UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE

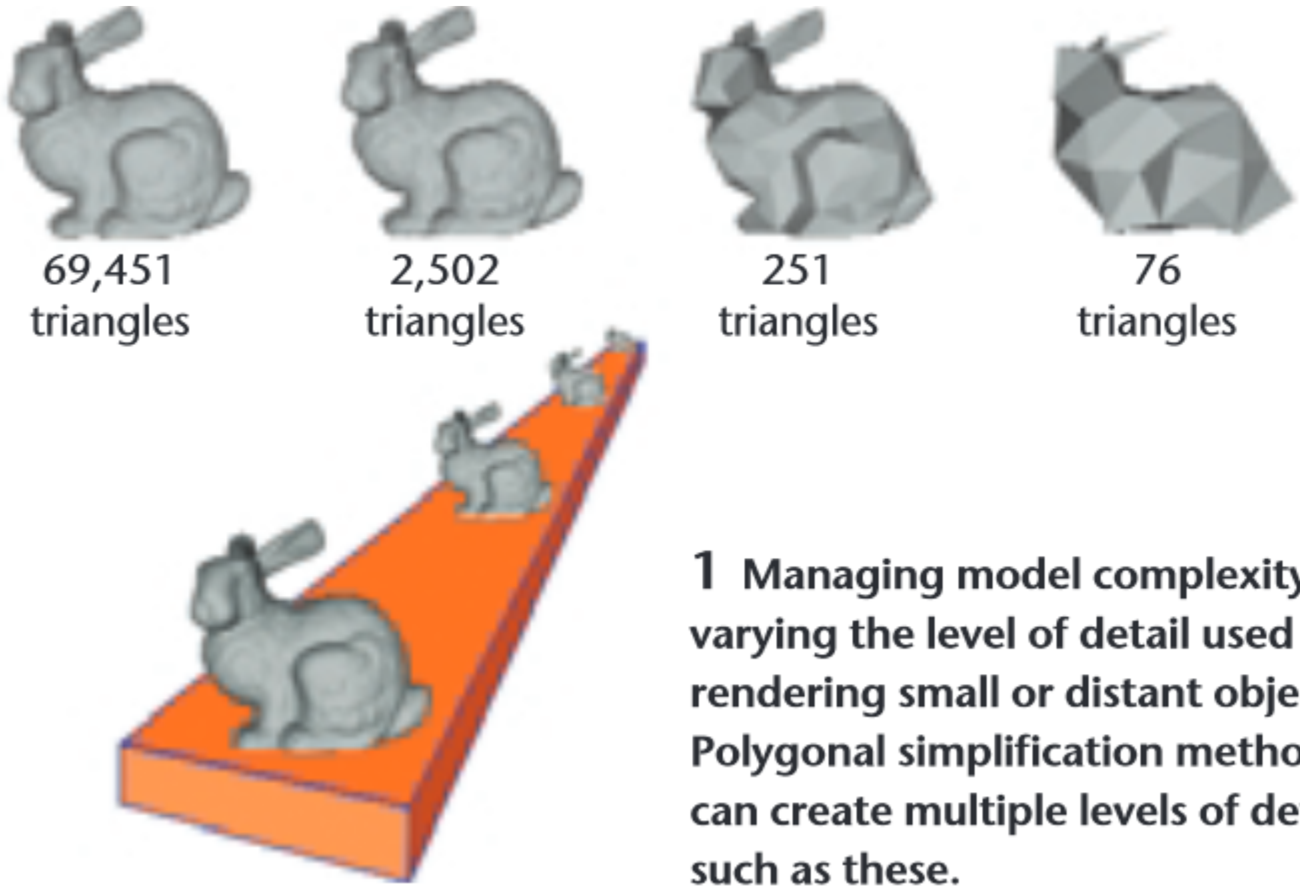
Mesh Simplification

Presented by :
Keerti Sekhar Sahoo
Xiao Mao

Goal

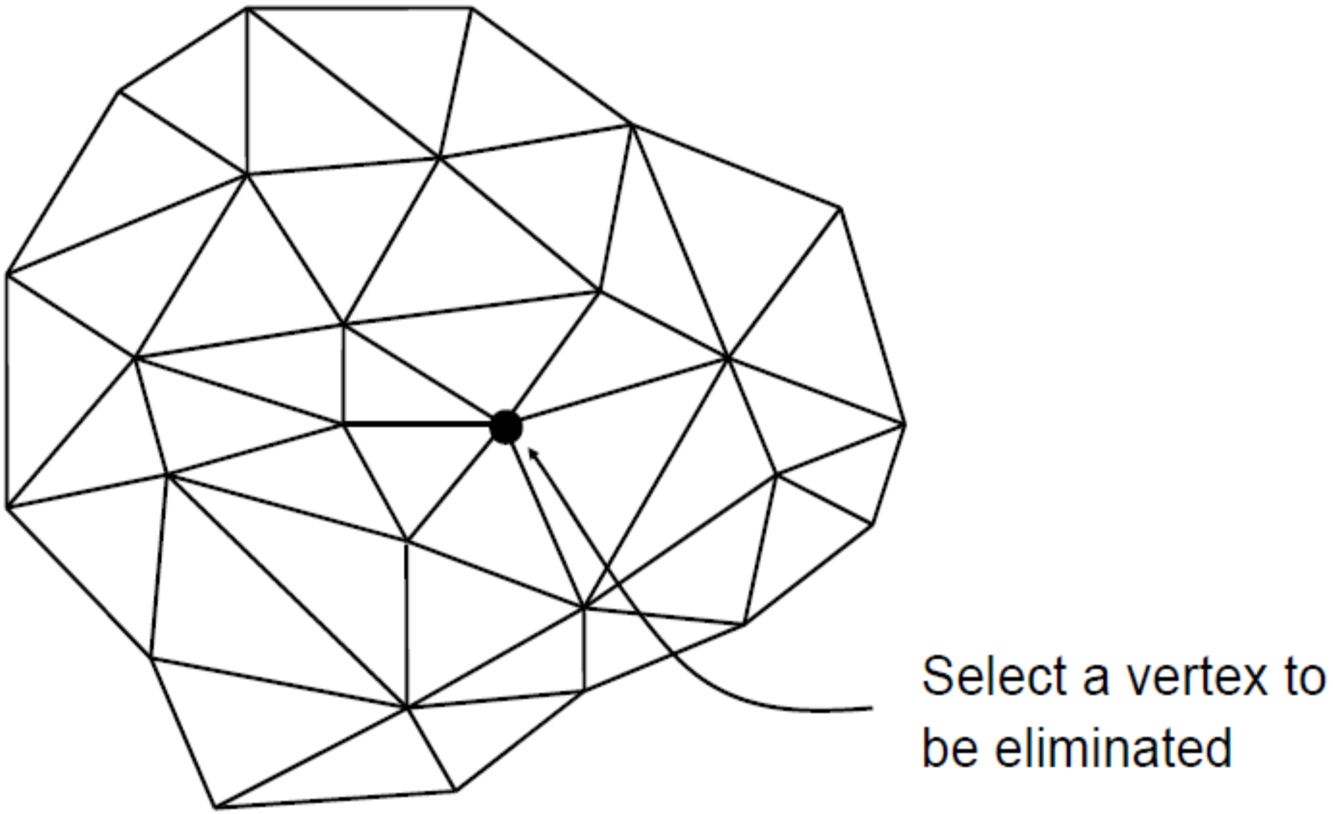
Reduce the amount of vertices and faces of a mesh and try to maintain the original topology.

