OpenGL Notes [a]

Stu Pomerantz

smp@psc.edu

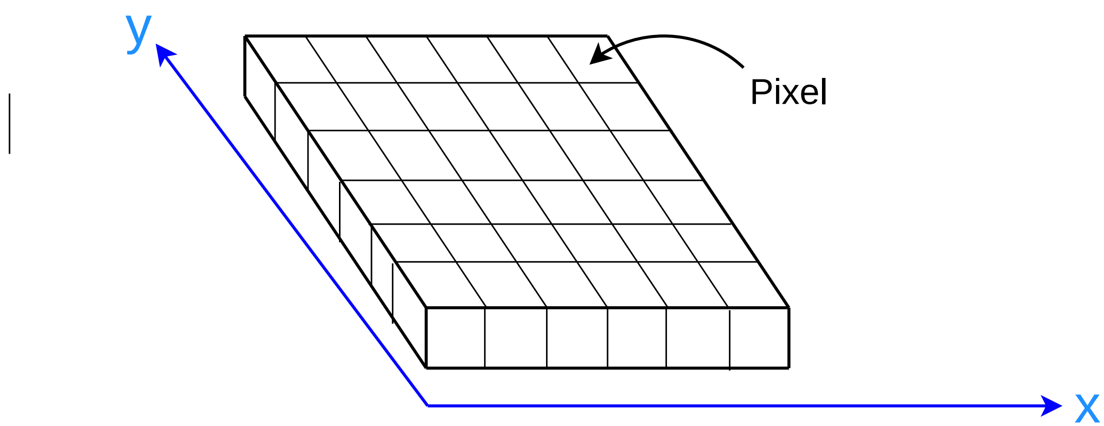
`http://www.psc.edu/~smp`

October 20, 2004

[a] Most material is adapted from: OpenGL ARB, et. al, "The OpenGL Programming Guide", Third Ed., Reading: Addison-Wesley, 1999

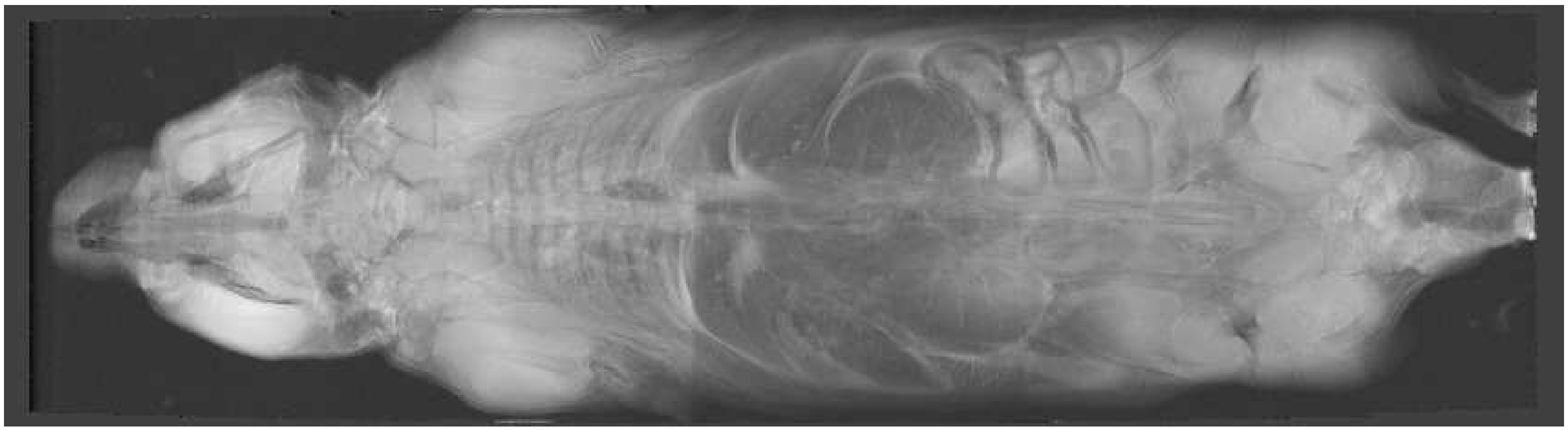
Image

- Can think of an image as a rectangular array values.
- Sometimes referred to as an image *plane* or *surface*.
- Values could be scalars or vectors (RGB colors for example).



