CITS4241 Visualisation
Lecture 15

Decimation

Large datasets…

- **The Problem**
  - Modern digital measuring devices, *eg*
    - **Laser scanners**
    - **Steered beam ultrasound**
    - **Satellite radar**

    generate datasets of large sizes
  - Sensor Accuracy
    - **Music CD** **16bits @ 44kHz**
    - **Digital Camera**
      - Spatial 640 x 480 pixels
      - Bits per pixel 8
    - **Monitor screen** **1200 x 1000 pixels**
    - **Analogue to Digital Converter**
      - Fast Flash - 8 bits @ 1GHz
      - General 12 bits @ 1MHz
      - High accuracy 16+ bits @ 100kHz

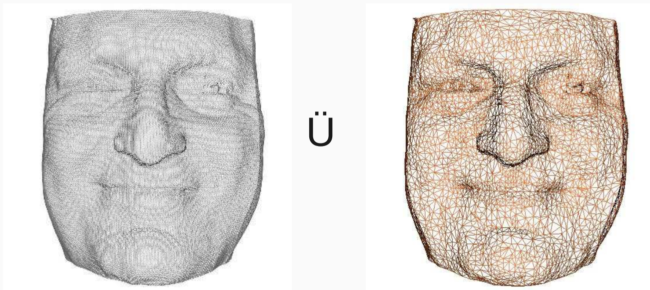
Large datasets…

- **3D volumetric data**
  - 512 x 512 pixels per slice
  - (say) 100 slices

    ð **511 x 511 x 99 > 25x10^6 voxels!**
- **Visualise this data by contouring …using marching cubes**
- **Contour surface will consist of what shape?**
  - Average number of triangles per voxel?
    - **4?**
  - How many triangles likely?
    - **4 x 25 x 10^6 = 10^8 triangles! (worst case)**

Aim

- **Reduce the number of triangles …**



- Decimation **compresses the polygon database**
  - Compressions of 10:1 are possible

Aim - Polygon Reduction

- **Vertex deletion (triangle decimation)**
  - Based on some measurable criterion
- **No new vertices introduced**
- **"Unimportant" vertices deleted**
- **"Holes" created must be filled**

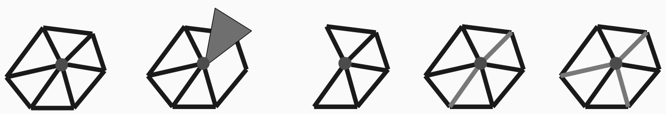
Decimation Algorithm

For each point in a triangle mesh ...

1. **Classify the point**
   - Characterize the local geometry and topology for the point
2. **Apply a decimation criterion to determine whether a point can be deleted**
3. **After deleting the point in Step 2, retriangulate the resulting hole**

Decimation Algorithm

1. **Classify point**

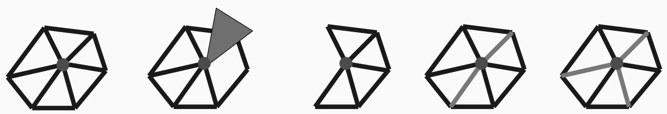


Simple Complex Boundary Interior Edge Corner

- **A point is simple if**
  - it is surrounded by complete cycle of triangles and
  - the edge that uses the point is shared by <u>exactly</u> 2 triangles

Decimation Algorithm

1. **Classify point**

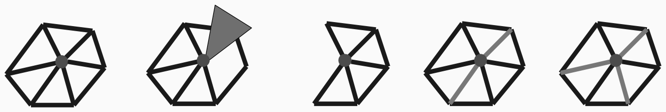


Simple Complex Boundary Interior Edge Corner

- **A point is complex if**
  - it is used by a triangle not in the cycle of triangles, or
  - the edge that uses the point is not used by 2 triangles

Decimation Algorithm

1. **Classify point**

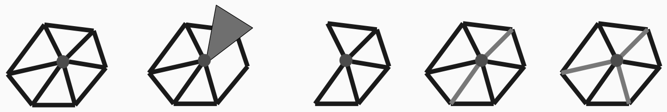


Simple **Complex** **Boundary** **Interior Edge** **Corner**

- **A point is said to be a boundary point if**
  - it is on the boundary of a mesh, i.e., within a semicycle of triangles

Decimation Algorithm

1. **Classify point**



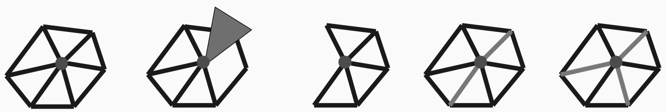
Simple **Complex** **Boundary** **Interior Edge** **Corner**

- **A simple point can be further classified as an interior edge point if**
  - it is used by 2 feature edges

    Note: if the surface normals of two adjacent triangles sharing the point make an angle larger than a specified **feature angle** then the common edge is called a **feature edge**

Decimation Algorithm

1. **Classify point**



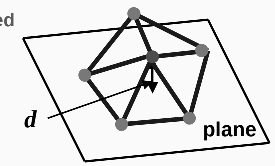
Simple **Complex** **Boundary** **Interior Edge** **Corner**

- **A simple point can be further classified as a corner point if**
  - it is used by > 2 feature edges

    Note: if the surface normals of two adjacent triangles sharing the point make an angle larger than a specified **feature angle** then the common edge is called a **feature edge**

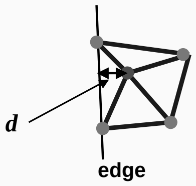
Decimation Algorithm

2. **Applying a decimation criterion to determine if a point can be deleted**
   - Estimate the error that would result by deleting the point and replacing it with another (simpler) triangulation
   - A number of error measures possible; the simplest measure is based on
     - **Distance measures of local planarity or local collinearity**
       - Is distance d < pre-defined error measure?
       - Yes à point can be deleted



