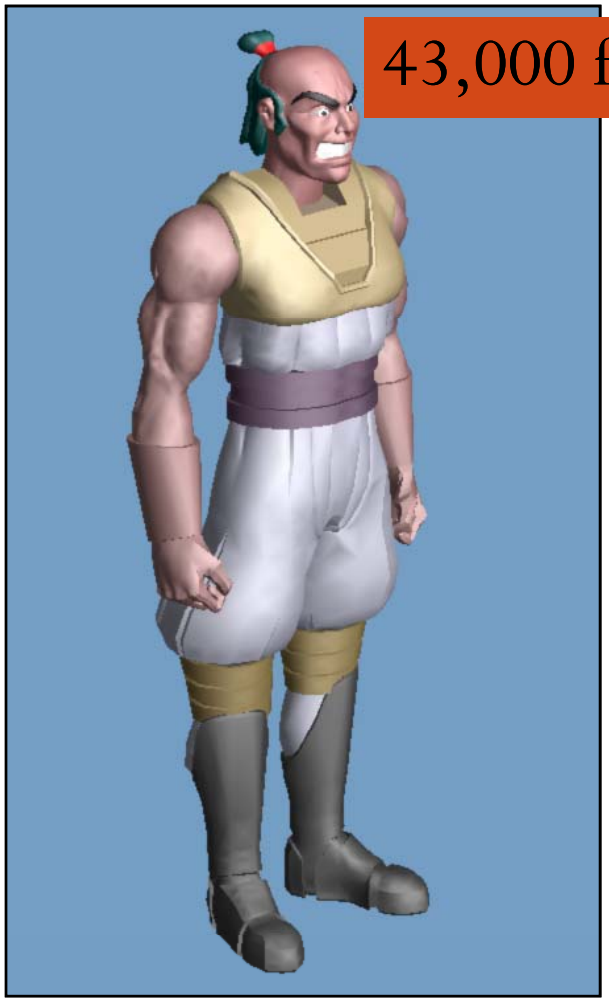
# Progressive Meshes

Xin (Shane) Li
March, 2010

# Progressive Meshes

- Motivations
- Progressive Triangular Meshes
  - Connectivity
  - Geometry
- Progressive Tetrahedral Meshes (Progressive Simplicial Complex)

# Complex Meshes



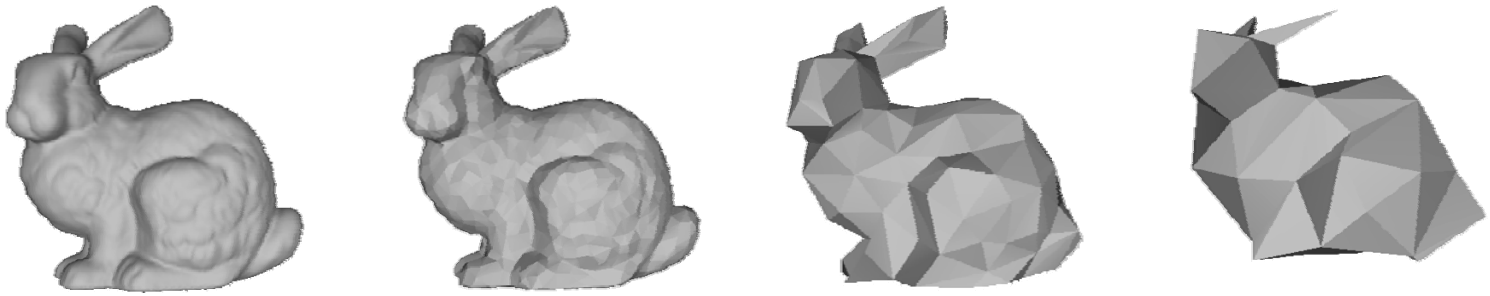


Challenges:

- 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…

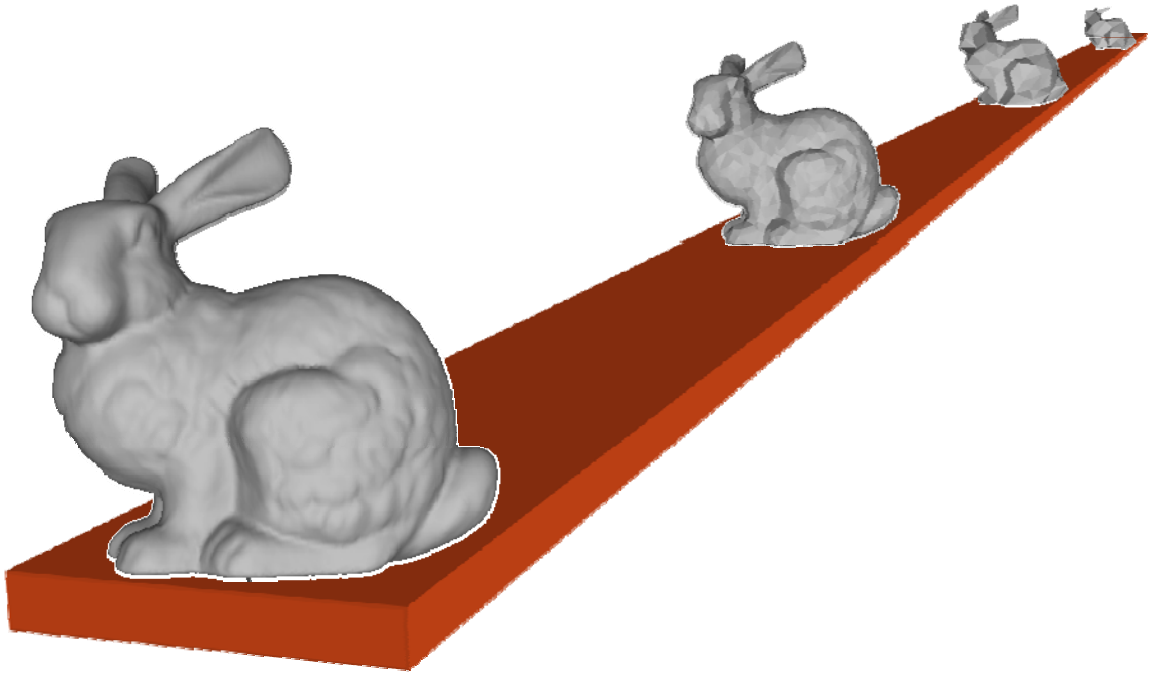


69,451 polys 2,502 polys 251 polys 76 polys

Courtesy Stanford 3D Scanning Repository

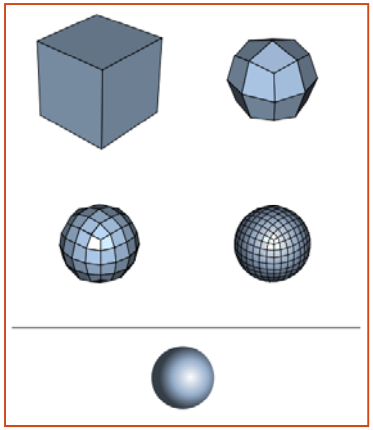
# Level of Detail

- Distant objects use coarser LODs:

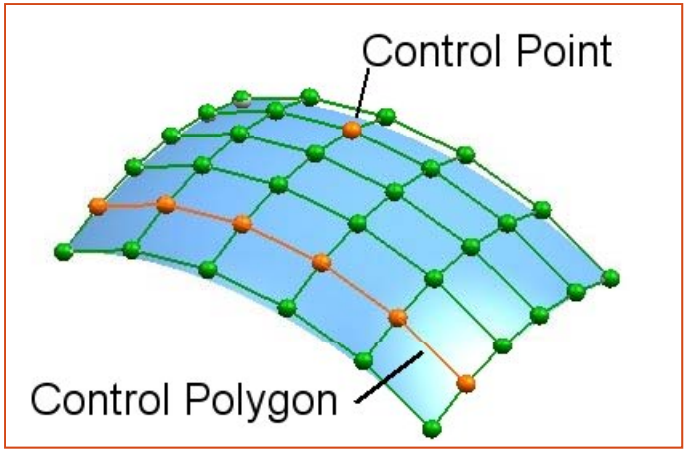


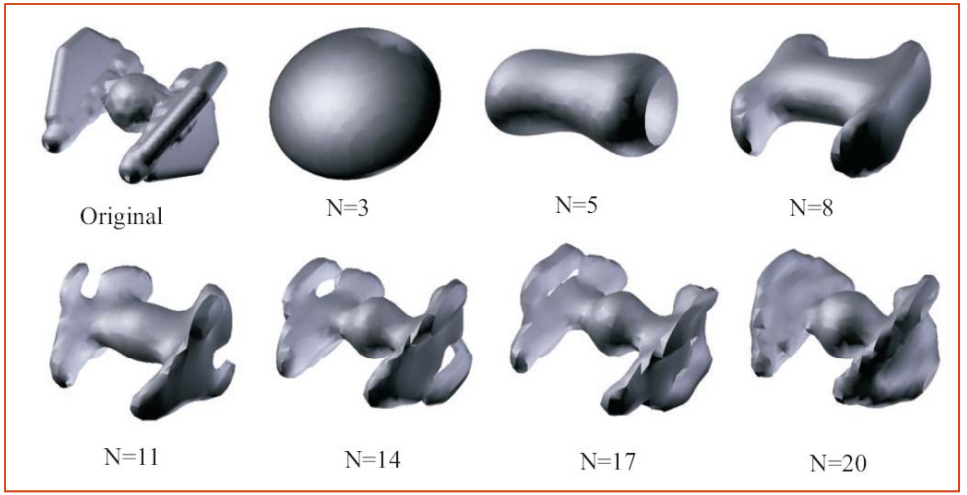
# Multiresolutional Modeling, Processing and Analysis

A webpage about Multiresolutional modeling by Michael Garland:
http://www.cs.cmu.edu/afs/cs/user/garland/www/multires/index.html



- Subdivision Surface
- Spline
- Wavelet
- …



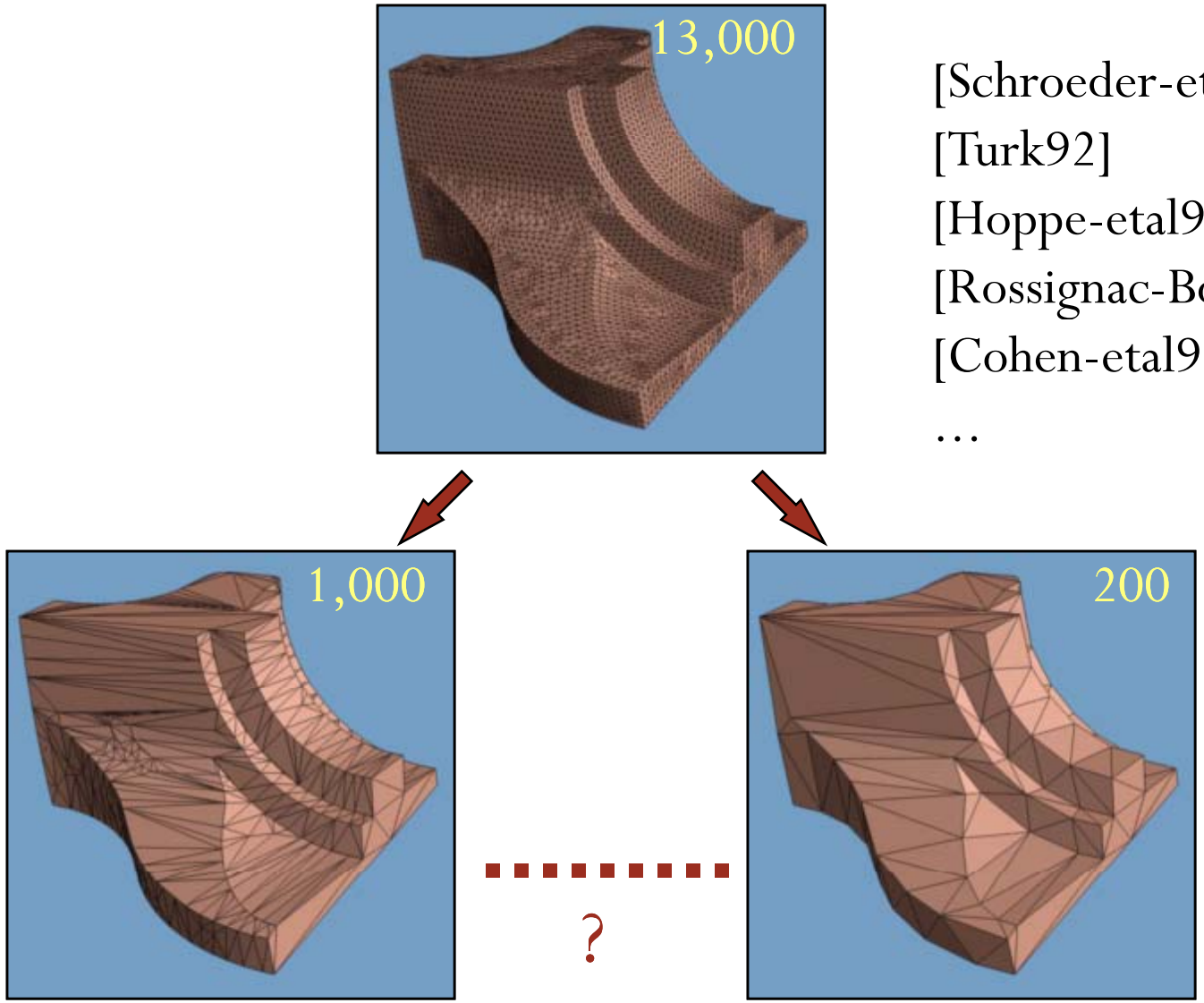



# Motivations

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

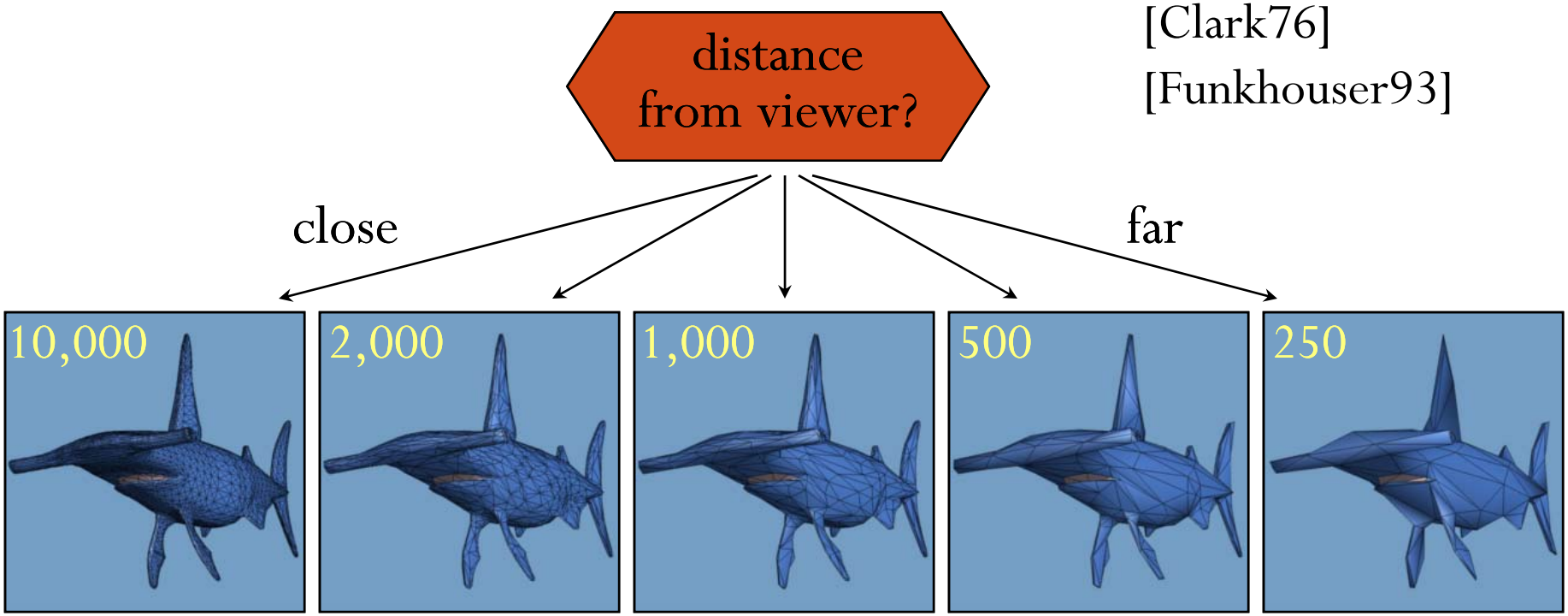
- A mesh simplification procedure for general input meshes
  - Preserve various properties (colors, normals, …)
  - Lossless
  - Continuous-resolution
  - Efficient (time and space)
  - Progressive transmission

