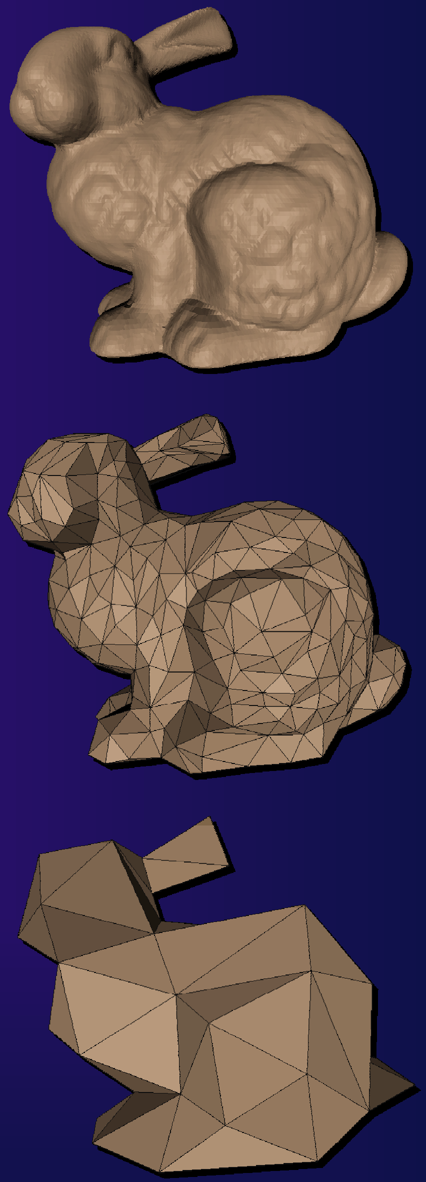
Multiresolution Modeling: Survey & Future Opportunities

Michael Garland

University of Illinois at Urbana-Champaign, USA

September 1999



The Problem of Detail

Graphics systems are awash in model data

- very detailed CAD databases
- high-precision surface scans

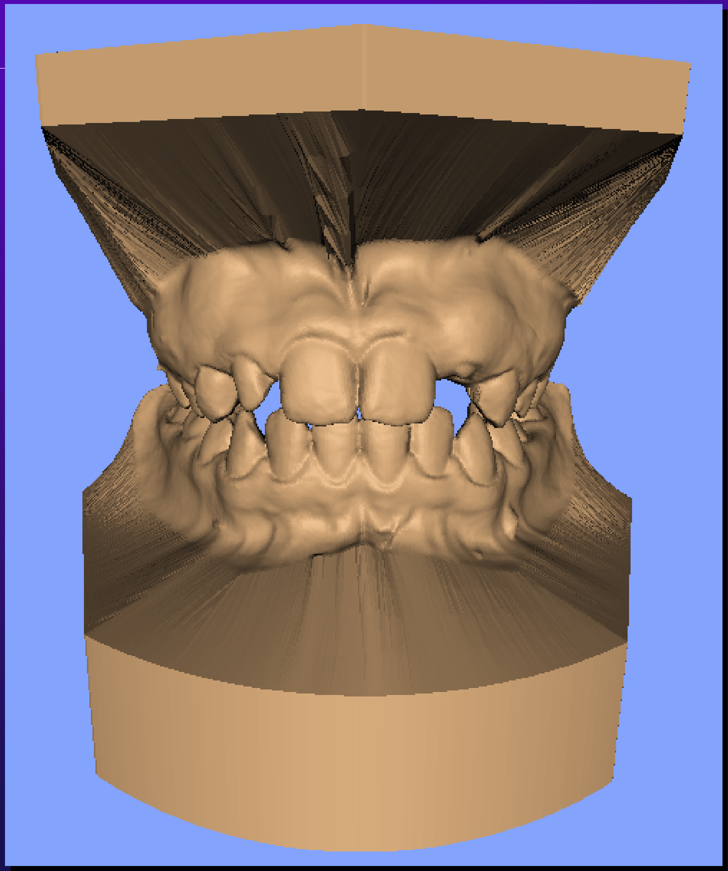
Available resources are always constrained

