CS271 Computer Graphics II

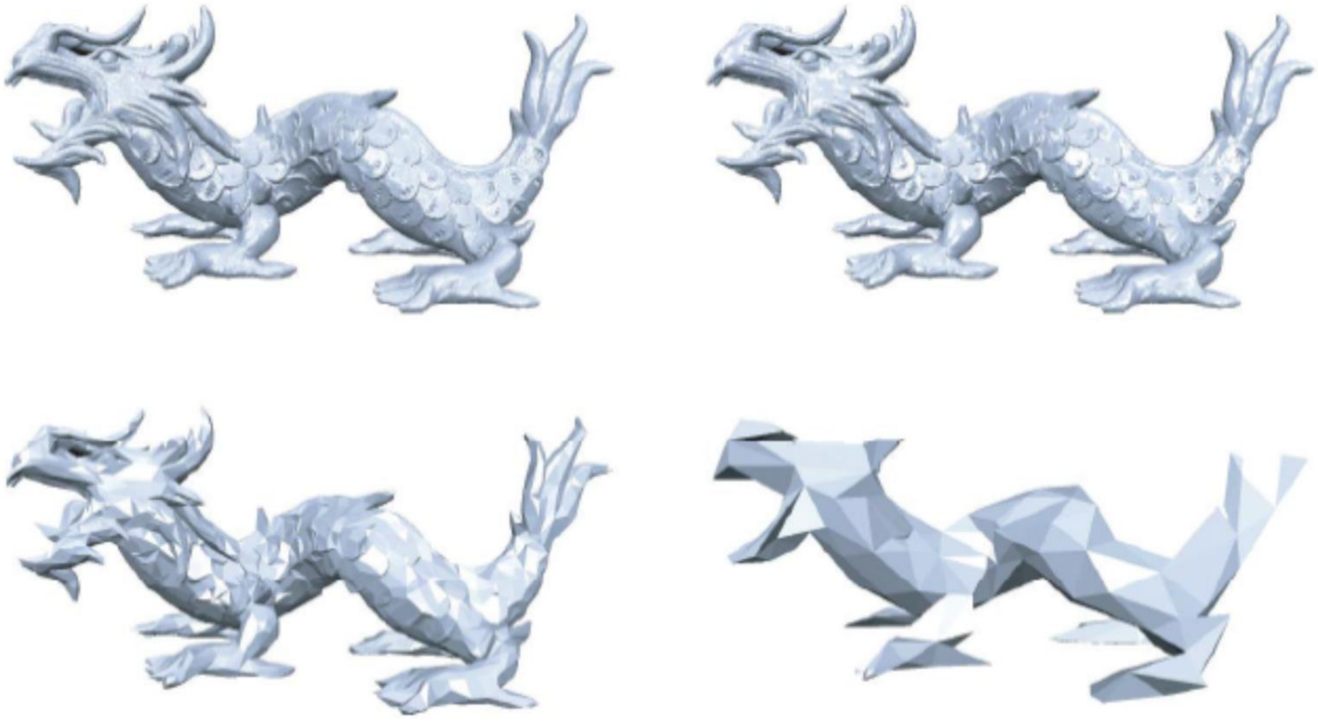
Lecture 6

Mesh Simplification

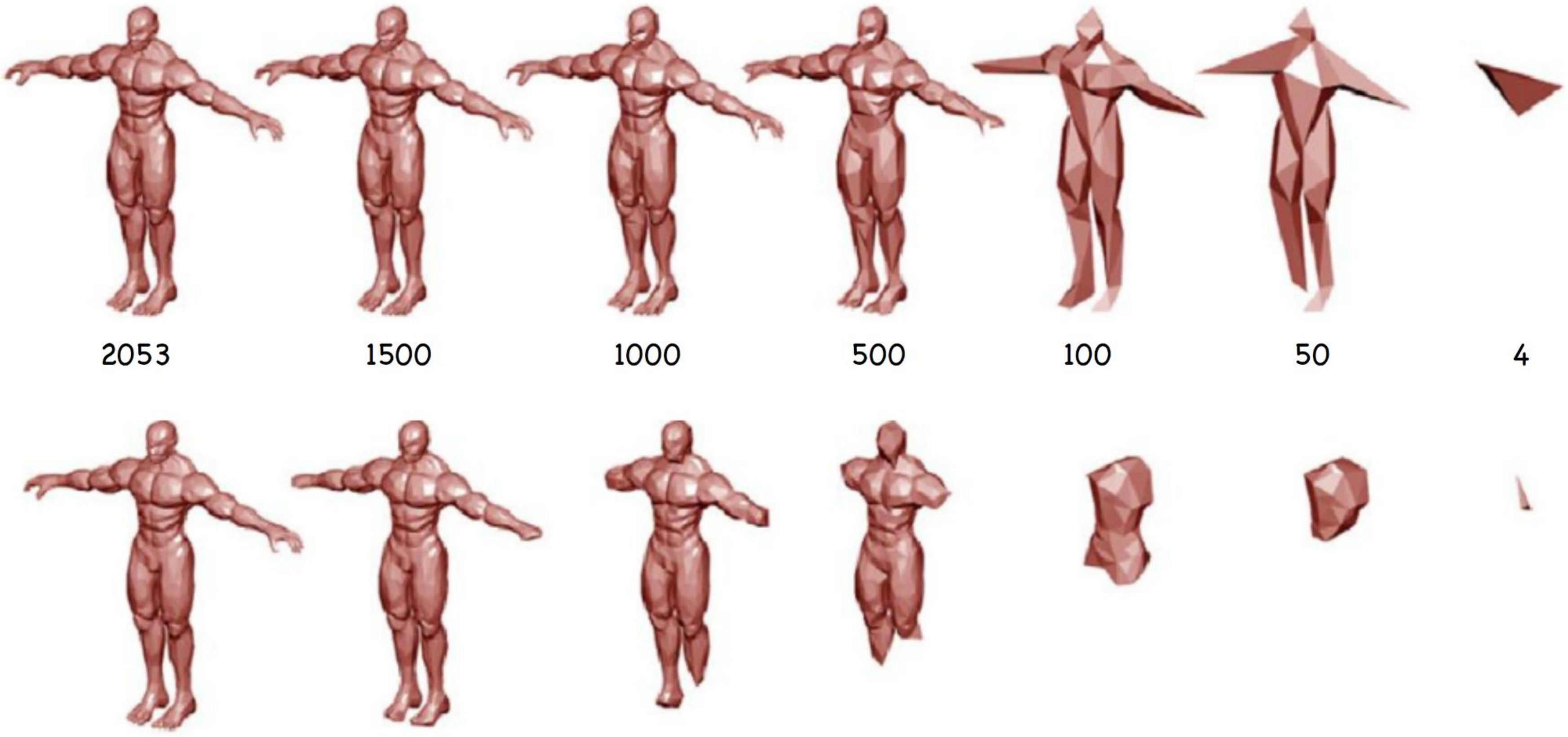
Definition

Simplification, decimation, approximation, downsampling

- Transform a given polygonal mesh into another mesh with fewer faces, edges, and vertices
- The simplification or approximation procedure is usually controlled by user-defined quality criteria

