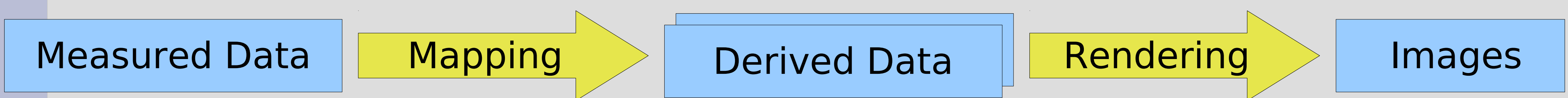
Surface Rendering

Leonid I. Dimitrov and Miloš Šrámek

Austrian Academy of Sciences

Volume Visualization by Mapping

- In general:
  - Measured densities are mapped to visual attributes (transparency, color)



- Surface rendering

  - Extra data structure: a surface model
  - Basic primitive: 2D patches (polygons)

Surface Rendering Basics

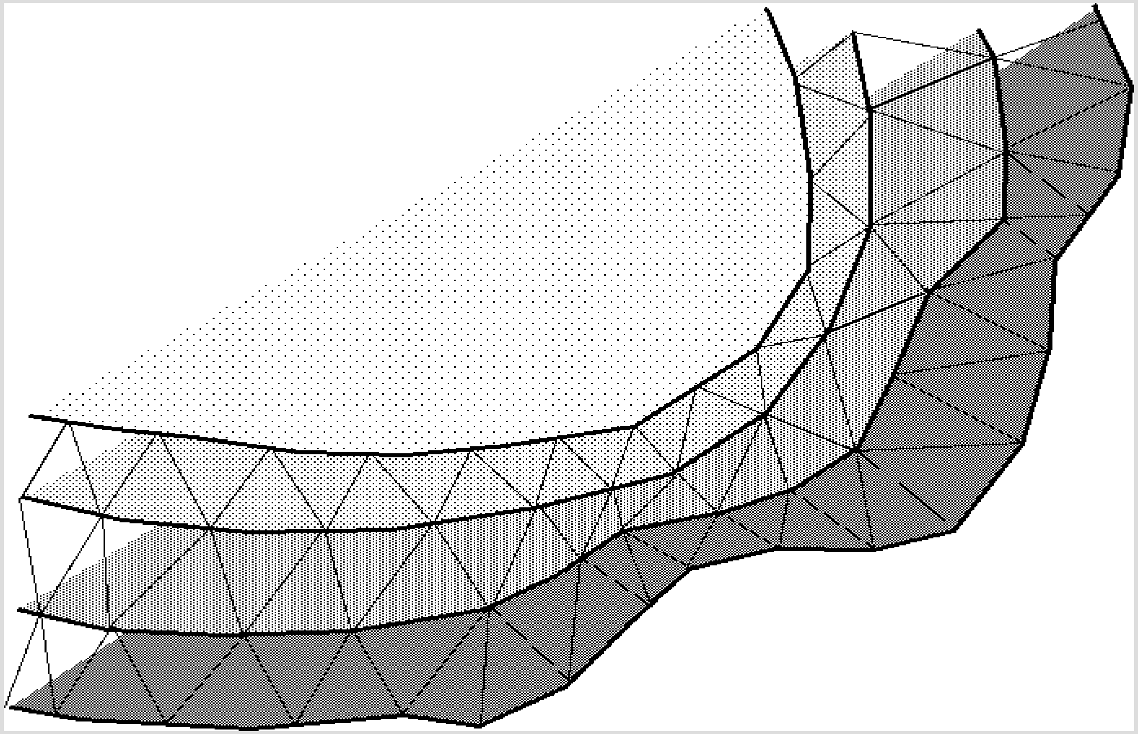
- Another view/model of volumetric data
- Reduces the volume visualization problem to conventional CG-methods
- Poses new problems:
  - How to find the surfaces?
    - Based on volume data segmentation
  - How to find their normals?
    - From segmented or from original data

Surface Extraction

- Contour tracking (Keppel 75, Fuchs 77)
- Opaque cubes (Herman and Liu 1979)
- Marching cubes (Lorensen and Cline 1987)
- Dividing cubes (- ,, -)
- Marching tetrahedra (- ,, -)
- Marching triangles (Hilton et al, 1996)
- Surface Detection by Ray Casting (Höhne et al, 1988)

Contour Tracking (1)

- Contour detection in slices
- Surface approximation by means of patches
- Rendering:
  - standard approaches

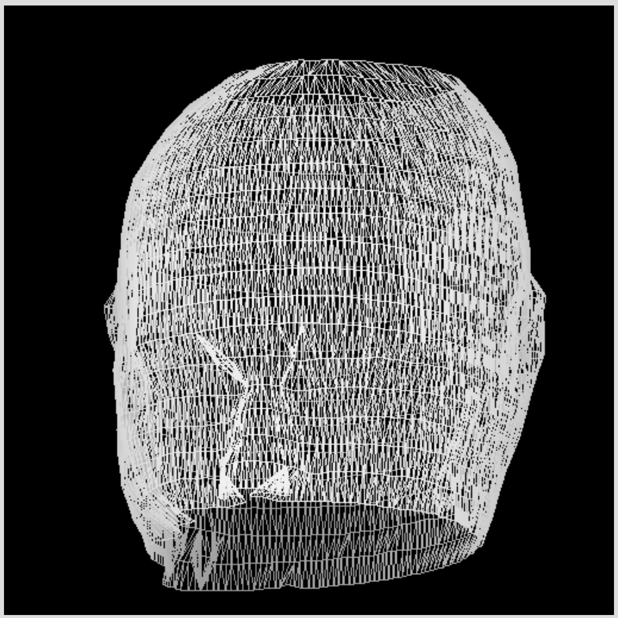


Contour Tracking (2)

