Level Set Models for Computer Graphics

David E. Breen
Department of Computer Science
Drexel University
Philadelphia, PA   USA

Overview

• Level Set Model Introduction

APPLICATIONS

• 3D Volumetric Metamorphosis
  – Feature-based User Controls
• Contour-based Surface Reconstruction
• Surface Editing
  – CSG Modeling
  – Handle-based
  – Sketch-based
  – Multiresolution

What is a Level Set Model?

• A numerical technique for tracking interfaces

• Level-set models are defined as an iso-surface, i.e. a level set, of a dynamic implicit function $\phi$

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

• $k \in \mathbb{R}$ is the iso-value
• $\bar{x} \in \mathbb{R}^3$ is a point in space on the iso-surface
• $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ is a scalar function

What is a Level Set Model?

• A deformable implicit model, $\Phi(X, t) = 0$
• Sampled representation of dimension $n+1$
  – Images (2D) represent curves
  – Volumes (3D) represent surfaces
• Change level set by modifying samples
• Change sample values by solving a Partial Differential Equation (PDE)

This is a Level Set Model!

of a curve!
Changing the image moves the curve

Red curve is defined by $\phi(x,y) = 127$

What is a Level Set Model?

- $\phi(x)$ is not defined by a specific equation
- $\phi(x)$ is represented by a regular 3D sampling
  - Signed distance volume dataset
- Level Set model is deformed by evolving the Level Set equation on the sampling
  - Osher & Sethian 1988
  $$\frac{\partial \phi}{\partial t} = -\nabla \phi F(\rho(x), \partial \phi, \partial^2 \phi, ...)$$
- Connects changes in sample values to changes of the level set curve/surface

The Speed Function

$$\frac{\partial \phi}{\partial t} = -\nabla \phi F(\rho(x), \partial \phi, \partial^2 \phi, ...)$$

- LS model deformation is controlled by $F()$
- $F(x,...)$ defines the speed of the LS surface in direction of the surface normal $N$ at each point $x$ on the surface
- $F()$ is defined by a user for each application in order to achieve a specific goal
- We have defined speed functions for several computer graphics applications

Advantages of LS Models

- Always produce closed, non-self-intersecting (simple) surfaces

Deforming Mesh May Intersect

No Self-Intersection with Level Set Deformations
Important for GCAD, Digital Manufacturing and Analysis

Figurine automatically manufactured from a level set model

Advantages of LS Models

• Easily change topological genus
  – Holes may close or open
  – Separate pieces come together/split apart
• Ideal for complex deformable models of unknown and/or changing genus

Mug-to-Chain Morph

Level Set Segmentation

Advantages of LS Models

• Concise interface for control → F( )
• Free of mesh connectivity and quality issues
• No need to reparameterize or remesh during deformation

Disadvantages of LS Models

• No inherent parameterization (?)
  – Pedersen 1995, mesh parameterization methods
• Computationally expensive (?)
  – Narrow Band methods are O(surface area)!
• High memory requirements (?)
• Cannot represent fine or sharp features (?)
• Cannot control genus (?)
3D Volumetric Metamorphosis

Advantages of Level Set Morphing

- Morphing objects can change genus and number of components
- No restrictions on shape or mesh structure of morphing objects
  - as long as they are closed
  - if you can convert your object into a level set, you can morph it
- Guaranteed convergence
  - as long as objects spatially overlap

Level Set Morphing Movements

- Each point on surface moves in the direction of local normal. Step-size proportional to signed distance to target $\gamma_B$

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

- Regions outside contract
- Regions inside expand
- Guaranteed convergence
Level Set Morphing

Movements

- Each point on surface moves in the direction of local normal. Step-size proportional to signed distance to target $g_B$.
- Regions outside contract
- Regions inside expand
- Guaranteed convergence
- Not moving points!

Different Alignments Produce Different Morphs

Dart-to-Jet Morph

Morphing Challenge

- How to morph between these three models?

Morphing Between Different Types of Models

- Combining
  - A variety of scan conversion algorithms
  - A flexible metamorphosis (morphing) technique based on level set models
- Produces
  - A technique for morphing between different types of geometric models
1 Minute of Fame

- Tar Monster Morphing Sequence
  Scooby-Doo 2, 2004

But Specifying Object Overlap Is Insufficient For Control

- Morphing between a horse and a camel

Solution: Add Feature Correspondences

- User identifies features/regions on the source that should morph into specified features/regions on the target

Horse-to-Camel Morph: Much Better!
Horse-to-Camel Morph: An Added (Free!) Bonus

Change of topology! Front knees join.

