Remeshing I
Remeshing

Given a 3D mesh, find a “better” discrete representation of the underlying surface.
What is a good mesh?

- Equal edge lengths
- Equilateral triangles
- Valence close to 6
What is a good mesh?

• Equal edge lengths
• Equilateral triangles
• Valence close to 6
• Uniform vs. adaptive sampling
What is a good mesh?

- Equal edge lengths
- Equilateral triangles
- Valence close to 6
- Uniform vs. adaptive sampling
- Feature preservation
What is a good mesh?

- Equal edge lengths
- Equilateral triangles
- Valence close to 6
- Uniform vs. adaptive sampling
- Feature preservation
- Alignment to curvature lines
- Isotropic vs. anisotropic
What is a good mesh?

- Equal edge lengths
- Equilateral triangles
- Valence close to 6
- Uniform vs. adaptive sampling
- Feature preservation
- Alignment to curvature lines
- Isotropic vs. anisotropic
- Triangles vs. quadrangles
Applications

– Design
– Reverse engineering
– Simulation
– Visualization
Local Structure

- Element **type**
  - Triangle
  - Quadrangle
  - Polygon
Local Structure

- Element shape (isotropy vs anisotropy)

[Diagram showing the transition from low isotropy to high isotropy]
Local Structure

Element distribution (sizing, grading)
Local Structure

• Element orientation
Local Structure

• Element orientation
Isotropic Remeshing

Well-shaped elements

– for processing & simulation (numerical stability & efficiency)
Two Fundamental Approaches

• **Parameterization-based**
  – map to 2D domain / 2D problem
  – computationally more expensive
  – works even for coarse resolution remeshing

• **Surface-oriented**
  – operate directly on the surface
  – treat surface as a set of points / polygons in space
  – efficient for high resolution remeshing
Parameterization Based
Parameterization Based
Parameterization Based

parameterize
Parameterization Based

re-parameterize
Parameterization Based

apply
Surface Oriented
Surface Oriented

sample
Surface Oriented

filter
Surface Oriented

back-project
Variational Methods

- Minimize a global energy
  - Both discrete and continuous degrees of freedom
    - Connectivity, geometry
- What to optimize
  - Edge lengths, angles, triangle shapes,…
- Isotropic point sampling $\rightarrow$ well shaped triangles
  - Simple and effective
Parameterization-Based Remeshing

[Alliez et al. ‘03]

- Compute 2D parameterization
  - Conformal: only area distortion

- Sample 2D domain
  - Density based on area distortion

- Triangulate
- Project back to 3D
Isotropic 2D Sampling

- Density based random sampling
  - Does not guarantee uniform distance between samples
Sampling Energy

• Given sites $x_i$ and regions $R_i$, minimize

$$E(x_1, \ldots, x_k, R_1, \ldots, R_k) = \sum_{i=1}^{k} \int_{x \in R_i} \| x - x_i \|^2 \, dx$$

• Spreads out points
Sampling Energy

• Given sites $x_i$ and regions $R_i$, minimize

• If sites are fixed, energy is minimized by the **Voronoi Tessellation**

  Voronoi cell $R_i = \text{All points closer to } x_i \text{ than to other } x_j$
Sampling Energy

• Given sites $x_i$ and regions $R_i$, minimize

$$E(x_1, \ldots, x_k, R_1, \ldots, R_k) = \sum_{i=1}^{k} \int_{x \in R_i} \| x - x_i \|^2 \, dx$$

• If sites are fixed, energy is minimized by the Voronoi Tessellation

• Among all Voronoi tessellations, energy is minimized when sites are **centroids of cells** = **Centroidal Voronoi Tessellation**
Varying Density

\[
E(x_1, \ldots, x_k, R_1, \ldots, R_k) = \sum_{i=1\ldots k} \int_{x \in R_i} \rho(x) \| x - x_i \|^2 \, dx
\]
Initial Sample Scatter

