A Remeshing Approach to Multiresolution Modeling

Mario Botsch      Leif Kobbelt
Multiresolution Modeling

Shape deformation with intuitive detail preservation
Multiresolution Modeling

Frequency decomposition

Change low frequencies

Local frame details
Multiresolution Modeling
Multiresolution Modeling
Multiresolution Modeling

Decomposition → Detail Information → Editing → Reconstruction
Two Different Meshes

- User interaction
- Decomposition operator
- Deformation operator
- Reconstruction operator
- Responsible for robustness & efficiency
Detail Encoding

Displacements in normal direction
Detail Encoding

Displacements in normal direction

Independent tesselations!
Remeshing?

- Features, sharp edges
- Hand-crafted triangulation
- Low frequency surface
- No aliasing problems

Detailed → Remesh base surface

Base
Multiresolution Modeling

Decomposition → Remeshing → Detail Information → Editing → Reconstruction
Outline

• Introduction
• Freeform Modeling
• Remeshing
• Results
Modeling Requirements

• Per-vertex interpolation constraints
  • Arbitrary support
• Physically plausible behaviour
  • Stiffness, smoothness
Boundary Constraint Modeling

- Prescribe boundary constraints
- vertex positions
- $C^0 - C^2$ continuities

- Constraint energy minimization

$$E_k(S) = \int F_k (S_u^k, S_u^{k-1}v, \ldots, S_v^k)$$

- Euler-Lagrange PDE:

$$\Delta^k(S) = 0$$
Energy Functionals

Membrane
\[ \Delta S = 0 \]

Thin-Plate
\[ \Delta^2 S = 0 \]

\[ \Delta^3 S = 0 \]
Modeling Metaphor

- Support region (blue)
- Handle regions (green)
- Fixed vertices (grey)
Discretization $\rightarrow$ Linear System

\[ h = \{h_1, \ldots, h_H\} \]
\[ p = \{p_1, \ldots, p_P\} \]
\[ f = \{f_1, \ldots, f_F\} \]

\[
\begin{pmatrix}
\Delta^k \\
0
\end{pmatrix}
\begin{pmatrix}
I_{F+H} \\
0
\end{pmatrix}
\begin{pmatrix}
p f \\
h
\end{pmatrix}
= 
\begin{pmatrix}
0 f \\
h
\end{pmatrix}
\]

$\Delta^k p = b$
Laplace Discretization

\[ \Delta (p) := \frac{2}{A(p)} \sum_{q_i} (\cot \alpha_i + \cot \beta_i) (p - q_i) \]
Problems

- Degenerate triangles
- Matrix no longer positive definite
- Reconstruction operator unstable
- Matrix unsymmetric
- Better solvers for symmetric matrices

$$\Delta (p) := \frac{2}{A(p)} \sum_{q_i} (\cot \alpha_i + \cot \beta_i) (p - q_i)$$
Why not uniform Laplacian?

Irregular Tessellation

Uniform Weights

Cotangent Weights
Uniform Laplacian Discretization?

Real-world meshes are irregular...
Outline

• Introduction
• Freeform Modeling
• Remeshing
• Results
Remeshing Objectives

- Numerical robustness
  - Triangle roundness
  - Isotropic remeshing

- Computational efficiency
  - Fast linear system solver
  - Symmetric matrix
Isotropic Remeshing

• No global parameterization
  • Explicit remeshing instead

• Several related works:
  • Kobbelt et al. 2000
  • Vorsatz et al. 2003
  • Surazhsky et al. 2003
Isotropisch Remeshing

Specify target edge length $L$

Iterate:

1. Split edges longer than $e_{\text{max}}$
2. Collapse edges shorter than $e_{\text{min}}$
3. Flip edges to get valence 6
4. Relaxation by tangential smoothing

Optimal thresholds?
**Edge Length Thresholds**

\[ |e_{\text{max}} - L| = \left| \frac{1}{2} e_{\text{max}} - L \right| \]

\[ \Rightarrow e_{\text{max}} = \frac{4}{3} L \]

\[ |e_{\text{min}} - L| = \left| \frac{3}{2} e_{\text{min}} - L \right| \]

\[ \Rightarrow e_{\text{min}} = \frac{4}{5} L \]
Remeshing Results

Original

\((\frac{1}{2}, 2)\)

