Introduction to Scientific Visualization

Volume Rendering

February 13, 2018
Overview

• Scalar Volumes
• Ray casting
Isosurfaces

• Level sets
  • Images (2D) $\rightarrow$ curves
  • Volumes (3D) $\rightarrow$ surfaces

• Marching cubes (Lorensen & Cline 87)
  • Triangulated surface at a given isovalue
Isosurfacing is limited

- Isosurfacing is "binary"
  - point inside isosurface?
  - voxel contributes to image?

- Is a hard, sharp boundary necessarily appropriate for the visualization task?

Slice  Isosurface  Volume Rendering
Spirit of volume rendering

- “Every voxel contributes to image"
- Greater flexibility

Marc Levoy, 1988
"Display of Surfaces from Volume Data"
Pipelines: Isosurface vs. Vol Rendering

The standard line - "no intermediate geometric structures"

Volume Data → Isosurface extraction → Triangles

Volume Data → volume rendering

 rendered

image

image
The standard line - "no intermediate geometric structures"

Pipelines: Isosurface vs. Vol Rendering
What is Volume Rendering?

• Any rendering process that maps from volume data to an image without introducing binary distinctions / intermediate geometry

• How to make the data visible?
What is Volume Rendering?

• Any rendering process which maps from volume data to an image without introducing binary distinctions / intermediate geometry

• How to make the data visible?

→ color and opacity
Direct volume rendering

• Directly get a 3D representation of the volume data
• The data is considered to represent a semi-transparent light-emitting medium
  — Even gaseous phenomena can be simulated
• Approaches are based on the laws of physics (light emission, absorption, scattering)
• The volume data is used as a whole (look inside, see all interior structures)
Isosurfacing is Limited

- Isosurfacing poor for ...
  - measured, "real-world" (noisy) data
  - amorphous, "soft" objects

virtual angiography  bovine combustion simulation
Fundamentals (Physics)

• Density attenuation

• Kajiya: exponential decay of light intensity

\[ e^{-\tau \int_{t_1}^{t_2} \sigma(t) dt} \]

• Volume Rendering Integral
Fundamentals (Physics)

- Density attenuation
- Kajiya: exponential decay of light intensity
  \[ e^{-\tau \int_{t_1}^{t_2} \sigma(t) dt} \]
- Volume Rendering Integral
  \[ c(R) = \int_0^D c(s(x(t))) \mu(s(x(t))) e^{-\int_0^t \mu(s(u)) du} dt \]
Fundamentals (Physics)

- Density attenuation
  
  - Kajiya: **exponential decay of light intensity**
    
    \[ e^{-\tau \int_{t_1}^{t_2} \sigma(t) dt} \]

- Volume Rendering Integral
  
  \[ c(R) = \int_0^D \underbrace{c(s(x(t)))\mu(s(x(t)))}_{\text{emitted color}} e^{-\int_0^t \mu(s(u))du} dt \]
Fundamentals (Physics)

• Density attenuation

• Kajiya: exponential decay of light intensity

\[ e^{-\tau \int_{t_1}^{t_2} \sigma(t)dt} \]

• Volume Rendering Integral

\[ c(R) = \int_0^D c(s(x(t)))\mu(s(x(t)))e^{-\int_0^t \mu(s(u))du}dt \]
Fundamentals (Physics)

• Density attenuation

• Kajiya: exponential decay of light intensity

\[ e^{-\tau \int_{t_1}^{t_2} \sigma(t) \, dt} \]

• Volume Rendering Integral

\[ c(R) = \int_{0}^{D} c(s(x(t))) \mu(s(x(t))) e^{-\int_{0}^{t} \mu(s(u)) \, du} \, dt \]
Fundamentals (Physics)

• Density attenuation

• Kajiya: exponential decay of light intensity

\[ e^{-\tau \int_{t_1}^{t_2} \sigma(t) \, dt} \]

• Volume Rendering Integral

\[ c(R) = \int_0^D \left[ c(s(x(t))) \mu(s(x(t))) e^{-\int_0^t \mu(s(u)) \, du} \right] \, dt \]
Fundamentals (Physics)