# Mesh Simplification




[Schroeder-etal92]
[Turk92]
[Hoppe-etal93]
[Rossignac-Borrel93]
[Cohen-etal96]
…

# Level-of-detail (LOD)



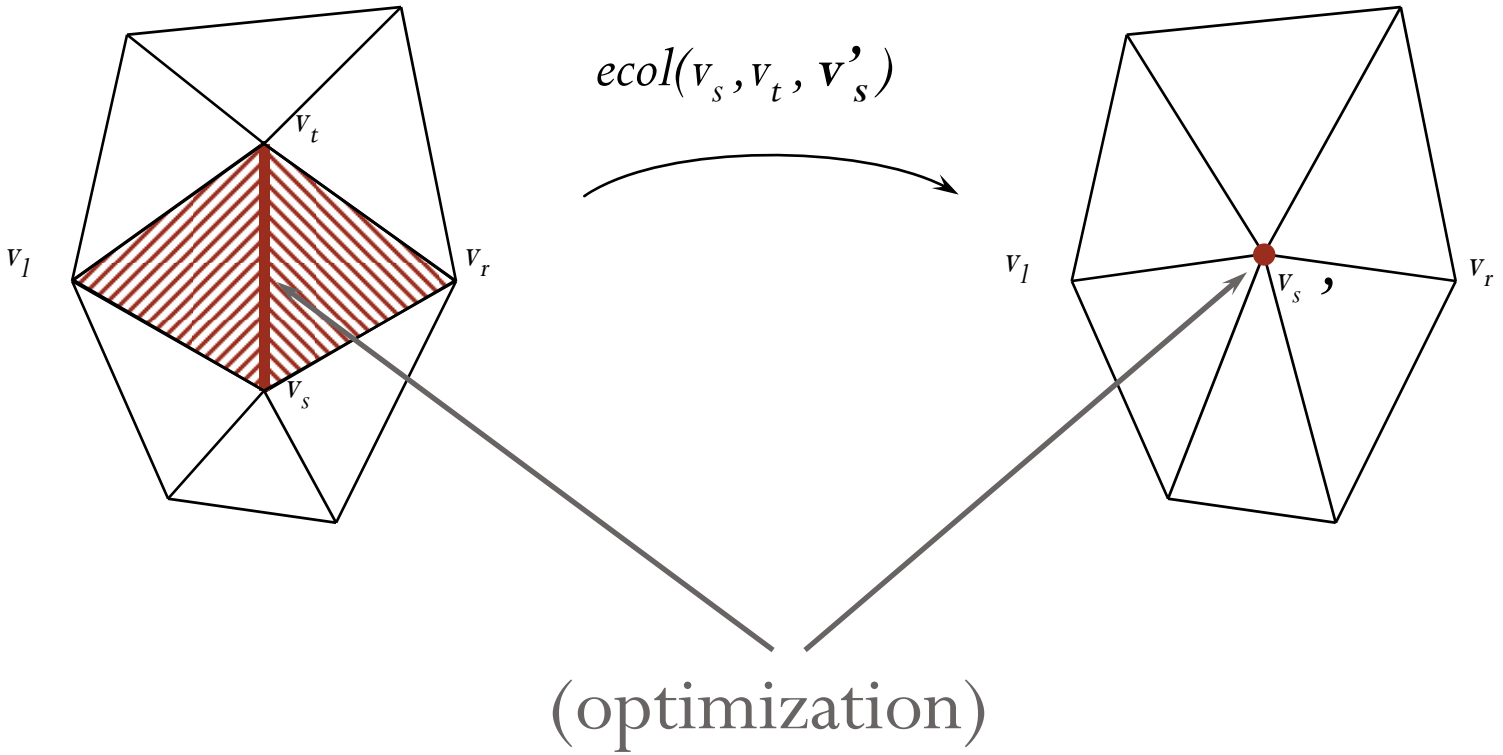

[Clark76]
[Funkhouser93]

Concern: transitions may "pop"

→ would like smooth LOD

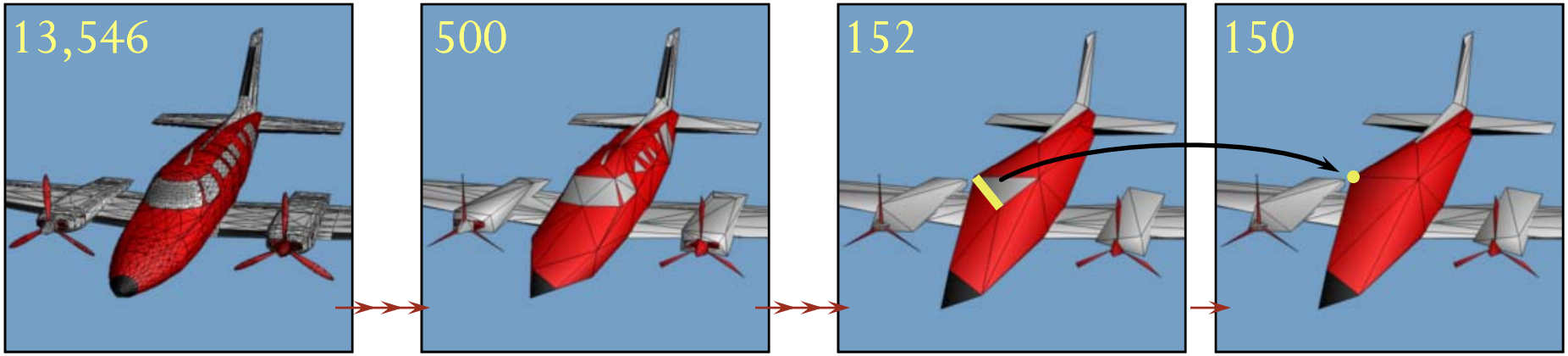
# Mesh simplification procedure

- Idea: apply sequence of edge collapses:




Can be easily implemented using Half-Edge Data Structure!

