Progressive Meshes

Xin (Shane) Li

Progressive Meshes

- Motivations
- For Surfaces: Progressive Triangular Meshes
  - Connectivity
  - Geometry
- For Solids: Progressive Tetrahedral Meshes

Complex Meshes

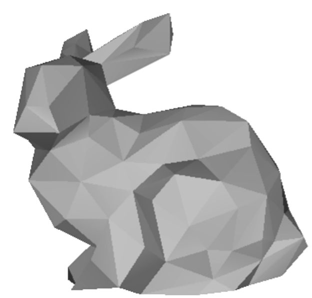
<u>Challenges:</u>
- Expensive to store, transmit, render, and edit

Level of Detail

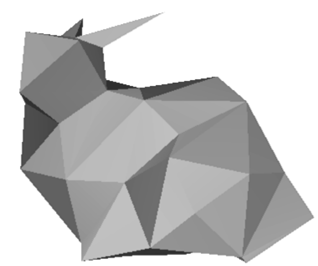
- Decreasing the complexity of a 3D object representation
  - as it moves away from the viewer
  - or based on other metrics (object importance, eye-space position…)
- Applied on geometry, texture, material…

69,451 faces

2,502 faces



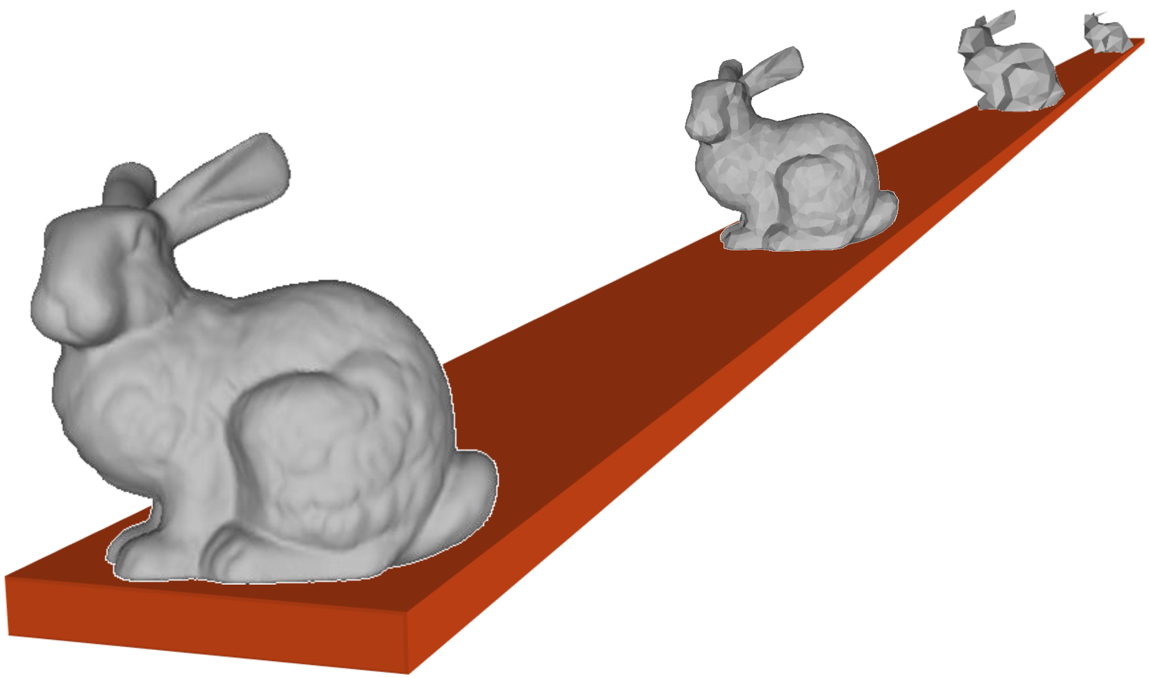
251 polys



76 polys

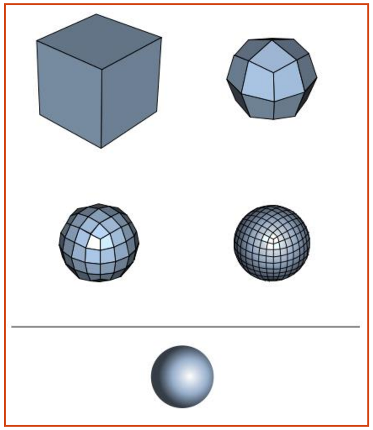
Level of Detail

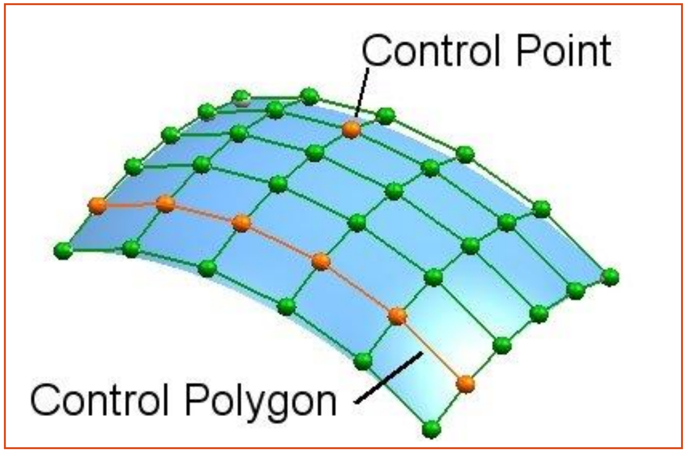
- Distant objects use coarser LODs:

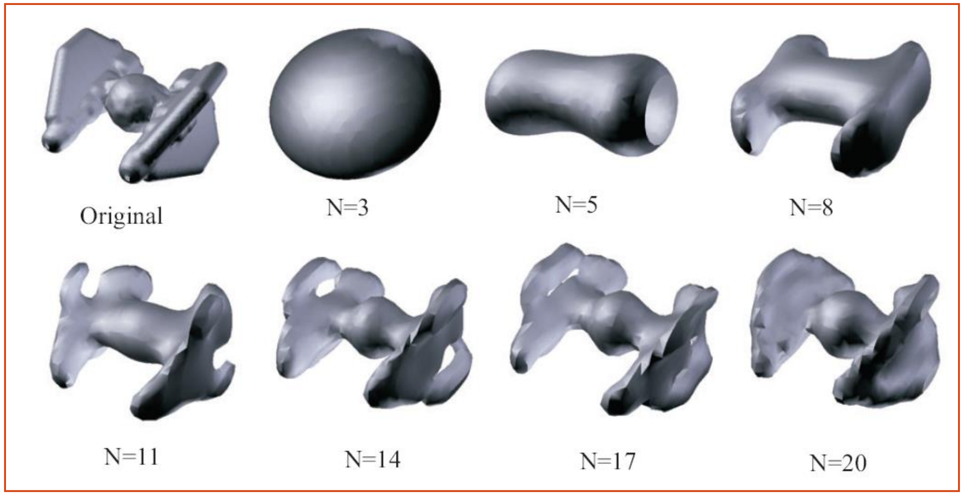


Multiresolutional Modeling, Processing and Analysis

- Subdivision Surface, splines, wavelets…
- Image pyramid…
- Quad/oct tree, BSP tree…





