Isosurfaces and Level-Sets for Volume Processing

Ross T. Whitaker

SCI Institute School of Computing

University of Utah

Introduction

Isosurfaces

- An Implicit Representation

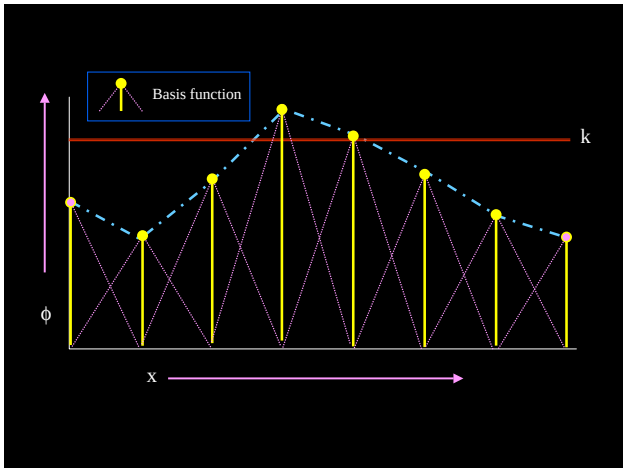
$$\phi : \begin{array}{c} U \\ x,y,z \end{array} \mapsto \begin{array}{c} \mathbb{R} \\ k \end{array}$$

- $U \subset \mathbb{R}^3$ — domain of the volume/range of surface model.
- A surface S is a set of ponts

$$S = \{\bar{x} | \phi(\bar{x}) = k\}$$

Local Basis Functions

A Volume-Based Representation (1D)



Introduction

Motivation

- Framework for Volume Processing/Filtering
  - Level-set methods for nonlinear image/volume filtering
  - Denoising & Reconstruction
- Surface-Modeling Technology
  - Many degrees of freedom
  - Topologically flexible
  - Limits resolution rather than shape

How Do We Represent ϕ

- Linear combination of global basis functions
  - "Blobby" models — *Blinn 1982*
  - Deformation by modifying size, position, number, etc. — *Muraki 1991*
- Linear combination of local local basis functions
  - Can define local deformations in terms of nearest neighbors
  - Many degrees of freedom
  - Well-defined relationship between surface movements and voxel values

Isosurface Visualization

- Triangulation (mesh extraction)
  - E.g. *marching cubes* (*Lorensen et. al. 87*)
  - Other related approaches
- "Direct" visualization
  - E.g. volume rendering
- Note: not the domain of this talk

Surface Normals

- Exists for every point in D.

$$\bar{n}(\bar{x}) = \frac{\nabla\phi(\bar{x})}{|\nabla\phi(\bar{x})|} \quad \text{where} \quad \bar{x} \in D.$$

- Gives normal to level set passing through that point
- Covention — inside or out (be consistent)
- How to compute? (e.g. central differences)

Local Basis Functions

- Geometry defined by *local* operations
- Continuous mathematics on ϕ
- Voxel manipulations determined by well-defined numerical methods

The Geometry of Isosurfaces

- Surface normals
- Curvature
- Goal: express the geometry of the isosurface in terms of derivatives of ϕ

Differential Geometry

Things to Remember

- Surface shape is locally quadratic
- Principal directions — $\bar{e}_1, \bar{e}_2$
- Principal curvatures — k_1, k_1
- Mean curvature — $H = \frac{k_1+k_2}{2}$
- Gaussian curvature — $K = k_1 \times k_2$
- Total curvature — $D^2 = k_1^2 + k_2^2 = 4H^2 - 2K$
- Note: $k_{1,2} = H \pm \sqrt{H^2 - K}$

Second-Order Structure of The Level Sets

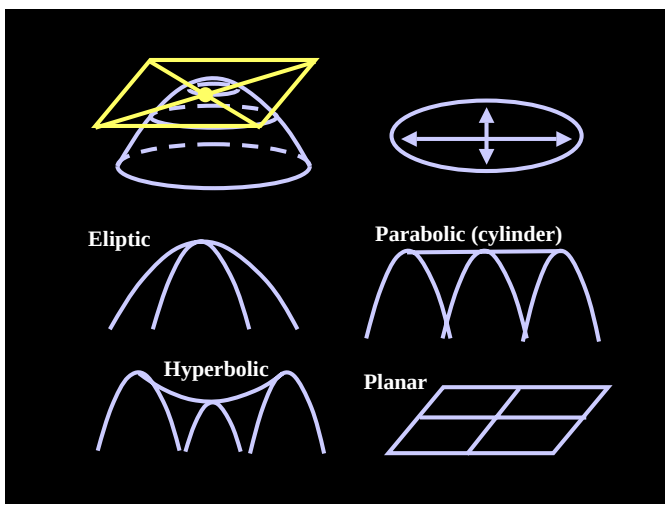
Mean Curvature

$$H(\bar{x}) = \frac{k_1 + k_2}{2} = \frac{1}{2}\mathrm{Tr}(\mathrm{N}) = \frac{1}{2}\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}$$