The level of details of output should be acceptable compared to the original geometry.

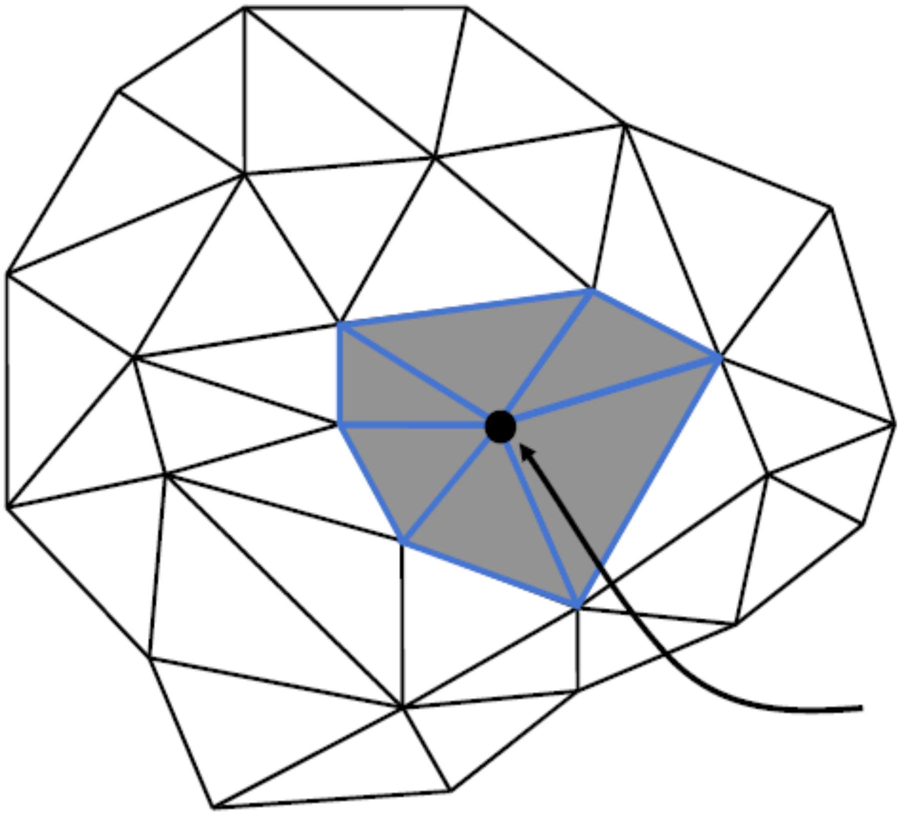


1 Managing model complexity by varying the level of detail used for rendering small or distant objects. Polygonal simplification methods can create multiple levels of detail such as these.

