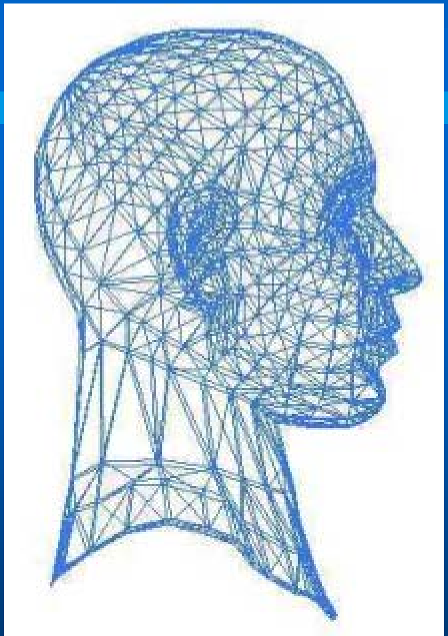
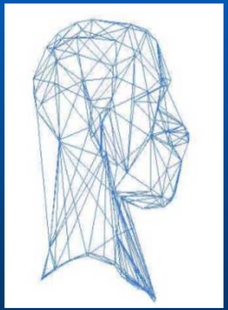


Simplification and multiresolution representation for surface meshes



T. Firsova

Overview

Introduction

Preliminaries

Error metrics

Simplification of Surfaces

Multiresolution Models

Introduction

Motivation

- Triangle meshes are the most popular (drawing) primitives in Computer Graphics
    - Hardware support for rendering
    - Triangles are the basic elements for piecewise linear interpolation
    - Triangles are basic elements for mesh generation
- Triangle meshes are large

Data sources

Laser digitizing

Digital terrain modeling

Tessellation of CAD models

Isosurface generation

Application areas

- Computer Graphics - real time rendering, animation, transmission
- Scientific visualization - clean up of marching cubes isosurfaces
- Computer Vision - acquired range data (noisy), model fitting
- Computer Aided Design - tessellations of curves surface models

Large meshes

Large meshes mean:

- Large memory requirements
- Slower rendering
- Slower transmission
- Slower computation - more expensive analysis

Problems and solution

Problems:

- Storage
- Rendering
- Transmission over the network
- Analysis (e.g. Finite Element computations)

Solution:

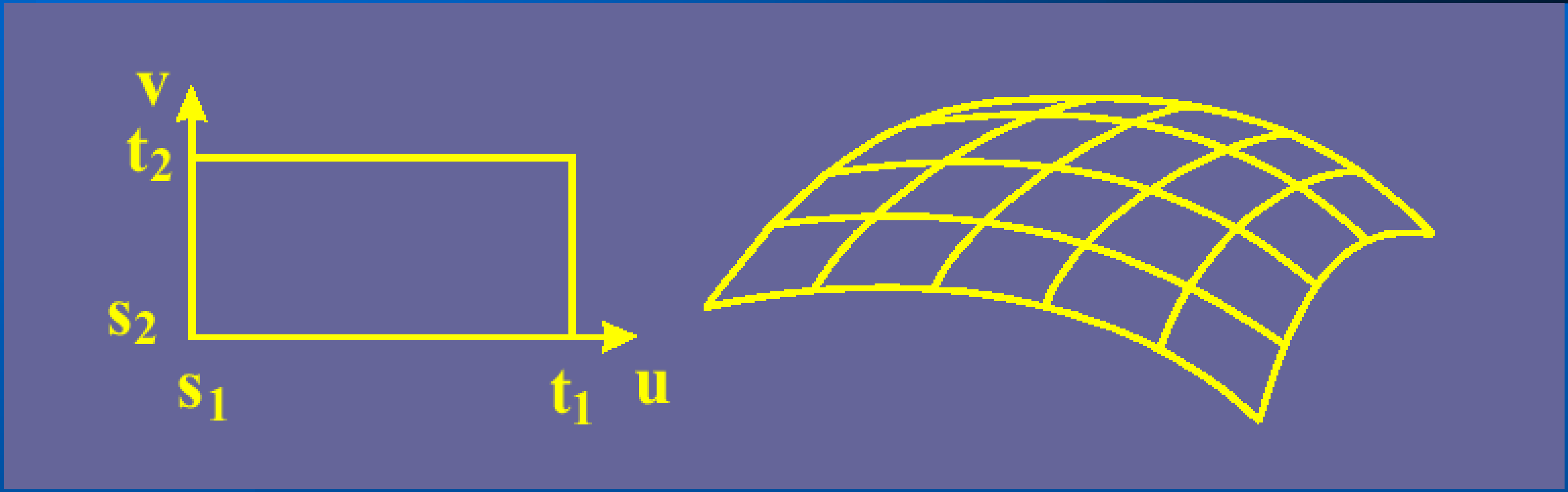
- Mesh simplification
- Multiresolution representations of meshes

Preliminaries

Preliminaries

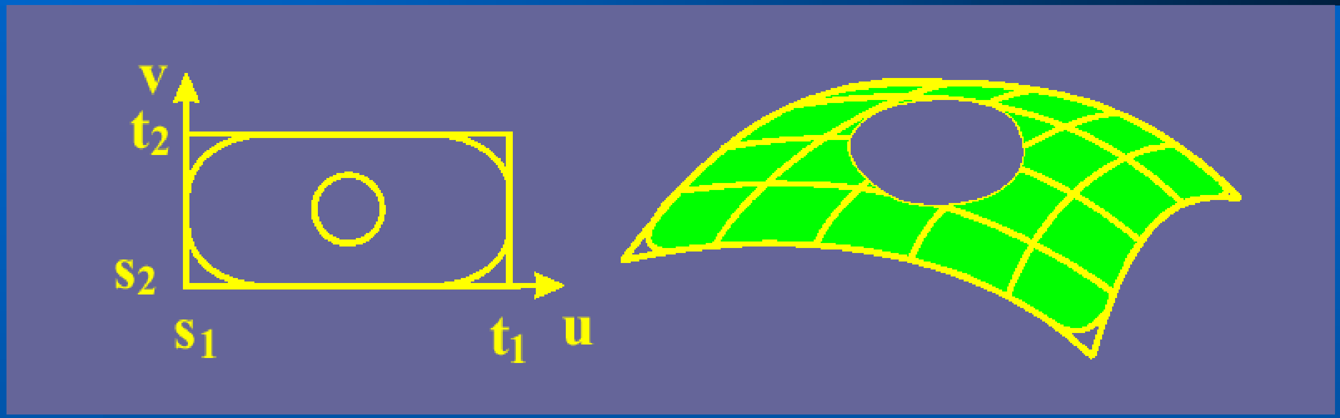
- Parametric surfaces
- Scalar fields (e.g. terrain)
- Simplicial meshes
- Manifold triangulations
- Data structures for polygonal and triangle meshes

Parametric surfaces



A *parametric patch* $\varphi(\Omega)$ is the image of continuous function $\varphi : \Omega \rightarrow R^3$,
where Ω is a compact domain in R^2 .
R^3 – *physical space;*
R^2 – *parametric space;*

Parametric surfaces

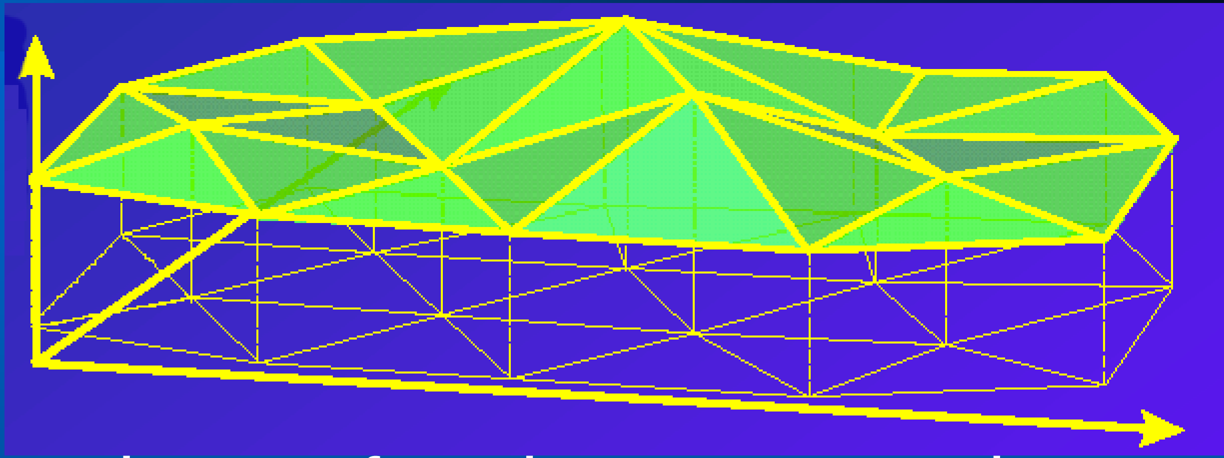


The boundary of domain Ω, $\partial\Omega$, is formed by a finite set of closed curves, called ***trimming curves***.

A ***parametric surface*** is a collection of parametric patches P = $\{\varphi_1(\Omega_1)\ ,\ \varphi_2(\ \Omega_2)\ ,\ldots\ \varphi_k(\ \Omega_k)\ \}$,such that for each pair of patches $\varphi_i(\Omega_i)\ ,\ \varphi_j(\Omega_j)\ ,\ i \neq j,$:

$$\varphi_i(\Omega_i) \cap \varphi_j(\Omega_j) = \partial\varphi_i(\Omega_i) \cap \partial\varphi_j(\Omega_j)$$

Scalar fields



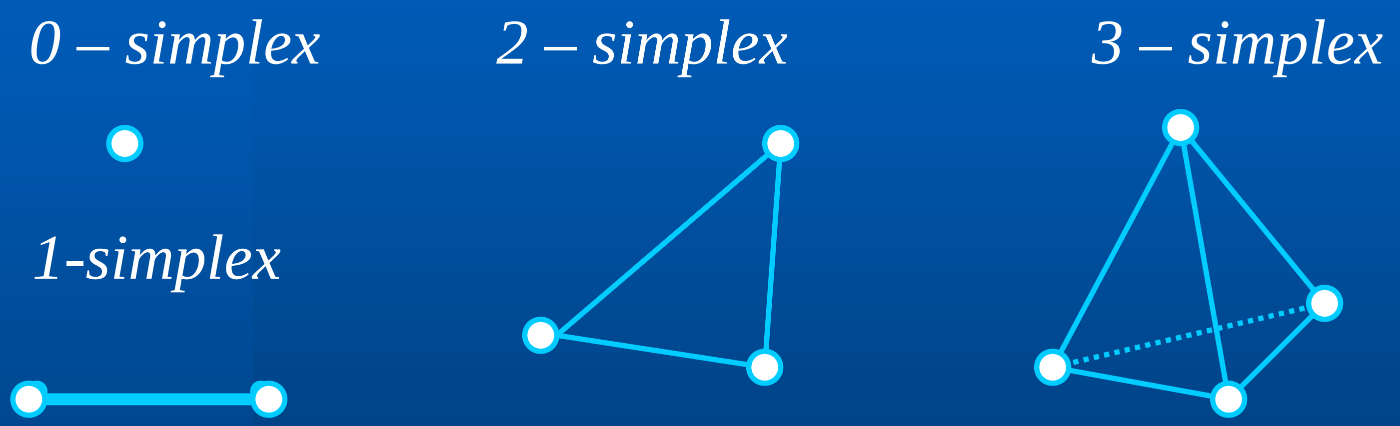
A ***scalar field*** is a continuous function $\varphi : \Omega \to R$, where Ω is a compact domain in R^k , $k \geq 1$

The image of φ embedded in R^{k+1} space, i.e.,
$F = \{(X, \varphi(X)) \,/\, X \in \Omega\} \subset R^{k+1}$ is called a ***hypersurface***.

For k=2 F is called an *explicit surface*, (also terms: non-parametric surface, height field, $2^1/_2$ D-surface, topographic surface)

Simplices

A ***d-simplex*** *S* is the convex combination of *d+1* linearly independent points. *d* is called *dimension* of the simplex.

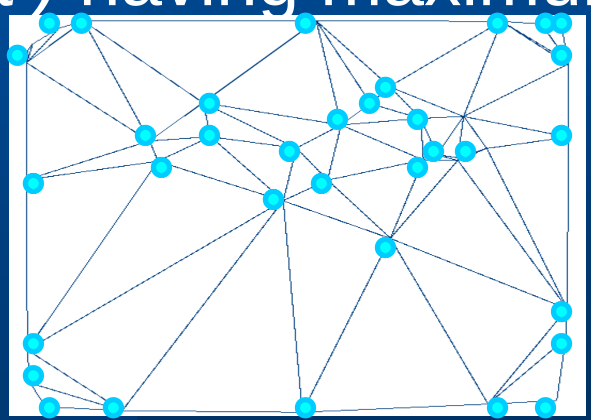


The ***boundary*** *dS* of a simplex consists of all *(d-k)*-simplices contained in *S (k>0)*.
The simplexes in the boundary of *S* are called ***faces***.

Simplicial Mesh

A finite set T of simplices in R^n is a ***simplicial mesh*** when the following conditions hold:

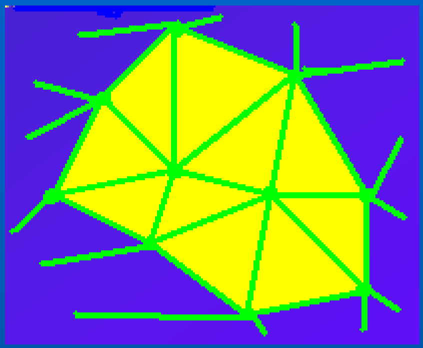
- For each simplex $t \in T$ all faces of t belong to T;
- For each pair of simplexes $t_0, t_1 \in T$, either $t_0 \cap t_1 = \emptyset$ or $t_0 \cap t_1$ is a simplex of T;
- Each simplex t is a face of some simplex t' (possibly $t \equiv t'$) having maximum order among all simplices of T.



