A Developer's Survey of Polygonal Simplification algorithms

CS 563 Advanced Topics in
Computer Graphics

Fan Wu

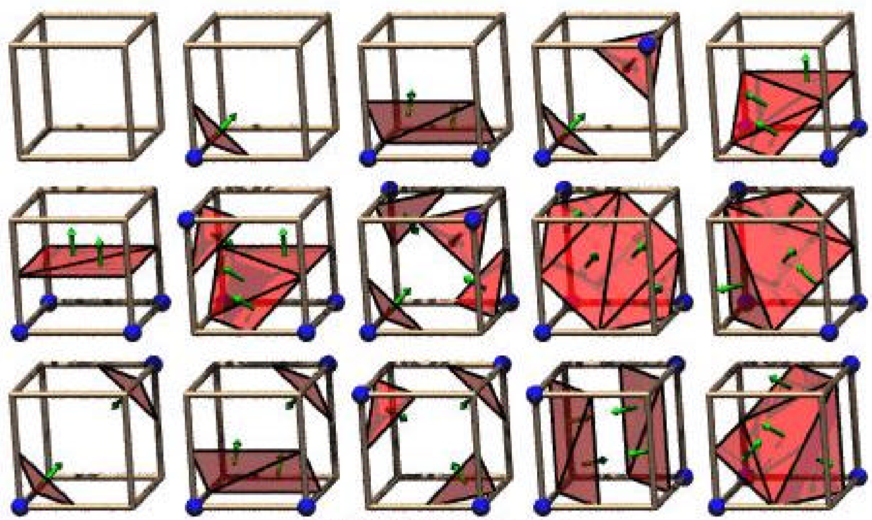
Mar. 31, 2005

Some questions to ask…

- Why simplification?
- What are my models like?
- What matters to me most?

Why?

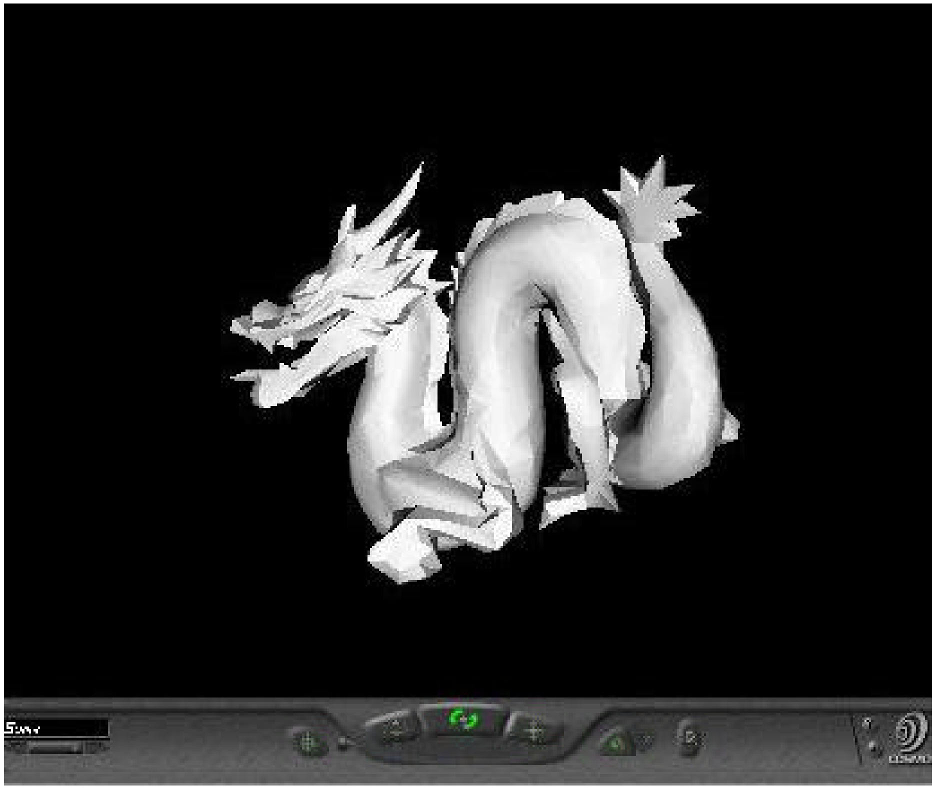
- Eliminate redundant geometry



The 15 Cube Combinations

Why?(Cont.)

- Download models from the web

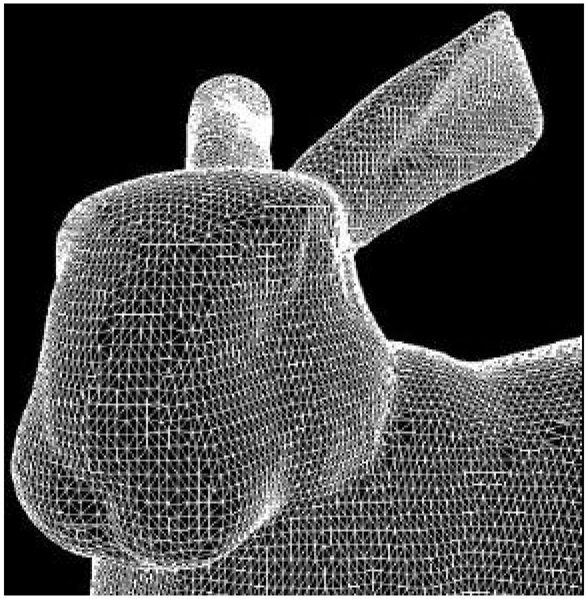


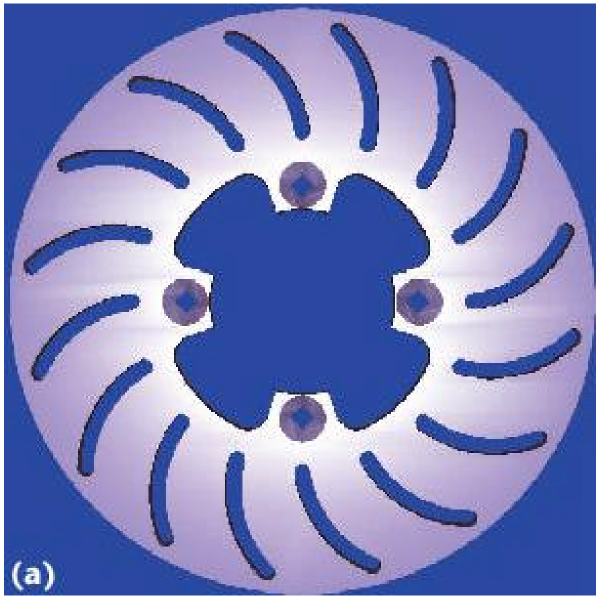
Why?(Cont.)

- Improve rendering speed (generating level of details) –Produce LODs

What are my models like?

- Smooth, organic forms
- Machine parts with sharp corners





❖Different models need different simplification algorithms

What are my models like? (Cont.)

- Precomputed lighting and textures
- Small number of large complex object
- Multiple moderately complex objects –video game
- Large number of small objects –CAD models

What's important for you?

- Geometric accuracy?
  - Volume deviation
- Visual fidelity? (how to measure?)
- Pre-processing time?
- Run-time rendering time?
- Automatic?

