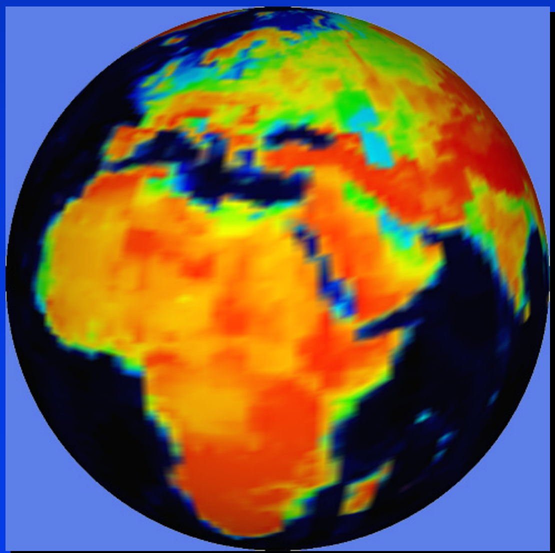
Simplifying Surfaces with Color and Texture using Quadric Error Metrics



Michael Garland
Paul S. Heckbert

Carnegie Mellon University

October 1998

Overview

Often want to simplify overly complex models

- Too much for hardware, or simply over-sampled
- Common sources: Scanning & Reconstruction

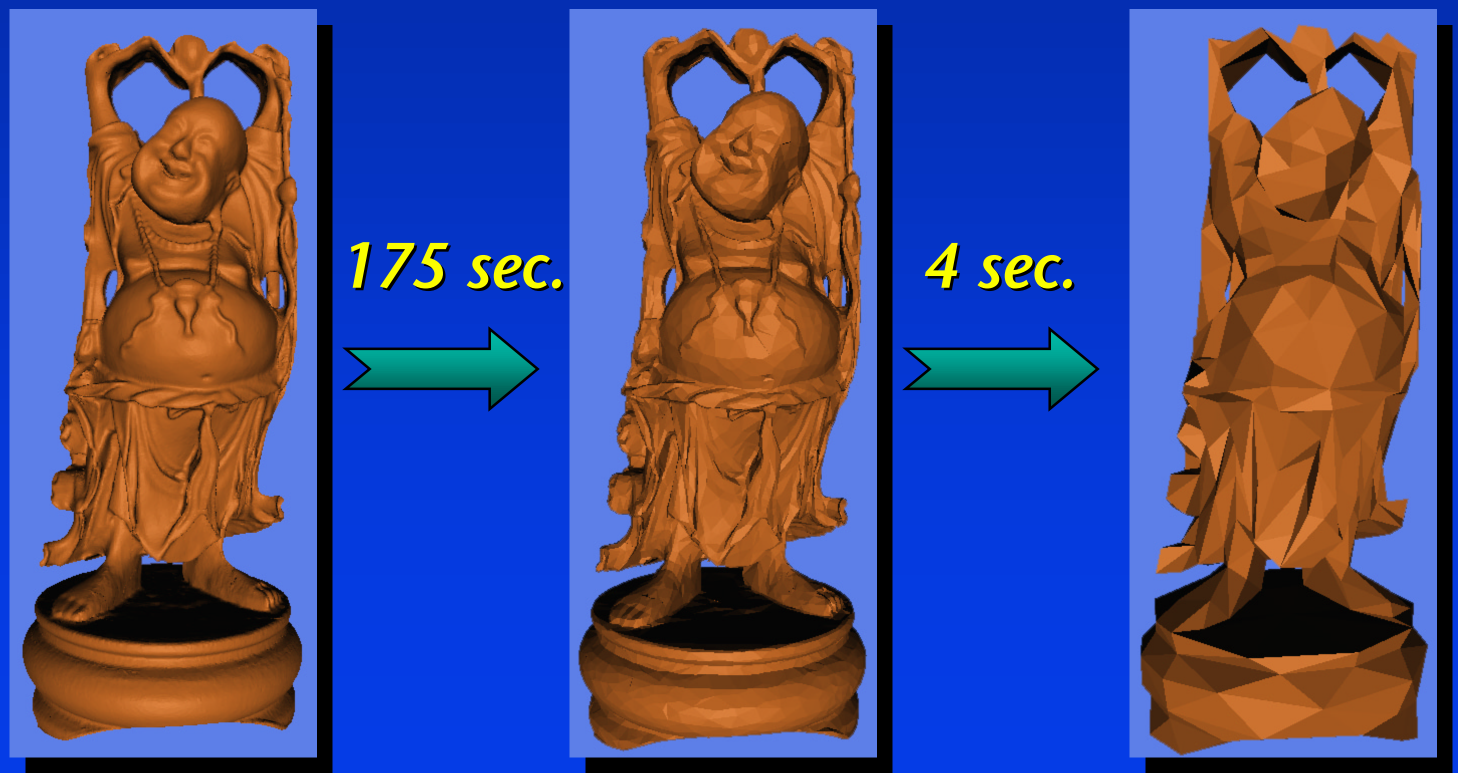
Quadric-based simplification **[SIGGRAPH 97]**