Image

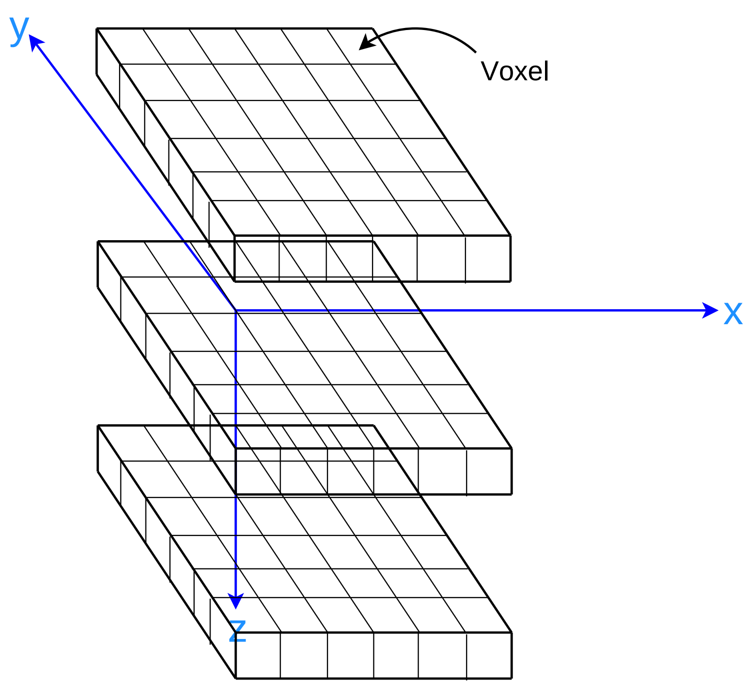
We know how to display images.



Mouse MRI, Duke Center for In-Vivo Microscopy (CIVM)

Volume

- Can think of a volume as a cubic array values.
- Many image planes "stacked" on top of one another.
- Values could be scalars or vectors (RGB colors for example).



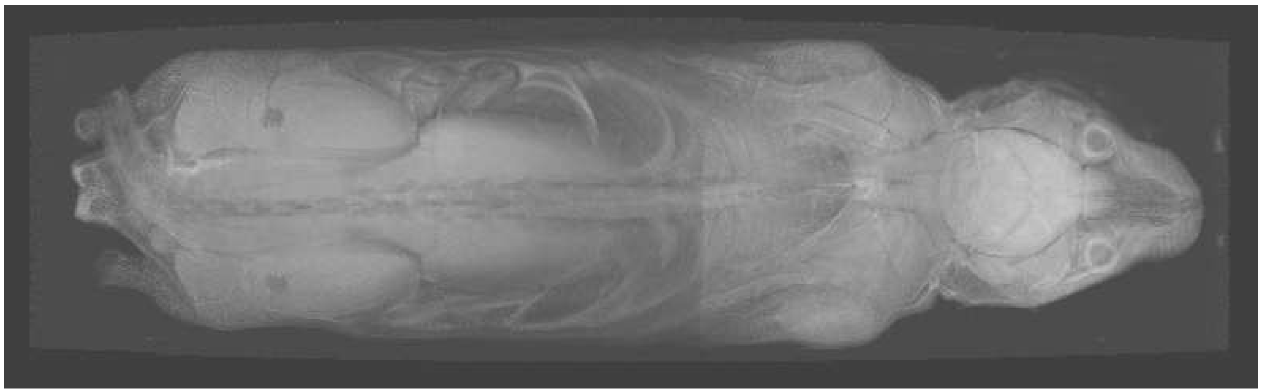
Volume

How can volume data be displayed ? Could just display each plane:

Two planes of the CIVM Mouse MRI data.