Remove an unimportant vertex

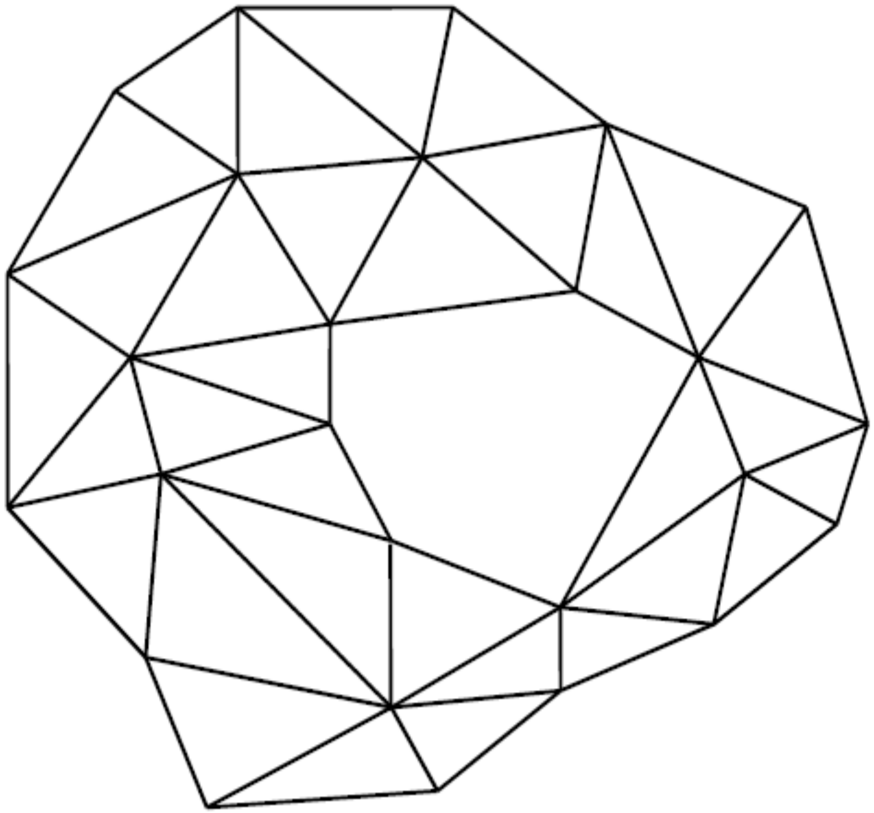


Remove an unimportant vertex



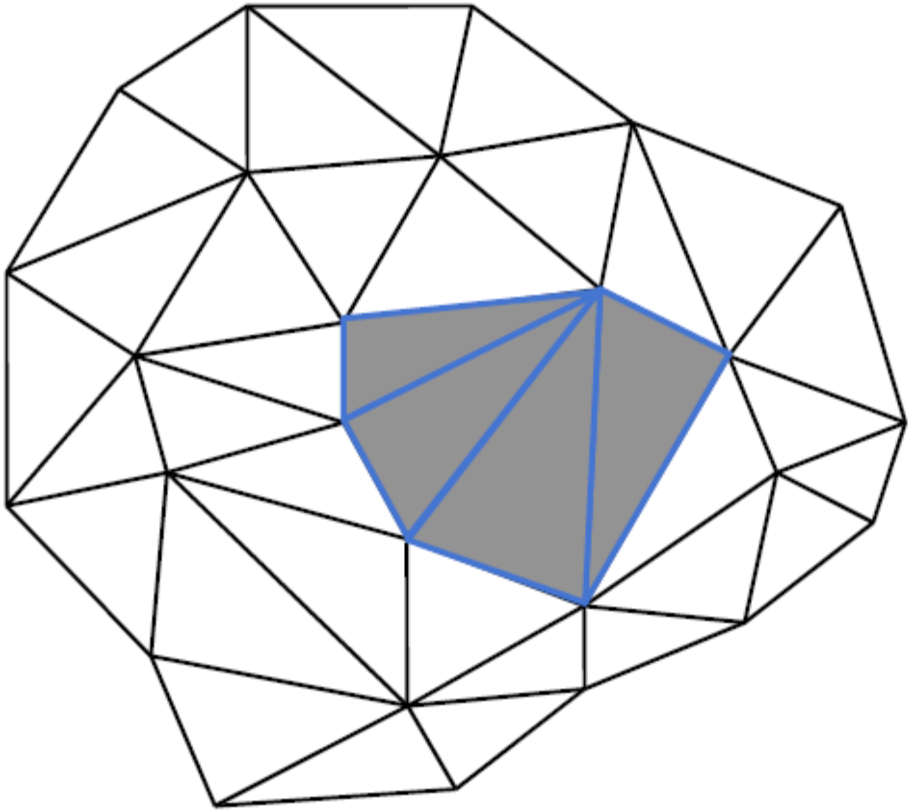
Select all triangles sharing this vertex

Remove an unimportant vertex



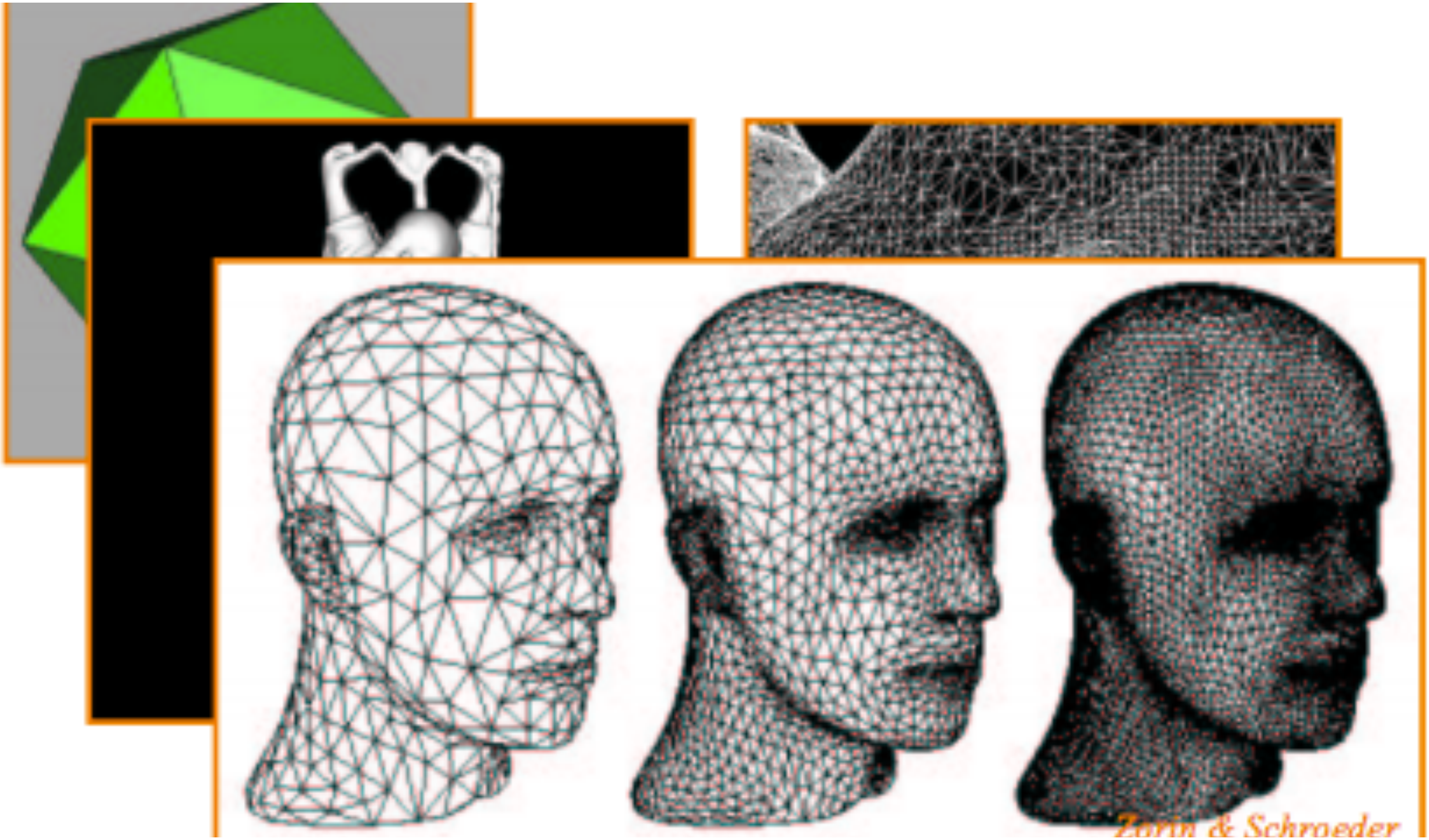
Remove the selected triangles, creating the hole

Remove an unimportant vertex



Fill the hole with
new triangles

Polygonal Meshes



Popularity of Polygonal Meshes

Polygons are popular in interactive computer graphics because of their mathematical simplicity.

There are various methods to simplify the polygonal geometry of small, distant and unimportant portions of the model and reduce the rendering process at the same time maintaining the quality of the scene’s outcome.