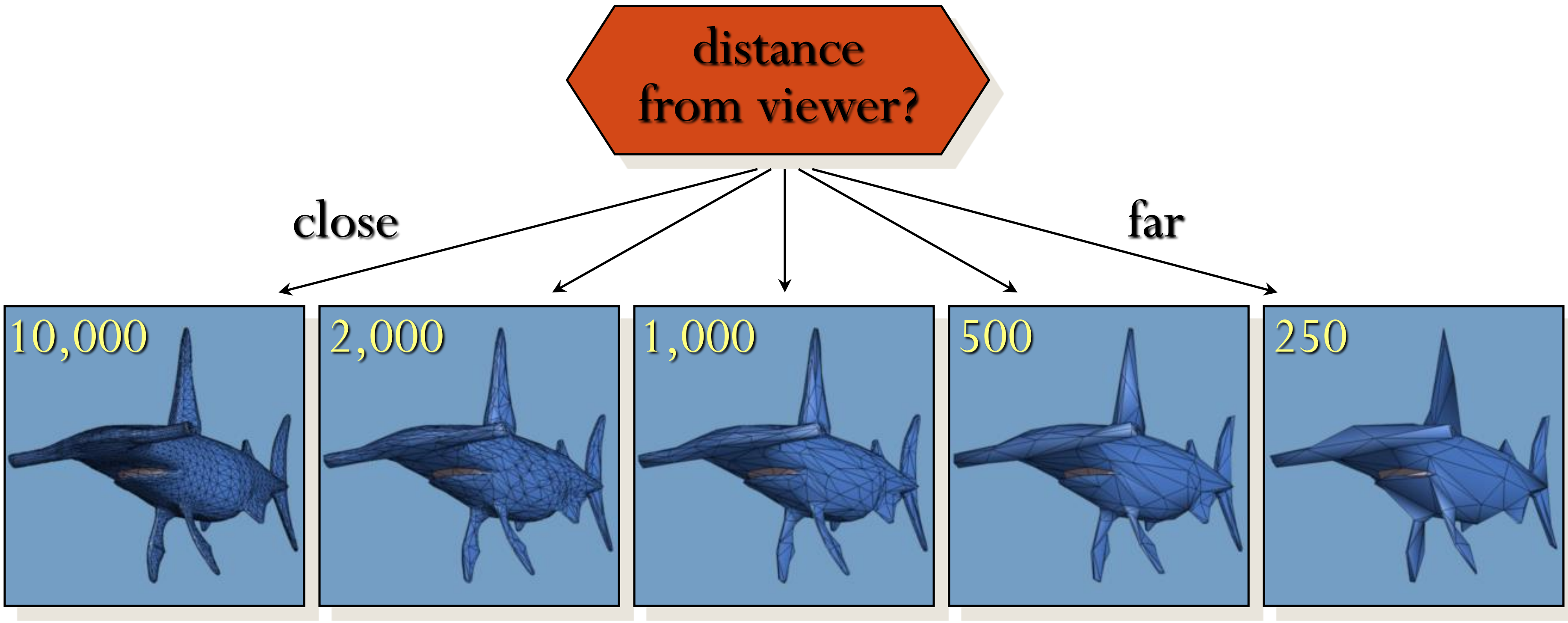


Motivations

- Applications of multiresolution techniques : Compression, Progressive transmission and display, Level-of-detail Control, Multiresolution editing…

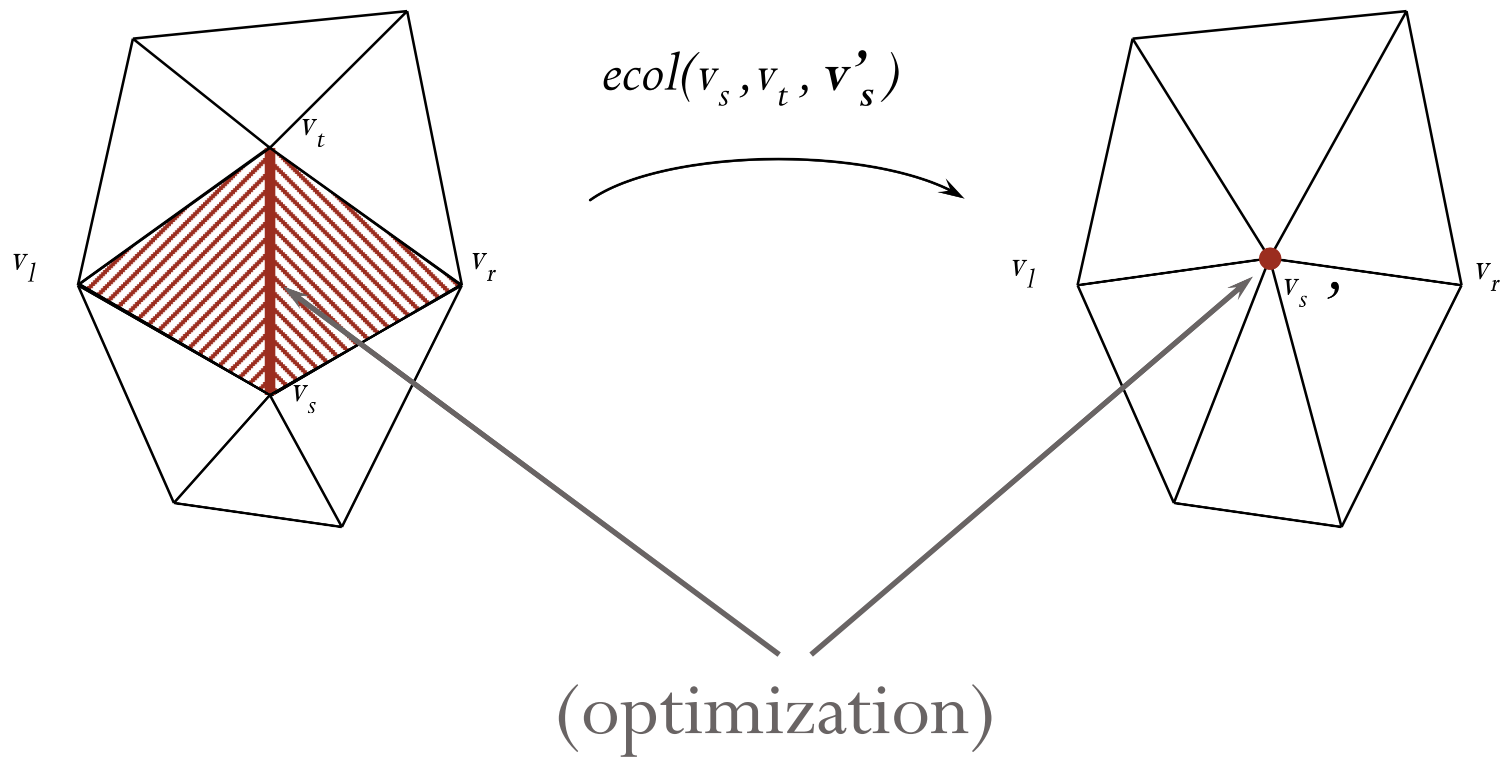
- A mesh simplification procedure for general input meshes
  - Preserve various properties (colors, normals, …)
  - Lossless
  - Continuous-resolution
  - Efficient (time and space)
  - Directly applicable for Progressive transmission