Taxonomy of SA

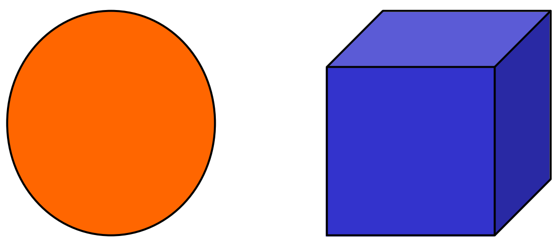
- Simplification algorithms can be classified into groups based on different criteria:
  - Topology preservation
  - Simplification Mechanism
  - Static/Dynamic/View-dependent

Topology

- Refers to the connected polygonal mesh's structure
  - Genus: the number of holes (handles) in the object
    - Sphere and Cube: 0
    - Doughnut and Coffee Cup : 1

Topology

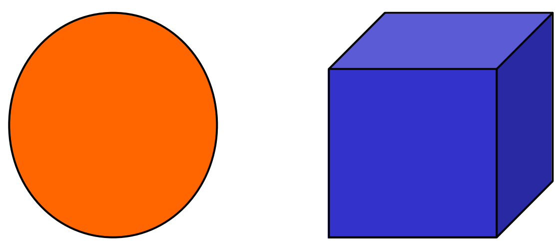
- Refers to the connected polygonal mesh's structure
  - Genus: the number of holes (handles) in the object
    - Sphere and Cube: 0



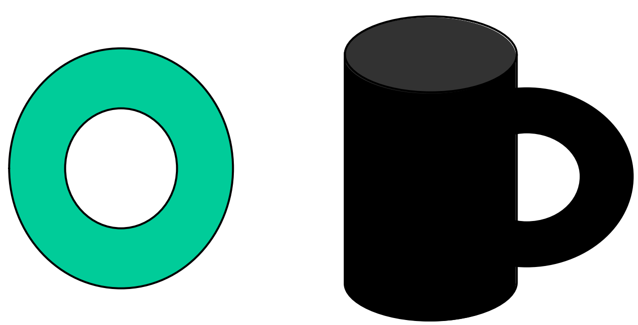
    - Doughnut and Coffee Cup : 1

Topology

- Refers to the connected polygonal mesh’s structure
  - Genus: the number of holes (handles) in the object
    - Sphere and Cube: 0



    - Doughnut and Coffee Cup : 1

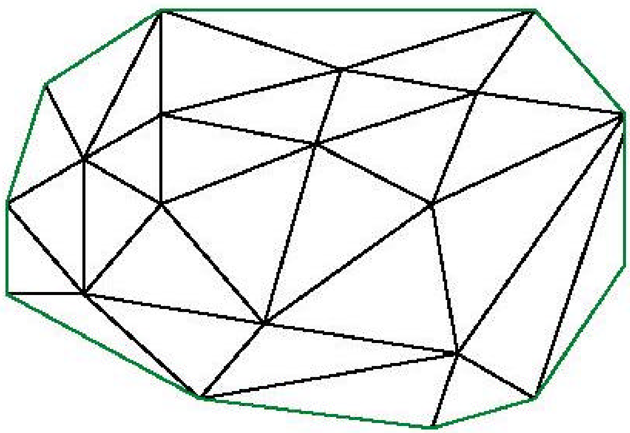


Topology (Cont.)

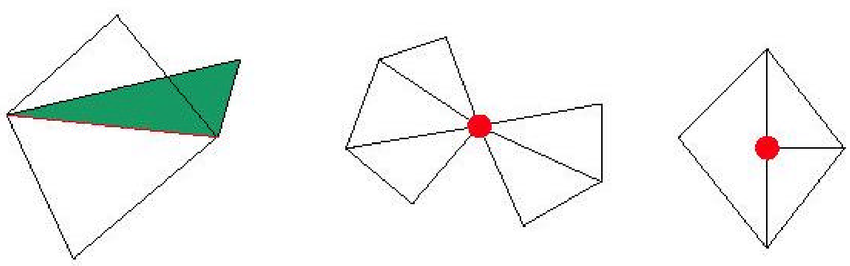
- 2D manifold (local property)
  - Exact two triangle share an edge
  - Every vertex is shared by a ring of triangles to form a surface (local topology is a disc)
  - Every triangle shares edges with exact three triangles

Topology (Cont.)

- Examples of manifold and non-manifold

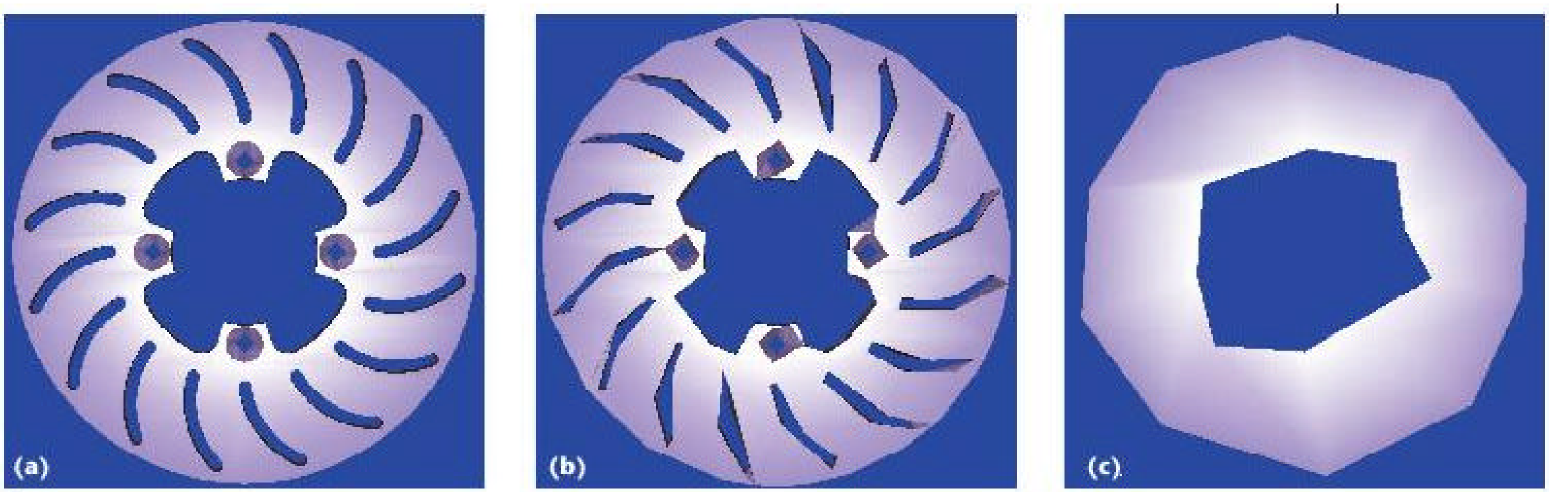


2 A 2D manifold with a boundary (boundary edges in green). One or two triangles share each edge and a connected ring of triangles shares each vertex.



Topology and SA

- Topology-preserving algorithms
  - Preserve manifold connectivity at every step (don't close up or create holes)



Yes Yes No

Topology and SA (Cont.)

Topology-preserving algorithms

- ☺ Good visual fidelity
- ☹ Limited simplification
- ☹ Often require the model to be a manifold to begin with (less robust)

Topology and SA (Cont.)

