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
Level Set Models
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 \( \phi \)

\[
S = \{ \vec{x} \mid \phi(\vec{x}, t) = k \}
\]

- \( k \in \mathbb{R} \) is the iso-value
- \( \vec{x} \in \mathbb{R}^3 \) is a point in space on the iso-surface
- \( \varphi : \mathbb{R}^3 \rightarrow \mathbb{R} \) 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
    \[
    \frac{\partial \phi}{\partial t} = -|\nabla \phi| F(\vec{x}, \partial \phi, \partial^2 \phi, ...)
    \]
- Connects changes in sample values to changes of the level set curve/surface
The Speed Function

\[
\frac{\partial \phi}{\partial t} = -|\nabla \phi|F(\ddot{x}, \partial \phi, \partial^2 \phi, \ldots)
\]

- LS model deformation is controlled by \( F(\ ) \)
- \( F(X, \ldots) \) 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
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 $\rightarrow F(\ )$
- 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 $O(\text{surface area})$
- High memory requirements (?)
- Cannot represent fine or sharp features (?)
- Cannot control genus (?)
  - Han et al. 2001, Bischoff & Kobbelt 2003, Ségonne et al. 2005
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 $\gamma_B$
Each point on surface moves in the direction of local normal. Step-size proportional to signed distance to target \( \gamma_B \)

\[
\frac{\partial \phi(\bar{X})}{\partial t} = |\nabla \phi(\bar{X})| \gamma_B(\bar{X})
\]
Each point on surface moves in the direction of local normal. Step-size proportional to signed distance to target $\gamma_B$.

$$\frac{\partial \phi(\vec{X})}{\partial t} = |\nabla \phi(\vec{X})| \gamma_B(\vec{X})$$

$F(\vec{X}) = \gamma_B(\vec{X})$
Level Set Morphing Movements

- Each point on surface moves in the direction of local normal. Step-size proportional to signed distance to target $\gamma_B$
  \[ \frac{\partial \phi(X)}{\partial t} = |\nabla \phi(X)| \gamma_B(X) \]
- 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 $\gamma_B$.

$$\frac{\partial \phi(\vec{X})}{\partial t} = |\nabla \phi(\vec{X})| \gamma_B(\vec{X})$$

• Regions outside contract.
• Regions inside expand.
• Guaranteed convergence.
• Not moving points!
Different Alignments Produce Different Morphs
Dart-to-Jet Morph
Dart-to-Jet Morph
Morphing Challenge

- How to morph between these three models?

Polygonal Mesh

CSG Model

MRI Scan
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
Not so good! 😞
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?

- Correspondences 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?

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

- Human Face to Stormtrooper - Scene Setup
  - 26 features
  - 100 frames
  - Morph timestep = 0.160
  - Warp step = linear

R. Campos (Drexel)
How to Incorporate Feature Correspondences?

- Apply the full warp defined by the correspondences and do the morph.
How to Incorporate Feature Correspondences?

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

Partial Warp: incrementally applied for $u = 0 .. 1$

Warp Step ($u$)

Mesh Extraction

Incremental Warp Transform

Warp Step
Combing 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!

Subtracted random spatial noise from the source
Timings of Morph and Warp Steps Can Be Independently Adjusted

- \( dt(\text{Frame } \#) \)
- \( u(\text{Frame } \#) \)
- Default timing curves
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
- Use a combination of Lagrangian and Eulerian techniques
Level Set Approach

• Sweep out the surface that connects the contours
  – Use 2D level set morphing
Level Set Approach

- Sweep out the surface that connects the contours
  - Use 2D level set morphing
  - Map time into Z
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

Surface Reconstruction Via Contour Metamorphosis: An Eulerian Approach With Lagrangian Particle Tracking

Ola Nilsson  David Breen  Ken Museth
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
$$\frac{\partial \phi}{\partial t} = -\|\vec{\nabla} \phi\| F(\bar{x}, \phi, \ldots)$$

\[ F = D_q(d) G(\gamma) F(\gamma) \]
Level-Set Blending

Position

Paste (CSG Union)

Blend

Distance

Curvature

Cut-off func.

Filter func.

\[ F_{blend} = D_c(d)G(K)\alpha K \]
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
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 \((x_s)\) and drag it
- Radius of influence \((r)\) around \(x_s\)
- Surface moves within sphere around \(x_s\)
- Simple yet useful for designing and creating handle-like structures

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

- \(x\) : point on the surface being evaluated
- \(d_s(x)\) = geodesic distance to \(x_s\) from \(x\)
- \(\alpha\) : shape parameter
Pulling on a point, arbitrary ROI

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

$$F(x) = f(d_{out}(x)) \cdot \left( \frac{\max(d_{in}^{x_s}(x)) - d_{in}^{x_s}(x)}{\max(d_{in}^{x_s}(x))} \right)^\alpha$$

$$f(d) = \begin{cases} 
1/2 - 1/2 \times \cos(\pi \times d(x)/\epsilon) & d \leq \epsilon \\
1.0 & d > \epsilon 
\end{cases}$$