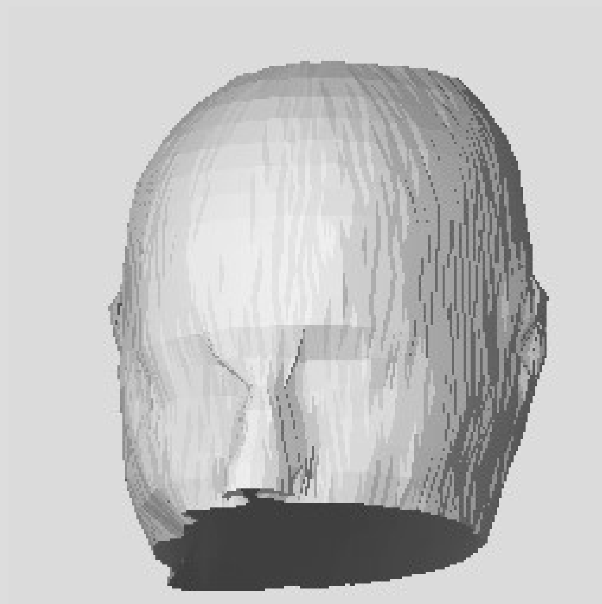
Special cases to consider:

- Only convex contours
- Convex and concave contours
- 1 contour – n contours
- m contours – n contours