- CPU, space, graphics speed, network bandwidth

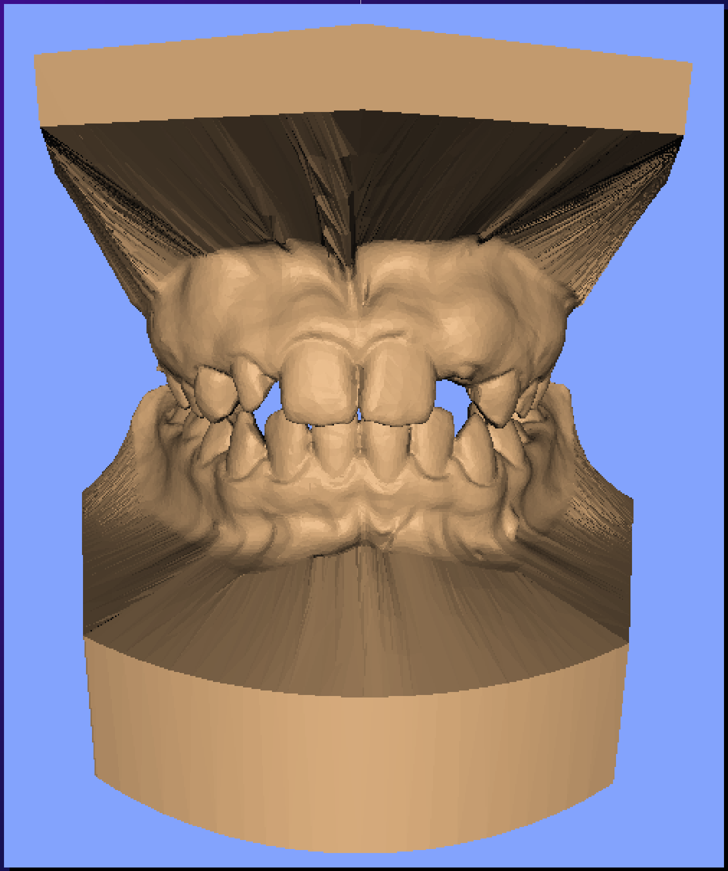
We need economical models

- want the minimum level of detail (LOD) required

A Non-Economical Model

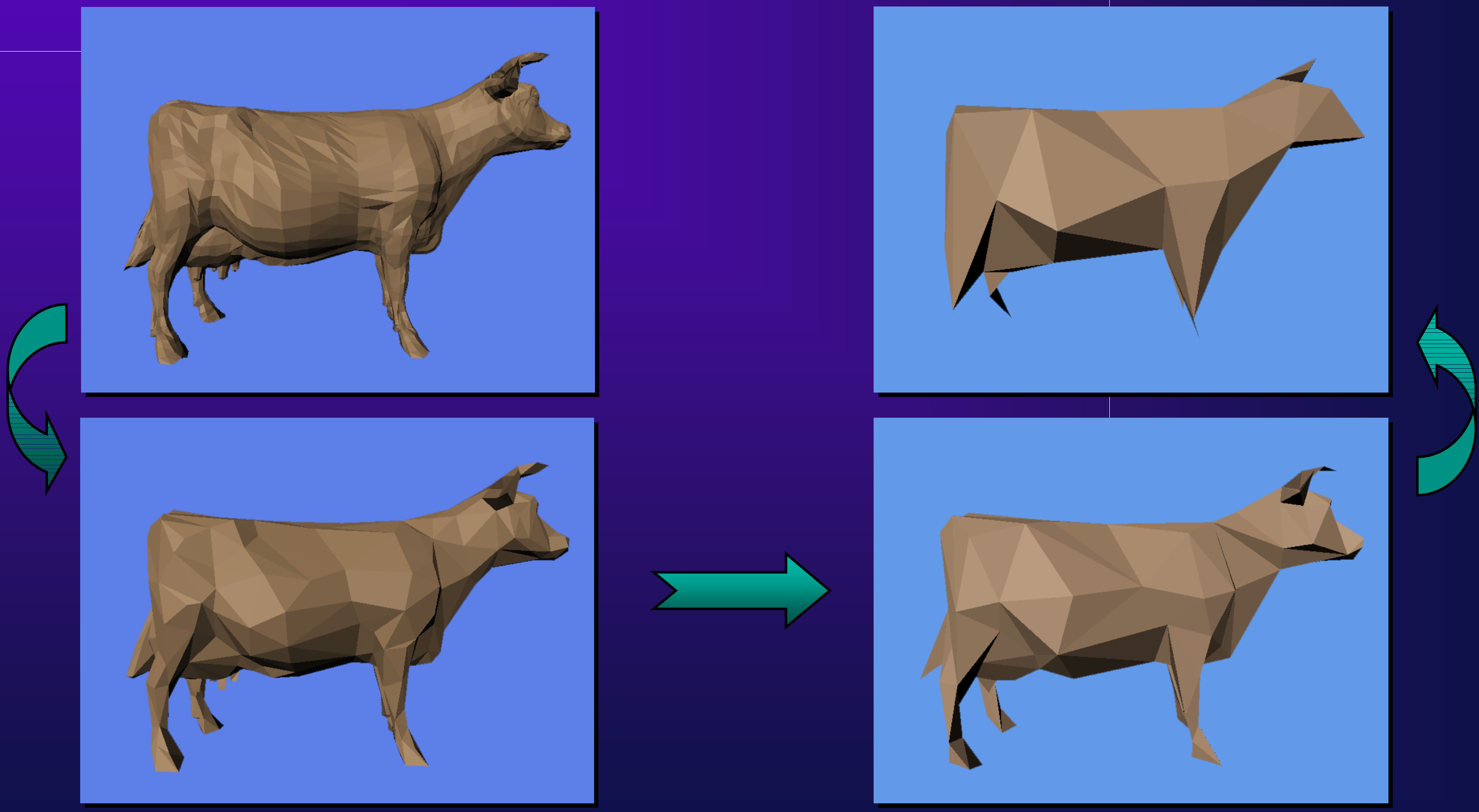


424,376 faces



60,000 faces

Automatic Surface Simplification



Automatic Surface Simplification

Produce approximations with fewer triangles

- should be as similar as possible to original
- want computationally efficient process

Need criteria for assessing similarity of models

- for display, visual similarity is the ultimate goal
- similarity of shape is often used instead
  - *generally easier to compute*
  - *lends itself more to applications other than display*

Focus on Polygonal Models

Polygonal surfaces are ubiquitous

- only primitive widely supported in hardware
- near-universal support in software packages
- output of most scanning & reconstruction systems

Switching representations is no solution

- indeed, some suffer from the same problem
- many applications *want* polygons

Will always assume models are triangulated

Historical Background

Function approximation [y=f(x)]

- long history in mathematical literature

Piecewise linear curve approximation

- various fields: graphics, cartography, vision, …

Height field (i.e., terrain) triangulation

- research back to at least early 70's
- important for flight simulators

Other Related Fields

Geometry compression

- simplification is a kind of lossy compression

Surface smoothing

- reduces geometric complexity of shape

Mesh generation

- finite element analysis (e.g., solving PDE's)
- need appropriate mesh for good solution
- overly complex mesh makes solution slow

Overview of Simplification Methods

Manual preparation has been widely used

- skilled humans produce excellent results
- very labor intensive, and thus costly

Most common kinds of automatic methods

- vertex clustering
- vertex decimation
- iterative contraction

Optimal Approximations

Achieve given error with fewest triangles

- no mesh with fewer triangles meets error limit

Computationally feasible for curves

- $O(n)$ for functions of one variable
- but $O(n^2 \log n)$ for plane curves

Intractable for surfaces

- NP-hard to find optimal height field [Agarwal–Suri 94]
- must also be the case for surfaces

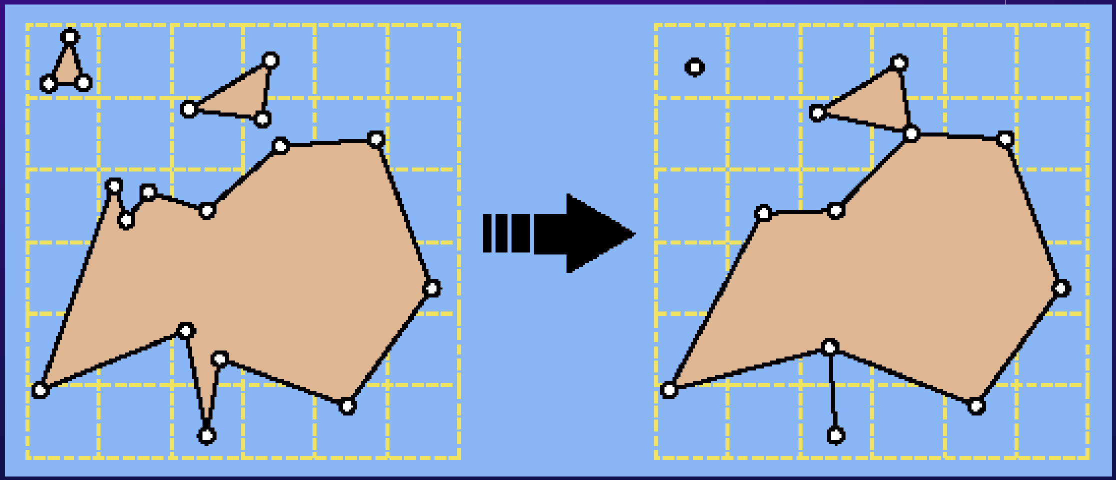
Vertex Clustering

Partition space into cells

- grids [Rossignac-Borrel], spheres [Low-Tan], octrees, ...

