Raycasting

Direct volume rendering

Volume rendering (sometimes called direct volume rendering) stands for methods that generate images directly from 3D scalar data.

"Directly" means: no intermediate geometry (such as an isosurface) is generated.

Volume rendering techniques
• depend strongly on the grid type
• exist for structured and unstructured grids
• are predominantly applied to uniform grids (3D images).

Raycasting

Raycasting is historically the first volume rendering technique. It has common with raytracing:
• image-space method: main loop is over pixels of output image
• a view ray per pixel (or per subpixel) is traced backward
• samples are taken along the ray and composited to a single color

Differences are:
• no secondary (reflected, shadow) rays
• transmitted ray is not refracted
• more elaborate compositing functions
• samples are taken at intervals (not at object intersections)

Ray templates

A ray template (Yagel 1991) is a voxelized ray which by translating generates all view rays. Ray templates speed up the sampling process, but are obviously restricted to orthographic views.

Algorithm:
• Rename volume axes such that z is the one "most orthogonal" to the image plane.
• Create ray template with 3D version of Bresenham algorithm, giving 26-connected rays which are functional in z coordinate (have exactly one voxel per z-layer)
• Translate ray template in base plane, not in image plane
Ray templates

Incorrect: translated in image plane

Correct: translated in base plane

Compositing

Two simple compositing functions can be used for previewing:

- Maximum intensity projection (MIP):
  - maximum of sampled values
  - result resembles X-ray image
- Local maximum intensity projection (LMIP):
  - first local maximum which is above a prescribed threshold
  - approximates occlusion
  - faster & better (!)

\[
\text{Compositing}
\]

Assume that each sample on a view ray has color and opacity:

\[
(C_{f,0}, \alpha_{f,0}, \ldots, C_{f,N}, \alpha_{f,N}) \quad C_i \in [0,1]^3, \alpha_i \in [0,1]
\]

where the 0th sample is next to the camera and the Nth one is a (fully opaque) background sample:

\[
\begin{align*}
C_{\text{background}} &= C_{f,0} \\
\alpha_{\text{background}} &= 1
\end{align*}
\]

\(\alpha\)-compositing can be defined recursively:

Let \(C_{f}^b\) denote the composite color of samples \(f, f+1, \ldots, b\)

Recursion formula for back-to-front compositing:

\[
C_{f}^b = \alpha_{f} C_{f} + (1-\alpha_{f}) C_{f+1}^b
\]

\(\alpha\)-compositing reveals the closed formula for \(\alpha\)-compositing:

The first few generations, written with transparency \(T_j = 1-\alpha_j\):

\[
\begin{align*}
C_0^b &= \alpha_0 C_0 \\
C_1^b &= \alpha_1 C_1 + \alpha_1 \alpha_0 T_{b-1} \\
C_2^b &= \alpha_2 C_2 + \alpha_2 C_1 + \alpha_2 \alpha_0 T_{b-2} + \alpha_2 \alpha_1 T_{b-1} \\
C_3^b &= \alpha_3 C_3 + \alpha_3 C_2 + \alpha_3 \alpha_0 T_{b-3} + \alpha_3 \alpha_1 T_{b-2} + \alpha_3 \alpha_2 T_{b-1} + \alpha_3 \alpha_0 T_{b-2} + \alpha_3 \alpha_1 T_{b-1} \cdot T_{b-2}
\end{align*}
\]

front-to-back compositing can be derived from the closed formula:

Let \(T_j^F\) denote the composite transparency of samples \(f, f+1, \ldots, b\)

Then the simultaneous recursion for front-to-back compositing is:

\[
C_i^f = \alpha_i C_i \\
T_i^F = 1 - \alpha_i \\
C_i^{f+1} = C_i^f + \alpha_{i+1} C_{i+1} T_i^F \\
T_i^{F+1} = (1-\alpha_{i+1}) T_i^F
\]

Advantage of front-to-back compositing: early ray termination when composite transparency falls below a threshold.
The emission-absorption model

How realistic is $\alpha$-compositing?
The emission-absorption model (Sabella 1988) yields a basic volume rendering equation

$$L(x) = \int e(x') e^{-\frac{\tau(x')}{\epsilon(x')}} dx'$$

The equation describes the radiance (power per unit area per solid angle [W/m²/sr]) arriving along a ray at the position $x$ on this ray. The emission function $\epsilon(x)$ describes the photons "emitted" by the volume along the ray. The absorption function $\tau(x)$ is the probability that a photon traveling over a unit distance is lost by absorption.

The emission-absorption model is based on Boltzmann's transport equation in statistical physics, but completely ignores scattering. In more general models $\tau(x)$ is an extinction function having both an absorption term and a scattering term.

Instead, in the emission-absorption model:
- incident scattering is modeled by the emission function
- loss by scattering can be thought to be part of the absorption.

Transfer functions

Transfer functions map raw voxel data to opacities and colors as needed for the $\alpha$-compositing.

Inputs of TF (one or more):
- voxel value $s(x)$
- gradient magnitude $\nabla s(x)$
- higher derivatives of $s(x)$

Opacity transfer function $\alpha(s(x))$
Color transfer function $C(s(x))$ or premultiplied: $\tilde{C}(s(x))$

In general TF don't depend on spatial location, except for focus+context techniques.

