Simplification

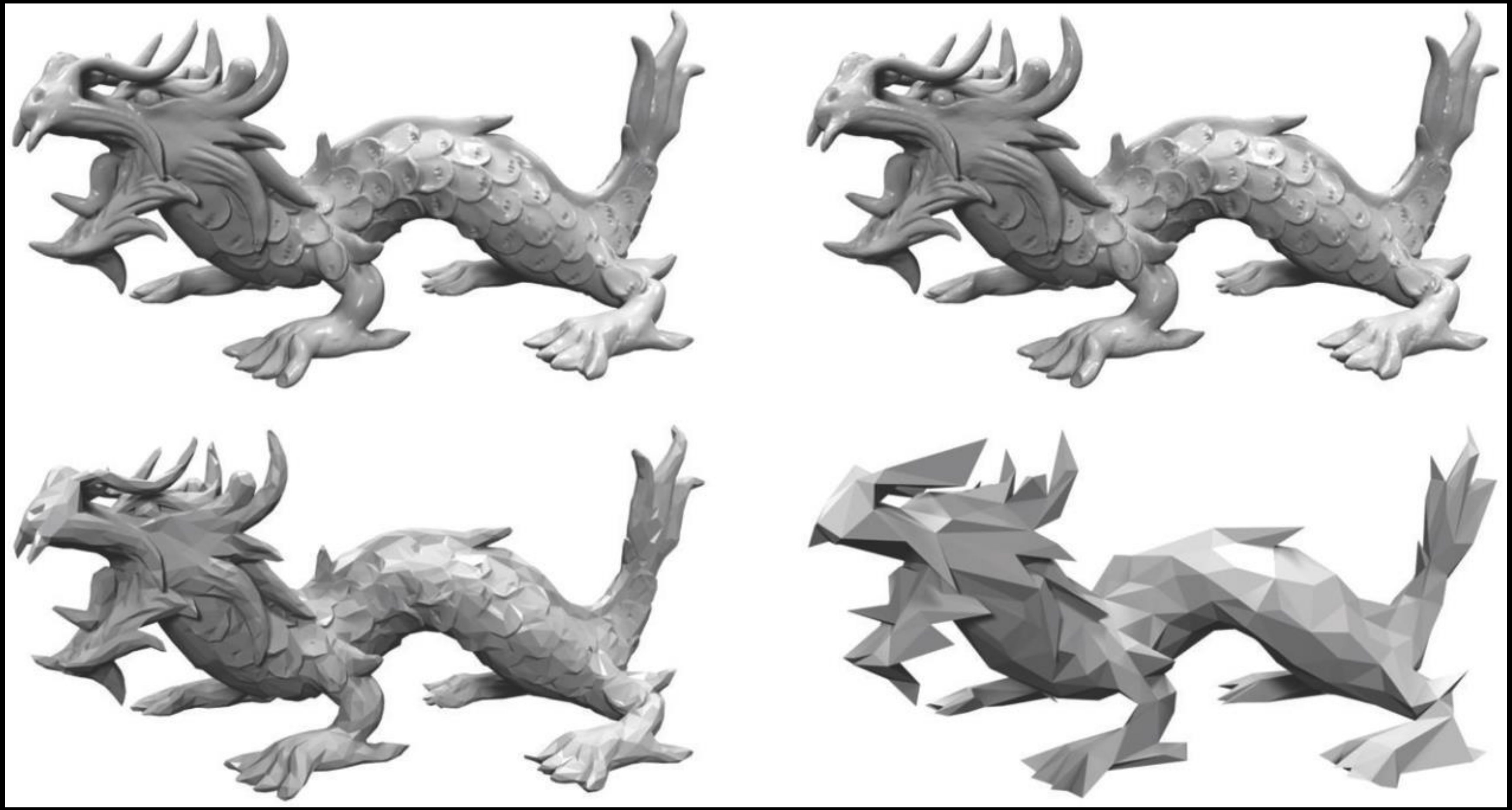
Xiao-Ming Fu

Outlines

- Definition
- Local operations
- Quadric error metric
- Variational shape approximation

Simplification and approximation

- Transform a given polygonal mesh into another mesh with **fewer** faces, edges, and vertices.