Level-of-detail (LOD)



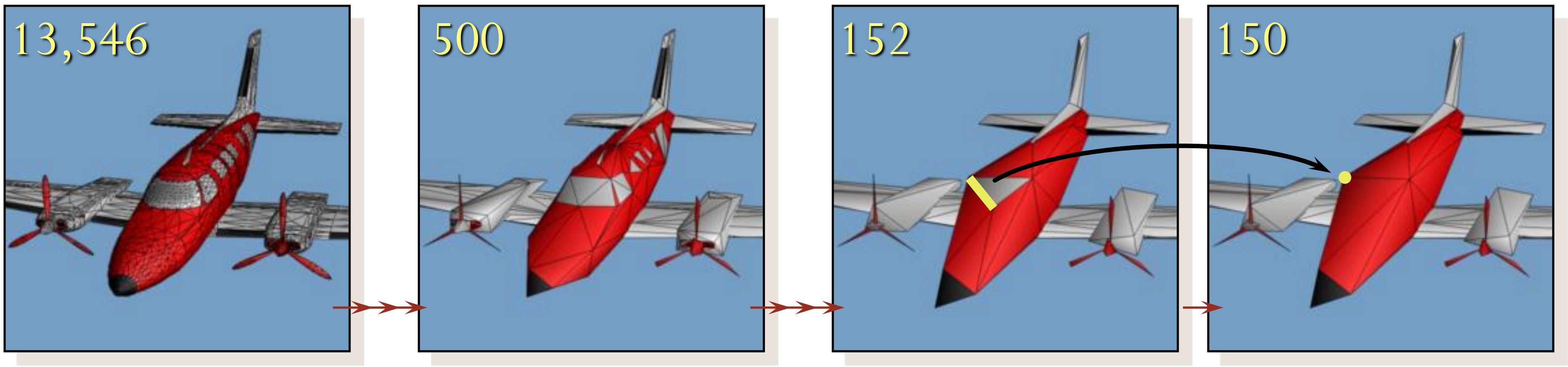
Mesh simplification procedure

- Idea: apply sequence of edge collapses:



Connectivity change can be easily implemented using Half-Edge Data Structure!

Simplification process

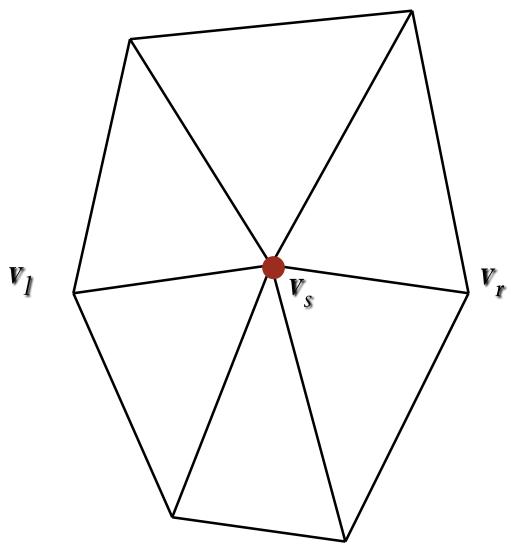


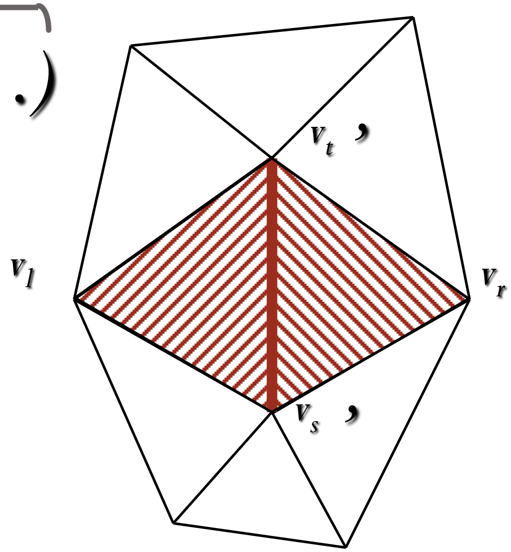
$$\hat{M} = M^n \xrightarrow{ecol_{n-1}} \rightarrow\rightarrow\rightarrow M^{175} \rightarrow\rightarrow\rightarrow \xrightarrow{ecol_i} M^1 \xrightarrow{ecol_0} M^0$$