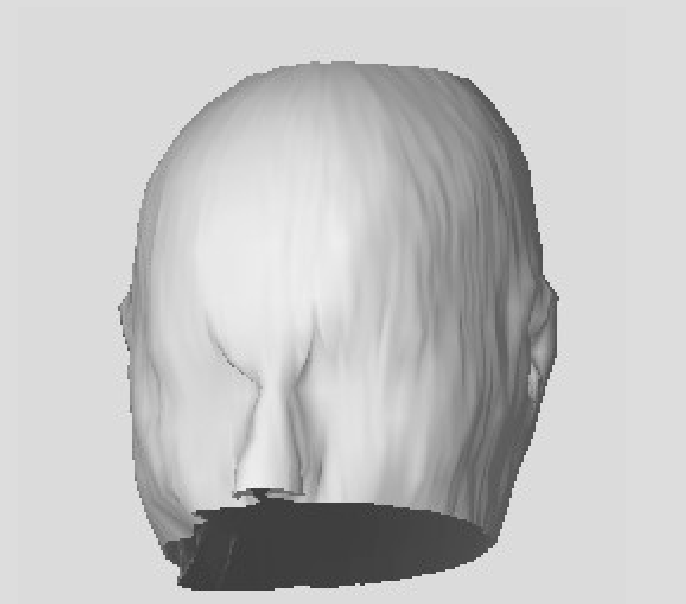
Contour Tracking Example



Wireframe model



Triangle Model Flat shading



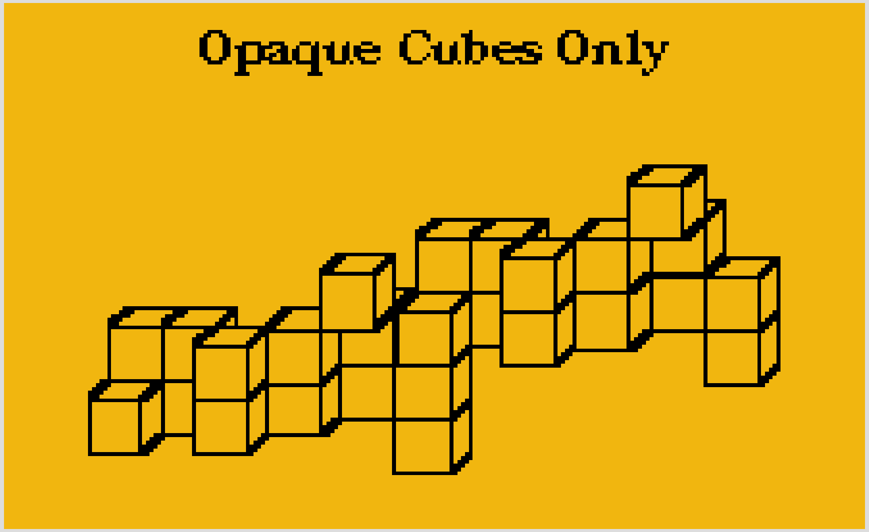
Triangle Model Gourard shading

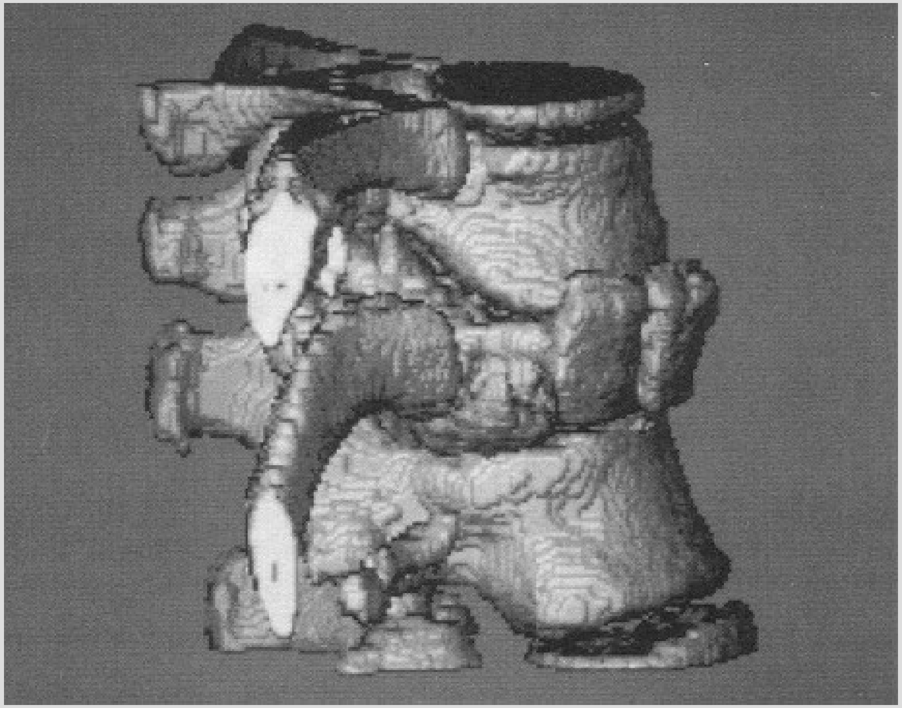
Contour Tracking Summary

- Complex topologies require complex algorithms
- Ambiguities possible – interaction required
- Usually used only for simple objects

Opaque Cubes

- Herman and Liu: „Cuberille“
- Represent „on“-cells as hexahedra
- Render six faces with traditional methods
- “Jaggy” images





Marching cubes
(Lorensen & Cline 1987)

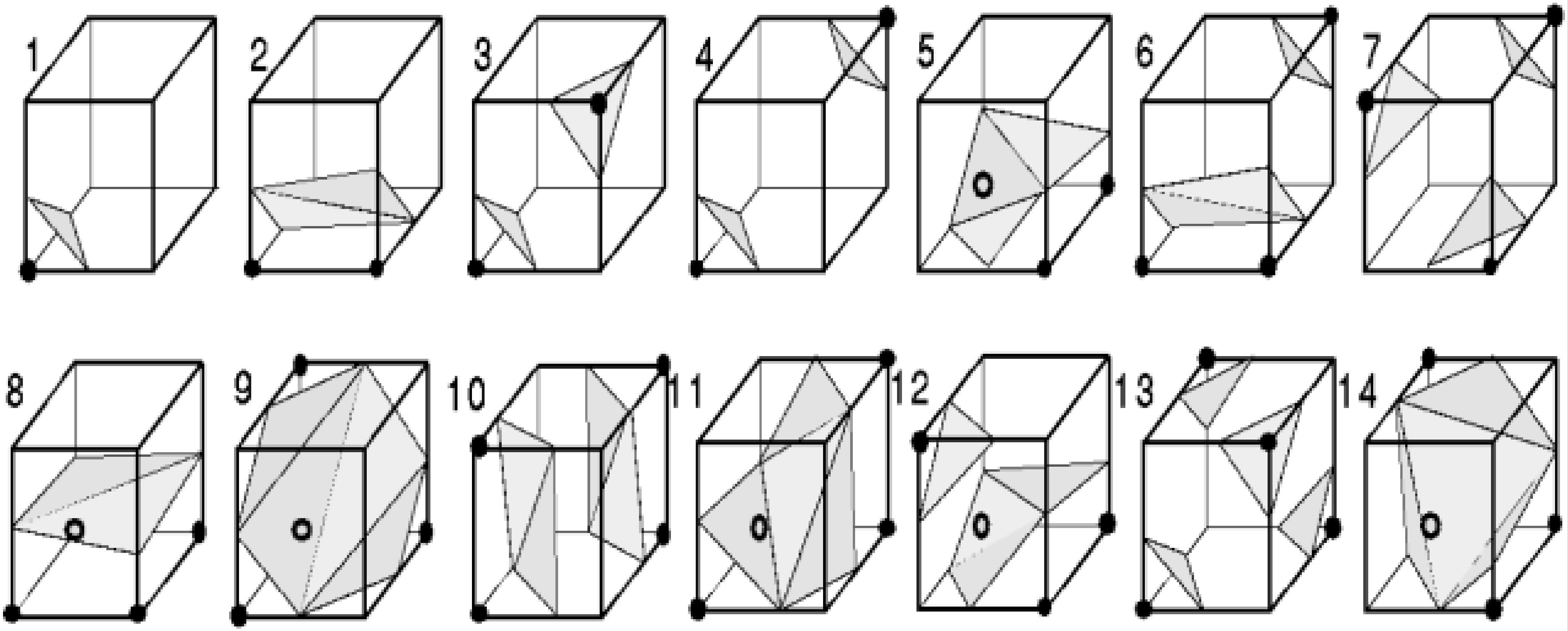
Tessellation on the cell level

- Rectilinear grid
- Basic cell – 8 samples
- Thresholding on level T:
  - All values above or below T: no surface in the cell
  - Values both below and above T: cell intersected by surface