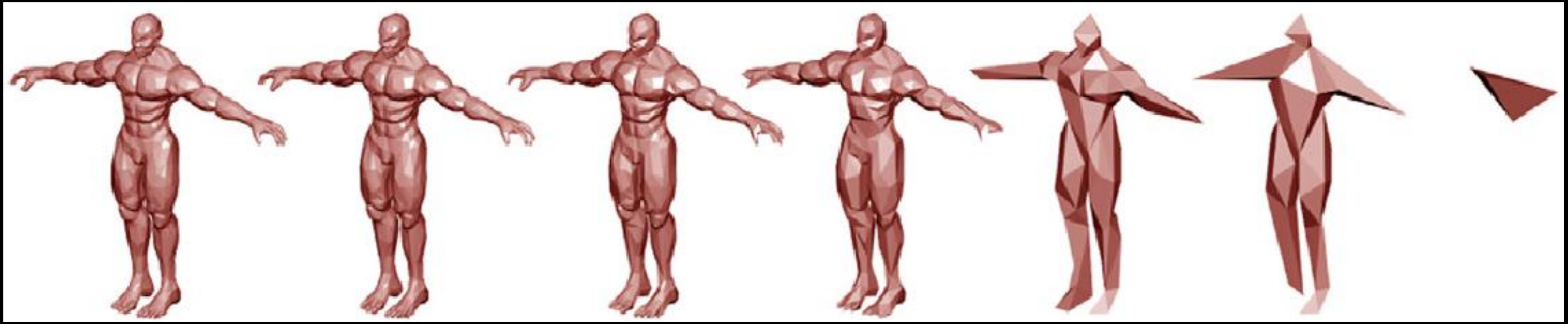


Simplification and approximation

- Transform a given polygonal mesh into another mesh with **fewer** faces, edges, and vertices.

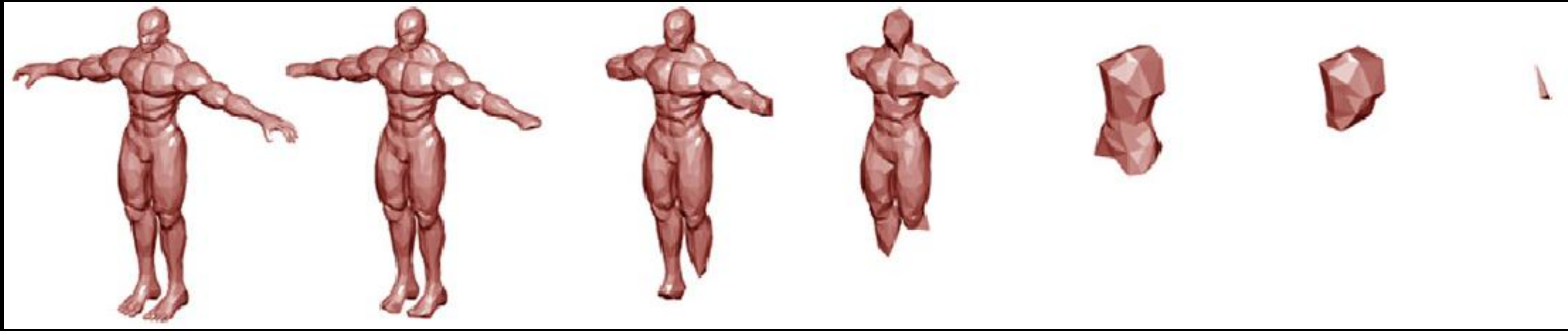
- The simplification or approximation procedure is usually controlled by user-defined **quality criteria**.

