Surface Visualization of 3D data

a) Voxel / rasterize approach
   - Assume value is constant throughout voxel i.e. value at cell center vs vertices.
   - Preprocess: threshold (e.g. find bone structure)
   - Use gradient shading
   - i.e. approximate a surface normal to voxels

\[ F = \sum dx \ dy \ dz \]

b) Tiling (assume values defined at vertices)
   - Take parallel slices of 3D data
   - Obtain contours of each slice e.g. with marching squares
   - Find corresponding contour(s) from one slice to the next.
   - Handle bifurcation if any
   - Triangulate
c) Marching Cubes (Lorensen (HyK) & Cline '87)
- Assume values defined at vertices.
- Vertex state is either 1 or 0 (above or below) threshold.

- Find state for cell:
  i.e. $2^6$ possible states
  ⇒ reduced to 15 by symmetry.
  see page 161

  e.g. 

- Visit each cell,
- Determine state for each cell
- Index into table to determine location of surfaces.
- Linear Interpolate to find edge intersections
- Triangulate.
Ambiguity:

2D case:

3D case:
⇒ arbitrary pairing may result in holes.

Alternatives:

1. Split hexahedron (cubes) into tetrahedra, use marching tetrahedra.
   in 2D: split into Δ's
   have problems with "bumps"
   e.g.

   ![Diagram](image)
   threshold = 2.5

2. Look at additional neighboring cells.

3. Add complementary cases.
Attributes:
- color / texture
- contour
- normals
- flow on surface
- #

Lots of polygons:
- Simplification of meshes.
- View dependent iso-surfaces

Mini-cubes / spheres.
- alternative before DVR.
- transparency.
Direct Volume Rendering:

- No intermediate geometry
- Accumulate information along 1D line through volume
- 2 general methods:
  1) Image order: ray casting
  2) Object order: splatting
- Others:
  1) Texture slabs
  2) Fourier volume rendering
- No thresholding needed (optional)
- View-dependent
- Uses all data vs only see some data
- Fuzzy vs sharp appearance

\[
B = \int J D a_s \cos \theta e^{-\mu r} dr
\]

- \(\theta\) = angle between I & E at each voxel along ray
- B = cumulative attenuated info along ray
- J = decay constant
- D a_s = accumulated densities between voxel & light source
Image-Order:

Raycasting, Levoy '88

3D density data \( \rightarrow \) 3D color \( C(x,y,z) \)
e.g. CT scan \( \rightarrow \) 3D opacity \( \alpha(x,y,z) \)

\( C(x,y,z) = \) determined by gradient ("surface"
normal) & lighting (independent of other voxels i.e.
does not attenuate lighting of voxel between
volume & light)

\( \alpha(x,y,z) = \) determined by mapping density
values to different types of tissues.

Raycast - combine \( C \) & \( \alpha \) into

\[ C(R) = \text{color seen by ray } R. \]

\[ C(R) = \sum_{k=0}^{K} \left( C(R,k) \cdot \alpha(R,k) \cdot \prod_{i=k+1}^{K} (1 - \alpha(R,i)) \right) \]

\( (R,k) \rightarrow k\text{th voxel along ray} \)
\( C(R,0) \rightarrow \text{color of background} \)
\( \alpha(R,0) \rightarrow 1 \)

For each pixel
- shoot ray
- calculate \( C(R) \)
Variations & Issues:

1. MIP - Maximum intensity projection.

   - Scalar value along ray
   - Distance along ray

   Could be "intensity", color, raw data, density, ...

   - A = maximum intensity
   - B = average intensity

   \( A \Rightarrow \) set \( C(R) = A \)
   \( B \Rightarrow \) set \( C(R) = B \)
   
   \( C(R) = \) distance to reach certain accumulated value.

2. Sampling

   - Regular sampling
   - Cell intersection
(a) rays - parallel / perspective

(b) Starting point if regular sampling

(c) data vs color

A) calculate color at each vertex, then trilinear interpolate to sample

B) use data at each vertex, trilinear interpolate to sample, convert to color at that point based on interpolated values.

(F) Termination e.g. accumulated opacity.
Object-Order:

For each voxel
- project (shadows) voxel to projection screen
- apply filtering to 'splat'
- alpha-blending

-may be done in
  a) front-to-back
  b) back-to-front

* discrete process which produces holes on the periphery or when perspective projection gets extreme, countered by distributing the energy across multiple pixels via a footprint table

Coherent Projection (parallel projection only)

each voxel:
Transfer Function:

1D
2D

\[ \text{opacity} \]
\[ \text{density} \]

"threshold"
"iso-surface"

multiple
material

"mod ill percentage transfer function"

To highlight regions where there is a sharp transition, use gradient magnitude T.F.