Modeling aspects of 3d Photography

1. Outline

Given:

Dense triangular mesh M, possibly with colored vertices.

Will talk about:

a. Mesh simplification and multiresolution analysis of meshes

b. Mesh parameterization

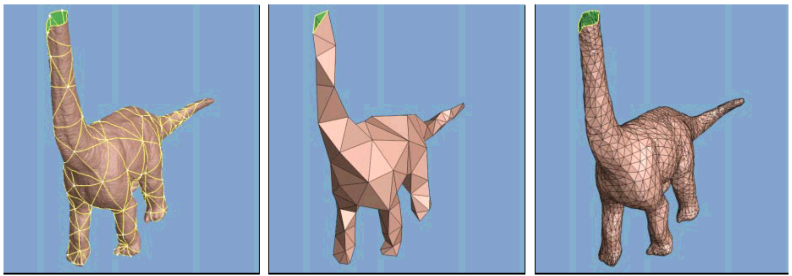
c. Conversion to other surface representations

a. Mesh simplification and multiresolution analysis of meshes

Decompose original mesh M into

- simple base mesh M_0
- series of correction terms of decreasing magnitude

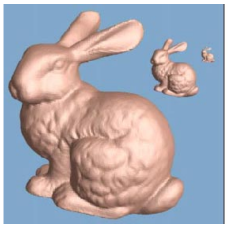
By successively adding correction terms, can generate sequence of meshes M_0, M_1, M_2,… approaching M.



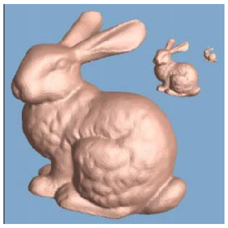
(g) Original mesh (103,713 faces) (h) Base mesh (229 faces) (i) Approx. ($\epsilon = 1.5\%$; 4,725 faces)

Motivation

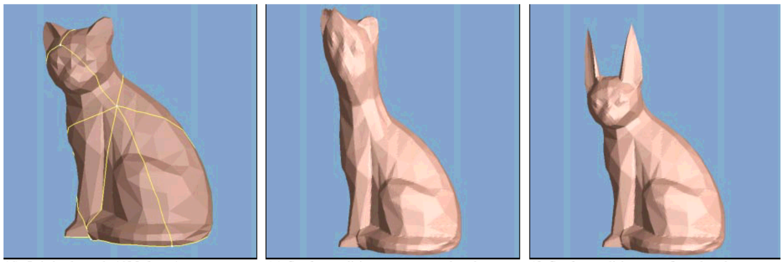
- Compression
- Progressive transmission
- Level-of-detail control
- Multiresolution editing



Full resolution: 70K faces



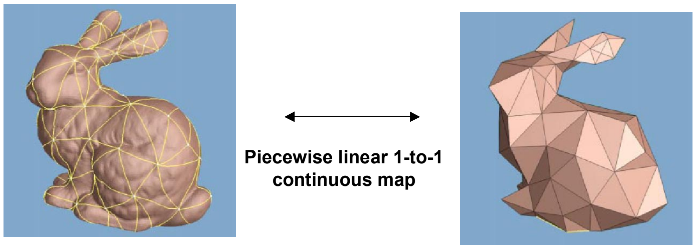
LoD control: 38K - 4.5K - 1.9K faces



(d) Original mesh (698 faces) (e) Surface editing at a coarse level (f) Surface editing at a finer level

b. Mesh parameterization

Establish correspondence between points on original mesh and points on the meshes M_0, M_1,…..



Piecewise linear 1-to-1 continuous map

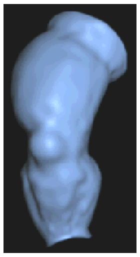
Motivation

Texture mapping --- can cover up lack of geometric detail

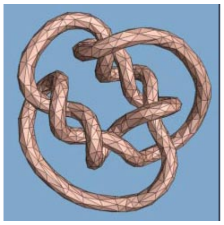
c. Conversion to other surface representations

Approximate mesh by subdivision surface, spline patches,…….

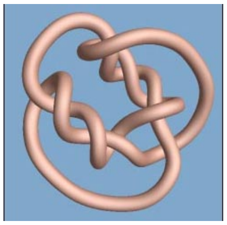
Mesh (25K vertices)



B-spline surface
27 x 36 control points



Mesh



Subdivision surface

Motivation

- Parsimonious representation of smooth surfaces
- Use of models in CAD systems

2. Mesh simplification and multiresolution analysis of meshes

2.1 Representation of meshes

A mesh is a pair $M = (K, V)$:

- The simplicial complex K determines the connectivity of the vertices, edges, and faces of the surface.
- The vertex positions $V = (\underline{v}_1, \ldots, \underline{v}_m)$ determine the geometric shape of the surface in R^3.

Definition of K:

- K is a collection of subsets of $\{1, \ldots, m\}$
- $\{1\}, \ldots, \{m\} \in K$
- $\{i, j\} \in K$ iff $\underline{v}_i$ and $\underline{v}_j$ are connected by edge in S.
- $\{i, j, k\} \in K$ iff $\underline{v}_i, \underline{v}_j, \underline{v}_k$ are vertices of a face of S

Define topological realization $|K| \subset R^m$:

- mesh in R^m
- obtained by identifying 1-simplices $\{1\}, \ldots, \{m\}$ with $\underline{e}_1, \ldots, \underline{e}_m$.