Marching cubes

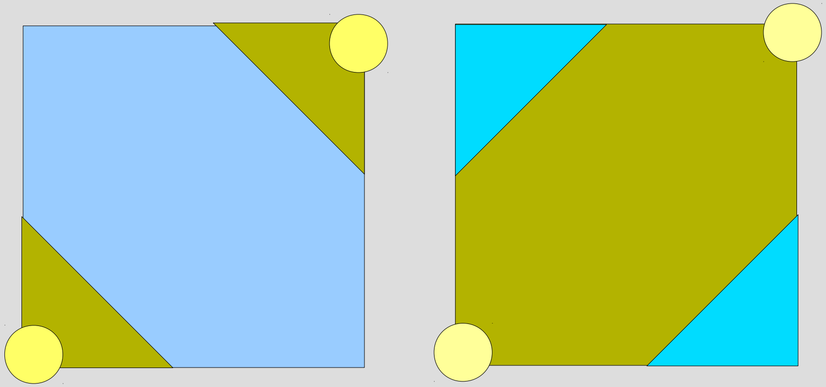
- Processing cell-by-cell (marching)
- Surface cell: surface approximation by 1-5 triangles
  - 256 basic possibilities
  - symmetry “black-white” $\Rightarrow$ 128 combinations
  - rotational symmetry: $\Rightarrow$ only 14 possibilities
- Vertex position: estimated by interpolation along edges

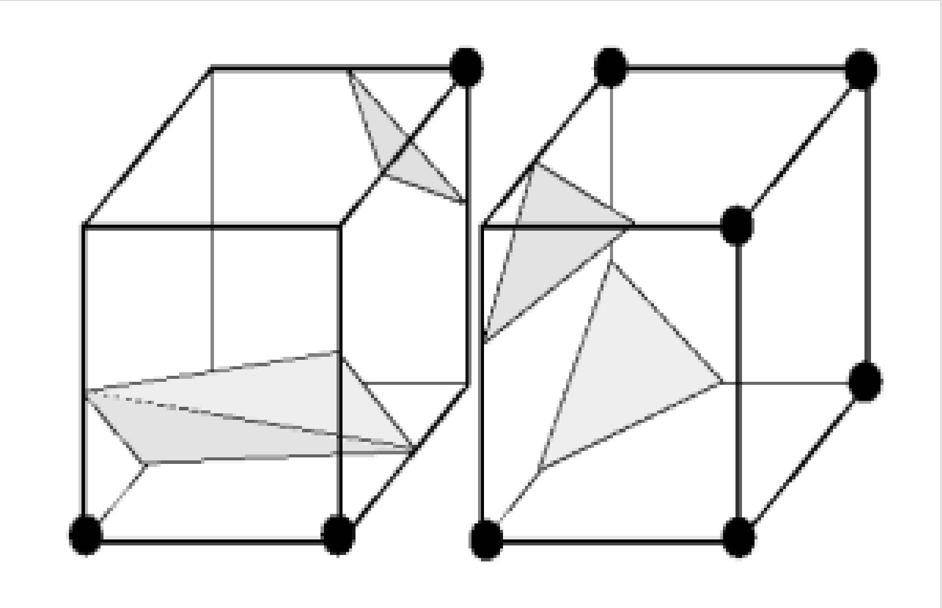
MC: triangulation table



MC ambiguity (1)