Opposite: Topology-modifying algorithms

- ☺ Permit drastic simplification
- ☹ Poor visual quality/Popping effect
- ☹ Often insensitive to topological features

✓Work best when drastic simplification is needed (visualization or complex scenes)

Topology and SA Summary

Questions to ask before using the algorithm:

- Topology-preserving or Topology-modifying?
- Topology-tolerant or Topology-sensitive?

Mechanism

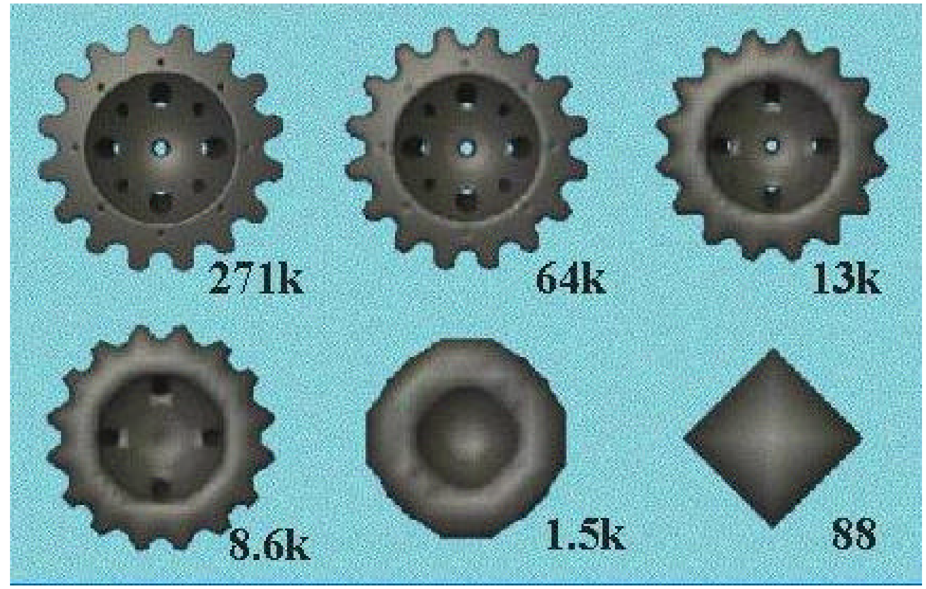
- Strategies used to remove polygons
  - Sampling
  - Adaptive Subdivision
  - Decimation
  - Vertex Merging

(1) Sampling

- Sample the surface:
  - Embedding a 3D volumetric grid around the surface
- Remove high-frequency (fine) features
  - Applying image processing to perform low-pass filter
- Recover the surface from filtered samples:
  - Recover the simplified surface from the low-passed volume

Sampling (Cont.)

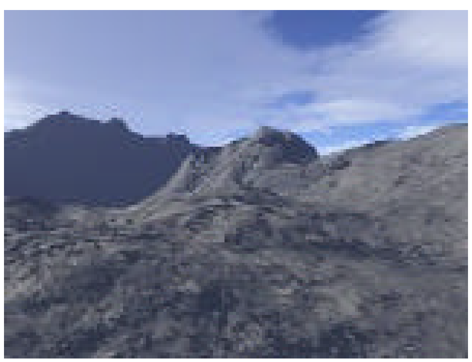
- ☺ High frequency features are gradually removed
- ☺ high frequency features are hard to sample

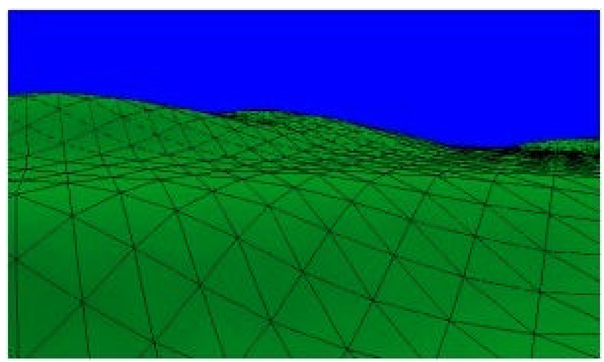


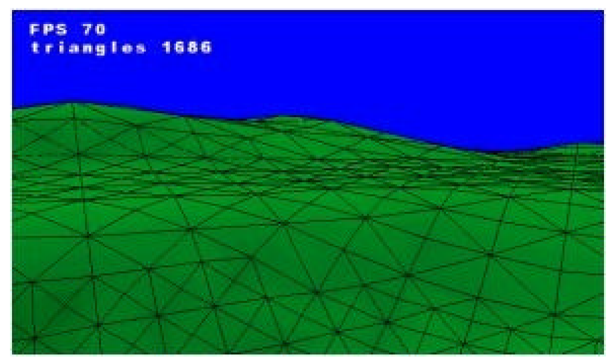
✓Work best for smooth objects

(2) Adaptive Subdivision

- Find a base mesh
- Recursively subdivide the base mesh to approach the initial mesh
- Example: Terrain simplification (rectangle as the base mesh)





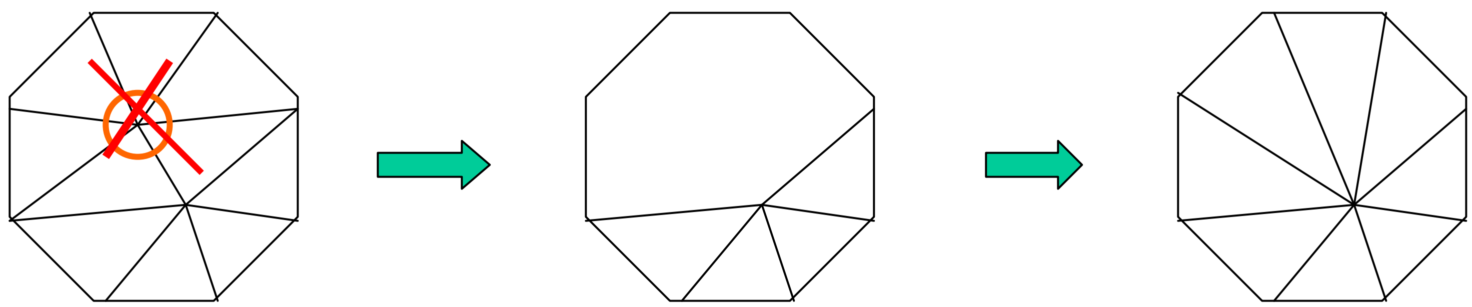


Adaptive Subdivision (Cont.)

- Work best when the base mesh is easily found (can be very hard)
- Preserve topology
- Multi-resolution editing (low level feature naturally propagate to finer level)

(3) Decimation

- Iteratively remove vertices from the mesh
- Remove the associated triangles
- Retriangulate the resulting holes at each step



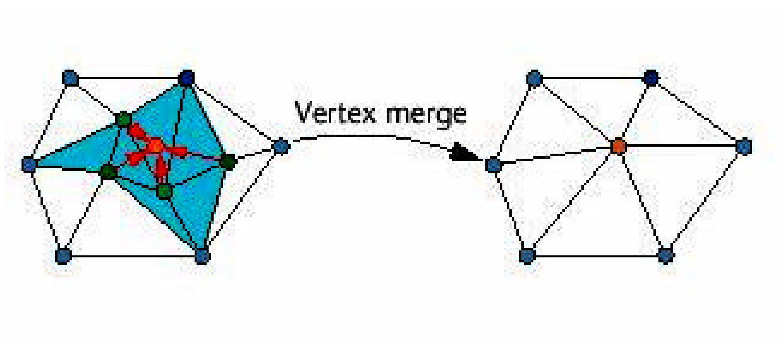
Decimation (Cont.)