Define linear map $\Phi_V : R^m \to R^3$ by

$$\Phi_V(\underline{e}_i) = \underline{v}_i, \quad i = 1, \ldots, m.$$

For every point $\underline{y} \in S$ there is unique point $\underline{b} \in |K|$ with

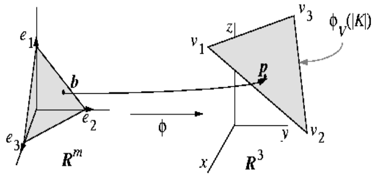
$$\Phi_V(\underline{b}) = \underline{y}$$

$\underline{b}$: barycentric coordinates of $\underline{y}$

Note: $\underline{b}$ has at most 3 non-zero components.

Simplicial complex K
vertices: $\{1\}, \{2\}, \{3\}$
edges: $\{1,2\}, \{2,3\}, \{1,3\}$
faces: $\{1,2,3\}$

Topological realization $|K|$ Geometric realization (V)



2.2 Fitting a mesh to a set of points

Given:

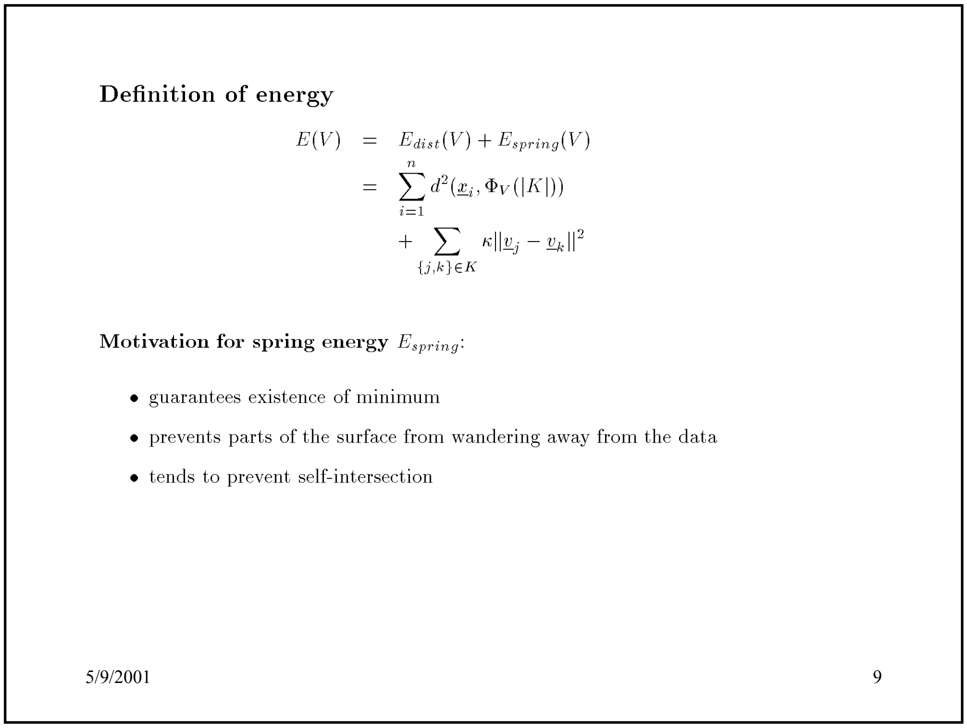
- Set of points $\underline{x}_1, \ldots, \underline{x}_n \in R^3$
- Initial mesh $M = (K, V)$

Goal:

Change vertex positions $V = \{\underline{v}_1, \ldots, \underline{v}_n\}$ to improve approximation of points by mesh.

Approach:

- Define energy function $E(V)$ that measures distance of mesh from points.
- Minimize $E(V)$ over vertex positions using numerical optimization.