Need of Mesh Simplification

- **Rendering 3D graphical models with fine geometric details.**
- **Storage optimization.**
- **Bandwidth optimization.**

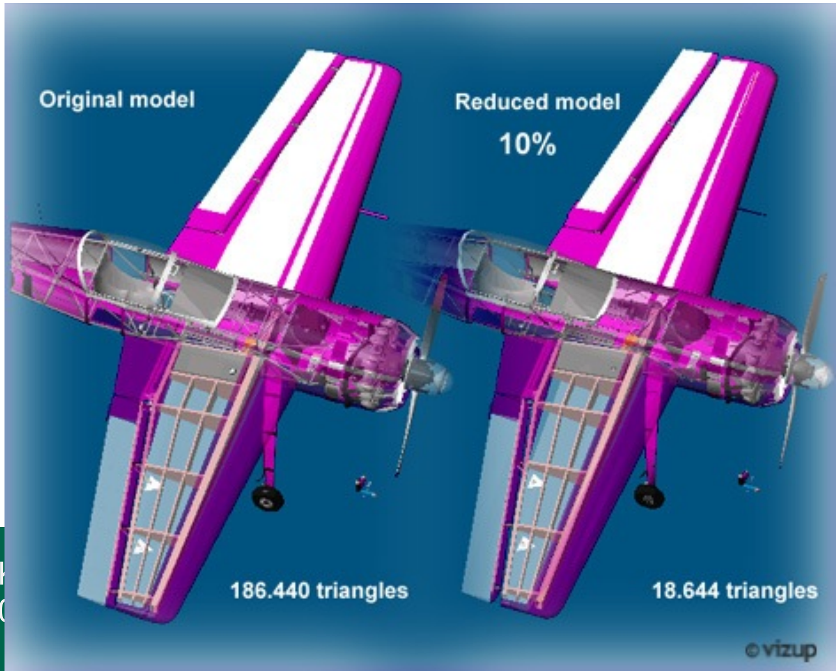
Eliminate redundant geometry

- Surface with lots of co planar triangles are merged into few larger polygons.

- Effective way to reduce the level of complexity of the mesh and doesn't introduce any geometric error.
  e.g. Primitives those are not visible like the inner side of the object

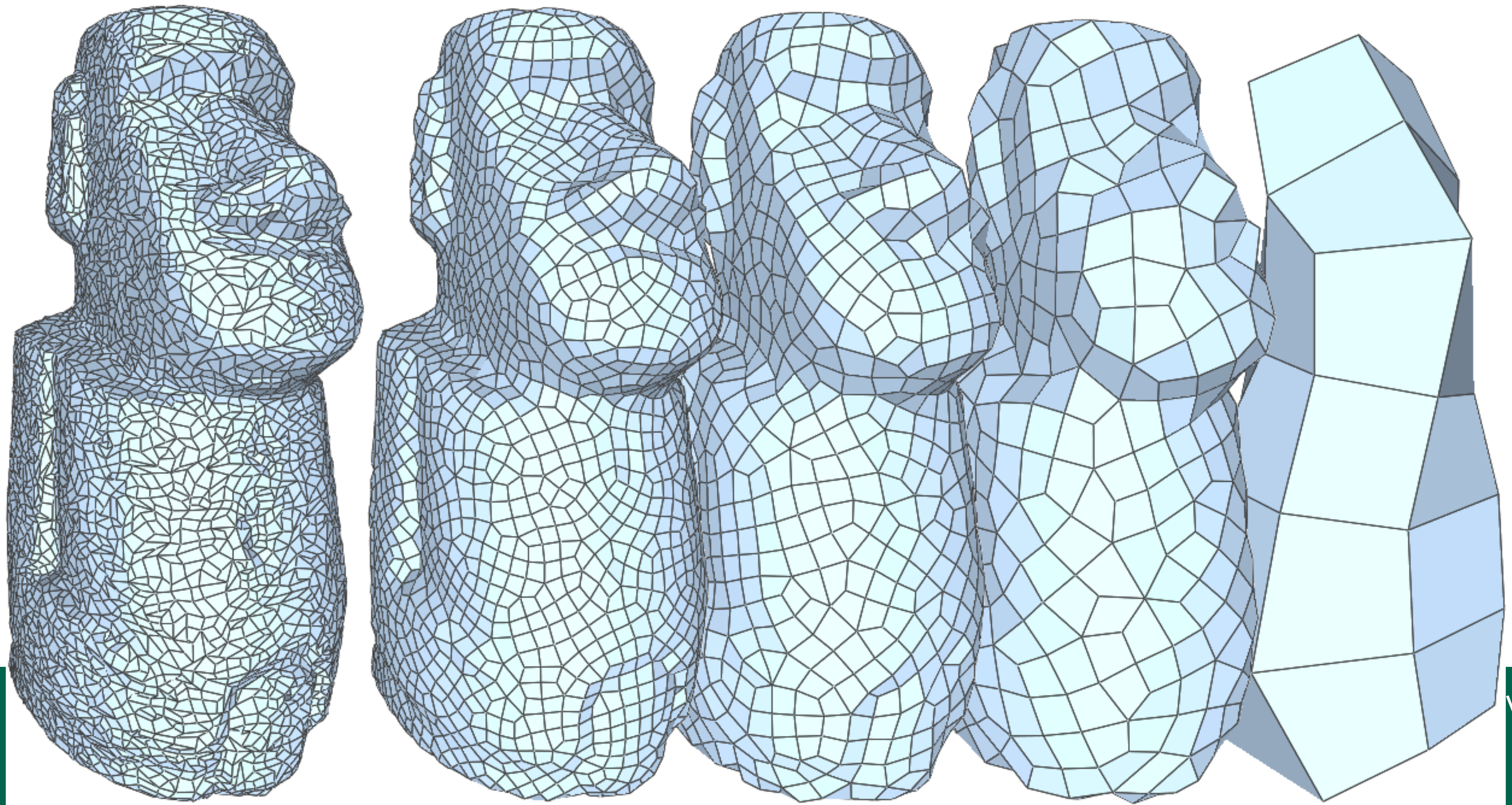
Reduce Model Size

- Downloading models from website with considering the optimization of bandwidth and storage space.
- Implementation needs geometric compression.



Improving Running Time Performance

Generate levels of details(LOD) of the object.
Higher LOD leads to a more realistic object.