Merge all vertices within the same cell

- triangles with multiple corners in one cell will degenerate



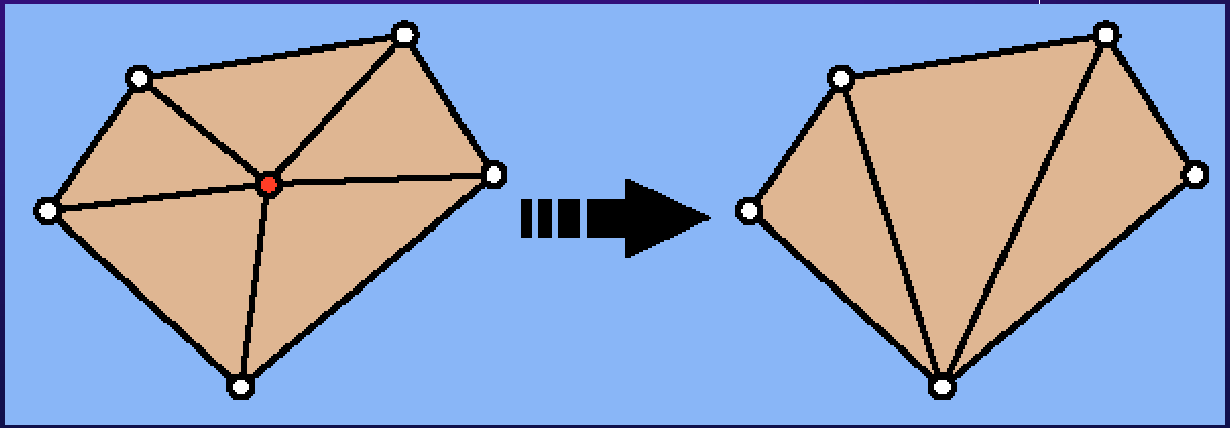
Vertex Decimation

Starting with original model, iteratively

- rank vertices according to their importance
- select unimportant vertex, remove it, and retriangulate hole

A fairly common technique

- Schroeder *et al*, Soucy–Laurendeau, Klein *et al*, Ciampalini *et al*



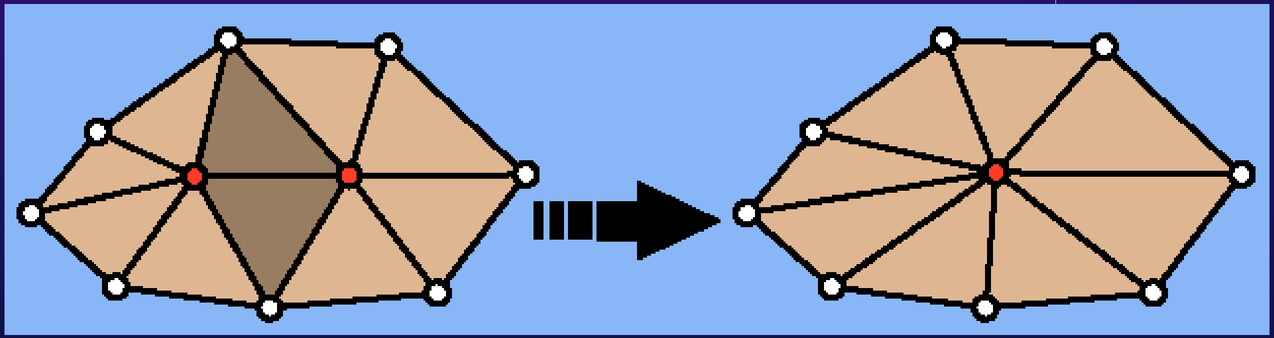
Iterative Contraction

Contraction can operate on any set of vertices

- edges (or vertex pairs) are most common, faces also used

Starting with the original model, iteratively

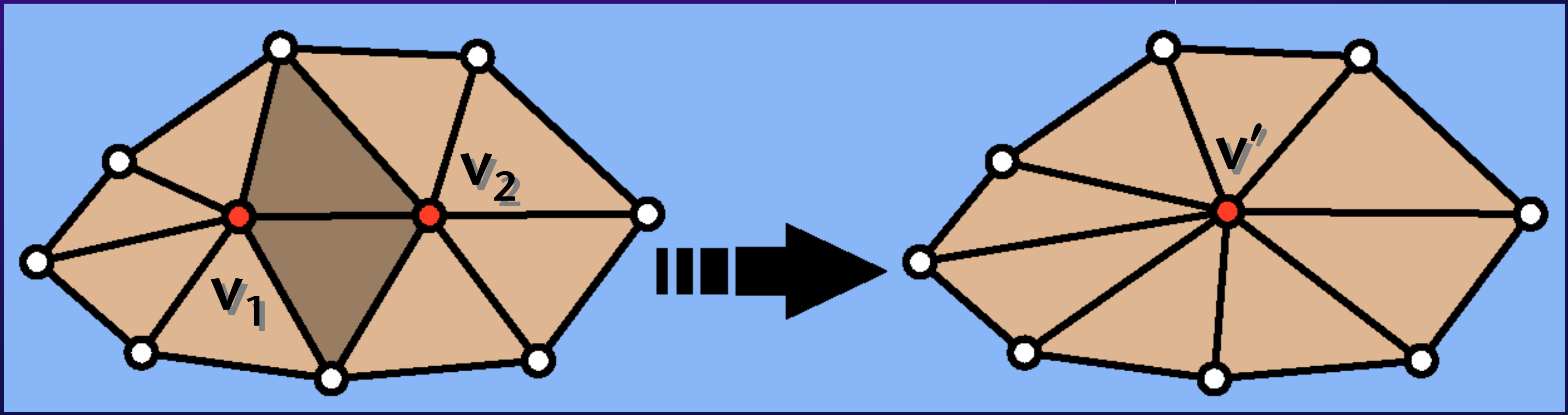
- rank all edges with some cost metric
- contract minimum cost edge
- update edge costs