Voronoi tessellation
Optimized Sample Placement

centroidal Voronoi tessellation
Uniform Remeshing

back-project

50kV
Lloyd Algorithm

• Alternate:
  – Voronoi partitioning
  – Move sites to respective centroids
Centroidal Voronoi Diagrams
Centroidal Voronoi Diagrams
Centroidal Voronoi Diagrams
Centroidal Voronoi Diagrams
Centroidal Voronoi Tessellation

- Lloyd converges slowly
  - Stop when points “stop” moving

- Recent fast algorithm: direct optimization of the energy using quasi-Newton
  “On centroidal voronoi tessellation—energy smoothness and fast computation” [Liu et al., TOG ’09]
Uniform vs. Adaptive
Uniform Sampling
Adaptive Sampling
Limitations

• Closed meshes
  – Need a good cut
  – Free boundary parameterization
  – Stitch seams afterwards

• Protruding legs
  – Sampling
  – Numerical problems
Smart Cut, Free Boundary
Free vs. Fixed Boundary
Visible Seams

visible seam
Naive Cut, Numerical Problems
Direct Surface Remeshing [Botsch et al. ’04]

• Avoid global parameterization
  – Numerically very sensitive
  – Topological restrictions

• Avoid local parameterizations
  – Expensive computations

• Use local operators & back-projections
  – Resampling of 100k triangles in < 5s
Local Remeshing Operators

- Edge Collapse
- Edge Split
- Edge Flip
- Vertex Shift
Isotropic Remeshing

Specify target edge length $L$

Compute edge length range $[L_{\text{min}}, L_{\text{max}}]$

Iterate:

1. Split edges longer than $L_{\text{max}}$
2. Collapse edges shorter than $L_{\text{min}}$
3. Flip edges to get closer to valence 6
4. Vertex shift by tangential relaxation
5. Project vertices onto reference mesh
Edge Collapse / Split

\[ \frac{1}{2} L_{\text{max}} \quad \frac{1}{2} L_{\text{max}} \]

\[ L_{\text{max}} \]

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

\[ \frac{3}{2} L_{\text{min}} \quad \frac{3}{2} L_{\text{min}} \]

\[ L_{\text{min}} \quad L_{\text{min}} \]

\[ \Rightarrow L_{\text{min}} = \frac{4}{5} L \]
Edge Flip

• Improve valences
  – Avg. valence is 6 (Euler)
  – Reduce variation

• Optimal valence is
  – 6 for interior vertices
  – 4 for boundary vertices
Edge Flip

- Improve valences
  - Avg. valence is 6 (Euler)
  - Reduce variation
- Optimal valence is
  - 6 for interior vertices
  - 4 for boundary vertices
- 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
  – Bary-center of one-ring neighbors

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

- Local “spring” relaxation
  - Uniform Laplacian smoothing
  - Bary-center of one-ring neighbors

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

- Local “spring” relaxation
  - Uniform Laplacian smoothing
  - Bary-center of one-ring neighbors
    \[ c_i = \frac{1}{\text{valence}(v_i)} \sum_{j \in N(v_i)} p_j \]

- Keep vertex (approx.) on surface
  - Restrict movement to tangent plane
    \[ p_i \leftarrow p_i - n_i n_i^T (c_i - p_i) \]
Vertex Projection

• Project vertices onto original reference mesh
• Assign position & interpolated normal
Remeshing Results

Original

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

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

Edge length deviation (%)

Deviation from 60°
Feature Preservation?
Feature Preservation

• Define features
  – Sharp edges
  – Material boundaries

• Adjust local operators
  – Don’t move corners
  – Collapse only along features
  – Don’t flip feature edges
  – Univariate smoothing
  – Project to feature curves
Adaptive Remeshing

- Precompute max. curvature on reference mesh
- Target edge length locally determined by curvature
- Adjust split / collapse criteria
Next Time:
Quad Dominant Meshing