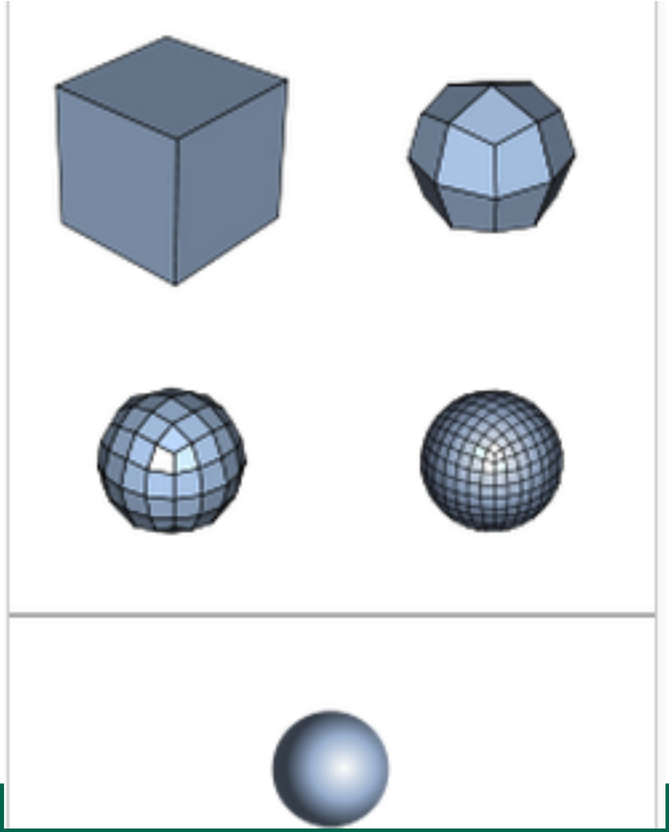
Mechanisms

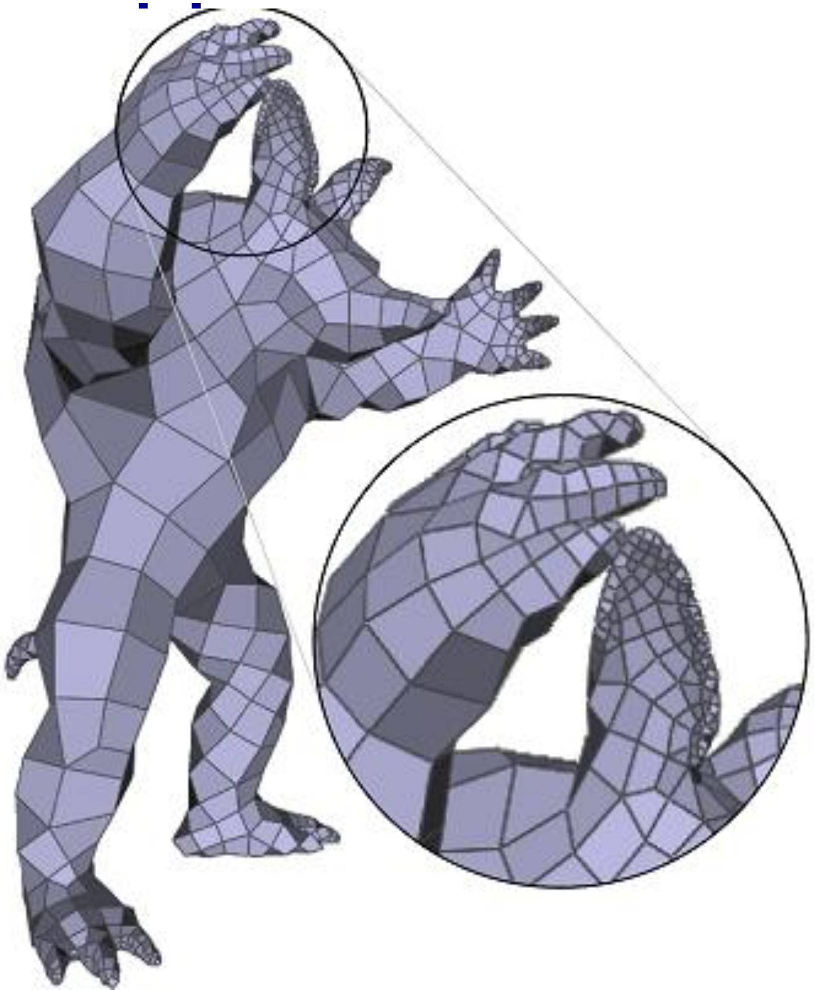
Sampling Algorithm

- **Samples the initial model's geometry. e.g the points on the model's surface or voxels stacked on the model in a 3D grid.**
- **More elaborate the difficult to code approach.**

Adaptive Subdivision

➢ **A simple base mesh is recursively subdivided to closely approximate the initial**





