Mesh Simplification using Four-Face Clusters

Luiz Velho

Visgraf Laboratory

IMPA – Instituto de Matemática Pura e Aplicada

Rio de Janeiro, Brasil

Outline

- Simplification and Multiresolution
- Hierarchical 4-K Meshes
- Four-Face Cluster Simplification
- Examples

Mesh Simplification

- Basic Algorithm:

  while (requirements not satisfied) **do**

  (1) select candidate modifications

  (2) apply simplification operator

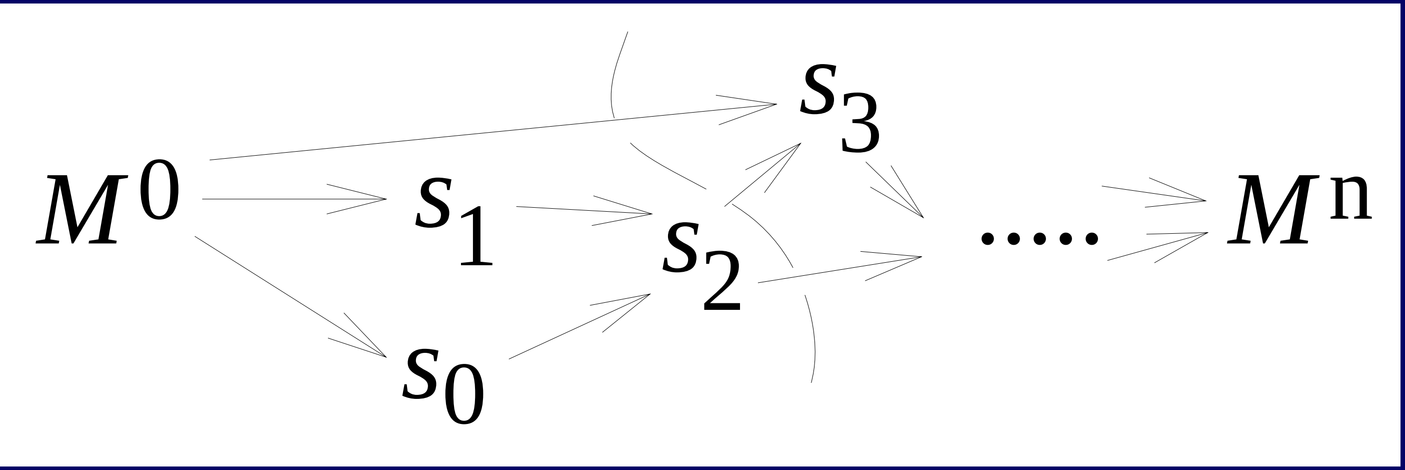
- Builds a Multiresolution

$$M^0 \xrightarrow{s_0} M^1 \xrightarrow{s_1} \cdots M^{n-1} \xrightarrow{s_{n-1}} M_n$$

* *Progressive Structure*

Variable-Resolution Meshes

- Adaptation required in many applications
  - Example: *View-dependent Geometry*
- Need more powerful hierarchical structure: (DAG)
  - Dependency among simplification operations



* *Mesh is a* cut *in the graph*

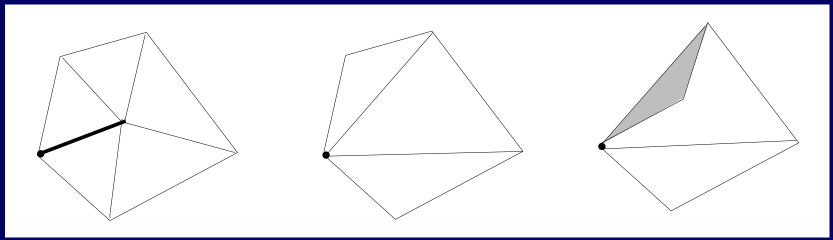
Simplification and Adapted Meshes

- Two-Step Approach:

  1 Generate Progressive Structure (Simplification)

  2 Build Vertex Tree (Dependency Analysis)

* Problem: *fold-over may occur if vertex neigborhood is not the same*



* Solution: *enforce original vertex neighgothood*

  – Easier when building Variable-Resolution structure during simplification

Four-Face Cluster Simplification

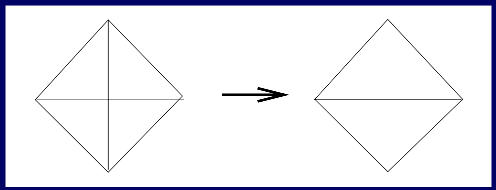
- Single-Step Approach
  - Integration of Simplification and DAG Construction
- Output is a Variable-Resolution Structure
  - 4-K Mesh
- Based on Classic Components
  - Half-Edge Collapse Variant
  - Quadric Error Metric