$$H(\bar{x}) = \frac{\begin{array}{c}\phi_x^2(\phi_{yy}+\phi_{zz}) + \phi_y^2(\phi_{xx}+\phi_{zz}) + \\ \phi_z^2(\phi_{xx}+\phi_{yy}) - 2\phi_x\phi_y\phi_{xy} - \\ 2\phi_x\phi_z\phi_{xz} - 2\phi_y\phi_z\phi_{yz}\end{array}}{2(\phi_x^2+\phi_y^2+\phi_z^2)^{3/2}}$$

Second-Order Structure

A Short Course In Differential Geometry



Second-Order Structure of The Level Sets

- Gradient of the normal $\longrightarrow$ matrix:

$$N = [\bar{n}_x \;\; \bar{n}_y \;\; \bar{n}_z]$$

- Second-order geometry of level set $\longrightarrow$ invariants of N
- Eigenvalues:
  - $e_1 = k_1$
  - $e_2 = k_2$
  - $e_3 = 0$

Second-Order Structure of The Level Sets

Gaussian Curvature

$$K(\bar{x}) = k_1 k_2 = e_1 e_2 + e_1 e_3 + e_2 e_3 = 2H^2(\bar{x}) - \frac{1}{2} D^2(\bar{x})$$

$$K(\bar{x}) = \frac{\begin{array}{c} \phi_z^2(\phi_{xx}\phi_{yy} - \phi_{xy}\phi_{xy}) + \phi_y^2(\phi_{xx}\phi_{zz} - \phi_{xz}\phi_{xz}) \\ +\phi_x^2(\phi_{yy}\phi_{zz} - \phi_{yz}\phi_{yz}) + 2\,[\phi_x\phi_y(\phi_{xz}\phi_{yz} - \phi_{xy}\phi_{zz}) \\ + \phi_x\phi_z(\phi_{xy}\phi_{yz} - \phi_{xz}\phi_{yy}) + \phi_y\phi_z(\phi_{xy}\phi_{xz} - \phi_{yz}\phi_{xx})] \end{array}}{(\phi_x^2 + \phi_y^2 + \phi_z^2)^2}$$

Deformable Models

- Basic terminology/mathematics
- Deformation processes (i.e. types)
- Level-Set Surface Models

Second-Order Structure of The Level Sets

Total Curvature

$$D^2(\bar{x}) = k_1^2 + k_2^2 = ||N||$$

Example: Total Curvature of An Isosurface

Isosurface of greyscale volume

Total curvature (lighter is greater)

Deformable Models

Regular Surfaces

- Locally parametric and changing over time — $\bar{S} = \bar{S}(r,s,t)$
- Surface normals $\bar{n} = \bar{n}(r,s,t)$
- Parameterization is just a place holder
  - Geometry will be reexpressed without
  - It will be "forgotten" as the model deforms

Surface Deformation Examples

- Forcing function

$$\frac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x})$$

- Expand/Contract

$$\frac{\partial \bar{x}}{\partial t} = G(\bar{x})\bar{n}$$

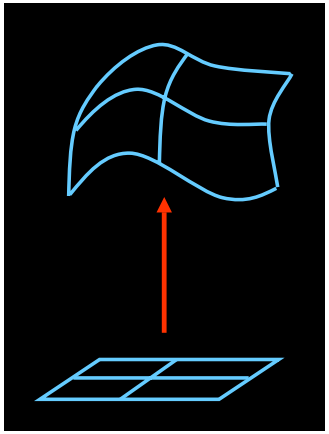
Deformable Models

Regular Surfaces

- Model only local deformation in terms of local surface structure
- Surface — $\mathscr{S} \subset \mathbb{R}^3$
- Local mapping

$$\bar{S} : \underset{r}{V} \times \underset{s}{V} \mapsto \underset{x,y,z}{\mathbb{R}^3},$$

where $V \times V \subset \mathbb{R}^2$



Deformable Surfaces

- Points on surface, $\bar{x} = \bar{S}(r,x,t)$, move in a prescribed manner
- Depends on position and local surface structure of $\bar{S}$.

$$\frac{\partial \bar{x}}{\partial t} = \bar{G}(\bar{x}, \bar{n}, \bar{S}_r, \bar{S}_s, \bar{S}_{rr}, \bar{S}_{rs}, \bar{S}_{ss}, \ldots),$$

- Note: independence from parameterization $\longrightarrow$ assume (r,s) is an orthnormal parameterization.