- Density attenuation
- Kajiya: exponential decay of light intensity
  \[ e^{-\tau \int_{t_1}^{t_2} \sigma(t) dt} \]
- Volume Rendering Integral
  
  Integral along ray

  \[ c(R) = \int_0^D c(s(x(t))) \mu(s(x(t))) e^{-\int_0^t \mu(s(u)) du} dt \]
Volume Rendering Integral

\[
c(R) = \int_{0}^{D} c(s(x(t)))\mu(s(x(t)))e^{-\int_{0}^{t}\mu(s(u(t)))du}dt
\]

- \(x\): position along ray \(R\)
- \(c\): color associated with value \(s\)
- \(s\): scalar value at \(x\)
- \(\mu\): density/opacity associated with that value
DVR

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
DVR

Absorption

Emission

active scattering
active scattering
Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
DVR

Absorption

Emission

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
**DVR**

- Emission and absorption of light

\[
I(s) = I(s_0) e^{-\tau(s_0, s)} \quad \tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s) ds
\]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
**DVR**

*Integration*

- **Emission and absorption of light**

\[
I(s) = I(s_0) e^{-\tau(s_0, s)}
\]

\[
\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s) ds
\]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
DVR

• Emission and absorption of light

\[ I(s) = I(s_0) e^{-\tau(s_0, s)} + \int_{s_0}^{s} q(\dot{s}) e^{-\tau(s, \dot{s})} d\dot{s} \]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
• Emission and absorption of light

\[ I(s) = I(s_0)e^{-\tau(s_0, s)} + \int_{s_0}^{s} q(\dot{s})e^{-\tau(s, \dot{s})} d\dot{s} \]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
DVR

Resample along ray

\[ I(s_i)I(s_{i+1}) \]

\[ s_0 \quad s_i \quad s_{i+1} \]

\[ I(s_0) \quad q(s_i), A(s_i) \quad q(s_{i+1}), A(s_{i+1}) \]

\[ \alpha = A(s_{i+1}) \]

\[ I(s_{i+1}) = \alpha q(s_{i+1}) + (1 - \alpha)I(s_i) \]

\[ = q(s_{i+1}) \text{ OVER } I(s_i) \]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
DVR

Resample along ray

\[ I(s_0) \quad q(s_i), \quad A(s_i) \quad q(s_{i+1}), \quad A(s_{i+1}) \]

\[ \alpha = A(s_{i+1}) \]

\[ I(s_{i+1}) = \alpha q(s_{i+1}) + (1 - \alpha)I(s_i) \]

\[ = q(s_{i+1}) \text{ OVER } I(s_i) \]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
DVR

Resample along ray

\[ I(s_i)I(s_{i+1}) \]

\[ \text{TF}(s_i) \]

\[ I(s_0) \quad q(s_i), \quad A(s_i) \quad q(s_{i+1}), \quad A(s_{i+1}) \]

\[ \alpha = A(s_{i+1}) \]

\[ I(s_{i+1}) = \alpha q(s_{i+1}) + (1 - \alpha)I(s_i) \]

\[ = q(s_{i+1}) \overset{\text{OVER}}{\text{I}} I(s_i) \]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
DVR

Resample along ray

\[ I(s_i)I(s_{i+1}) \]
\[ s_0 \quad s_i \quad s_{i+1} \]

\[ TF(s_i) \quad TF(s_{i+1}) \]

\[ I(s_0) \quad q(s_i), A(s_i) \quad q(s_{i+1}), A(s_{i+1}) \]

\[ \alpha = A(s_{i+1}) \]

\[ I(s_{i+1}) = \alpha q(s_{i+1}) + (1 - \alpha)I(s_i) \]

\[ = q(s_{i+1}) \text{ OVER } I(s_i) \]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
DVR

Resample along ray

\[ I(s_i)I(s_{i+1}) \]

\[ s_0 \quad s_i \quad s_{i+1} \]

\[ TF(s_i) \]

\[ TF(s_{i+1}) \]

\[ I(s_0) \quad q(s_i), A(s_i) \quad q(s_{i+1}), A(s_{i+1}) \]

