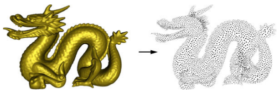
Efficient Simplification of Point-sampled Surfaces



Point-Based Computer Graphics Mark Pauly 1

Overview

- Introduction
- Local surface analysis
- Simplification methods
- Error measurement
- Comparison

Point-Based Computer Graphics Mark Pauly 2

Introduction

- Point-based models are often sampled very densely
- Many applications require coarser approximations, e.g. for efficient
  - Storage
  - Transmission
  - Processing
  - Rendering

⇨ we need simplification methods for reducing the complexity of point-based surfaces

Point-Based Computer Graphics Mark Pauly 3

Introduction

- We transfer different simplification methods from triangle meshes to point clouds:
  - Incremental clustering
  - Hierarchical clustering
  - Iterative simplification
  - Particle simulation
- Depending on the intended use, each method has its pros and cons (see comparison)

Point-Based Computer Graphics Mark Pauly 4

Local Surface Analysis

- Cloud of point samples describes underlying (manifold) surface
- We need:
  - mechanisms for locally approximating the surface ⇨ MLS approach
  - fast estimation of tangent plane and curvature ⇨ principal component analysis of local neighborhood

Point-Based Computer Graphics Mark Pauly 5

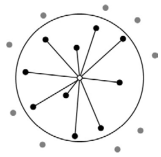
Neighborhood

- No explicit connectivity between samples (as with triangle meshes)
- Replace geodesic proximity with spatial proximity (requires sufficiently high sampling density!)
- Compute neighborhood according to Euclidean distance

Point-Based Computer Graphics Mark Pauly 6

Neighborhood

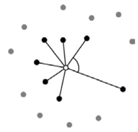
- k-nearest neighbors



  - can be quickly computed using spatial data-structures (e.g. kd-tree, octree, bsp-tree)
  - requires isotropic point distribution

Neighborhood

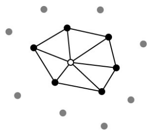
- Improvement: angle criterion (Linsen)



  - project points onto tangent plane
  - sort neighbors according to angle
  - include more points if angle between subsequent points is above some threshold

Neighborhood

- Local Delaunay triangulation (Floater)



  - project points into tangent plane
  - compute local Voronoi diagram

Covariance Analysis

- Covariance matrix of local neighborhood N:

$$\mathbf{C} = \begin{bmatrix} \mathbf{p}_{i_1} - \bar{\mathbf{p}} \\ \cdots \\ \mathbf{p}_{i_n} - \bar{\mathbf{p}} \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{p}_{i_1} - \bar{\mathbf{p}} \\ \cdots \\ \mathbf{p}_{i_n} - \bar{\mathbf{p}} \end{bmatrix}, \qquad i_j \in N$$

- with centroid $\bar{\mathbf{p}} = \frac{1}{|N|} \sum_{i \in N} \mathbf{p}_i$

Covariance Analysis

- Consider the eigenproblem:

$$\mathbf{C} \cdot \mathbf{v}_l = \lambda_l \cdot \mathbf{v}_l, \qquad l \in \{0,1,2\}$$

- C is a 3x3, positive semi-definite matrix
  - ⇨ All eigenvalues are real-valued
  - ⇨ The eigenvector with smallest eigenvalue defines the least-squares plane through the points in the neighborhood, i.e. approximates the surface normal

Covariance Analysis

- The total variation is given as:

$$\sum_{i \in N} |\mathbf{p}_i - \bar{\mathbf{p}}|^2 = \lambda_0 + \lambda_1 + \lambda_2$$

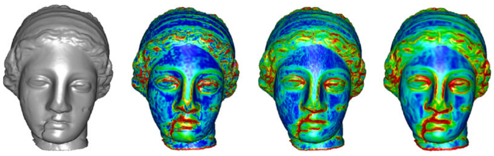
- We define surface variation as:

$$\sigma_n(\mathbf{p}) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \qquad \lambda_0 \le \lambda_1 \le \lambda_2$$