4-K Meshes

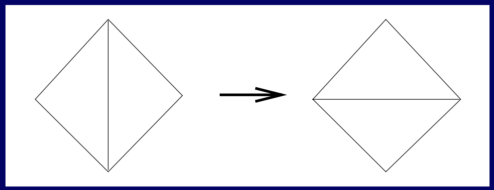
* *Particular Case of Multi-Triangulation*

- Simplest Type
- Good Properties: Expressive Power / Growth Rate / Height and Width

- Basic Operations
  - Degree-4 Vertex Removal / Insertion

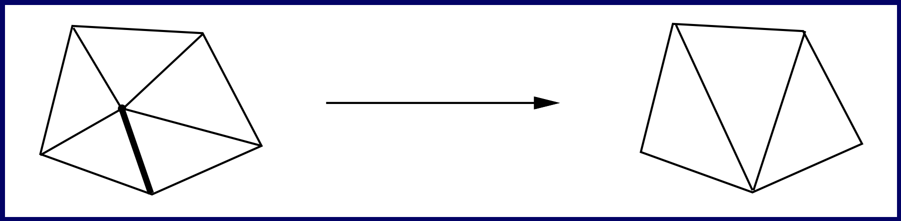


  - Edge Swap

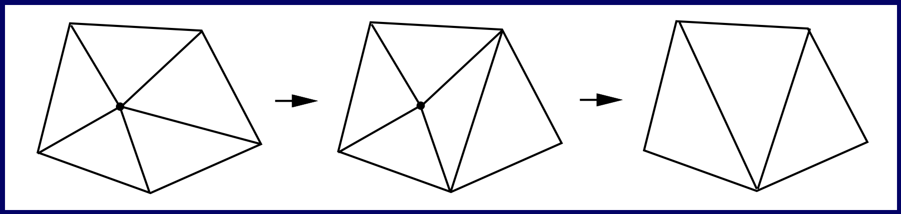


Simplification of 4-K Meshes using Edge Collapse

- Decomposition of a General Edge Collapse
  - Edge Collapse



  - Edge Swap + Deg(4) Vertex Removal



Parallel Application of Simplification Operations

- Steps of the Method

  Repeat for N refinement levels:

  1. Rank vertices based on mesh quality criteria (quadric error metric)
  2. Select an independent set of clusters that covers most of the mesh
  3. Simplify clusters using edge swaps and deg(4) vertex removals

* *Logarithmic Height*

Step 1: Measuring Error with Quadrics

- Initialization of Vertex Quadrics: $Q_v = \sum \alpha_f Q_f$
  - Area-weighted sum of incident face quadrics

- Error at Vertex v: $E(v) = \alpha C(v) + \beta S(v)$
  - Cost of Half-Edge Collapse with u fixed

$$C(v) = \min_{u \in N_1(v)} (Q_v + Q_u)(u)$$

  - Cost to make $\deg(v) = 4$

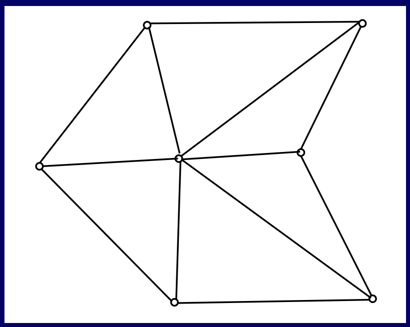
$$S(v) = \sum_{(s,t) \in I} Q_s(t)$$

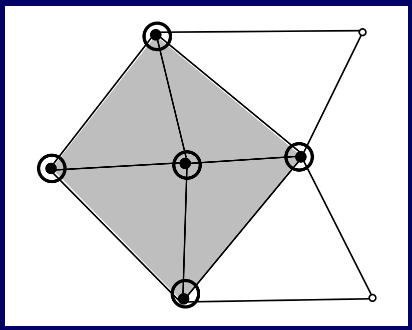
$Q_s(t)$ *is the volume squared of thetrahedron defined by* (u, v, s, t)