Edge Contraction

A single edge contraction $(v_1,v_2) \rightarrow v'$ is performed by

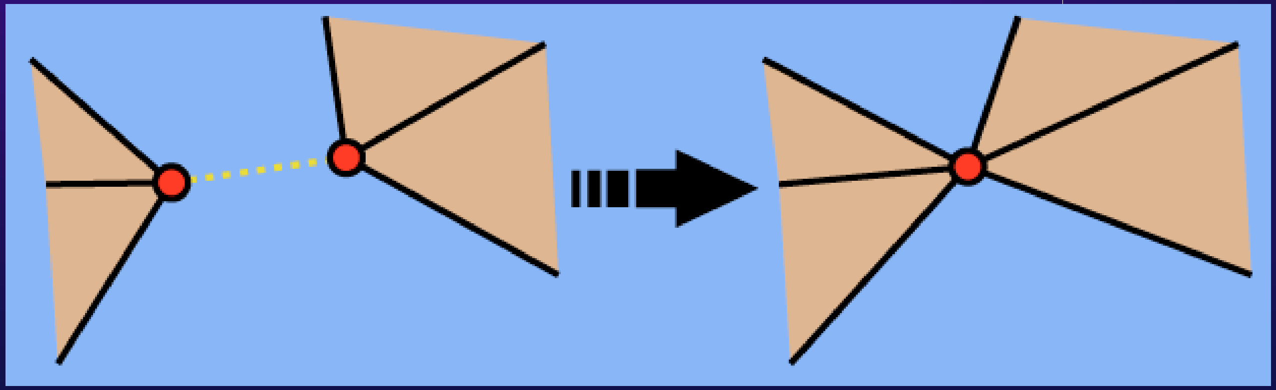
- moving v_1 and v_2 to position v'
- replacing all occurrences of v_2 with v_1
- removing v_2 and all degenerate triangles



Vertex Pair Contraction

Can also easily contract any pair of vertices

- fundamental operation is exactly the same
- joins previously unconnected areas
- can be used to achieve topological simplification



Iterative Edge Contraction

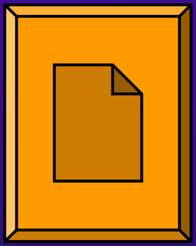
Currently the most popular technique

- Hoppe, Garland–Heckbert, Lindstrom-Turk, Ronfard-Rossignac, Guéziec, and several others
- simpler operation than vertex removal
- well-defined on any simplicial complex

Also induces hierarchy on the surface

- a very important by-product
- enables several multiresolution applications

Demo



Cost Metrics for Contraction

Used to rank edges during simplification

- reflects amount of geometric error introduced
- the main differentiating feature among algorithms

Must address two interrelated problems

- what is the best contraction to perform?
- what is the best position v′ for remaining vertex?
  - *can just choose one of the endpoints*
  - *but can often do better by optimizing position of v′*

Cost Metrics for Contraction

Simple heuristics

- edge length, dihedral angle, surrounding area, …

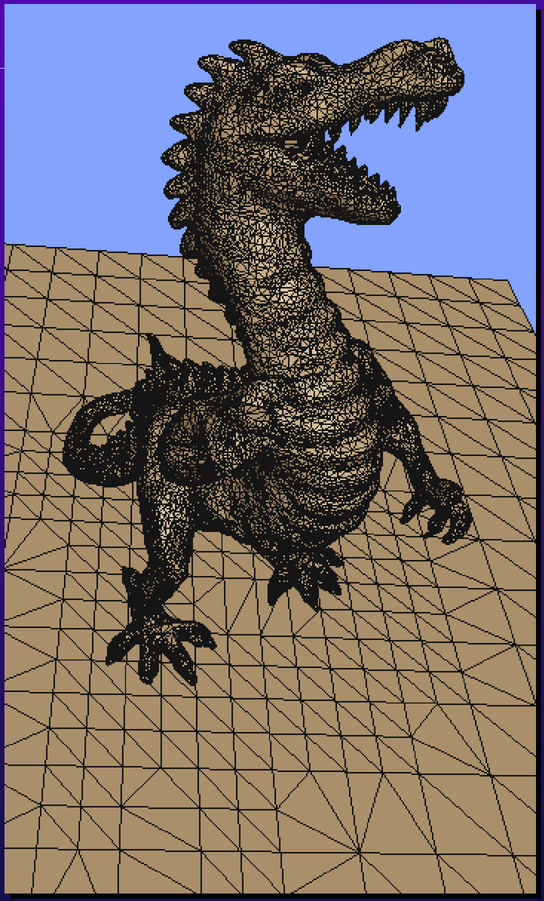
Sample distances to original surface

- projection to closest point [Hoppe]
- restricted projection [Soucy–Laurendeau, Klein *et al*, Ciampalini *et al*]

Alternative characterization of error

- quadric error metrics [Garland–Heckbert]
- local volume preservation [Lindstrom–Turk]

Must Also Consider Attributes



Mesh for solution

Radiosity solution

Must Also Consider Attributes

50,761 faces

10,000 faces

Simplification Summary

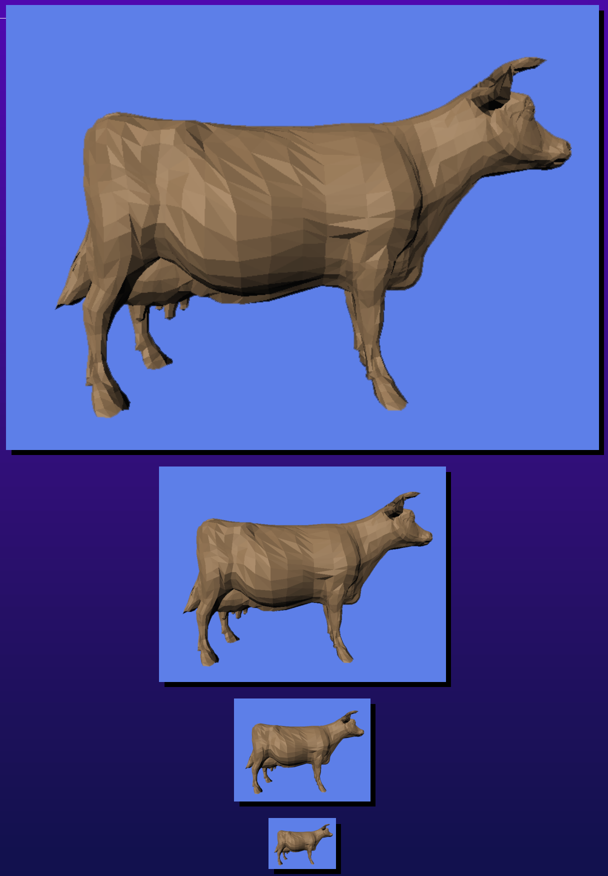
Spectrum of effective methods developed

- high quality; very slow *[Hoppe et al 93, Hoppe 96]*
- good quality; varying speed
  [Schroeder et al 92; Klein et al 96; Ciampalini et al 97; Guéziec 95 Garland-Heckbert 97; Ronfard-Rossignac 96; Lindstrom-Turk 98]
- lower quality; very fast *[Rossignac–Borrel 93; Low–Tan 96]*
- result is usually produced by transforming original

Various other differentiating factors

- is topology simplified? restricted to manifolds?
- are attributes simplified or re-sampled into maps?