- $d_{in}(x)$: geodesic distance to $x_s$ from $x$
- $\max(d_{in}(x))$: maximum of these values over all points in the ROI
- $d_{out}(x)$: geodesic distance to the boundary curve from $x$
- $f(d)$: ROI transition function
- $\alpha$: 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

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

- \( r \): thickness of the ROI
- \( d_{in}(x) \): geodesic distance between \( x \) and the curve
Pulling on a curve on the surface, arbitrary ROI

- Closed curve ($C_1$) on the surface define an ROI
- Curve ($C_2$) on the surface used as a handle.
- Clicking and dragging the handle deforms the surface

\[ F(x) = f(d_{out}(x)) \times \left( \frac{\max(d_{in}^{cs}(x)) - d_{in}^{cs}(x)}{\max(d_{in}^{cs}(x))} \right)^{\alpha} \]

- $d_{in}(x)$: geodesic distance to curve $C_2$
- $d_{out}(x)$: geodesic distance to $C_1$ from $x$
- $\max(d_{in}(x))$: maximum over all points in the ROI
- $f(d)$: ROI transition function
- $\alpha$: shape parameter
Changing alpha

(a) $\alpha = 1$
(b) $\alpha = 2$
(c) $\alpha = 3$
(d) $\alpha = 4$
(e) $\alpha = 6$
(f) $\alpha = 9$
Surface Detailing/Carving

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

\[
F(x) = \begin{cases} 
0 & f_{se}(V) > 0 \\
\beta \cdot f_{se}(V) & f_{se}(V) \leq 0
\end{cases}
\]

\[V = x - x_c\]

- \(x_c\): tool center
- \(f_{se}(V)\): implicit superellipsoid equation
- \(\beta = -1\) drives the surface outwards
- \(\beta = 1\) drives the surface inwards
Interactive Smoothing

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

\[ F'(x) = \gamma \cdot g(d_g(x)) \cdot \kappa(x) \]

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

• \( \gamma \): constant that controls the amount of smoothing
• \( d_g \): Euclidean distance from \( x \) to cursor location
• \( \kappa \): mean curvature
• \( r \): radius of influence
Level Set Surface Editing
Sketch-based Operators
Single Cross-section Curve

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

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

- \( d_{up}(x) \): \( x_{cs} \) is closest point on \( C_s \)
- \( x_{cd} \) is corresponding point on \( C_d \)
- Distance is \( |x_{cs} - x_{cd}| \)
- \( d_{in}(x) \): geodesic distance to \( C_s \) from \( x \)
- \( d_{out}(x) \): geodesic distance to \( B \) from \( x \)
- Max functions taken over all points in the ROI
Single Cross-section Curve
**Multiple Cross-Section Curves**

\[
F(x) = \frac{d_{up}(x)}{\max(d_{up}(x))} \ast f(d_{out}(x)) \ast \frac{\max(d_{in}(x)) - d_{in}(x)}{\max(d_{in}(x))}.
\]

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

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

\[
F(x) = \sum_{c=1}^{N_c} \alpha_c(x) \ast F_c(x)
\]

- \(N_c\) is the number of cross-section curves
- \(\alpha_c(x)\) is the weight function that determines the effect of curve \(c\) on point \(x\).
  - What is \(\alpha_c(x)\)?
    - Results with \(\alpha(c, x) = 1.0\) shown below
    - Ideas: \(\alpha_c(x)\) decreases linearly/exponentially/? with \(d_{in}(x)\)
# Blending Functions

![Blending Functions Diagram]

**Method**
- $\alpha^1_c(x) = \frac{1}{2} + \frac{1}{2} \times \cos\left(\frac{d_{in}(x)}{\max(d_{in}(x))} \pi\right)$
- $\alpha^2_c(x) = 1 - \frac{d_{in}(x)}{\max(d_{in}(x))}$
- $\alpha^3_c(x) = \frac{1}{N_c}$
- $\alpha^4_c(x) = \frac{1}{d_{in}(x)}$

<table>
<thead>
<tr>
<th>Method</th>
<th>Criteria</th>
<th>Fits to curves</th>
<th>Smooth Transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closest curve</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>$\alpha^1_c$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$\alpha^2_c$</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>$\alpha^3_c$</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$\alpha^4_c$</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>
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

Sketching on the surface
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

\[
F_C(x) = \frac{d_{up}(\bar{x})}{\max(d_{up}(\bar{x}))} \times \frac{\max(d_{in}(\bar{x})) - d_{in}(\bar{x})}{\max(d_{in}(\bar{x}))} \\
F(x) = \sum_{c=1}^{N_C} \alpha_c(x) \times F_c(x)
\]
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
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
• 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
Results: Handle-Based

Creating a bear
Results: Handle-Based

Creating a fantasy character
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>