Curvature-preserved criteria



2053 1500 1000 500 100 50 4

Curvature-removed criteria



2053 1500 1000 500 100 50 4

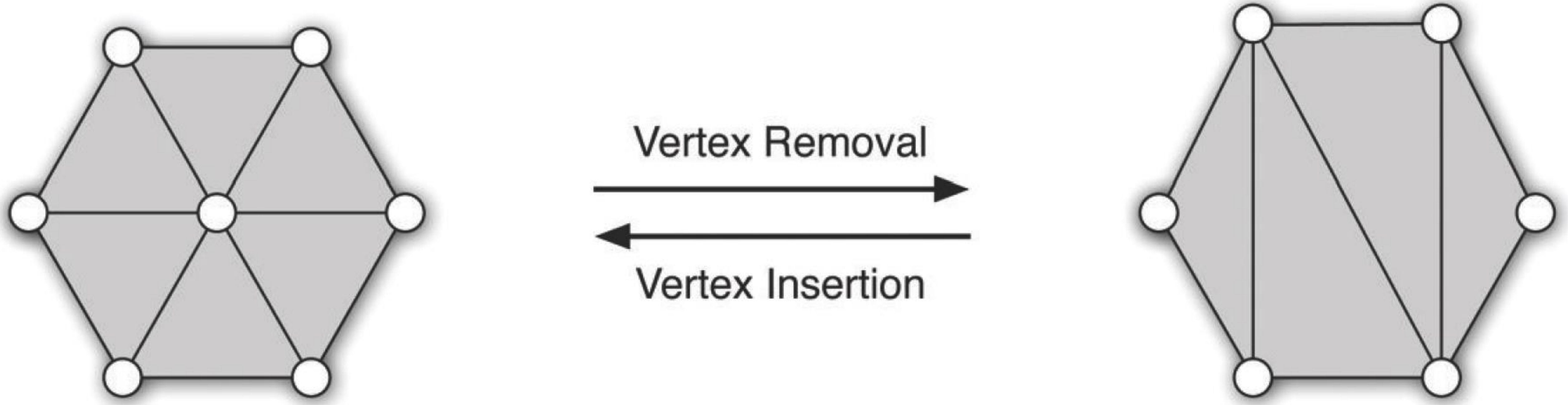
Simplification applications

- Adjust the complexity of a geometric 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

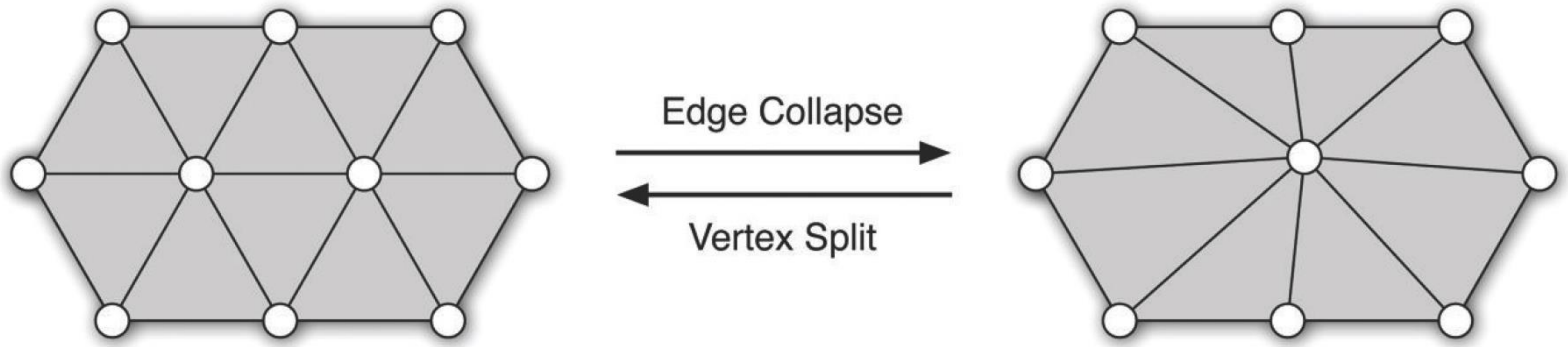
Outlines

- Definition
- Local operations
- Quadric error metric
- Variational shape approximation

