Marching cubes

Input: 3D voxel data (e.g. medical scan) OR 2D isosurface embedded in 3D

Output: Triangulation of object boundary or isosurface

Voxel = volume element (3D, usually cube)
Pixel = picture element (2D, usually square)

Each voxel has a scalar value
Goal: triangulation separates space w/value < α from space w/value > α

Define \( f : \mathbb{R}^3 \rightarrow \mathbb{R} \)

Isosurface: \( \{ p : f(p) = \alpha \} \)

How defined? Implicit surface modeling, surface reconstruction, ...

2 possibilities: continuous \( f \) e.g. signed distance function OR binary \( f \) (inside/outside) e.g. solid modeler

Computing \( f \) is often dominant cost.

2 possibilities: \( f \) is known only at voxel centers (usual for voxel data) OR \( f \) can be sampled at any point (usual for isosurfaces)
Marching Triangles

1. Choose background grid.

   ![Hexagonal grid](image1)  
   (good for pixel data)
   ![Equilateral grid](image2)  
   (good for isocontours)

2. Sample $f(p)$ at each grid vertex $p$.

   Label vertex "+" if $f(p) > \alpha$ (inside).
   "-" if $f(p) < \alpha$ (outside).
   
   
   "+" if $f(p) = \alpha$ (perturb the value a bit).

3. For each edge with one + endpoint and one - endpoint, compute a cut point $c$ where $f(c) \approx \alpha$.

   If $f$ cannot be sampled anywhere
   Use bisection.
   else if $f$ is continuously-valued
   Use interpolation.
   else
   Use midpoint.
   (consider smoothing the output later)

4. If a background triangle has differing signs, output a segment.
Marching squares

1. Choose grid.

2. Sample $f$ at each vertex.

3. Compute cut points.

4. All corners the same sign: \[ \begin{array}{cccc}
\uparrow & \uparrow \\
\downarrow & \downarrow \\
\end{array} \] No segment generated.

   One corner differs:

   \[ \begin{array}{cccc}
\uparrow \downarrow & \uparrow \\
\downarrow \uparrow & \downarrow \\
\end{array} \] \hspace{1cm} One segment.

   Two corners differ:

   \[ \begin{array}{cccc}
\uparrow \downarrow \uparrow & \uparrow \\
\downarrow \uparrow \downarrow & \downarrow \\
\end{array} \] \hspace{1cm} Two segments.

Ambiguity: \[ \begin{array}{ccc}
\square & \text{or} & \square \\
\end{array} \] ?

"Doesn't matter" vs. "Redo w/finer grid" vs. "Try to guess better choice".

Heuristic [Nielson & Hamann]: Assume $f$ is bilinear function.

$$ f(x, y) = Axy + Bx + Cy + D $$

Then $f(x, y) = \alpha$ is hyperbola w/axis-aligned asymptotes.

$$ (f_{01} - \alpha)(f_{10} - \alpha) > (f_{00} - \alpha)(f_{11} - \alpha) \quad \text{or} \quad (f_{01} - \alpha)(f_{10} - \alpha) < (f_{00} - \alpha)(f_{11} - \alpha) $$
Marching tetrahedra

1. Choose grid. Can't be equilateral!
   - Even vertices
   - Odd vertices

Better alternative: body centered cubic lattice.

2. Sample f at vertices.
3. Cut points.
4. One corner differs:
   Generate one triangle.

Two corners differ:

"Doesn't matter" vs. "choose dihedral angle nearest 180°"

(dot product of unit normal vectors is closest to one)
Marching cubes (Lorensen Cline '87; Wyvill McSheeters Wyvill '86)

1. Choose grid.
2. Sample f.
3. Cut points.
4. Stencils.

Important: No triangle should lie on a cube face! [Surface could fold over on itself.]

Wyvill:

Look-up table: each vertex is + or −, giving $2^8 = 128$ entries.
Each entry is list of triangles indexing cut points (12 edges of cube).
Symmetries: rotation, reflection, swapping + & -. $\rightarrow$ 15 distinct cases.

Problem:

"Hole" in surface... not watertight, not manifold.

Solution [Nielsen Hamann '91]: Use bilinear function to disambiguate ☐ vs. ☐ on faces.

$\rightarrow$ 34 distinct cases.
(Some of which introduce vertex inside cube, like Wyvill.)