Volume rendering

Robert A. Drebin, Loren Carpenter, Pat Hanrahan
SIG ’88 vol 22

Presented for Comp250 by Alexandra Lauric
The Goal

• Given a volume (3D Image), render a 2D image as output. The process is called “Volume Rendering”.

![Diagram](image)
Definitions

• **Slice** – 2D image
• **Volume** – series of slices
• **Value** – number associated with each point
• **Level /Iso Surface** – the set of points having the same value.
Applications

• Medical imaging
• Paleontology and Archaeology
• Oil and Gas Exploration
• Security applications
Related Work

• **Scene-based**
  – Maximum Intensity Projection
  – Additive reprojection
  – Surface Rendering (Iso Surface)

• **Object-based**
  – Octree representation
  – Polygonal Models
The algorithm

- render volumes containing a mixture of materials.
- both the interior of the materials and the boundary between materials are colored.
- artifacts caused by aliasing and quantization are avoided.
Steps

Input raw data
Classification
Material Percentage Volumes
Surface Extraction
Lighting Model
Viewing & Projection
## Input Raw Data

<table>
<thead>
<tr>
<th>Type</th>
<th>Spatial Resolution</th>
<th>Intensity Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT</td>
<td>512x512</td>
<td>12 Bits</td>
</tr>
<tr>
<td>MRI</td>
<td>256x256</td>
<td>12 Bits</td>
</tr>
</tbody>
</table>
Classification 1/3

• “Label” each voxel. – give meaning to it.
  Probabilistic classifier: each voxel is a mixture of materials.

  \[ Pr(I) = \sum_{i=1}^{n} p_i P_i(I) \]

  \( Pr(I) \) – probability that voxel has value \( I \).
  \( p_i \) – percentage of material \( i \) in voxel
  \( P_i \) – Probability that material \( i \) has value \( I \)
  \( n \) – number of material
Classification 2/3

• When using CT, the distribution $P_j(I)$ are known in advanced.

• The bayesian estimate of the percentage of each material within a voxel of intensity $I$ is given by:

$$p_i(I) = \frac{P_i(I)}{\sum_{j=1}^{n} P_j(I)}$$
• Determine the classification function from slide histogram.

• Assume that no more than two materials overlap.

• Assume linearity between peaks.
Material Percentage Volumes

• We have the material percentage in each voxel → we can create a material property volume. (we color the volume)
• Associate each material \(i\) with a color \(C_i\).
• Calculate the color of each voxel.

\[
C = \sum_{i=1}^{n} p_i C_i \\
C_i = (\alpha_i R_i, \alpha_i G_i, \alpha_i B_i, \alpha_i)
\]
Surface Extraction

• For each voxel compute:
  – “strength”
  – normal

• Calculate density volume: \( D = \sum_{i=1}^{n} p_i \rho_i \)
  \( D \) – total \( \rho \) of a voxel
  \( \rho_i \) – material assigned density

• Combine two materials by assigning the same density.

• Compute normal to surface from \( \rho \) volume gradient:
  \( N_x = D_{x-1} - D_x \)
  \( N_y = D_{y-1} - D_y \)
  \( N_z = D_{z-1} - D_z \)
Voxel Light Model

A ray of light is travelling in the voxel. It enters in the back, crosses the surface and exits from the front.

• assume light rays are not attenuated as they travel through the volume.

resulting color:

\[ I' = C + (1-\alpha_c)I = C \over I \]

voxel contains a surface:

\[ I' = \left( C_F \over \left( C_S \over \left( C_B \over I \right) \right) \right) \]

surface reflected color:

\[ C_S = f(N, L)C_{Diffuse} + g(E, N, L)C_{Light-source} \]
Viewing & Projection 1/3

- Use parallel merge projection
- To compute the final image, we project the image through Z plane.
  \[ I_z = \frac{C_z}{I_z} \]
- The images are merged to the final image from back to front (\( I_0 \) is the final image).
- The initial image \( I_n \) is set to background.
Viewing & Projection 2/3

• What we want to view the scene from another angle?
• To preserve the simplicity of the parallel merge projection, consider the viewing coordinate system fixed and transform the scene (the volume).
• Transform each individual slice and put it back in the volume.
Viewing & Projection 3/3

• Transform volume to lie in the viewing coordinate system.

\[ T = P_z(z_e) \ R_z(\psi) \ R_y(\Phi) \ R_z(\theta) \]

- \(P_z\) – perspective transform
- \(R_z, R_y\) – rotations about y and z axis
Results 1/2
Results 2/2
Thank you.