Static Resolution is Not Enough



Model used in variety of contexts

- many machines; variable capacity
- projected screen size will vary

Context dictates required detail

- LOD should vary with context
- context varies over time
- with what level of coherence?
  - *generally high coherence in view*
  - *possibly poor coherence in load*

Need Multiresolution Models

Encode wide range of levels of detail

- extract appropriate approximations at run time
- must have low overhead
  - *space consumed by representation*
  - *cost of changing level of detail while rendering*
- can be generated via simplification process

Image pyramids (mip-maps) a good example

- very successful technique for raster images

Discrete Multiresolution Models

Given a model, build a set of approximations

- can be produced by any simplification system
- at run time, simply select which to render

Inter-frame switching causes visual "popping"

- can smooth transition with image blending
- or use geometry blending: geomorphing [Hoppe]

Supported by several software packages

- RenderMan, Open Inventor, IRIS Performer, ...

Limits of Discrete Models

We may need varying LOD over surface

- large surface, oblique view (eg. viewer on terrain)
  - *need high detail near the viewer*
  - *need less detail far away*
- single LOD will be inappropriate
  - *either excessively detailed in the distance (wasteful)*
  - *or insufficiently detailed near the viewer (visual artifacts)*

Doesn't really exploit available coherence

- small view change may cause large model change

Progressive Meshes

We get more than just final approximation

- sequence of contractions
- corresponding intermediate approximations

Re-encode as progressive mesh (PM) [Hoppe 96]

- take final approximation to be base mesh
- reverse of contraction sequence is a split sequence
- can reconstruct any intermediate model
- allows for progressive transmission & compression

PM's a Limited Multiresolution

More flexibility is required

- local addition/subtraction of triangles
  - *as conditions change, make small updates in LOD*
  - *this is the multi-triangulation framework [De Floriani et al]*
- may require novel approximations

Must encode dependency of contractions

- PM's imply dependency on all earlier contractions
- but we can reorder non-overlapping contractions

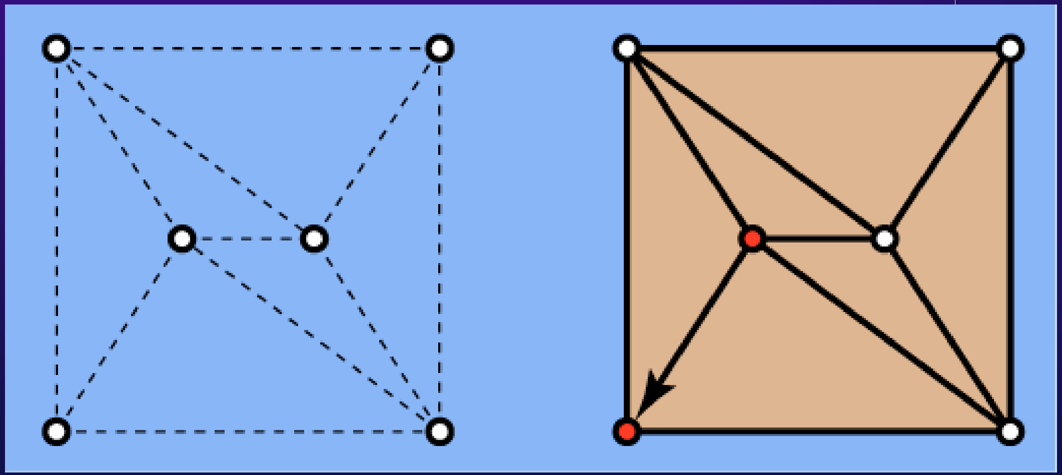
Structure Induced on Surface

Every vertex on approximation corresponds to

- a connected set of vertices on the original
- hence a region on the surface: the union of neighborhoods

Initial conditions

- every vertex set is a singleton, every region a neighborhood



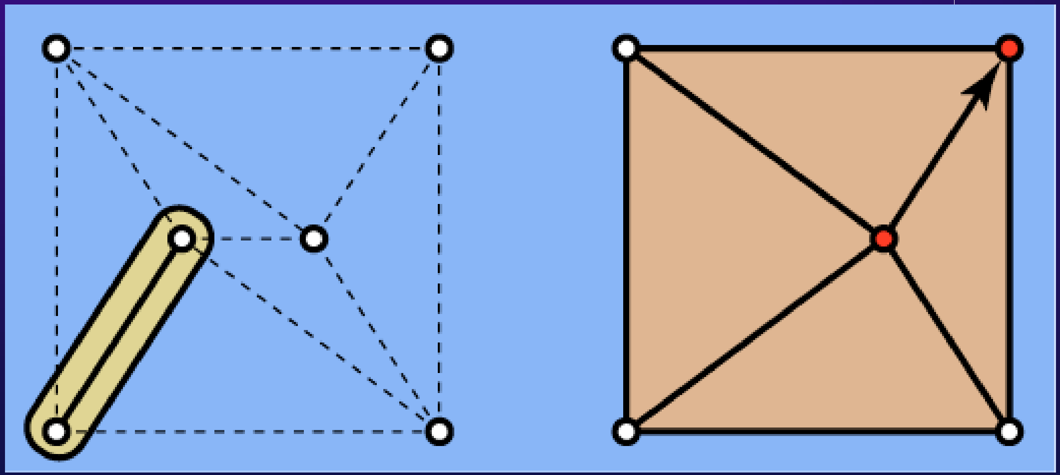
Structure Induced on Surface

A contraction merges corresponding vertex sets

- remaining vertex accumulates larger surface region

When merging regions, can link them by a mesh edge

- as shown on left hand side

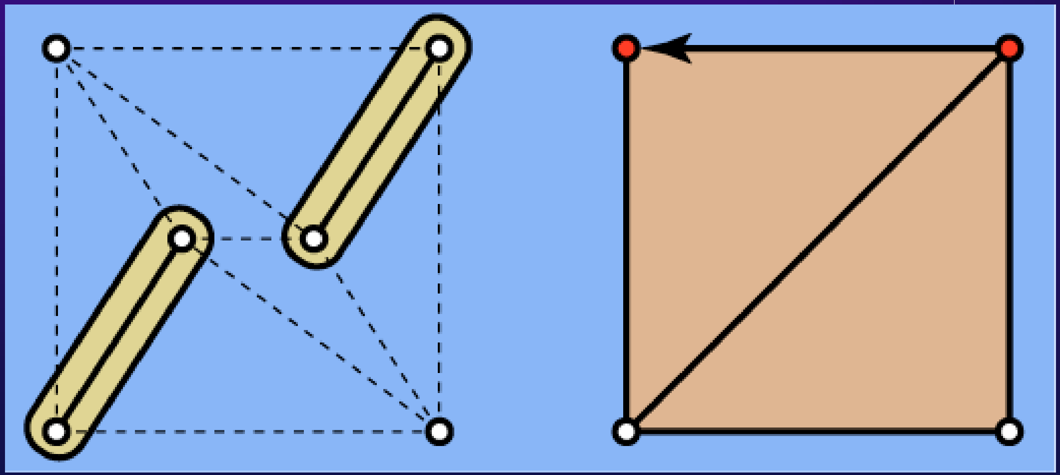


Structure Induced on Surface

Links within single region form spanning tree

- links within all regions form spanning forest
- any contraction order within regions is (topologically) valid