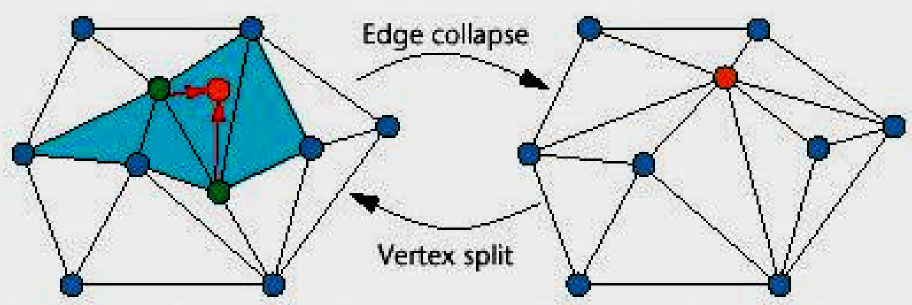
- Simple to code and very fast
- Topology preserving
- No drastic simplification
- Topology tolerant

Vertex Merging

- Collapsing two or more vertices into a single vertex
- Reconnecting the points with the remaining adjacent vertices
- Different algorithms have varying sophistication in which vertices to merge in what order

Vertex Merging (Cont.)





6 The edge-collapse operation merges exactly two vertices that share an edge. This eliminates two triangles from the mesh (one if the edge lies on a boundary). A vertex split is the dual of an edge collapse, introducing two triangles.

Mechanism Summary

- Sampling
- Adaptive Subdivision
- Decimation
- Vertex Merging

Static/Dynamic/View-dependent

- Static: Compute a fixed set of simplifications or LODs offline
- Dynamic: Encoding a continuous spectrum of LODs as opposed to a fixed set
  - Better granularity
  - Support progressive transmission

Static/Dynamic/View-dependent

- View-dependent: Extending Dynamic methods – LOD selections are based on view dependent criteria:
  - Distance to the eye
  - Silhouette regions
  - ☺ better fidelity for a given polygon count
  - ☹ increase the run time CPU load for choosing the LODs

A Brief catalog of algorithms

- A Brief catalog of algorithms
  - How they treat topology
  - Whether they use static, dynamic, or view-dependent simplification

Triangle mesh decimation

- 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.

Triangle mesh decimation

- Overall
  - Topology
    - Topology preserving
    - Topology tolerant
  - Mechanism
    - Decimation
  - Simple and fast
  - only limited to manifold surfaces

Triangle mesh decimation

▪Topology-modifying algorithm

- Topology
  - Topology modifying
- Mechanism
  - Decimation
- Decimation code
  - http://public.kitware.com/VTK/

Vertex clustering

- Uniform rectilinear partitioning of vertices into 3D grid cells
- Replace all vertices in a grid cell by a single representative vertex

Vertex clustering

- New algorithm:
  - Use error quadrics instead of grading each vertex with an important value
  - Compared to original vertex pair contraction algorithm, the sequence of contractions is determined by the cluster grid rather than by the mesh geometry
  - Floating-Cell clustering. Assign an importance to vertices and sort the vertices by importance

Vertex clustering

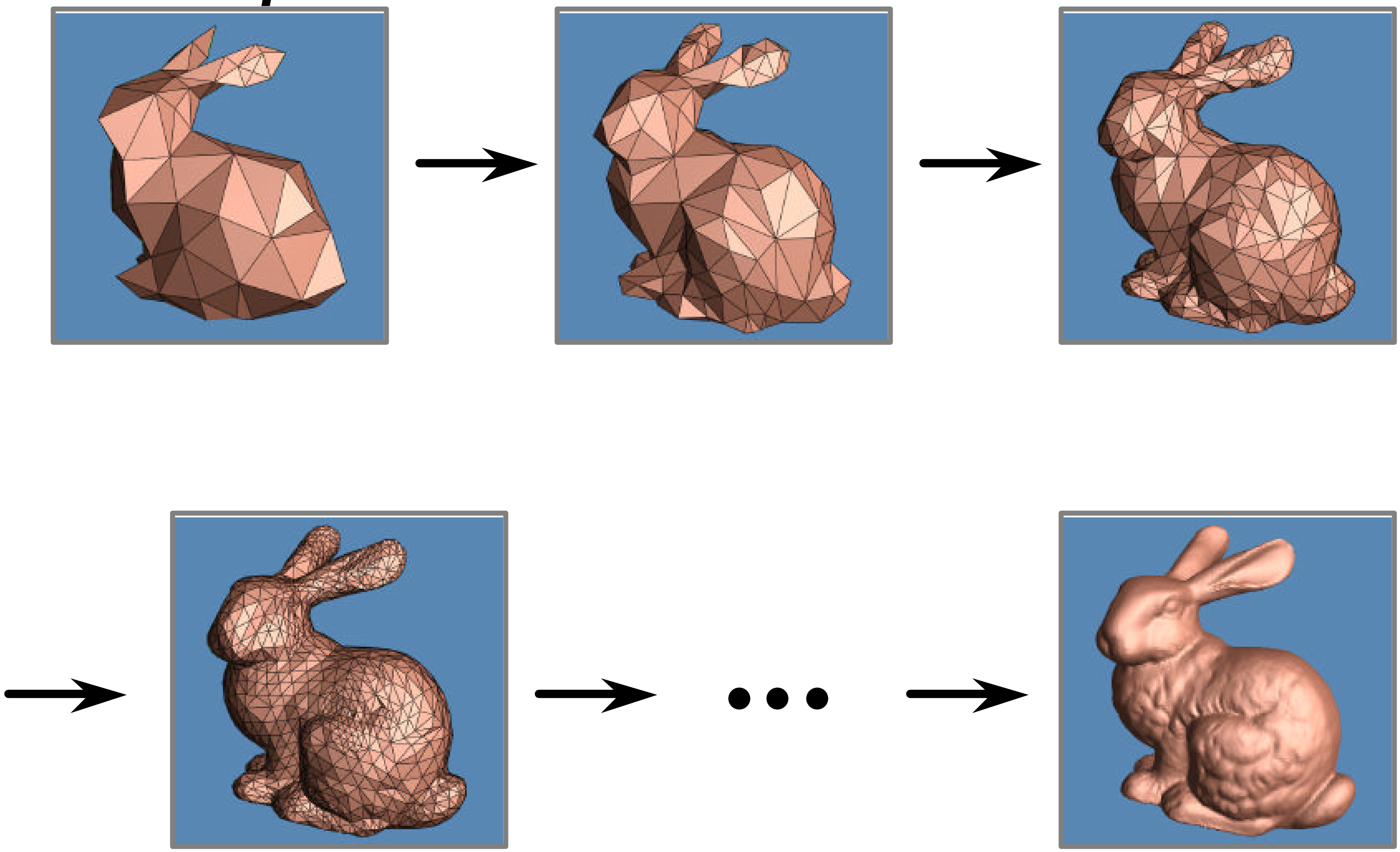
- Overall
  - Topology
    - Topology modifying
    - Topology insensitive
  - Fast
  - Poor quality