Triangulation - 2-simplicial mesh embedded in either R^2, or R^3

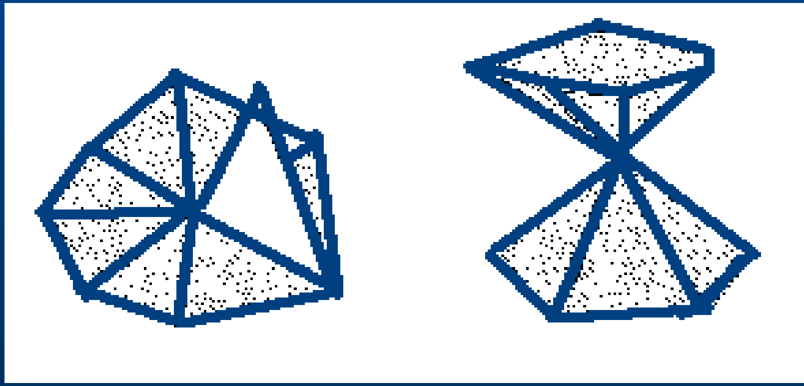
Manifold surfaces

Manifold surface (2-manifold) S: subset of R^k, for some $k \geq 3$, such that each point of *S* has an open neighborhood homeomorphic to the open disc in R^2



A 2-manifold with boundary is homeomorphic to a simplicial complex C of dimension 2 satisfying the following conditions:

- Every 1-simplex in C is manifold: incident to one or two 2-simplexes.
- For every 0-simplex *v* in C:
  - *v.star* - does not contain non-manifold 1-simplexes
  - Set of *v.star* 0-simplexes is connected



Definition: Let S be a simplex in a simplicial complex C. **Star(S)** is the set of simplexes of which S is face.

Manifold Triangulations

Euler relations:

Let v, e, f be the number of vertices, edges and faces of 2-manifold mesh

For mesh isomorphic to a sphere (genus one):
$v - e + f = 2 \quad \rightarrow \quad e \approx 3v \quad \text{and} \quad f \approx 2v$

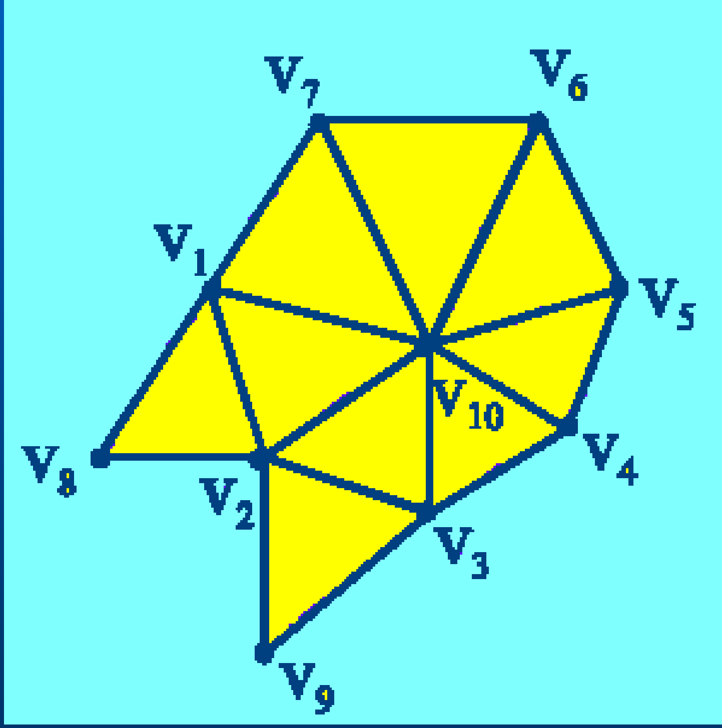
For mesh with boundary:
$e = 3(v-1) - b$
$t = 2(n-1) - b$
b – number vertices on border

Data structures

Geometry (location of the vertices in 3D-space)
Connectivity (triangles, adjacency relations between triangles)

Vertex list: $v_i = (x_i, y_i, z_i)$

$v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}$



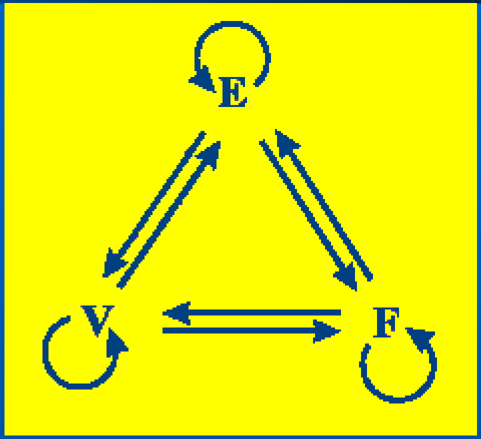
Triangle list:
(1, 2, 10), (2,3,10), (3,4,10)(4,5,10), (5,6,10), (6,7,10),(7,1,10), (1,8,2), (2,9,3)

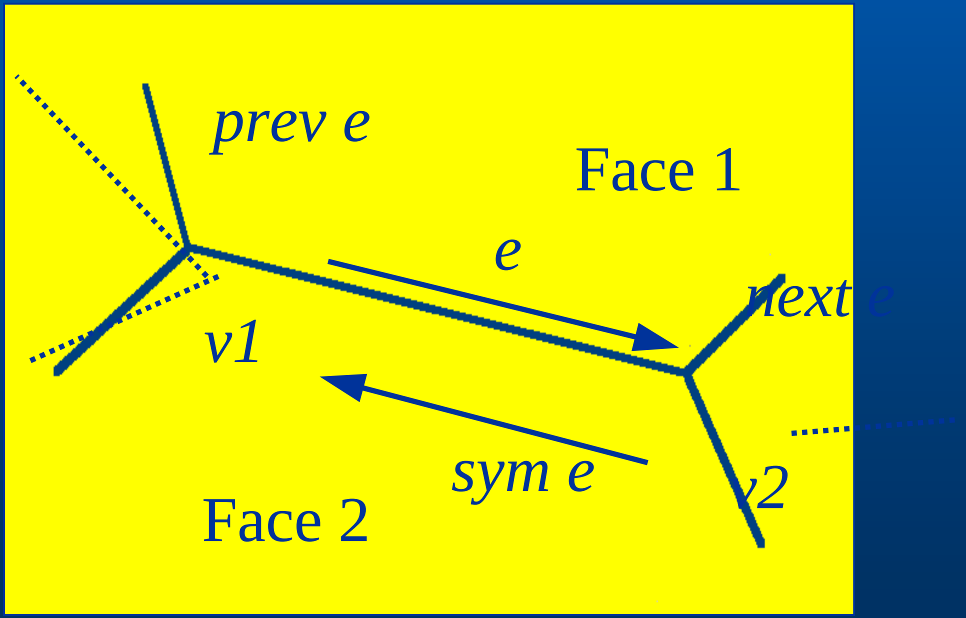
+ each vertex is stored only once
- no direct adjacency relations are stored

2-manifold data structures

Data structure for triangulation coding usually contains:

- list of main objects (one or few from the set V, E, F)
- some set of mutual adjacency relations.





winged-edge data structure

DCEL

2-manifold data structures

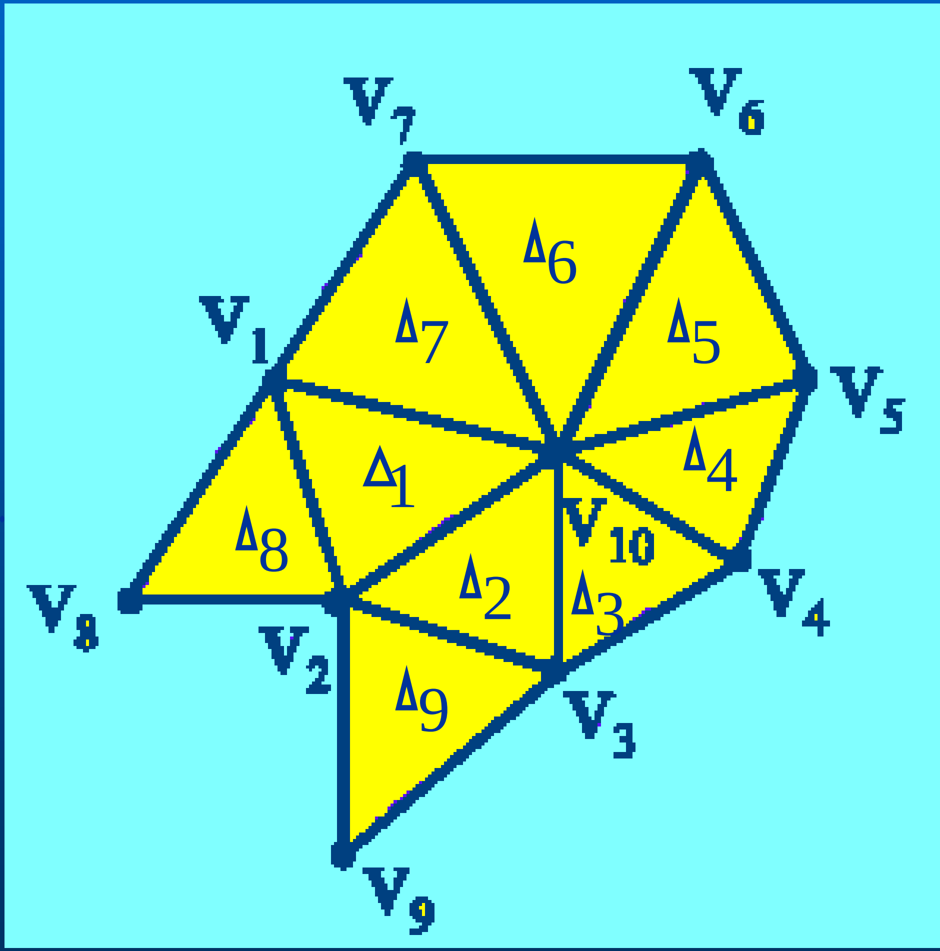
Face based

Triangle list:

(1,2,10),(2,3,10),(3,4,10)(4,5,10),
(5,6,10), (6,7,10),(7,1,10),(1,8,2),
(2,9,3)

Neighborhood list:

(2, 7, 8), (3,1,9), (4,2,-1),
(5,3,-1), (6,4,-1), (7,5,-1),
(1,6,-1), (-1,1,-1), (-1,2,-1)

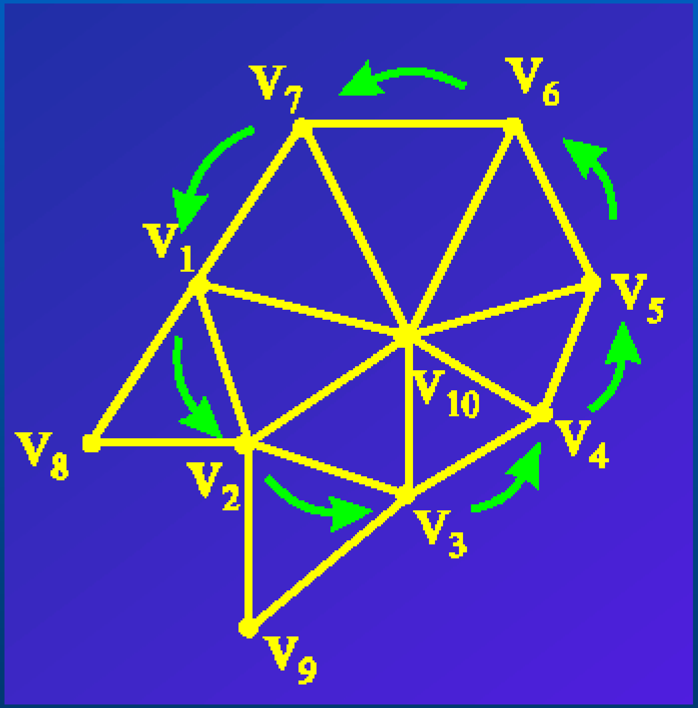


+ adds adjacency relations to standard format
+ supports breath first traversal of the mesh
- no direct access to the neighborhood of a vertex
- relatively high storage requirements

2-manifold data structures

Vertex based

v10: (1,2,3,4,5,6,7)
v1 : (~8,2,10,7)
v2 : (~9,3,10,1,8)
v3 : (~4,10,2,9)
v4 : (~5,10,3)
v5 : (~6,10,4)
v6 : (~7,10,5)
v7 : (~1,10,6)
v8 : (~2,1)
v9 : (~3,2)



+ the neighborhood of a vertex is accessible constant time
+ the boundary of a mesh with only one border can be extracted in time linear to the output size
+ storage efficient

Error metrics

Error on manifold surfaces

S – input manifold surface

T – approximating simplicial mesh for *S*

E(T,S) – error function

a. Let *S* is known at every point and function $f : S \to T$ is defined, then we can define a difference function $\delta_T : S \to R$ as $\delta_T(X) = |X - f(X)|$ and

$$E(T,S) = || \delta_T ||_S$$

where $|| . ||_S$ – some norm on functions defined over *S* – e.g., L_2 norm or L_∞ norm