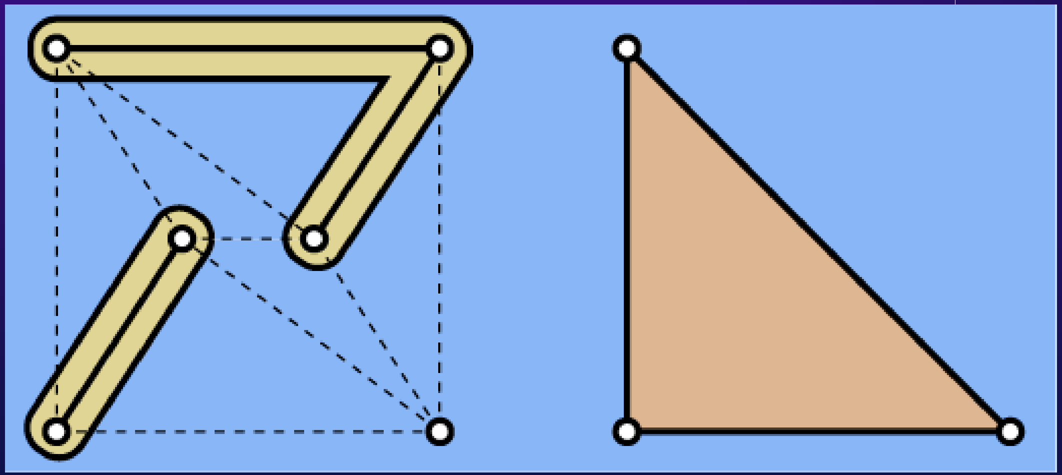
Regions always completely partition original surface