* (Take into account triangle aspect ratio and mesh fold-over)

Step 2: Selecting Clusters

- Independent Set Computation:
  - Cluster Marking



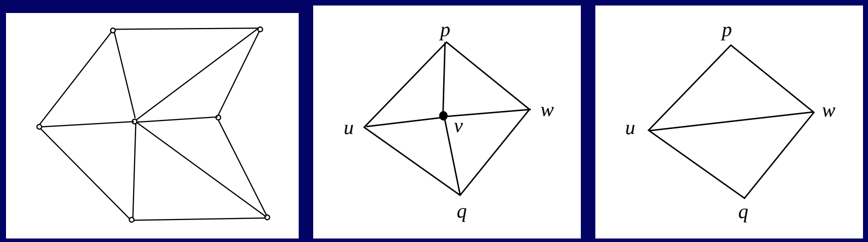


* OBS: Mark Four-Face Clusters *after* Edge-Swaps

Step 3: Simplify Clusters and Update

- Simplification of Cluster associated with Vertex v

  - Perform Sequence Edge Swaps such that $\deg(v) = 4$
  - Remove vertex v (e.g. Half-Edge Collapse (u, v))



- Update Error

  – Add Quadric of v to endpoints of new edge (u, w)

$$Q_i = (Q_v + \delta_i Q_i), \qquad i = u, w, \qquad \text{where} \quad \delta_i = 1 - \frac{C_i}{C_u + C_w}$$

  – Recompute Costs of Neighbors $p \in N_1(u) \cup N_1(w)$

Algorithm

Simplify_4k(M, n)

assign quadrics;

for all ($v \in$ M) **do**

compute $E(v)$

put v into queue

for (j = 1 to n) **do**

while (queue not empty) **do**

get v from queue

if (v not marked) **then**

perform edge swaps in $N_1(v)$

remove vertex (v)