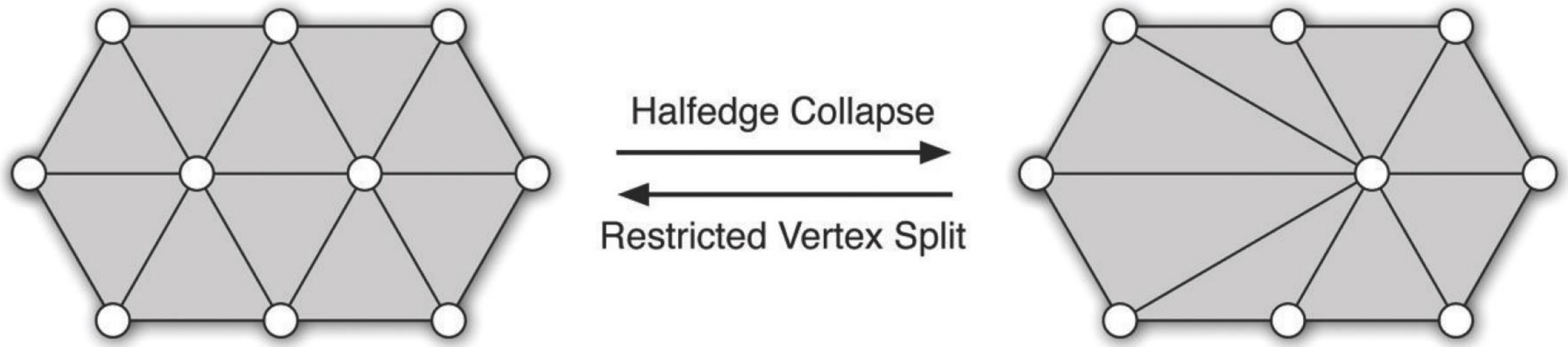
Vertex removal



Edge collapse



Half-edge collapse



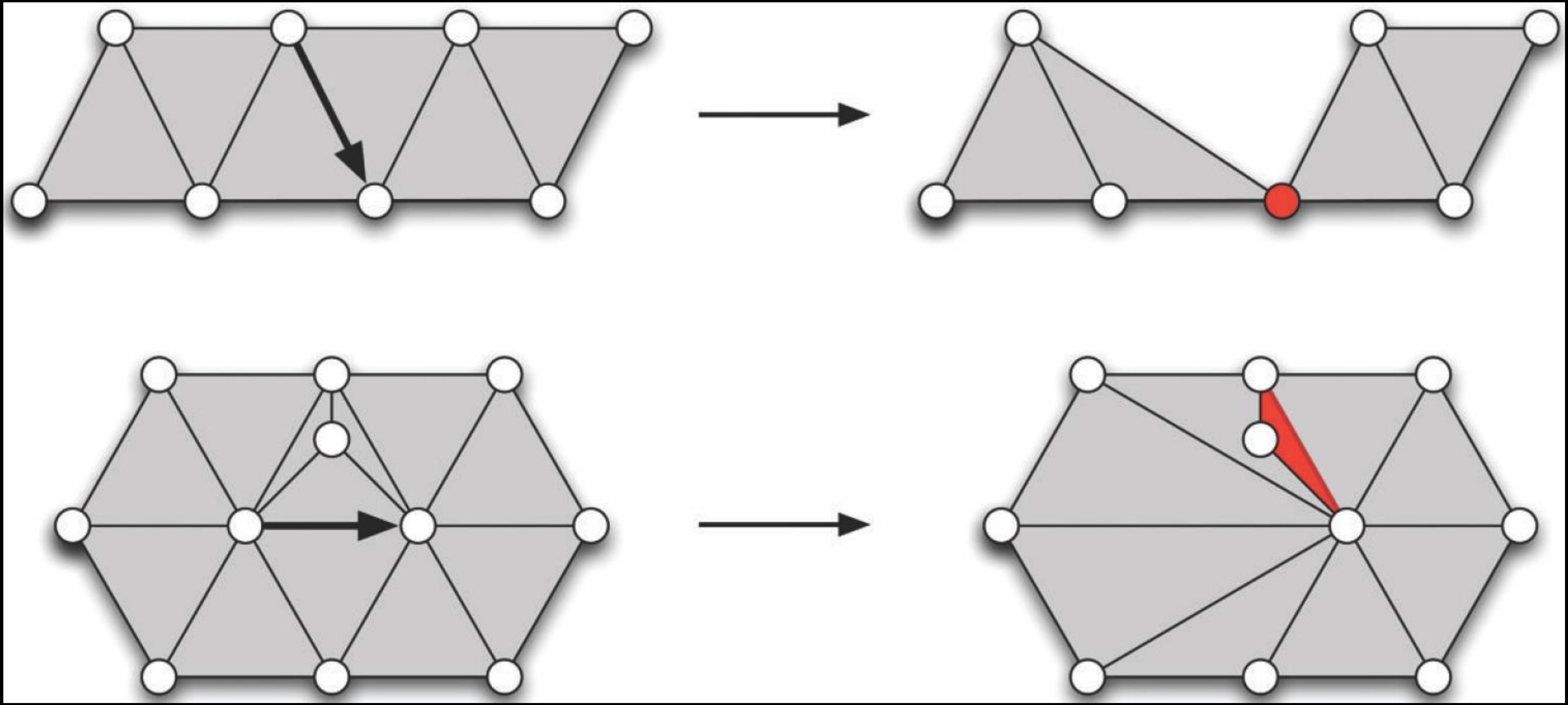
After collapse: n(E) – 3, n(V) – 1, n(F) – 2.

According to Euler' formula: 2 - 2m = n(V) + n(F) – n(E).

Half-edge collapsing would not change the genus of a mesh.