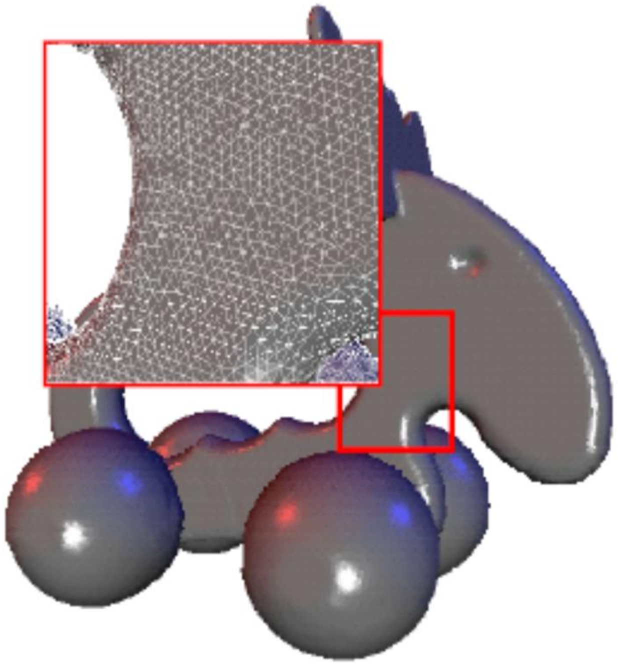


Curvature-preserved vs. Curvature-removed Criteria

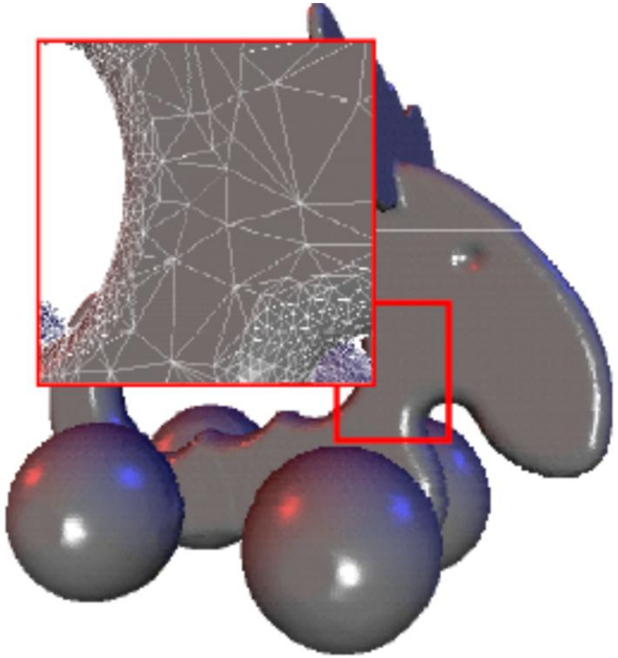


Mesh Simplification Applications

- Oversampled 3D scan data



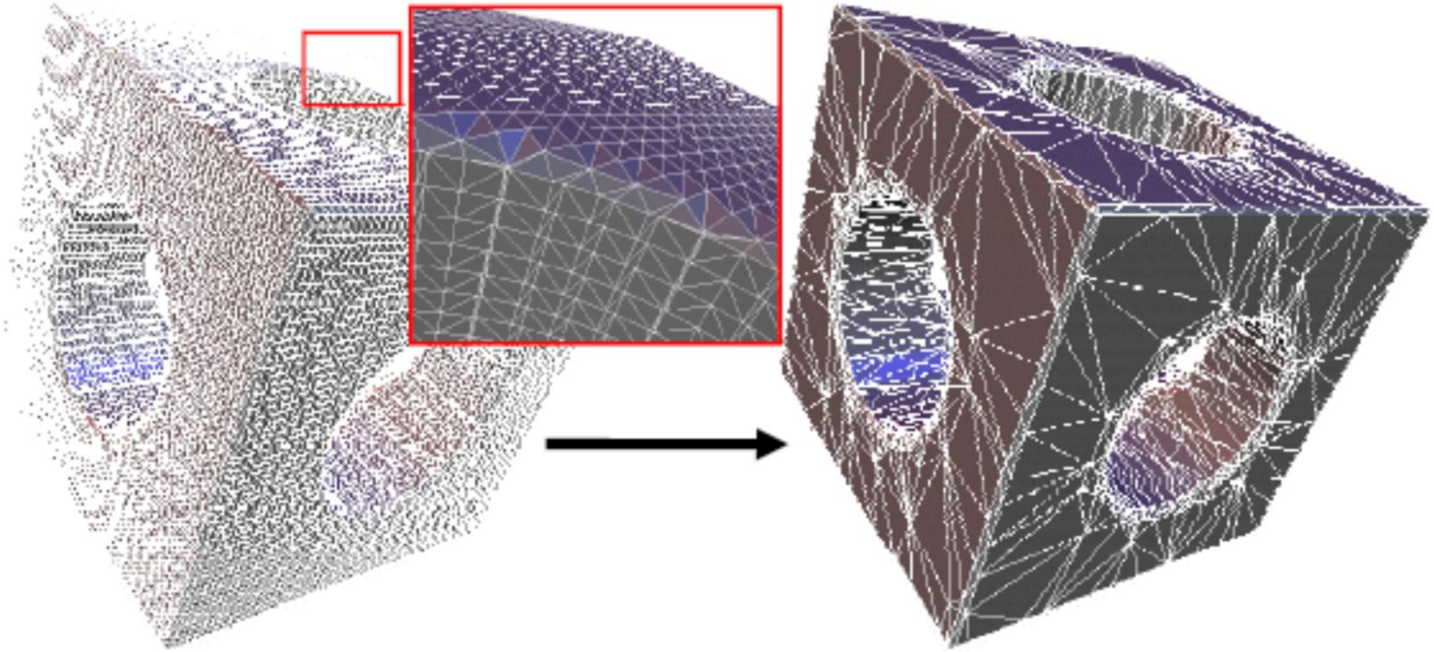
~150k triangles



~80k triangles

Mesh Simplification Applications

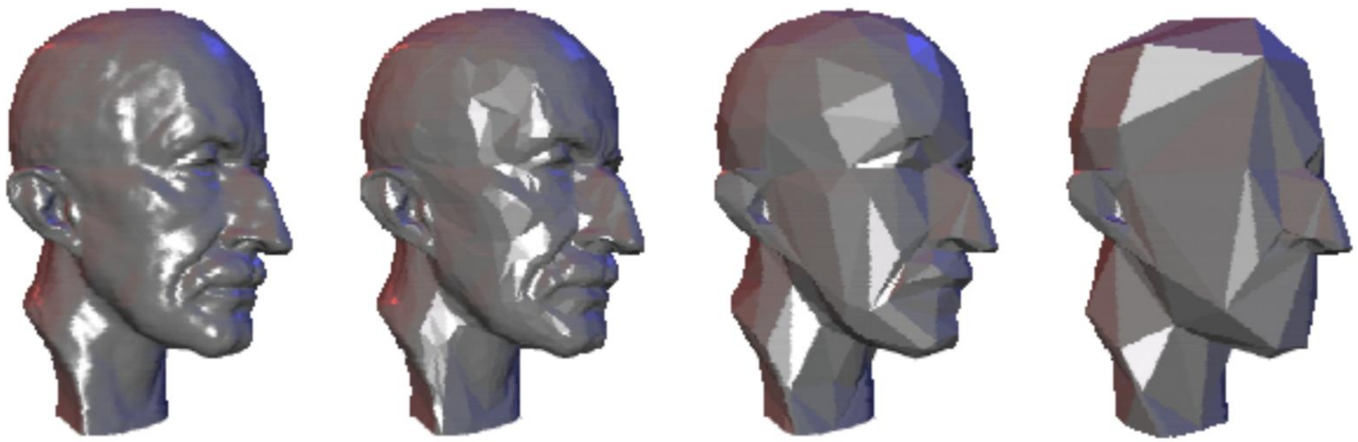
- Over-tessellation: e.g., iso-surface extraction



Mesh Simplification Applications

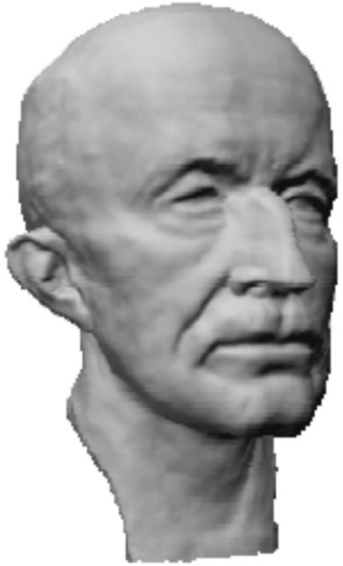
Multi-resolution hierarchies for

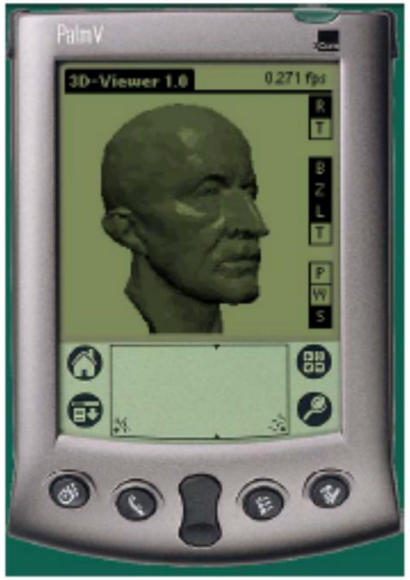
- Efficient geometry processing
- Level-of-detail (LOD) rendering