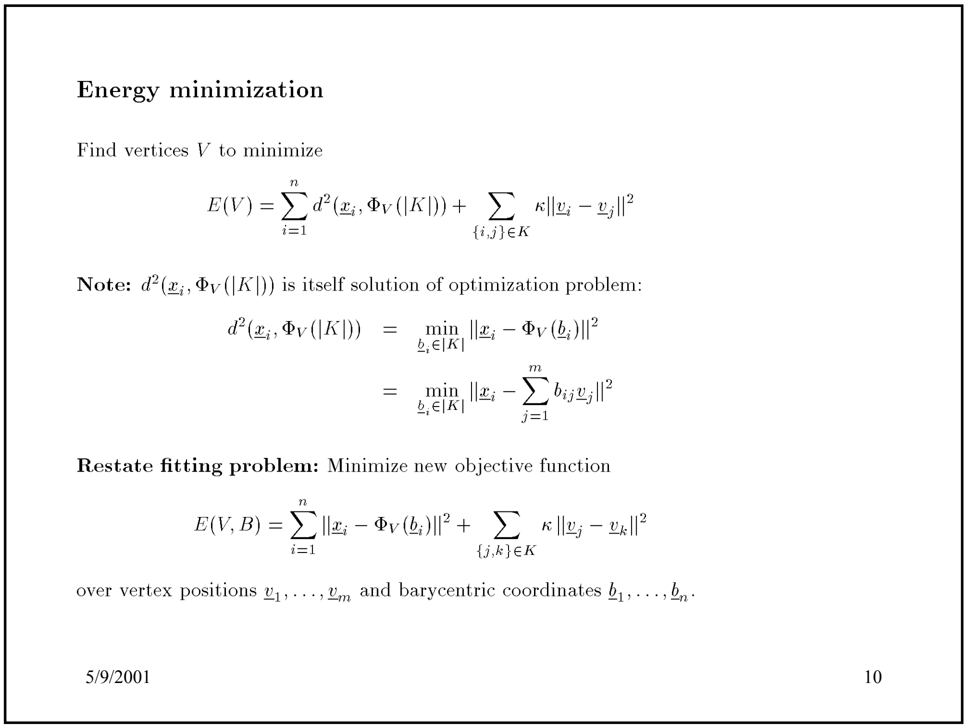


Definition of energy

$$\begin{aligned} E(V) &= E_{dist}(V) + E_{spring}(V) \\ &= \sum_{i=1}^{n} d^2(\underline{x}_i, \Phi_V(|K|)) \\ &\quad + \sum_{\{j,k\}\in K} \kappa \|\underline{v}_j - \underline{v}_k\|^2 \end{aligned}$$

Motivation for spring energy E_{spring}:

- guarantees existence of minimum
- prevents parts of the surface from wandering away from the data
- tends to prevent self-intersection



Energy minimization

Find vertices V to minimize

$$E(V) = \sum_{i=1}^{n} d^2(\underline{x}_i, \Phi_V(|K|)) + \sum_{\{i,j\}\in K} \kappa \|\underline{v}_i - \underline{v}_j\|^2$$

Note: $d^2(\underline{x}_i, \Phi_V(|K|))$ is itself solution of optimization problem:

$$\begin{aligned} d^2(\underline{x}_i, \Phi_V(|K|)) &= \min_{\underline{b}_i \in |K|} \|\underline{x}_i - \Phi_V(\underline{b}_i)\|^2 \\ &= \min_{\underline{b}_i \in |K|} \|\underline{x}_i - \sum_{j=1}^{m} b_{ij}\underline{v}_j\|^2 \end{aligned}$$

Restate fitting problem: Minimize new objective function

$$E(V, B) = \sum_{i=1}^{n} \|\underline{x}_i - \Phi_V(\underline{b}_i)\|^2 + \sum_{\{j,k\}\in K} \kappa \, \|\underline{v}_j - \underline{v}_k\|^2$$

over vertex positions $\underline{v}_1, \ldots, \underline{v}_m$ and barycentric coordinates $\underline{b}_1, \ldots, \underline{b}_n$.

Optimization method

Want to minimize

$$E(V,B) = \sum_{i=1}^{n} ||\underline{x}_i - \Phi_V(\underline{b}_i)||^2 + \sum_{\{j,k\}\in K} \kappa\, ||\underline{v}_j - \underline{v}_k||^2$$

over vertex positions $\underline{v}_1, \ldots, \underline{v}_m$ and barycentric coordinates $\underline{b}_1, \ldots, \underline{b}_n$.

Suggests alternating minimization scheme:

- For fixed $\underline{v}$'s, can find optimal $\underline{b}$'s by projection
- For fixed $\underline{b}$'s, can find optimal $\underline{v}$'s by solving 3 *linear* LS problems

Note:

- can use continuity between iterations in projection step
- LS problems are large but sparse — use conjugate gradients

2.3 Basics of mesh simplification

Given:

Complex mesh M_{orig}, possibly with associated information — face colors, face textures, normals at the vertices (or at the corners), sharp edges, etc.

Goal:

Generate sequence of simpler meshes $M_0, M_1, M_2, \ldots$ approximating M_{orig}.

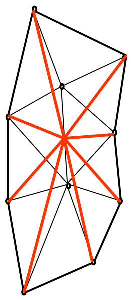
Important characteristics

- Are the simplified meshes surfaces (no singularities)?
- Can the topological type change?
- Efficient ”undo”?
- Smooth ”geomorphing” between approximations?
- Texture mapping?

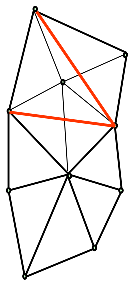
Outline for a class of procedures

Define elementary simplification operation

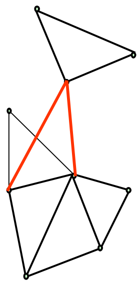
- collapse an edge
- remove a vertex and re-triangulate — more general than edge collapse
- merge two vertices, not necessarily connected by an edge



Edge collapse



Vertex removal



Vertex merge

Outline for a class of procedures

- Define elementary simplification operation (edge collapse, vertex removal, vertex merge)
- Define error metric that measures the discrepancy between a simplified mesh and the original mesh M_{orig}.
- Given the current simplified mesh M_j, collapse the edge (remove the vertex,..) that leads to the smallest increase in error.

2.4 Progressive meshes (Hoppe 1997)

Given:

Complex mesh M_{orig}, with associated attributes:

- discrete attributes – colors – for faces
- scalar attributes – normals – for corners = (vertex, face) pairs

Goal:

Generate sequence of simpler meshes $M_0, M_1, M_2, \ldots$ approximating M_{orig}.