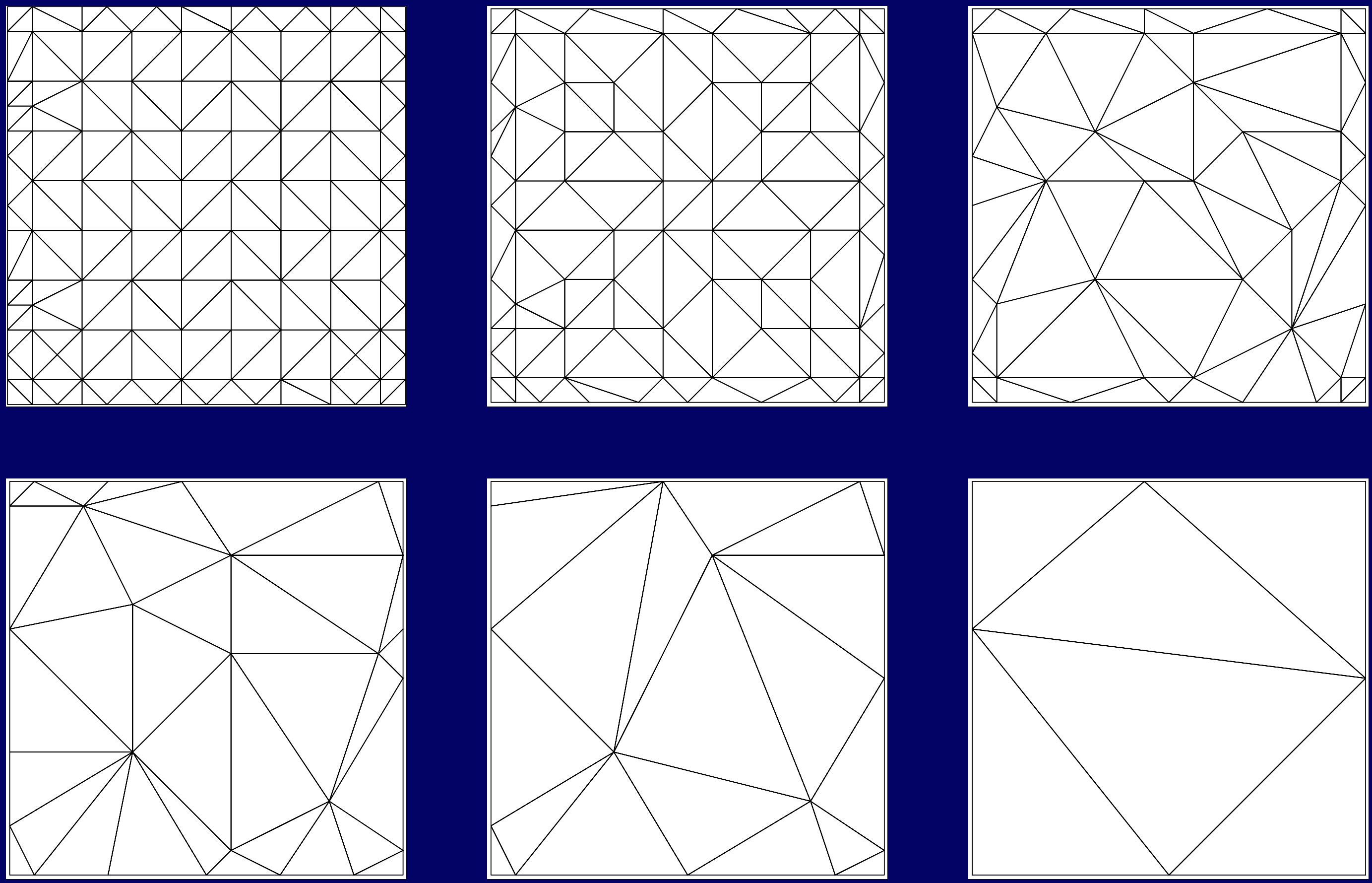
recompute quadrics Q_u and Q_w

update queue for $p \in N_1(u) \cup N_1(w)$

Examples

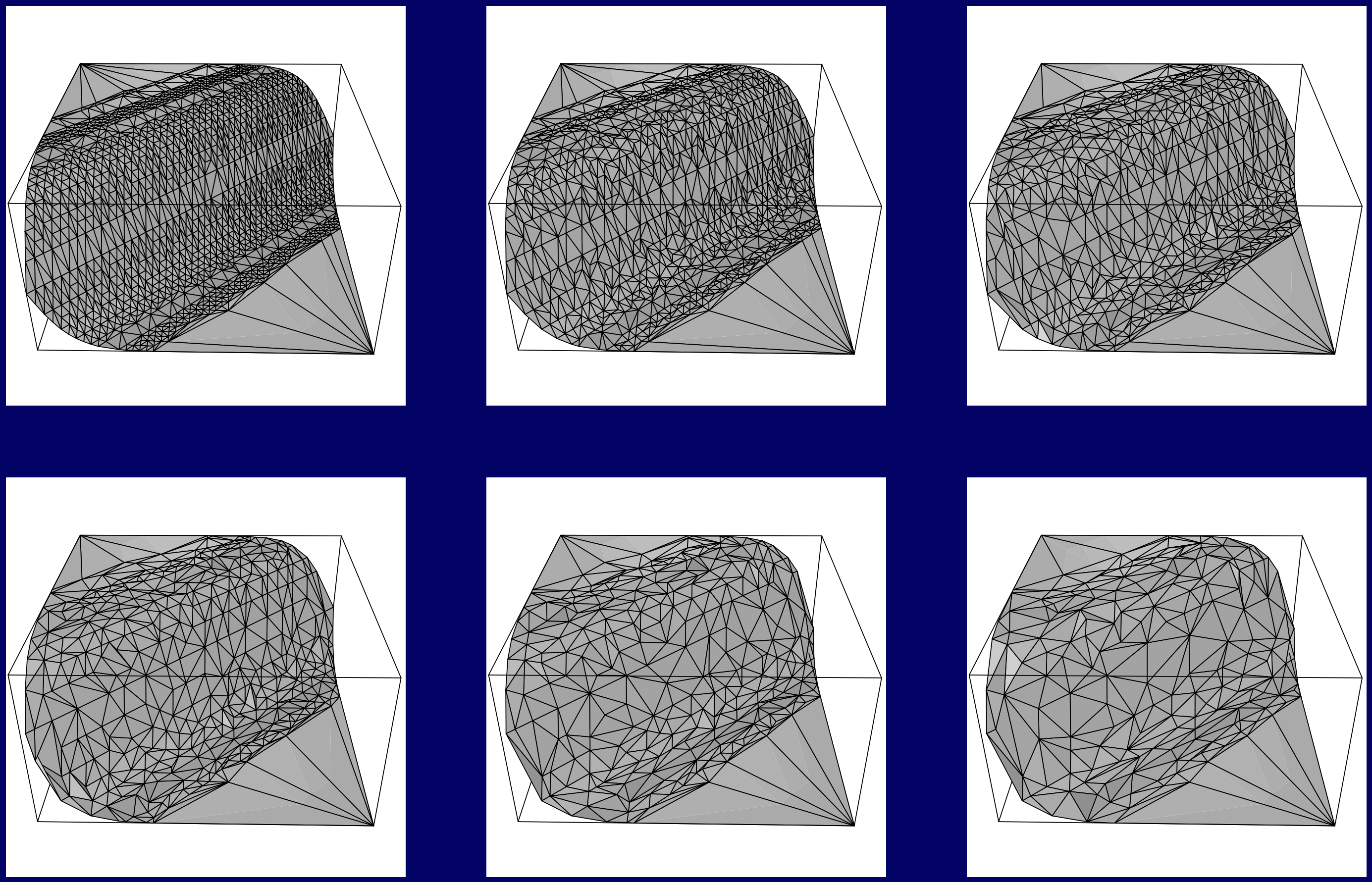
- Simplification Sequence
  - Planar Triangulation
  - Height Surface
  - Cow
  - Bunny
- Adapted Meshes
  - Bunny