Error on manifold surfaces

b. Let S is known only at a finite set of sample points D and for each triangle $t \in T$ subset D_t of T is known (D_t is approximated by t), then for each point $X \in D$ we can define the difference $\delta_T(X)$ as the Euclidean distance between X and the triangle t related to it

$$E(T,S) = \| \delta_T \|_D$$

where $\| . \|_D$ – discrete norm – e.g., the mean square or the maximum of differences over data

Error on manifold surfaces

c. No direct correspondence between a point of *S* and its representative in T is known. Some combination of Hausdorff distance and usual norms can be adopted.
For instance, we can define

$$\delta_T(X) = \min_{Y \in T} | X, Y |, \; X \in S$$

$$\delta_S(Y) = \min_{X \in S} | X, Y |, \; Y \in T$$

$$E(T,S) = \oplus(\| \delta_T \|_S , \| \delta_S \|_T$$

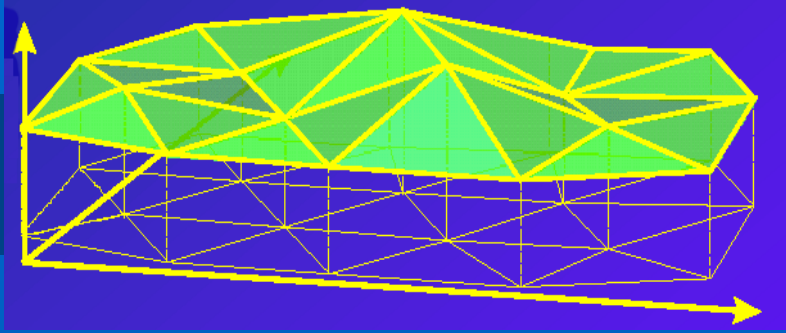
where $\oplus$ can be: an average, or a max, or a min, or a projection operator

Error on parametric surfaces

- In parametric space, domain Ω and trimmed curves of $\partial\Omega$ of each patch are approximated with $\tilde{\Omega}$ and $\partial\tilde{\Omega}$ (polygonal)
- $\tilde{\Omega}$ is triangulated and each $t \in \tilde{T}$ is projected into physical space through a function to obtain mesh T approximating S

$$
\begin{array}{ccc}
\Omega & \overset{\tilde{\rho}}{\rightarrow} & \tilde{\Omega} \\
\varphi\downarrow & & \downarrow\tilde{\varphi} \\
S & \overset{\rho}{\rightarrow} & T
\end{array}
$$

Error on scalar fields



Domain Ω is approximated through a poligonal $\tilde{\Omega}$ where a triangulation $\tilde{T}$ is defined

Approximating mesh T is defined by the image of piecewise linear function $\tilde{\varphi} : \tilde{\Omega} \rightarrow R$

An error function – analogous to that of parametric surfaces

If surface F is known only on a finite set of samples D, $\Omega \equiv \tilde{\Omega}$, therefore $e(X) = | \varphi(X) - \tilde{\varphi}(X) |$ and

$$E(T,F) = || e(X) ||_D$$

where $|| . ||_D$ – discrete norm – e.g. the mean square error or the maximum error at all data points

Surface simplification

Simplification Goals

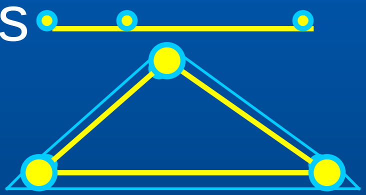
- Treat large meshes (> 1000 M triangles)
- High processing rates (O(n) time complexity)
- Controlled approximation of the original model : approximation error less than a predefined error tolerance e in 3D
- Form „good“ approximations to original mesh: visual, geometric, data-dependent; preservation of details

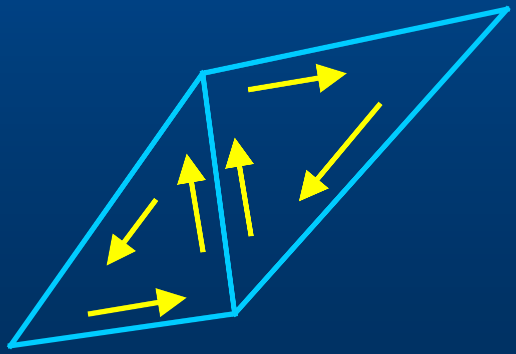
Simplification goals (cont .)

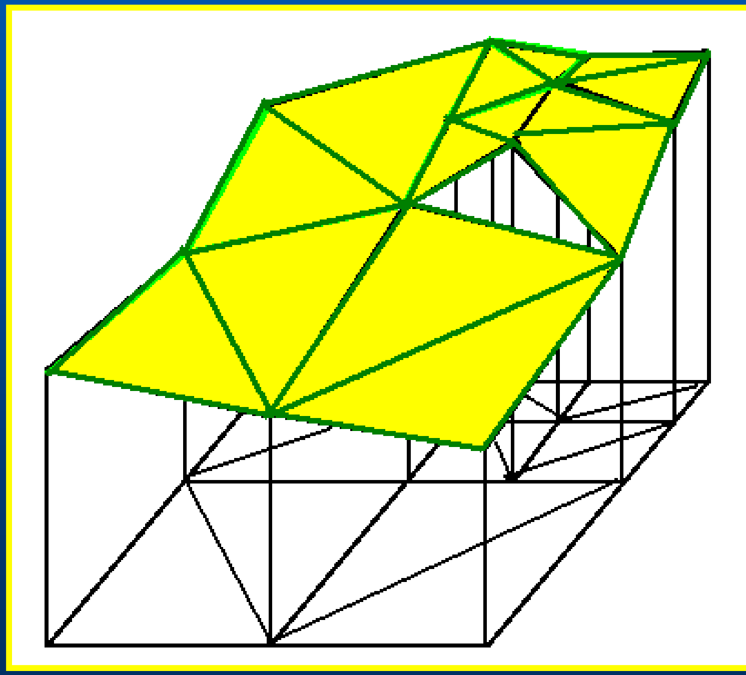
- Mapping of the original vertices to the reduced triangulation (e.g. for texture)
- Equiangularity of the reduced triangulations
- Several levels of detail (LOD)
- Smooth transition between different levels
- Simple hierarchies of different LOD
- Merging of different LODs

Simplification goals (cont .)

- Conform triangulations
- Immune against the following anomalies (in tessellated 2-manifold models)
  - Degenerate triangles
  - Duplicate triangles
  - Degenerate edges
  - Inconsistent edges







Surface mesh simplification

Optimal simplification strategy?

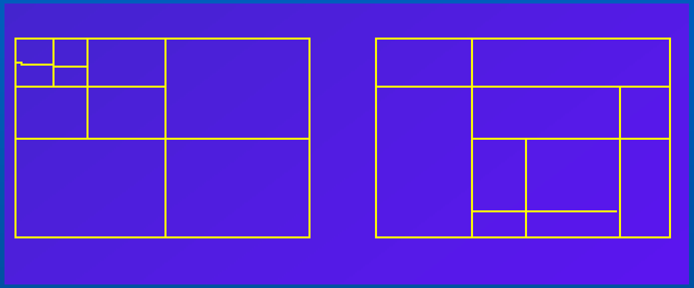
Refinement strategies

- Hierarchical triangulations
- Delaunay pyramid

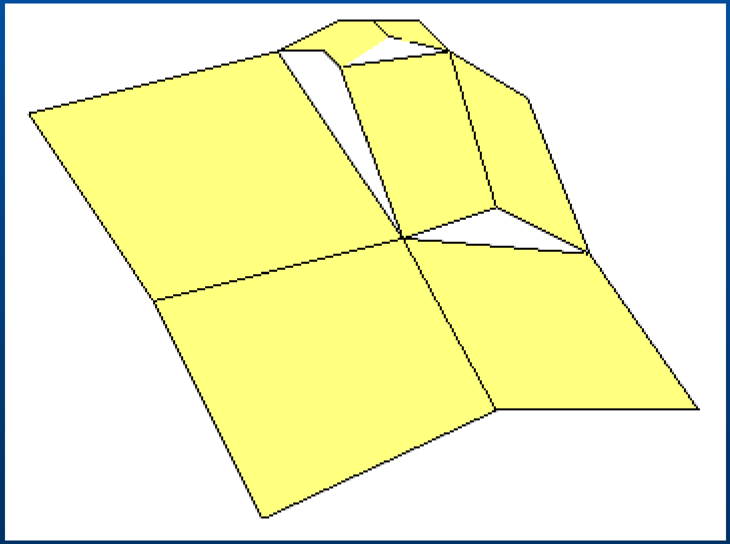
Decimation methods

- Vertex, edge, face decimation
- Clustering methods

Approximation of parametric surfaces using hierarchical subdivison



Outline: Construct approximating mesh by recursively subdividing a surface.
Examples: Quadtrees, kD-trees…

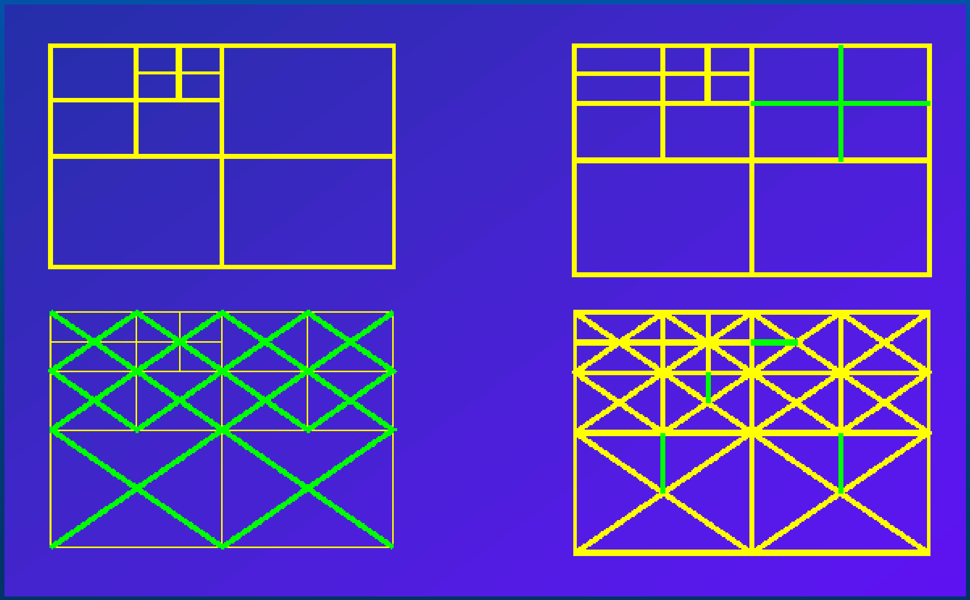


Restricted quadtrees

(von Herzen, Barr 1987)

Restricted quadtree: adjacent leaves are allowed to differ for no more than one level

Each quadrant is triangulated according to predefined patterns:



Summary (Quadtrees)

+ Canonical strategy for the refinement
+ Simple measurement of approximation error
+ Suitable structure for FE-computations
+ Very compact data structure, no need to encode information on connectivity, dependencies, etc.
+ Very fast to traverse
+ Easy to extend for the decimation approach

Summary (Quadtrees , cont)

- data must be distributed on a regular grid
- patches must have rectangular or triangular domain
- difficult to handle trimming curves
- difficult to extend to nonparametric surfaces
- features in the data set not aligned to the regular grid cannot be represented well

Refinement algorithms

Given: parameterized surface or scalar field (over rectangular or triangular domain)

General outline of refinement algorithms:

M^{or} : original mesh (mostly defined by surface points)

max_error : allowed approximation error between original and simplified mesh

M^0 : initial Mesh (rectangle, two triangles)

$M := M^0$

$e := ||M - M^{or}||$

While *(* e > *max_error)*

  Insert *one or more points into the triangulation M*

  $e := ||M - M^{or}||$

Refinement algorithms

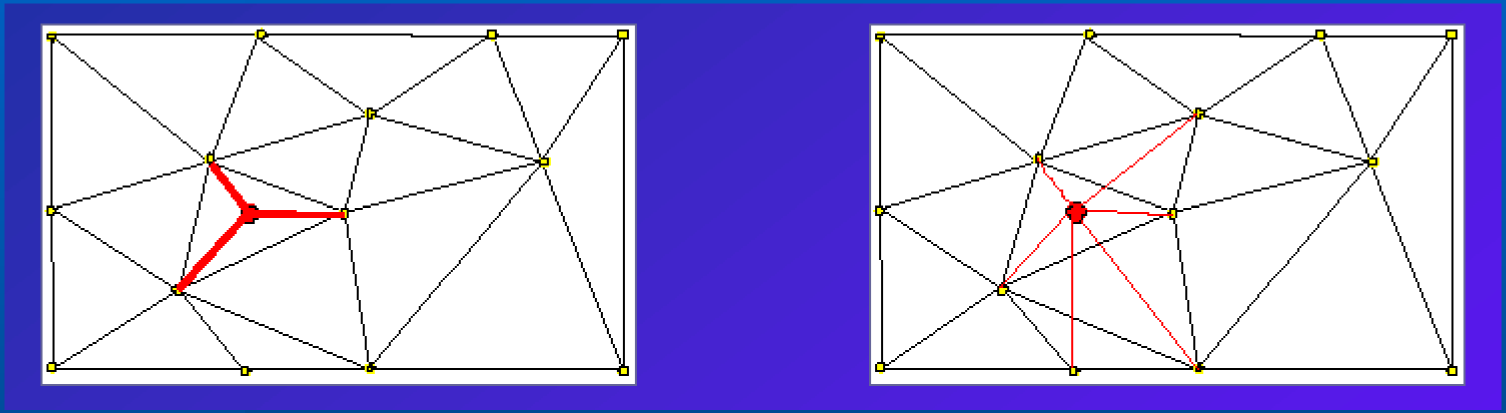
What vertex is inserted on current step?