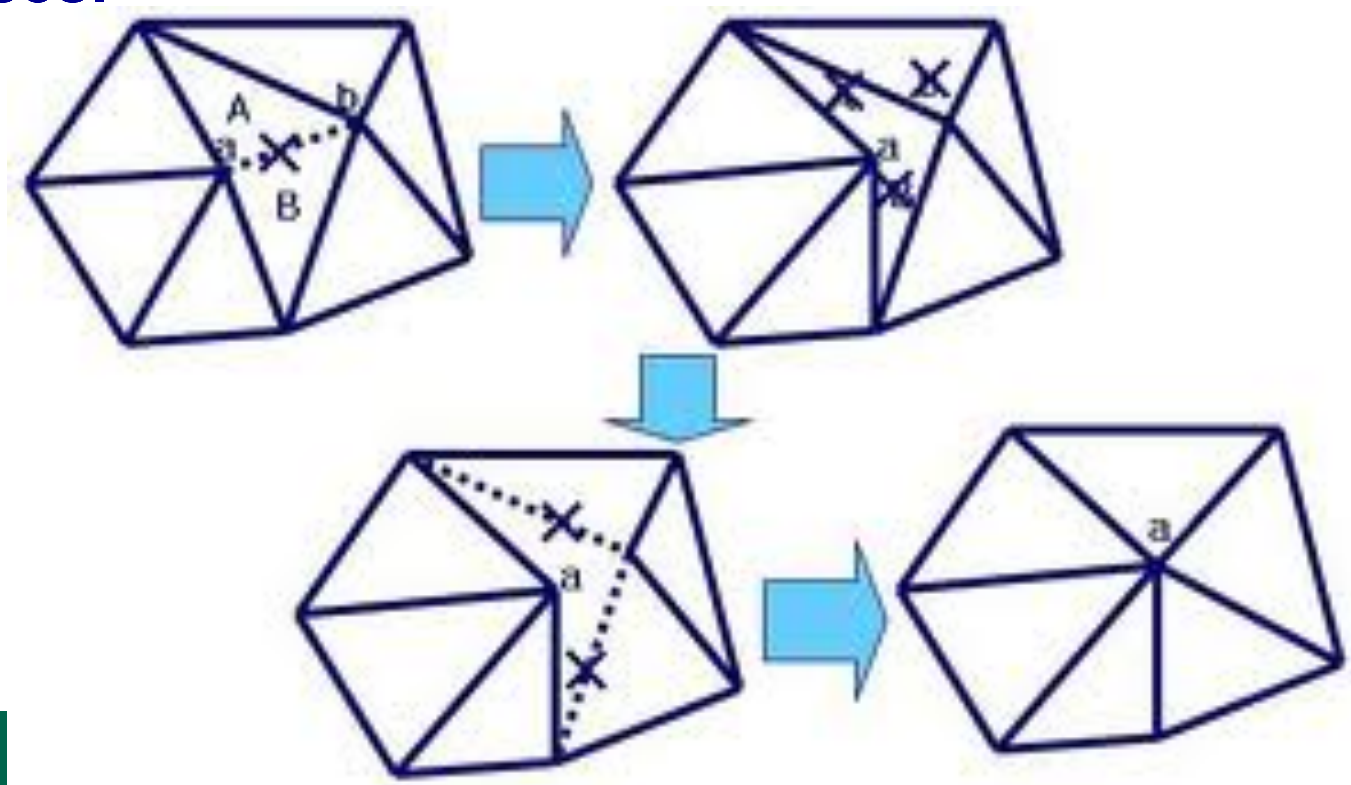
OF
CO

Decimation

- **Iteratively removes vertices or faces from the mesh and re-triangulating the resulting hole after each step .**
- **These algorithms are fast.**

Vertex merging

Operates by collapsing two or more vertices into a single vertex which in turn can be merged with other vertices.



Decimation of Triangle Meshes

- **Reduce the total number of triangles in a triangle mesh**
- **Preserve the original topology and a good approximation of the original geometry**

Overview

A multiple-pass algorithm

- During each pass, perform the following three basic steps on every vertex:
- Classify the local geometry and topology for this given vertex
- Use the decimation criterion to decide if the vertex can be deleted.
- If the point is deleted, re-triangulate the resulting hole.
- This vertex removal process repeats, with possible adjustment of the decimation criteria, until some termination condition is met.

Decimation process

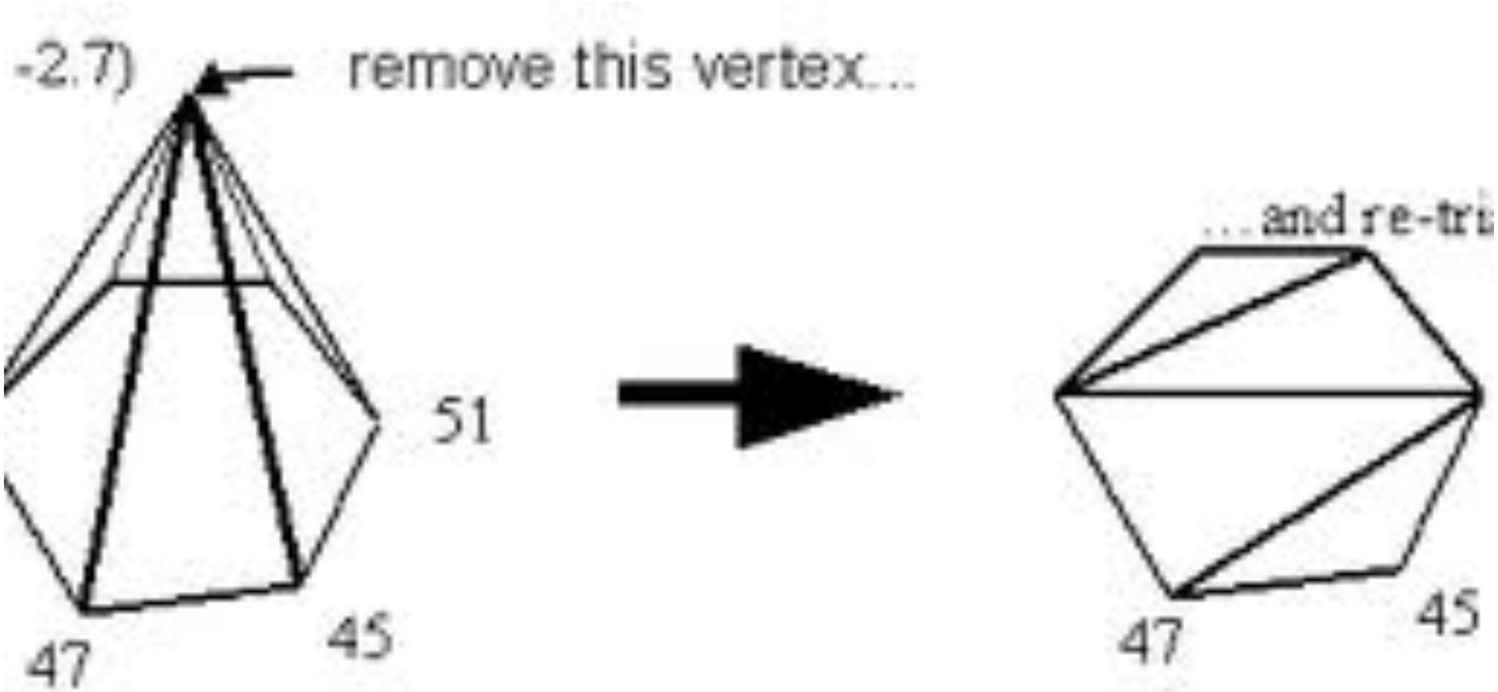


Fig. 4. Vertex decimation

Steps

Basically for each vertex, three steps are involved:

- Characterize the local vertex geometry and topology
- Evaluate the decimation criteria
- Triangulate the resulting hole.

