Shape Manipulation

Model Simplification
- Problem
  - Models are too big to:
    - Display interactively
    - Display at all
- Typical domains
  - 3D scanning – very verbose
  - Visualization of scientific data
  - Century man-year CAD/CAM efforts: plants, ships and planes
  - Results of procedural models
  - Tuning for interactive environments
- Basic approach
  - Define a primitive simplification *operator*: e.g. edge collapse
  - Define a *measure* of this operator's effect on model quality
  - Use a greedy algorithm:
    - Initialize the a queue of operations, sorted by measure
    - Repeat
      - Apply operator at front of queue
      - Adjust affected queue entries
    - Until target [error, # polys] reached
- Brief taxonomy based on quantization
  - Pairwise merge algorithms
    - Operators are defined as merges of surface patches
    - Common example: edge collapse (Hoppe, Garland)
    - Usually best quality, fairly slow
  - Pruning algorithms
    - Operators are defined as removals of a patch at a time
    - Common example: decimations (Schroeder)
    - Good quality, slower
  - Splitting
    - Operators are splitting of coarse patches
    - Simplification in fact becomes refinement
    - Common example: rsimp (Watson)
    - Decent quality, fast
  - Product codes
    - Patches are defined by a quantization of the model space
    - Not greedy: linear time
    - Common example: vertex clustering (Rossignac)
    - Poor quality, very fast
- Critique
  - +: works well and used widely
  - -: Still difficult to simply models out of core
  - -: Still difficult to handle the relationship of geometry and attributes
- Related fields
  - Level of detail (LOD)
    - How to use simplified models in interactive display?
  - Measuring visual fidelity
    - Which primitive simplification preserves appearance?

Surface fitting
- Problem
  - We have a cloud of points from a scanning apparatus
  - How do we turn these points into a surface model?
- Basic example (Turk & Levoy)
  - Make multiple partial scans

- Fit surface to each scan (fairly simple)
- Repeat
  - Choose two partial scans
  - Align them to one another using iterated closest points
  - Join the surfaces topologically
  - Tune the local geometry with comparisons to point cloud
- Until all partial scans are joined

Critique

- +: widely used
- -: didn't address reflectance & surface properties
- -: what about parametric surfaces?

Shape comparison

Problem

- How to sort & and query for shapes?

Basic example (Osada, Funkhouser et al)

- Define a *shape function*
  - Should be invariant under rigid transforms, small detail variation
  - E.g. distance between two random model points
  - E.g. volume of tetrahedron between four model points
- Use it to create a *shape distribution*
  - Make a lot of random samples of shape function
  - Form a histogram
  - Fit a piecewise linear probability distribution function
  - Result, a vector with N integers
- Compare shape distributions
  - Use existing function comparison approaches
  - They use Minkowski sum:
    - Shape distr A, B
    - sum((A-B)^N)^(1/N)

Critique

- +: Good for rough classification
- -: Not good at details

Issues

- What is a good shape function?
- How to query?