OpenMesh: collapse(), is_collapse_ok().

Topologically illegal (half-)edge collapses



Outlines

- Definition
- Local operations
- Quadric error metric
- Variational shape approximation

Incremental algorithms

- Removing one vertex at a time
- The iterative decimation procedure can take **arbitrary user-defined criteria** into account, according to **which the next removal operation is chosen**.

Quadric error metric (QEM)

- The squared distance of a point x from the plane P_i:

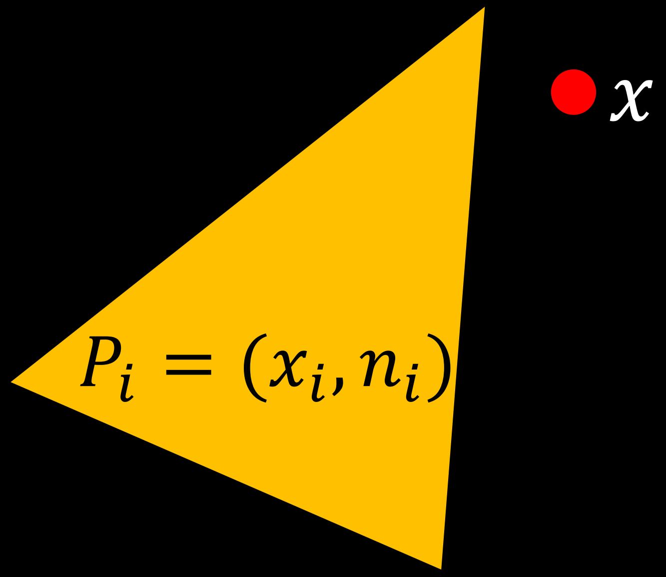
$$d(x, P_i) = \left(n_i^T x - d_i\right)^2$$
$$d_i = n_i^T x_i$$

Denote $\bar{x} = (x, 1)$ and $\overline{n_i} = (n_i, -d_i)$.

Then:

$$d(x, P_i) = \left(\overline{n_i}^T \bar{x}\right)^2 = \bar{x}^T \overline{n_i}\,\overline{n_i}^T \bar{x} =: \bar{x}^T Q_i \bar{x}$$