Feature Edge

- **A feature edge exists if the angle between the surface normals of two adjacent triangles is greater than a user-specified "feature angle".**

Characterize Local Geometry and Topology

Each vertex is assigned one of five possible classifications:

Simple vertex

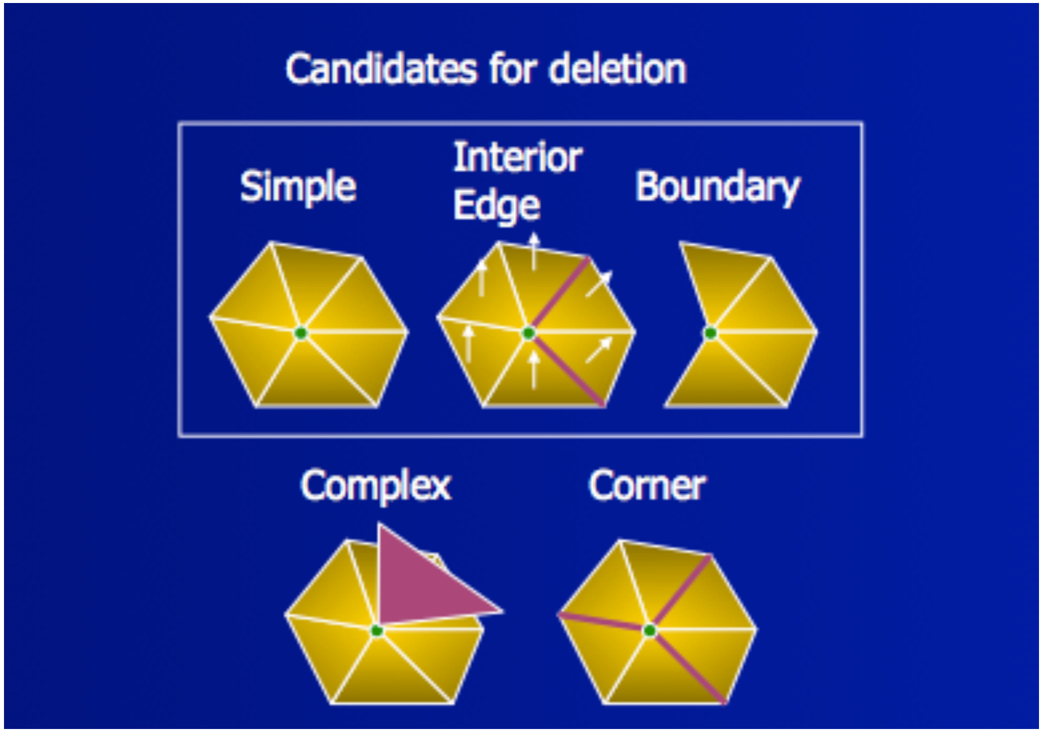
Complex vertex

Boundary vertex

Interior edge vertex

Corner vertex

Classification of edges



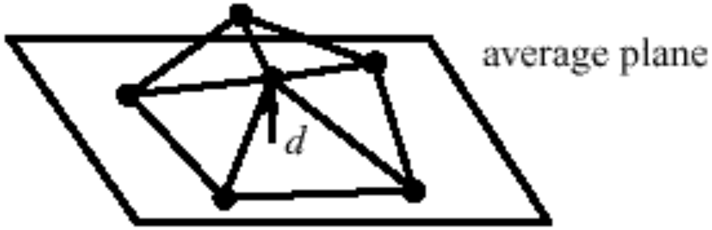
Evaluate the Decimation Criteria

- **Complex vertices are not deleted from the mesh.**
- **Use the distance to plane criterion for simple vertices.**
- **Use the distance to edge criterion for boundary and interior edge vertices.**

Criterion for Simple Vertices

Use the distance to plane criterion.

If the vertex is within the specified distance to the average plane, it can be deleted. Otherwise, it is retained.



Criterion for Boundary &Interior Edge Vertices

Use the distance to edge criterion.

If the distance to the line defined by two vertices creating the boundary or feature edges is less than a specified value, the vertex can be deleted.

Criterion for Corner Vertices

Corner vertices are usually not deleted to keep the sharp features.

But it is not always desirable to retain feature edges.

Meshes containing areas of relatively small triangles with large feature angles

Small triangles which are the result of ‘noise’ in the original mesh.

Use the distance to plane criterion.

Triangulate the Hole

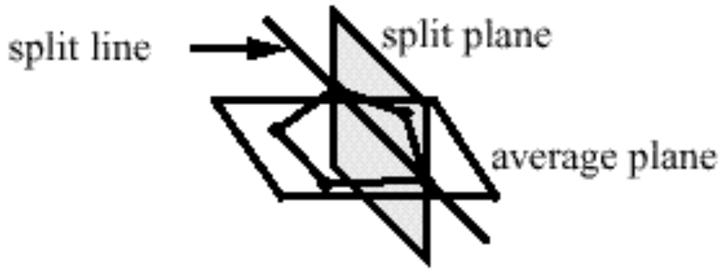
Deleting a vertex and its associated triangles creates one(simple or boundary vertex) or two loops(interior edge vertex).
The resulting hole should be triangulated.
From the Euler relation, it follows that removal of a simple, corner, interior edge vertex reduces the mesh by exactly two triangles. For boundary vertex, exactly one triangles

Recursive Splitting Method

The author used a recursive loop splitting method.

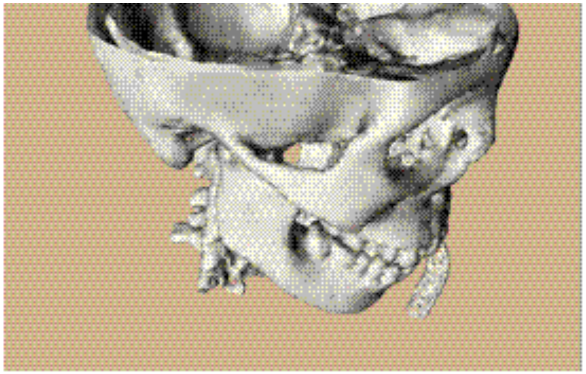
Divided the loop into two halves along a line defined from two non-neighboring vertices in the loop.

Each new loop is divided until only three vertices remain in each loop.