Inverse Direction: Reconstruction

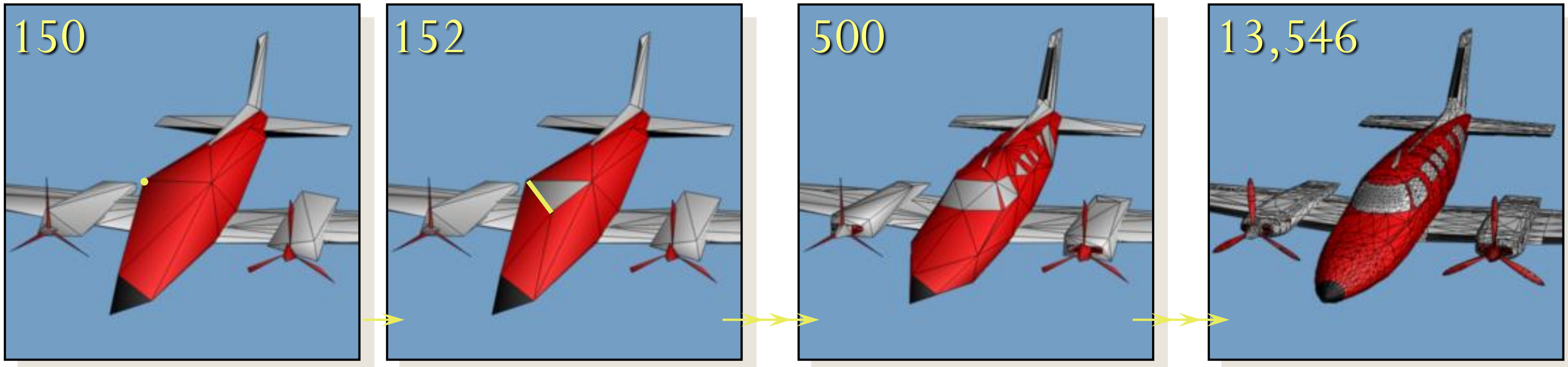
Vertex split transformation:

$$vspl(v_s, v_l, v_r, \underbrace{v'_s, v'_t, \ldots}_{attributes})$$





Reconstruction process



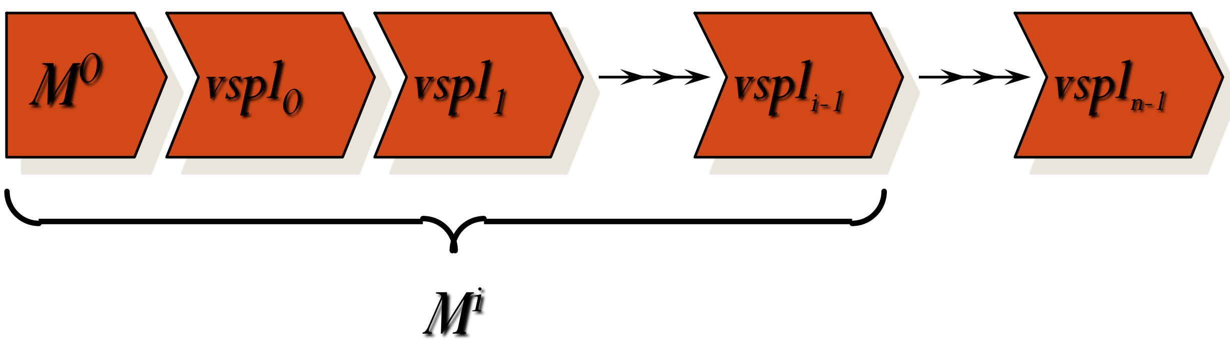
$M^0 \longrightarrow M^1 \longrightarrow \cdots \longrightarrow M^{175} \longrightarrow \cdots \longrightarrow M^n = \hat{M}$

$vspl_0$ … $vspl_i$ … $vspl_{n-1}$

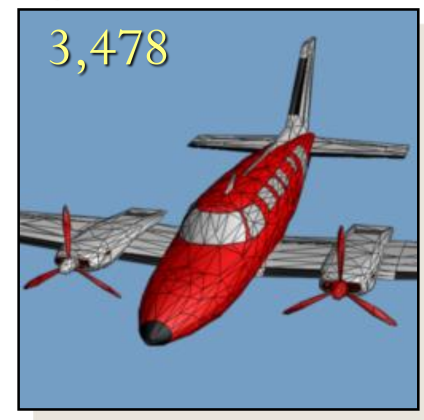
progressive mesh (PM) representation

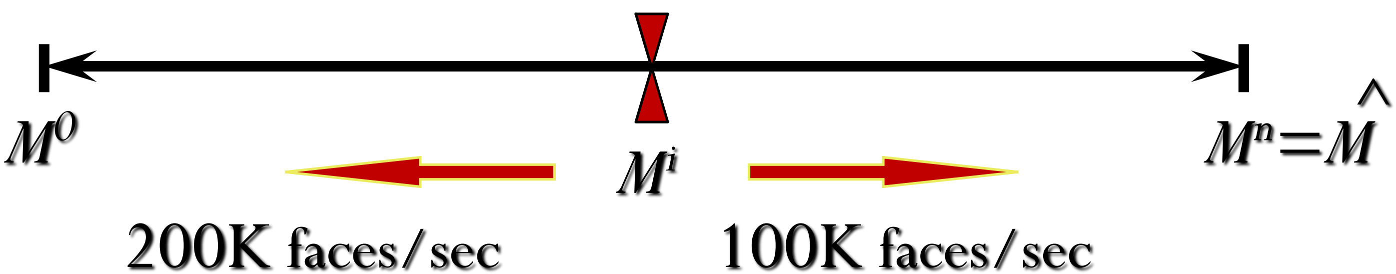
Continuous-resolution LOD

From PM, extract M^i of any desired complexity.



3,478 faces?