- Several tessellation possibilities for certain vertex configuration



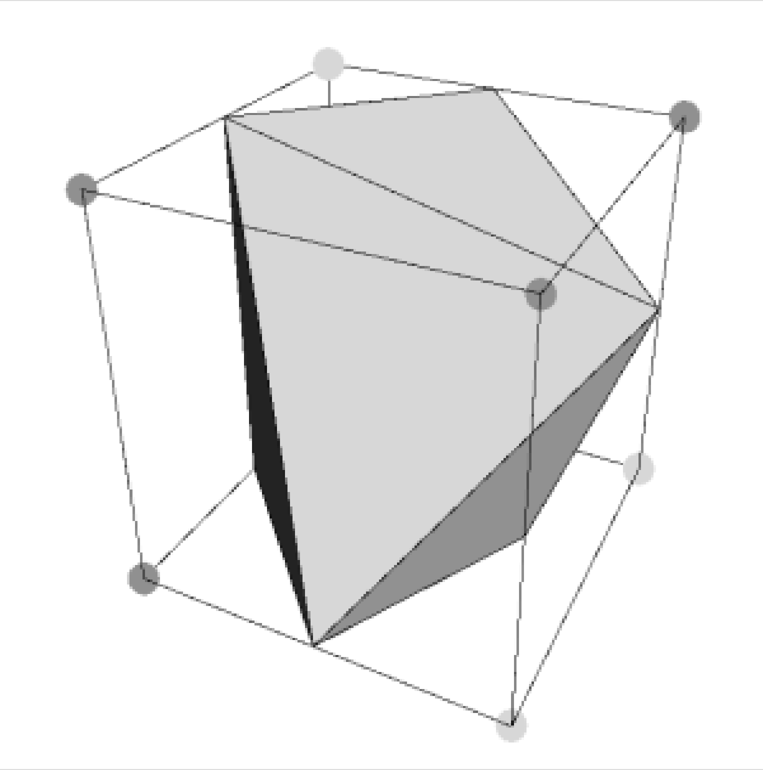


- Surface holes may appear
- Correction needed

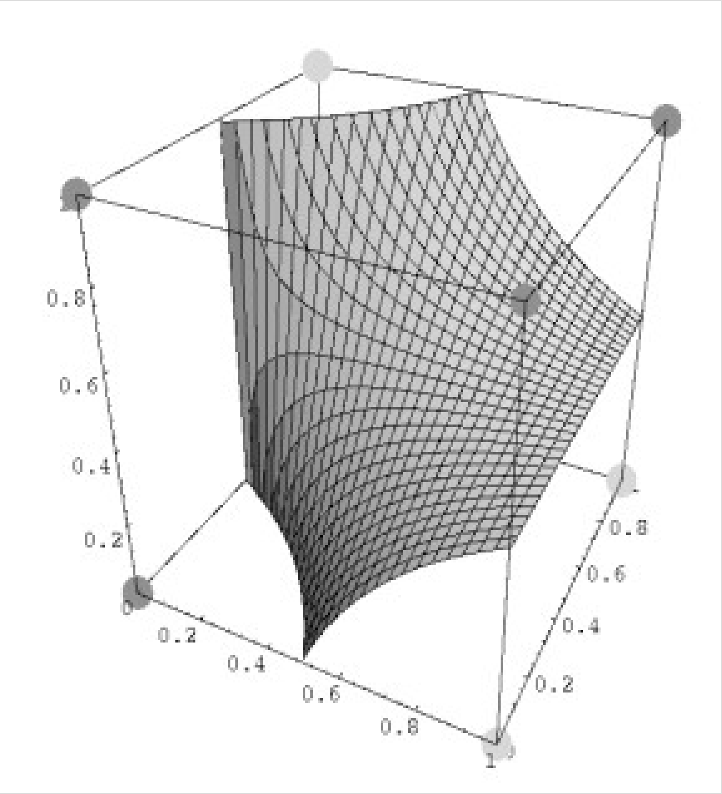
Incorrect combination in 3D

MC vs. Isosurfacing