\((\frac{4}{5}, \frac{4}{3})\)
Isotropic Remeshing

• Leads to well-shaped triangles
• Increased robustness
• But matrix still unsymmetric
• Because of Voronoi areas $A(p)$
• Equalize areas !

$$\Delta (p) := \frac{2}{A(p)} \sum_{q_i} (\cot \alpha_i + \cot \beta_i) (p - q_i)$$
Area Equalization

- Assign mass $A(p)$ to each vertex $p$
- Mass weighted centroid

$$
\mathbf{g}_i := \frac{1}{\sum_{q_i} A(q_i)} \sum_{q_i} A(q_i) \mathbf{q}_i
$$

- Tangential update

$$
\mathbf{p}_i \mapsto \mathbf{p}_i + \lambda (I - \mathbf{n}_i \mathbf{n}_i^T) (\mathbf{g}_i - \mathbf{p}_i)
$$
Remeshing Results

Original  \( (\frac{1}{2}, 2) \)  \( (\frac{4}{5}, \frac{4}{3}) \)  Area Eq.
Remeshing Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Edge length deviation (%)</th>
<th>Deviation from 60°</th>
<th>Area deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1/2, 2)</td>
<td>55</td>
<td>6.7</td>
<td>34</td>
</tr>
<tr>
<td>(4/5, 4/3)</td>
<td>27</td>
<td>4.0</td>
<td>13</td>
</tr>
<tr>
<td>Area Eq.</td>
<td>21</td>
<td>5.6</td>
<td>4</td>
</tr>
</tbody>
</table>
Area Equalization Remeshing

- Efficient algorithm
  - Projection instead of local parametrization
  - Remesh 100k triangles in <5 sec
- Very regular mesh
  - Inner angles close to 60°
  - Relative mean area error <5%
Outline

• Introduction
• Freeform Modeling
• Remeshing
• Results
Increased Robustness

- No degenerate triangles
  - Matrix is positive definite

- No obtuse angles
  - Cotangent weights are positive
  - More reliable Laplacian discretization
Symmetric Laplace Matrix

- Replace Voronoi areas by their mean

\[ \bar{\Delta}(p) := \frac{2}{A} \sum_{q_i} (\cot \alpha_i + \cot \beta_i) (p - q_i) \]

- Matrix becomes symmetric

\[ \bar{\Delta}^k p = b \]

- Small low-frequency errors (~0.7%)
  - Compensated by detail encoding (~0.2%)
Different Solvers

• Iterative solvers
  • Not suitable for large systems: $O(n^2)$

• Multigrid solvers
  • Robust and efficient: $O(n)$
  • Quite complicated to implement

• Direct solvers ?
Direct Solvers

- Naive direct solvers are $O(n^3)$
  - Not suitable for large systems

- System is sparse, not band-limited
  - Band-limitation by reordering

- Band-limited factorizing solvers
  - Factorization: $O(bn^2)$
  - Solving: $O(bn)$
Direct Solvers

- **Unsymmetric systems:**
  - Band-limited LU factorization
  - Requires pivoting for stability
  - Compromises band-limiting permutations

- **Symmetric systems:**
  - Band-limited Cholesky factorization
  - Backward stable, exploits symmetry
Comparison

- Iterative solvers
  - Not suitable for large systems: $O(n^2)$

- Multigrid solvers
  - Robust and efficient: $O(n)$
  - Quite complicated to implement

- Direct solvers
  - Same linear complexity
  - Faster by an order of magnitude
  - Considerably easier to use
## Comparison (15k DoF)

<table>
<thead>
<tr>
<th>Method</th>
<th>Precomputation</th>
<th>XYZ Solution</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterative</td>
<td>7.2s</td>
<td>7.4s</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Multigrid</td>
<td>4.5s</td>
<td>0.8s</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Direct</td>
<td>2.4s</td>
<td>0.07s</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>
Multigrid vs. Direct

Precomputation

XYZ Solution

Multigrid

Cholesky

Computer Graphics Group
Mario Botsch
System Overview

Remeshing (per model) → Factorization (per modification) → Back-Subst. (per frame)

Displacements
Conclusion

• Multiresolution framework
  • Independent tesselations
  • Remesh smooth base surface

• Area equalizing isotropic remeshing
  • Improves numerical robustness
  • Yields symmetric matrix

• Allows for direct solvers
  • Significantly faster
  • Easier to use