Mesh Simplification Applications

- Adaption to hardware capabilities





Mesh Simplification

Adjust the complexity of a geometry data set

- Since many decimation schemes work iteratively, i.e., they decimate a mesh by removing one vertex at a time, they usually can be inverted
- Hierarchical method

Problem Statement

- Given:M = (V, F)
- Find: M' = (V', F') such that
- $|V'| = n < |V|$ and $\|M - M'\|$ is minimal, or
- $\|M - M'\| < \varepsilon$ and $|V'|$ is minimal

Respect additional fairness criteria

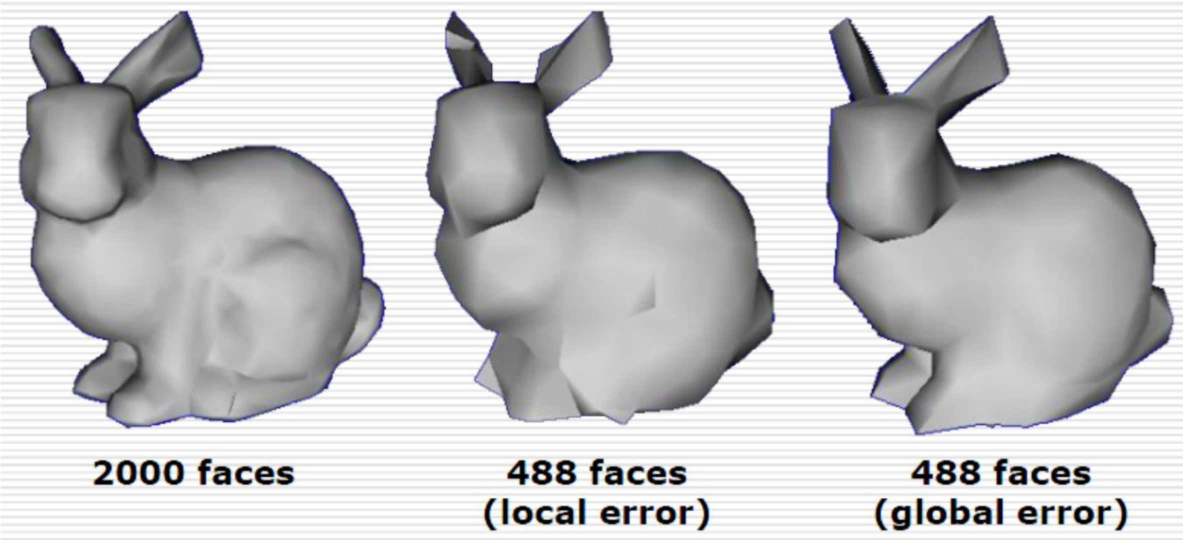
- Normal deviation, triangle shape, scalar attributes, etc.

