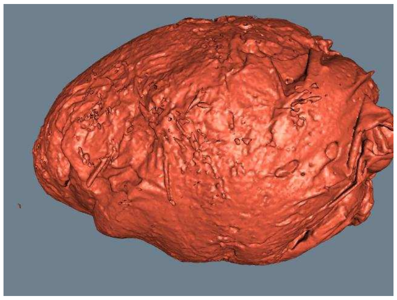
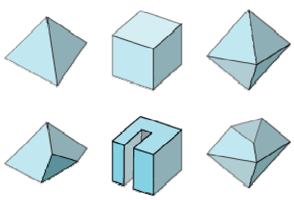
Goal



Lorensen

1

Polyhedron



2

Polyhedron

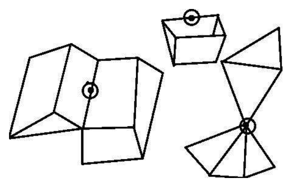
A finite collection of planar bounded convex polygonal faces such that

1. The faces intersect properly
2. The neighborhood of every point is topologically an open disk
3. The surface is connected

Euler formula (genus 0): V-E+F=2

3

Not a 2-manifold



4

Information flow

1. Data acquisition
2. Image processing
3. Surface reconstruction
4. Display

5

Marching cubes (Lorensen & Cline, 87)

- Intermediate geometric representation
- High-resolution 3D surface construction algorithm
- Create a polygonal representation of constant density surfaces from a 3D array of data
- Result can be displayed with conventional graphics rendering algorithms

6

Steps

1. Define a surface value = threshold (by user)
2. Algorithm:
   1. Find cubes intersected by the surface
   2. Examine cubes in the boundary cells and produce a set of connected polygons
   3. Calculate normals at each vertex

7

Cube

Locate the surface in a logical cube created from eight pixels

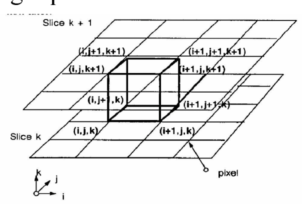


Figure 2. Marching Cube.

8

Locating the surface

1. Determine how the surface intersects the cube
   a. Assign 1 to a vertex if the data value exceeds threshold (=inside)
      Surface intersects cube edges where one vertex is out and one is in
   b. Determine the topology of the surface within a cube
   c. Find the location of the intersection
2. March to the next cube

9

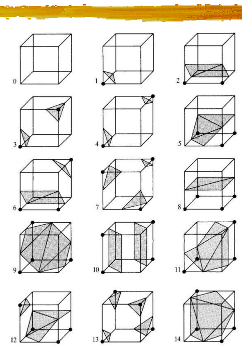
Cases

Reduce the number of cases by

1. Complementary cases
2. Rotational symmetry

10

Triangulated cubes



11

Edge table

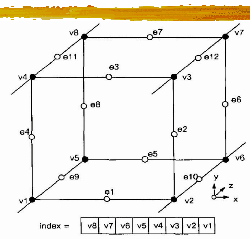


Figure 4. Cube Numbering.

8-bit index ⟶ pointer into an edge table

12

Shading

1. Calculate gradient for every cube vertex

$$G_x(i,j,k) = \frac{D(i+1,j,k) - D(i-1,j,k)}{\Delta_x}$$
$$G_y(i,j,k) = \frac{D(i,j+1,k) - D(i,j-1,k)}{\Delta_y}$$
$$G_z(i,j,k) = \frac{D(i,j,k+1) - D(i,j,k-1)}{\Delta_z}$$

2. Find normals at every triangle vertex
3. Gouraud shading

13

Summary

1. Read four slices into memory
2. Create a cube from 2 slices
3. Calculate an index by comparing to threshold
4. Lookup the list of edges in table
5. Find surface-edge intersection – interpolation
6. Calculate normal at each cube vertex
   Interpolate the normal to each triangle vertex
7. Output triangle vertices and normals

14

Enhancement

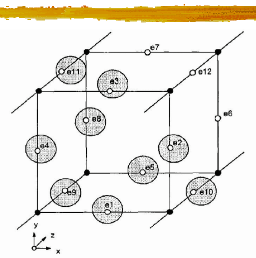


Figure 5. Coherence.

Coherence – only three new edges

15

Results

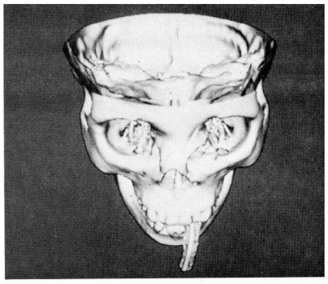


Figure 8. Bone Surface.

Results

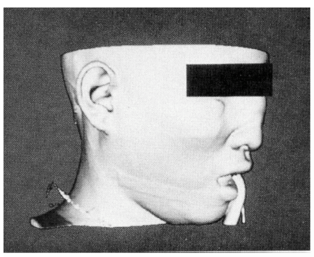


Figure 9. Soft Tissue Surface.

Results

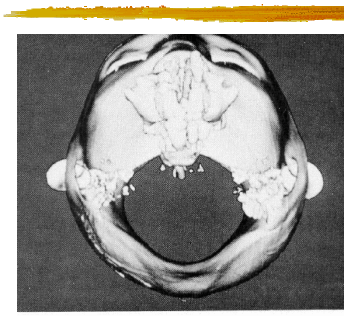


Figure 10. Soft Tissue, Top View.

Results

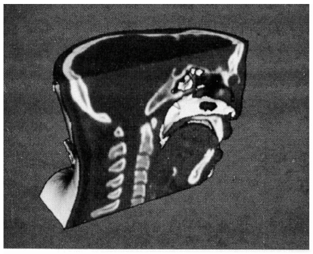


Figure 11. Sagittal Cut with Texture Mapping.

19

Results

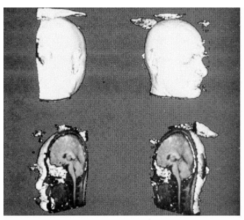
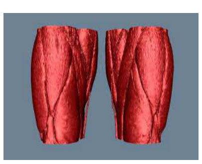


Figure 12. Rotated Sequence of Cut MR Brain.

20

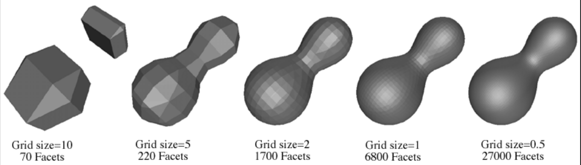
Results



Legs, visible man (Lorensen)

21

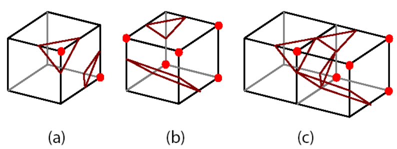
Resolution



www.mhri.edu.au/~pdb/modellig/polygonise

22

Holes



23