- Isosurface definition by interpolation & thresholding

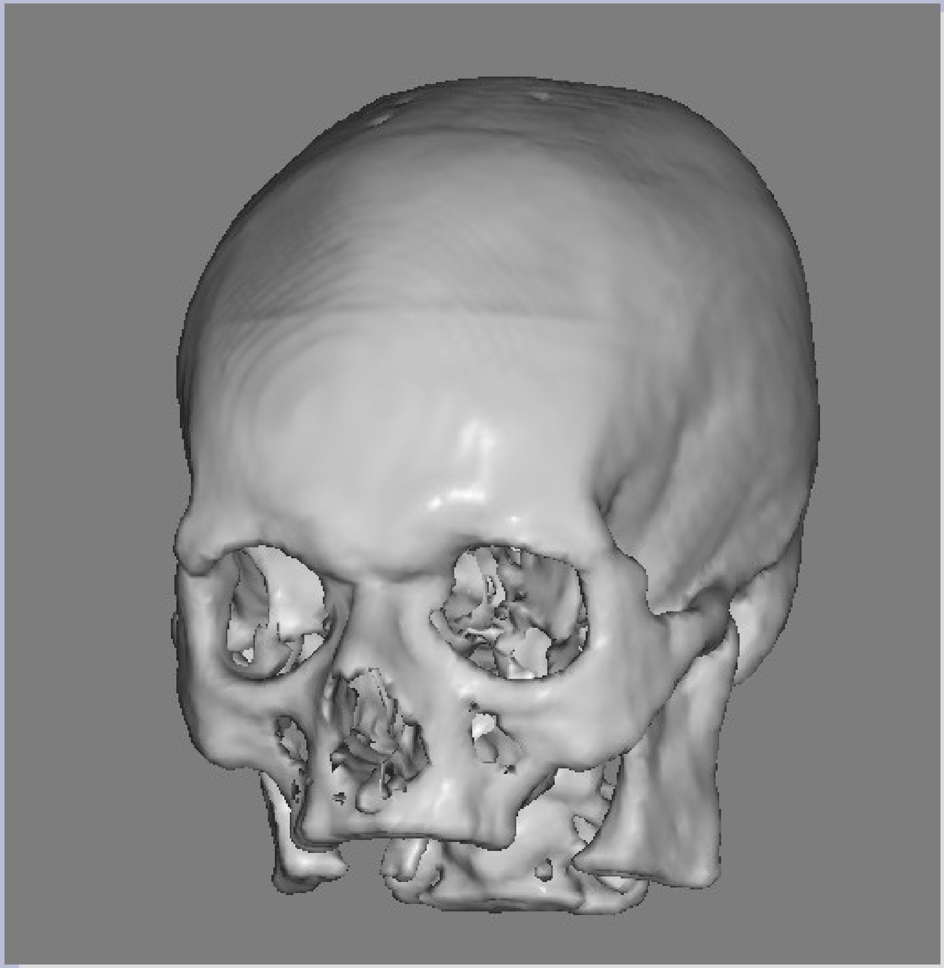


Triangulation by the MC algorithm

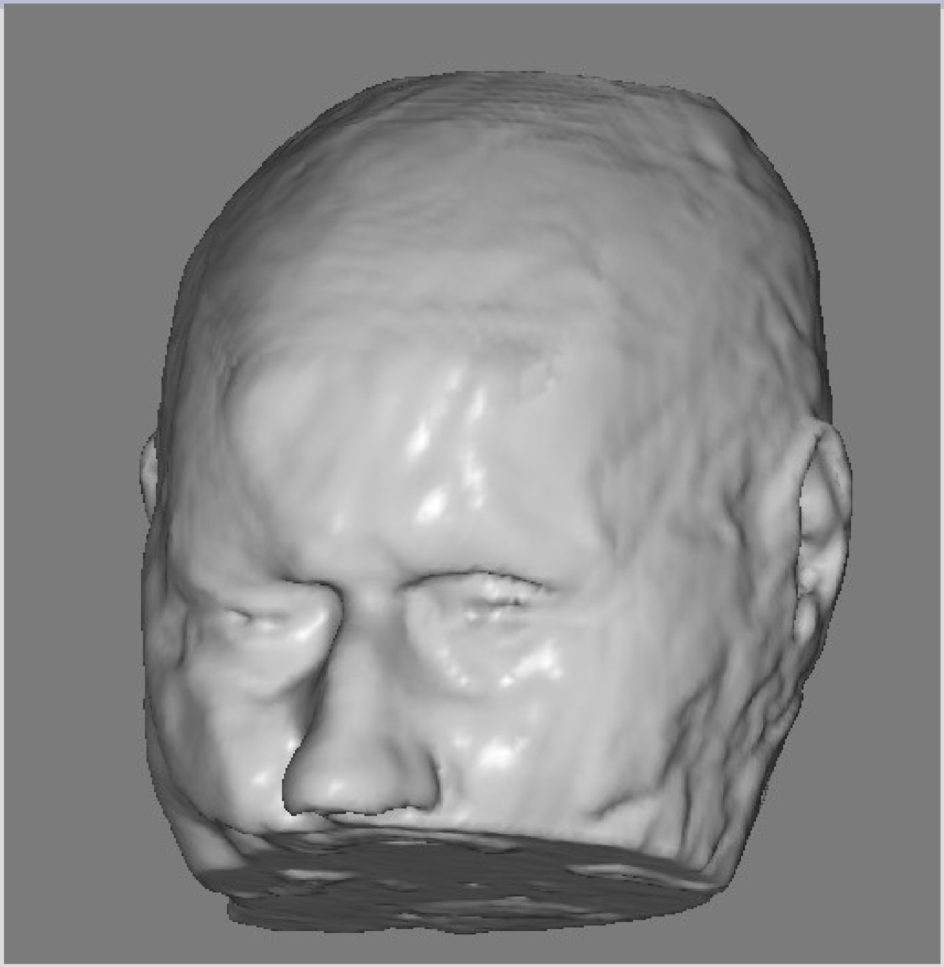


Interpolation surface within a single cell (trilinear interpolation)