Solution :

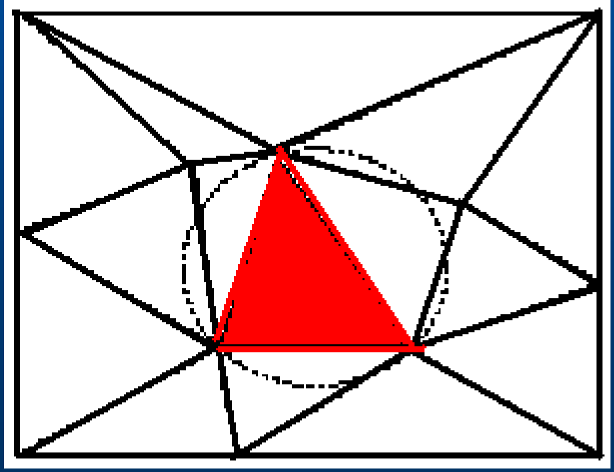
Insert recursively that vertex into the domain triangulation that causes the ***highest error*** until the L_∞ error between original M^{or} and simplified M triangulation is less than a predefined threshold *max_error*.

The Delaunay Pyramid

How to insert a new vertex into the current triangulation?



Solution: Use a Delaunay triangulation in the domain



A triangulation of a point set $P \subset R^2$ is called ***Delaunay triangulatio***n, if in the inner of the circumcircle of each triangle there is no point $p \in P$.

Delaunay Pyramid

R.Klein, 1997

Advantages of the Delaunay triangulation:

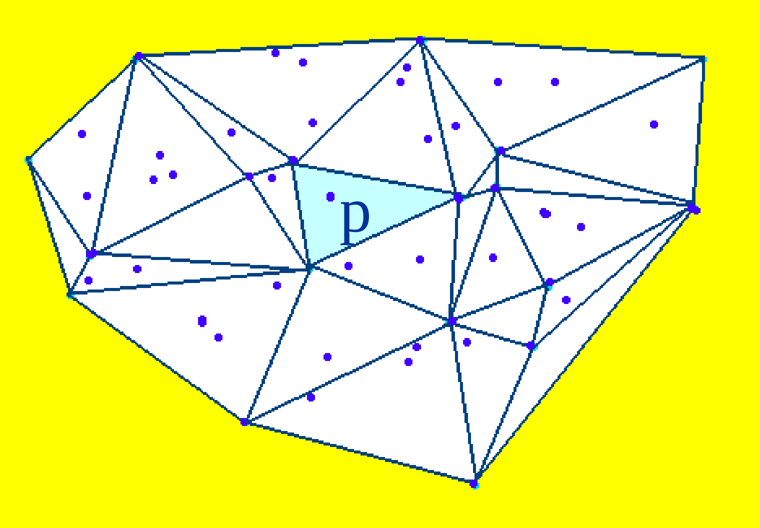
- The location of the inserted vertices may be arbitrary.
- Delaunay triangulation maximizes the minimal angle in the triangulation $\Rightarrow$ good aspect ratios of triangles.
- The Delaunay triangulation of a point set is unique.
- Connectivity is implicitly given. Only a sequence of vertices in the domain must be stored. Vertices on the triangulation of the surface can either be computed by evaluating the parametric function or the z-values of the vertices are stored.
- Insertion and removal algorithms allow to change between different levels of detail.

Fast insertion algorithm

M. Garland, P. Heckbert, 1997

Problems:

- Find the vertex causing the maximum approximation error
- Find the triangle containing the new point p to be inserted



	Naïve Algorithm:	Optimized algorithm:
Selection	– O(n)	– O(log i)
Insertion	– O(i)	– O(i)
Recalculation	– O(ni)	– O(ni)
Worst case	– $O(m^2 n)$	– O(mn)
Average	– O(mn)	– O((m+n)log m)

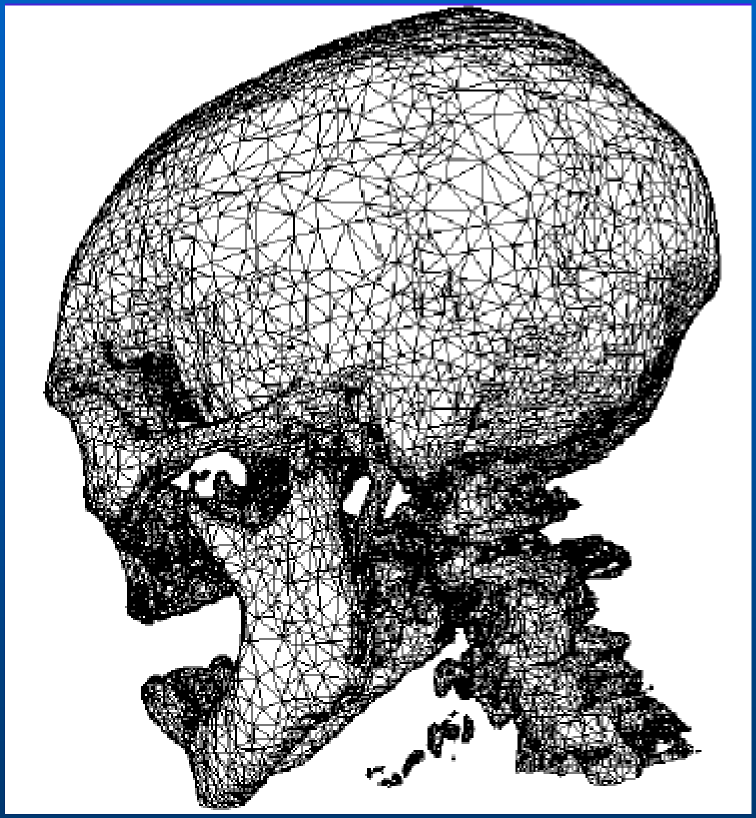
Summary (Delaunay Pyramid)

+ ***Extremely compact*** model for parametric and manifold surfaces
+ Storing the simplest mesh plus the sorted vertices inserted during refinement tagged with the approximation error of the triangles incident to the vertices delivers a ***finite set of different LOD*** with ***controlled approximation error***.
+ Easy incorporation of borders, feature points, edges, etc.

R. Klein, W. Strasser, 1998

Simplification of not necessarily parameterized surfaces

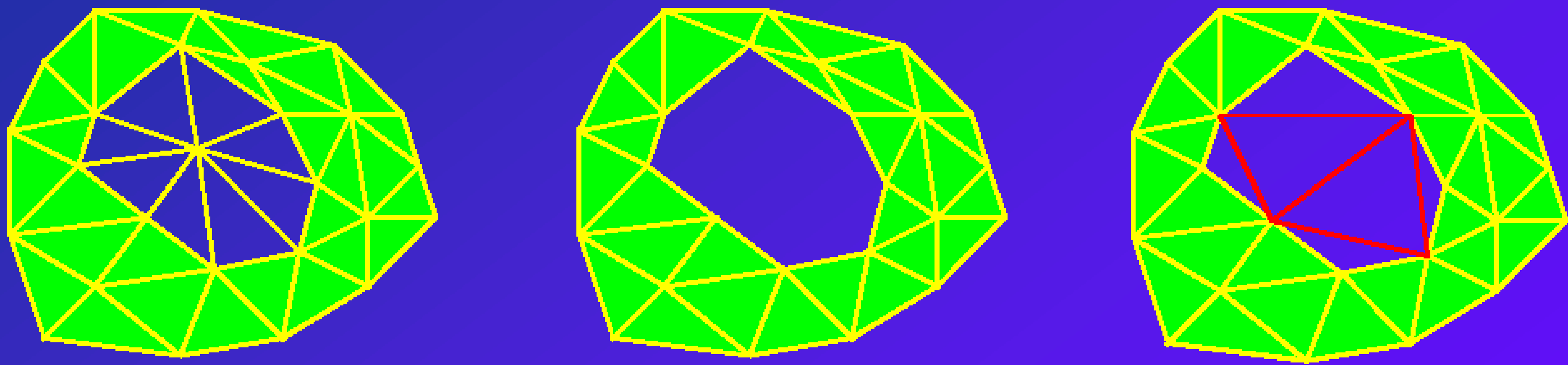
In many cases no common parameterization of the surface is known.
There are even surfaces for that a common parameterization do not exist
Need for simplification strategies that are not based on a parameterization of the surface!



Schroeder et. al. 1992:
Decimation of triangle meshes

Vertex decimation

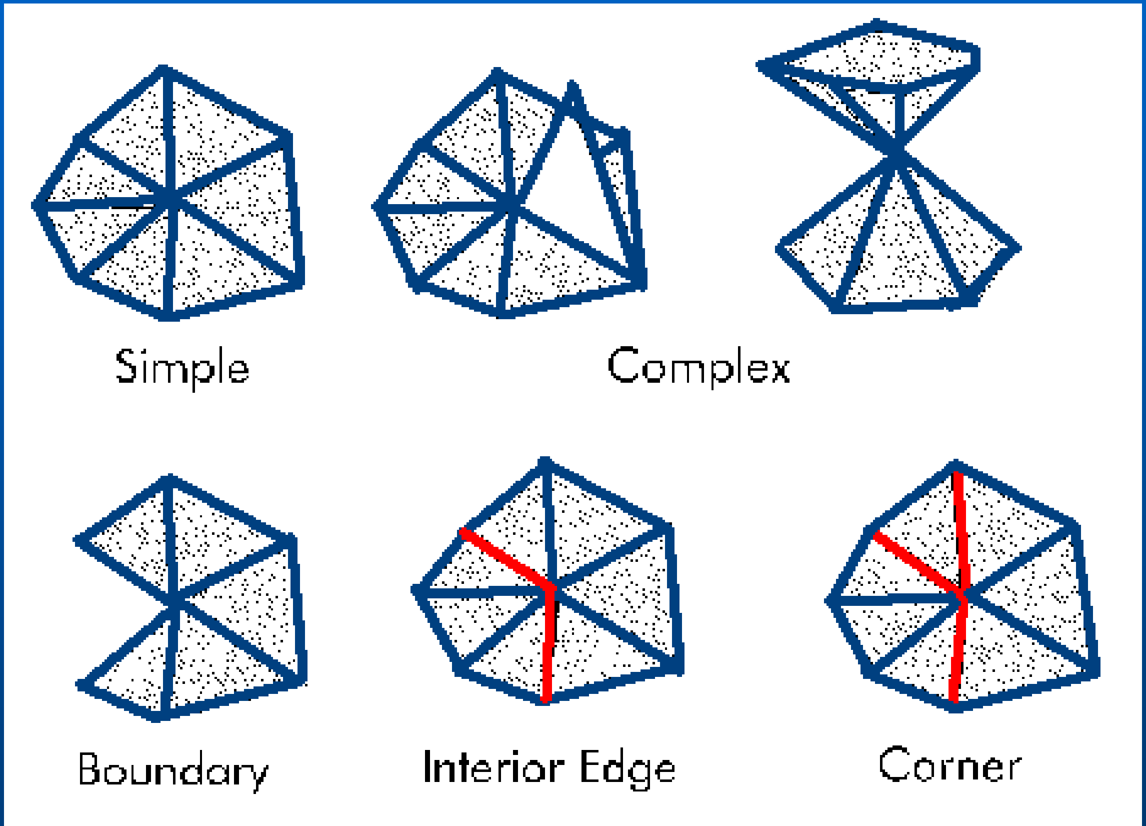
Simplification by ***successively removing*** vertices:



Evaluate local topology and geometry	Delete vertex and incident triangles	Retriangulate remaining hole