Level-Set Surface Deformation

Two approaches

- Static:

$$\phi(\bar{x}(t)) = k(t) \implies \nabla\phi(\bar{x}) \cdot \frac{\partial \bar{x}}{\partial t} = \frac{\mathrm{d}k(t)}{\mathrm{d}t}$$

  Eikonal Equation

- Dynamic:

$$\phi(\bar{x}(t), t) = k \implies \frac{\partial \phi}{\partial t} = -\nabla\phi \cdot \frac{\mathrm{d}\bar{x}}{\mathrm{d}t}$$

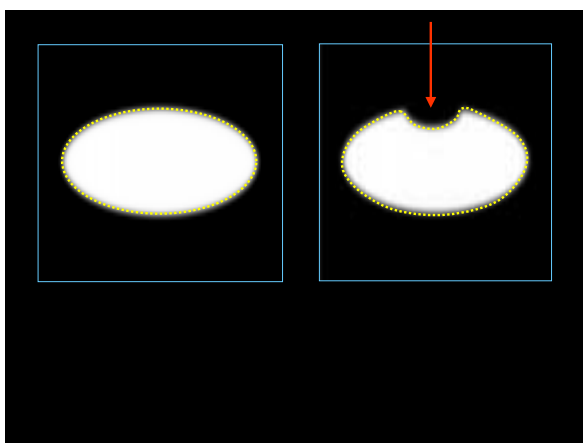
Sethian, 1999

Implications of this Strategy

- No memory (no internal structure)
- No specific topology (closed)
- Many degrees of freedom $\longrightarrow$ complex shapes
- Must solve 3D, nonlinear pde to deform
- Can/must specify the deformation pointwise

Level-Set Surface Deformation

- Represent a surface/contour implicit
- Change shape by modifying greyscale values



Level-Set Surface Models

Dynamic Approach

- Embed the surface motion in ϕ

$$\frac{\partial \phi}{\partial t} = -\nabla\phi \cdot \frac{\mathrm{d}\bar{x}}{\mathrm{d}t} = -\nabla\phi \cdot \bar{F}(\bar{x}, \mathrm{D}\phi, \mathrm{D}^2\phi, \ldots),$$

  where $D^n\phi$ are nth-order derivatives of ϕ

- Applies to *all* level sets of ϕ

Level-Set Surface Deformation Examples

- Forcing function

$$\frac{\partial \bar{x}}{\partial t} = \bar{F}(\bar{x}). \implies \phi_t = -\bar{F} \cdot \nabla \phi$$

- Expand/contract

$$\frac{\partial \bar{x}}{\partial t} = G(\bar{x})\bar{n} \implies \phi_t = G|\nabla \phi|$$

Finite Differences

- Time — forward differences

$$u^{n+1}_{i,j,k} = u^n_{i,j,k} + \Delta t \Delta u^n_{i,j,k},$$

- Space (to approxmiate $\Delta u^n_{i,j,k}$ — depends
  - Second-order terms — central differences
  - First-order terms — up-wind schemes

Deformation Modes

	Effect	Parametric Evolution	Level-Set Evolution	Parameter Assumptions
1	External force	$\bar{F}$	$\bar{F} \cdot \nabla \phi$	None
2	Expansion/ contraction	$G(\bar{x})\bar{N}$	$G(\bar{x})\,\lvert\nabla \phi(\bar{x},t)\rvert$	None
3	Mean curvature	$S_{rr} + S_{ss}$	$H\lvert\nabla \phi\rvert$	Orthonormal
4	Gauss curvature	$S_{rr} \times S_{ss}$	$K\lvert\nabla \phi\rvert$	Orthonormal
5	Second order	S_{rr} or S_{ss}	$\left(H \pm \sqrt{H^2 - K}\right)\lvert\nabla \phi\rvert$	Principal curvatures

Numerical Methods

- Finite differences
- Up-wind schemes
- Narrow-band and sparse-field methods
- Notation
  - $u^n_{i,j,k}$ — nth time step, i, j, k on grid
  - $\bar{x}_{i,j,k}$ — 3D position of i, j, k grid point
  - $u^n_{i,j,k}$ discrete sampling of $\phi(\bar{x}_{i,j,k}, t_n)$
  - assume w.o.l.g. grid spacing is 1 unit