Planar triangulation



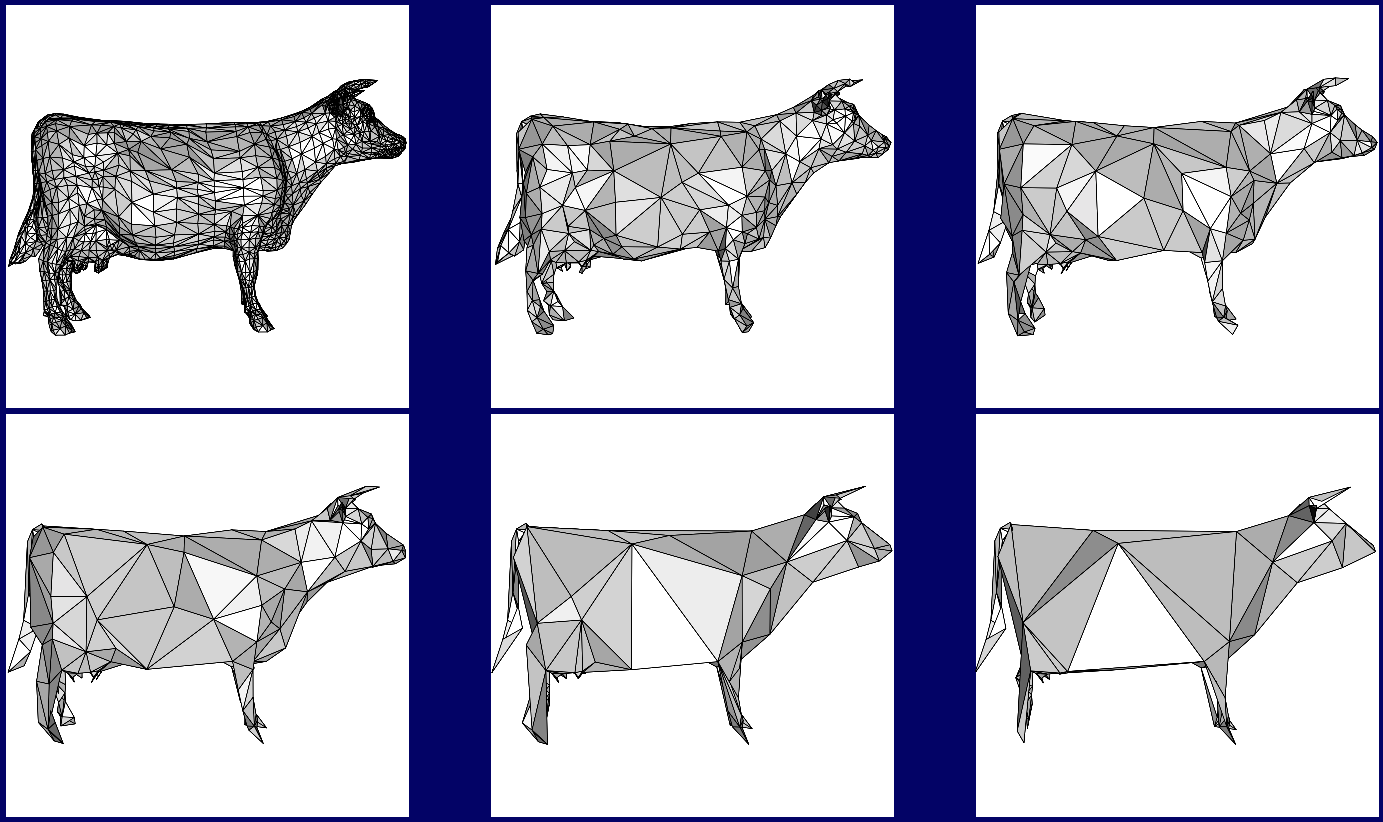
- Simplified meshes with 186, 132, 69, 35, 19, 6 triangles