Mesh Simplification

Start with the original fine mesh

- **Simply progressively**
  E.g., collapse edges, vertex clustering

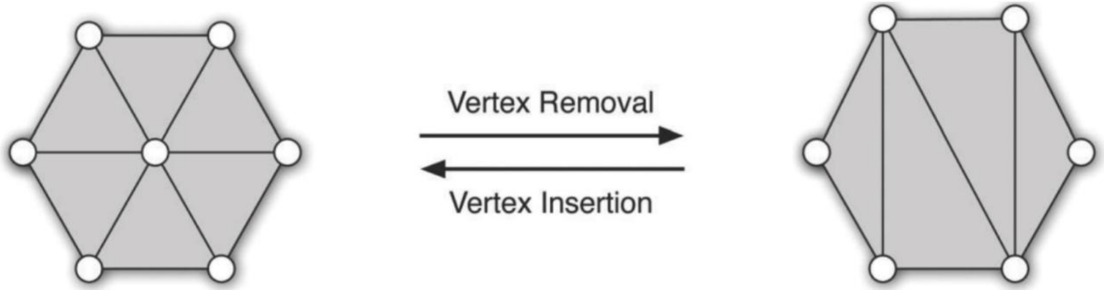
- **Aim to keep original appearance**
  Normal deviation, triangle shape, scalar attributes
  Error control



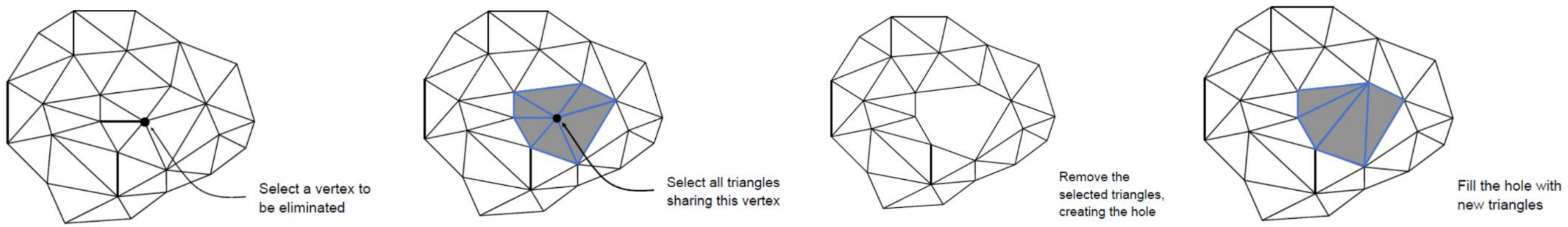
Local Operations

Local Simplification Operator

- Vertex Removal



- Deletion & Insertion
- Reversible



Select a vertex to be eliminated

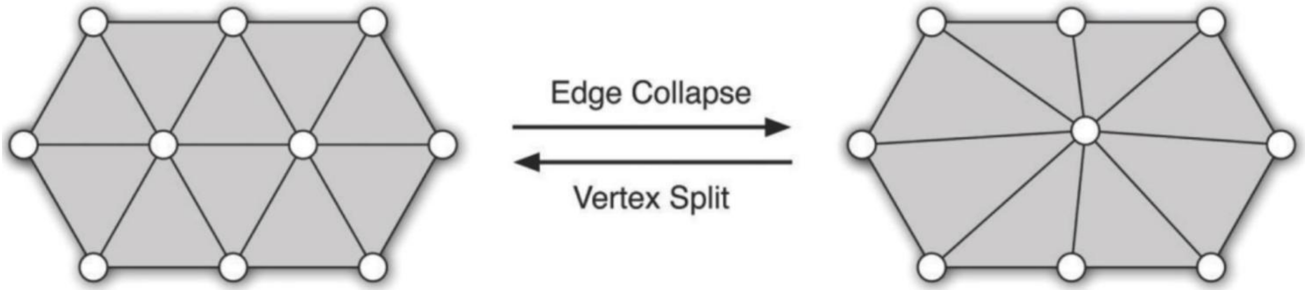
Select all triangles sharing the vertex

Remove the selected triangles, creating the hole

Fill the hole with new triangles

Local Simplification Operator

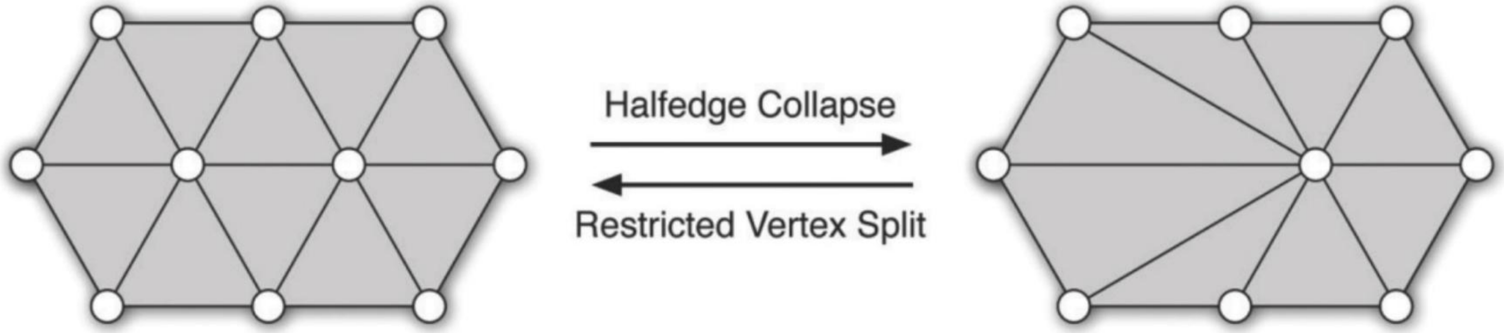
- **Edge Collapse**



- Merge two adjacent vertices
- Simple to implement
- Well-suited for implementing geomorphing