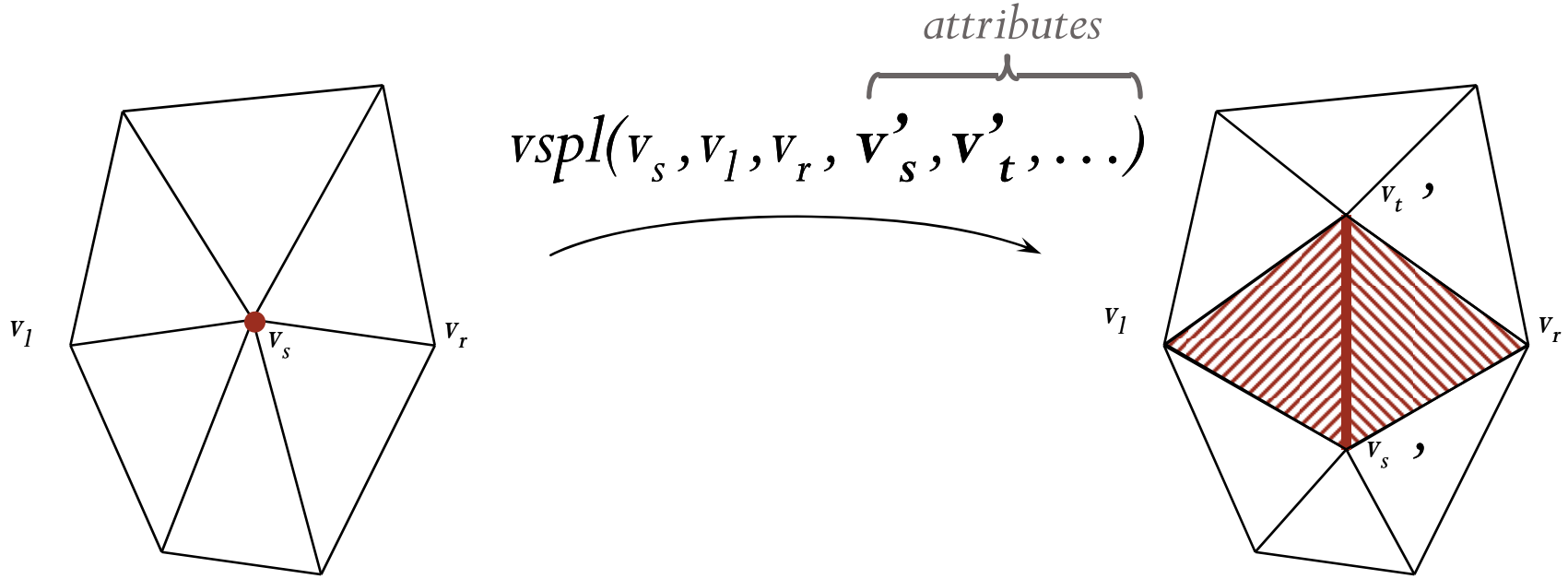
# Simplification process



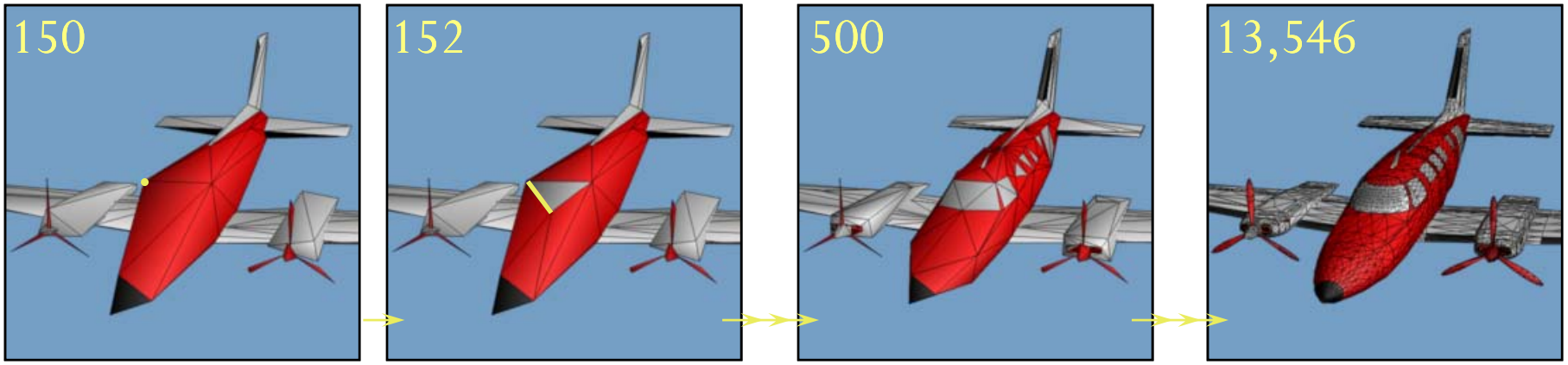

$\hat{M} = M^n \xrightarrow{ecol_{n-1}} \cdots \rightarrow M^{175} \cdots \rightarrow \xrightarrow{ecol_i} M^1 \xrightarrow{ecol_0} M^0$

# Invertible

Vertex split transformation:



# Reconstruction process



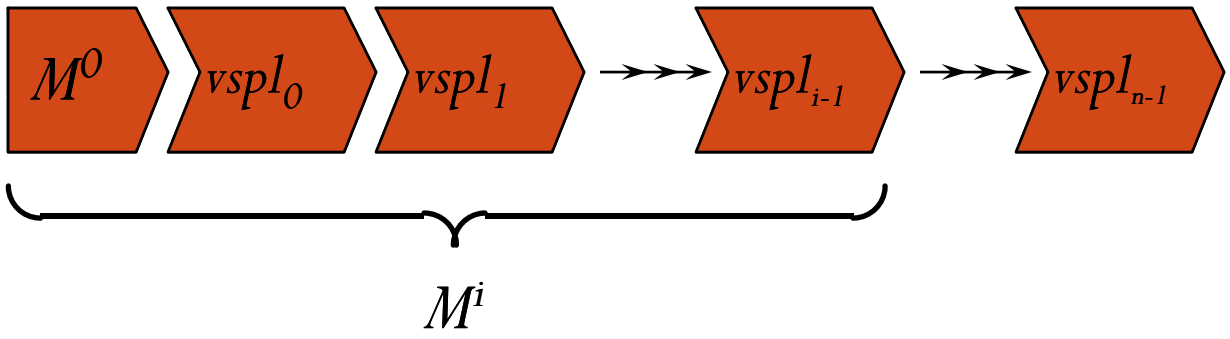

$M^0 \longrightarrow M^1 \longrightarrow \cdots \longrightarrow M^{175} \longrightarrow \cdots \longrightarrow M^n = \hat{M}$

$vspl_0 \quad \ldots \quad vspl_i \quad \ldots \quad vspl_{n-1}$

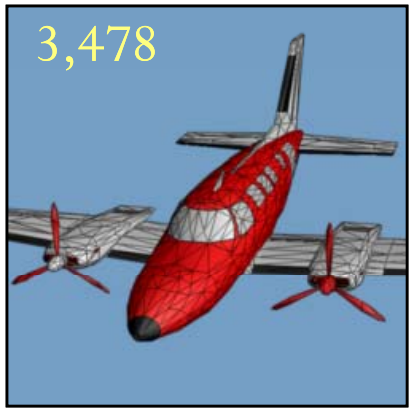
*progressive mesh (PM)* representation

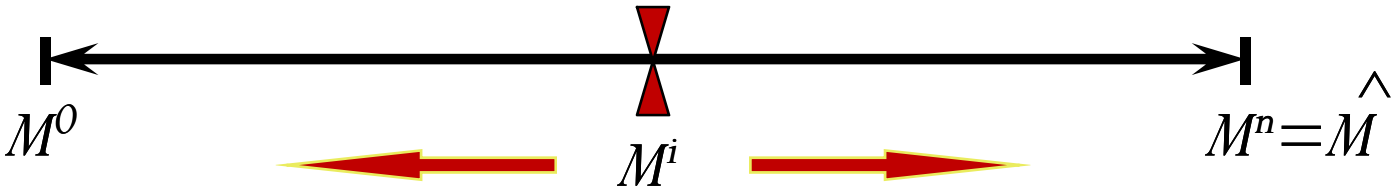
# Continuous-resolution LOD

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




3,478 faces?