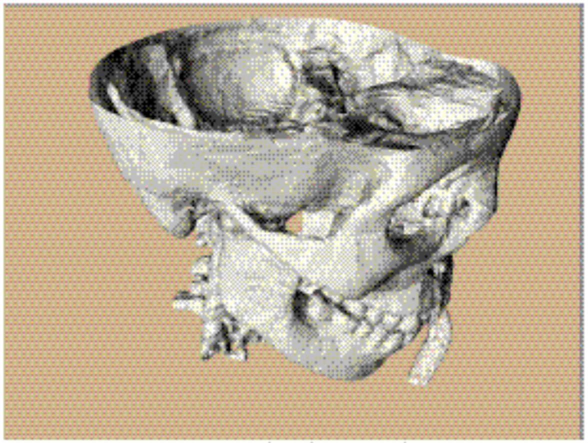


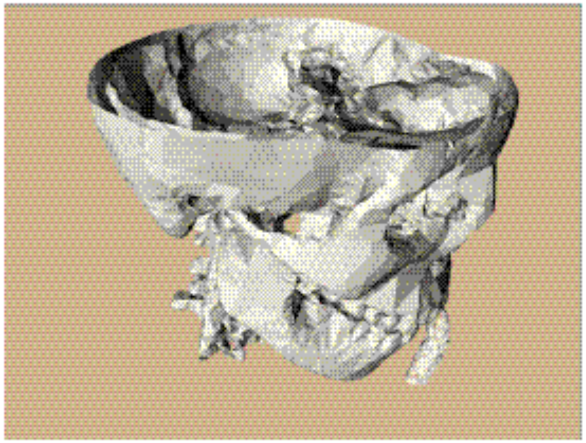
Results

Full Resolution
(569K Gouraud shaded triangles)



75% decimated
(142K Gouraud shaded triangles)





Vertex Clustering

Grading

Clustering

Synthesis

Adjustment of normals

Grading

A weight value is computed for each vertex according to its visual importance (e.g. size of the faces bounded by the vertex).

Clustering

Create clustering-cell

Size of clustering-cell depends on level of detail

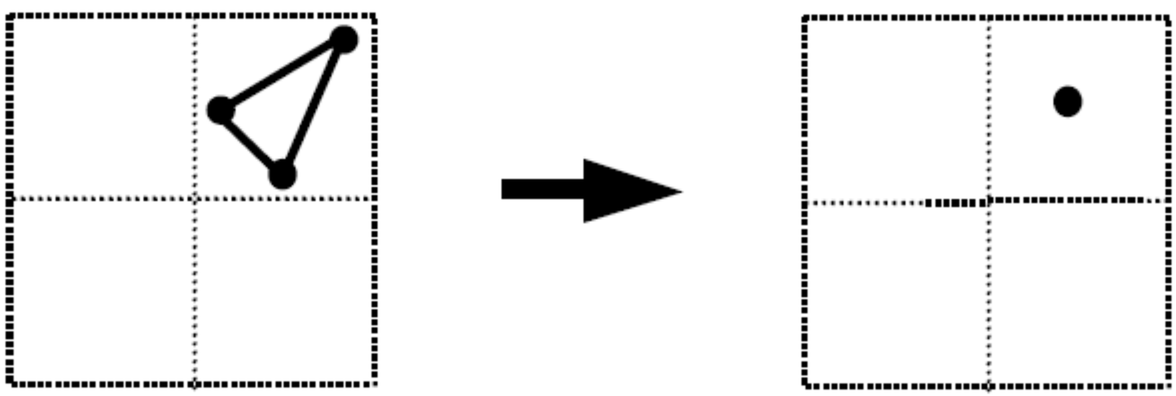
Keep a vertex representative per cell

Clusters $p \leftrightarrow \{p_0,\ldots,p_n\}$, $q \leftrightarrow \{q_0,\ldots,q_m\}$
Connect (p,q) if there was an edge (p_i,q_j)

Synthesis

The representative of P_1, P_2, ..., P_k in the same cell is their weighted average:

$$\mathbf{P} = \frac{w_1\mathbf{P}_1 + w_2\mathbf{P}_2 + \ldots + w_k\mathbf{P}_k}{w_1 + w_2 + \ldots + w_k}$$



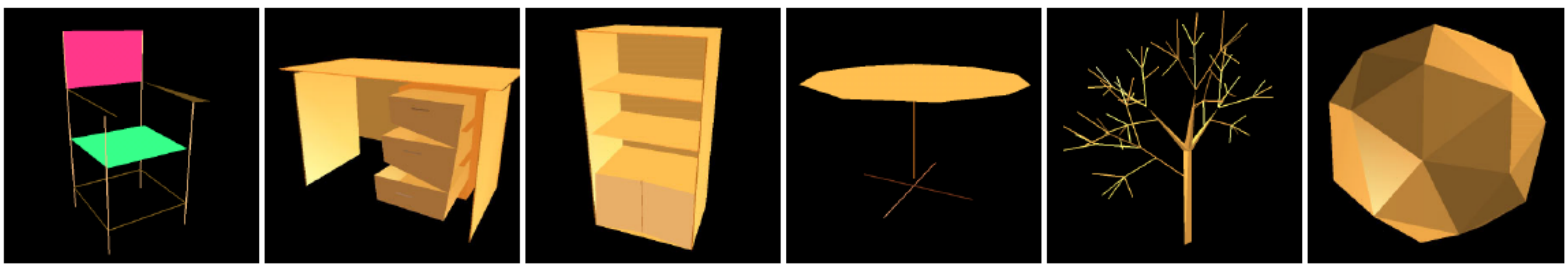
Adjustment of normals

Normals of resulting edges and triangles should be reconstructed

Result

(a) (b) (c) (d) (e) (f)

Original mesh



(a) (b) (c) (d) (e) (f)

Simplified mesh

Result

plate	vertices	triangle	edges	points
original models				
1(a)	782	1528	0	0
1(b)	1254	2512	0	0
1(c)	406	808	0	0
1(d)	1947	3840	0	0
1(e)	2720	5100	0	0
1(f)	1742	3480	0	0
simplified using floating-cell clustering				
3(a)	20	10	9	0
3(b)	60	109	10	0
3(c)	23	40	6	1
3(d)	16	20	3	0
3(e)	95	18	78	0
3(f)	32	60	0	0

References

http://webdocs.cs.ualberta.ca/~anup/Courses/604_3DTV/Presentation_files/Polygon_Simplification/luebke01developers.pdf

https://www.comp.nus.edu.sg/~lowkl/publications/lowkl_i3d1997.pdf

http://www.cs.mtu.edu/~shene/COURSES/cs3621/SLIDES/Simplification.pdf

http://graphics.stanford.edu/courses/cs468-10-fall/LectureSlides/08_Simplification.pdf

References

http://www.cs.princeton.edu/courses/archive/fall08/cos526/lec/526-08-simplify.pdf

Mesh Simplification Viewer

http://www.jsomers.com/vipm_demo/meshsimp.html

Thank you!