Second-Order Operators

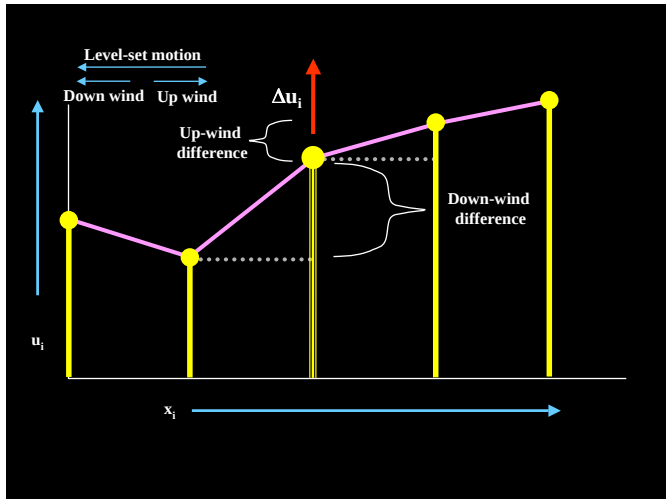
Tightest Fitting

$$\delta_{xx} u_{i,j,k} \triangleq \delta_x^{(-)} \delta_x^{(+)} u_{i+1,j,k} = u_{i+1,j,k} + u_{i-1,j,k} - 2u_{i,j,k}$$

$$\delta_{zz} u_{i,j,k} \triangleq \delta_z^{(-)} \delta_z^{(+)} u_{i+1,j,k} = u_{i,j,k+1} + u_{i,j,k-1} - 2u_{i,j,k}$$

$$\delta_{xy} u_{i,j,k} \triangleq \delta_x \delta_y u_{i,j,k}$$

Up-Wind Scheme



First-Order Operators

$$\delta_x^{(+)} u_{i,j,k} \triangleq (u_{i+1,j,k} - u_{i,j,k})$$

$$\delta_x^{(-)} u_{i,j,k} \triangleq (u_{i,j,k} - u_{i-1,j,k})$$

$$\delta_x u_{i,j,k} \triangleq \frac{1}{2}(u_{i+1,j,k} - u_{i-1,j,k})$$

Up-Wind Scheme

Osher and Sethian, 1988

- "Non-oscillatory" scheme
- Use *one-sided* difference operators in gradient of ϕ
- Avoid overshoot — create no new contours
- Note: choice of derivatives depends on data

Up-Wind Scheme

- Let
$$\bar{F}(\bar{x}) = (F^{(1)}(\bar{x}), F^{(2)}(\bar{x}), F^{(3)}(\bar{x}))$$

- Then
$$\bar{F}(\bar{x}_{i,j,k}) \cdot \nabla\phi(\bar{x}_{i,j,k}, t) \approx \sum_{q=1}^{2} F^{(q)}(\bar{x}_{i,j,k}) \begin{cases} \delta_q^+ u_{i,j,k}^n & \text{for } F^{(q)}(\bar{x}_{i,j,k}) > 0 \\ \delta_q^- u_{i,j,k}^n & \text{for } F^{(q)}(\bar{x}_{i,j,k}) < 0 \end{cases}$$

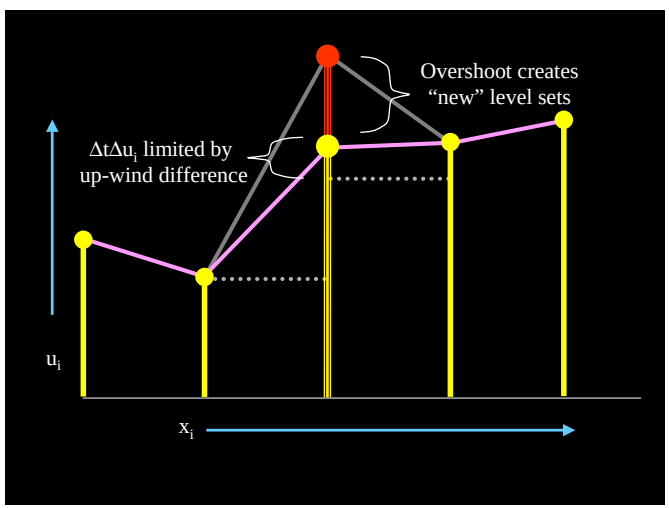
Time Steps

Limited For Stability

Up-wind scheme:
$$\Delta t_G \leq \frac{1}{3 \sup_{i,j,k \in X} \{|\nabla G(x_i, y_j, z_k)|\}}$$

