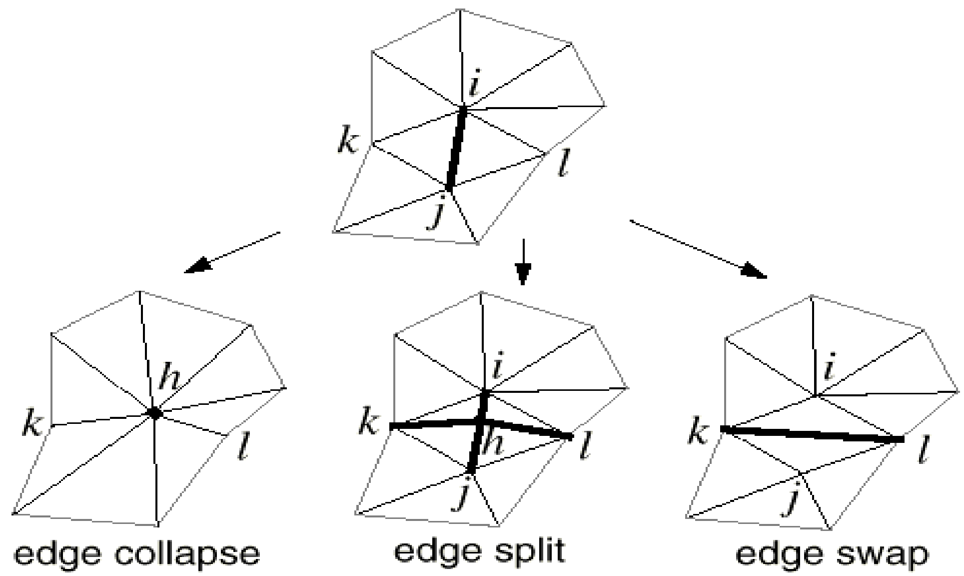
Simplification (part four): simplification algorithms for free-form meshes

Leila De Floriani and Enrico Puppo
Department of Computer Science
University of Genova

- *Incremental techniques based on local modifications*:
  - Energy function optimization: mesh optimization and progressive meshes
  - Mesh decimation
  - Simplification envelopes
- *Examples of non-incremental techniques:*
  - Face-merging
  - Re-tiling
  - Vertex-clustering

Mesh Optimization [Hoppe et al. `93]

- Simplification based on the iterative execution of:
  - *edge collapse*
  - *edge split*
  - *edge swap*



Mesh Optimization [Hoppe et al. `93]

- Quality of approximation evaluated through an energy function:

  $E(\square) = E_{dist}(\square) + E_{rep}(\square) + E_{spring}(\square)$ where

  - **E_{dist}** : sum of squared distances of the points of S* from □
    - measures error with respect to input mesh
  - **E_{rep}** : factor proportional to the number of vertices in □
    - measures the degree of simplification
  - **E_{spring}** : sum of the edge lengths
    - measures the shape of triangles in the mesh

Mesh Optimization [Hoppe et al. `93]

Algorithm structure

- Outer minimization cycle:
  - choose a legal action (edge collapse/swap/split) which reduces the energy function
  - perform the action and update the mesh
- Inner minimization cycle:
  - optimize the vertex positions of current mesh with respect to the initial mesh
- but (to reduce complexity):
  - legal action selection is random
  - inner minimization is solved in a fixed number of iterations

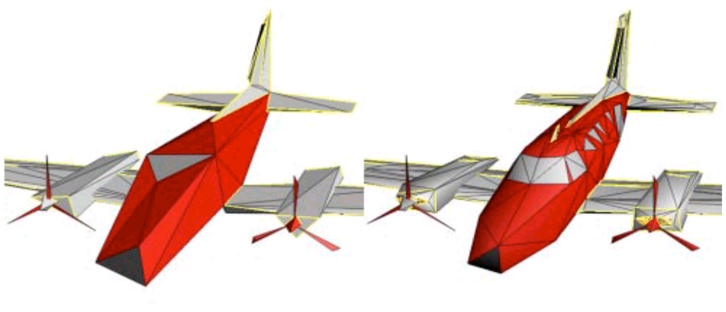
Mesh Optimization - Properties

- high quality of the results
- re-sampled vertices
- topology preserved
- high processing times
- not easy to implement and to use (selection of tuning parameters)

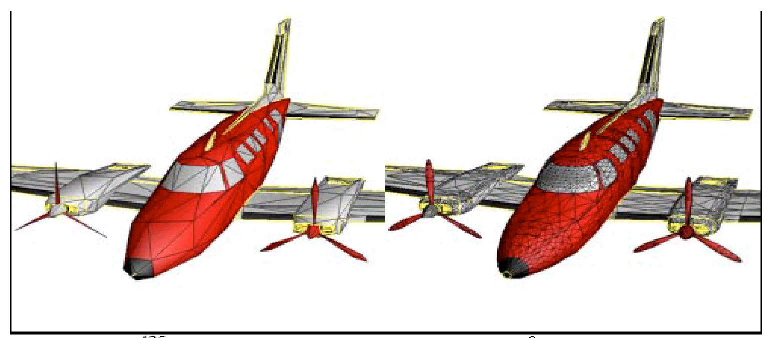
Progressive Meshes [Hoppe `96]