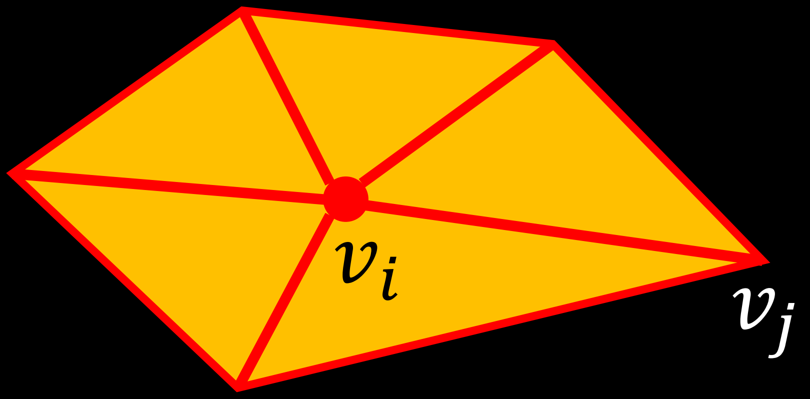
Quadratic error Matrix



Quadratic error Matrix Q

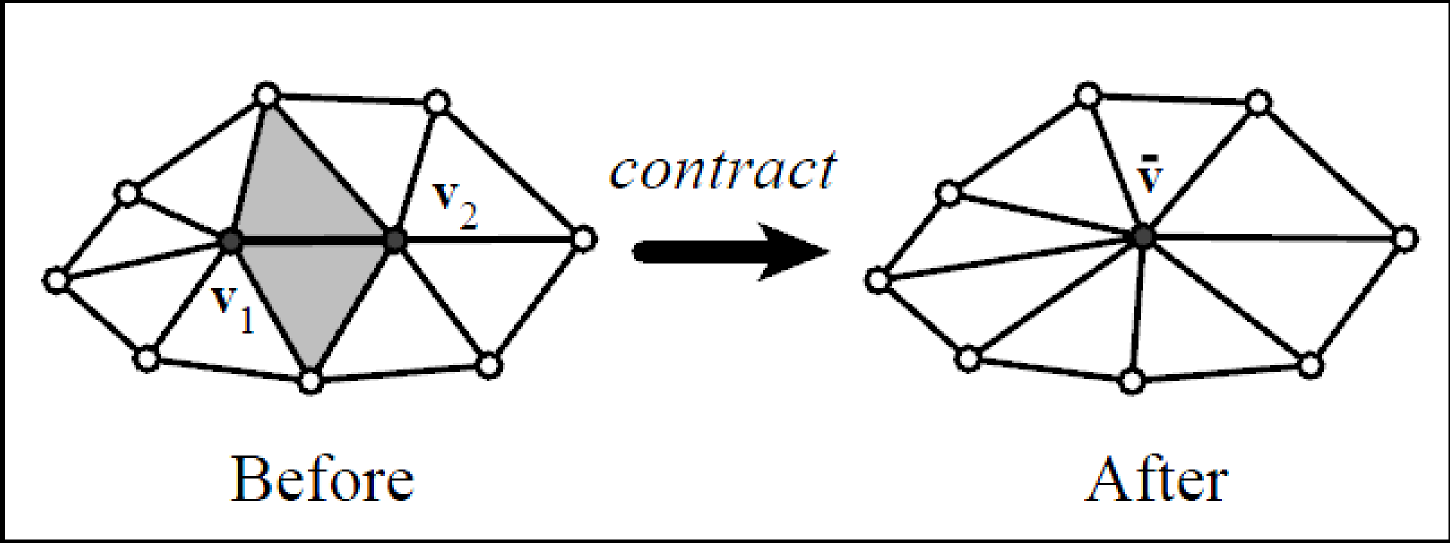
- On vertices

$$Q_i^v = \sum_{j \in \Omega(i)} Q_j$$



- On edge

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



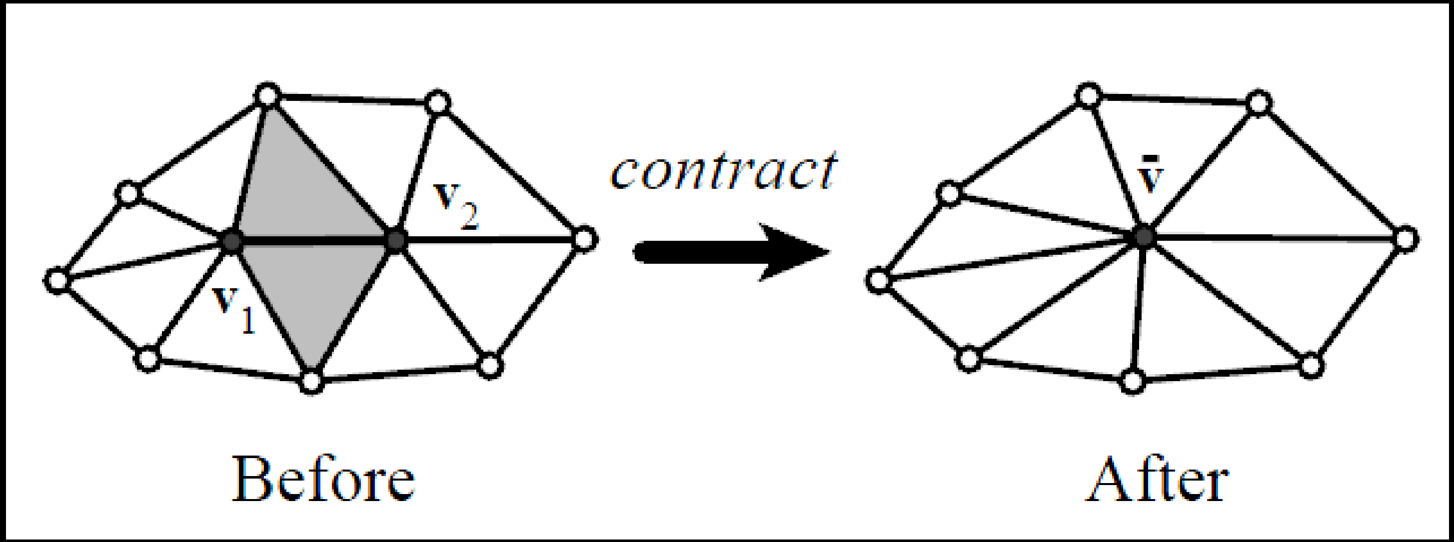
QEM error

- QEM error On edge:

$$\bar{v} = \arg\min_{v} v^T Q^e v$$

Note: Q^e may not a full rank matric

- Q on $\bar{v}$ is just Q^e.