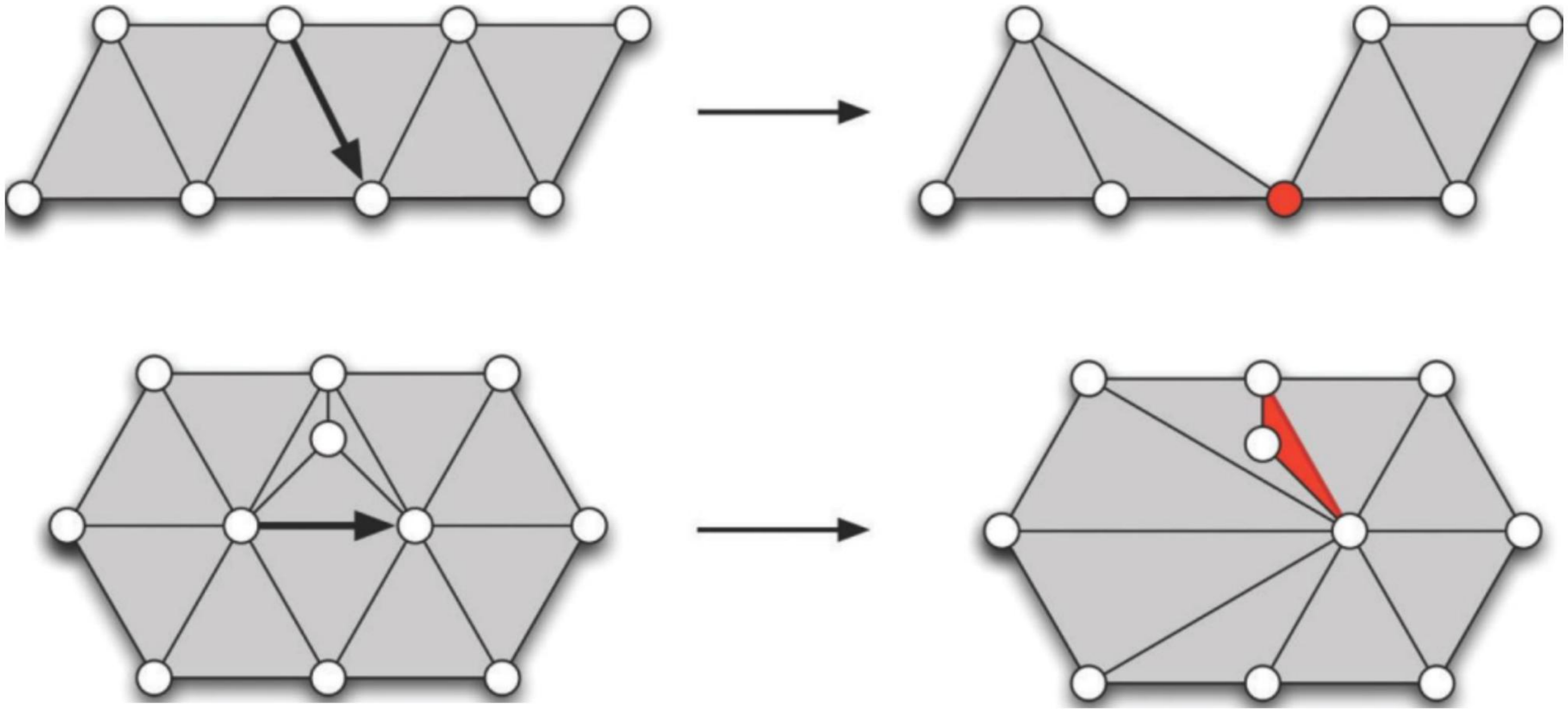
Local Simplification Operator

- **Half-edge Collapse**



- **Collapse edge into one end point**
  Special case of vertex removal
  Special case of edge collapse
- After collapse: n(E) - 3, n(V) – 1, n(F) – 2
- According to Euler Formula: unchanged
- Half-edge collapsing would not change the genus of a mesh
- Should determine whether collapse is ok (may introduce non-manifold structure)

Topologically Illegal (half-)edge Collapses



Simplification via Edge Collapse

One popular scheme: iteratively collapse edges

Greedy algorithm from a general overview:

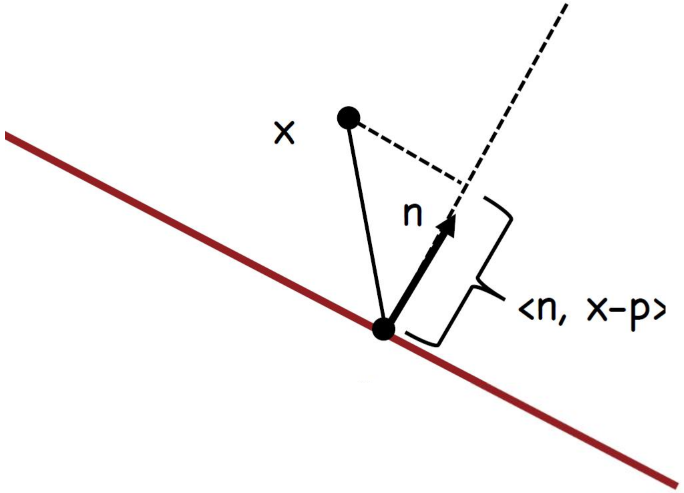
- Assign each edge a cost
- Collapse edge with least cost
- Repeat until target number of elements is reached
- Particularly effective cost function: **Quadric Error Metric**

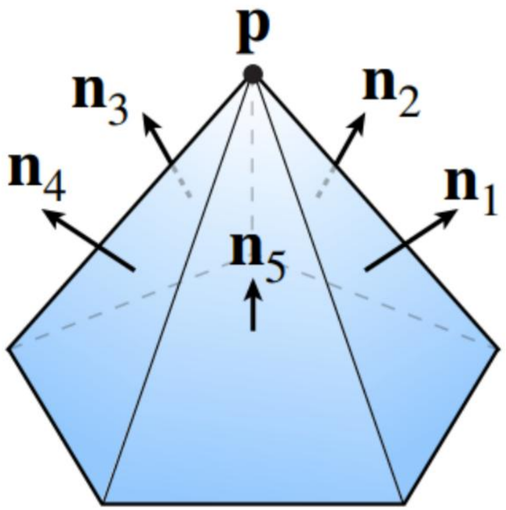
Quadric Error Metric

Quadric Error Metric (QEM)

Approximate distance to a collection of triangles

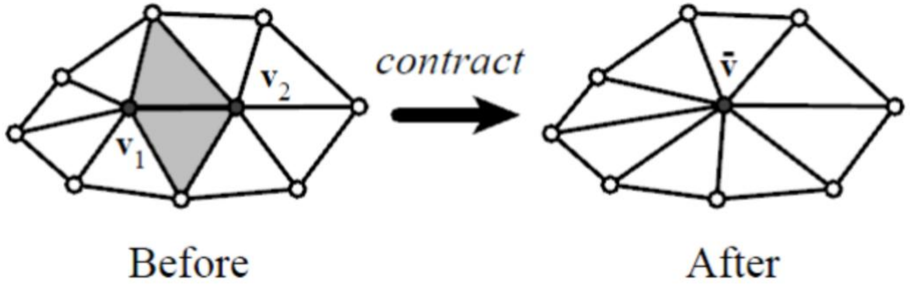
- Q: Distance to plane w/ normal n passing through point p?
- A: dist(x) = <n, x> - <n, p> = <n, x-p>
- Quadric error is then sum of squared point-to-plane distances