Vertex decimation

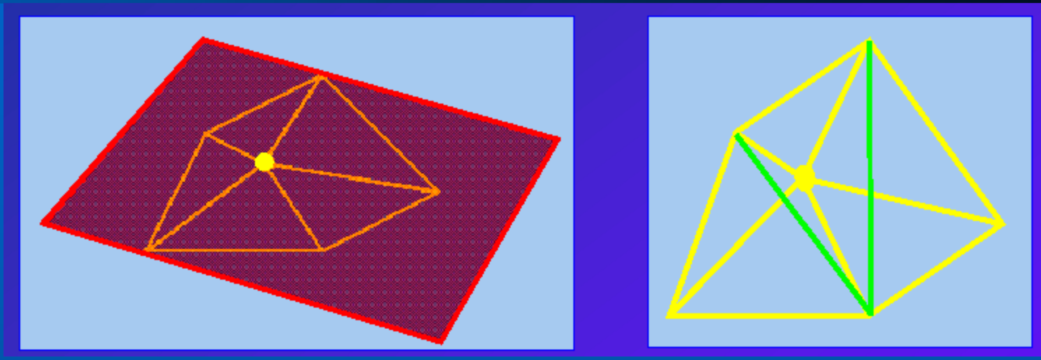
Schroeder et. al. 1992, 1997

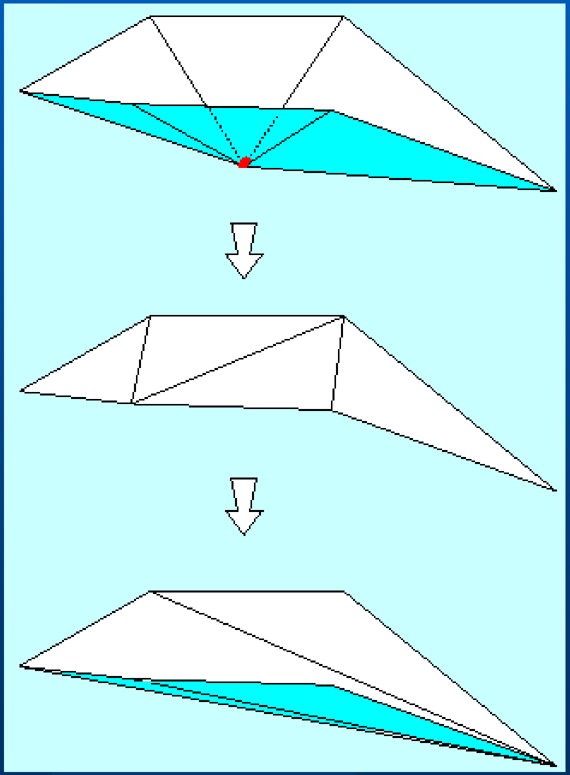


Evaluation of the local topology

Vertex decimation

Retriangulation of the hole in a suitable plane



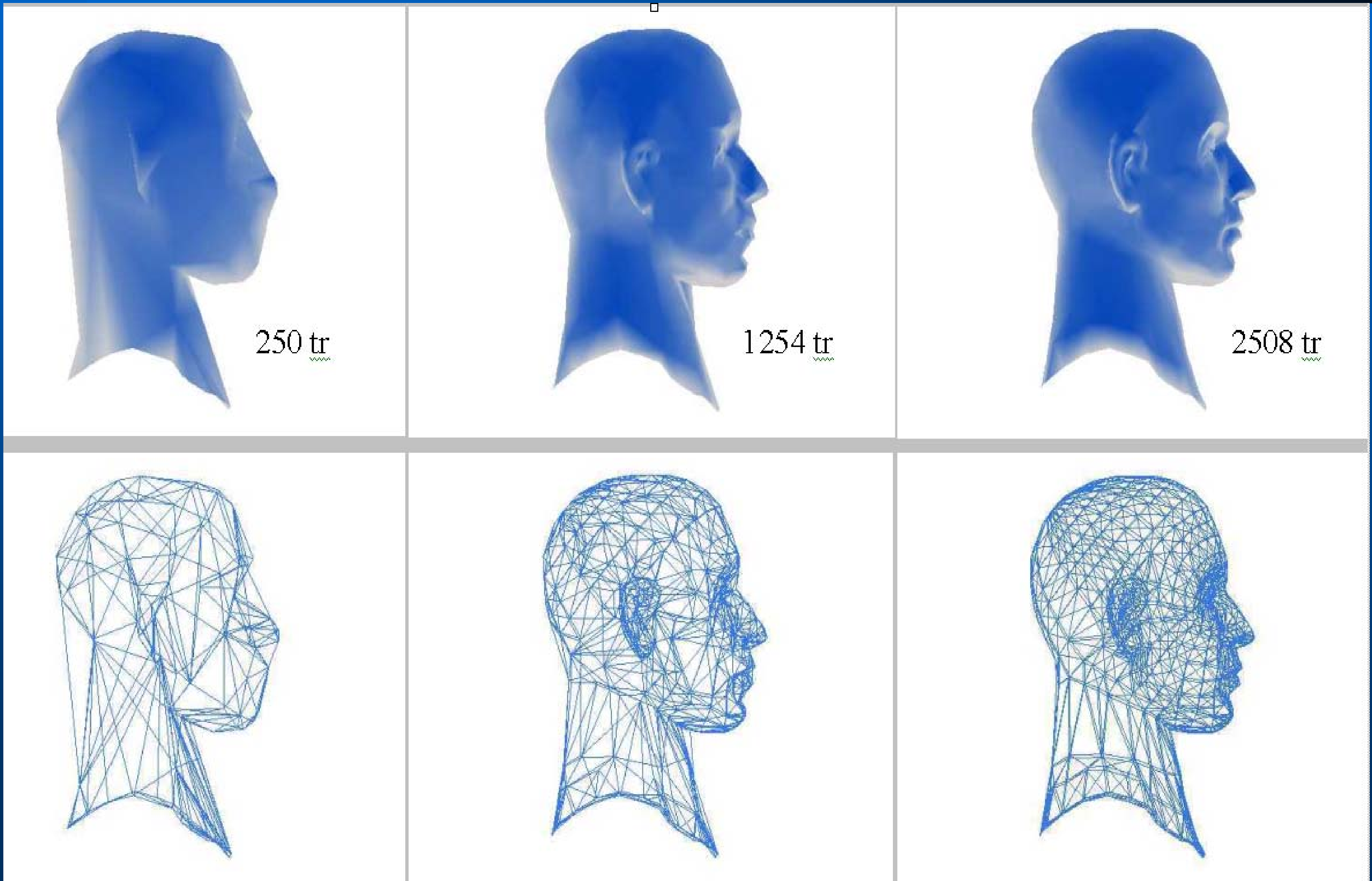


Different retriangulation strategies:

- Constrained Delaunay
- Data dependent (find triangulation that approximates the geometry in the neighborhood of the removed vertex best.) For a fast implementation edge swapping can be used.

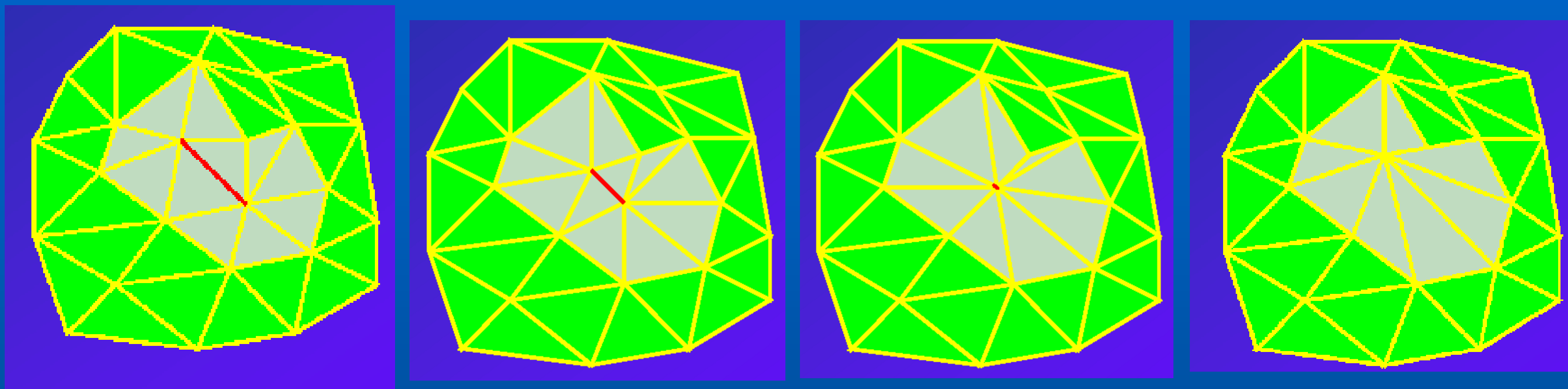
Example from Puppo, Scopigno, 1997

Vertex decimation

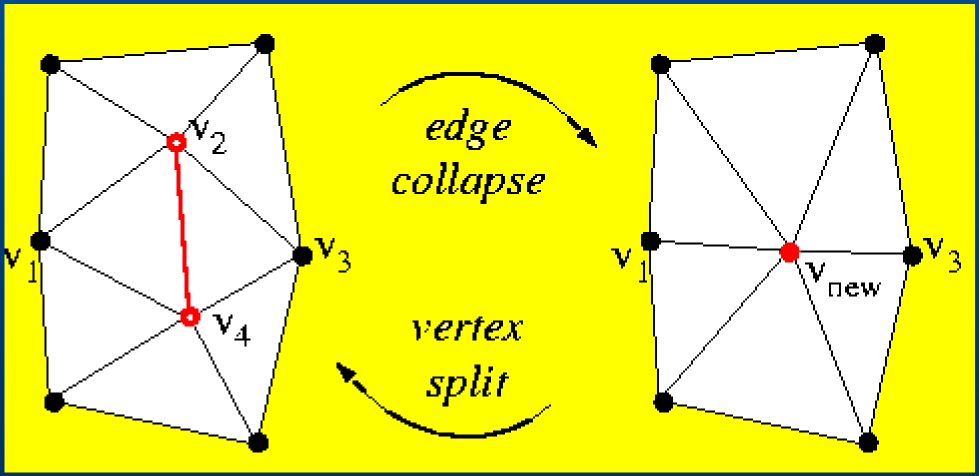


Edge Collapse

(Gueziec95, Hoppe96, Ronfrard-Rossignac96)

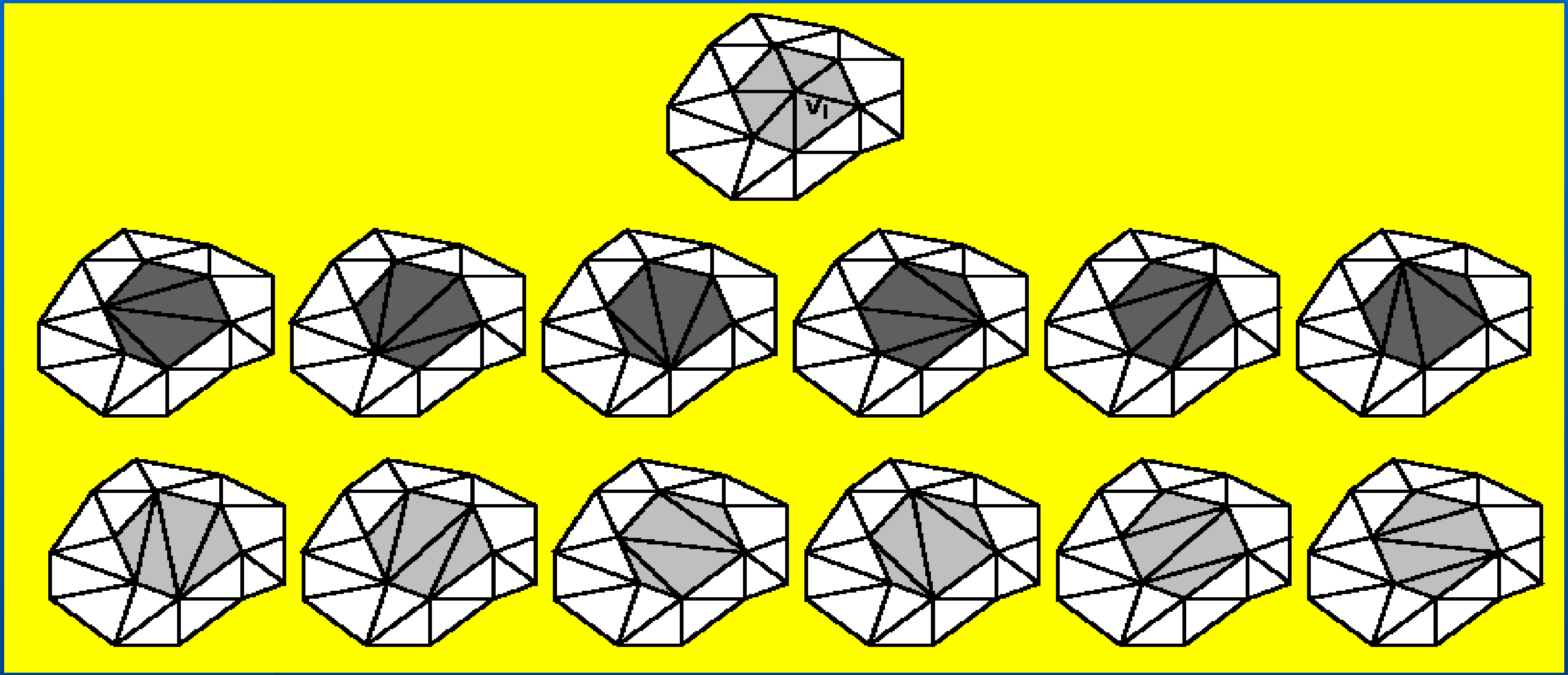


- ***Sufficient*** for simplification
- ***Simple*** collapse operation
- ***Simple inverse*** split operation
- Position of new vertex can be optimized



- Collapsing to one of the original vertices is called ***half egde*** collapse

Vertex decimation or half edge collapse?

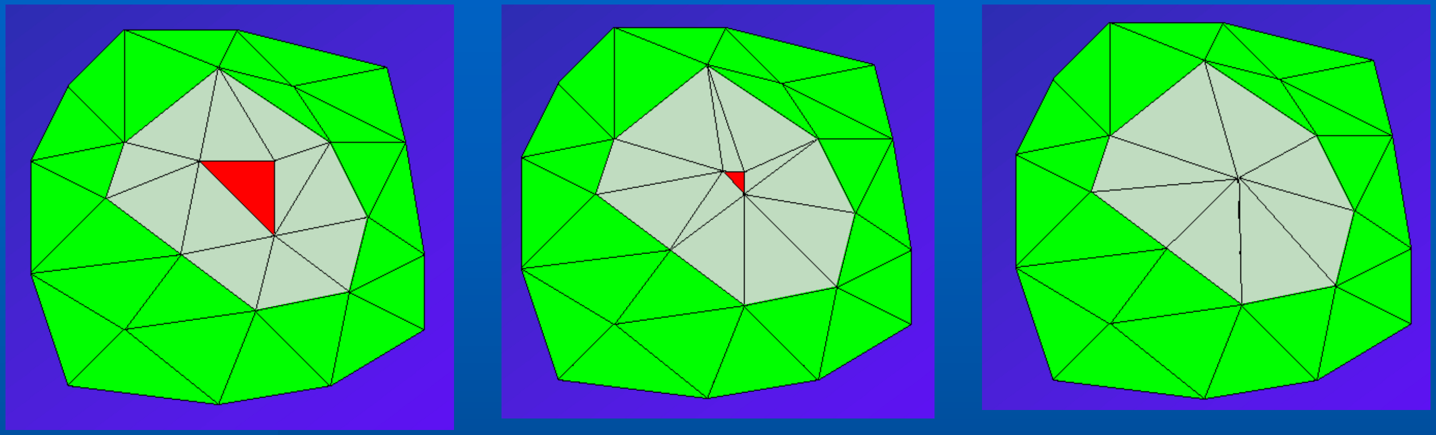


Half edge collapse can be interpreted as special kind of vertex removal

For some applications vertex removal is superior to edge collapse, especially if there are requirements on the quality of the triangulation

Triangle Collapse

(Hamman, 1994)



Can be realized as two successive edge collapse operations

New vertex position can be optimized

Decimation algorithms

In which order should vertices, edges or triangles be removed or collapsed?