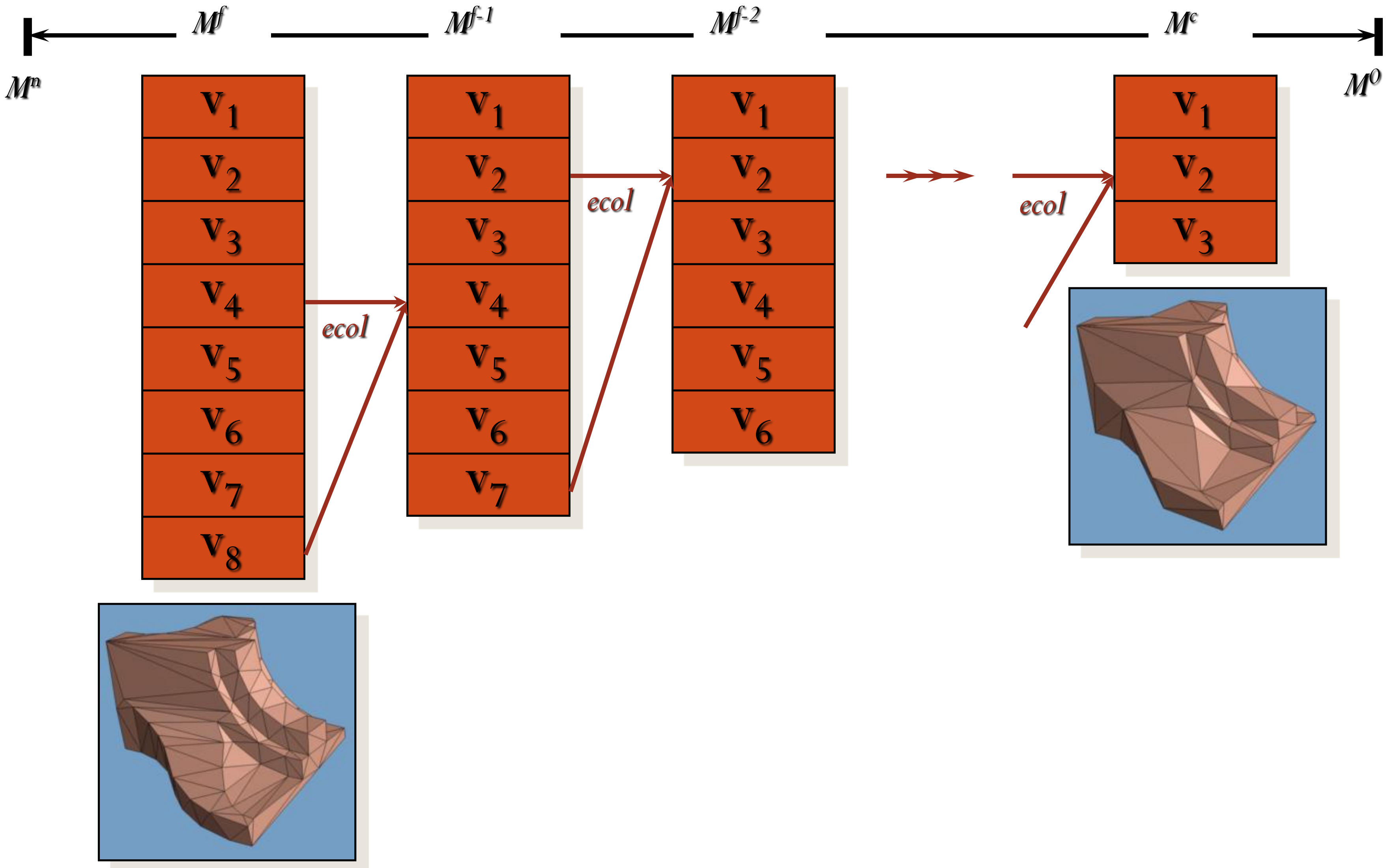




Video

(166 MHz Pentium)

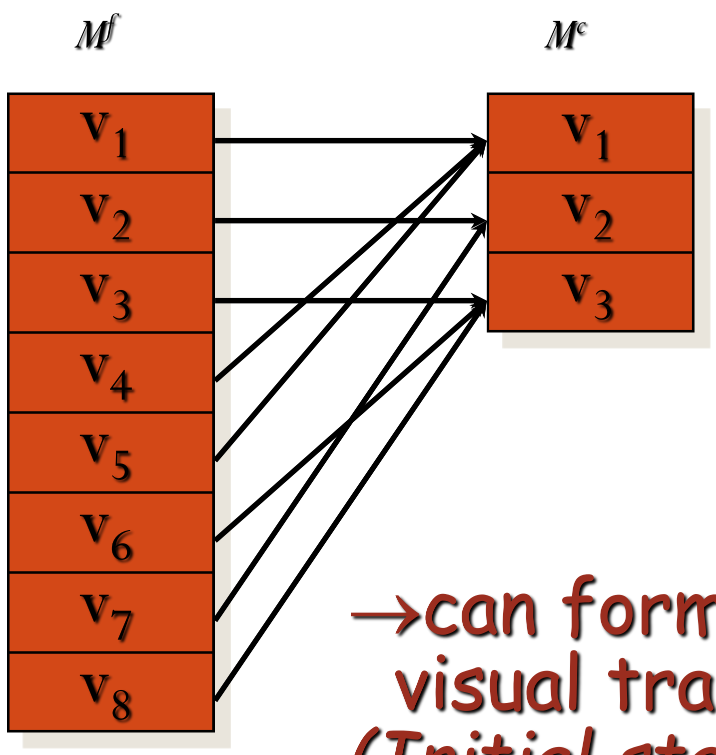
Property: Vertex correspondence

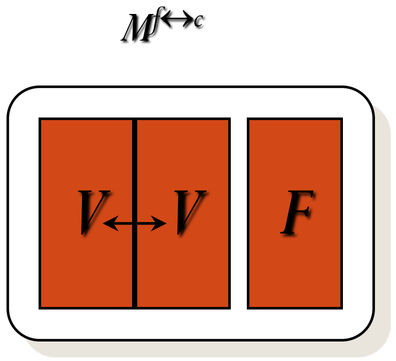


Application: Smooth transitions

Correspondence is a surjection:

Video





→can form a smooth visual transition: *geomorph*

(Initial status: topologically M^f, geometrically $M^{f\text{->}c}$)

