11.1 Remeshing
Outline

- *What is remeshing?*
- *Why remeshing?*
- *How to do remeshing?*
• What is remeshing?

• Why remeshing?

• How to do remeshing?
Definition

**Given a 3D mesh**
- Already a manifold mesh

**Compute another mesh**
- Satisfy some quality requirements
- Approximate well the input mesh
• *What* is remeshing?

• *Why* remeshing?

• *How* to do remeshing?
Motivation

Unsatisfactory “raw” mesh

- By scanning or implicit representations
Motivation

Unsatisfactory “raw” mesh

• By scanning or implicit representations

Improve mesh quality for further use
Motivation

Unsatisfactory “raw” mesh

- By scanning or implicit representations

Improve mesh quality for further use

- Modeling: easy processing
- Simulation: numerical robustness
- ……

Quality requirements

- Local structure
- Global structure
Local structure

Element type

- Triangles vs. quadrangles

all-triangle mesh

all-quad mesh

quad-dominant mesh
Local structure

Element type

• Triangles vs. quadrangles
Local structure

Element type
- Triangles vs. quadrangles

Element shape
- Isotropic vs. anisotropic
Local structure

Element type
- Triangles vs. quadrangles

Element shape
- Isotropic vs. anisotropic

Element distribution
- Uniform vs. adaptive
Local structure

Element type
- Triangles vs. quadrangles

Element shape
- Isotropic vs. anisotropic

Element distribution
- Uniform vs. adaptive

Element alignment
- Preserve sharp features and curvature lines
## Valence of a *regular* vertex

<table>
<thead>
<tr>
<th></th>
<th>Interior vertex</th>
<th>Boundary vertex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle mesh</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Quadrangle mesh</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
Valence of a *regular* vertex

<table>
<thead>
<tr>
<th></th>
<th>Interior vertex</th>
<th>Boundary vertex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triangle mesh</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Quadrangle mesh</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Different types of mesh structure

- Irregular
- Semi-regular: multi-resolution analysis / modeling
- Highly regular: numerical simulation
- Regular: only possible for special models
• What is remeshing?

• Why remeshing?

• How to do remeshing?
• What is remeshing?

• Why remeshing?

• How to do remeshing?
  - Isotropic remeshing
  - Anisotropic remeshing
• What is remeshing?

• Why remeshing?

• How to do remeshing?
  - Isotropic remeshing
  - Anisotropic remeshing
Isotropic remeshing

Incremental remeshing
- Simple to implement and robust
- Not need parameterization
- Efficient for high-resolution input

Variational remeshing
- Energy minimization
- Parameterization-based → expensive
- Works for coarse input mesh

Greedy remeshing
Isotropic remeshing

Incremental remeshing
- Simple to implement and robust
- Not need parameterization
- Efficient for high-resolution input

Variational remeshing
- Energy minimization
- Parameterization-based → expensive
- Works for coarse input mesh

Greedy remeshing
Local remeshing operators

- Edge Collapse
- Edge Split
- Edge Flip
- Vertex Shift
Specify target edge length $L$

$$L_{\text{max}} = \frac{4}{3} \times L; \quad L_{\text{min}} = \frac{4}{5} \times L;$$

Iterate:

1. **Split** edges longer than $L_{\text{max}}$
2. **Collapse** edges shorter than $L_{\text{min}}$
3. **Flip** edges to get closer to optimal valence
4. Vertex **shift** by tangential relaxation
5. **Project** vertices onto reference mesh
Edge split

Split edges longer than $L_{\text{max}}$

$$|L_{\text{max}} - L| = \left| \frac{1}{2} L_{\text{max}} - L \right|$$

$$\Rightarrow L_{\text{max}} = \frac{4}{3} L$$
Edge collapse

Collapse edges shorter than $L_{\text{min}}$
Optimal valence

- 6 for interior vertices
- 4 for boundary vertices
**Optimal valence**

- 6 for interior vertices
- 4 for boundary vertices

**Improve valences**

- Minimize valence excess

\[
\sum_{i=1}^{4} (\text{valence}(v_i) - \text{opt\_valence}(v_i))^2
\]
Vertex shift

Local “spring” relaxation

- Uniform Laplacian smoothing
- Barycenter of one-ring neighborhood

\[ c_i = \frac{1}{\text{valence}(v_i)} \sum_{j \in N(v_i)} p_j \]
Vertex shift

Local “spring” relaxation

- Uniform Laplacian smoothing
- Barycenter of one-ring neighborhood

\[ c_i = \frac{1}{\text{valence}(v_i)} \sum_{j \in N(v_i)} p_j \]
Local “spring” relaxation

- Uniform Laplacian smoothing
- Barycenter of one-ring neighborhood

\[ c_i = \frac{1}{\text{valence}(v_i)} \sum_{j \in N(v_i)} p_j \]

Keep vertex (approx.) on surface

- Restrict movement to tangent plane

\[ p_i \leftarrow p_i + \lambda (I - n_i n_i^T)(c_i - p_i) \]
Onto original reference mesh

- Find closest triangle
- Use BSP to accelerate $\mathcal{O}(\log n)$
- Barycentric interpolation to compute position & normal
Specify target edge length $L$

Iterate:

1. Split edges longer than $L_{\text{max}}$
2. Collapse edges shorter than $L_{\text{min}}$
3. Flip edges to get closer to optimal valence
4. Vertex shift by tangential relaxation
5. Project vertices onto reference mesh
Adaptive remeshing
Adaptive remeshing