Multiresolution analysis of arbitrary meshes

- Main idea
  - Uses a compact wavelet representation to guide a recursive subdivision process
  - multiresolution representation of mesh M = base shape M^0 + sum of local correction terms (wavelet terms)

Multiresolution analysis of arbitrary meshes

base shape M^0



mesh M

Multiresolution analysis of arbitrary meshes

- Overall
  - Possesses the disadvantages of strict topology-preserving approaches
  - The fidelity of the resulting is high for smooth, organic forms
  - Difficulty in capturing sharp features

Voxel-based object simplification

- Main idea
  - Sampling
    - Polygonal
    - Range scanned
    - Volume
    - Mathematical function
  - Controlled Filtering, Use low-pass filtering to create frequency-limited versions of the sampled volume
  - Reconstruction
    - Volume reconstruction
    - Surface reconstruction

Voxel-based object simplification

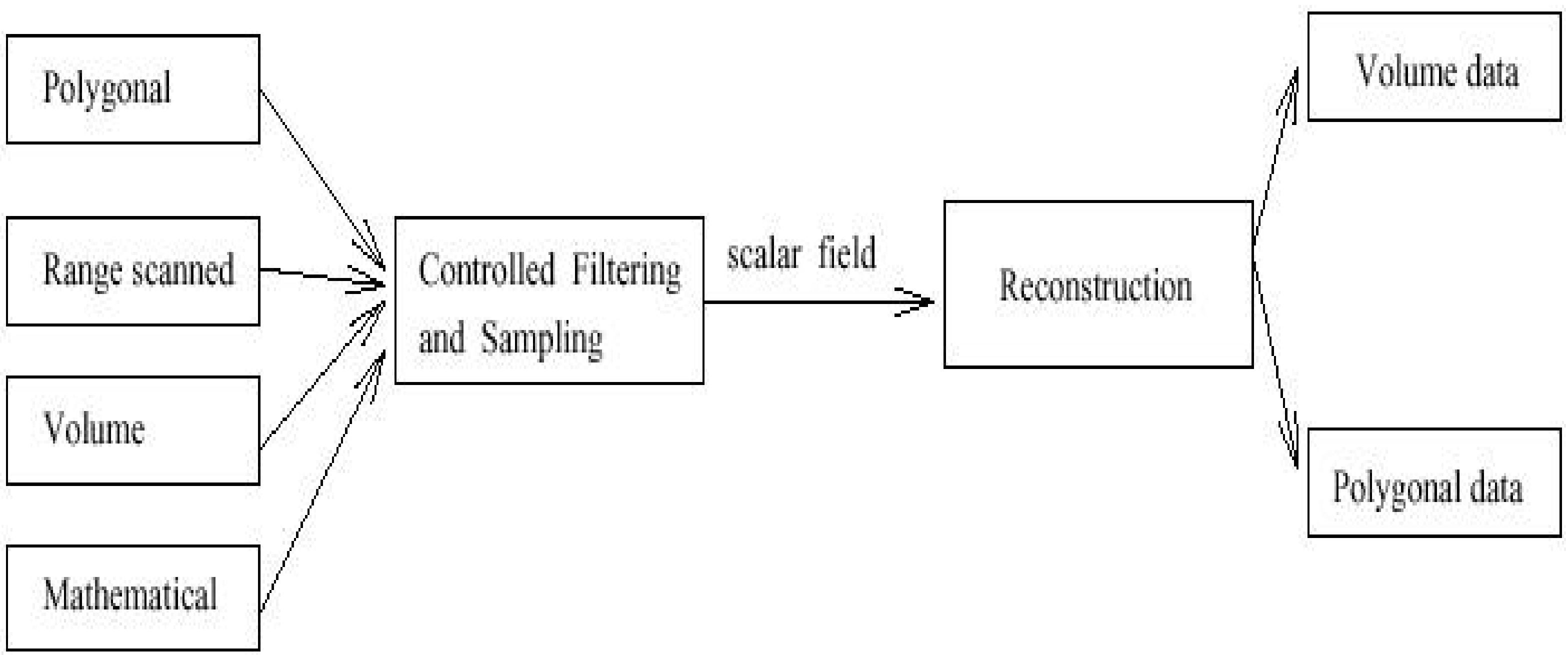


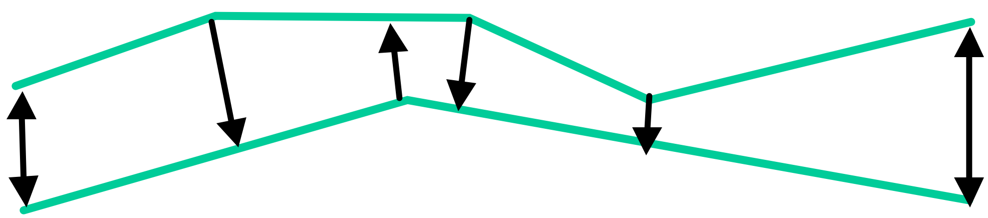
Fig. 1: *Pipeline for controlled topology simplification.*

Multiresolution analysis of arbitrary meshes

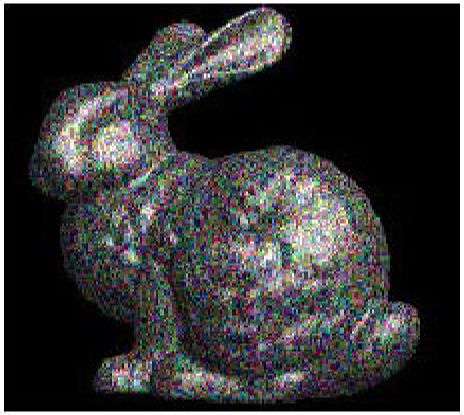
- Overall
  - High-frequency details are eliminated
  - Not topology tolerant

Simplification Envelopes

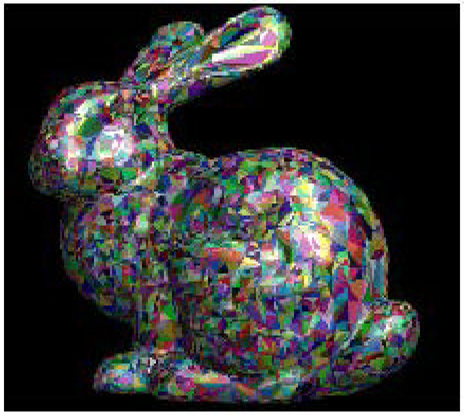
- Main idea
  - Bound error by requiring simplified surface to lie within surfaces offset by e



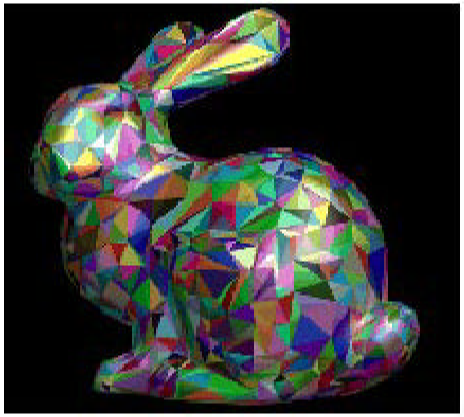
Simplification Envelopes



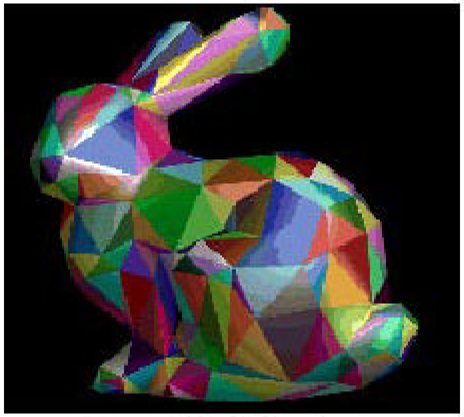
(a) bunny model: 69,451 triangles



(a) $\epsilon = 1/16\%$, 10,793 triangles



(a) $\epsilon = 1/4\%$, 2,204 triangles



(a) $\epsilon = 1\%$, 575 triangles

Cohen, Varshney, et al., "Simplification Envelopes," *Proceedings of SIGGRAPH 96*