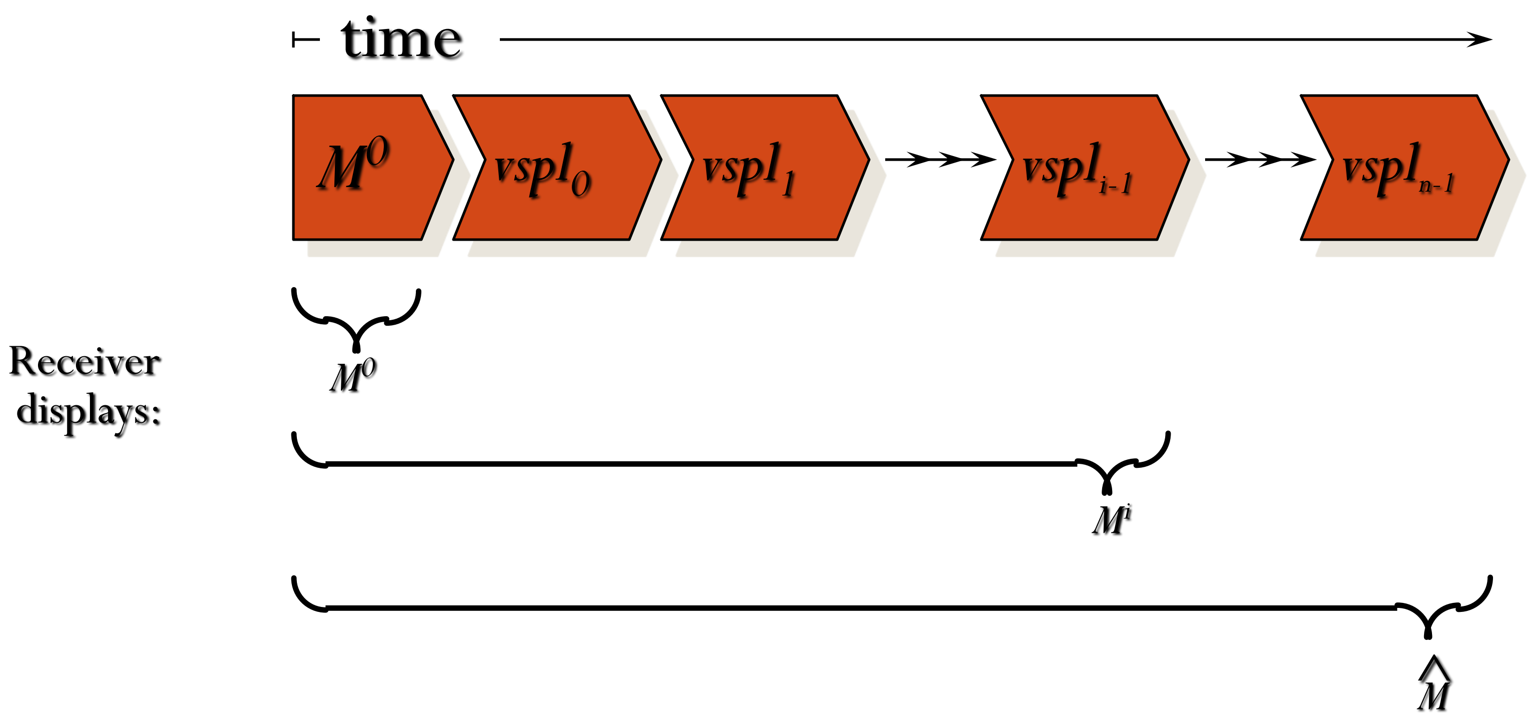
Morphing by Linear Interpolation

- Source mesh M1={V1, ..., Vn}
- Target mesh M2={U1, ..., Un}
- The linear interpolation:

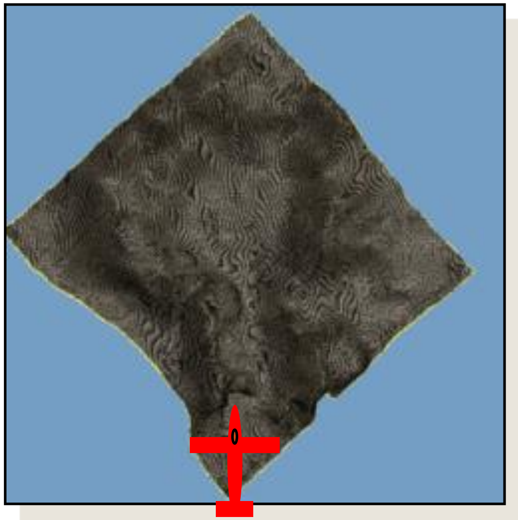
$$M(t) = \{V1*(1-t)+U1*t, \ldots, Vn*(1-t)+Un*t\}$$

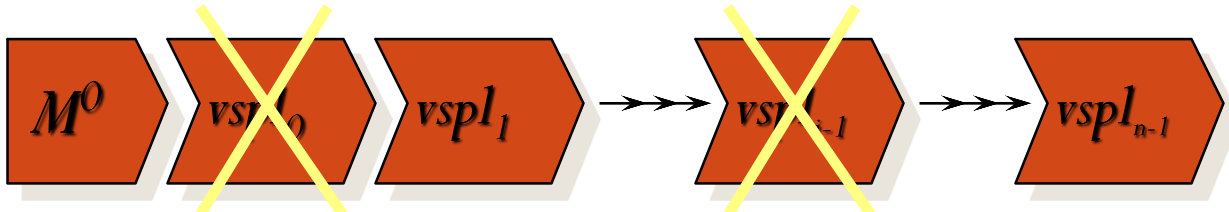
Application: Progressive transmission

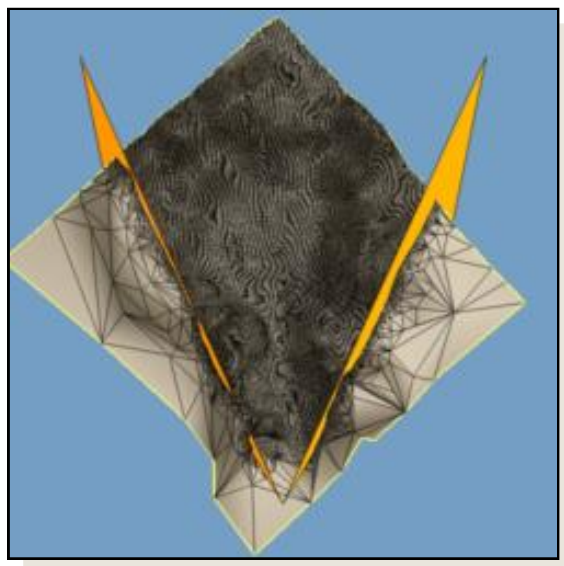
Transmit records progressively:

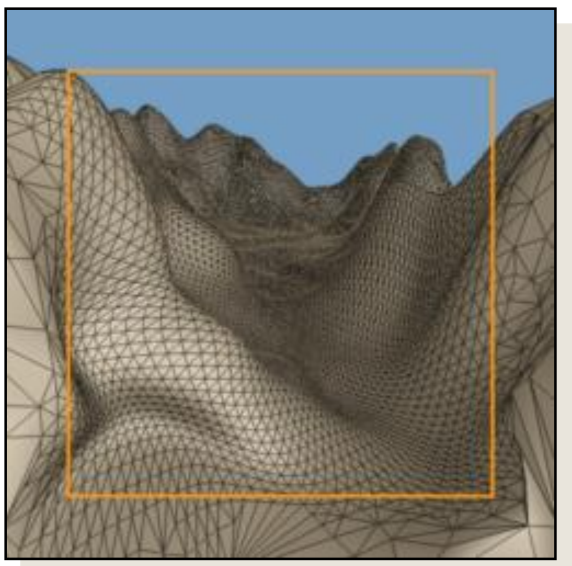


Application: Selective refinement







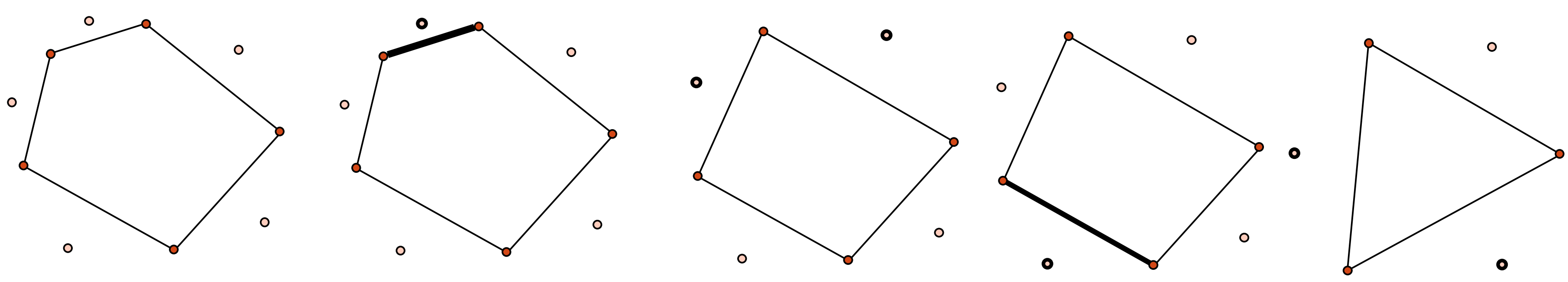


(e.g. view frustum)

Where to place the new vertex?