How to Incorporate Feature Correspondences?

- User provides a target and source models, and their corresponding features

How to Incorporate Feature Correspondences?

- Extract a mesh from the level set model
- Incrementally apply the warp to the mesh

How to Incorporate Feature Correspondences?

- Correlations define a warp that deforms the source close to the shape of the target
- Morph Step: transfers surface details
- Warp Step: large-scale deformation

How to Incorporate Feature Correspondences?

- Apply the full warp defined by the correspondences and do the morph

Applying incremental warp just to the source

Combining These Two Steps Produces The Desired Result

Warped Morph Only

Morph + Mesh Warp
Combing These Two Steps Produces The Desired Result

Don't Care About the Genus or Number of Components!

Timings of Morph and Warp Steps Can Be Independently Adjusted

Two Retimed Morphs

Contour-based Surface Reconstruction

Problem Statement

Create a smooth surface from parallel contours
Problem Statement
Create a smooth surface from parallel contours

Input Data
- Frequently it is easier to segment 3D datasets one slice (2D image) at a time
- The 2D segmentations need to be combined into a 3D object

Level Set Approach
- Sweep out the surface that connects the contours
  - Use 2D level set morphing
  - Map time into Z
  - Constrain the speed of the sweeping motion at contours
Level Set Approach

• Sweep out the surface that connects the contours
  – Use 2D level set morphing
  – Map time into Z
  – Constrain the speed of the sweeping motion at contours
  – Use a combination of Lagrangian and Eulerian techniques

3D Reconstruction as a 2D Morphing Process

Mouse Embryo Result

Pelvis Result

Level Set Surface Editing

CSG Modeling

Do Level Sets Have Something to Offer Geometric Modeling?

• A representation that
  – easily changes genus
  – is manufacturable after every edit operation
• More importantly need to tweak output from simulations and special effects
A Biomedical Application

Repairing a thresholded medical 3D scan

Initial Level Set Editing

- CSG operations
- Automatic blending
- Local/Global smoothing
- Embossing

Level-Set Editing Framework

Input Data

- Scanned Volumes
- Distance Calculation
- Volume Rendering

Operations

- 3D Scan Conversion
- Paste CSG Union
- Level Set Blending
- Level Set Point Attraction
- Embossing

Initial Level Set Blending

- CSG operations
- Automatic blending
- Local/Global smoothing
- Embossing

Speed Function Building Blocks

\[
\frac{\partial \phi}{\partial t} = -|\nabla \phi| F(\phi, \ldots) \quad F = D(x)(\gamma y) F(y)
\]

- Distance-based cut-off function
- Geometric property filter function
- Provides user control of local geometric properties
- Function of geometric measure

Global Smoothing with a Morphological Opening

Laser scan reconstruction with spiky errors
Global Smoothing with a Morphological Opening

- Offset surface inwards

Global Smoothing with a Morphological Opening

- Offset surface outwards to create smoothed surface

Creating The Dragon

- Original dragon superellipsoid Cut (CSG difference)
- Mirror & position
- Paste & LS blending
- Affine transformation
- Level set blending
- Copy (CSG intersection)
- Affine transformation
- Level set blending
- Mirror & position
- Position supertoroid
- Paste (CSG union)
- Level set blending
- Final new dragon

Important for GCAD, Digital Manufacturing and Analysis

- Figurine automatically manufactured from a level set model

Repairing a Damaged Bust

- Nose copied from another model, then blended
- Left chin copied from right chin, then blended
- Hair sharpened (~ smoothing)

Level Set Surface Editing

- Handle-based Operators
Pulling a point, symmetric ROI

- Click on a point on the surface (x₀) and drag it
- Radius of influence (r) around x₀
- Surface moves within sphere around x₀
- Simple yet useful for designing and creating handle-like structures

\[ F(x) = \begin{cases} \cos^{\alpha/2} \cdot d(x) / r & d(x) \leq r \\ 0 & d(x) > r \end{cases} \]

- \( x \): point on the surface being evaluated
- \( d(x) \): geodesic distance to \( x \) from \( x₀ \)
- \( \alpha \): shape parameter

Pulling on a point, arbitrary ROI

- A closed boundary curve on the surface designates an ROI
- Click and drag within the ROI
- All points in the ROI move outward as a function of the geodesic distance to \( x₀ \)
- Speed gradually decreases to zero at the boundary curve.
- Stop when \( x₀ \) reaches cursor or the user releases the mouse button

\[ F(x) = f(d_{\text{ROI}}(x)) = \left( \frac{\max(d_{\text{ROI}}(x)^2) - d_{\text{ROI}}(x)^2}{\max(d_{\text{ROI}}(x)^2)} \right)^{\alpha} \]