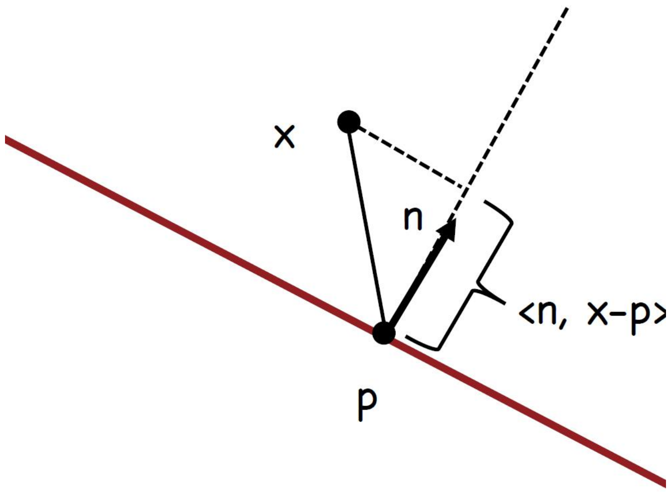
$$Q(x) := \sum_{i=1}^{k} < n_i, x - p >^2$$

$$Q^e = Q_1^v + Q_2^v$$



Quadric Error – Homogeneous Coordinates

Suppose in coordinates we have
- A query point **x = (x, y, z)**
- A normal **n = (a, b, c)**
- An offset **d := <n, p>**



In **homogeneous coordinates**, let
- **u := (x, y, z, 1)**
- **v := (a, b, c, d)**

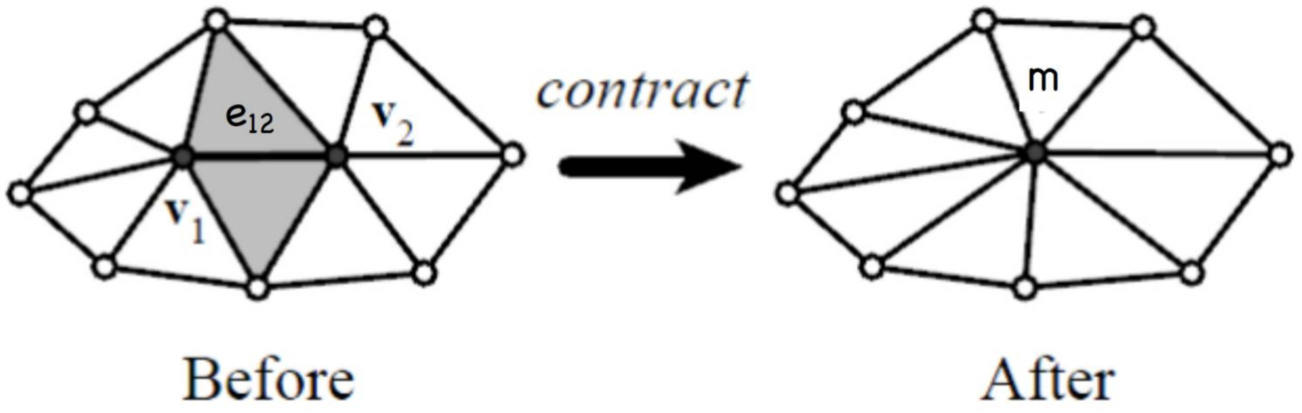
- Signed distance to plane is then just **<u, v> = ax + by + cz + d**
- Squared distance is **$\langle u, v\rangle^2 = u^T(vv^T)u =: u^TKu$**
- Matrix **$K = vv^T$** encodes squared distance to plane

$$K = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

Key idea: sum of matrices K distance to union of planes
$u^TK_1u + u^TK_2u = u^T(K_1+K_2)u$

Quadric Error of Edge Collapse

- How much does it cost to collapse an edge e12?
  Idea: compute midpoint m, measure error $Q(m) = m^T(K_1+K_2)m$
- Error becomes "score" for e_{12}, determining priority

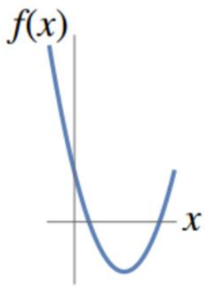


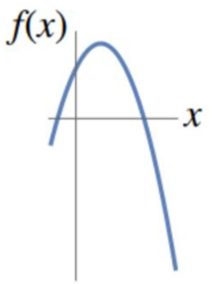
- Better idea: find point x that minimize error!
- But how to minimize quadric error?

Revisit: Minimizing a Quadratic Function

Suppose you have a function $f(x) = ax^2 + bx + c$

- Q: What does the graph of this function look like?
- Q: How do we find the minimum?





- A: Find where the function looks "flat" if we zoom in really close
- i.e., find point x where 1st derivative vanishes:

$$f'(x) = 0$$
$$2ax + b = 0$$
$$x = -b/2a$$

Minimizing Quadratic Polynomial

Not much harder to minimize a quadratic polynomial in **n** variables

- Can always write in terms of a symmetric matrix A
- E.g., in 2D: $\mathbf{f(x, y) = ax^2 + bxy + cy^2 + dx + ey + g}$

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \quad A = \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} d \\ e \end{bmatrix}$$

$$\mathbf{f(x, y) = x^T A x + u^T x + g}$$

(will have the same form for any **n**)