Characteristics

- Are the simplified meshes surfaces (no singularities)? Yes
- Can the topological type change? No
- Efficient "undo"? Yes
- Smooth "geomorphing" between approximations? Yes
- Texture mapping? No (at the time)

Sketch of simplification method

Elementary simplification operation: edge collapse

Error metric (assuming no attributes):

Sample points $\underline{x}_1, \ldots, \underline{x}_n$ from M_{orig} (vertices $\underline{v}_1, \ldots, v_m$ + possibly additional points.)

Measure error of a mesh $M = (K, V)$ by

$$E(M) = \sum_i d^2(\underline{x}_i, M) + \kappa \sum_{\{i,j\} \in K} ||\underline{v}_i - \underline{v}_j||^2$$

Simplification step:

- For each (legally collapsible) edge of current mesh $M = (K, V)$
  - Collapse edge: $K \to K'$
  - Find vertex positions V' minimizing $E(M')$
- Collapse edge for which $E(M')$ is smallest

Computational shortcuts:

- Collapsing an edge will not change optimal positions of vertices far away from edge ⇒
  - Only optimize over position of newly generated vertex
  - Only use those points $\underline{x}_i$ that project onto the disc surrounding the edge
- Initially generate priority queue of edges sorted according to increase in error incurred by collapse

  Collapse edge in front of queue

  Update errors for edges in neighborhood of collapsed edge

Dealing with attributes

Quite complex. Add terms to error metric that penalize changes in topology of discontinuity curves. For details, see paper.

Key insights

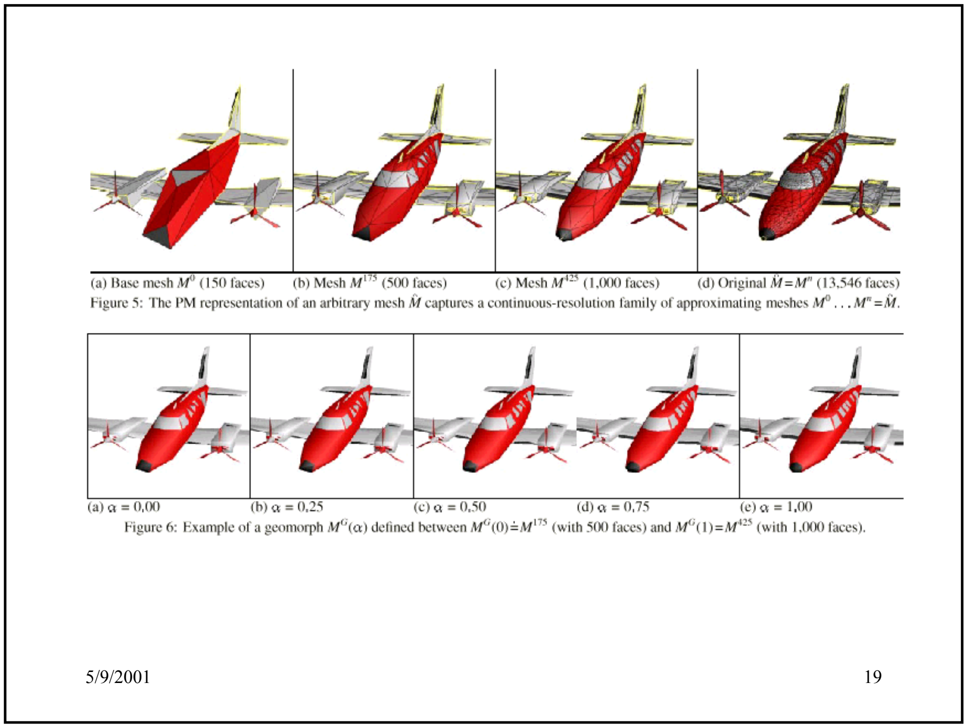
Can store information for vertex split that undoes edge collapse:

- original positions of vertices that were merged
- incident edges of original vertices
- attributes of corners and faces

Can represent original mesh by simplest (base) mesh and vertex splits.

Allows progressive transmission and selective refinement.

Can smoothly morph between meshes (obvious for consecutive meshes in sequence)