- Fast method producing quality results
- Convenient characterization of error/shape

Generalization required to handle properties

- Color, texture, surface normals, etc.

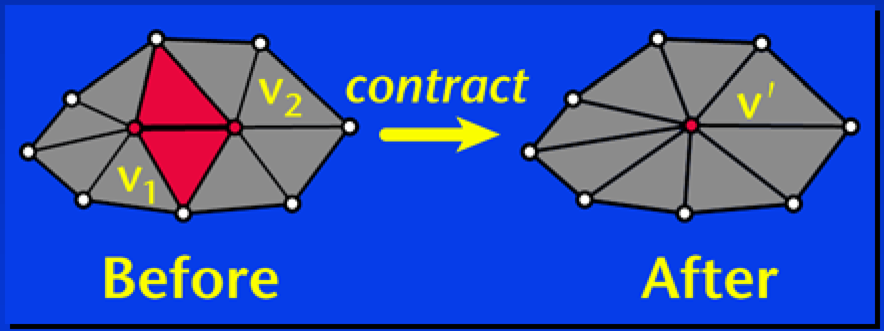
Geometric Surface Simplification



1,087,716 faces *20,000 faces* *1,000 faces*

Fundamental Quadric-Based Simplification Algorithm

Iterative edge contraction



Quadric error metric

- Each vertex has (conceptual) set of planes
- Vertex error = sum of squared distances to planes
- Combine sets when contracting vertex pair

Measuring Error with Quadrics

Given a plane, we can define a quadric Q

$$Q = (\mathbf{A}, \mathbf{b}, c)$$

measuring squared distance to given plane as

$$Q(\mathbf{v}) = \mathbf{v}^{\mathsf{T}}\mathbf{A}\mathbf{v} + 2\,\mathbf{b}^{\mathsf{T}}\mathbf{v} + c$$

$$\begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + 2 \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + c$$

