Optimizing Mesh Compression

Presented By: Andrew Clements

Outline

- Introduction
- Background
- Mesh Optimization
- Bounding Volumes
  - Spheres
  - Quadrics
- Results & Discussion

Introduction

- Goal: Reduce number of bits used to encode geometry in a mesh
- Idea: Optimize mesh for uniformity
- How: Perturb vertices without introducing significant error
  - Use quadrics, sphere as bounding volumes
- Evaluation: Use Metro to compare L2 error

Background

- Lee uses bounding cubes
- Only attempts locally driven optimization
- Conclusion: does not improve the bit rate
- Global optimization may be more promising

Mesh Optimization

- Want triangles as equilateral as possible
  - Minimize an objective function, subject to constraints
- Linear programming techniques are known for optimizing 2D planar meshes
  - can maximize edge lengths, area, angles,…
- No previously known techniques in 3D to approach the globally optimal solution

Laplacian Smoothing

- Approximate the globally optimal solution by using Laplacian smoothing
  - No guarantee on optimality
- $R = (1 - \lambda) + \lambda C$
  - where $C_{ab} = \begin{cases} 1/d_a & \text{if (a, b) is an edge, } d_a \text{ the valence of i} \\ 0 & \text{otherwise} \end{cases}$
- Using $\lambda = 0.5$ will guarantee convergence

Bounding Volumes

- Define a region where a vertex may move
  - Spheres with given radii
  - Quadrics with given error tolerance ε

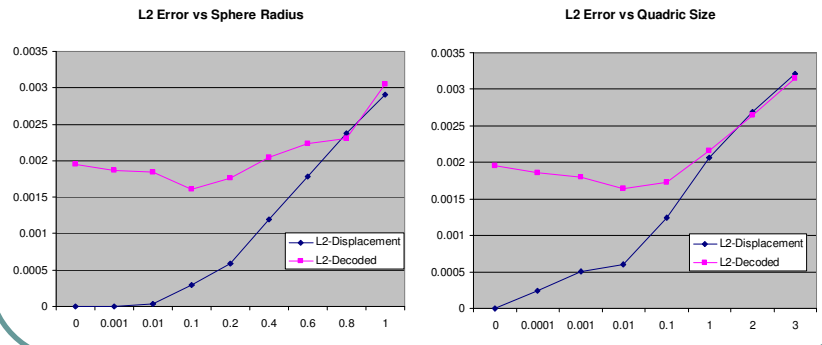
Represented as: $ax^2 + 2bxy + 2cxz + 2dx + ey^2 + 2fyz + 2gy + hz^2 + 2iz + j = \varepsilon$

Or $Q(\mathbf{v}) = \mathbf{v}^T \begin{bmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{bmatrix} \mathbf{v} = \varepsilon, \quad \text{where } \mathbf{v} = [x\ y\ z\ 1]^T$

- If as a result of Laplacian smoothing, the vertex location is outside the volume, move the vertex back inside the volume

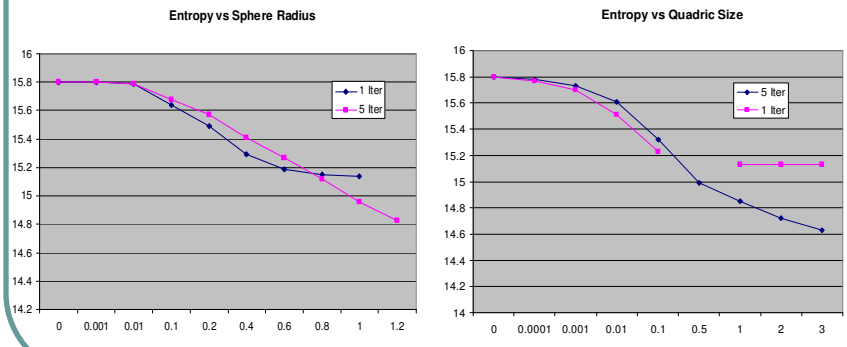
Results – L2 Error

- Graph of L2 errors using Metro on Egypt model (1 smoothing, using A.A.)



Results - Entropy

- Graph of entropy amount on Egypt model



Discussion

- Pure gains in entropy encoding are limited to 0.5 bits / vertex
- Gains level off after a certain point
  - Need more smoothing iterations to continue gains
- Visual differences in output meshes are negligible