Heigh surface



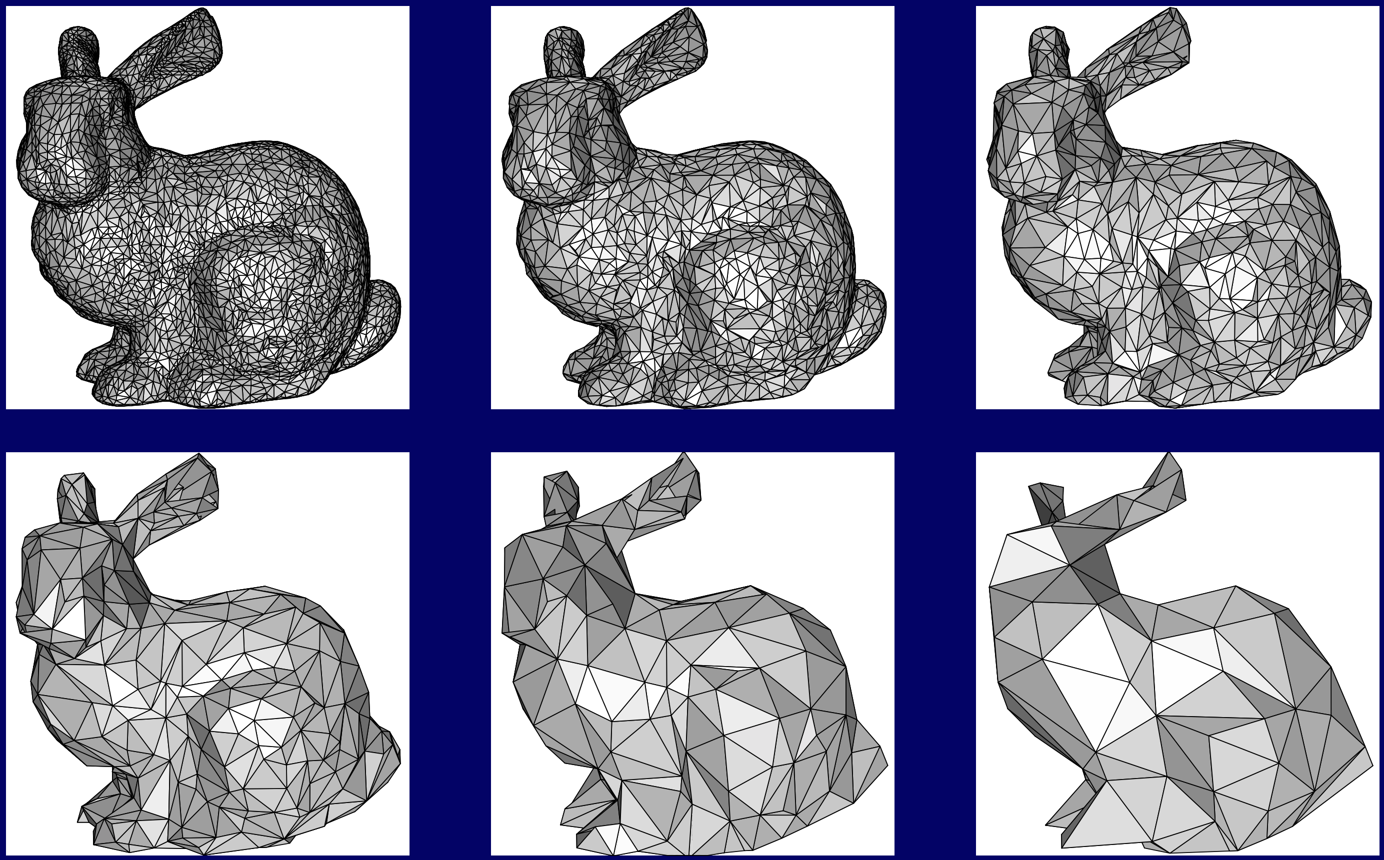
- Simplified meshes with 2432, 1594, 1103, 749, 500, 338 triangles