- A simple strategy: pick shortest edge + use the midpoint
- *An locally optimal solution:
  - To measure the shape error due to the edge collapse: QEM ["Surface Simplification Using Quadric Error Metrics," Garland and Heckbert 97]
    - After an edge collapse, the error of a new vertex is the sum of squared distances to originally associated planes
    - Iteratively collapse the edge (u,v) with the lowest E(u,v) into the minimizer P(u,v), update the minimizers and QEMs of the new edges (suitable data structure: heap)

An analogous example in 2D:



A Quadratic Energy to Measure the Error

- Squared distance of point p to plane q:

$$p = (x, y, z, 1)^T, \quad q = (a, b, c, d)^T$$

$$dist(q, p)^2 = (q^T p)^2 = p^T (q q^T) p =: p^T Q_q p$$

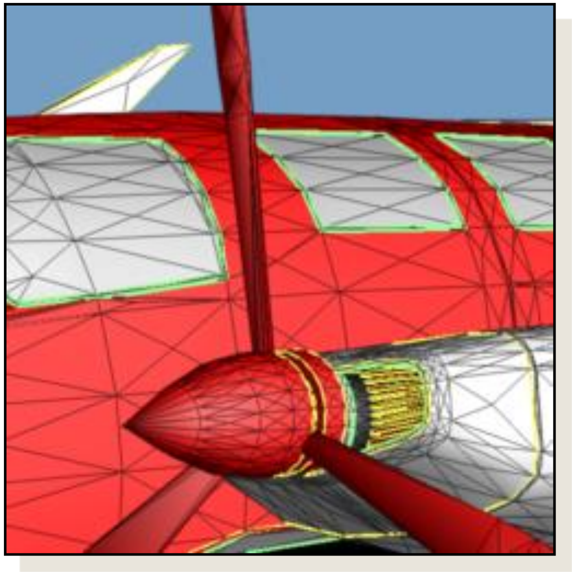
$$Q_q = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

- • Sum distances to planes of neighboring triangles:

$$\sum_i dist(q_i, p)^2 = \sum_i p^T Q_{q_i} p = p^T \left(\sum_i Q_{q_i} \right) p =: p^T Q_p p$$

More Criteria to pick edges

- An energy to describe the deviation of *appearance* :
  - geometric shape
  - scalar fields (e.g. color)
  - Feature (discontinuity) curves
  - …



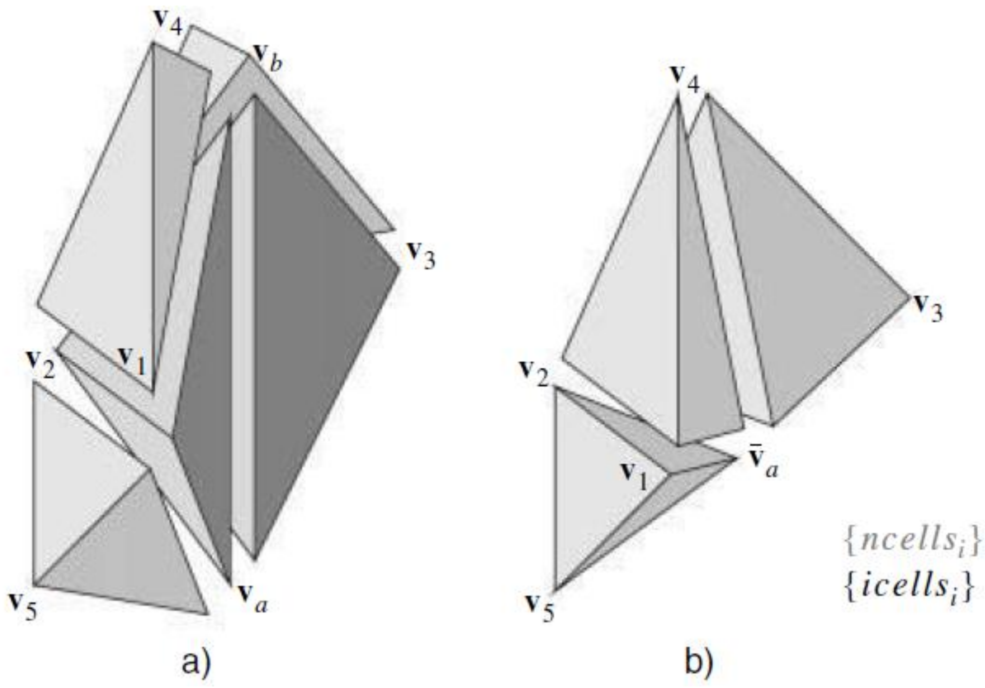
- In general, greedy strategy to always collapse edge resulting in smallest ΔE

Summary

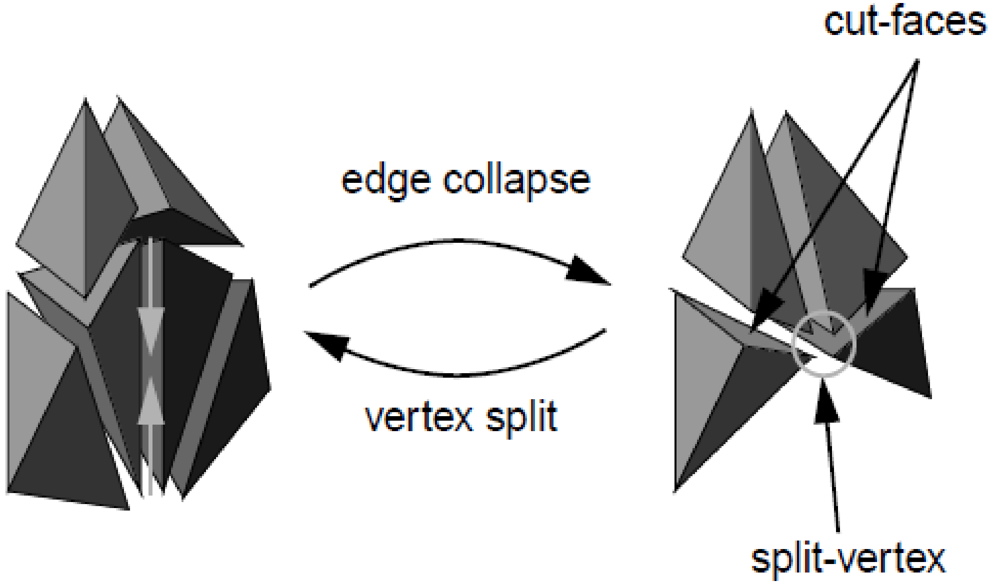
- With the half-edge data structure, can you implement this idea using the simplest strategy (pick shortest edge, use mid-point)
- Now consider the generalization of PM to 3D…