- \( d_{\text{ROI}}(x) \): geodesic distance from \( x \)
- \( \alpha \): shape parameter

Pulling a curve on the surface, symmetric ROI

- Draw an open curve on the surface
- Click on a point on the curve and drag the point
- ROI is symmetric around the curve
- Speed decreases with the distance from the curve

\[ F(x) = \begin{cases} 0 & d_{\text{ROI}}(x) > r \\ 1.0 - \frac{d_{\text{ROI}}(x)}{r} & d_{\text{ROI}}(x) \leq r \end{cases} \]

- \( d_{\text{ROI}}(x) \): distance to curve C₂
- \( r \): thickness of the ROI

Pulling on a curve on the surface, arbitrary ROI

- Closed curve (C₁) on the surface define an ROI
- Curve (C₂) on the surface used as a handle.
- Clicking and dragging the handle deforms the surface

\[ F(x) = f(d_{\text{ROI}}(x)) \left( \frac{\max(d_{\text{ROI}}(x)^2) - d_{\text{ROI}}(x)^2}{\max(d_{\text{ROI}}(x)^2)} \right)^{\alpha} \]

- \( d_{\text{ROI}}(x) \): geodesic distance from \( x \)
- \( \alpha \): shape parameter

Changing alpha

- Superellipsoid-shaped tool
- Add or subtract features
- Adjustable tool size
- Can be used like an “eraser”

\[ F(x) = \begin{cases} 0 & f_{\text{ROI}}(V) > 0 \\ f_{\text{ROI}}(V) & f_{\text{ROI}}(V) \leq 0 \end{cases} \]

- \( x₀ \): tool center
- \( f_{\text{ROI}}(V) \): implicit superellipsoid equation
- \( \beta > 1 \): drives the surface outward
- \( \beta < 1 \): drives the surface inward

Surface Detailing/Carving

- Superellipsoid-shaped tool
- Add or subtract features
- Adjustable tool size
- Can be used like an “eraser”
Interactive Smoothing

- Click & drag tool over the area that needs smoothing
- Curvature-based speed function
- Fixed radius of influence
- Adjustable tool size and intensity

\[ F(x) = \gamma \cdot g(d(x)) \cdot s(x) \]

- \( g(d) = \begin{cases} 
1.0 & d - r - \epsilon \\
1/2 + 1/2 \cdot \cos(c \cdot (d - r + \epsilon)) & r - \epsilon < d \leq r \\
0.0 & d > r
\end{cases} \]

- \( \gamma \): constant that controls the amount of smoothing
- \( s \): mean curvature
- \( r \): radius of influence

Level Set Surface Editing

Sketch-based Operators

Single Cross-section Curve

\[ F(x) = \frac{d_{ds}(x)}{\max(d_{ds}(x))} \cdot f(d_{ds}(x)) \cdot \frac{\max(d_{ds}(x)) - d_{ds}(x)}{\max(d_{ds}(x))} \]

\[ f(d) = \begin{cases} 
1.0 & d > s \\
(d/s)^2 & d \leq s
\end{cases} \]

- \( d_{ds}(x) \): closest point on \( C_s \)
- \( x \) is corresponding point on \( C_d \)
- \( d_{ds}(x) \): geodesic distance to \( C_s \) from \( x \)
- \( d_{ds}(x) \): geodesic distance to \( B \) from \( x \)
- Max functions taken over all points in the ROI

Multiple Cross-section Curves

- The speed function is a combination of speed functions calculated per curve

\[ F(x) = \sum a_c(x) \cdot f(d) \]

- \( N \) is the number of cross-section curves
- \( a(x) \) is the weight function that determines the effect of curve \( c \) on point \( x \)
- What is \( a(x) \)?
  - \( x \) is closest point on \( C_s \)
  - \( d_{ds}(x) \): closest point on \( C_d \)
  - Results with \( a(x) \cdot x \) shown below
  - \( a(x) \) decreases linearly/exponentially/linearly?