Second-order terms:
$$\Delta t_H \leq \frac{1}{6}$$

Up-Wind Scheme



Second-Order Terms

Example: Mean Curvature

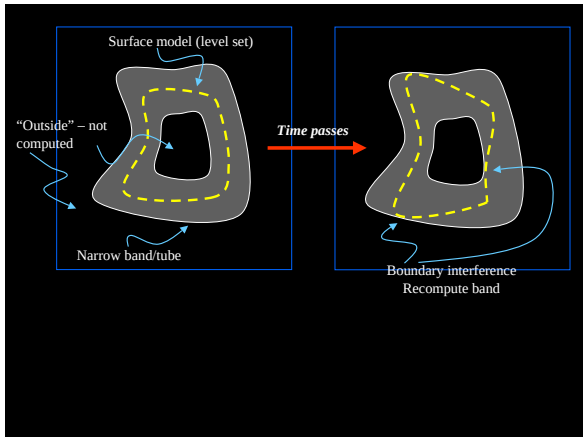
Central differences — plug and chug

$$H_{i,j,k}^n = \frac{\begin{array}{c}\left(\left(\delta_y u_{i,j,k}^n\right)^2 + \left(\delta_z u_{i,j,k}^n\right)^2\right)\delta_{xx} u_{i,j,k}^n \\ + \left(\left(\delta_z u_{i,j,k}^n\right)^2 + \left(\delta_x u_{i,j,k}^n\right)^2\right)\delta_{yy} u_{i,j,k}^n + \left(\left(\delta_x u_{i,j,k}^n\right)^2 + \left(\delta_y u_{i,j,k}^n\right)^2\right)\delta_{zz} u_{i,j,k}^n \\ -2\delta_x u_{i,j,k}^n \delta_y u_{i,j,k}^n \delta_{xy} u_{i,j,k}^n - 2\delta_y u_{i,j,k}^n \delta_z u_{i,j,k}^n \delta_{yz} u_{i,j,k}^n - 2\delta_z u_{i,j,k}^n \delta_x u_{i,j,k}^n \delta_{zx} u_{i,j,k}^n\end{array}}{\left(\delta_x u_{i,j,k}^n\right)^2 + \left(\delta_y u_{i,j,k}^n\right)^2 + \left(\delta_z u_{i,j,k}^n\right)^2}$$

Narrow-Band Algorithm

Adalsteinson and Sethian, 1995

Restablish the neighborhood (embedding) when necessary



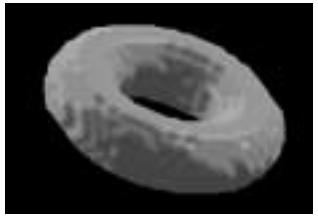
Sparse-Field Algorithm

Behavior

Fast: execution time grows with surface elements

Accurate: active set avoids shocks—sub voxel surface fitting

Fitting of surface model to toriodal data (dist. trans.)



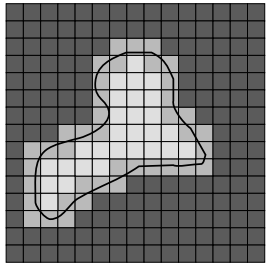
Conventional — Shocks

Sparse-Field

Narrow-Band Methods

Evolution of level set, $\phi(\bar{x},t) = k$, not impacted by choice of embedding

Perform calculations for surface evolution only in a neighborhood of the surface $\mathscr{S} = \{\bar{x} | \phi(\bar{x}) = k\}$