- **Q: How do we find a critical point (min/max/saddle)?**
- **A: Set derivative to zero!**

$$2A\mathbf{x} + \mathbf{u} = 0$$

$$\mathbf{x = -1/2\ A^{-1}u}$$

立志 成才 报国 裕民

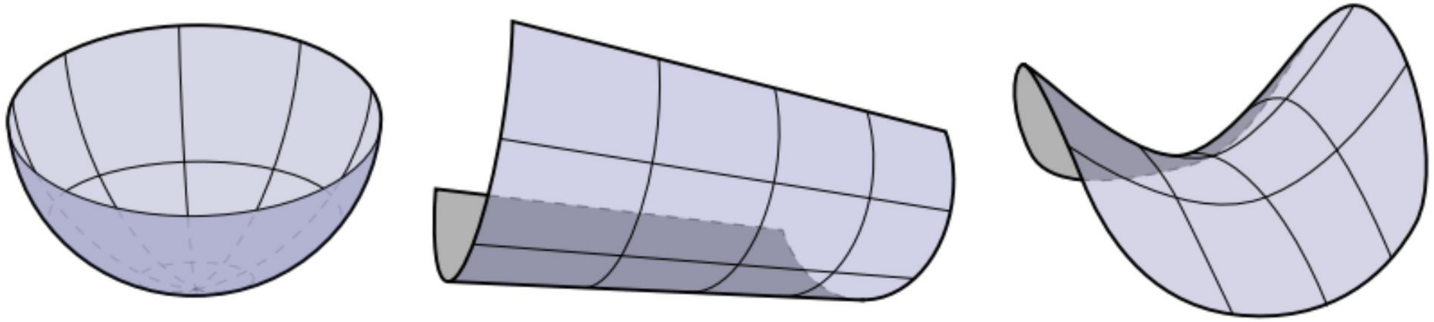
Positive Definite Quadratic Form

Just like our 1D parabola, critical point is not always a min!

- Q: In 2D, 3D, nD, when do we get a minimum?
- A: When matrix A is positive-definite:

$$\mathbf{X}^T\mathbf{A}\mathbf{X} > 0 \quad \forall \mathbf{X}$$

- **1D:** Must have xax = $ax^2 > 0$, i.e., a is positive!
- **2D:** Graph of function looks like a "bowl":



positive definite　　positive semidefinite　　indefinite

Minimizing Quadric Error

Find “best” point for edge collapse by minimizing quadratic form

$$\min_{\mathbf{u}\in\mathbb{R}^4} \mathbf{u}^T K \mathbf{u}$$

- Already know 4th (homogeneous) coordinate for a point is 1
- So, break up our quadratic function into two pieces:

$$\begin{bmatrix} \mathbf{x}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} B & \mathbf{w} \\ \mathbf{w}^\mathsf{T} & d^2 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$
$$= \mathbf{x}^\mathsf{T} B \mathbf{x} + 2\mathbf{w}^\mathsf{T}\mathbf{x} + d^2$$

- Now we have a quadratic polynomial in the unknown position $\mathbf{x} \in \mathbb{R}^3$
- Can minimize as before:

$$2B\mathbf{x} + 2\mathbf{w} = 0 \quad \Longleftrightarrow \quad \mathbf{x} = -B^{-1}\mathbf{w}$$

QEM Simplification: Final Algorithm

- **Input: a mesh**
- **Output: a simplified mesh**

- **Initialization:**

  Compute **K** for each triangle (squared distance to plane)

  Set $\mathbf{K}_i$ at each vertex to sum of **Ks** from incident triangles

  For each edge $\mathbf{e}_{ij}$**:**

  Set $\mathbf{K}_{ij} = \mathbf{K}_i + \mathbf{K}_j$

  Find point x minimizing error, set cost to $\mathbf{K}_{ij}(\mathbf{x})$

- **Until we reach target number of triangles:**

  Collapse edge eij with smallest cost to optimal pont **x**

  Set quadric at new vertex to $\mathbf{K}_{ij}$

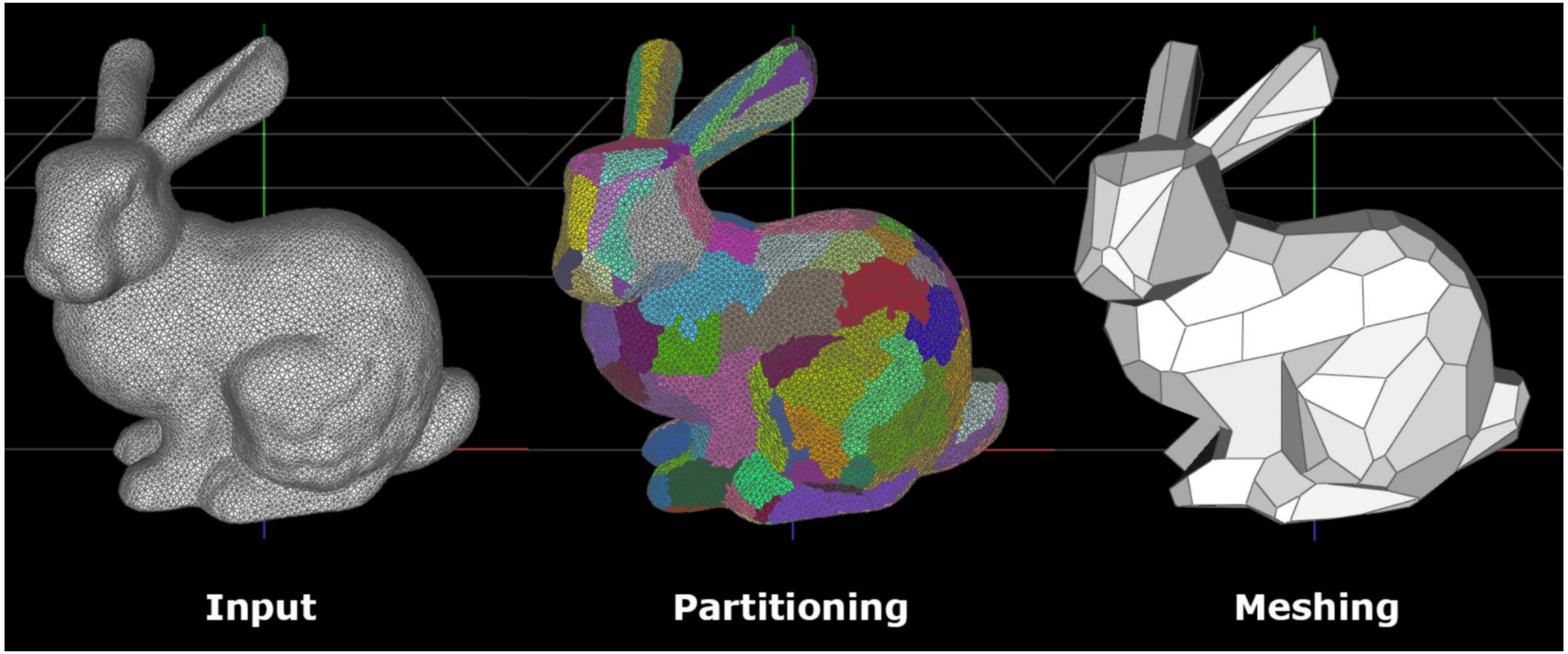
  Update cost of edges touching new vertex