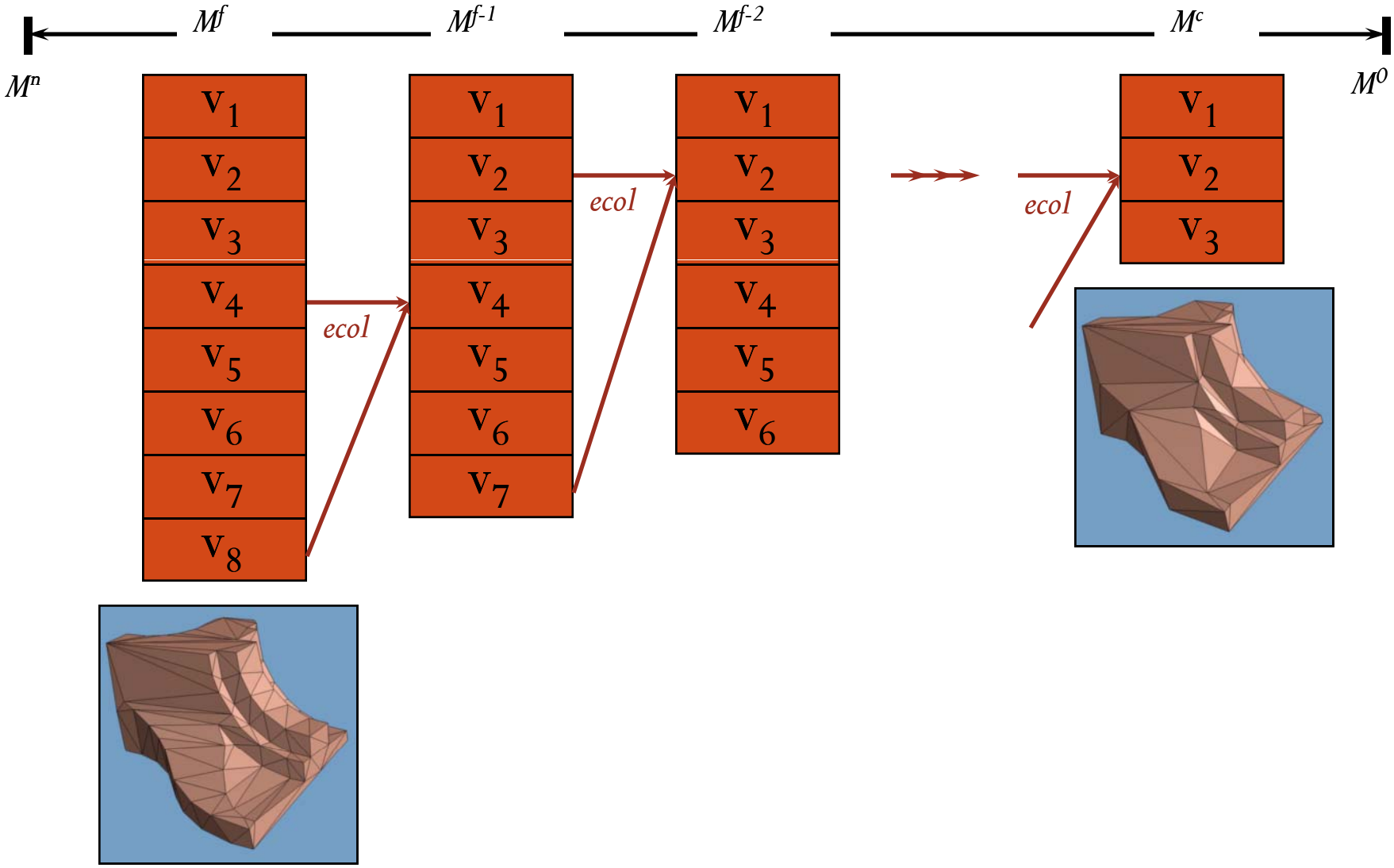

200K faces/sec!

100K faces/sec!

(166 MHz Pentium)

Video

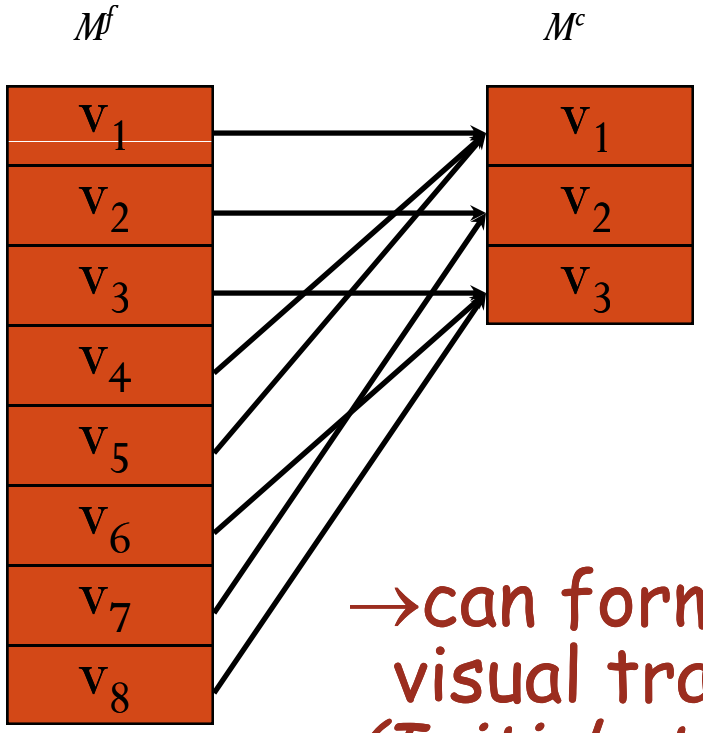
# Property: Vertex correspondence

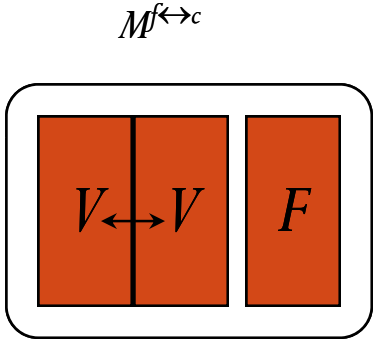


# Application: Smooth transitions

Correspondence is a surjection:

Video

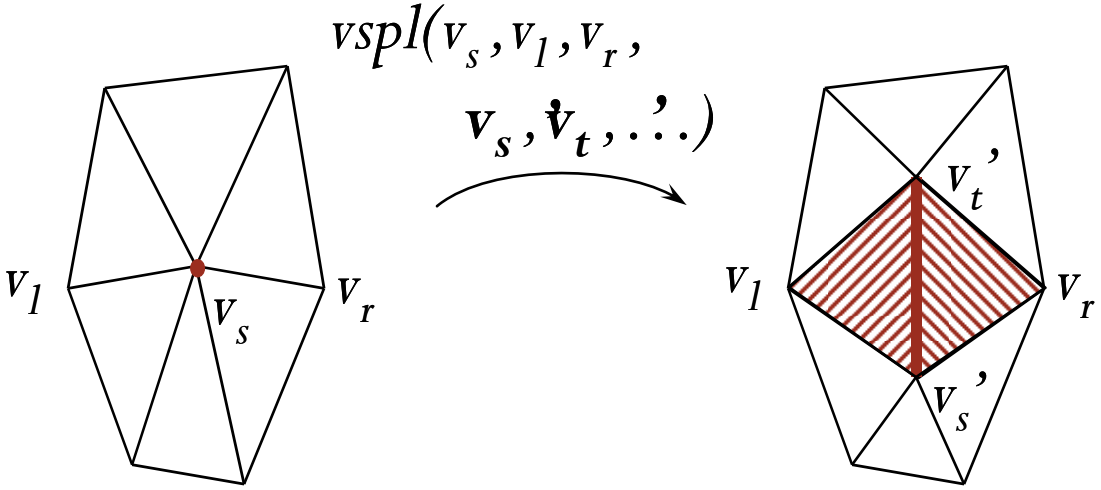



→can form a smooth visual transition: *geomorph*

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

# Application: Mesh compression




Record deltas:

- $v_t' - v_s$
- $v_s' - v_s$
- ...