QEM Algorithm

Input: a mesh
Output: a simplified mesh
Initialization:
Compute the Q^e matrices for all the edges.
Compute the optimal contraction target $\bar{v}$ for each edge.
While $N_v > n \; and \; Cost_{min} \; < t$

The error $\bar{v}^T Q^e \bar{v}$ becomes the *cost* of the edge.

Place all the edges in a priority queue keyed on cost with minimum cost edge at the top.

Remove the edge of the least cost from the heap , collapse this edge, and update the costs of all edges involving.

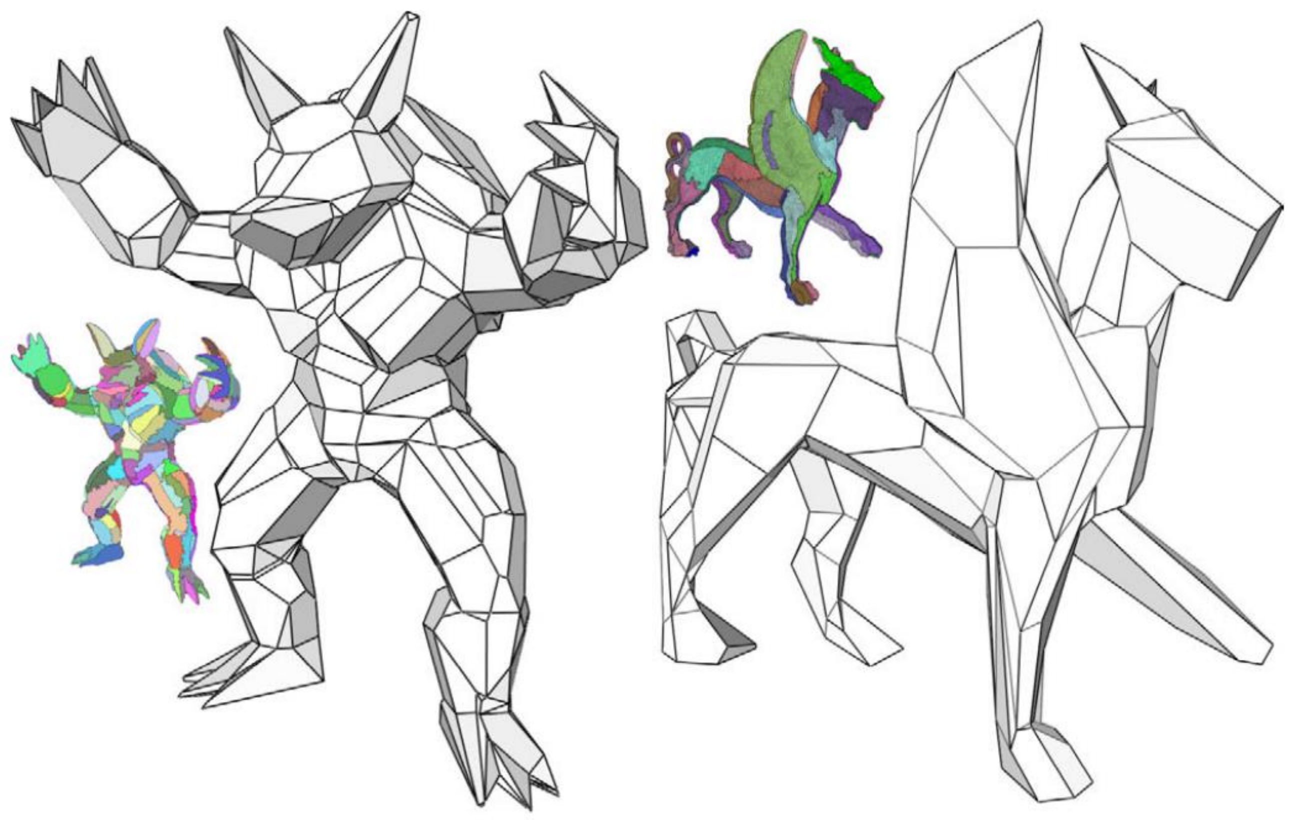
End;

Outlines

- Definition
- Local operations
- Quadric error metric
- Variational shape approximation