Back-to-front Compositing with \( \alpha = A(s_{i+1}) \)

\[ I(s_{i+1}) = \alpha q(s_{i+1}) + (1 - \alpha)I(s_i) \]

\[ = q(s_{i+1}) \text{ OVER } I(s_i) \]

Markus Hadwiger, IEEE Visualization 2002 Tutorial Notes
General Components

- Basic diagram
Color and Opacity Transfer Functions

• $C(p), \alpha(p)$ – $p$ is a point in volume

• Functions of input data $f(p)$
  • $C(f), \alpha(f)$ – these are 1D functions
  • Can include lighting affects
    • $C(f, N(p), L)$ where $N(p) = \text{grad}(f)$
  • Derivatives of $f$
    • $C(f, \text{grad}(f)), \alpha(f, \text{grad}(f))$
Transfer Functions (TFs)

Map data value $f$ to color and opacity.

$\alpha$  RGB

$f$
Transfer Functions (TFs)

Map data value $f$ to color and opacity

Human Tooth CT
Transfer Functions (TFs)

Map data value $f$ to color and opacity

Shading, Compositing…

Human Tooth CT
Transfer Functions (TFs)

Map data value $f$ to color and opacity

Shading, Compositing...
Volume Rendering Usefulness

Measured sources of volume data

- CT (computed tomography)
- PET (positron emission tomography)
- MRI (magnetic resonance imaging)
- Ultrasound
- Confocal Microscopy
Volume Rendering Usefulness

- Synthetic sources of volume data
- CFD (computational fluid dynamics)
- Voxelization of discrete geometry
Volume Rendering: Interfaces

Transfer function, with shading

Skin/Air
Bone/Soft tissue
Bone/Air
Concepts

- Voxels
  - basic unit of volume data
- Interpolation
  - trilinear common, others possible
- Gradient
  - direction of fastest change
- Compositing
  - "over operator"
Gradient

\[ \nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \]

\[ \approx \left( \frac{f(1,0,0) - f(-1,0,0)}{2}, \frac{f(0,1,0) - f(0,-1,0)}{2}, \frac{f(0,0,1) - f(0,0,-1)}{2} \right) \]

• Approximates "surface normal" (of isosurface)
Compositing: Over Operator

\[ \mathbf{c}_f = (0, 1, 0) \]
\[ \alpha_f = 0.4 \]  \hspace{1cm} \text{front}

\[ \mathbf{c}_b = (1, 0, 0) \]
\[ \alpha_b = 0.9 \]  \hspace{1cm} \text{back}

\[ \mathbf{c} = \alpha_f \mathbf{c}_f + (1 - \alpha_f) \alpha_b \mathbf{c}_b \]
\[ \alpha = \alpha_f + (1 - \alpha_f) \alpha_b \]

\[ \mathbf{c} = 0.4 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + (1 - 0.4) \times 0.9 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.54 \\ 0.4 \\ 0 \end{pmatrix} \]
Compositing: Over Operator

$$c_f = (0, 1, 1)$$
$$\alpha_f = 0.4$$

$$c_m = (0, 1, 0)$$
$$\alpha_m = 0.4$$

$$c_b = (1, 0, 0)$$
$$\alpha_b = 0.9$$

$$c = \alpha_f c_f + (1 - \alpha_f) \alpha_b c_b$$

$$c = \alpha c + (1 - \alpha) b$$

$$0.4 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + (1 - 0.4) \times 0.9 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.54 \\ 0.4 \\ 0 \end{pmatrix}$$

$$0.4 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + (1 - 0.4) \begin{pmatrix} 0.54 \\ 0.4 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.324 \\ 0.64 \\ 0.4 \end{pmatrix}$$
Compositing: Over Operator

\[
\begin{align*}
c &= \alpha_f c_f + (1 - \alpha_f) \alpha_b c_b \\
\alpha &= \alpha_f + (1 - \alpha_f) \alpha_b
\end{align*}
\]

Order Matters!