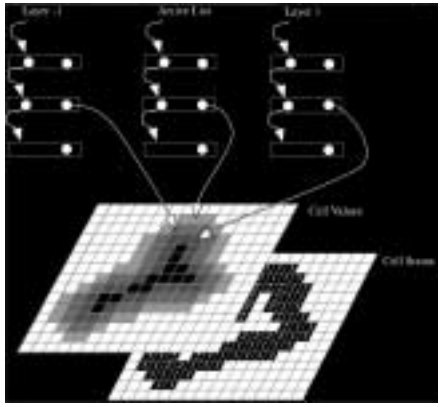


Sparse-Field Algorithm

Whitaker 1998

Maintain a thin band of pixels around an *active set*

Update active set and then consecutive layers



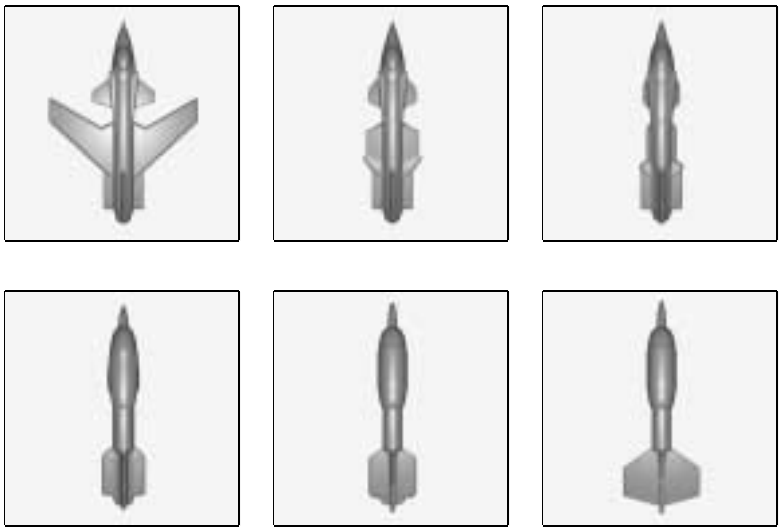
Surface Morphing

Whitaker and Breen, 1998; Breen and Whitaker 2000

Two parts of morphing:

1. Warping — coordinate transformation
2. Blending — "filling in" details ⟵ level-set surface models
   - Fourier domain — *Hughes 1992*
   - Linear interpolation — *Lerios et al. 1995*
   - Distance trans. — *Cohen-Or et al. 1998*

Example



Applications

- Surface morphing
- Filleting
- Antialiasing binary volumes
- Inverse problems (surface estimation)
- Other

Surface Morphing

Strategy

Expansion/contraction — depending on signed-distance transform to "target"

$$\frac{d\bar{x}}{dt} = \gamma_B(\bar{x}(t))\,\bar{n}(\bar{x}(t))\;\forall\,\bar{x}(t) \in \mathscr{S}_t$$

$$\Updownarrow$$

$$\frac{\partial\phi(\bar{x})}{\partial t} = |\nabla\phi(\bar{x})|\gamma_B(\bar{x})$$

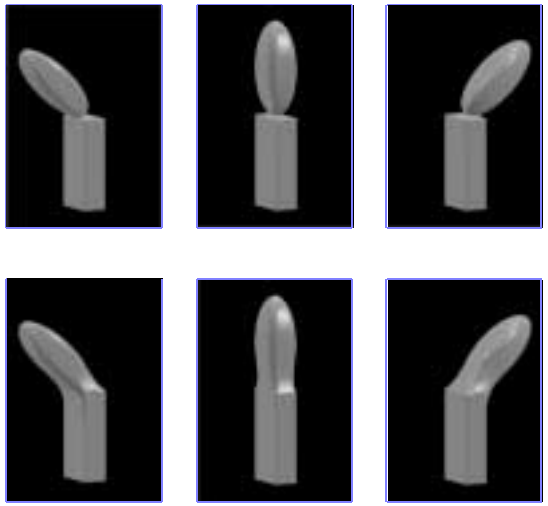
Filleting and Blending

Whitaker and Breen, 1998

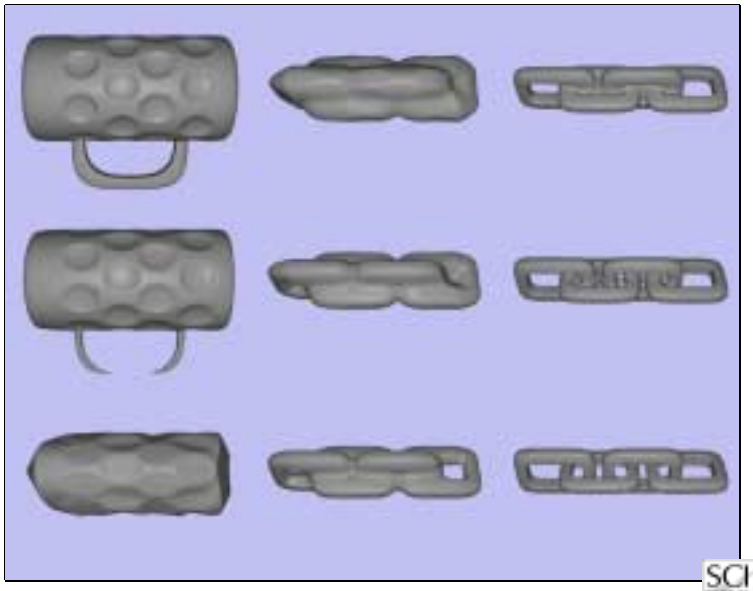
Modify mean curvature flow to “filling in” material