  - measures the fraction of variation along the surface normal, i.e. quantifies how strong the surface deviates from the tangent plane ⇨ estimate for curvature

Covariance Analysis

- Comparison with curvature:



original | mean curvature | variation n=20 | variation n=50

Surface Simplification

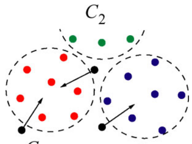
- Incremental clustering
- Hierarchical clustering
- Iterative simplification
- Particle simulation

Incremental Clustering

- Clustering by region-growing:
  - Start with random seed point
  - Successively add nearest points to cluster until cluster reaches maximum size
  - Choose new seed from remaining points
- Growth of clusters can also be bounded by surface variation
  - ⇨ Curvature adaptive clustering

Incremental Clustering

- Incremental growth leads to internal fragmentation
  - ⇨ assign stray samples to closest cluster

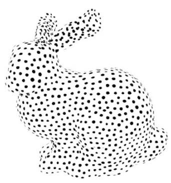


- Note: this can increase maximum size and variation bounds!

Incremental Clustering

- Replace each cluster by its centroid

original model with color-coded clusters (34,384 points)



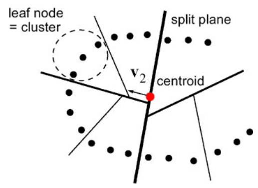
simplified model (1,000 points)

Hierarchical Clustering

- Top-down approach using binary space partition:
- Split the point cloud if:
  - Size is larger than user-specified maximum or
  - Surface variation is above maximum threshold
- Split plane defined by centroid and axis of greatest variation (= eigenvector of covariance matrix with largest associated eigenvector)
- Leaf nodes of the tree correspond to clusters

Hierarchical Clustering

- 2D example



Hierarchical Clustering

- Adaptive clustering

original model with color-coded clusters (34,384 points)

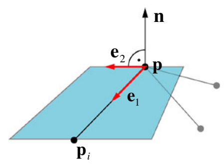
simplified model (1,000 points)

Iterative Simplification

- Iteratively contracts point pairs
  - ⇨ Each contraction reduces the number of points by one
- Contractions are arranged in priority queue according to quadric error metric (Garland and Heckbert)
- Quadric measures cost of contraction and determines optimal position for contracted sample
- Equivalent to QSlim except for definition of approximating planes

Iterative Simplification

- Quadric measures the squared distance to a set of planes defined over *edges* of neighborhood
  - plane spanned by vectors $\mathbf{e}_1 = \mathbf{p}_i - \mathbf{p}$ and $\mathbf{e}_2 = \mathbf{e}_1 \times \mathbf{n}$



Iterative Simplification

original model (187,664 points)

simplified model (1,000 points)

remaining point pair contraction candidates

Particle Simulation

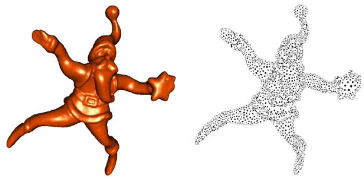
- Resample surface by distributing particles on the surface
- Particles move on surface according to inter-particle repelling forces
- Particle relaxation terminates when equilibrium is reached (requires damping)
- Can also be used for up-sampling!

Particle Simulation

- Initialization
  - randomly spread particles
- Repulsion
  - linear repulsion force $F_i(\mathbf{p}) = k(r - \|\mathbf{p} - \mathbf{p}_i\|) \cdot (\mathbf{p} - \mathbf{p}_i)$
  - ⇨ only need to consider neighborhood of radius r
- Projection
  - keep particles on surface by projecting onto tangent plane of closest point
  - apply full MLS projection at end of simulation

Particle Simulation

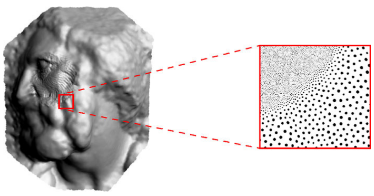
- Adaptive simulation
  - Adjust repulsion radius according to surface variation
    ⇨ more samples in regions of high variation



original model (75,781 points) — simplified model (6,000 points)

