9.2 Decimation
Parameterization

- **isometric** \( I(u, v) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \)

- **conformal** \( I(u, v) = s(u, v) \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \)

- **equiareal** \( \text{det}(I(u, v)) = 1 \)

\[
I = \begin{pmatrix}
\mathbf{x}_u^T \mathbf{x}_u & \mathbf{x}_u^T \mathbf{x}_v \\
\mathbf{x}_u^T \mathbf{x}_v & \mathbf{x}_v^T \mathbf{x}_v
\end{pmatrix}
\]
Harmonic Maps

- minimize Dirichlet energy: 
  \[ \int_{\Omega} \| \nabla x \|^2 = \int_{\Omega} \| x_u \|^2 + \| x_v \|^2 \, du \, dv \]
- Euler-Lagrange PDE 
  \[ \Delta x(u, v) = 0 \]

Discrete Harmonic Maps

Convex Combination Maps

original mesh  uniform weights  cotan weights  mean value
Last Time

fixed vs. open boundaries

cutting the mesh → disk topology

texture atlases

constrained parameterization
Mesh Optimization

Smoothing
• Low geometric noise

Fairing
• Simplest shape

Decimation
• Low complexity

Remeshing
• Triangle Shape
Mesh Decimation

Oversampled 3D scan data

~150k triangles

~80k triangles
Mesh Decimation

Over tesselation: e.g., Iso-surface extraction
Mesh Decimation

Multi-resolution hierarchies for

• efficient geometry processing
• level-of-detail (LOD) rendering
Mesh Decimation

Adaptation to hardware capabilities
Mesh Decimation

Adaptation to hardware capabilities
Size-Quality Tradeoff
Given $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, find $\mathcal{M}' = (\mathcal{V}', \mathcal{F}')$ such that

- $|\mathcal{V}'| = n < |\mathcal{V}|$ and $\|\mathcal{M} - \mathcal{M}'\|$ is minimal, or
- $\|\mathcal{M} - \mathcal{M}'\| < \epsilon$ and $|\mathcal{V}'|$ is minimal
Problem Statement

Given $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, find $\mathcal{M}' = (\mathcal{V}', \mathcal{F}')$ such that

- $|\mathcal{V}'| = n < |\mathcal{V}|$ and $\|\mathcal{M} - \mathcal{M}'\|$ is minimal, or
- $\|\mathcal{M} - \mathcal{M}'\| < \epsilon$ and $|\mathcal{V}'|$ is minimal

NP hard

- Look for sub-optimal solution

Respect additional fairness criteria

- Normal deviation, triangle shape, colors,…
Mesh Decimation methods

• Vertex Clustering

• Iterative Decimation
Vertex Clustering

- Cluster Generation
- Computing a representative
- Mesh generation
- Topology changes
**Cluster Generation**

- Uniform 3D grid
- Map vertices to cluster cells
- Computing a representative
- Mesh generation
- Topology changes
Cluster Generation
- Hierarchical approach
- Top-down or bottom-up
- Computing a representative
- Mesh generation
- Topology changes
Vertex Clustering

- Cluster Generation
- **Computing a representative**
  - Average/median vertex position
  - Error quadrics
- Mesh generation
- Topology changes
Computing a Representative

average vertex position → low pass filter
Computing a Representative

median vertex position $\rightarrow$ sub-sampling
Computing a Representative

error quadrics → feature preservation
Squared distance to plane

\[ p = (x, y, z, 1)^T, \quad q = (a, b, c, d)^T \]

\[ \text{dist}(q, p)^2 = (q^T p)^2 = p^T (qq^T) p =: p^T Q_q p \]
Error Quadrics

Squared distance to plane

\[ p = (x, y, z, 1)^T, \quad q = (a, b, c, d)^T \]

\[ \text{dist}(q, p)^2 = (q^T p)^2 = p^T (qq^T) p =: p^T Q_q p \]

\[ Q_q = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \]
Sum of distances to vertex planes

\[
\sum_i \text{dist}(q_i, p)^2 =
\]
Error Quadrics

Sum of distances to vertex planes

\[ \sum_i \text{dist}(q_i, p)^2 = \sum_i p^T Q_{q_i} p = p^T \left( \sum_i Q_{q_i} \right) p =: p^T Q_p p \]

Point that minimizes the error

\[
\begin{bmatrix}
q_{11} & q_{12} & q_{13} & q_{14} \\
q_{21} & q_{22} & q_{23} & q_{24} \\
q_{31} & q_{32} & q_{33} & q_{34} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
p^* = \begin{bmatrix}
0 \\
0 \\
0 \\
1
\end{bmatrix}
\]
Comparison

average

median

error quadric
Vertex Clustering

- Cluster Generation
- Computing a representative
- **Mesh generation**
  - Clusters $\mathbf{p} \leftrightarrow \{p_0, \ldots, p_n\}$, $\mathbf{q} \leftrightarrow \{q_0, \ldots, q_n\}$
  - Connect $(\mathbf{p}, \mathbf{q})$ if there was an edge $(p_i, q_j)$
- Topology changes
Vertex Clustering

- Cluster Generation
- Computing a representative
- Mesh generation

**Topology changes**
- If different sheets pass through on cell
- Can be non-manifold
Vertex Clustering

- Cluster Generation
- Computing a representative
- Mesh generation
- **Topology changes**
  - If different sheets pass through on cell
  - Can be non-manifold
Mesh Decimation methods