Schroeder et. al.: No special order, traverse vertex list several time and check which vertices can be removed

Idea (Douglas- Peucker): Build priority queue of vertices sorted by the approximation error

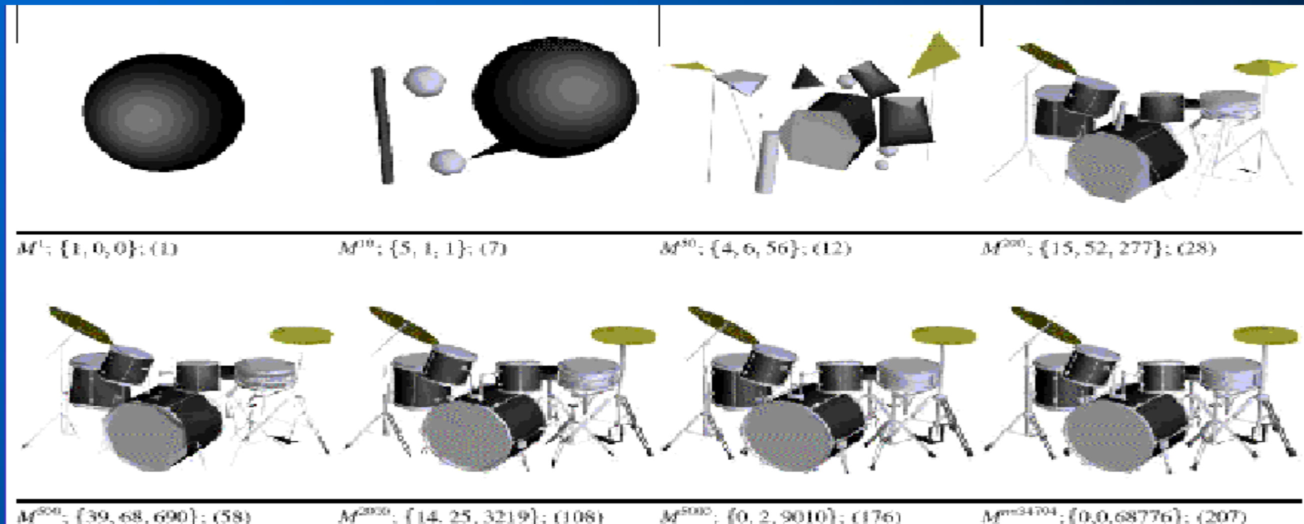
Decimation algorithm outline

Simplification

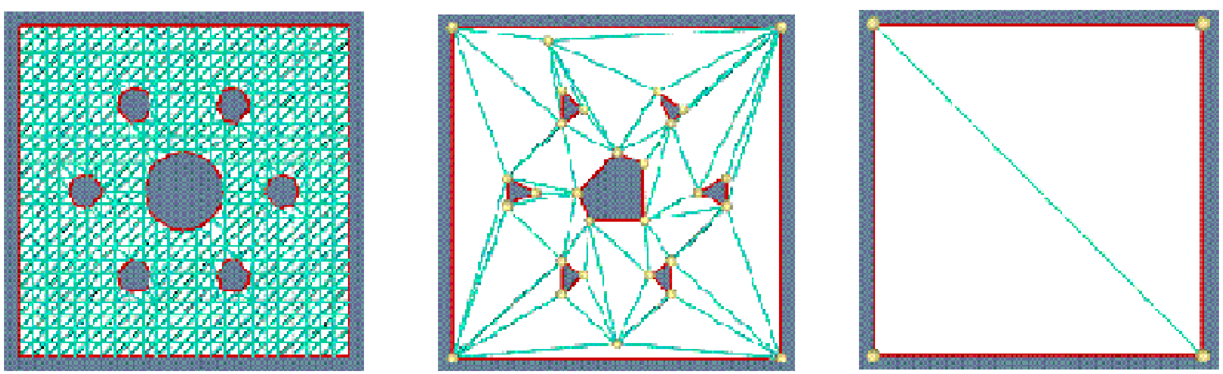
Until error high or approximation small enough

- Find vertex, edge or triangle that introduce the least error
- Perform local simplification operation
- Update priority queue
- Save sequence of simplification operations and their inverse

Topology modifying algorithms



Popovic, Hoppe, Siggraph 97



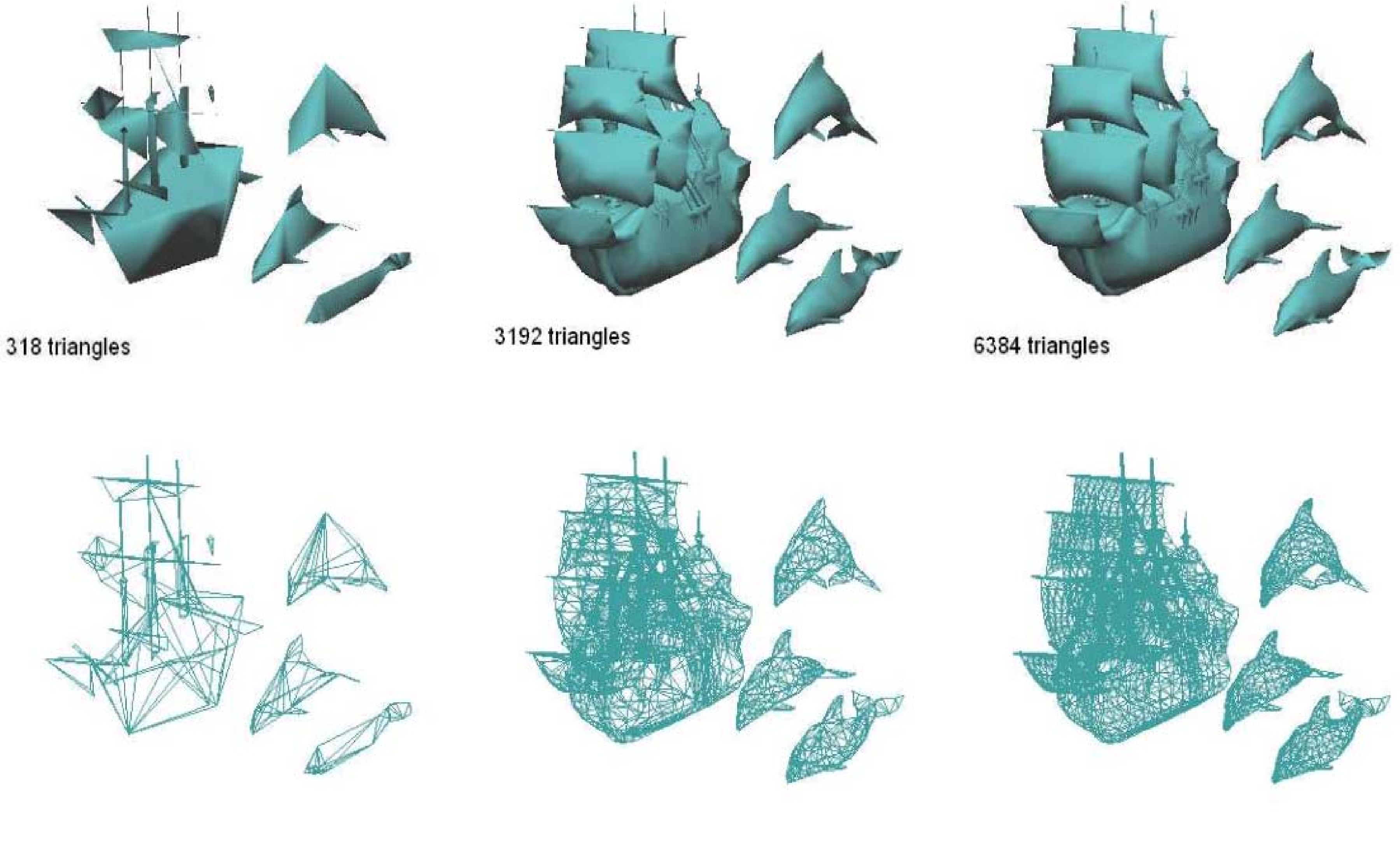
a) Original model
1,132 triangles

b) Topology unchanged
43 triangles

c) Topology modified
(2 triangles)

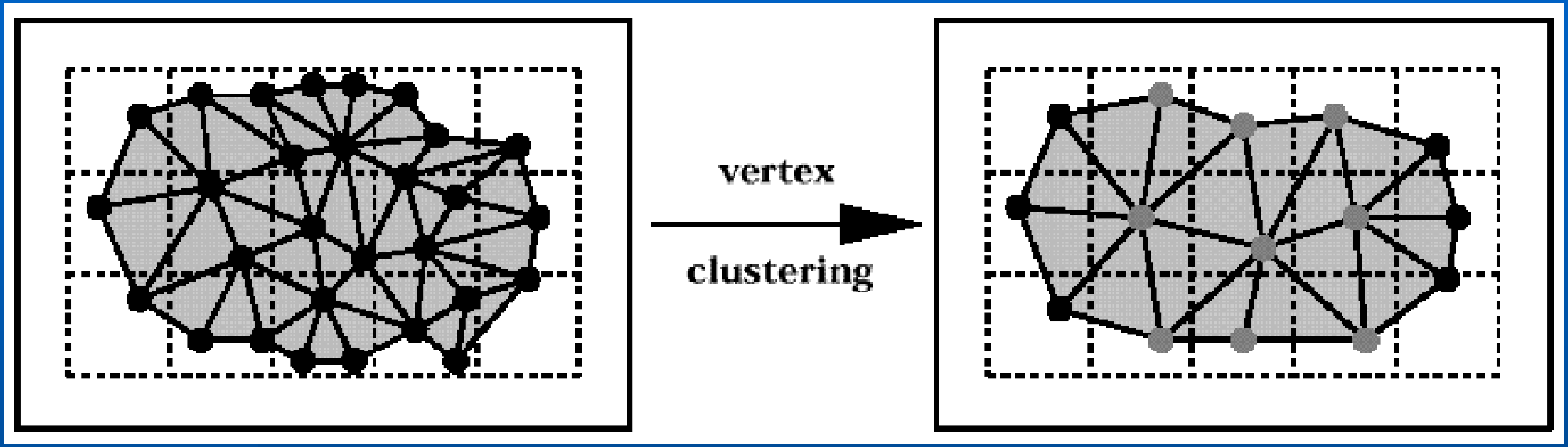
William Schroeder, Siggraph 97

Topology modifying algorithms



Vertex Clustering

Borel, Rossignac 1993



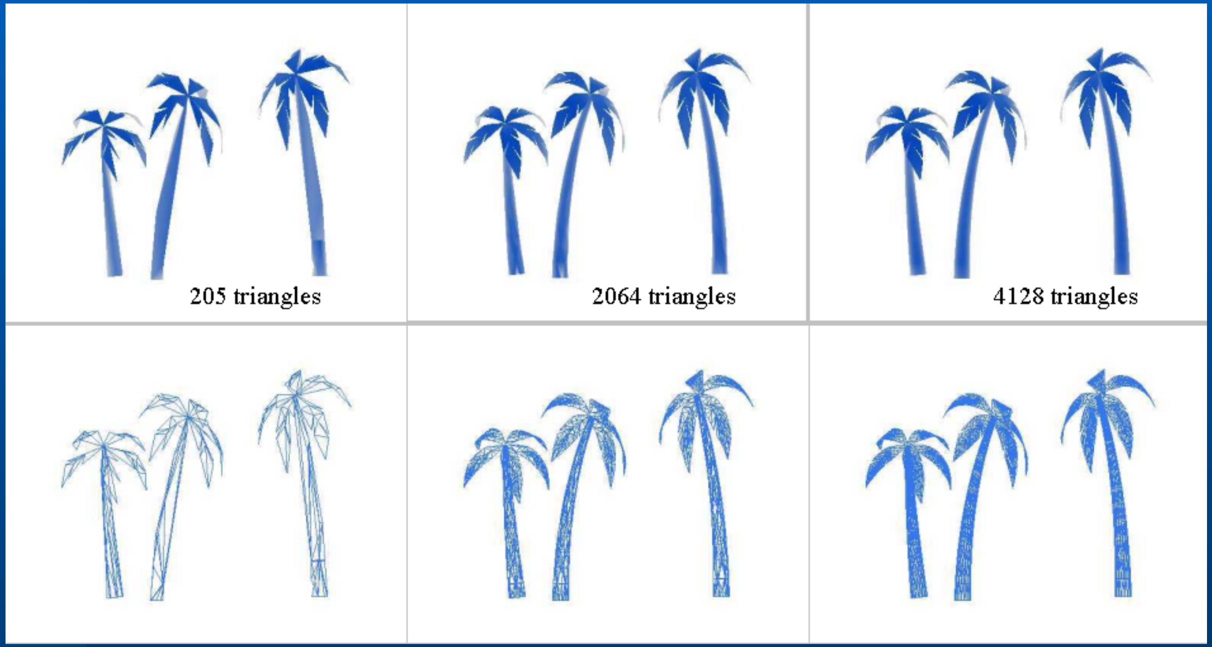
- Error bounded in a Hausdorff sense
- Simple to implement
- Very fast
- Not topology preserving
- Bad geometric accuracy of the original mesh
- Produces very crude approximation

Multiresolution modeling

Multiresolution model is a model that can provide different representations, depending on the level of detail required.