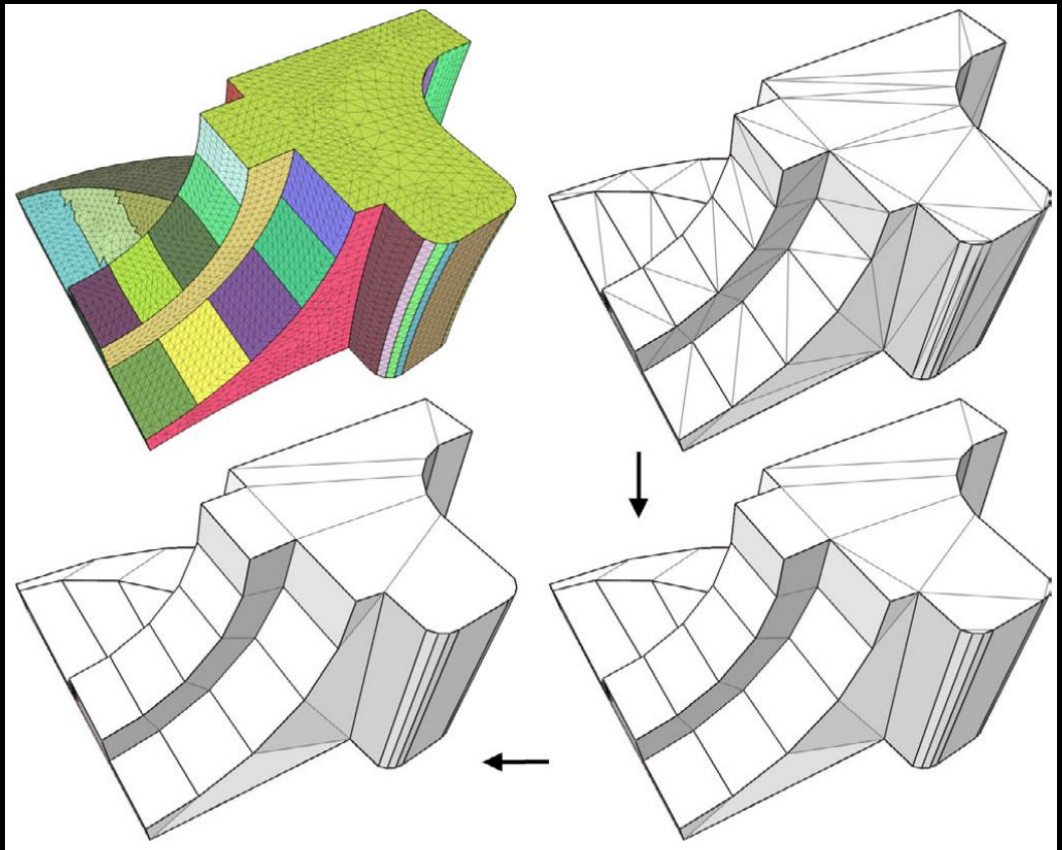
Variational shape approximation (VSA)

- 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 x_i with normal direction n_i.



Region representation

$$R_1 \cup \cdots \cup R_k = M$$

M: a triangle mesh

$R = \{R_1, \dots, R_k\}$: a partition of M into k regions.

Proxies: $P = \{P_1, \dots, P_k\}$, $P_i = (x_i, n_i)$

Distance metrics between R_i and P_i

- The squared orthogonal distance of x from the plane P_i.

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

- 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 k and an error metric E (L^2 or $L^{2,1}$), find a set $R = \{R_1, \ldots, R_k\}$ of regions and a set $P = \{P_1, \ldots, P_k\}$ 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

- 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
- Proxy fitting phase
  - the partitioning is kept fixed and the proxies are adjusted

- Initialization
  - randomly picking k triangles as R
  - The planes of k triangles are used to initialize P

Geometry partitioning phase

- Modifies the set R of regions to achieve a lower approximation error while keeping the proxies P fixed.

```
partition(R = {R_1, ..., R_k}, P = {P_1, ..., P_k})

  // find the seed triangles and initialize the priority queue
  queue = ∅
  for i = 1 to k do
    select the triangle t ∈ R_i that minimizes E(t, P_i)
    R_i = {t}
    set t to conquered
    for all neighbors r of t do
      insert (r, P_i) into queue

  // grow the regions
  while the queue is not empty do
    get (t, P_i) from the queue that minimizes E(t, P_i)
    if t is not conquered then
      set t to conquered
      R_i = R_i ∪ {t}
      for all neighbors r of t do
        if r is not conquered then
          insert (r, P_i) into queue
```

Proxy fitting phase

- The partition R is kept fixed, the proxies P_i are adjusted in order 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.