Measuring Error with Quadrics

Each vertex has an associated quadric

- Measures error at vertex
- Sum quadrics when contracting pair

Sum of endpoint quadrics determines v′

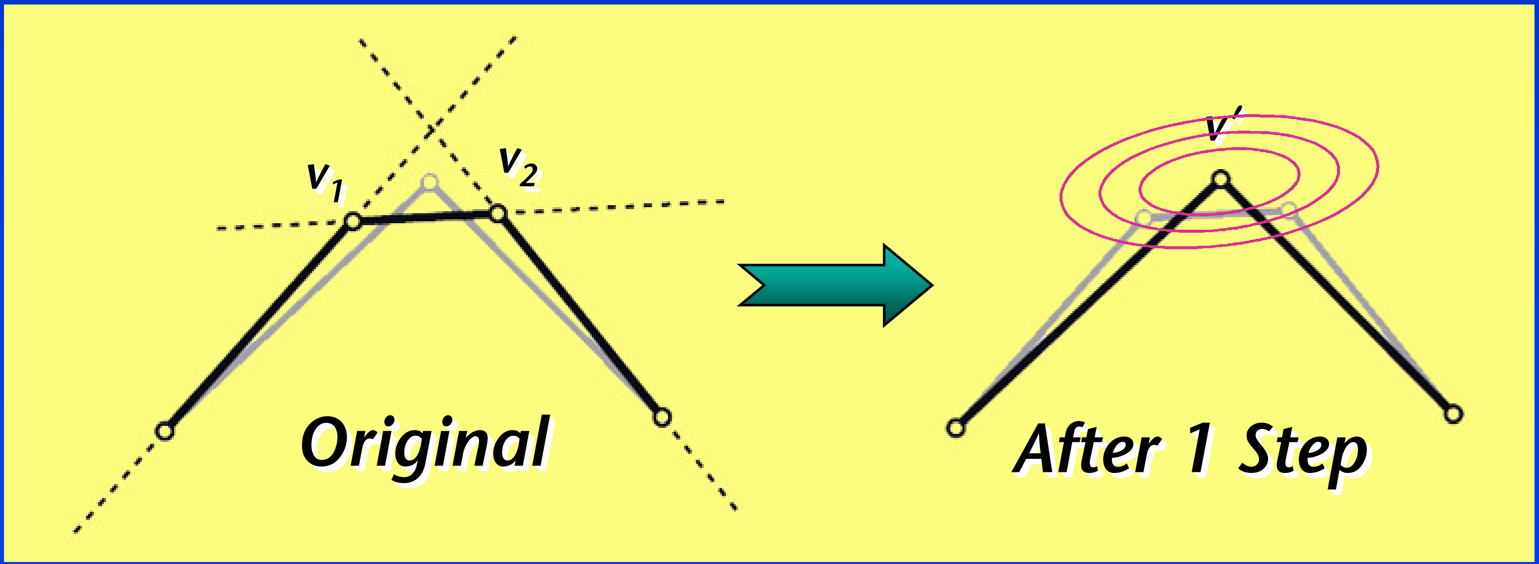
- Fixed placement: select v_1 or v_2
- Optimal placement: choose v' minimizing $Q(v')$

$\mathbf{v}' = -\mathbf{A}^{-1}\mathbf{b}$ is optimal vertex

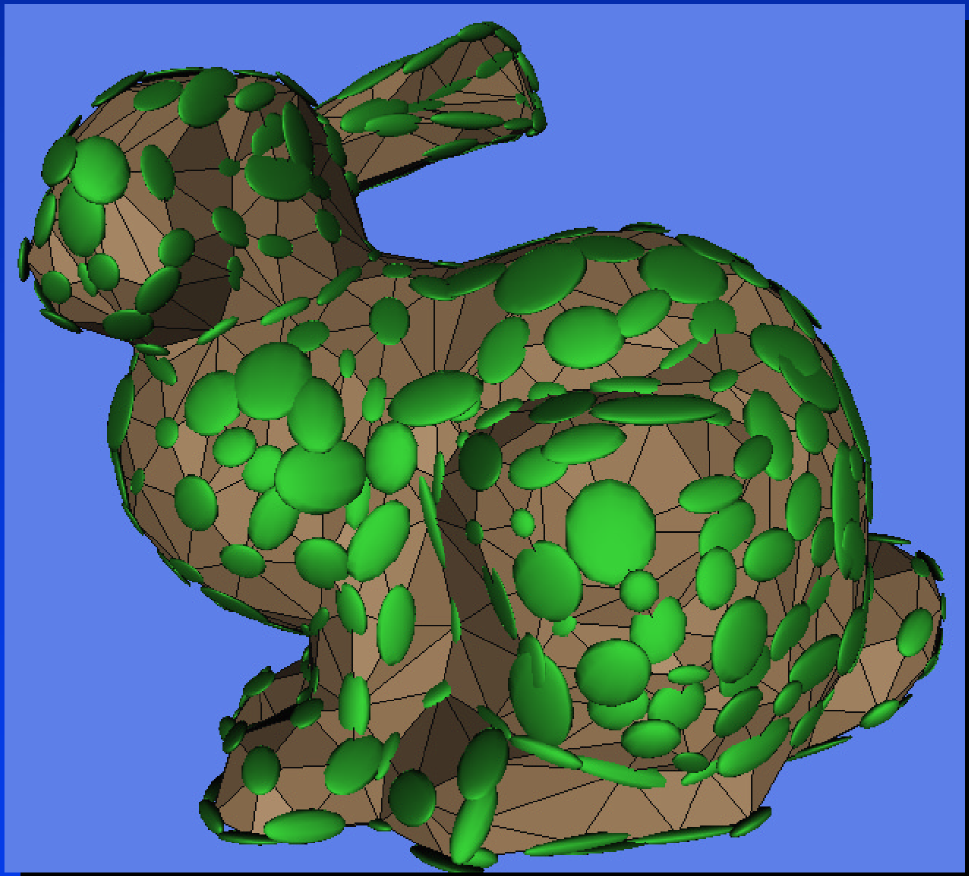
A Simple Example: Contraction & "Planes" in 2D