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \left[\max(k_1, 0) + \max(k_2, 0) \right]$$

Blending

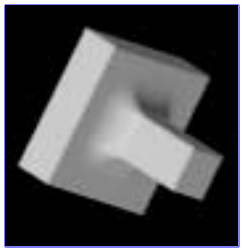


Example

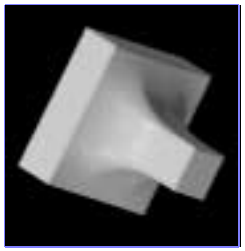


Filleting

Solid Models



t = 1.0



t = 4.0

Antialiasing Binary Volumes

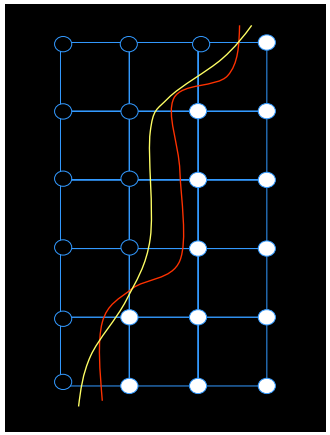
Whitaker, 2000

Treat the binary volume problem as an undercontrained *estimation problem*
Solution must:

- Enclose all of the inside points
- Enclose none of the outside points

Resolve ambiguity by imposing some other condition

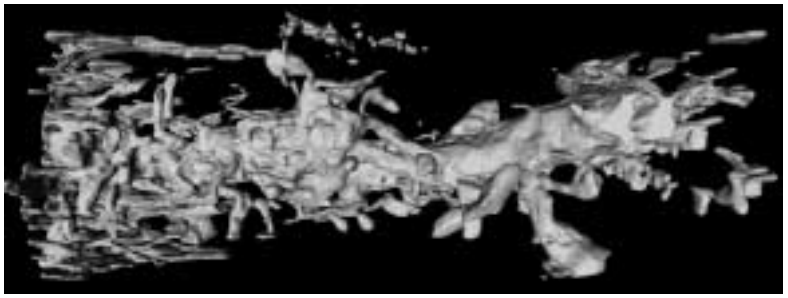
- Find a "preferred" solution
- E.g. minimal surface area



Inverse Problems

Estimating Surfaces In Ill-Posed Settings

- Santosa, 1996 — Image deblurring
- Dorn *et al.*, 2000 — Electromagnetic tomography
- Elangovan & Whitaker, 2001 — Tomography



Tranmission Electron Tomography, Data Courtesy of UC San Diego NCMIR

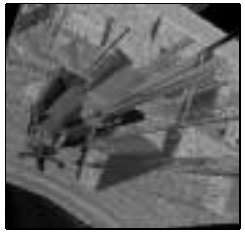
3D Reconstruction From Range Data

Whitaker, 1998

Estimate 3D surfaces from noisy range measurements taken from different points of view

A range image

Amplitude data

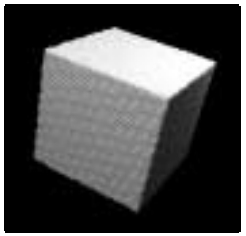


Surface rendering

Antialiasing Binary Volumes

Example

"Ideal" volume

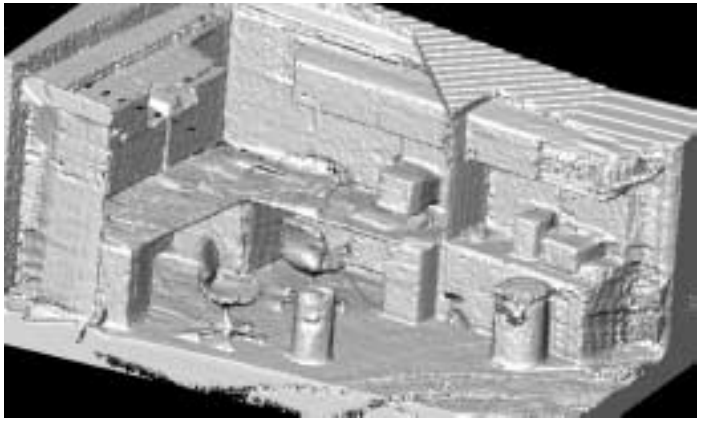


Binary volume

Surface Estimate

Initial Model

No prior
No deformation/fitting



Other Applications

- 2D/3D image segmentation *(Whitaker, 95; Casselles et al. 95; Malladi et al., 95; etc.)*
- Image/volume filtering *(Osher & Rudin, 90; Alvarez & Morel, 1994; Malladi & Sethian, 96; etc.)*
- Image blending *(Whitaker, 00)*
- Computer vision *(Kimia & Zucker, 92, 94)*
- Physical simulation *(Osher & Fedkiw, 00)*
- See also: Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999