- Compute maximum principle curvature on reference mesh
- Determine local target edge length from maximum curvature
- Adjust edge split / collapse criteria accordingly
Feature preservation
Define feature edges / vertices

- Large dihedral angles
- Material boundaries

Adjust local operators

- Do not touch corner vertices
- Do not flip feature edges
- Collapse along features
- Univariate smoothing
- Project to feature curves
Isotropic remeshing

Incremental remeshing
- Simple to implement and robust
- Not need parameterization
- Efficient for high-resolution input

Variational remeshing
- Energy minimization
- Parameterization-based → expensive
- Works for coarse input mesh

Greedy remeshing
Voronoi Diagram
Voronoi Diagram

Divide space into a number of cells
Voronoi Diagram

Divide space into a number of cells

Dual graph: Delaunay triangulation
Centroidal Voronoi Diagram

For each cell

The generating point $\bullet = \text{mass of center} \ +$

non CVD

CVD
Centroidal Voronoi Diagram

Compute CVD by Lloyd relaxation

1. Compute Voronoi diagram of given points \( p_i \)
2. Move points \( p_i \) to centroids \( c_i \) of their Voronoi cells \( V_i \)
3. Repeat steps 1 and 2 until satisfactory convergence

\[
p_i \leftarrow c_i = \frac{\int_{V_i} x \cdot \rho(x) \, dx}{\int_{V_i} \rho(x) \, dx}
\]
Compute CVD by Lloyd relaxation

1. Compute Voronoi diagram of given points \( p_i \)
2. Move points \( p_i \) to centroids \( c_i \) of their Voronoi cells \( V_i \)
3. Repeat steps 1 and 2 until satisfactory convergence

\[
p_i \leftarrow c_i = \frac{\int_{V_i} \mathbf{x} \cdot \rho(\mathbf{x}) \, d\mathbf{x}}{\int_{V_i} \rho(\mathbf{x}) \, d\mathbf{x}}
\]

CVD maximizes compactness

- Minimize the energy:

\[
\sum_i \int_{V_i} \rho(\mathbf{x}) \| \mathbf{x} - p_i \|^2 \, d\mathbf{x} \rightarrow \text{min}
\]
1. Conformal parameterization of input mesh
2. Compute local density
3. Perform in 2D parameter space
   A. Randomly sample according to local density
   B. Compute CVD by Lloyd relaxation
4. Lift 2D Delaunay triangulation to 3D
Variational remeshing
Adaptive remeshing
Feature preservation
• What is remeshing?
• Why remeshing?
• How to do remeshing?
  - Isotropic remeshing
  - Anisotropic remeshing
Anisotropic remeshing

Artist-designed models

- Conform to the anisotropy of a surface
Anisotropic remeshing

Anisotropic remeshing

Differential geometry

- A local *orthogonal* frame: *min/max* curvature directions and *normal*
3D curvature tensor

**Isotropic**
- Spherical ($k > 0$)
- Planar ($k = 0$)

**Anisotropic**
- Elliptic ($k_{\min} > 0$, $k_{\max} > 0$)
- Parabolic ($k_{\min} = 0$, $k_{\max} > 0$)
- Hyperbolic ($k_{\min} < 0$, $k_{\max} > 0$)

2 principal directions
Principal direction fields

min curvature  max curvature  overlay
Flattening to 2D

- One 3D tensor per vertex
- Discrete conformal parameterization
- 2D tensor field using barycentric coordinates

Piecewise linear interpolation of 2D tensors
• Regular case

- minor foliation
- major foliation
- principal foliations
2D direction fields

- Singularities

- **umbilic**
  (spherical point)
  2D tensor proportional to identity
Umbilics

wedge

trisector
Lines of curvature

- minor net
- major net
- overlay
Lines of curvature

minor net

major net
 Overlay curvature lines in anisotropic regions

 Add umbilical points in isotropic regions
Vertices

intersect lines of curvatures
Edges

straighten lines of curvatures + Delaunay triangulation near umbilics
Resolve T-junctions
Smoothing

quad-triangle subdivision
Anisotropic remeshing

Remeshing results

- Min curvature
- Max curvature
- Result
- Minor net
- Major net
- Overlay
Remeshing results

MeshLab

- meshlab.sourceforge.net
- open source
- available for Windows, MacOSX, and Linux

Graphite

- available for Windows
- MacOSX or Linux?
Remeshing via Graphite

“Mesh” → “remesh” → “pliant” →

- [Optional] flag border as feature
- [Optional] flag sharp edges as feature (dihedral angle)
- [Optional] estimate edge size (bounding box divisions)
- remesh (target edge length)
• Textbook: Chapter 6
• Alliez et al, “Interactive geometry remeshing”, SIGGRAPH 2002
• Alliez et al, “Isotropic surface remeshing”, SMI 2003
• Alliez et al, “Anisotropic polygonal remeshing”, SIGGRAPH 2003
• Botsch & Kobbelt, “A remeshing approach to multiresolution modeling”, Symp. on Geometry Processing 2004
• Marinov et al, “Direct anisotropic quad-dominant remeshing”, Pacific Graphics 2004
• Alliez et al, “Recent advances in remeshing of surfaces”, AIM@Shape state of the art report, 2006
Thanks!

http://cs621.hao-li.com