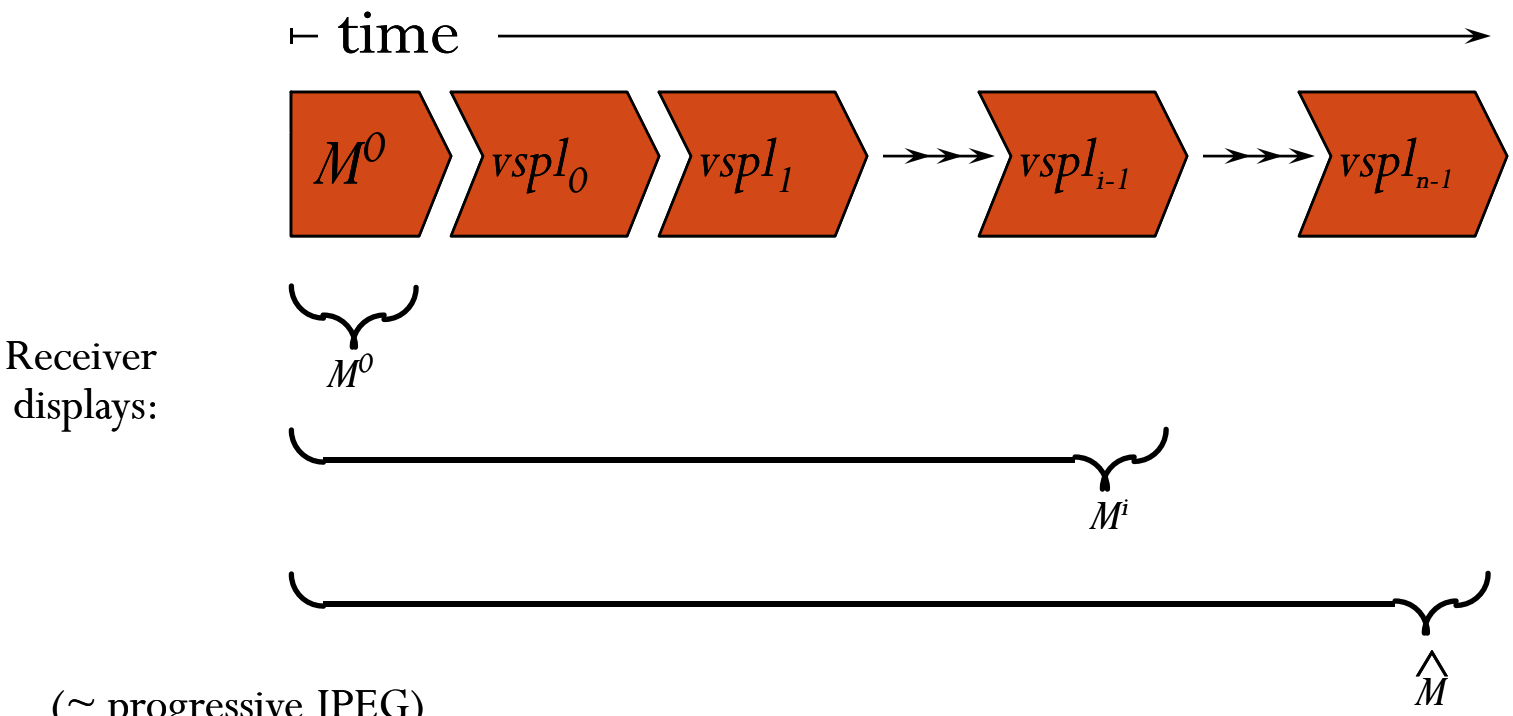
## Encoding of *vspl* records:

- connectivity: ~ good triangle strips
- attributes: excellent delta-encoding

[Deering95]

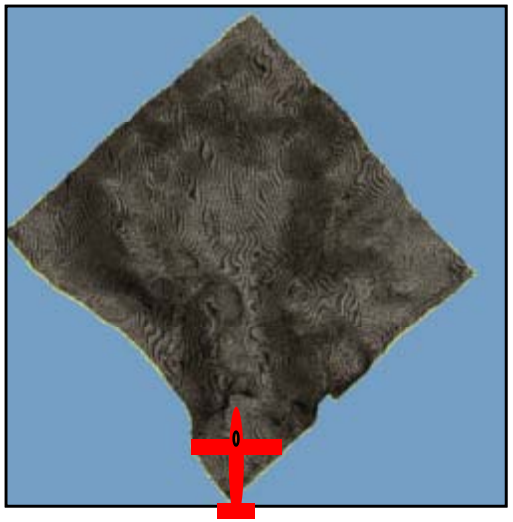
# Application: Progressive transmission

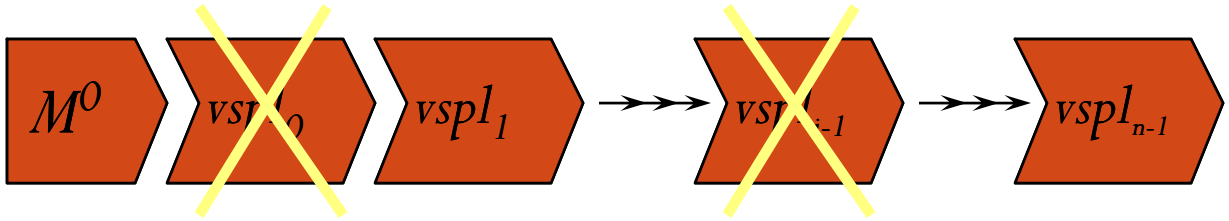
Transmit records progressively:

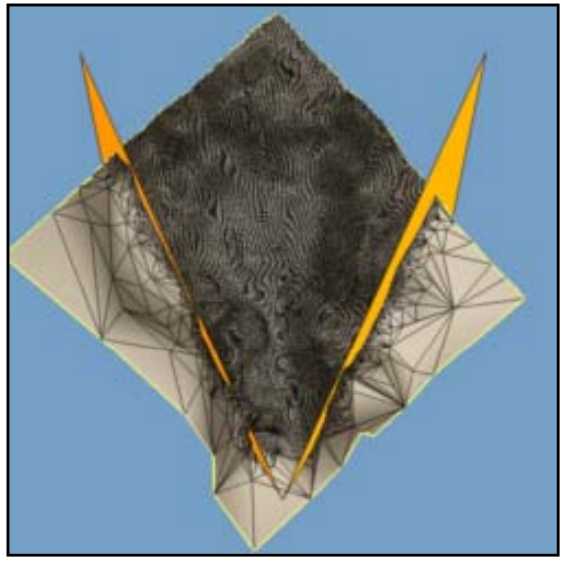


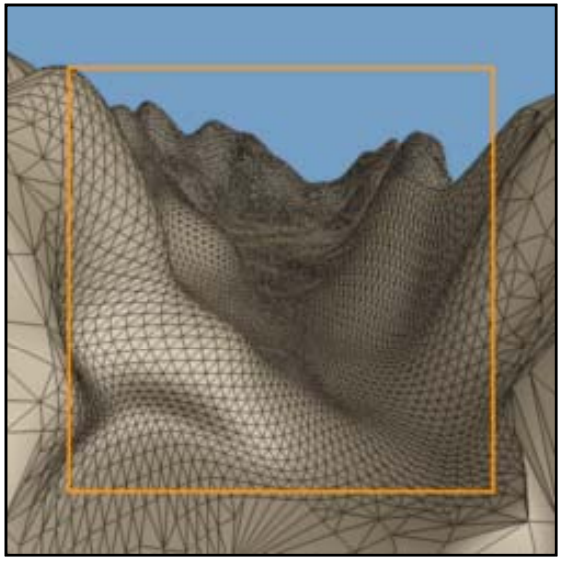

(~ progressive JPEG)

# Application: Selective refinement





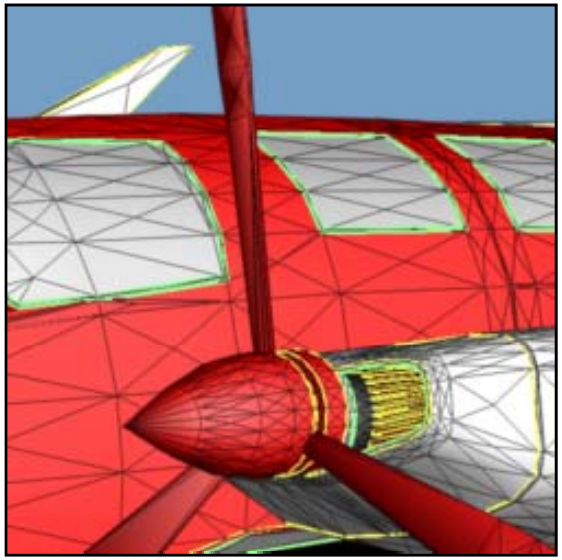





(e.g. view frustum)

# How to select edge collapses?

- Preserve *appearance*:
  - geometric shape
  - scalar fields (e.g. color)
  - discontinuity curves



$$E = \sum_{\text{points}} \int (e_{shape} + e_{scalars}) dA + \sum_{\text{points}} \int (e_{disc}) dL$$