Model Fitting To Range Data

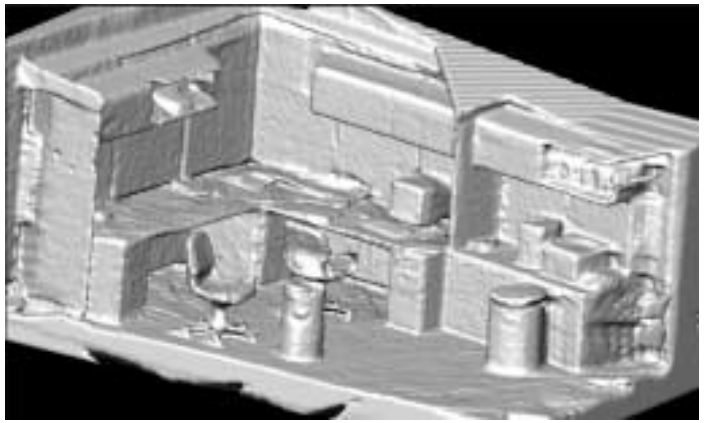
Combine *data term* with prior (surface area penalty)

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| G(\bar{x}) + H(\bar{x})$$

The data term, $G(\bar{x})$, includes inference from a *set* of registered range maps

Final Estimate

Prior
Deformation/fitting



VISPack

Volume-Image-Surface Package

Capabilities

- Image processing
- Volume processing
- Level-set surface modeling

Design

- *C++* — object oriented
- Data handles/copy on write
- Templates
- Functional interface — operator overloading
- File I/O (extensible) — no GUI

Level-Set Surface Modeling

$$\frac{\partial \phi}{\partial t} = \alpha \bar{F}(\bar{x}, \nabla\phi) \cdot \nabla\phi + \beta G(\bar{x}, \nabla\phi)|\nabla\phi| + \gamma|\nabla\phi| + \eta E(\mathrm{D}\phi, \mathrm{D}^2\phi)$$

Software

How Can I Get Started?

- VISPack
- NLM Insight Project

Operator Overloading

3D Edge Detector

```
Volume<float> dx, dy, dz;
Volume<float> vol_gauss = vol.gauss(0.5);
Volume<float> vol_out
   = (((dx = vol_gauss.dx()).power(2)*vol_gauss.dx(2)
   + ((dy = vol_gauss.dy()).power(2)*vol_gauss.dy(2)
   + ((dz = vol_gauss.dz()).power(2)*vol_gauss.dz(2)
   + dx*dy*(dx).dy() + dx*dz*(dx).dz())
   + dy*dz*(dy).dz()) )).zeroCrossings()
     && ((dx.power(2) + dy.power(2)) > T*T));
```

Example: Shape Morphing Object

Subclass Of *VoxModel*

1. *Source* and *target* volumes
2. Inialize to *source*
3. Define virtual method:
```
grow(float x, float y, float z,
    float nx, float ny, float nz)
```
4. call `iterate()` and save volumes/models at regular intervals

Conclusions

- Isosurface — representing and *manipulating* solid shapes
- Level set surface models — wide range of applications for processing volumes and surfaces
- Analytical — surface motion $\longrightarrow$ pde's on volume
- Lots of potential
  - Applications
  - New technologies

VoxModel Object

1. Creates a model from a volume
2. Calculates $\Delta u^n_{i,j,k}$, Δt, etc.
3. Virtual functions (subclasses) define $\bar{F}$, G
4. Parameters (α, β, γ, η) set by the subclass
5. Performs an update on the values of $u^n_{i,j,k}$
6. Maintains active set and updates the *layers*
7. Provides access to $u^n_{i,j,k}$ and list of active grid points.

NIH Insight Project

Level Sets (Among Other Things)

1. Open source, C++, object oriented library (API)
2. Dataflow architecture
3. Parallelism and Streaming
4. Generic framework for pde's
5. Level-set surface modeling capability
6. Release Jan. 2002
7. http://public.kitware.com/Insight/Web/index.htm