Volume Rendering

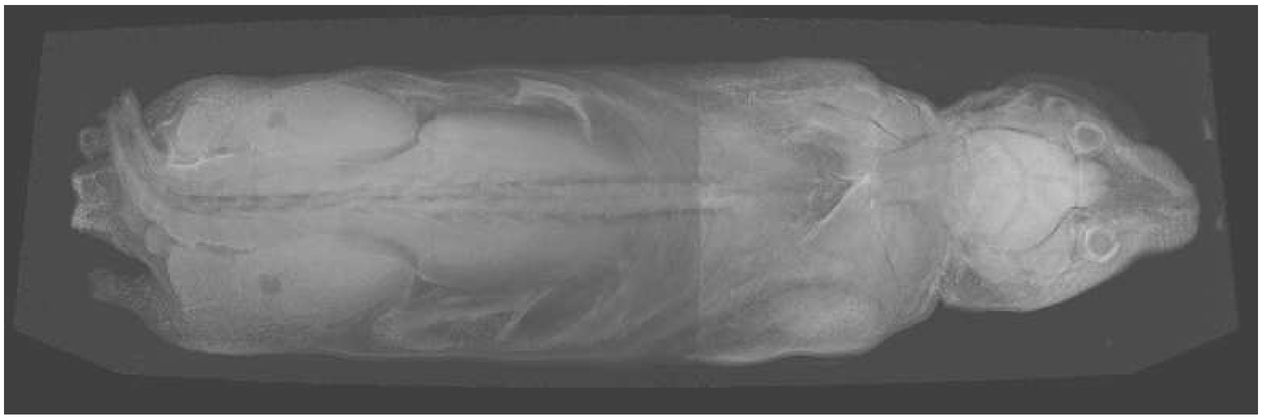
Another way do display volume data is *volume rendering*:



Volume Rendering of the CIVM Mouse MRI data.

Volume Rendering

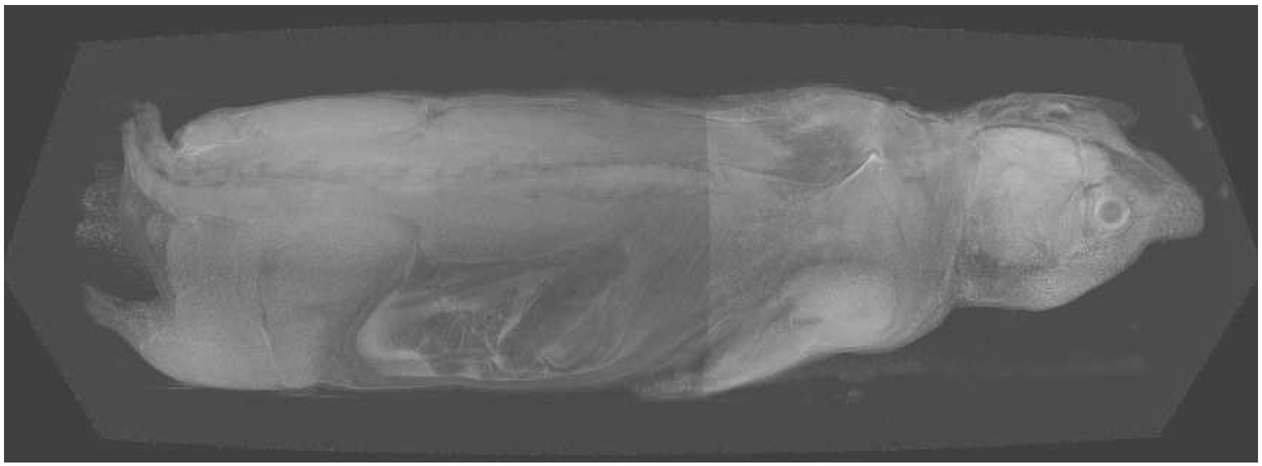
volume rendering:



Volume Rendering of the CIVM Mouse MRI data.