\[
\begin{align*}
c &= (0.324, 0.64, 0.4) \\
\alpha &= 0.964
\end{align*}
\]

\[
\begin{align*}
c &= (0.324, 0.64, 0.24) \\
\alpha &= 0.964
\end{align*}
\]
Pixel Compositing Schemes
Pixel Compositing Schemes

First

Depth
Pixel Compositing Schemes

Max intensity

First

Depth
Pixel Compositing Schemes

- Max intensity
- Average
- First

Depth
Pixel Compositing Schemes

Depth

Max intensity
Accumulate
Average
First
Compositing – First (Threshold)

Extracts iso-surfaces (again!)
Compositing - Average

Intensity vs. Depth

Average

Synthetic Reprojection
Compositing - MIP

Maximum Intensity Projection
Magnetic Resonance Angiogram
Compositing - Accumulate

Make transparent layers visible;
Uses a transfer function for color/opacity
Raycasting

- **Back to Front**
  - straightforward use of over operator
  - intuitively backwards

- **Front to Back**
  - intuitively right
  - not simple over operator
  - facilitates early ray termination
Raycasting: compositing

• Back to Front:

\[ C_{i+1} = a_i c_i + (1-a_i) C_i \]
Raycasting: compositing

• Front to Back:

\[ C_i; A_i \]

\[ c_i; a_i \]

\[ C_{i+1}; A_{i+1} \]

\[ C_{i+1} = C_i + (1 - A_i)a_i c_i \]

\[ A_{i+1} = A_i + (1 - A_i)a_i \]
Raycasting: compositing

Which is better?

• Front to Back:
  \[ C_{i+1} = C_i + (1 - A_i)a_i c_i \]
  \[ A_{i+1} = A_i + (1 - A_i)a_i \]

• Back to Front:
  \[ C_{i+1} = a_i c_i + (1-a_i)C_i \]
General Components

• Basic diagram
General Components

• Basic diagram
General Components

• Basic diagram
General Components

• Basic diagram
General Components

• Basic diagram
General Components

• Basic diagram
General Components

• Basic diagram
Ray-casting - Highlights

Advantages:

• Simple algorithm
• Inherently parallel
• Can add features (like a ray-tracer)

Disadvantages:

• Slow (lots of rays, lots of samples) though GPU friendly
• Must sample densely
• (Requires entire data set in memory)
Unstructured Volume Rendering

• How to accurately display a volume defined over an unstructured grid?

• Numerous approaches:
  • Ray casting
  • Ray tracing
  • Sweep plane algorithms (e.g., ZSWEEP)
  • PT algorithm of Shirley and Tuchman
Projected Tetrahedra

- Decompose each cell into tetrahedra
- Sort the tetrahedra in a back to front fashion
- Project each tetrahedron and render its decomposition into 3 or 4 triangles

Two different non-degenerate classes of the projected tetrahedra
Volume Density Optical Model

• For the Volume Density Optical Model of Williams et al. the emission and absorption along a light ray is defined by the transfer functions $\kappa(f(x,y,z))$ and $\rho(f(x,y,z))$ with $f(x,y,z)$ being the scalar function.

• Usually the transfer functions are given as a linear or piecewise linear function, or as a lookup table.
Tetrahedra Compositing

• For each rendered pixel the ray integral of the corresponding ray segment has to be computed

• Observation: The ray integral depends only on $S_f$, $S_b$, and I for the Volume Density Optical Model of Williams et al.
3D Texturing Approach

- Compute the three-dimensional ray integral by numerical integration and store the integrated chromaticity and opacity in a 3D texture

- Assign appropriate texture coords \((S_f, S_b, l)\) to the projected vertices of each tetrahedron
Pros / Cons of PT Method

• **Pros:**
  • Object order method
  • Hardware-accelerated approach
  • Per-pixel exact rendering

• **Cons:**
  • Sort Required
  • Slower than uniform volume rendering
Summary

Volume Ray Casting

• (Requires entire data set in memory)
• Can produce reflections, shadows, and complex illumination “relatively” easily
• Easily parallelizable (GPU friendly)