Lines defined by neighboring segments

- Determine position of new vertex
- Error isocontours shown on right



Visualizing Quadrics in 3-D



Quadric isosurfaces

- Are ellipsoids (maybe degenerate)
- Characterize shape
- Stretch in least curved directions
- Eigenvalues proportional to principal curvatures

Quadrics Give Good Results, But Ignores Attributes

Fundamental algorithm works well

- Simple to implement & simplification is fast
- High quality approximations
- Quadrics record useful information about shape

But many models have additional properties

- Color, texture, normals, etc.
- Need to simplify these as well

Gouraud Shaded Surface: Single RGB Value Per Vertex

Surface geometry

Radiosity solution

Surface Properties as Vertex Attributes

Each vertex has a set of properties

- Each property has *one* unique value per vertex
- Attributes are linearly interpolated over faces
- Primary example: one RGB color per vertex

Can't treat geometry & color separately

- Position and color are correlated
- Optimal position may lie off the surface
- Must synthesize new color for this position

Vertex Attributes Become Added Dimensions

Treat each vertex as a 6-vector [x y z r g b]

- Assume this 6-D space is Euclidean
  - *Of course, color space is only roughly Euclidean*
- Scale *xyz* space to unit cube for consistency

Triangle determines a 2-plane in 6-D space

- Can measure squared distance to this plane
- Distance along all perpendicular directions
  - *Generalized Pythagorean Theorem*

Generalized Quadric Metric

Squared distance to 2-plane has same form:

$$Q(\mathbf{v}) = \mathbf{v}^{\mathsf{T}}\mathbf{A}\mathbf{v} + 2\mathbf{b}^{\mathsf{T}}\mathbf{v} + c$$

- **A**: 6x6 matrix **v**,**b**: 6-vectors *c*: scalar (for RGB)
- Underlying algorithm remains the same

May want to selectively weight channels

- Relative importance of space & color
- Relative importance of red & green

Generalized Quadric Metric

Common property types

	Vertex	*Dimension*
• Color	[x y z r g b]	6x6 quadrics
• Texture	[x y z s t]	5x5 quadrics
• Normal	[x y z u v w]	6x6 quadrics
• Color+Normal	[x y z r g b u v w]	9x9 quadrics

Simplifying the Dragon Model

50,761 faces

20 sec.

10,000 faces

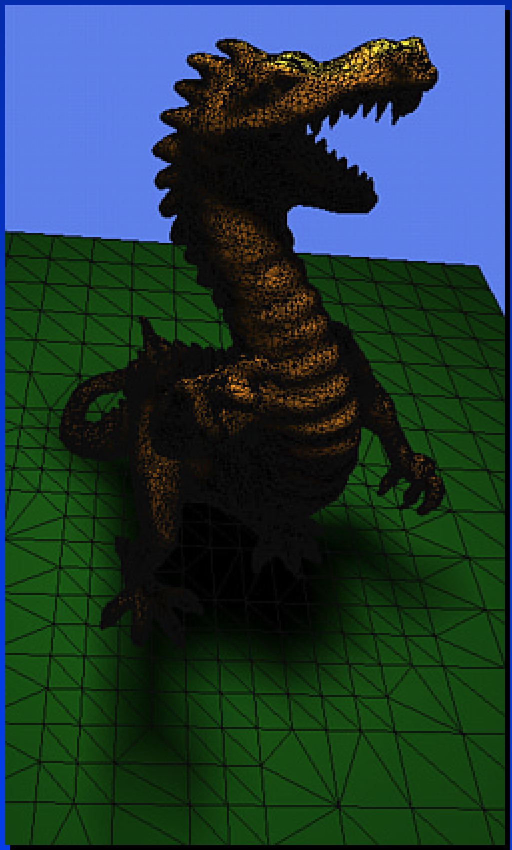
Simplifying the Dragon Model

50,761 faces

23 sec.