Progressive Tetrahedral Meshes

Edge Collapse

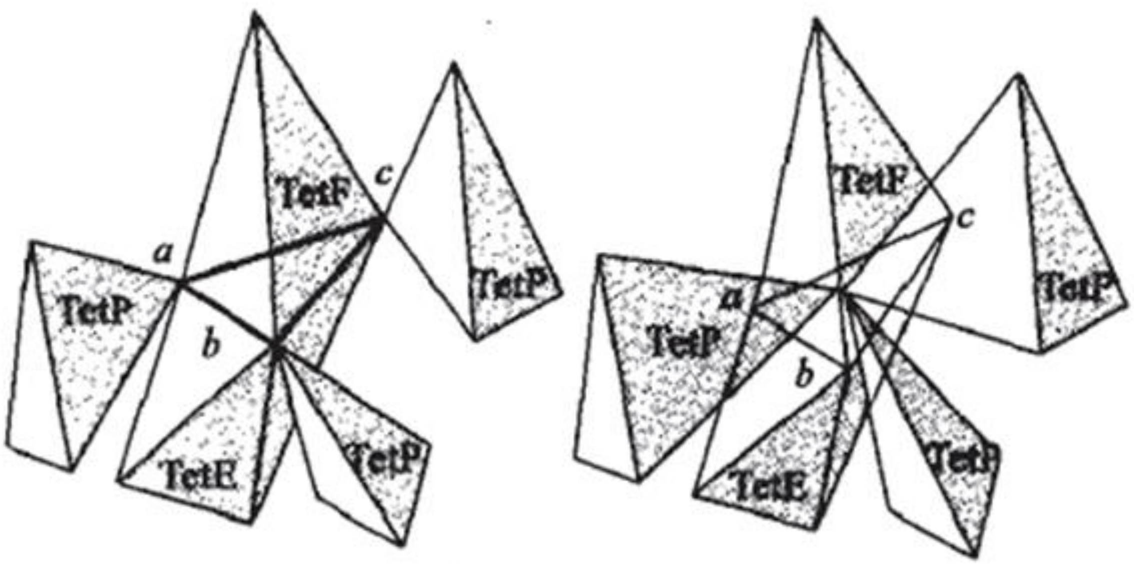


Progressive Tetrahedral Meshes



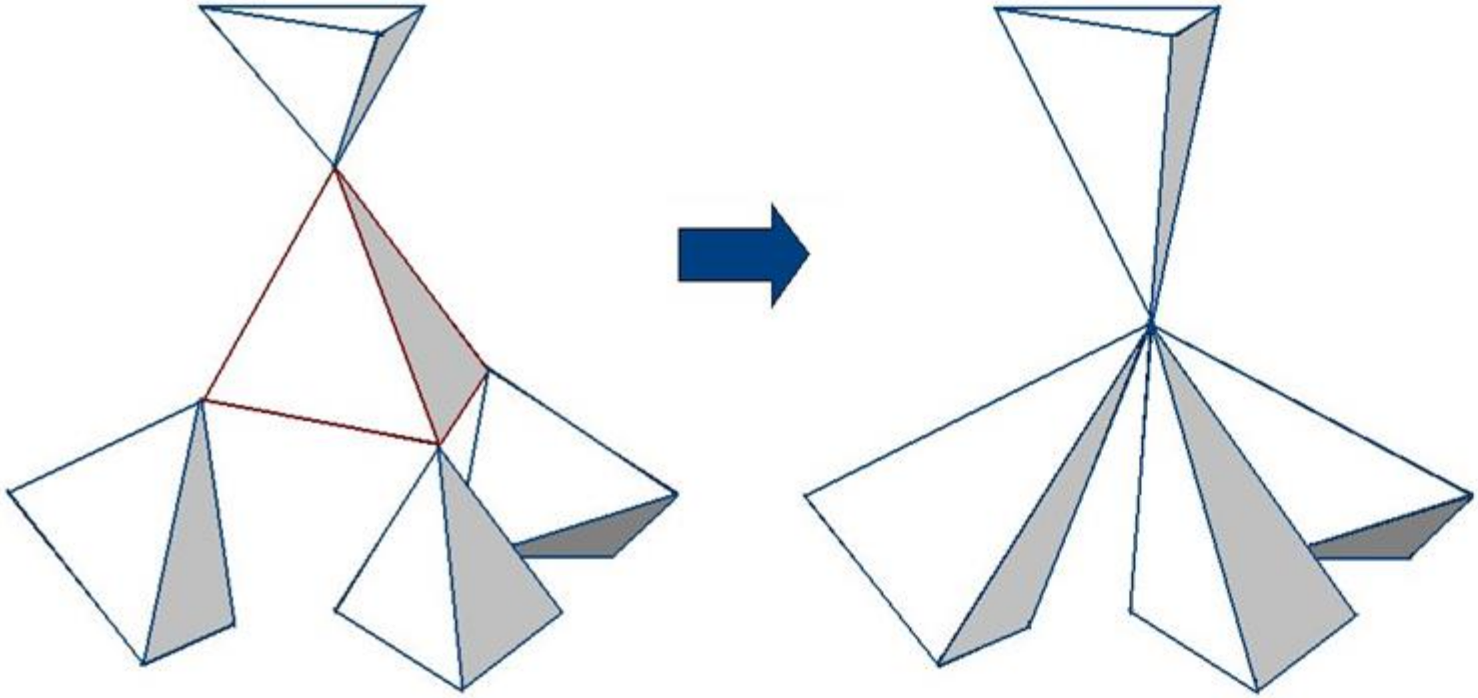
Progressive Tetrahedral Meshes

Is "Edge Collapse" the only way?



Progressive Tetrahedral Meshes

Is "Edge Collapse" the only way?



An application in Morphing

Course Project Candidate:

Inter-surface mapping and morphing