- It executes only *edge collapses* to reduce the *energy function*
- *Edge collapse* can be easily inverted: it stores a sequence of inverse *vertex splits* to support multi-resolution and progressive transmission
- Faster than mesh optimization

Examples:



(a) Base mesh M^0 (150 faces) (b) Mesh M^{175} (500 faces)



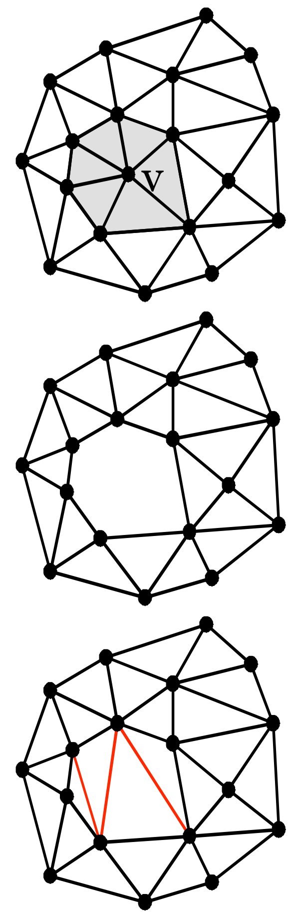
(c) Mesh M^{425} (1,000 faces) (d) Original $\hat{M} = M^n$ (13,546 faces)

Mesh Decimation [Schroeder et al, 92]

- Based on controlled removal of vertices
- Classify vertices as either removable or not (based on local topology / geometry and required precision)
- *Loop*
  - choose a removable vertex v
  - delete v and its incident faces
  - re-triangulate the hole

  until

  no more removable vertex exists or reduction rate fulfilled



Mesh Decimation [Schroeder et al, 92]

- General method (manifold/non-manifold input)
- *Algorithm phases:*
  - topological classification of vertices
  - evaluation of the decimation criterion (error evaluation)
  - re-triangulation of the removed triangles patch

- *Topological classification of vertices:* for each vertex v: find and characterize the link of v
  - *simple vertex:* link of v homeomorphic to a circle
  - *boundary vertex:* link of v homemorphic to an closed interval in R
  - *non-manifold vertex:* link of v formed by several connected components

- Non-manifold vertices are not *removable*

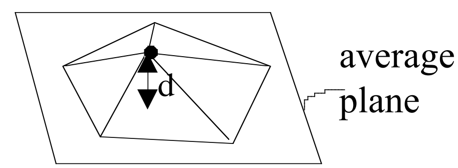
Mesh Decimation [Schroeder et al, 92]

- Decimation criterion: a vertex is *removable* iff
  - *simple vertex:* distance vertex to face-loop-average-plane is lower than e_{max}

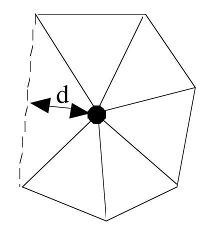
  or
  - *boundary vertex*: if distance vertex to new boundary edge is lower than e_{max}

  where e_{max} : value selected by the user

- Local error evaluation



d: distance to plane



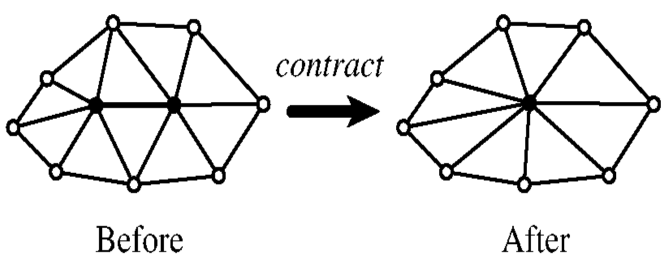
d: distance to edge

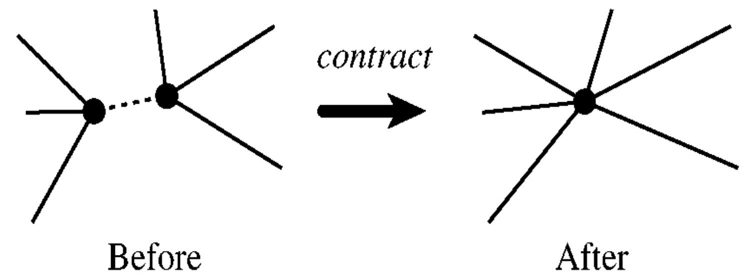
Mesh Decimation - Evaluation

- good efficiency (speed & reduction rate)
- simple implementation and use
- works on very large meshes
- preserves topology; vertices are a subset of the original ones
- implemented in the Visualization Toolkit (VTK), public domain
- error is not bounded (local evaluation ➡ accumulation of error)