- Vertex Clustering
- Iterative Decimation
Incremental Decimation

- **General Setup**
- Decimation operators
- Error metrics
- Fairness criteria
- Topology changes
General Setup

Repeat:

pick mesh region
apply decimation operator

Until no further reduction possible
For each region
  evaluate quality after decimation
  enqueue(quality, region)

Repeat:
  pick best mesh region
  apply decimation operator
  update queue
Until no further reduction possible
For each region
  evaluate quality after decimation
  enqueue(quality, region)

Repeat:
  pick best mesh region
  if error < ε
    apply decimation operator
    update queue
Until no further reduction possible
Incremental Decimation

- General Setup
- **Decimation operators**
- Error metrics
- Fairness criteria
- Topology changes
Decimation Operators

- What is a “region”?
- What are the DOFs for re-triangulation?
- Classification
  - topology-changing vs. topology-preserving
  - subsampling vs. filtering
  - inverse operation $\rightarrow$ progressive meshes
Select a vertex to be eliminated
Vertex Removal

Select all triangles sharing this vertex
Remove the selected triangles, creating a hole.
Fill the hole with triangles
Decimation Operators

- Remove vertex
- Re-triangulate hole
  - Combinatorial DOFs
  - Sub-sampling
Decimation Operators

- Merge two adjacent triangles
- Define new vertex position
  - Continuous DOF
  - Filtering

Edge Collapse

Vertex Split
Decimation Operators

- Collapse edge into one end point
  - Special vertex removal
  - Special edge collapse
- No DOFs
  - One operator per half-edge
  - Sub-sampling

H. Hoppe: Progressive Meshes
Edge Collapse
Edge Collapse
Edge Collapse
Edge Collapse
Edge Collapse
Edge Collapse
Edge Collapse (Flip!)
Application: Progressive Meshes
Priority Queue Updating
Incremental Decimation

- General Setup
- Decimation operators
- **Error metrics**
- Fairness criteria
- Topology changes
Local distance to mesh [Schröder et al. ‘92]

- Compute average plane
- No comparison to original geometry
Global Error Metrics

Simplification envelopes [Cohen al. ‘96]

• Compute (non-intersecting) offset surfaces
• Simplification guarantees to stay within bounds
(Two-sided) Hausdorff distance: Maximum distance between two shapes

$$d(A, B) := \max_{a \in A} \min_{b \in B} \| a - b \|$$

- In general $d(A, B) \neq d(B, A)$
- Computationally involved
Scan data: One-sided Hausdorff distance sufficient

- From original vertices to current surface
Global Error Metrics

Error quadrics [Garland, Heckbert 97]

- Squared distance to planes at vertex
- No bound on true error

\[ p_i^T Q_i p_i = 0, \quad i=\{1,2\} \]

\[ p_3^T Q_3 p_3 = \text{min} \]

\[ < \varepsilon \quad \rightarrow \quad \text{ok} \]
Global Error Metrics

Initialization:

• Assign each vertex the quadric built from all its incident triangles’ planes

Decimation:

• After collapsing edge $(p_1, p_2) \rightarrow p_3$, simply add the corresponding quadrics: $Q_3 = Q_1 + Q_2$

Memory consumption

• Quasi-global error metric with 10 floats per vertex
• $N =$ number of vertices

• Priority Queue for half edges
  • $6N \log(6N)$

• Error control
  • Local $O(1)$ ⇒ global $O(N)$
  • Local $O(N)$ ⇒ global $O(N^2)$
Incremental Decimation

- General Setup
- Decimation operators
- Error metrics
- **Fairness criteria**
- Topology changes
• Rate quality after decimation
• Approximation error
Fairness Criteria

- Rate quality after decimation
- Approximation error
- Triangle shape

\[
\frac{r_1}{e_1} < \frac{r_2}{e_2}
\]
Fairness Criteria

• Rate quality after decimation
  • Approximation error
  • Triangle shape
• Rate quality after decimation
  • Approximation error
  • Triangle shape
Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balance
Fairness Criteria

- Rate quality after decimation
  - Approximation error
  - Triangle shape
  - Dihedral angles
  - Valence balance
  - Color differences
  - ...

Incremental Decimation

- General Setup
- Decimation operators
- Error metrics
- Fairness criteria
- Topology changes
Fairness Criteria

- Merge vertices across non-edges
  - Changes mesh topology
  - Need spatial *neighborhood* information
  - Generates *non-manifold* meshes

![Diagram showing vertex contraction and separation](image-url)
Comparison

• **Vertex clustering**
  - fast but difficult to control simplified mesh
  - topology changes, non-manifold meshes
  - global error bound, but often not close to optimum

• **Iterative decimation with quadric error metrics**
  - good trade-off between mesh-quality and speed
  - explicit control over mesh topology
  - restricting normal deviation improves mesh quality
• Quadric-based simplification
  • http://graphics.cs.uiuc.edu/~garland/software/qslim.html
  • http://www.openmesh.org


• Kobbelt et al., A general framework for mesh decimation, Graphics Interface 1998.
Next Time

Remeshing
Thanks!

http://cs621.hao-li.com