Simplification Envelopes

- Overall
  - Topology preserving
  - Excellent choice where fidelity and topology preservation are crucial
  - Can be combined with most other meshes
  - The strict of preservation of topology curtails the capability for drastic simplification
  - http://www.cs.unc.edu/~geom/envelope.html

Appearance-preserving Simplification

- Main idea
  - Preserve three appearance attributes:
    - Surface Position
    - Surface Curvature
    - Material Color
  - Colors and normals stored in texture and normal maps
  - Texture deviation computed using parametric correspondence
  - Preserves colors and normals, bounding texture motion in object and screen space

Appearance-preserving Simplification

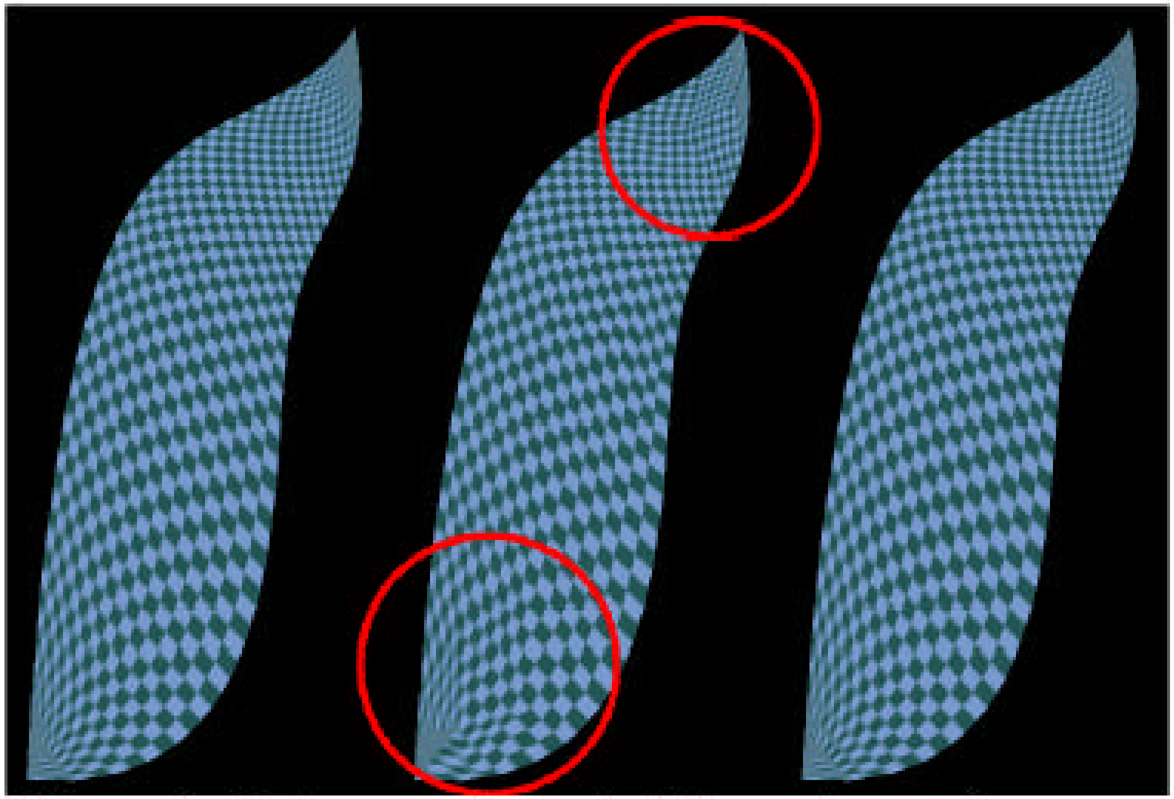


Figure 6: Texture coordinate deviation and correction on the lion's tail. *Left*: 1,740 triangles full resolution. *Middle and Right*: 0.25 mm maximum image deviation. *Middle*: 108 triangles, no texture deviation metric. *Right*: 434 triangles with texture metric.

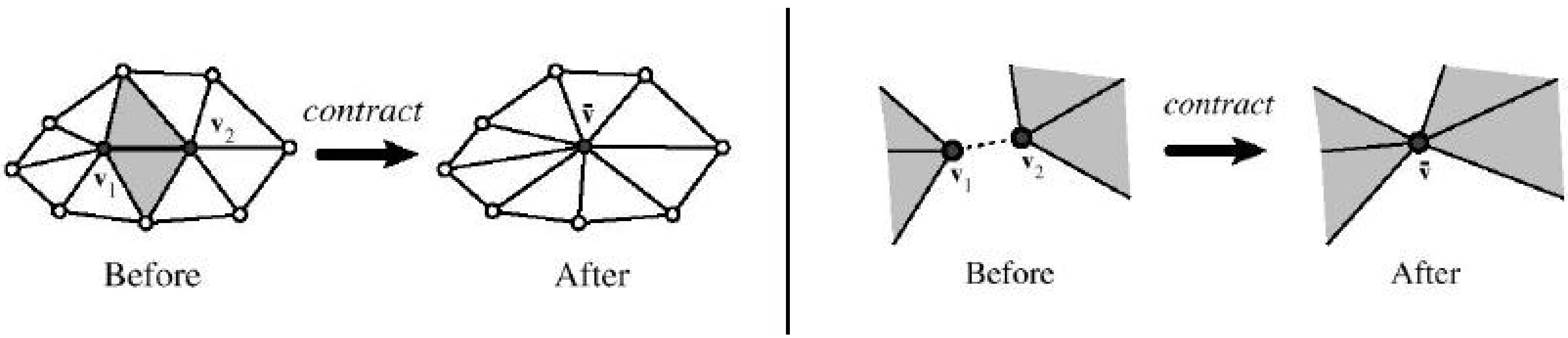
Cohen, Olano, and Manocha, "Appearance-Preserving Simplification," *Proceedings of SIGGRAPH 98.*