Decimation Algorithm

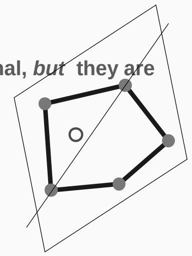
2. **Applying a decimation criterion to determine if a point can be deleted**
   - For a boundary or interior edge point…
     - it is considered to lay on an edge
     - its distance, d, from the edge is calculated
     - Is d < error criterion?
     - Yes à point can be deleted



Decimation Algorithm

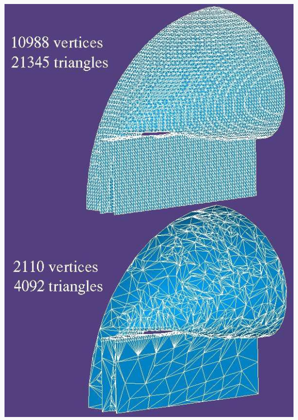
3. **'Hole' that is left by deleting a point is retriangulated**
   - A hole is defined by a loop of edges
   - Topologically, holes are two dimensional, *but* they are generally non-planar

- **Triangulation**
  - Choose an initial split plane
  - Creates two subloops
  - If all points in each subloop lie on opposite sides of the plane, then it's a valid one
  - Also, check aspect ratio à eliminate "skinny" triangles
  - Continue recursively until both subloops are triangulated



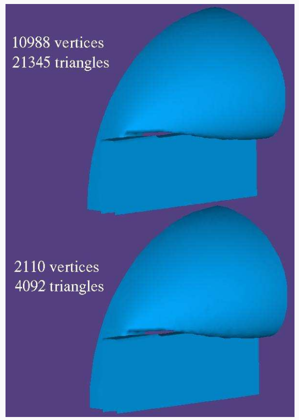
Examples

- **5:1 reduction (triangular mesh)**



Examples

- **5:1 reduction (shading)**



Advanced Decimation Techniques

- **Decimation - an active research area!**
- **Techniques**
  - Progressive Schemes
    - **Incremental transmission and reconstruction of meshes**
    - **Important for Web-based geometry visualisation**
  - Topology Modification
    - **Important for arbitrary levels of mesh reduction**

Progressive Mesh

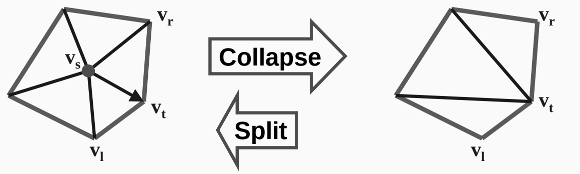
- **Series of triangle meshes, M^i, related by operations:**

$$M^n \rightarrow M^{n-1} \rightarrow \ldots \rightarrow M^1 \rightarrow M^0$$

- **M^n is the mesh at full resolution,**
- **M^0 is a simplified base mesh**
- **Critical characteristic of a progressive mesh**
  - Choose the mesh operations to be **invertible**
  - **Then operations can be applied in reverse order**

Progressive Mesh

- **Operations**
  - **Edge Collapse and Split** (due to deletion or addition of a point)



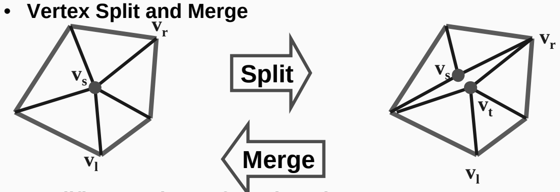
  - The operation is represented by five values:

    Edge_Collapse/Split(v_s, v_t, v_l, v_r, A)

    where v_s is the source vertex to be collapsed to the target vertex v_t; v_l and v_r are the two adjacent vertices of v_s and v_t; A represents the vertex attribute information after the operation.
  - Edge collapse occurs when we replace vertex v_s with v_t in the connectivity list of triangles that originally used vertex v_s.

Progressive Mesh Operations

- **Operations**
  - **Vertex Split and Merge**

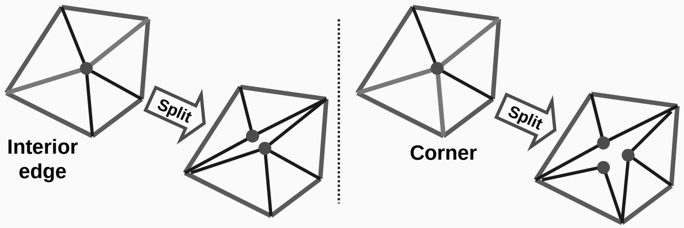


  - Modifies topology of mesh and
  - allows arbitrary levels of reduction
  - Relieves topological constraints
  - Only invoked when a valid edge collapse is not available
  - The operation can be represented with four values

    Vertex_Collapse/Split(v_s, v_t, v_l, A)

Progressive Mesh Operations

- **Operations**
  - **Vertex Split and Merge**



  - Classification of vertex determines splitting strategy
    - **Interior edge vertices: along feature edges**
    - **Corner vertices: along feature edges**

Progressive Mesh Operations

- **Allowing topology modification**
  - Arbitrary reduction
  - Whilst topology preserving reductions are used,
    - ð **Features (eg holes) remain**
      but **a circular hole may "collapse" to a triangular one**
  - Allow vertex splits
    - ð **Eventually, all features washed out**

Further reading

- **Polygon Reduction – Decimation**
  - Schroeder et al, "The Visualization Toolkit – An Object Oriented Approach to 3D Graphics", 1998, Chapter 9, pages 381 – 388