Decimation based on iterative contractions

- *Other techniques for mesh decimation:*
  - *edge collapse* to a vertex (half-edge and full-edge collapses)
  - *triangle collapse* to a vertex
  - *vertex-pair contraction* to a vertex



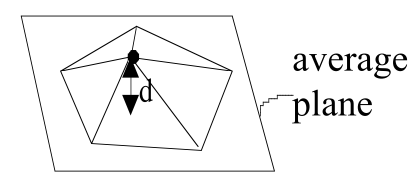


- *Topology preservation:*
  - Edge-collapse may implicitly alter the topology of the surface (e.g., by closing holes)
  - Vertex-pair contraction is used to merge several connected components (it may create non-manifold situations)

Error evaluation

- *Local error evaluation:* the error associated with a mesh □' is evaluated with respect to the approximated mesh □" on which the local modification has been performed
- *Global error evaluation:* the error associated with a mesh □' is evaluated with respect to the mesh □* at full resolution

- Heuristics proposed for *local error evaluation*:
  - *approximate* evaluation [Schroeder 92]: distance from an approximating plane

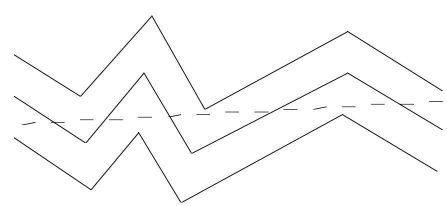


d: distance to plane

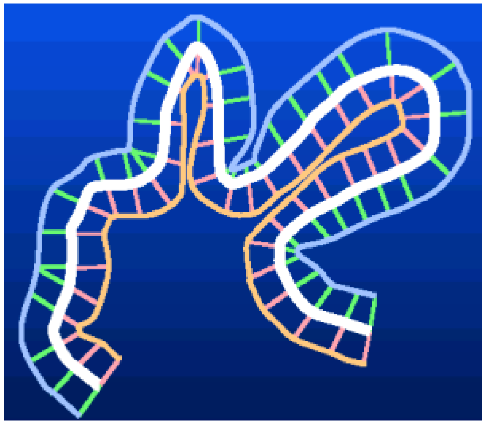
- Heuristics proposed for *global error evaluation:*
  - *vertex--to--simplified mesh* distance [Soucy96]: compute the distance of □' from the set of vertices V* of the original mesh

Simplification Envelopes [Cohen et al.'96]

- It does not provide an explicit evaluation of the accuracy, but ensures that the simplified mesh is within a target threshold distance form the input mesh
- Given the input mesh □
  - build two envelope meshes $□_{-}$ and $□_{+}$ at distance -□ and + □from □;
  - simplify □ (following a vertex decimation approach) by enforcing the decimation criterion: a candidate vertex may be removed *only if* the new triangle patch does not intersect either $□_{-}$ or $□_{+}$



Simplification Envelopes [Cohen et al.'96]



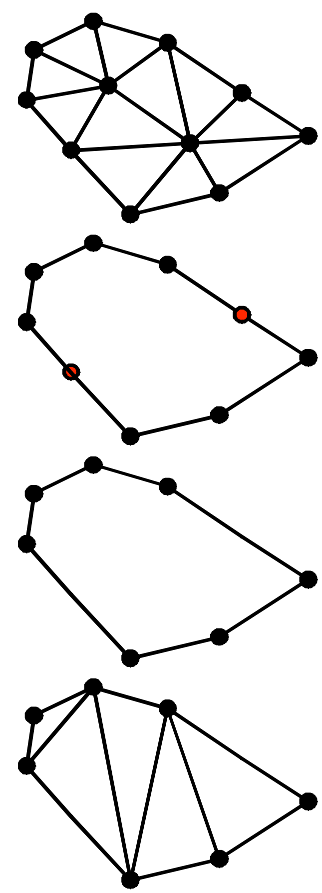
(drawing by A. Varshney)

- Envelopes do not self-intersect: thus, simplified mesh is *not self-intersecting*

- *Properties*
  - works for manifold surfaces
  - bounded approximation
  - construction of envelopes and intersection tests are expensive
  - vertices are a subset of the original data set
  - topology is preserved
  - available in public domain

Non-incremental techniques: Co-planar face merging

Superfaces [Kalvin, Taylor '96]



- The algorithm partitions the surface into connected disjoint regions based on a planarity criterion

