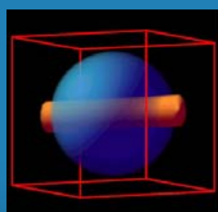
Introduction to *VOLUME RENDERING*



Presented by Zvi Devir

Few Methods of Representation

- Polygonal - Triangle Mesh
- Freeforms - parametric curves, patches...
- Solid Modelling - CGS (Constructive Solid Geometry)
- Space subdivision - Volume Rendering

… *And more*

What is Volume Rendering

Volume is a three-dimentional array of voxels. Just the same way an image is a 2D array of pixels.

3D images produced by CT, MRI or even mathematic scalar fields are easily represented as volumes.

Voxel is the basic element of the volume.
Typical volume size may be 128^3 voxels, but any other size is acceptable.

Volume Rendering means rendering the voxel-based data into viewable 2D image.

Why Volume Rendering?

Pros:

- Natural representation of CT/MRI images.
- Transparency effects (Fire, Smoke…).
- High quality

Cons:

- Huge data sets
- Computationally expensive.
- Cannot be embedded easily into polygonal scene

Rendering Methods

There are two categories of volume rendering algorithms:

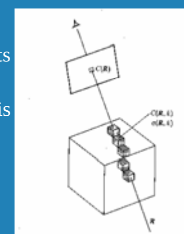
1. Ray casting algorithms (Object Order)
   - Basic ray-casting
   - Using octrees
2. Plane Composing (Image Order)
   - Basic slicing
   - Shear-Warp factorization
   - Transparent textures

Ray Casting

Ray casting is a method in which for every pixel in the image, a ray is cast through the volume. The ray intersects a line of voxels.

While passing, the color of the pixel is accumulated acording the voxels' color and transperacy.

Basic Comlexity = Depth*ImageSize



Ray Casting

The accumulation of the color is done using the following equations: $C^{out} = C^{in} + (1-\alpha^{in}) \cdot C^{*}_{voxel}$

$[\alpha^{out} = \alpha^{in} + (1-\alpha^{in}) \cdot \alpha_{voxel}]$

When $C^{*}_{voxel} = C_{voxel} \cdot \alpha_{voxel}$

Starting with $C_{in} = (0, 0, 0, 0)$

The accumulation is done front backwards.

If alpha is close enough to 1.0, the color will not change much. Therefore a threshold for alpha (the transparency) may be set, guarantees an early ray termination when possible.

Ray Casting - Performance Improvements

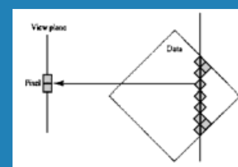
- Use of Octrees
  - Minimizes the number transparent voxels during the accumulation, since a group of transparent voxels may be represented as a single node in the octree.
  - More efficient memory usage.
- Interleaving methods
  - Sample every two (n) voxels as long as voxels are fully transparent.
  - Sample only ¼ of the points in the image and interpolate - Faster for interactive mode, but less quality.
- Pyramids, k-d trees and other data-structures.

Ray Casting - Improving Image Quality

- Multi Cast or Super Sampling:
  Instead of sampling one ray per pixel, sampling 4 rays per pixel.
  Better image... but four times longer to render.
- Ray subdividing:
  Used with perspective projection. When the rays draw away from each other, the sampling of the volume is not complete.
  The solution is to divide the ray when the rays density falls.

Plane Composing

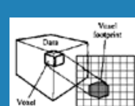
The plane composing (or “slicing”) method, divides the volume into slices.
During the rendering process, the slices are composes one over the other, producing the image.



Basic Complexity = VolumeSize

Plane Composing

- The slicing method works best in the case of parallel projection and the volume edge is parallel to the view plane.
  In this case, the voxels can be accessed easily as planes in the volume.
- However, when the view-plane is not parallel to the volume, a more complicated sampling algorithm is needed.
- Since a voxel is not projected onto one pixel, a filter should be used for the composition.



Plane Composing

The planes composing may be done from the rear side forwards, or front side backwards.

If the planes are composed read side forwards, the accumulation of the color is done using the following equations: $C^{[illegible]} = (1-\alpha^{[illegible]})\cdot C^{*} + C^{[illegible]}$

$[\alpha^{[illegible]} = (1-\alpha^{[illegible]})\cdot \alpha^{*} + \alpha^{[illegible]}]$

When $C^{*}_{voxel} = C_{voxel} \cdot \alpha_{voxel}$

Starting with $C_{in} = (0, 0, 0, 0)$

Otherwise, the ray casting accumulation equations remain valid.

Plane Composing - Performance Improvements

- Shear-warp factorization of the viewing matrix. (later)
- Information of min-max non-transparent voxels. (later)
- Again, interleaving methods
  - Sample only half of the planes.
  - Sample only ¼ of the points in the image...

And even,

- Use the hardware - graphic accelerators.
  - Rendering the trivial way - Sending the volume planes one by one as transparent textures to the AGP…
  - Back to front

Shadings & Classification

Raw volume data does not include normal or edges.
So edge detection and normals calculation should be done.

This may be accomplished using
- Marching Cubes algorithm
- Gradient estimations

In CT images, the data is only the density. Real colors and transparency are set by classifying the voxels using some criteria.
There are algorithms which allow dynamic change of the voxels classifications.

Bibliography and References

General:

- 3D Computer Graphics by Alan Watt
- *http://graphics.stanford.EDU/projects/volume/*
- *http://www.cs.wpi.edu/~matt/courses/cs563/talks/powwie/p1/ray-cast.htm*
- VolumePro - real time volume rendering using hardware. *http://www.rtviz.com/*

Fast Volume Rendering:

- Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation by Philippe Lacroute and Marc Levoy