MC example



422 400 triangles

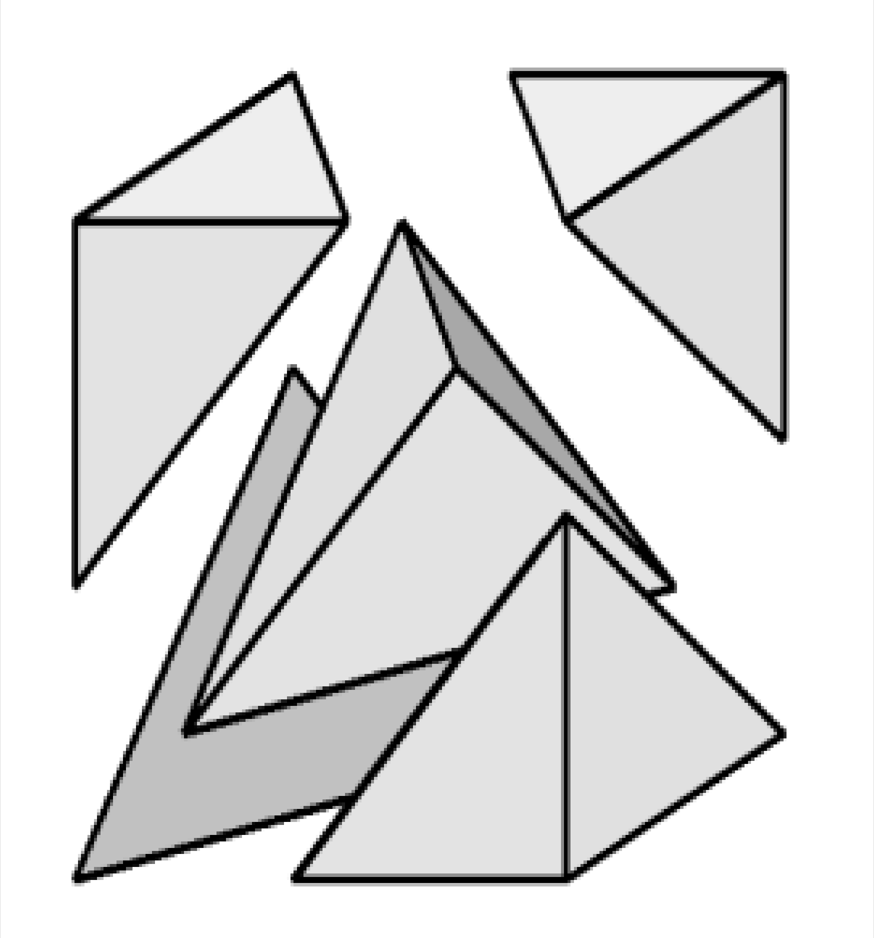


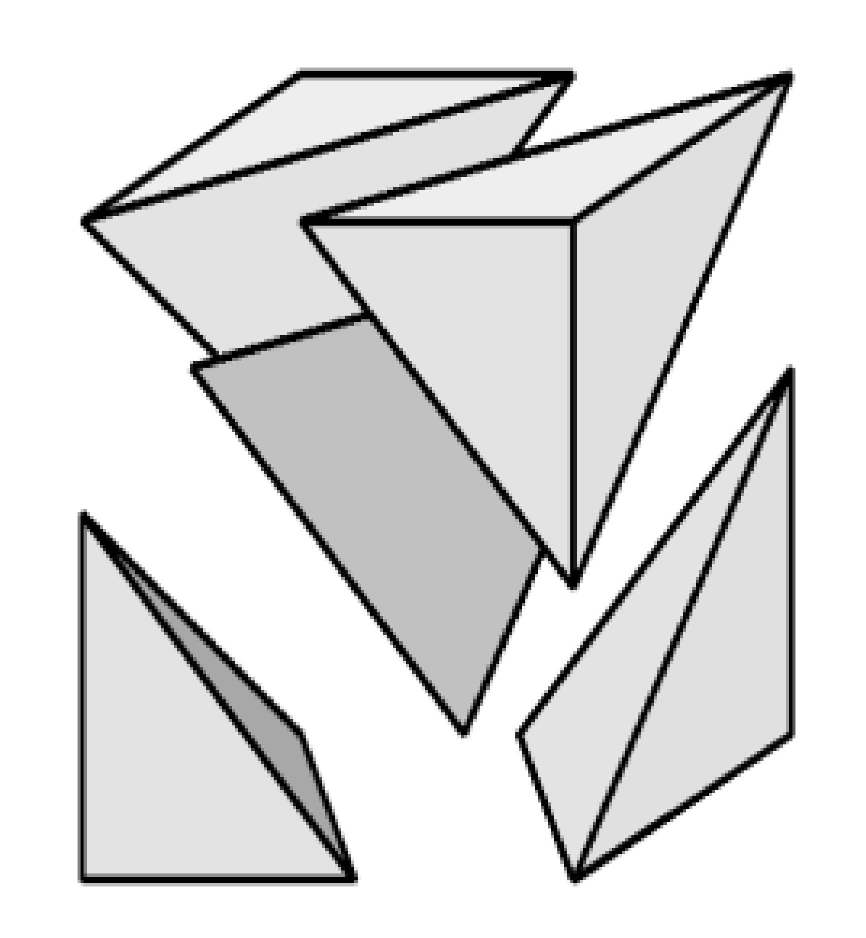
318 500 triangles

Marching Tetrahedra

- Solves ambiguity in MC tessellation
- Disambiguation by cell subdivision in tetrahedra:
  - 5 tetrahedra (2 possibilities)
  - 6 tetrahedra
  - 24 tetrahedra
- Doubles the number of triangles