# Selecting edge collapses

- Greedy algorithm: always collapse edge resulting in smallest $\Delta E$

Simplification rates: ~ 30 faces/sec

[Hoppe Siggraph 96]

- off-line process
- could use simpler heuristics

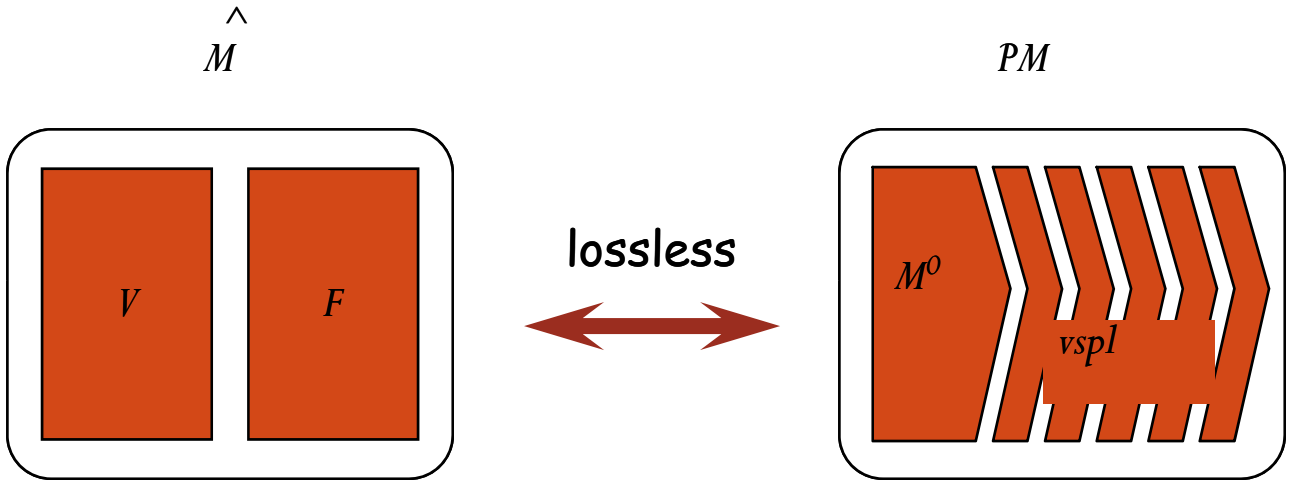
# * Fitting Energy

$$E_{dist}(K,V) = \sum_{i=1}^{n} d^2(\mathbf{x}_i, \phi_V(|K|))$$

$$\begin{aligned} E(K,V,B) &= \sum_{i=1}^{n} \|\mathbf{x}_i - \phi_V(\mathbf{b}_i)\|^2 + E_{spring}(K,V) \\ &= \sum_{i=1}^{n} \|\mathbf{x}_i - \phi_V(\mathbf{b}_i)\|^2 + \sum_{\{j,k\}\in K} \kappa \|\mathbf{v}_j - \mathbf{v}_k\|^2 \end{aligned}$$

- A non-linear optimization with probably not good "shape"
- Approximate the optimization: Global → Local

# Summary




- single resolution

- continuous-resolution
- smooth LOD
- space-efficient
- progressive

# Videos

# Summary

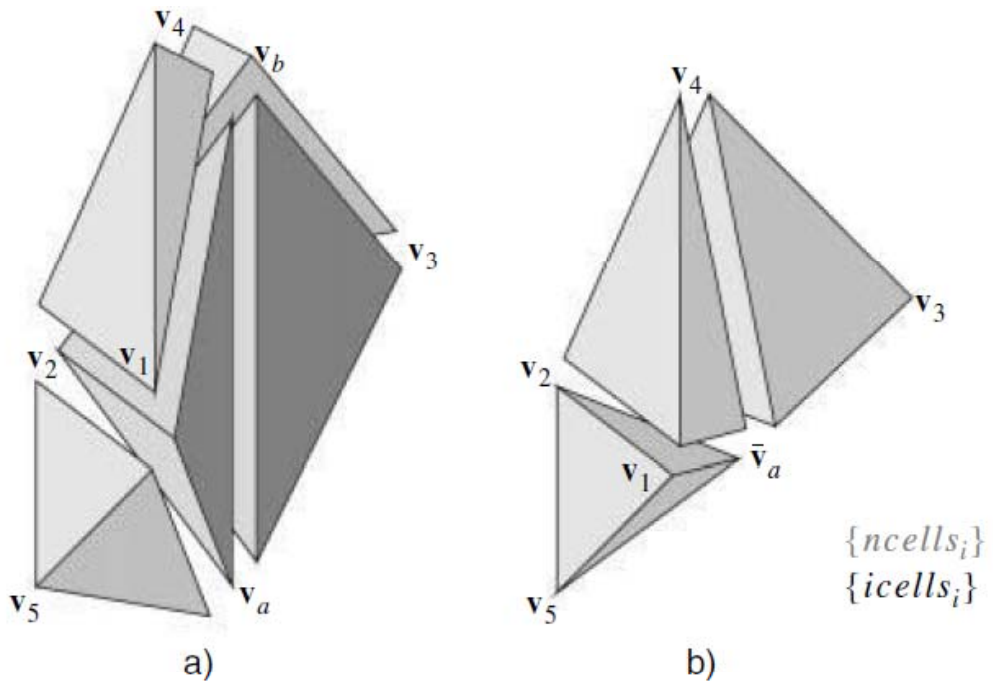
- Three issues that deserve more consideration:
  1. Correctness Detection
  2. Collapsing Edge Selection
  3. (New) Vertex Position
     1) Ideally: given n vertices → best approximation
     2) Practically: local optimization

# Summary

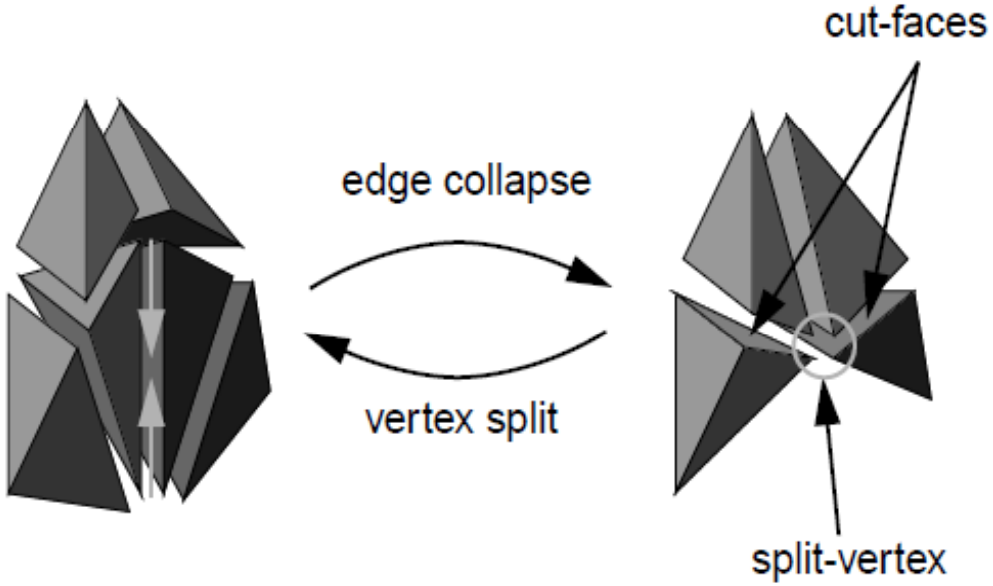
- Bottom line:
  - You got the concept and idea
  - And with the half-edge data structure, you can make this whole thing work
    - [Topologically Correctness] Shrink a complicated triangle mesh to a simple one, without changing Euler number
    - [Geometrically Roughly Right] Keep using the averaged spatial position
- Consider its generalization to 3D…