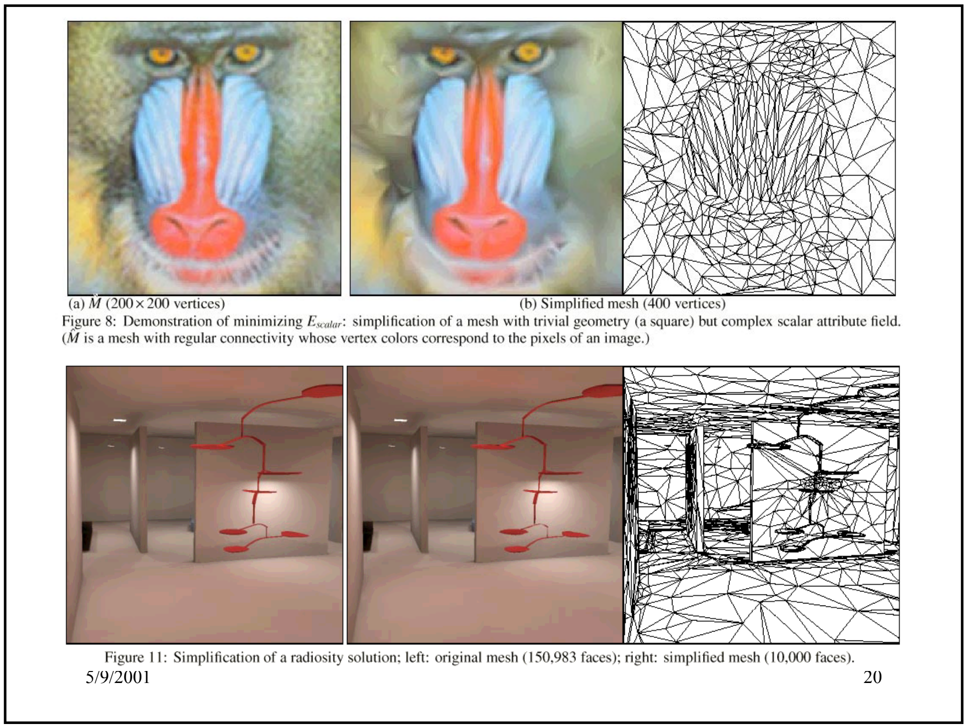


(a) Base mesh M^0 (150 faces) (b) Mesh M^{175} (500 faces) (c) Mesh M^{425} (1,000 faces) (d) Original $\hat{M}=M^n$ (13,546 faces)

Figure 5: The PM representation of an arbitrary mesh $\hat{M}$ captures a continuous-resolution family of approximating meshes $M^0 \ldots M^n=\hat{M}$.

(a) α = 0,00 (b) α = 0,25 (c) α = 0,50 (d) α = 0,75 (e) α = 1,00

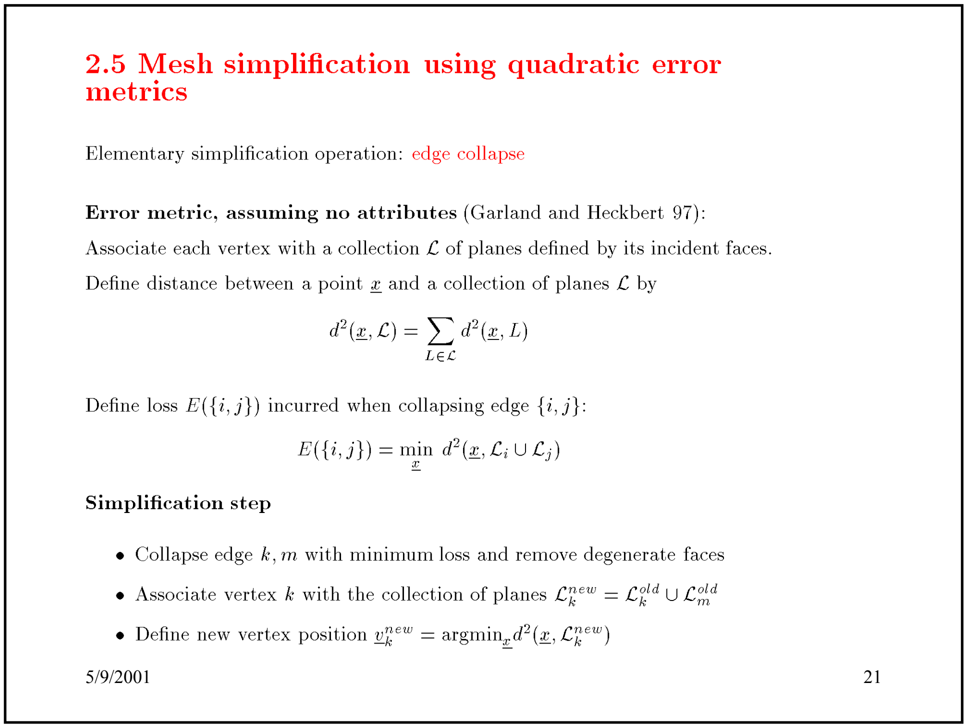
Figure 6: Example of a geomorph $M^G(\alpha)$ defined between $M^G(0)\doteq M^{175}$ (with 500 faces) and $M^G(1)=M^{425}$ (with 1,000 faces).



(a) $\hat{M}$ (200×200 vertices) (b) Simplified mesh (400 vertices)

Figure 8: Demonstration of minimizing E_{scalar}: simplification of a mesh with trivial geometry (a square) but complex scalar attribute field. ($\hat{M}$ is a mesh with regular connectivity whose vertex colors correspond to the pixels of an image.)

Figure 11: Simplification of a radiosity solution; left: original mesh (150,983 faces); right: simplified mesh (10,000 faces).



2.5 Mesh simplification using quadratic error metrics

Elementary simplification operation: edge collapse

Error metric, assuming no attributes (Garland and Heckbert 97):

Associate each vertex with a collection $\mathcal{L}$ of planes defined by its incident faces.