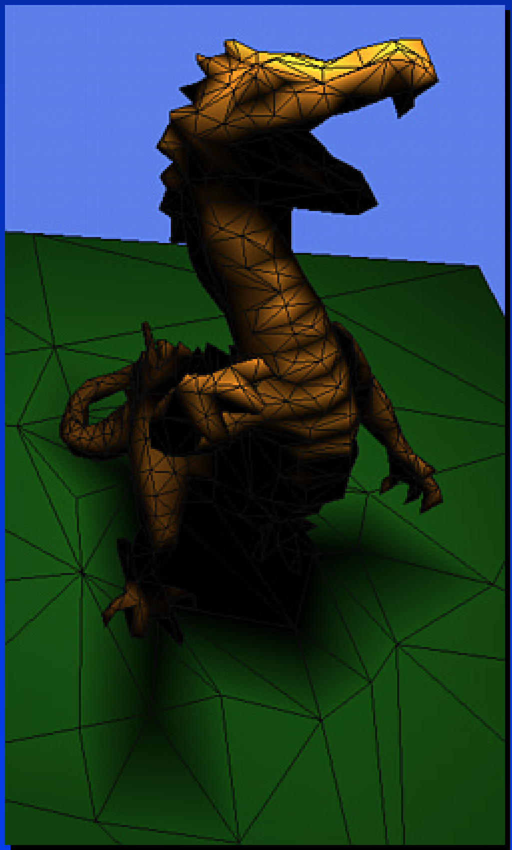
1,500 faces

Simplified Dragon Mesh



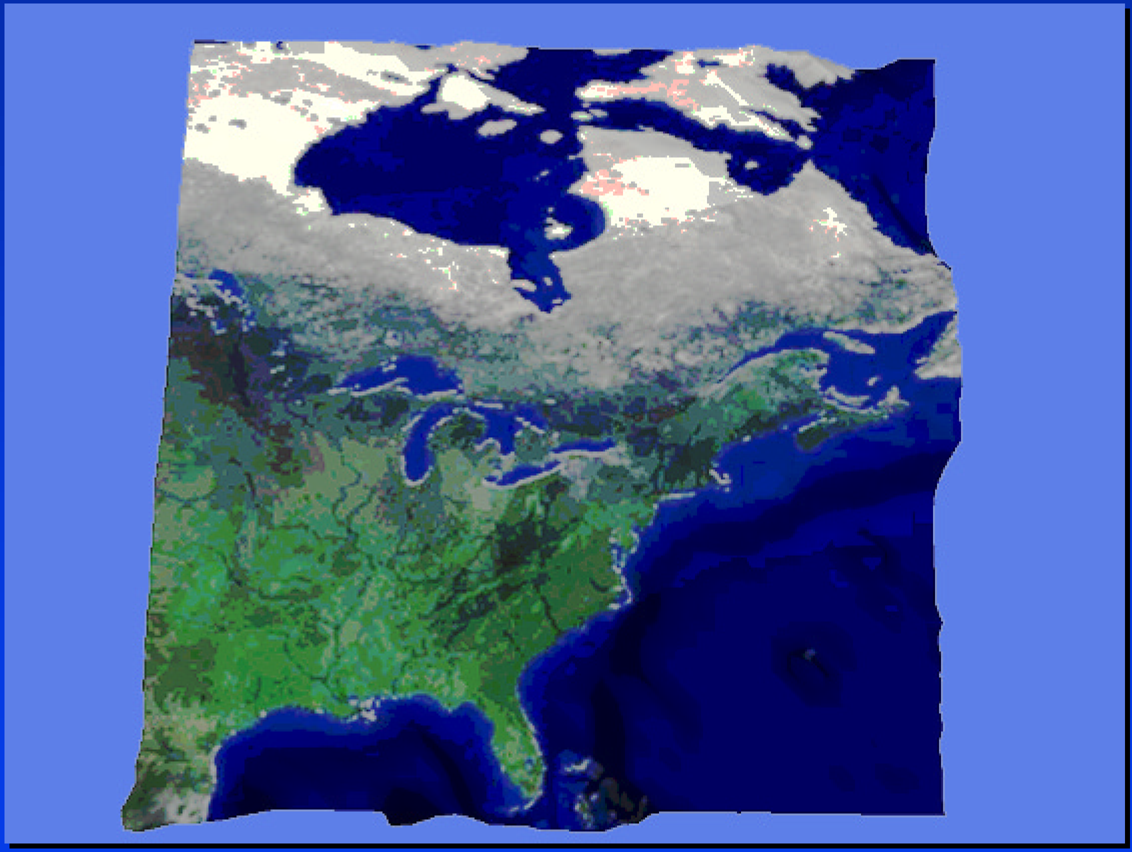
50,761 faces

23 sec.