Full Resolution	60,000 triangles	1000 triangles

Variational Shape Approximation

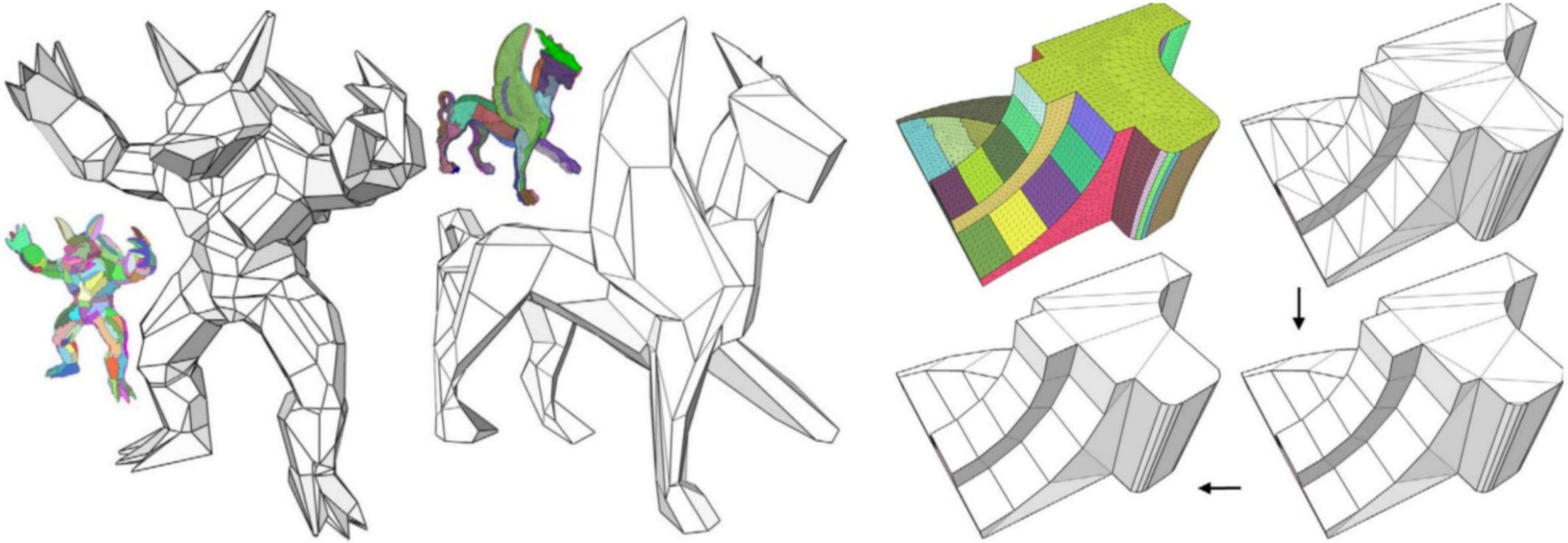
Variational Shape Approximation (VSA)

VSA is highly sensitive to features and symmetries and produces anisotropic meshes of high approximation quality



Variational Shape Approximation (VSA)

- The input shape is approximated by a set of proxies
- A plane in space through the point $\mathbf{x}_i$ with normal direction $\mathbf{n}_i$



Region Representation

- $\mathbf{M}$: a triangle mesh
- $\mathbf{R} = \{\mathbf{R}_1, \ldots, \mathbf{R}_k\}$: a partition of $\mathbf{M}$ into $\mathbf{k}$ regions $R_1 \cup \ldots \cup R_k = M$
- Proxies: $\mathbf{P} = \{\mathbf{P}_1, \ldots, \mathbf{P}_k\}$, $\mathbf{P}_i = (\mathbf{x}_i, \mathbf{n}_i)$

Distance metrics between $\mathbf{R}_i$ and $\mathbf{P}_i$

- The squared orthogonal distance of x from the plane $\mathbf{P}_i$

$$L^2(R_i, P_i) = \int_{x \in R_i} (n_i^T x - n_i x_i)^2 dA$$

- A measure of the normal field:

$$L^{2,1}(R_i, P_i) = \int_{x \in R_i} ||n(x) - n_i||^2 dA$$

Goal of VSA

Given a number $\mathbf{k}$ and an error metric $\mathbf{E(L_2 \text{ or } L_{2,1})}$, find a set $\mathbf{R} = \{\mathbf{R}_1, \ldots, \mathbf{R}_k\}$ of regions and a set $\mathbf{P} = \{\mathbf{P}_1, \ldots, \mathbf{P}_k\}$ of proxies such that the global distortion

$$E(R, P) = \sum_{i=1}^{k} E(R_i, P_i)$$

is minimized

Lloyd's Clustering Algorithm

上海科技大学 ShanghaiTech University

- The algorithm iteratively alternates between a **geometry partitioning** phase and a **proxy fitting** phase
- **Geometry partitioning phase**
  - A set of regions that best fit a **given** set of proxies
  - Modifies the set R of regions to achieve a lower approximation error while keeping the proxies P fixed
- **Proxy fitting phase**
  - The partitioning is kept **fixed,** and the proxies are adjusted to minimize approximation error
  - L^2 metric: the best proxy is the least-squares fitting plane
  - $L^{2,1}$ metric: the proxy normal n_i is just the area-weighted average of the triangle normals
- **Initialization**
  - Randomly picking k triangles as R
  - The planes of k triangles are used to initialize P

立志成才报国裕民

Related Papers

- Garland M, Heckbert P S. **Surface simplification using quadric error metrics**[C]//Proceedings of the 24th annual conference on Computer graphics and interactive techniques. 1997: 209-216.
- Cohen-Steiner D, Alliez P, Desbrun M. **Variational shape approximation**[M]//ACM SIGGRAPH 2004 Papers. 2004: 905-914.
- Skrodzki M, Zimmermann E, Polthier K. **Variational shape approximation of point set surfaces**[J]. Computer Aided Geometric Design, 2020, 80: 101875.
- Liu Y J, Xu C X, Fan D, et al. **Efficient construction and simplification of Delaunay meshes**[J]. ACM Transactions on Graphics (TOG), 2015, 34(6): 1-13.
- Yi R, Liu Y J, He Y**. Delaunay mesh simplification with differential evolution**[J]. ACM Transactions on Graphics (TOG), 2018, 37(6): 1-12.
- Liang Y, He F, Zeng X. **3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution**[J]. Integrated Computer-Aided Engineering, 2020, 27(4): 417-435.
- Liu H T D, Gillespie M, Chislett B, et al. **Surface simplification using intrinsic error metrics**[J]. ACM Transactions on Graphics, 2023, 42(4).