Define distance between a point $\underline{x}$ and a collection of planes $\mathcal{L}$ by

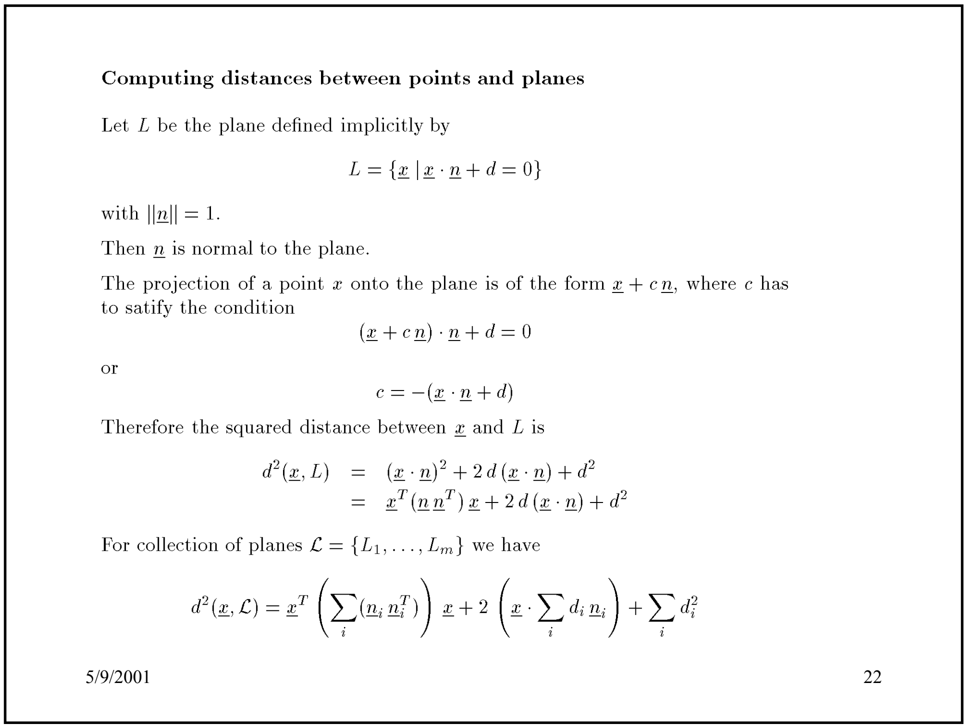
$$d^2(\underline{x}, \mathcal{L}) = \sum_{L \in \mathcal{L}} d^2(\underline{x}, L)$$

Define loss $E(\{i,j\})$ incurred when collapsing edge $\{i,j\}$:

$$E(\{i,j\}) = \min_{\underline{x}} \; d^2(\underline{x}, \mathcal{L}_i \cup \mathcal{L}_j)$$

Simplification step

- Collapse edge k, m with minimum loss and remove degenerate faces
- Associate vertex k with the collection of planes $\mathcal{L}_k^{new} = \mathcal{L}_k^{old} \cup \mathcal{L}_m^{old}$
- Define new vertex position $\underline{v}_k^{new} = \mathrm{argmin}_{\underline{x}} d^2(\underline{x}, \mathcal{L}_k^{new})$



Computing distances between points and planes

Let L be the plane defined implicitly by

$$L = \{\underline{x} \mid \underline{x} \cdot \underline{n} + d = 0\}$$

with $||\underline{n}|| = 1$.

Then $\underline{n}$ is normal to the plane.

The projection of a point x onto the plane is of the form $\underline{x} + c\,\underline{n}$, where c has to satisfy the condition

$$(\underline{x} + c\,\underline{n}) \cdot \underline{n} + d = 0$$

or

$$c = -(\underline{x} \cdot \underline{n} + d)$$

Therefore the squared distance between $\underline{x}$ and L is

$$\begin{aligned} d^2(\underline{x}, L) &= (\underline{x} \cdot \underline{n})^2 + 2\,d\,(\underline{x} \cdot \underline{n}) + d^2 \\ &= \underline{x}^T (\underline{n}\,\underline{n}^T)\,\underline{x} + 2\,d\,(\underline{x} \cdot \underline{n}) + d^2 \end{aligned}$$

For collection of planes $\mathcal{L} = \{L_1, \ldots, L_m\}$ we have

$$d^2(\underline{x}, \mathcal{L}) = \underline{x}^T \left(\sum_i (\underline{n}_i\,\underline{n}_i^T) \right) \underline{x} + 2 \left(\underline{x} \cdot \sum_i d_i\,\underline{n}_i \right) + \sum_i d_i^2$$

Recall:

$$d^2(\underline{x}, \mathcal{L}) = \underline{x}^T \left(\sum_i (\underline{n}_i \, \underline{n}_i^T) \right) \underline{x} + 2 \left(\underline{x} \cdot \sum_i d_i \, \underline{n}_i \right) + \sum_i d_i^2$$

Therefore:

- Easy to find $\underline{x}$ minimizing $d^2(\underline{x}, \mathcal{L})$ (quadratic function)
- No need to remember all normals and offsets associated with a vertex; enough to keep $\sum \underline{n}_i \, \underline{n}_i^T$, $\sum d_i \, \underline{n}_i$, and $\sum d_i^2$

Note: Iso-surfaces of $d^2(\underline{x}, \mathcal{L})$ are ellipses.

Lengths and directions of principal axes are related to mesh curvature.

Note: Quadratic form can be degenerate $\Rightarrow$ $\text{argmin}_{\underline{x}} \, d^2(\underline{x}, \mathcal{L})$ does not exist.