Appearance-preserving Simplification

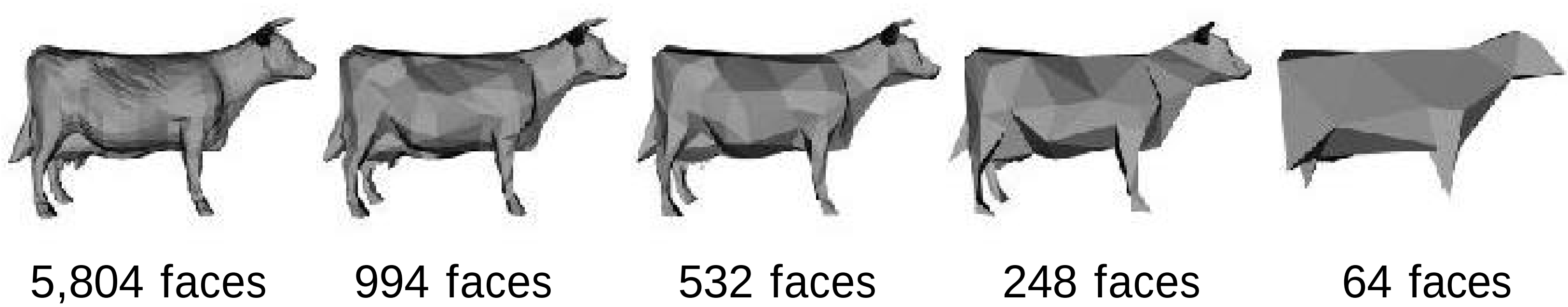
- Overall
  - Useful on models that don't require dynamic lighting
  - The standard for high-fidelity simplification

Quadric error metrics

- Main idea
  - Iteratively contract vertex pairs (a generalization of edge contraction)
  - As the algorithm proceeds, a geometric error approximation is maintained at each vertex of the model which is represented using quadric matrices
  - The algorithm proceeds until the simplification goals are satisfied.



Quadric error metrics



An example sequence of approximations generated by the algorithm. The entire sequence was constructed in about one second.

Quadric error metrics

- Overall
  - Continuous LODs
  - The algorithm can provide high efficiency, high quality and high generality
  - Running time approaches $O(n^2)$, but Erikson and Manocha proposed an adaptive threshold selection scheme that addresses this problem
  - Qslim. http://graphics.cs.uiuc.edu/~garland/software/qslim.html

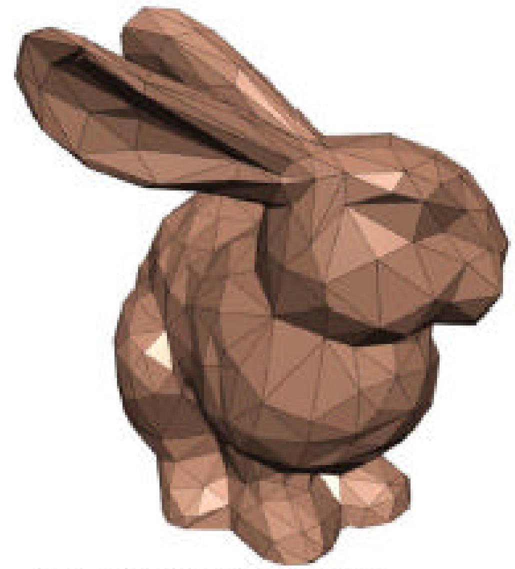
Image-driven Simplification

- Main idea
  - Use edge collapse
  - Two decisions
    - Where to place new vertex
    - How to order edge collapse (Lazy queue algorithm)
  - RMS root-mean-square (d_{RMS})

Image-driven Simplification

1a. Geometry–driven simplification.
$T = 1{,}336$. time = 1:05. $d_{RMS} = 4.10$.

1b. Original bunny model.
$T = 69{,}451$.



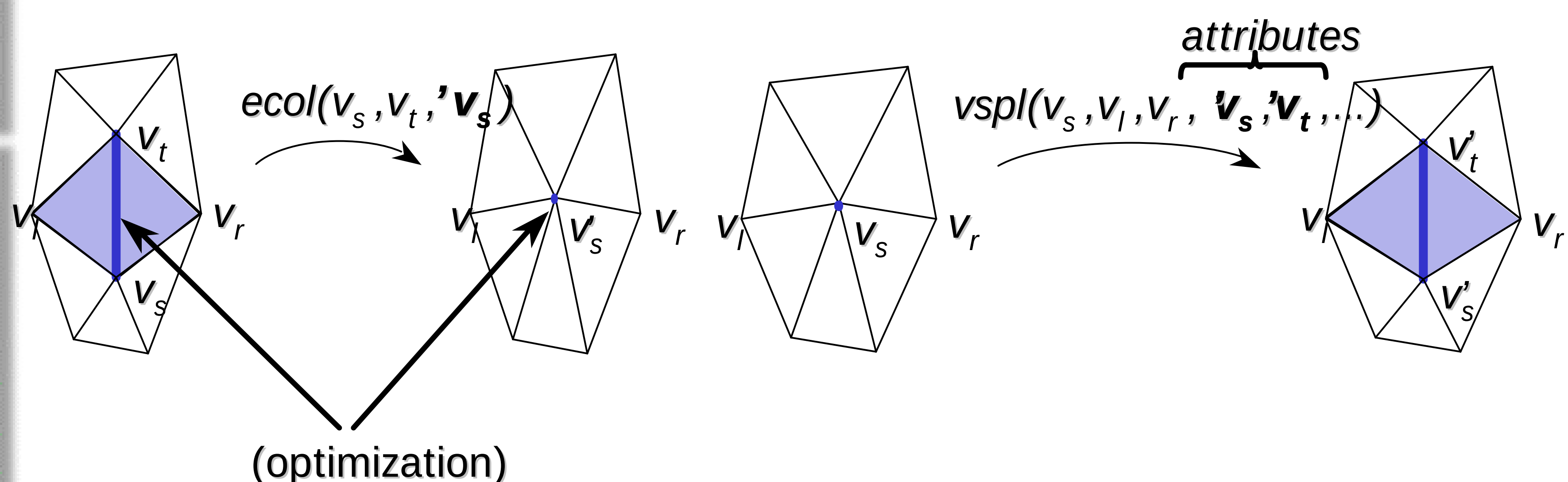
1c. Image–driven simplification.
$T = 1{,}333$. time = 12:15. $d_{RMS} = 3.75$.

Image-driven Simplification

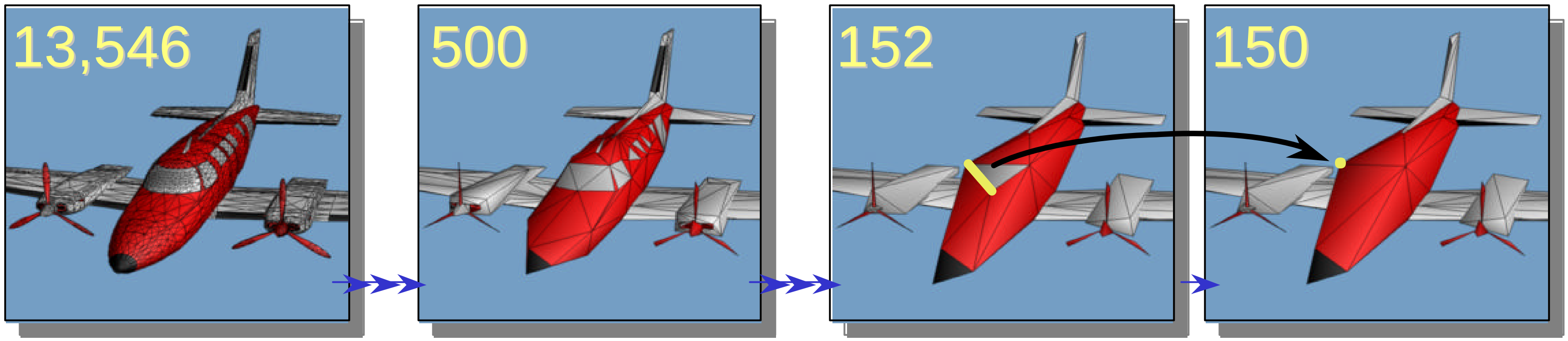
- Overall
  - Trade-off between simplification and the risk of missing an important effect
  - High-fidelity preservation of regions with drastic simplification of unseen model geometry
  - Slow