Particle Simulation

- User-controlled simulation
  - Adjust repulsion radius according to user input

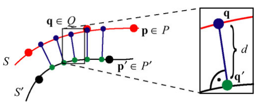


Measuring Error

- Measure the distance between two point-sampled surfaces using a sampling approach
- Maximum error: $\Delta_{\max}(S, S') = \max_{\mathbf{q} \in Q} d(\mathbf{q}, S')$
  - ⇨ Two-sided Hausdorff distance
- Mean error: $\Delta_{\text{avg}}(S, S') = \frac{1}{|Q|} \sum_{\mathbf{q} \in Q} d(\mathbf{q}, S')$
  - ⇨ Area-weighted integral of point-to-surface distances
- Q is an up-sampled version of the point cloud that describes the surface S

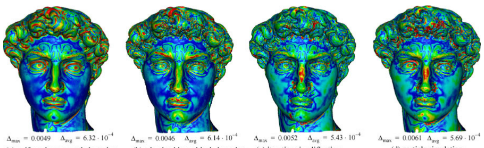
Measuring Error

- $d(\mathbf{q}, S')$ measures the distance of point $\mathbf{q}$ to surface S' using the MLS projection operator with linear basis functions



Comparison

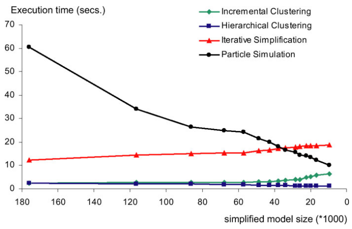
- Error estimate for Michelangelo's David simplified from 2,000,000 points to 5,000 points



$\Delta_{\max} = 0.0049$ $\Delta_{\text{avg}} = 6.32 \cdot 10^{-4}$ (a) uniform incremental clustering; $\Delta_{\max} = 0.0046$ $\Delta_{\text{avg}} = 6.14 \cdot 10^{-4}$ (b) adaptive hierarchical clustering; $\Delta_{\max} = 0.0052$ $\Delta_{\text{avg}} = 5.43 \cdot 10^{-4}$ (c) iterative simplification; $\Delta_{\max} = 0.0061$ $\Delta_{\text{avg}} = 5.69 \cdot 10^{-4}$ (d) particle simulation

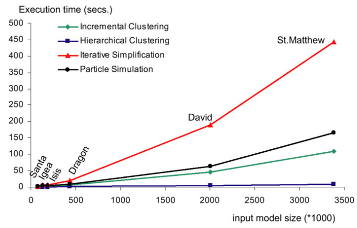
Comparison

- Execution time as a function of target model size (input: dragon, 535,545 points)



Comparison

- Execution time as a function of input model size (reduction to 1%)

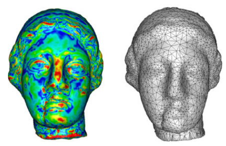


Comparison

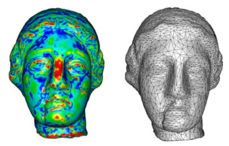
- Summary

	Efficiency	Surface Error	Control	Implementation
Incremental Clustering	+	-	-	+
Hierarchical Clustering	+	-	-	+
Iterative Simplification	-	+	o	o
Particle Simulation	o	+	+	-

Point-based vs. Mesh Simplification



point-based simplification with subsequent mesh reconstruction



mesh reconstruction with subsequent mesh simplification (QSlim)

⇨ point-based simplification saves an expensive surface reconstruction on the dense point cloud!

References

- Pauly, Gross: *Efficient Simplification of Point-sampled Surfaces*, IEEE Visualization 2002
- Shaffer, Garland: *Efficient Adaptive Simplification of Massive Meshes*, IEEE Visualization 2001
- Garland, Heckbert: *Surface Simplification using Quadric Error Metrics*, SIGGRAPH 1997
- Turk: *Re-Tiling Polygonal Surfaces*, SIGGRAPH 1992
- Alexa et al. *Point Set Surfaces*, IEEE Visualization 2001