In this case pick the best of the two original vertex positions for the new vertex.

Results

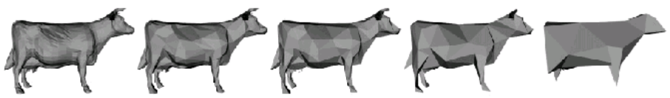


Figure 4: A sequence of approximations generated using our algorithm. The original model on the left has 5,804 faces. The approximations to the right have 994, 532, 248, and 64 faces respectively. Note that features such as horns and hooves continue to exist through many simplifications. Only at extremely low levels of detail do they begin to disappear.

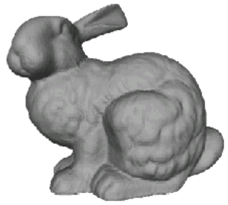


Figure 8: Original bunny model with 69,451 triangles. Rendered using flat shading just as in approximations below.

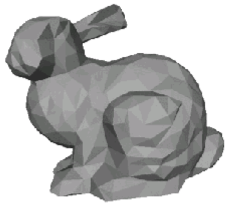


Figure 9: An approximation using only 1,000 triangles (generated in 15 seconds).

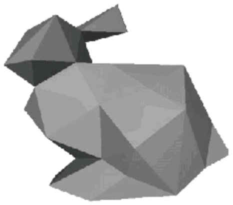


Figure 10: An approximation using only 100 triangles (generated in 15 seconds).

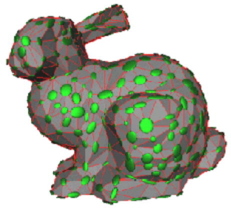


Figure 11: 1,000 face approximation. Error ellipsoids for each vertex are shown in green.

Incorporating surface attributes (Garland and Heckbert 98)

- Associate each vertex with a vector $\underline{v} = (\underline{p}, \underline{s}) \in R^{3+m}$.
  $\underline{p}$: geometric position;
  $\underline{s}$: attributes.
- Mesh now is a 2d surface in R^n. Everything generalizes in the obvious way.

Dealing with boundaries

- For each boundary edge, generate boundary plane orthogonal to incident face.
- Add boundary plane to plane collections associated with incident vertices.

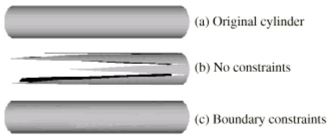


Figure 3: (a) Open-ended cylinder with 7,960 faces. (b) With unconstrained boundaries, this 2,460 face approximation quickly degenerates. (c) Using boundary constraints, the shape is preserved (also 2,460 faces).

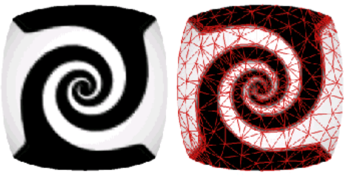


Figure 7: At left: a curved surface (18,050 faces) with colors at each vertex. At right: 1,000 face approximation. Notice that mesh edges follow the color contours.

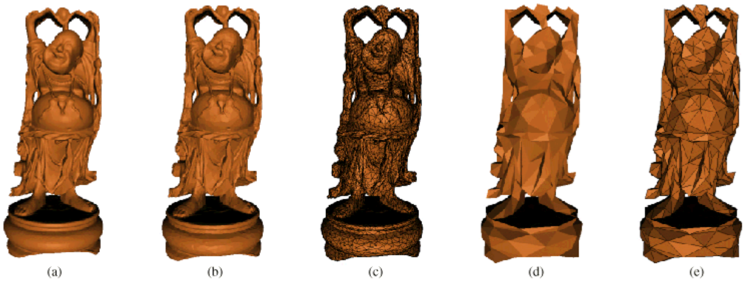


Figure 8: Simplifying geometry only: A very complex model of 1,085,634 faces (a) is simplified to 20,000 faces (b–c) and 1,000 faces (d–e).

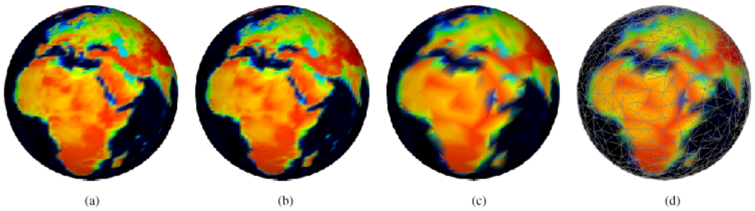


Figure 9: Simplifying geometry & color: A Gouraud-shaded surface of 73,728 faces (a) is reduced to 20,000 faces (b) and 3,000 faces (c–d).

Hoppe's quadratic metric (Hoppe 99)

Recap of G & H:

Each vertex i of original mesh M is associated with quadratic function Q_i

$$Q_i(\underline{x}) = d^2(\underline{x}, \mathcal{L}_i) = \sum_{L \in \mathcal{L}_i} d^2(\underline{x}, L)$$

where $\mathcal{L}_i$ is the collection of planes defined by the faces incident to vertex i.