Progressive meshes

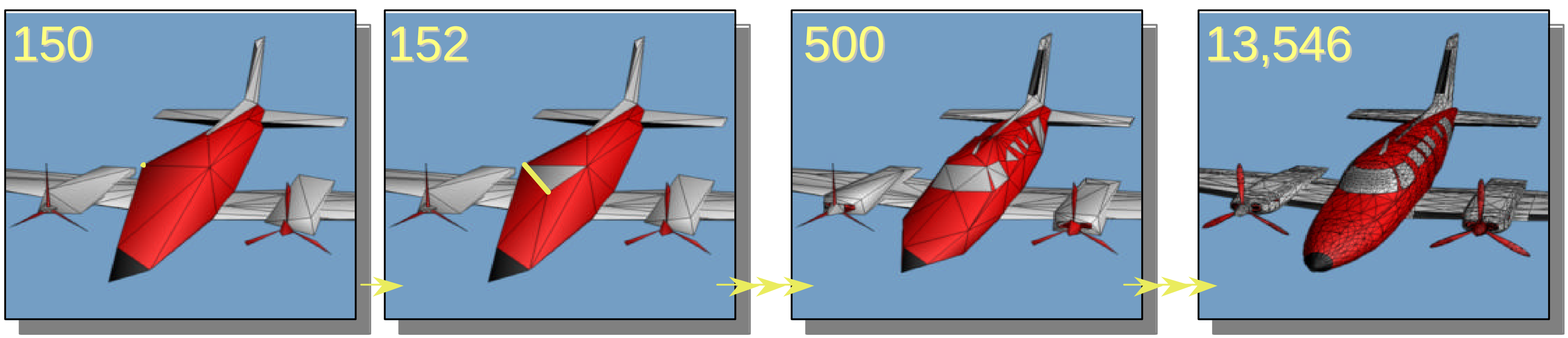
- Main idea
  - New mesh simplification procedure
    - Ecol
    - vspl
  - New representation
    - Progressive meshes



Progressive meshes



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



$$M^0 \xrightarrow{vspl_0} M^1 \xrightarrow{\ldots\ vspl_i\ \ldots} \ggg M^{175} \ggg \xrightarrow{vspl_{n-1}} M^n=\hat{M}$$

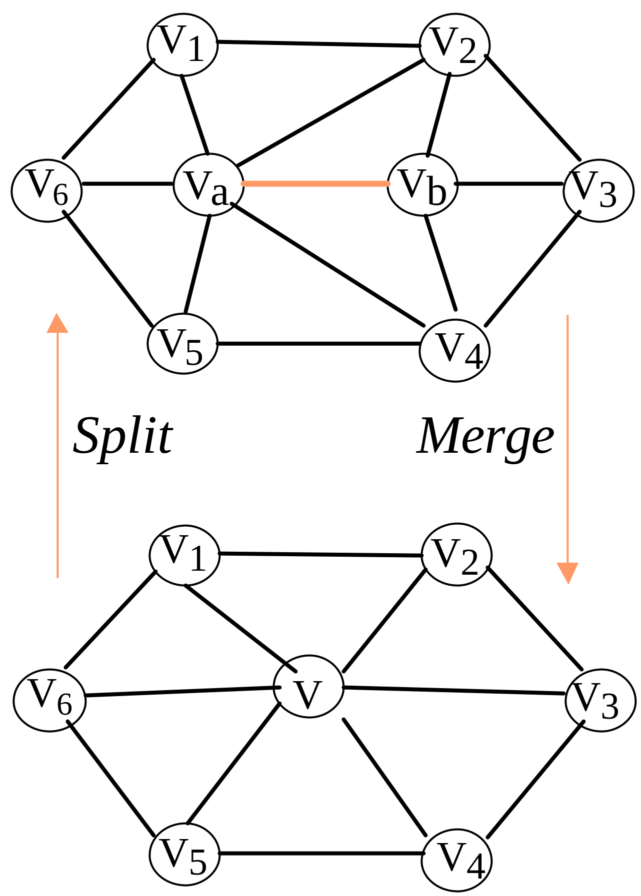
progressive mesh (PM) representation

Progressive meshes

- Overall
  - continuous-resolution
  - smooth LOD
  - space-efficient
  - progressive
  - Advantages of MRA
    - encodes geometry & color independently
    - supports multiresolution editing
  - Advantages of PM
    - lossless
    - more accurate
    - captures discrete attributes
    - captures discontinuities

Hierarchical dynamic simplification

- Main idea
  - Construct a view-dependent tree



Hierarchical dynamic simplification

- Overall
  - Not topology preserving
  - Topology tolerant
  - Dynamic
  - Simple to code
  - Any algorithm based on vertex merging can be used
  - fast, but limited by input geometry
  - VDSlib

    Http://vdslib.virginia.edu

Issues and Trends

- Converge on vertex merging as the underlying mechanism
- Hierarchical vertex merging provides general framework
- Still lack an agree-upon definition of fidelity
- Perceptual metric is hard to get

Reference

- [1] A Developer's Survey of Polygonal Simplification Algorithms, David P. Lueke *IEEE CG&A, May/June*, 2001
- [2] Topology Preserving Edge Contraction, Tamal Dey and Herbert Edelsbrunner
- [3] Decimation of Triangle Meshes, W. Schroeder, J. Zarge, and W. Lorenson, *ACM SIGGRAPH'92*
- [4] Simplification Using Quadric Error Metrics, M. Garland and P. Heckbert, *ACM SIGGRAPH'97*
- [5] Voxel-Based Object Simplification, T. He et al, *IEEE Visualization '95*
- [6] Simplification Envelopes, J. Cohen et al, *ACM SIGGRAPH 96*
- [7] Appearance Perserving Simplification, J. Cohen et al, *ACM SIGGRAPH 98*
- [8] View-Dependent Simplification of Arbitrary Polygonal Environment, D. Luebke and C. Erikson , *ACM SIGGRAPH'97*
- [9] Out-of-Core Simplification of Large Polygonal Models, P. Lindstrom, *ACM SIGGRAPH 2000*
- [10] Image-Driven Simplification, P. Lindstrom and G. Turk, *ACM Transactions on Graphics, vol.19, no. 3*
- [11] Geometric Compression, M. Deering, *ACM SIGGRAPH 95*

Reference

- [7] Appearance Perserving Simplification, J. Cohen et al, *ACM SIGGRAPH 98*
- [8] View-Dependent Simplification of Arbitrary Polygonal Environment, D. Luebke and C. Erikson , *ACM SIGGRAPH'97*
- [9] Out-of-Core Simplification of Large Polygonal Models, P. Lindstrom, *ACM SIGGRAPH 2000*
- [10] Image-Driven Simplification, P. Lindstrom and G. Turk, *ACM Transactions on Graphics, vol.19, no. 3*
- [11] Geometric Compression, M. Deering, *ACM SIGGRAPH 95*