Discrete multiresolution model

Discrete multiresolution model consist of a set of increasingly simpler approximations (set of discrete LODs) and the threshold parameters to control the switching between them.



! We unable to vary the level of detail over different parts of the model

Selective refinement

All schemes based on a local simplification operation naturally defines a *linear sequence of LOD approximations*

For real-time applications *view-dependent refinement* is required:

- View-dependent visualization of large objects with guaranteed screen space geometric error
- View-frustum culling
- Surface orientation (back-facing parts as coarse as possible)
- Illumination based refinement

Continuous Multiresolution model

Desirable properties:

Continuity through domain (can't be cracks due to abrupt transition between different LODs)

Efficiency – (must support efficient – short time - query processing)

Optimal size (model size mustn’t be considerably higher than the size of the mesh at the higher resolution)

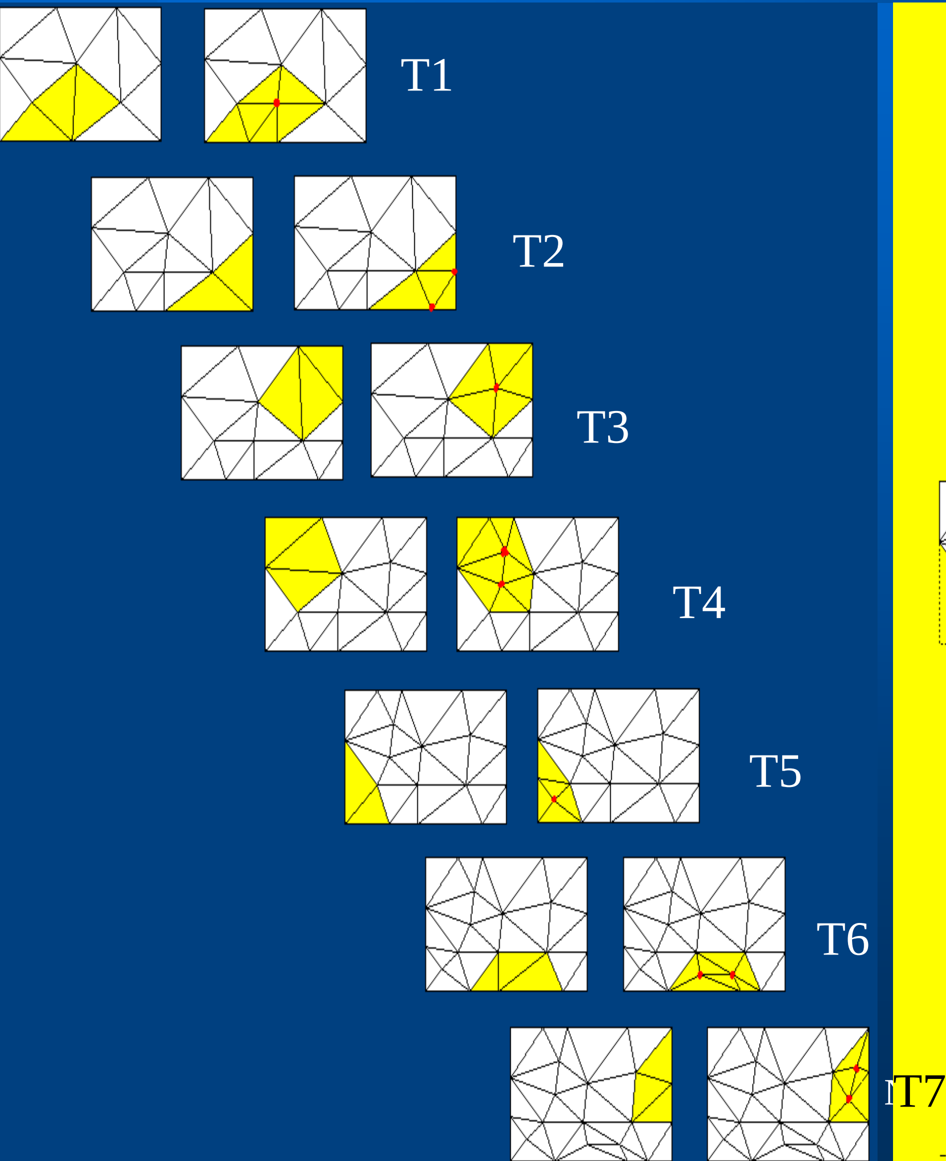
Continuity across resolution (abrupt changes should be avoided in changing a representation into another at a close LOD)

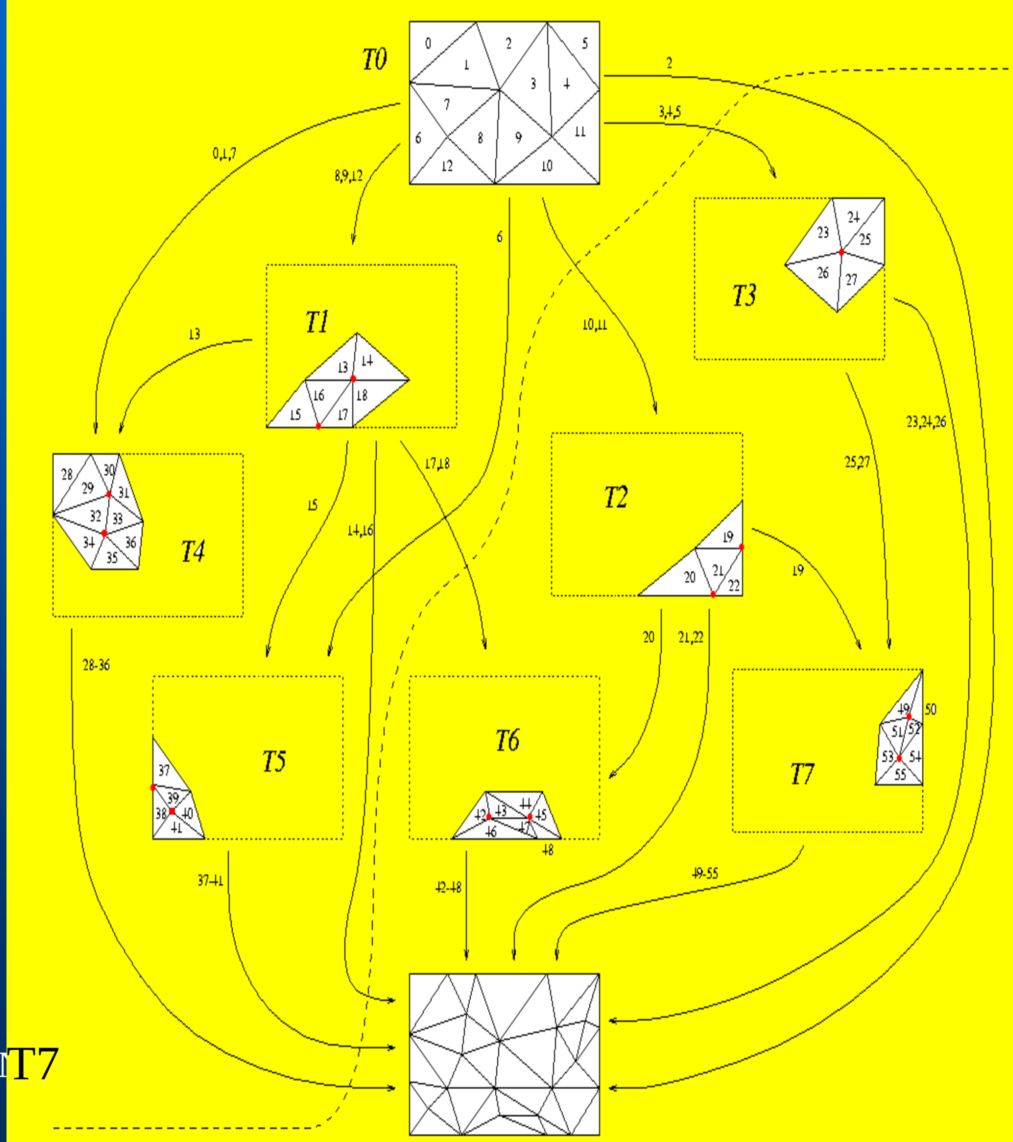
The multi - triangulation ***(Puppo 1996)***

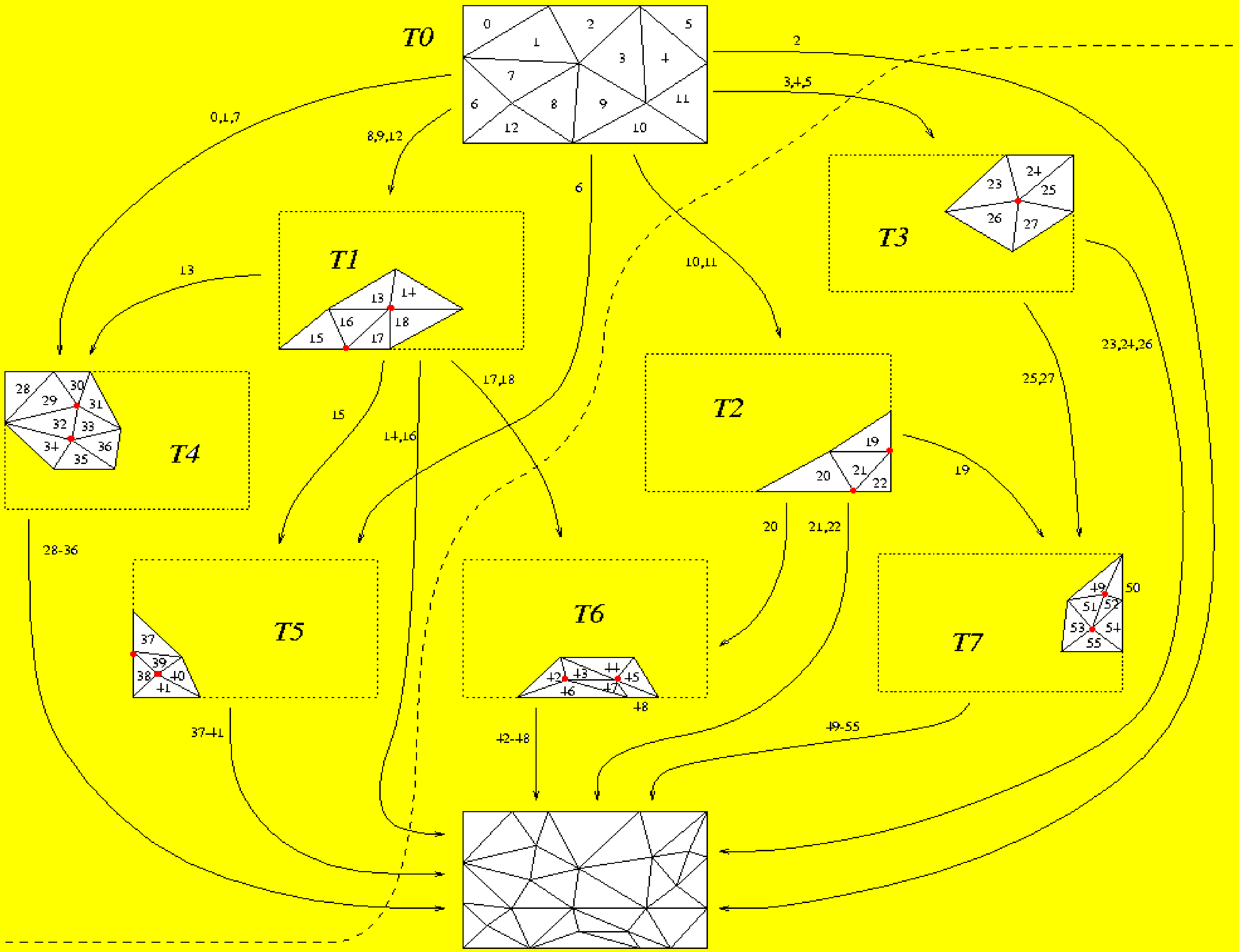
- The multi triangulation (*MT*) is a ***general framework*** for multiresolution meshes
- All multiresolution meshes described in the literature can be interpreted as ***special cases of the MT***
- ***MT*** can be obtained as the evolution of a mesh through ***iterative local modifications***
- Hierarchy of meshes forms ***directed acyclic graph (DAG)*** of fragments
- ***Local modification*** (or ***local update***) - operation that replaces group of triangles with another group of triangles covering the same area

