Lecture 22: Volume Rendering

1. Rendering or visualizing voxel-based data

- mathematical model
  - e.g., computational fluid model

- collected from real model
  - e.g., medical imaging (computed tomographic system)

- visible human project (1998)
  - e.g., 15GB voxel data set for male
  - e.g., 40 GB voxel data set for female
2. Rendering voxels in binary partitioned space

- 3D array of points which represent a 3D sampling of the reality

- A 3D sample is a binary occupancy
3. **Intermediate geometric representation techniques**
   - extract their surface and then use conventional techniques to render the surface.

   - 2D tracking to extract the contour of the slices (segmentation)

   - extract a surface between constructive contours
4. Marching cubes algorithm

(1) iso-value surface detector

- find the cubes or voxels that are intersected by the surface.

- a set of connected polygons are produced for rendering.

- actual surface is built up by fitting a polygon or polygons through each voxel that is deemed to contain a surface

- each cube is in the boundary set, its vertex is inside, outside or on the surface.
• 14 possible relationship between an iso-surface and the cube that it passes through

• iso-surface separates the surface points with other points.

• adjacent surface patches are slope (1st order $C^1$) continuous at the vertices, but not necessarily along the edge of the patch.
(2) Implementation

- match step:
  - pixel by pixel \( i(x) \)
  - tow by row \( j(y) \)
  - slice by slice \( k(z) \)

- vertex and edge numbering

- cube topology
  - single facet consists of a list of vertex indices
  - anti-clockwise (for the outward side of a facet)
e.g.,

facet data [45]
= 11, 10, 8, 8, 10, 0, 10, 1, 0, 5, 4, 9

edge data [45]
= 11, 1, 0, 4, 5, 8, 9, 10, −1, −1, −1, −1

vertex 0, 2, 3, 5 are outside the model

cube entity outside surface: 11111111 = 255
cube entity inside surface: 00000000 = 0

\[2^0 + 2^2 + 2^3 + 2^5 = 45\]