Cow model



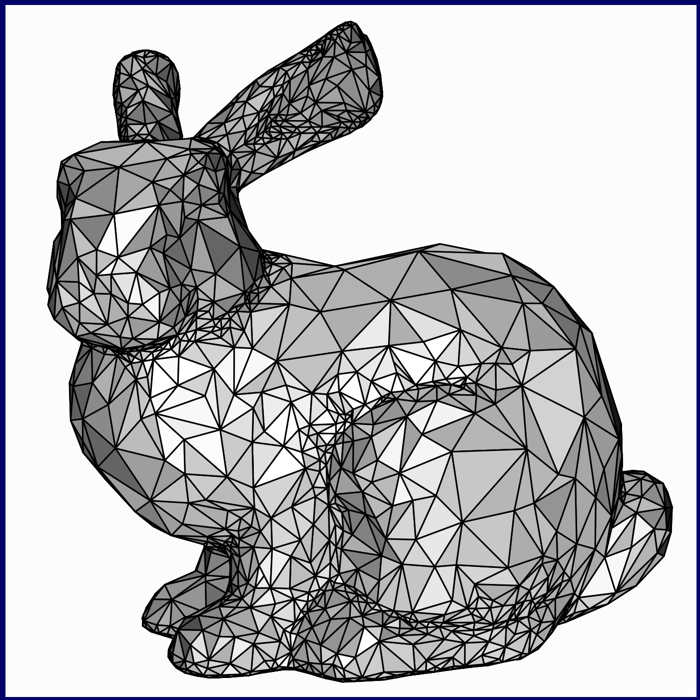
Simplified meshes with 5800, 1200, 700, 400, 300, 200 triangles

Stanford Bunny

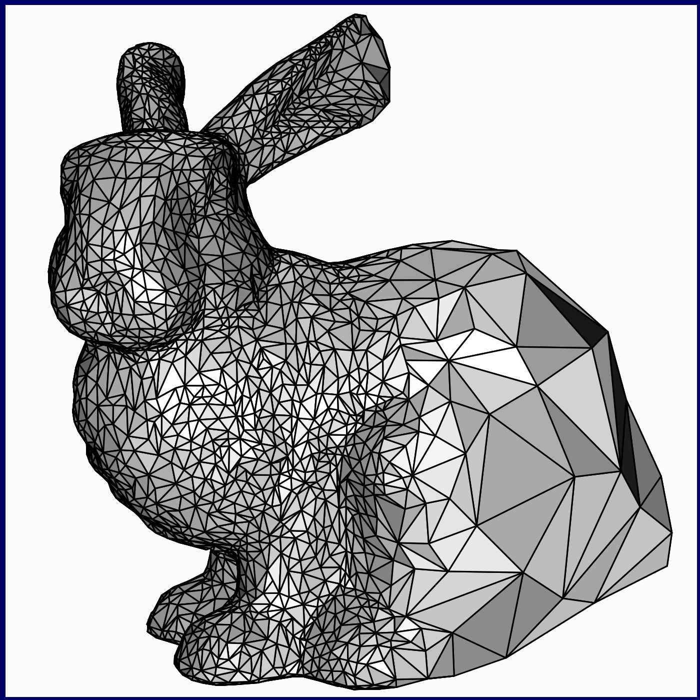


Simplified meshes with 10000, 4577, 2106, 988, 463, 245 triangles

Adapted meshes



Curvature



Region Selection

Final Remarks

- Conclusions
  - Single-Step Simplification and Hierarchy Construction
  - Simple Algorithm Based on Classic Components
  * Same Principles can be used for Sequencial Simplification
- Future Work
  - Out-of Core Extension
  - Feature Detection: Tagged Meshes