The multi - triangulation

from Puppo, Scopigno, 1997

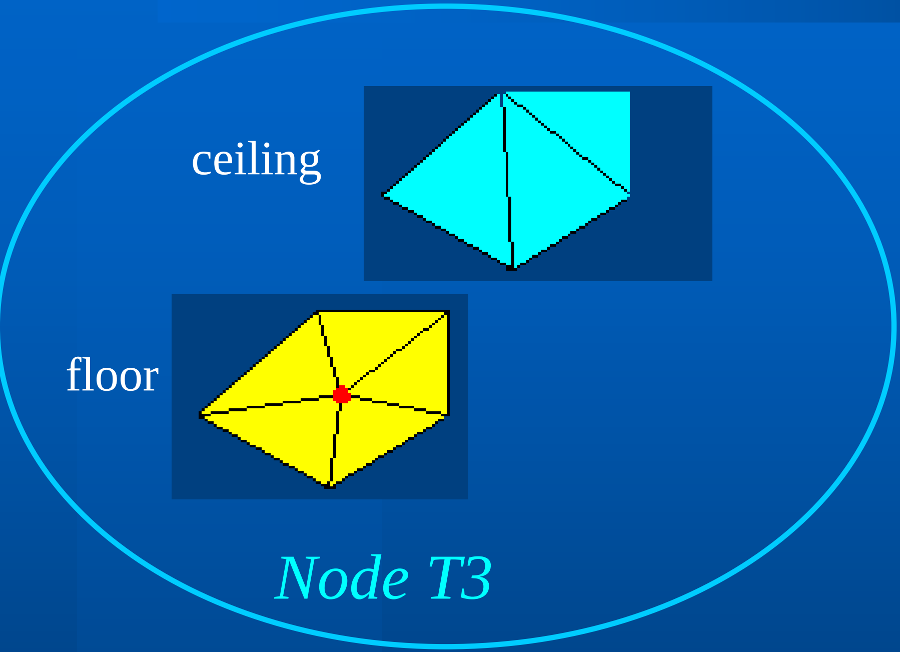






T0
T1
T2
T3
T4
T5
T6
T7
0,1,7
8,9,12
6
10,11
2
3,4,5
13
15
14,16
17,18
20
21,22
19
25,27
23,24,26
28-36
37-41
42-48
49-55

Data structure of the MT



structure ***Arc***
Upper node index
Lower node index
First triangle index
Last triangle index
Next arc index (the same dest. Node)

structure ***Node***
First outgoing arc;
Last outgoing arc;
First entering arc index
Last entering arc index
Number entering arcs

structure ***MT***
Vertex array;
Triangle array;
Arc array;
Node array;

structure ***Triangle***
Vertex Index [3];
Triangle error
Arc index;

The multi - triangulation (cont.)

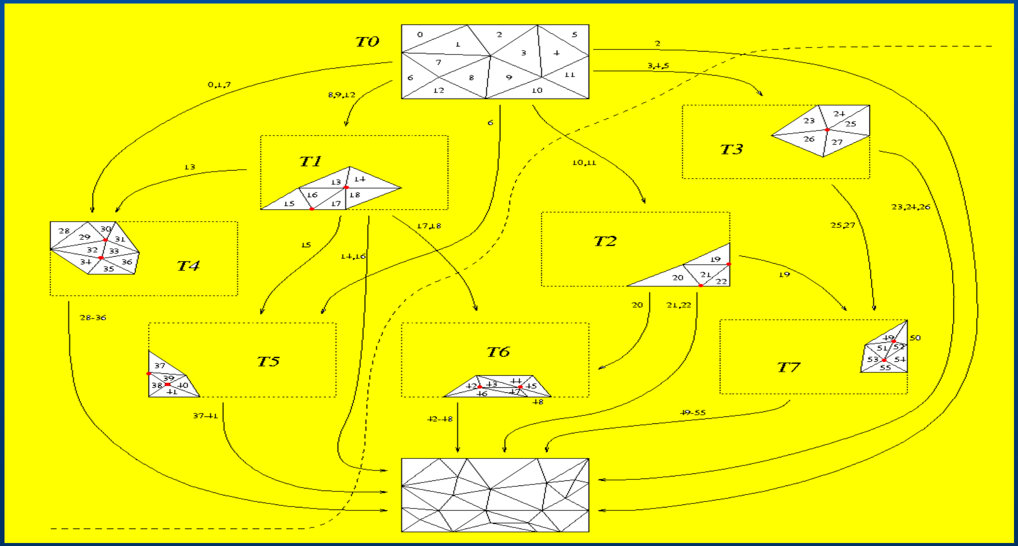
- Can easily be extended to two manifold surfaces embedded in 3D
- Independent of the different local simplification strategies (vertex removal, edge- and triangle collapse)
- Fast algorithms to refine and coarsen in distinct areas
- High storage costs

Extraction variable resolution meshes from MT

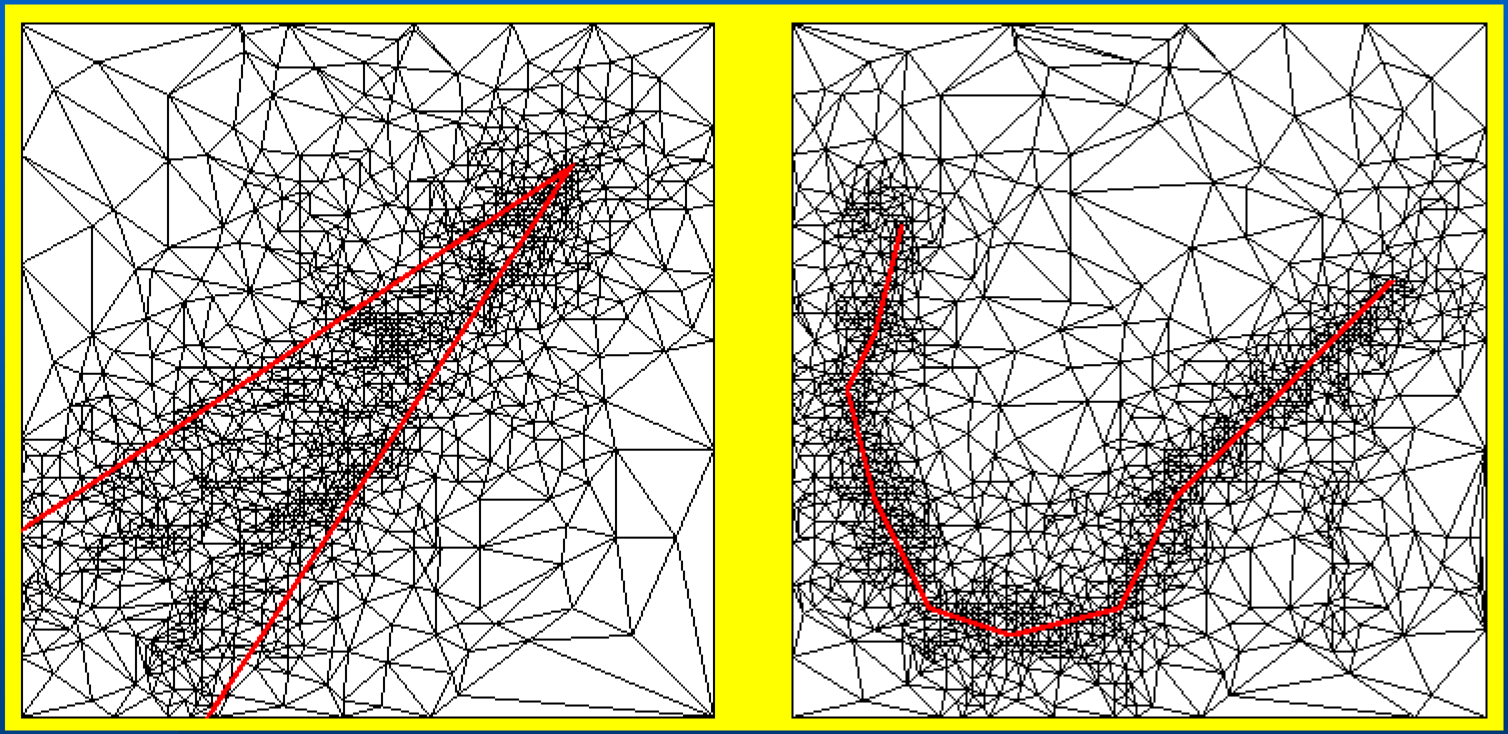
Boolean predicate *C* on the triangles :
For a given triangle *t*, *C(t)* is true if and if the resolution of *t* is acceptable.

Given a MT *M* and a resolution predicate *C* it is possible to extract a triangulation satisfying *C* in linear time in the size of *M*.

For flight simulators:
$C(t) = (\varepsilon_t \leq \min_{p \in t} \tau(p))$,
where ε_t - triangle error;
$\tau(p) = K^*dist(p,v)$;
v – view point

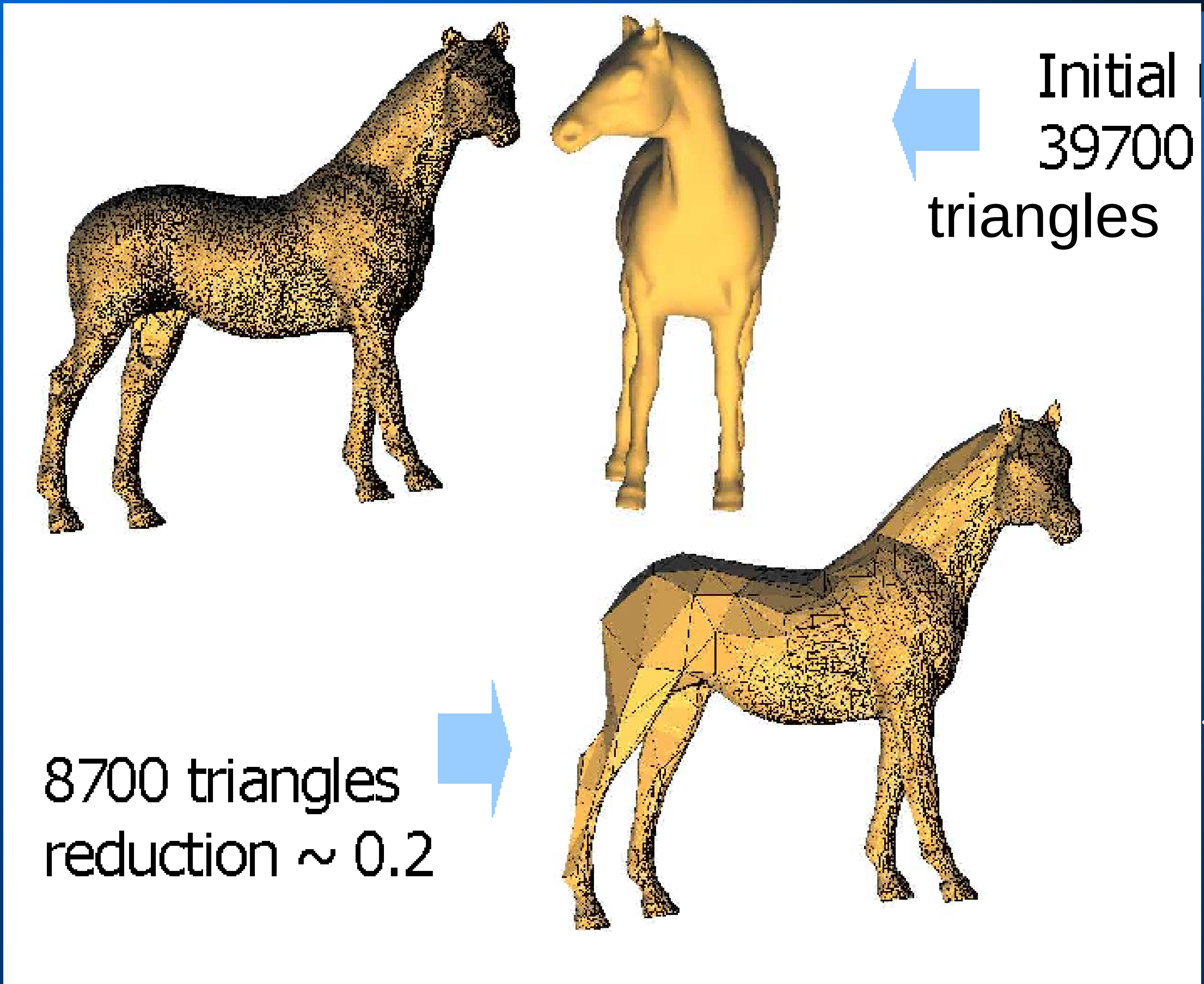


Examples of extracted models



From L.Floriani, P. Magillo “Efficient Implementation of Multi-Triangulation”, 1998

View dependent approximation



← Front view

References

1. M. Garland, P. Heckbert. Fast Triangular Approximation of Terrain and Height Fields. Course Notes for SIGGRAPH'97 LosAngeles,1997
2. W.J. Schroeder. A Topology Modifying Progressive Decimation Algorithm. Course Notes for SIGGRAPH'97, Los Angeles, 1997.
3. E.Puppo, Variable Resolution Triangulations. 1998, http://www.ima.ge.cnr.it/STAFF/puppo-en.html
4. E.Puppo, R.Scopigno, 1997, Simplification, LOD, and multiresolution – Principles and applications, Eurographics’97 Tutorial Notes
5. R.Klein,J.Kramer, 1997, Building multiresolution models for fast interactive visualization, In Proceedings of the SCCG
6. R.Klein, 1997, Multiresolution representations for surfaces meshes
7. L.De Floriani, P.Magillo, E. Puppo, 1997, Building and traversing a surface at variable resolution, Proceedings IEEE Visualization’97, Phoenix, AZ
8. L.De Floriani, P.Magillo, E. Puppo, Selective Refinement of Surface Meshes: Data Structures and Algorithms, 1998
9. R.Klein, W. Strasser, Handling of Very Large 3D-Surface-Datasets Using Mesh Simplification and Multiresolution Modeling, 1998, http://www.gris.uni-tuebingen.de/people/staff/reinhard/CGI-Tutorial/ CGI-Tutorial.pdf