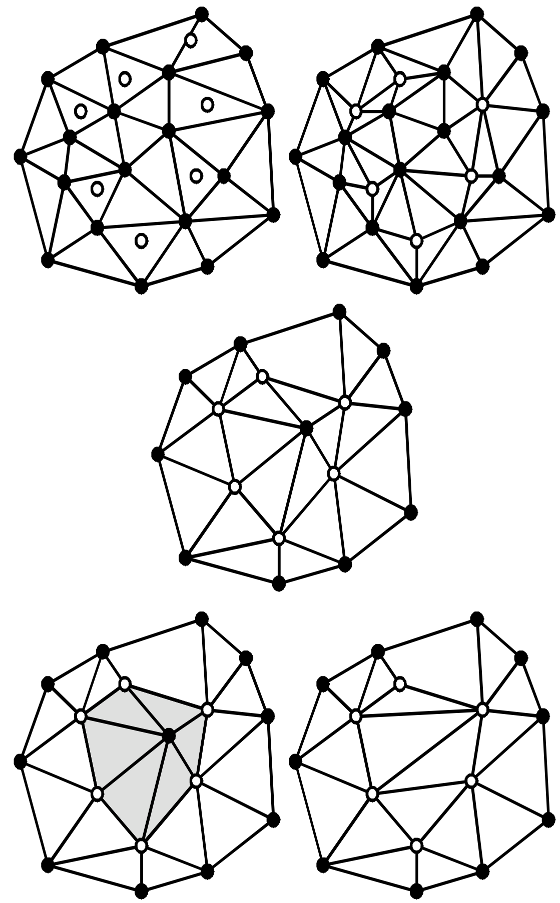
- Each region is replaced by a polygonal patch whose boundary is simplified, and the resulting region is re-triangulated

Properties:

- simple and efficient heuristics
- vertices are a subset of the original set
- geometric discontinuities are preserved (e.g. sharp edges)
- restricted to manifold surfaces
- topology is preserved

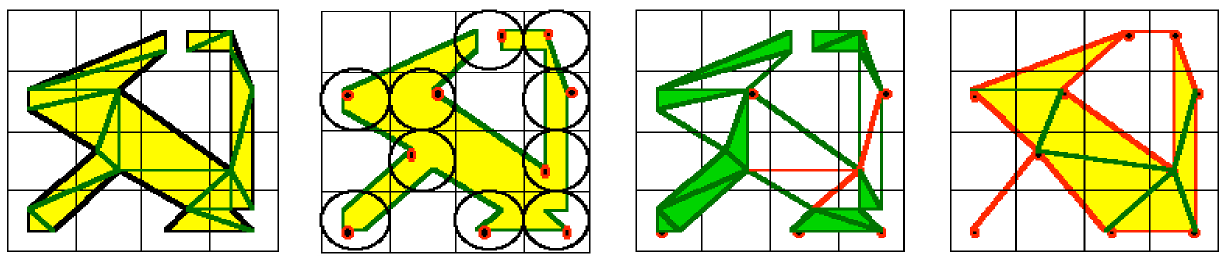
Non-incremental techniques: Re-tiling
[Turk `92]

- Distribute a new set of vertices on the original triangular mesh (points positioned using a relaxation technique to allow optimal surface curvature representation)
- Remove (part of) the original vertices
- Use local re-triangulation



Non-incremental techniques: vertex clustering [Rossignac, Borrel `93]

- detect and unify *clusters* of nearby vertices by superimposing a grid on the mesh
- all faces with two or three vertices in a cluster are removed



- *Properties:*
  - high efficiency
  - low quality approximations: the error is bounded by the grid cell size
  - it does not preserve topology
  - extended to perform simplification out-of-core (Lindstrom, 2000)

Survey papers on mesh simplification

- P.Heckbert and M.Garland, Survey of Polygonal Surface Simplification Algorithms, Course Notes SIGGRAPH1997.
  http://graphics.cs.uiuc.edu/~garland/papers/simp.pdf

- P. Cignoni, C. Montani, R. Scopigno, A Comparison of Mesh Simplification Algorithms, Computer and Graphics, 22(1), 1998
  http://vcg.iei.pi.cnr.it/bibliography.html

- M. Garland, Multiresolution modeling: Survey and Future Opportunities, Eurographics'99, State-Of-The-ART Report.
  http://graphics.cs.uiuc.edu/~garland/papers/STAR99.pdf

- P. Lindstrom , G.Turk, Evaluation of Memory-less Simplification, IEEE Transactions on Visualization and Computer Graphics, 5(2), 1999
  http://www.gvu.gatech.edu/people/peter.lindstrom/papers/tvcg99/tvcg99.pdf

- D.Luebke, Developer's Survey of Polygonal Simplification Algorithms, IEEE Computer Graphics &Applications (May 2001)
  http://www.cs.virginia.edu/~luebke/publications/pdf/cg+a.2001.pdf