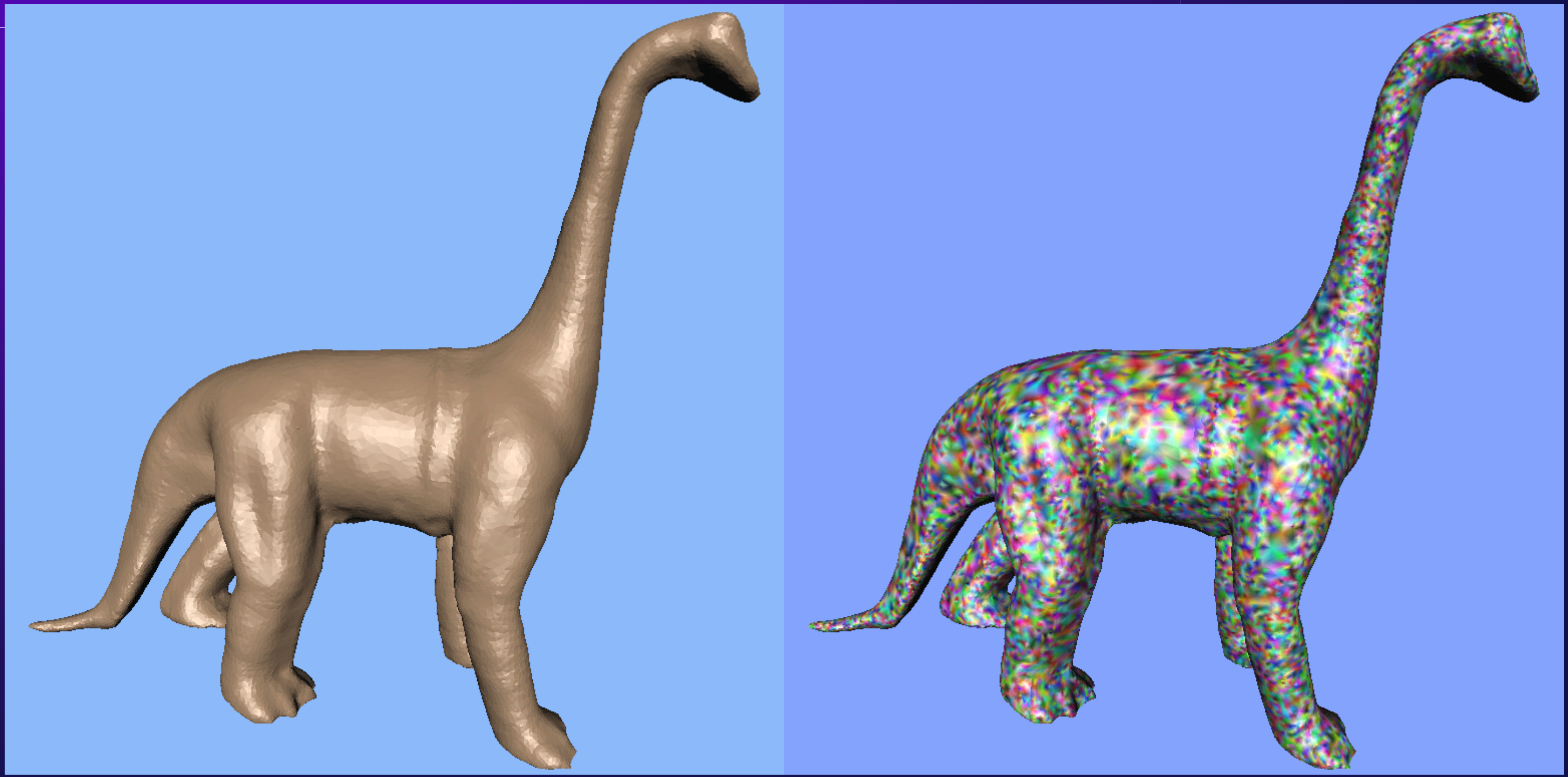
Structure Induced on Surface

Pair-wise merging forms hierarchy

- binary tree of vertices
- also a binary tree of surface regions

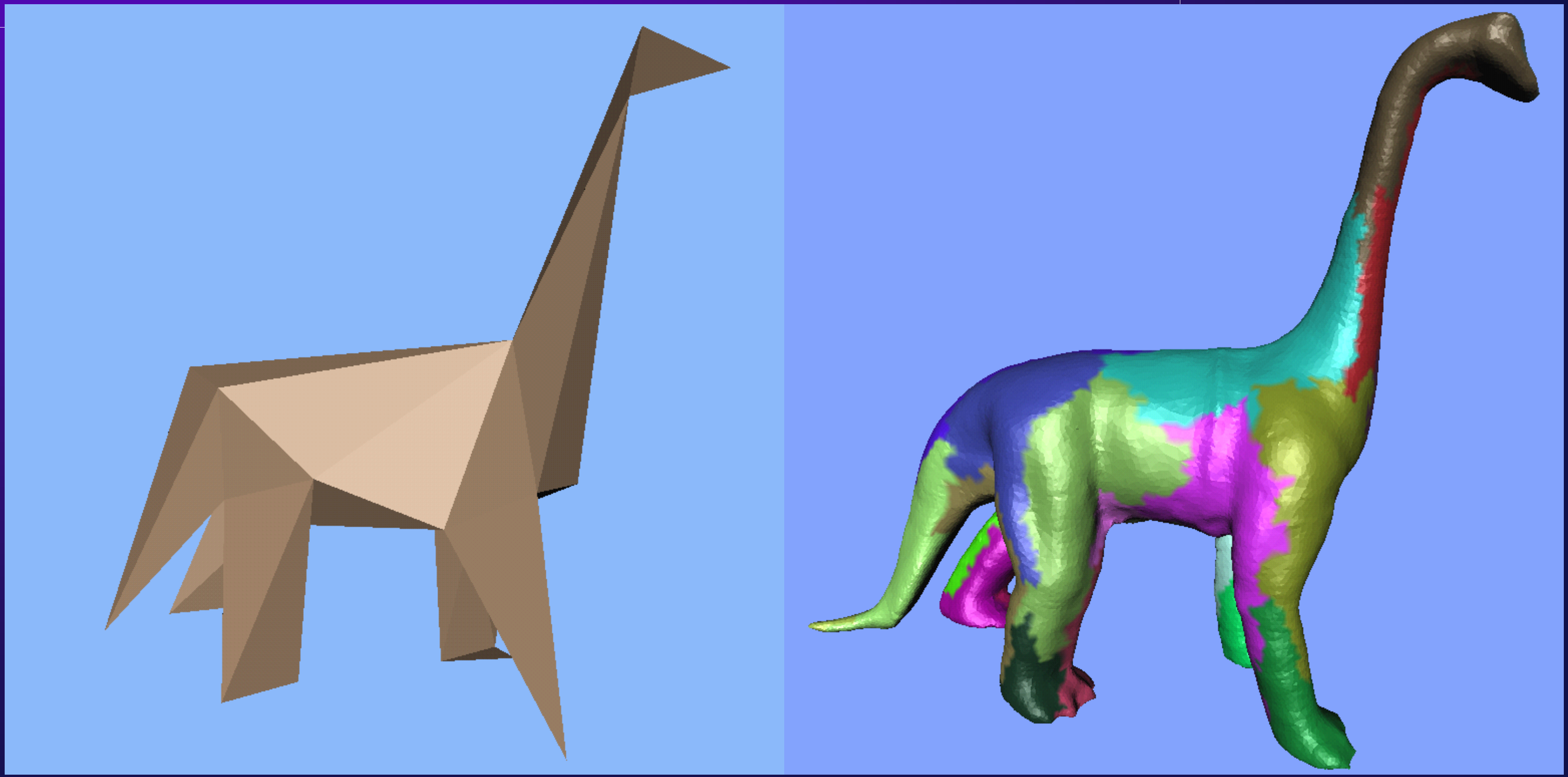


Example: Initial Vertex Neighborhoods

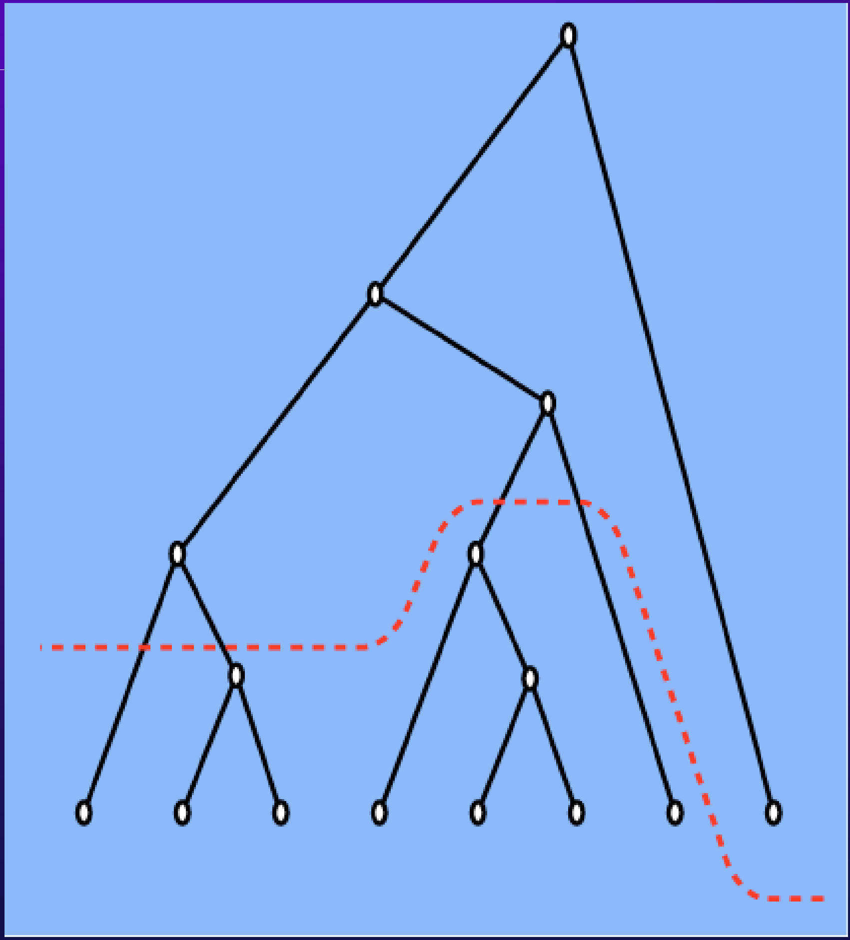


Example:
After 99% of vertices removed

Example:
After 99.9% of vertices removed



Vertex Hierarchies



A cut through the tree

- contract all below cut
- remaining leaves are active
- determines partition
- and an approximation

Encodes dependencies

- PM's assume total order
- disjoint subtrees are indep.
- novel approximations arise
- but must avoid fold-over

Vertex Hierarchies for View-Dependent Refinement

Multiresolution representation for display

- incrementally move cut between frames
  [Xia-Varshney, Hoppe, Luebke-Erickson]
- move up/down where less/more detail needed
- relies on frame-to-frame coherence
- can accommodate geomorphing [Hoppe]

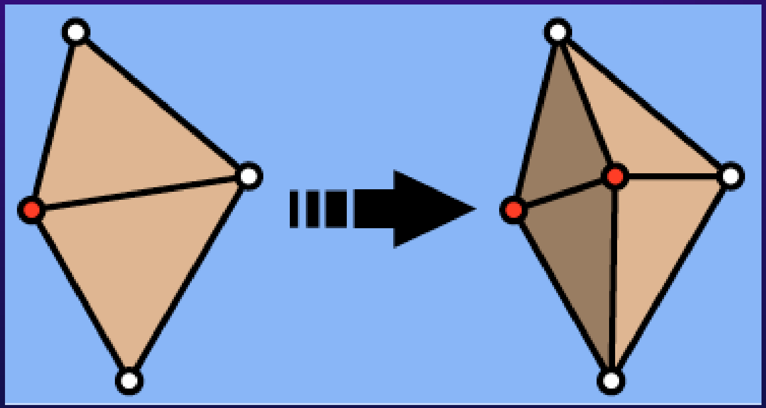
Most common application of vertex hierarchy

- uses hierarchy only to guide active front evolution
- more flexibility & overhead than discrete multires.