Blending Functions

<table>
<thead>
<tr>
<th>Method Criteria</th>
<th>Fits to curve</th>
<th>Smooth Transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1(x) )</td>
<td>✔️ ✔️ ✔️</td>
<td>✔️ ✔️</td>
</tr>
<tr>
<td>( \alpha_2(x) )</td>
<td>✔️ ✔️ ✔️</td>
<td>✔️ ✔️</td>
</tr>
<tr>
<td>( \alpha_3(x) )</td>
<td>✔️ ✔️ ✔️</td>
<td>✔️ ✔️</td>
</tr>
<tr>
<td>( \alpha_4(x) )</td>
<td>✔️ ✔️ ✔️</td>
<td>✔️ ✔️</td>
</tr>
</tbody>
</table>

- \( x \) is closest point on \( C_s \)
- Distance between curves
- Table shows criteria based on \( a(x) \) and \( d_{ds}(x) \)
Multiple Curve Handles

- All cross-section curves initially drawn on the surface
- Approximate 3D intersection points—closest point algorithm
- Curves stay connected
- Curve changes are propagated to intersecting curves via intersection points
- Use multiple curve speed functions to evolve surface

Localized Catmull-Rom Splines

- We have developed techniques that expand and generalize the result of modifying one C-R control point
  - Provides greater flexibility, control and expressiveness
  - Allows the user to define the range and type of influence that manipulation of a single control point may produce on a C-R curve
  - Provides a versatile and powerful localized, multiresolution curve editing capability

Global Deformations with Multiple Curves

- Create a 3D outline sketch of a model
- Initialize a simple level-set model inside (example: sphere)
- Deform the model until it reaches the sketch curves

\[ F_c(x) = \frac{d_{up} (x)}{\max (d_{up} (x))} \frac{\max (d_{in} (x)) - d_{up} (x)}{\max (d_{in} (x))} F(x) = \sum \alpha_c (x) \cdot F_c (x) \]

General Framework

1. User input is translated into level-set speed functions
2. The level-set PDE is solved on a portion of the narrow band by the VISPACK library
3. The resulting edited model is displayed in the UI

VISPACK Narrow-Band Data Structures

- A set of doubly linked lists point into 3D array
- The order of voxels within each list is arbitrary
- Requires that level-set PDE be solved over the whole surface

Voxel Storage

- Voxels stored in a 3D spatial hash table
- Many advantages
  - Sparse
  - Constant time random access
  - “Out-of-the-box” computing
- Disadvantage
  - No info away from the narrow band
Results: Handle-Based

Creating a bear

Results: Sketch-Based

Creating a duck and a shark

Results: Sketch-Based

Creating a shamrock and a bear

Results: Performance

FPS inversely proportional to # voxels in ROI

<table>
<thead>
<tr>
<th>Model</th>
<th>Resolution</th>
<th>fps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shark</td>
<td>85 x 99 x 54</td>
<td>66-90</td>
</tr>
<tr>
<td>Duck</td>
<td>79 x 67 x 37</td>
<td>66-90</td>
</tr>
<tr>
<td>Shamrock</td>
<td>45 x 50 x 35</td>
<td>83</td>
</tr>
</tbody>
</table>
**High-Resolution Models Using Spatial Hash Tables**

Dimensions:
- Dense volume: 944 × 2048 × 1709
- Sparse volume: 3,304,030,208 voxels

**Results**

High-Resolution Models

Dimensions:
- Dense volume: 654 × 600 × 1794
- Sparse volume: 703,965,600 voxels

**Open level set!**

Level Set Surface Editing

Multiresolution Operators

- Smoothing & Differencing
- Details stored in particles
- Binomial filtering for smoothing
- LS evolution restores details
- Speed function based on info from particles

Hierarchical LS Model
LS Multiresolution Modeling

- Changes cascaded post edit
- Local narrow band
- Fast Marching Methods
- Blended using CRS

Multires Results

- Editing a model at different resolutions

Geometric Texture Transfer

- Details particles can be transferred from one model to another to add texture
- “Erase” capital

Geometric Texture Transfer

- Copy details particles from back of head
Geometric Texture Transfer

- Paste particles on top of head and add details with an LS evolution

Summary

- Level Set Model Introduction

APPLICATIONS

- 3D Metamorphosis
- Contour-based Surface Reconstruction
- Surface Editing

Papers


Acknowledgements

- Collaborators: Al Barr, Cindy Grimm, Ken Museth, Ross Whitaker
- Students: Rafael Campos, Manolya Eyiyurekli, Jia Mao, Sean Mauch, Ola Nilsson
- This research was supported by the NSF STC for Graphics and Visualization, NSF grants ACI-99-82273, ACI-00-89915 and CCF-070244, and CAPES (Brazil) grant 88888.075221/2012-00.