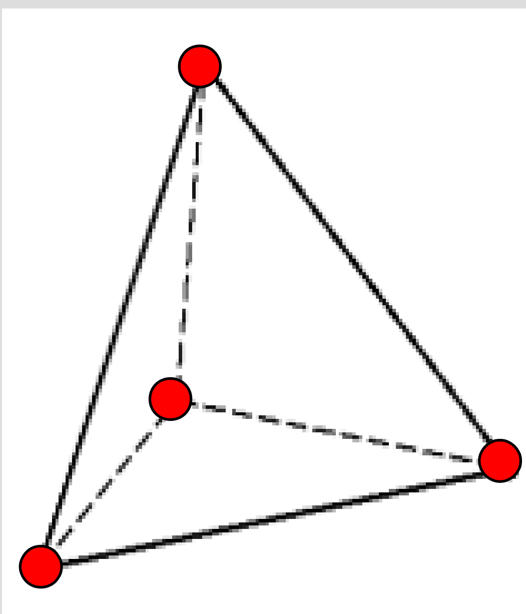
Cell subdivision in 5 tetrahedra



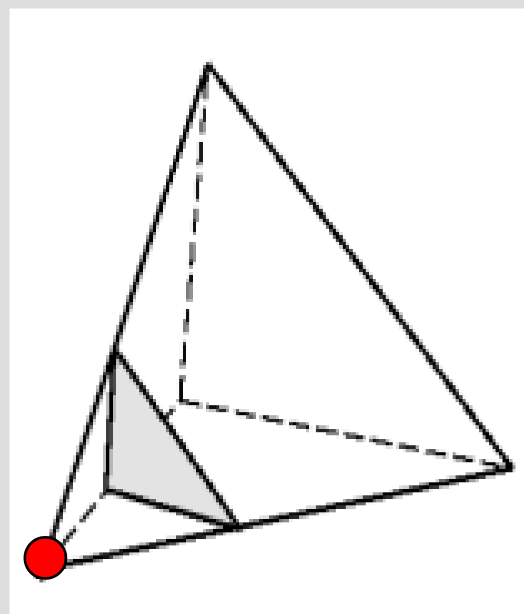


Tessellation of tetrahedra

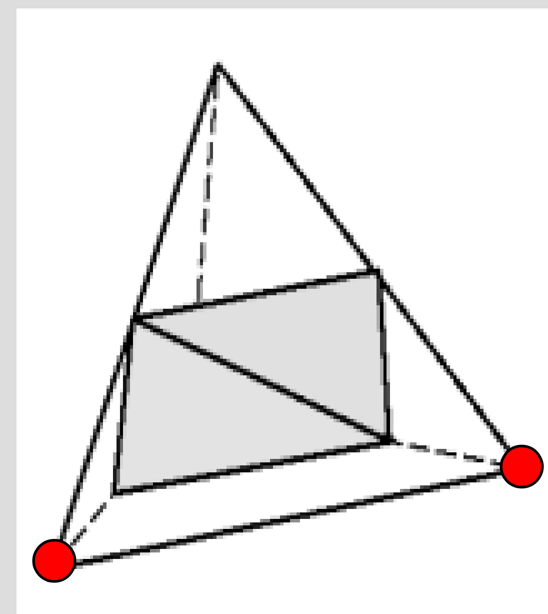
- No ambiguities



No surface



1 triangle



2 triangles

Dividing Cubes

- Specify a threshold value
- Identify voxels bracketing the isosurface
- Project voxel to image plane
  - If voxel projection covers less than a pixel, render it
  - Otherwise subdivide voxel and continue

Surface Extraction Advantages

- Uses known rendering methods
- Can take advantages of hardware
- View/light changes require only re-rendering (no pre-processing)
- Compact storage and transmission

Surface Extraction Disadvantages

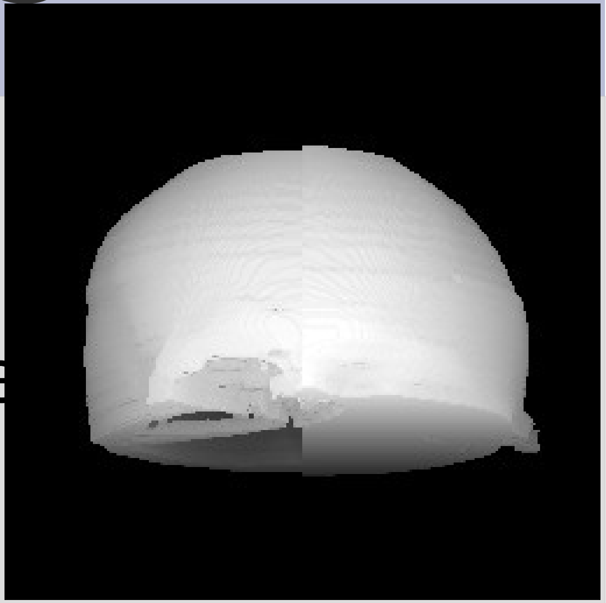
- Requires binary classification
- Throws away data
- Handles small features poorly
  - False positives and negatives
- Requires user intervention sometimes
- Cannot represent translucent data and weak surfaces

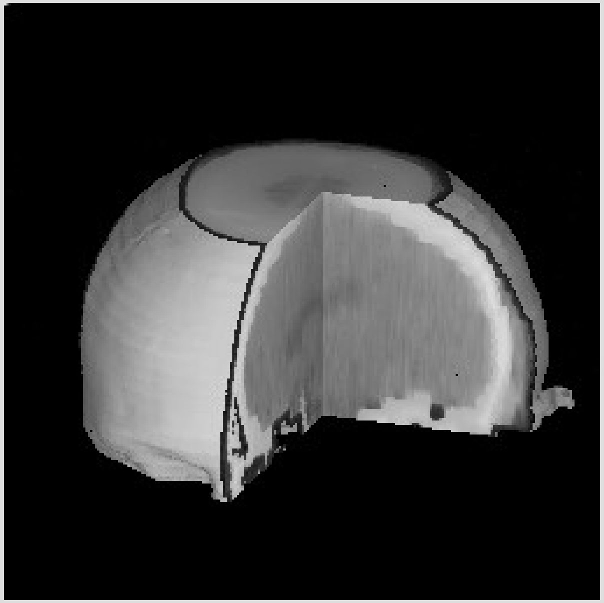
Surface shading

- Depth shading
  - Distance to observer stored
  - Shading in postprocessing possible
- Lambert shading

$$I = \vec{n} \cdot \vec{p}$$

  - $\vec{n}$ - Surface normal vector
  - $\vec{p}$ - Light direction
- Phong shading (with highlights)
- Others





Estimation of Surface Normal Vector

Using **Gradient**: $\vec{n} = \frac{\vec{g}}{\|\vec{g}\|}$

$$\vec{g} = \nabla f(x, y, z) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$$

- Gradient direction: direction of function's largest grow
- Gradient magnitude: rate of function change in that direction

Estimation of the Gradient Vector

- Most common option: central differences:

$$\nabla f(x_i, y_j, z_k) = \begin{pmatrix} \dfrac{f(x_{i+1}, y_j, z_k) - f(x_{i-1}, y_j, z_k)}{2} \\ \dfrac{f(x_i, y_{j+1}, z_k) - f(x_i, y_{j-1}, z_k)}{2} \\ \dfrac{f(x_i, y_j, z_{k+1}) - f(x_i, y_j, z_{k-1})}{2} \end{pmatrix}$$

- Other methods:
  - intermediate differences, ...