Note: If there are m vertex attributes, then L will be a plane in R^{3+m}

Loss $E(\{i,j\})$ incurred when collapsing edge $\{i,j\}$ is defined as

$$E(\{i,j\}) = \min_{\underline{x}} \left(Q_i(\underline{x}) + Q_j(\underline{x})\right)$$

Algorithm:

- Collapse edge $\{k,l\}$ with minimum loss
- $Q_k^{new} = Q_k^{old} + Q_l^{old}$
- $\underline{v}_k^{new} = \text{argmin}_{\underline{x}}\ Q_k^{new}(\underline{x})$

Hoppe changes definition of $Q_i(\underline{x})$.

Let $f = \{i,j,k\}$ be a face incident on vertex i.

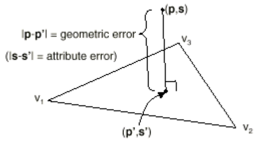
Will define quadratic function Q_i^f associated with vertex i and face f.

Let L be the plane in R^3 defined by the geometric positions $\underline{p}_i, \underline{p}_j, \underline{p}_k$ of the vertices.

Let $h(\underline{s})$ be the linear function on L defined by $h(\underline{p}_i) = \underline{s}_i$, $h(\underline{p}_j) = \underline{s}_j$, $h(\underline{p}_k) = \underline{s}_k$.

Define

$$Q_i^f(\underline{x}) = Q_I^f((\underline{p}, \underline{s})) = d^2(\underline{p}, L) + ||\underline{s} - h(P_L(\underline{p}))||^2$$



where $P_L(\underline{p})$ = projection of $\underline{p}$ on plane L.

So: $Q_i^f((\underline{p}, \underline{s})) =$

squared distance of geometric position $\underline{p}$ to plane L defined by face f +
squared difference between attribute vector $\underline{s}$ and its predicted value at $P_L(\underline{p})$.

Quadratic function $Q_i(\underline{x}) = \sum_f Q_i^f(\underline{x})$, where the sum is over the faces incident on vertex i.

Additional enhancements suggested in Hoppe (99)

Deal with attribute discontinuities by associating attributes with (vertex, face) pairs (wedges).

Memorlyless simplification: At each step, define quadratic functions Q_i based on current mesh, not on original mesh.

Surprise: Can work better.

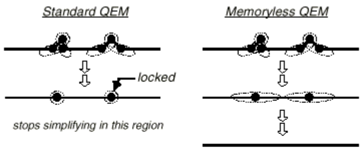
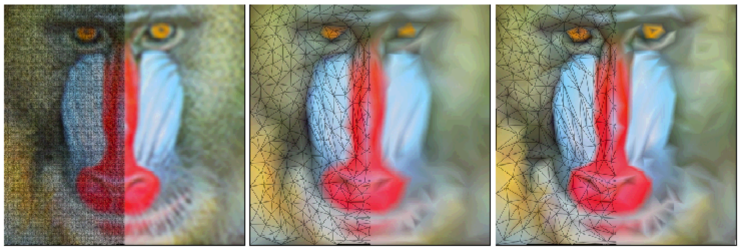
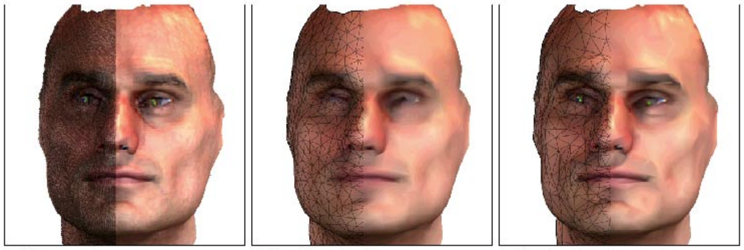


Figure 6: Illustration of standard QEM and memoryless QEM simplification. The dashed ovals symbolize the shapes of the quadric functionals Q_p^v; (a) in the standard scheme they are are computed once in a preprocess and subsequently summed during simplification; (b) in the memoryless scheme they are computed using the mesh simplified so far.



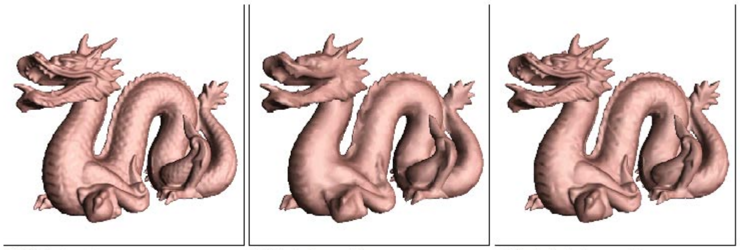
(a) Original mesh (79,202 faces) (b) Simplified using Q from [7] (c) Simplified using our new Q

Figure 3: Result of simplifying a vertex-colored 200×200 mesh down to 1,000 faces using the previous QEM [7] and using our new QEM. Mesh edges are rendered on the left half of each image. Weights ${}_i$ relating color to geometric accuracy are set to 1.



(a) Original mesh (b) Simplified using [7] (c) Simplified using our scheme

Figure 8: Simplification of a vertex-colored mesh of 135,133 faces down to 1,500 faces.



(a) Original mesh

(b) Q is just geometric error

(c) Q also includes normals

Figure 11: Simplification of a mesh of 920,000 faces down to 10,000 faces. For the geometric simplification in (b), normals are simply carried through. In (c) we optimize both geometry and normals, using _ = 0.02 for normals.