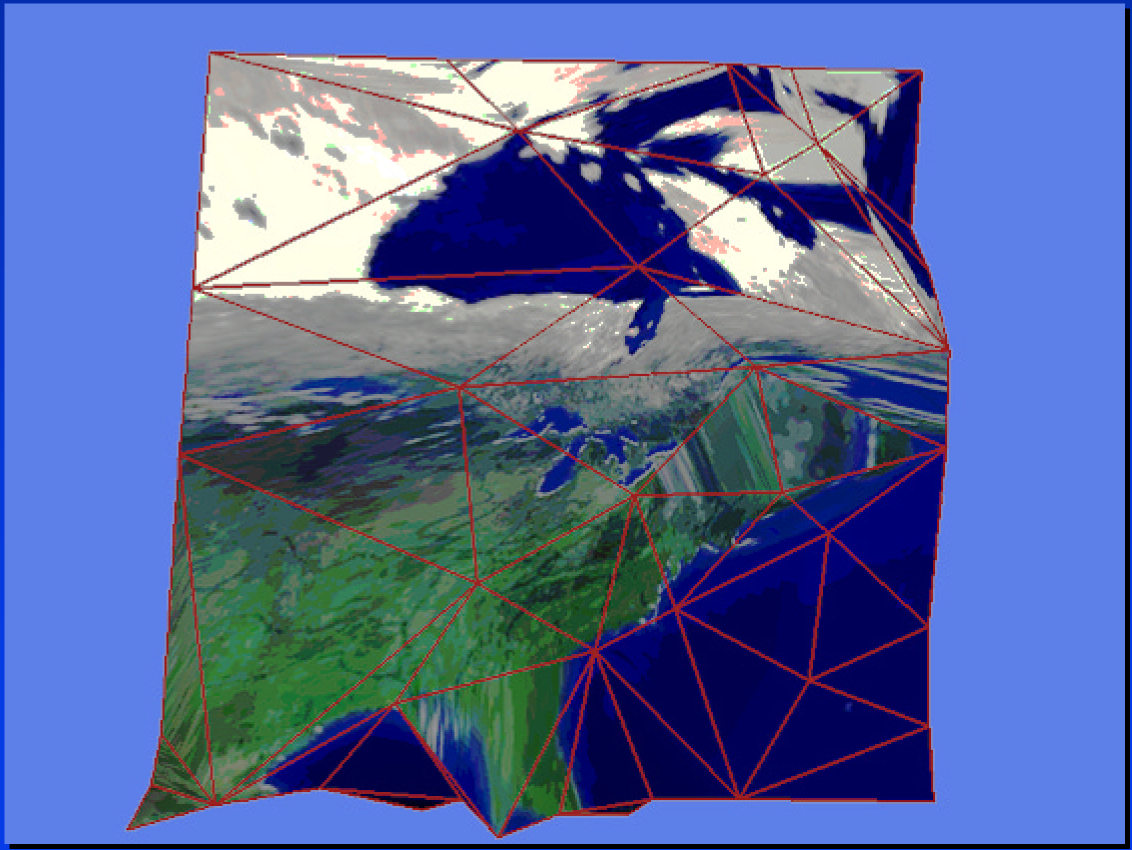


1,500 faces

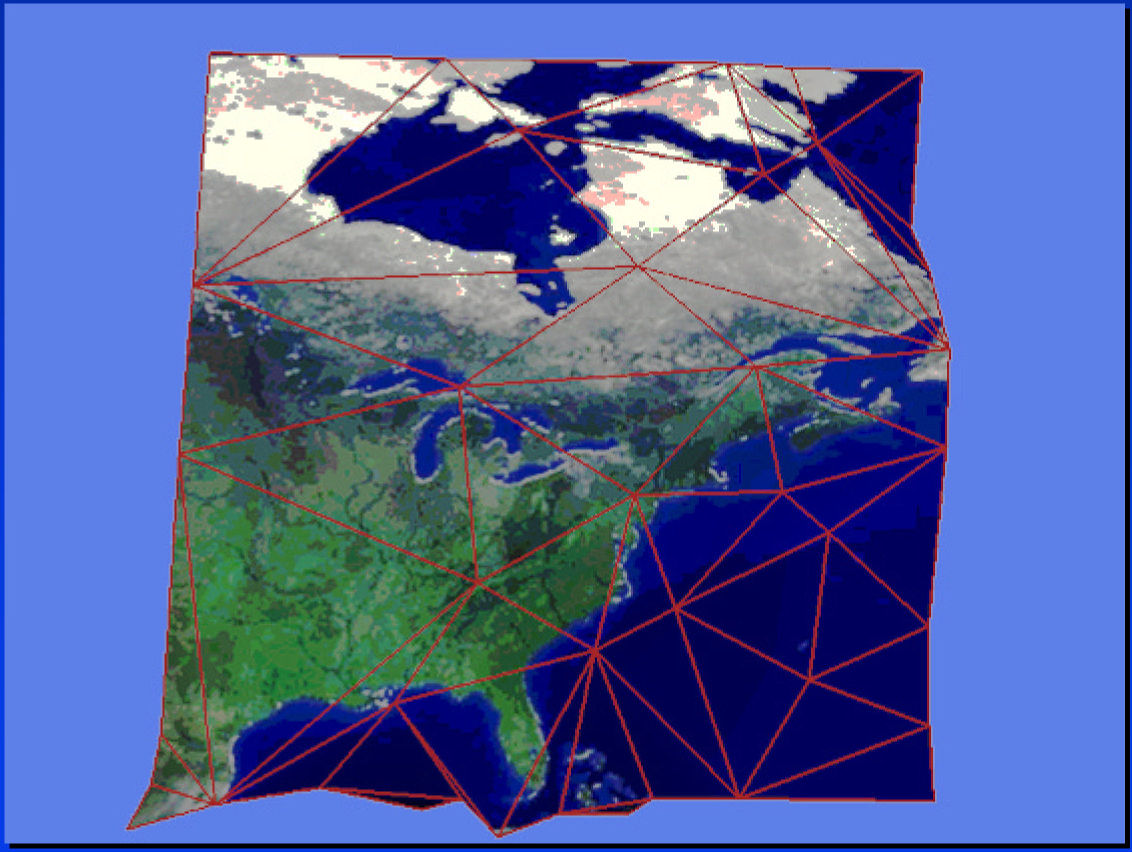
A Sample Textured Surface



Simplifying Geometry Only: Same Texture Coordinates



Simplifying with *xyzst* Quadrics; New Texture Coordinates



Related Work

Fairly few algorithms address attributes

- Re-sampling height field attributes
- Mesh optimization methods [Hoppe 96]
- Attribute map preservation [Cohen et al 98]

In comparison, our algorithm is

- Not limited to simple height field re-sampling
- Faster than other general algorithms
- Still capable of producing quality results

Summary

Generalized quadric metric

- Handles surfaces with per vertex attributes
- Rapidly produces quality approximations

Future work

- Handling attribute discontinuities
- Other attribute types
- Attribute characteristics necessary for success

Further Details Online

Online papers,

Sample models,

Experimental code

http://www.cs.cmu.edu/~garland/quadrics/