# Progressive Tetrahedral Meshes

## Edge Collapse

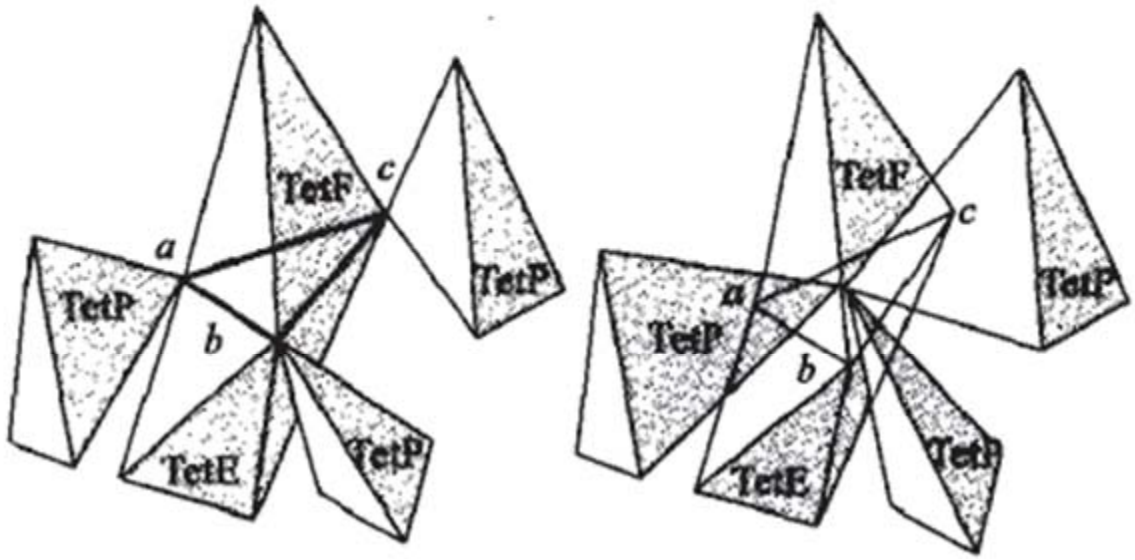


# Progressive Tetrahedral Meshes



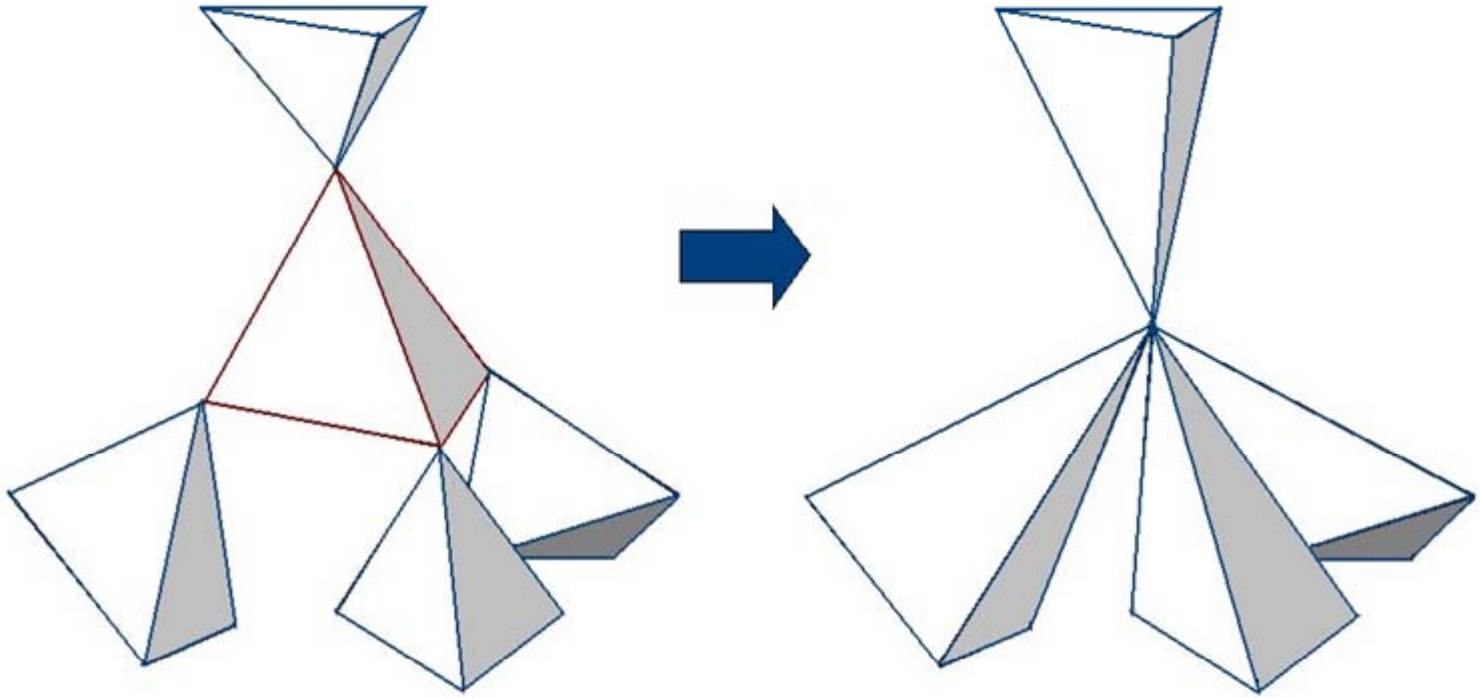
# Progressive Tetrahedral Meshes

Is "Edge Collapse" the only way?



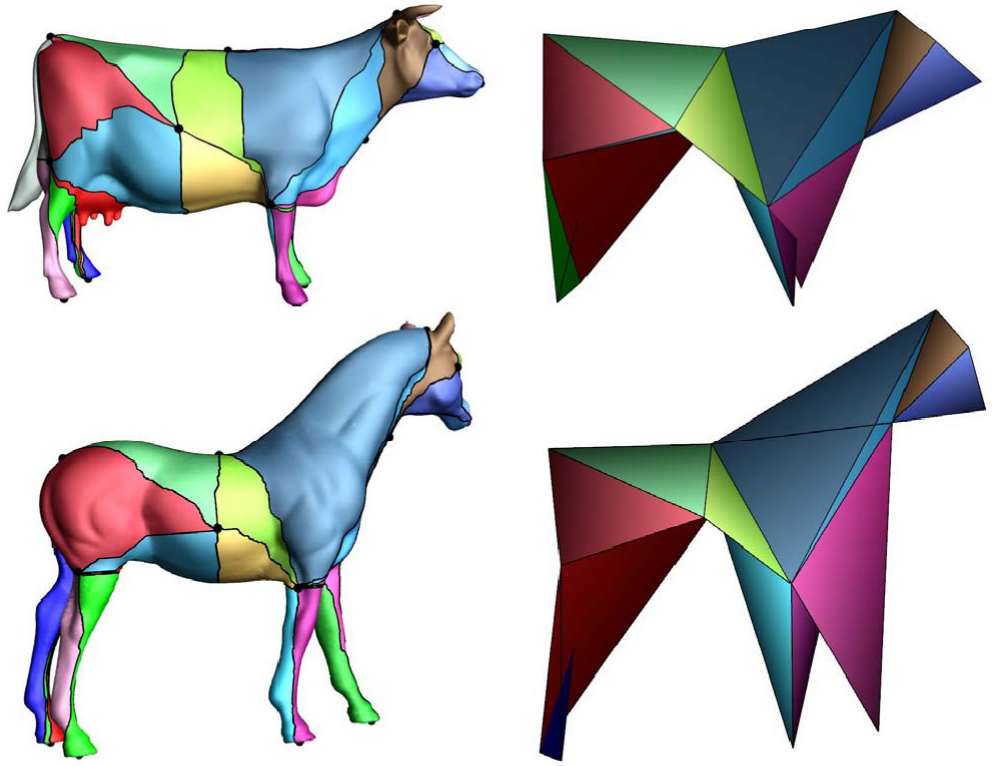
# Progressive Tetrahedral Meshes

Is "Edge Collapse" the only way?



# Some applications

Inter-surface mapping and morphing



# Some applications

Dynamic Collision Detection Video

And many more in visualization, vision, and CAGD…