Example of a bivariate (=2D) transfer function:

By choosing different opacity transfer functions different types of applications can be achieved.

Examples:
- standard application
- isosurface
- 3D edge detector

Discrete version of emission-absorption model

$$L(x) = \sum_{i} e_i \Delta x \sum_{j} e_j \Delta x \left(1 - \alpha_i\right)$$

matches the $\alpha$-compositing formula

$$C_i = \sum_{j} \alpha_j C_j$$

and gives interpretations of "opacity" and "color":

$$\alpha = 1 - e^{-\frac{\tau}{\epsilon}}$$
$$\alpha C = e^{-\frac{\tau}{\epsilon}}$$

The product $\tilde{C} = \alpha C_i$ is called a premultiplied or associated color.
Example: bivariate transfer function for isosurface of constant "thickness".

\[ \text{opacity} \quad \text{gradient magnitude} \parallel s(x) \parallel \]

The color transfer function allows to make a simple classification.

Example:

\[ \text{color (RGB)} \]

Better (but more expensive) classification than with a transfer function is obtained by segmentation (typically slice by slice, semi-automatic).

Pre-classified volume data.
Example: "virtual frog" dataset (Lawrence Berkeley Labs).

Volume rendering of segmented volume data (from VTK book).

Example: classifications with different transfer functions (McGill Univ.)

Example: temperature data for opacity and color TF (Nissan Res.Ctr.)

In pre-classification, the voxels can also be lit:

- The gradient is perpendicular to the local isosurface. It can be used as a normal vector for a Phong lighting (without rendering the isosurface itself).
- Reflection coefficients can be assigned by a separate transfer function ("materials" instead of colors only).
- The diffuse lighting can be applied to the entire volume dataset as a pre-processing since it is independent of the viewing direction.
Pre- vs. post-classification

For quality reasons, current volume rendering implementations often use post-classification.

Pre-Classification:
1. Transfer functions are applied to voxels
2. Results are interpolated to sample locations.

Post-Classification:
1. Raw data are interpolated to sample locations.
2. Transfer functions are applied to sampled data.

Preintegration

Idea (Engel 2001):
- simulate infinitely many interpolated samples between two successive samples \( s_i = s(x_i) \) and \( s_{i+1} = s(x_{i+1}) \)
- assuming:
  - field \( s(x) \) varies linearly between samples
  - transfer functions don’t depend on derivatives

The discrete formula for opacity at a sample was

\[
\alpha_i = 1 - e^{-\int_{x_i}^{x_{i+1}} r(s) ds}
\]

The continuous version, for a sample interval \([x_i, x_{i+1}]\), is

\[
\alpha_i = 1 - \int_{x_i}^{x_{i+1}} r(s) ds
\]

Assuming now \( s(x) \) to be linear between samples, we get

\[
\alpha_i = 1 - e^{-\int_{x_i}^{x_{i+1}} r(s) ds} \frac{d}{d} \quad \text{with} \quad d = \|x_{i+1} - x_i\|
\]

which is called a preintegrated opacity transfer function.

Preintegration

The integral

\[
\int_{s_i}^{s_{i+1}} r(s) ds = \int_{s_i}^{s_{i+1}} \frac{r(s) ds - r(x_i) ds}{d}
\]

can be evaluated by two lookups in a precomputed table of

\[
\int_{s_i}^{s_{i+1}} r(s) ds
\]

Alternatively, the preintegrated opacity TF could be tabulated for all possible combinations of \((s_i, s_{i+1}, d)\), especially if the sampling distance \(d\) is chosen constant.
Preintegration

The composite color of the same interval

\[ C_i = \int_{s_i}^{s_{i+1}} e(s(x)) e^{-\int_{s_i}^{s} \alpha(x) \, dx} \, ds \]

simplifies for linear \( s(x) \) to

\[ C_i = \frac{d}{s_{i+1} - s_i} \int_{s_i}^{s_{i+1}} e(s(x)) e^{-\int_{s_i}^{s} \alpha(x) \, ds} \, ds \]

which is a preintegrated color transfer function.

Again, it can be tabulated for all combinations of \((s_i, s_{i+1}, d)\), or it can be approximately calculated from tabulated values of \(\int_{s_i}^{s_{i+1}} e(s') ds'\) and \(\int_{s_i}^{s_{i+1}} e(s') ds'\) (moving the exponential term out of the integral).

Raycasting hardware

Today’s GPUs are well suited for volume rendering. Object-space methods are more straightforward, but also some raycasting implementations on GPUs have been done.

In contrast, two examples of special raycasting hardware:

- **VIZARD (1997, U. Tübingen):**
  - FPGA-based system
  - long pre-processing
  - 10 Hz for 256\(^3\) volume
  - perspective view raycasting

- **volumePRO (1999, MERL):**
  - PCI board
  - no pre-processing
  - 30 Hz for 256\(^3\) volume
  - orthographic raycasting
  - ray templates
  - shear-warp method
  - z-supersampling

**Example: with / without supersampling**

Some special features:
- cropping
- clipping planes

**ASIC chip of VolumePRO vg500 board has 4 parallel pipelines with stages:**
- interpolation
- gradient estimation
- classification/shading
- compositing

Image is rendered in a single pass through the volume.

Off-chip memory for slices (shown in blue)