<table>
<thead>
<tr>
<th>Model</th>
<th>Editing Details</th>
<th>Speed (fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lake model, Dimensions: 161 x 161 x 101</td>
<td>Plants and rocks</td>
<td>Pulling on a point, symmetric ROI</td>
</tr>
<tr>
<td></td>
<td>Surface on the rightmost plant</td>
<td>Surface detailing</td>
</tr>
<tr>
<td></td>
<td>Log</td>
<td>Surface Detailing</td>
</tr>
<tr>
<td></td>
<td>Animal (body)</td>
<td>Sketch-based editing</td>
</tr>
<tr>
<td></td>
<td>Animal (eyes)</td>
<td>Pulling on a curve, symmetric ROI</td>
</tr>
<tr>
<td></td>
<td>Animal (eyeballs)</td>
<td>Surface Detailing</td>
</tr>
<tr>
<td>Mannequin Head, Dimensions: 360 x 435 x 510</td>
<td>Hair, eyes and eyebrows</td>
<td>Surface Detailing</td>
</tr>
<tr>
<td></td>
<td>Horns</td>
<td>Pulling on a point, symmetric ROI</td>
</tr>
<tr>
<td></td>
<td>Nose and ears</td>
<td>Sketch-based editing</td>
</tr>
<tr>
<td></td>
<td>Chin</td>
<td>Pulling on a curve, symmetric ROI</td>
</tr>
<tr>
<td>Octopus, Dimensions: 322 x 322 x 202</td>
<td>Body and arms</td>
<td>Pulling on a point, symmetric ROI</td>
</tr>
<tr>
<td></td>
<td>Head</td>
<td>Sketch-based editing</td>
</tr>
<tr>
<td></td>
<td>Eyes</td>
<td>Interactive carving</td>
</tr>
<tr>
<td></td>
<td>Nose</td>
<td>Surface detailing</td>
</tr>
<tr>
<td>Teapot, Dimensions: 156 x 232 x 124</td>
<td>Erasing spout and top handle</td>
<td>Interactive carving</td>
</tr>
<tr>
<td></td>
<td>New handles</td>
<td>Pulling on a point, symmetric ROI</td>
</tr>
<tr>
<td>Cartoon frog, Dimensions: 401 x 401 x 401</td>
<td>Mouth</td>
<td>Interactive Carving</td>
</tr>
<tr>
<td></td>
<td>Tongue</td>
<td>Surface detailing</td>
</tr>
<tr>
<td></td>
<td>Eyes (balls)</td>
<td>Pulling on a point, symmetric ROI</td>
</tr>
<tr>
<td></td>
<td>Eyes (crosses)</td>
<td>Surface detailing</td>
</tr>
<tr>
<td>Cartoon bear, Dimensions: 320 x 320 x 600</td>
<td>Arms, claws and coat</td>
<td>Surface detailing</td>
</tr>
<tr>
<td></td>
<td>Legs, ears and nose</td>
<td>Pulling on a point, symmetric ROI</td>
</tr>
<tr>
<td></td>
<td>Eyes</td>
<td>Interactive carving</td>
</tr>
<tr>
<td>Aneurysm, Dimensions: 621 x 371 x 346</td>
<td>Splitting veins</td>
<td>Interactive carving</td>
</tr>
<tr>
<td></td>
<td>Connecting veins</td>
<td>Pulling on a point, symmetric ROI</td>
</tr>
</tbody>
</table>
High-Resolution Models Using Spatial Hash Tables

Dimensions:
944 × 2048 × 1709
3,304,030,208 voxels
(in dense volume)

28,582,546 voxels

26,236,562 voxels
Results
High-Resolution Models
Results
High-Res Models

Dimensions:
654 × 600 × 1794
703,965,600 voxels in dense volume vs. 11,852,818 voxels in sparse volume
Results

High-Resolution Models

Dimensions:

3081 × 2057 × 69

437,295,573 voxels in dense volume vs.

44,552,586 voxels in sparse volume

Open level set!
Level Set Surface Editing
Multiresolution Operators
Hierarchical LS Model

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

- Changes cascaded post-edit
- Local narrow-band

- Fast Marching Methods
- Blended using CBS
Multires Results

- Level 1
  - Original Level 1 model
  - Modified Level 1 model
  - Low resolution surface modification

- Level 2
  - Model after Level 1 changes are blended
  - Original Level 2 model

- Level 3
  - Model after Level 1 changes are blended
  - Model after Level 2 details are added

- Original Level 3 model
  - Model after Level 2 changes are blended
  - Model after Level 3 details are added
  - Modified Level 1 model
Multires Results

- Low resolution surface modification
- Level 1
- Level 2
- Level 3

Model after Level 1 changes are blended
Model after Level 2 details are added
Model after Level 3 details are added
Original Level 3 model
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

![Diagram showing geometric texture transfer](image_url)
Geometric Texture Transfer

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

APPLICATIONS

• Level Set Model Introduction

• 3D Metamorphosis

• Contour-based Surface Reconstruction

• Surface Editing
Papers


Questions?

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-0702441, and CAPES (Brazil) grant 88888.075221/2013-00.