Volume Rendering

Scalar Data Visualization

1. Iso-surface extraction: extract geometry first, then render the resulting polygons
2. Volume rendering: Direct display method project each data point onto the screen

Early attempts (pre-MCs)

The Cuberille Approach (1979)
- Each voxel has a value
- Each voxel has 6 faces
- Applying a threshold to perform binary classification
- Draw the visible faces of the boundary voxels as polygons

(Fairly jagged images)

Contour Tracking

Extract contours at each section and connect them together (1976 and after)

New Methods Were Needed

Better image quality is necessary
Finding object’s boundary sometimes can be difficult
The process of connecting boundary contours is also very complicated
The intermediate geometry size can be huge

Direct 2-D Display of 3D Objects

Tuy and Tuy 1984, IEEE CG & A
(one of the earliest volume rendering techniques)

Direct: No conversion from data to geometry
Basic Idea

Based on the idea of ray tracing

- Treat each pixel as a light source
- Emit light from the image to the object space
- The ray stops at the object boundary
- Calculate shading at the boundary point
- Assign the value to the pixel

Algorithm details

- Data Representation: (establish 3D volume and 2D screen space)
- Viewing
- Sampling
- Shading

Data Representation

3D volume data are represented by a finite number of cross sectional slices (a stack of images)

\[ N \times 2D \text{ arrays} = 3D \text{ array} \]

Data Representation (2)

What is a Voxel? - Two definitions

- A voxel is a cubic cell, which has a single value covering the entire cubic region
- A voxel is a data point at a corner of the cubic cell. The value of a point inside the cell is determined by interpolation

Viewing

Ray Casting

- Where to position the volume and image plane
- What is a 'ray'
- How to march a ray

Viewing (1)

1. Position the volume

Assuming the volume dimensions is \( w \times w \times w \)
We position the center of the volume at the world origin

\[ \text{volume center} = \frac{w}{2}, \frac{w}{2}, \frac{w}{2} \]  
(focal space)

Translated \( T(-w/2, -w/2, -w/2) \)

data to world matrix?

\[ \text{world to data matrix?} \]
2. Position the image plane

Assuming the distance between the image plane and the volume center is D, and initially the center of the image plane is (0,0,-D)

3. Rotate the image plane

A new position of the image plane can be defined in terms of three rotation angles $\alpha, \beta, \gamma$ with respect to x,y,z axes

Assuming the original view vector is [0,0,1], then the new view vector $g$ becomes:

$$
\begin{bmatrix}
\cos \beta & 0 & -\sin \beta \\
0 & 1 & 0 \\
\sin \beta & 0 & \cos \beta
\end{bmatrix}
\begin{bmatrix}
\cos \gamma & \sin \gamma & 0 \\
-\sin \gamma & \cos \gamma & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1
\end{bmatrix}
$$

Now,

R: the rotation matrix

$S = B - D \times g$

$U = [1,0,0] \times R$

$V = [0,1,0] \times R$

Image Plane: $L \times L$ pixels

Then

$E = S - \frac{L}{2} \times u - \frac{L}{2} \times v$

So

Each pixel $(i,j)$ has coordinates

$P = E + i \times u + j \times v$

We enumerate the pixels by changing $i$ and $j$ from 0 to $L-1$.

4. Cast rays

Remember for each pixel on the image plane $P = E + i \times u + j \times v$

and

the view vector $g = [0,0,1] \times R$

So the ray has the equation:

$Q = P + k (d \times g)$

$d$: the sampling distance at each step

$k = 0.12...$
**Sampling (1)**

In tuys' paper

\[ Q = P + K \times V \ (v=dxg) \]

At each step k, Q is rounded off to the nearest voxel (like the DDA algorithm)

Check if the voxel is on the boundary or not (compare against a threshold)

If yes, perform shading

**Shading**

- Take the voxel position, distance to the image plane, the object normal, and the light position into account

- The paper does not describe in detail, but you can imagine we can easily perform local illumination (diffusive or even specular).

- The distance can be used alone to provide an 3D depth cue (e.g. distant voxels are dimmer)

**Pros and Cons**

+ Require no boundary estimation/hidden surface removal
+ No display holes

- Binary object representation
- Flat lighting (head on illumination)
- Jagged surface
- No semi-transparencies

A more sophisticated classification and lighting model in [Levoy 88]