Further Refinement in Vertex Hierarchies

Also support synthetic refinement

- edge contraction is an inverse of edge split
- can synthesize temporary levels in tree by splitting edges
- fractal extrapolation of terrain surface, for example



Applications Beyond Display

Other important applications are appearing

- surface editing [Guskov *et al* 99]
- surface morphing [Lee *et al* 99]
- multiresolution radiosity [Willmott *et al* 99]

Still others seem promising

- hierarchical bounding volumes
- object matching
- shape analysis / feature extraction

Multiresolution Model Summary

Representations are available to support

- progressive transmission
- view-dependent refinement
- hierarchical computation (e.g., radiosity)

But limitations remain

- vertex hierarchies may over-constrain adaptation
- adaptation overhead not suitable for all cases
- interacting multiresolution objects largely ignored

Looking Ahead

We've reached a performance plateau

- broad range of methods for certain situations
- incremental improvement of existing methods

Major progress may require new techniques

- broader applicability of simplification
- higher quality approximations

Requires better understanding of performance

- how well, in general, does an algorithm perform?

Greater Generality

Many applications require non-rigid surfaces

- articulated models for animation

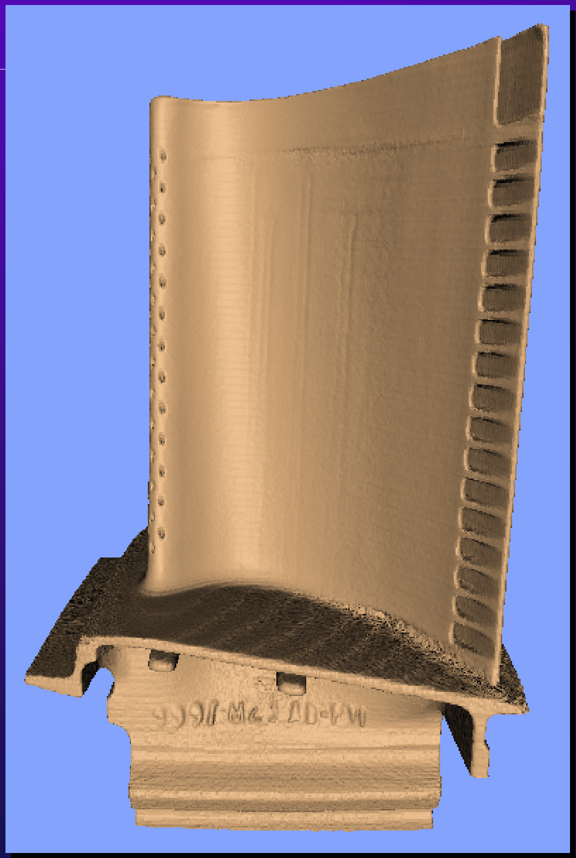
Other model types also have complexity issues

- tetrahedral volumes, spline patch surfaces, ...

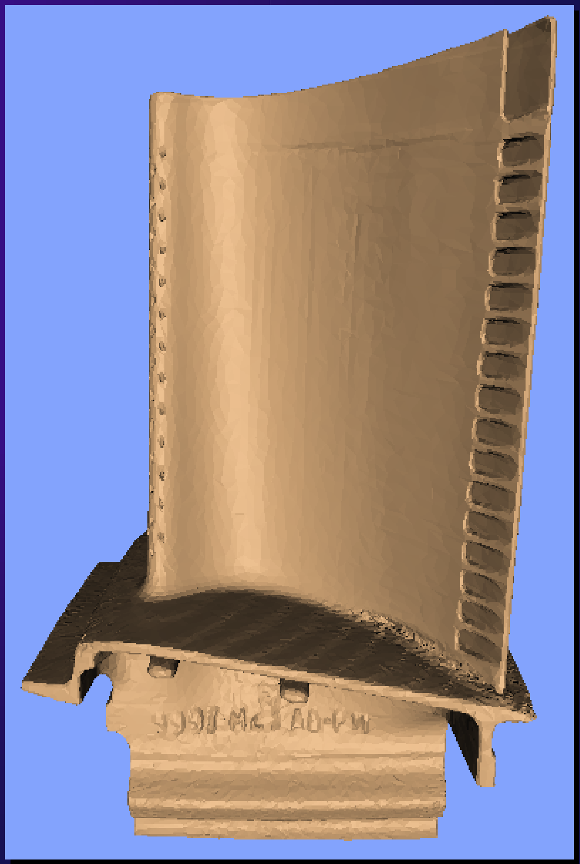
Need to handle extremely large data sets

- precise scans on the order of 10^9 triangles
- this is where simplification is needed the most
- even at 10^6 triangles, many algorithms fail

Too Large for Many Methods



1,765,388 faces



80,000 faces

Better Topological Simplification

Imperceptible holes & gaps can be removed

- most methods do this only implicitly

Few if any methods provide good control

- when exactly are holes removed?
- will holes above a certain size be preserved?

Requires a better understanding of the model

- when to simplify geometry vs. topology
- seems to benefit from more volumetric approach

Better Performance Analysis

Better criteria for evaluating similarity

- image-based metrics more appropriate for display
- metrics which accurately account for attributes

Most analysis has been case-based

- measure/compare performance on single data set

More thorough analysis is required

- theoretical analysis of quality [Heckbert-Garland 99]
- provably good approximations possible?

Higher Quality Approximations

Poor performance at extreme reduction levels

- algorithms do much worse than humans
- perhaps because all transform original into result

Simple iterative method is quite short-sighted

- only look one step ahead and never reconsider
- many consider only the local effect of operation

Consider separating analysis & synthesis

- first, build multi-level knowledge of surface shape
- then proceed with simplification

Alternative Frameworks

Greedy simplification is convenient but limited

- directly produce contraction sequence
- poor choices can never be reconsidered

Other, albeit expensive, approaches possible

- should produce a single sequence of contractions
- graph partitioning builds sequence in reverse
- more explicit optimization methods

Conclusions

Substantial progress since 1992

- simplification of 3D surfaces
- multiresolution representations (PM, hierarchies)
- application of multiresolution in different areas

There remains much room for improvement

- more effective, more general simplification
- better analysis and understanding of results
- other multiresolution representations

Acknowledgements

Funded in part by

- National Science Foundation
- Schlumberger Foundation

Sample models courtesy of

- Stanford graphics lab — bunny
- Iris Development — dental mold
- Viewpoint DataLabs — dragon
- GE/KitWare — turbine blade
- Andrew Willmott — dragon radiosity solution