Volume Rendering

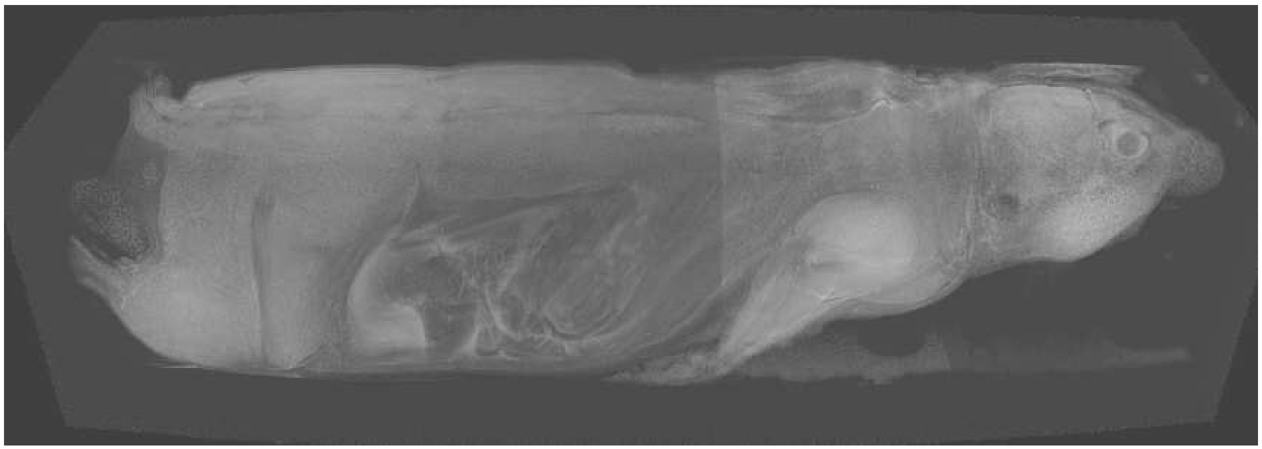
volume rendering:



Volume Rendering of the CIVM Mouse MRI data.

Volume Rendering

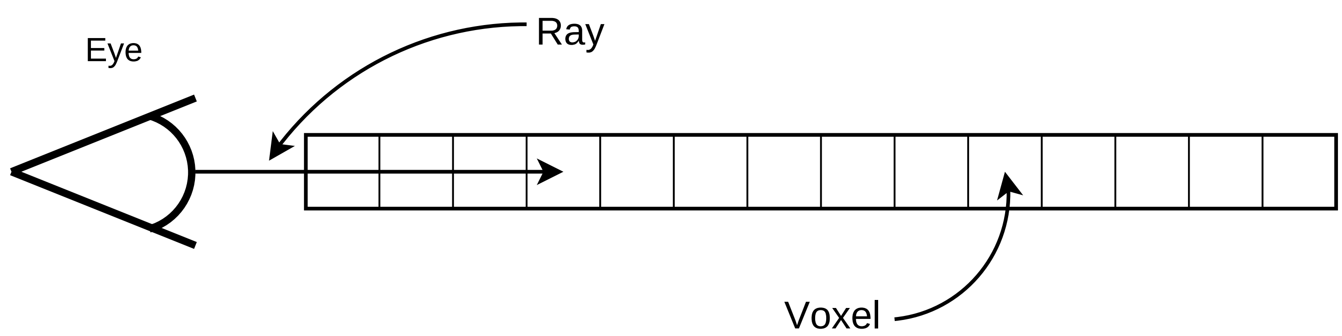
volume rendering:



Volume Rendering of the CIVM Mouse MRI data.

Volume Rendering

Volume rendering can be accomplished using a method called *ray casting*.



- Rays are projected from the eye into the volume.
- Voxels traversed by each *eye ray* may (or may not) contribute to the final image.
  - The contribution of each voxel depends on the light and material models.

Volume Rendering

The simplest model to consider is light absorption. In particular, imagine a single light emitting particle embedded in a medium that only absorbs (i.e. does not emit) light. This is how X-Ray and CAT work.

Let $I(s)$ be the light intensity at distance s through the medium. (This is what we ultimately want to compute).

Let I_0 be the intensity at $s = 0$ where the ray enters the volume.

Let $\tau(s)$ be the exinction coefficient. That is, it defines the rate that light is occluded.

Volume Rendering

The transparency of the medium between 0 and s is:

$$T(s) = e^{\int_0^s \tau(t)dt}$$

Finally, $I(s)$, the light intensity at distance s is:

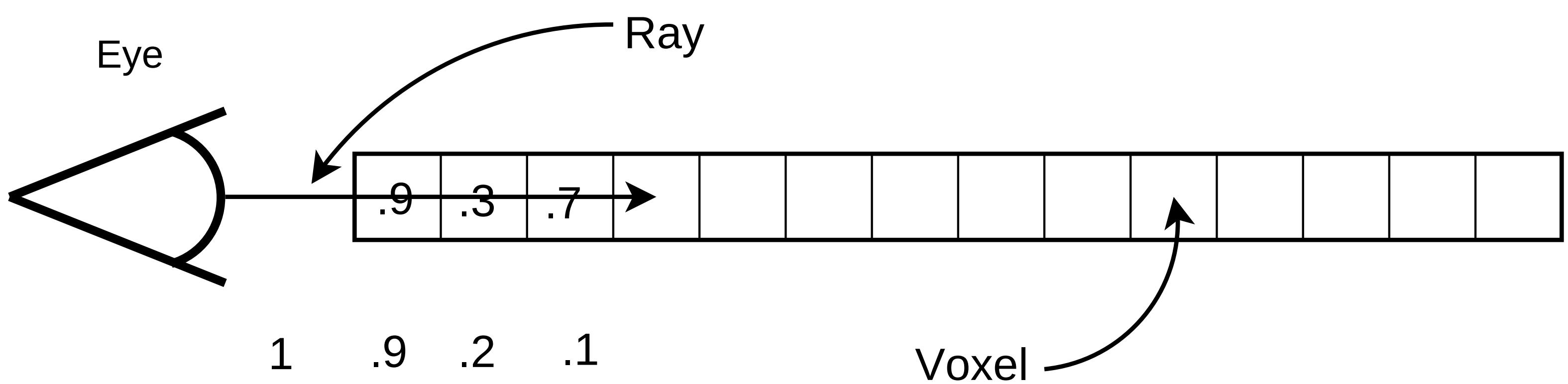
$$I(s) = I_0 \cdot T(s)$$

Reference: *Optical Models for Direct Volume Rendering*, Nelson Max.

Volume Rendering

It is important to note that this model is a physically correct model from the *emitter* point of view (not the eye point of view).

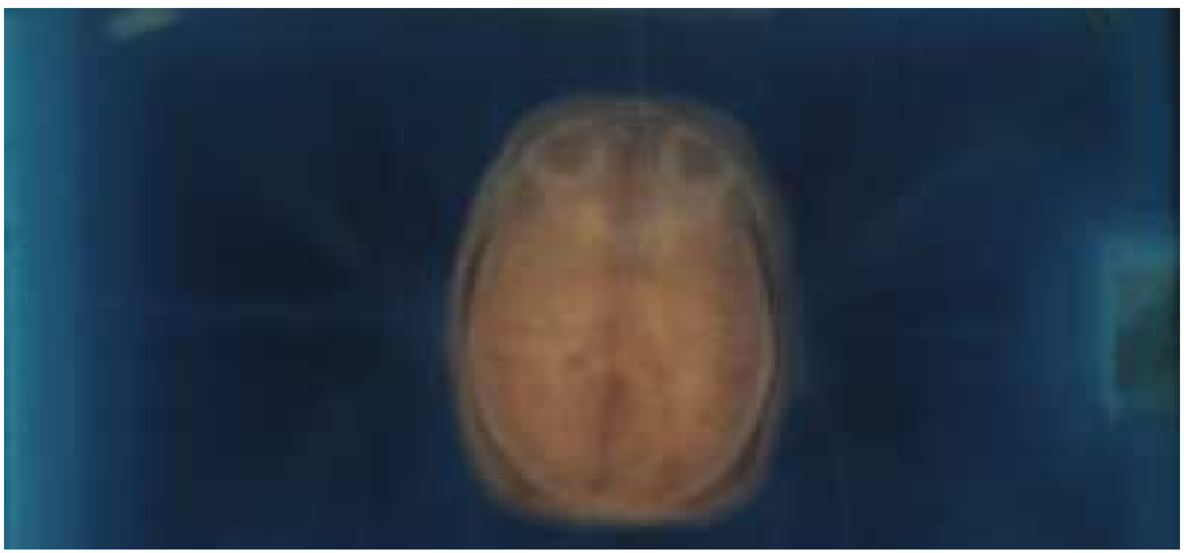
In volume rendering we use this model since it will produce a nice *rendition* but we *are not simulating* anything.



Volume Rendering

The light and material models rely on a *transfer function* which classifies voxels.

The NLM Visible Human Female, using a simple material and light model which averages voxels based on distance from the eye:



The brain.

Volume Rendering

The NLM Visible Human Female, using a simple material and light model which averages voxels based on distance from the eye:

Between the head and chest.

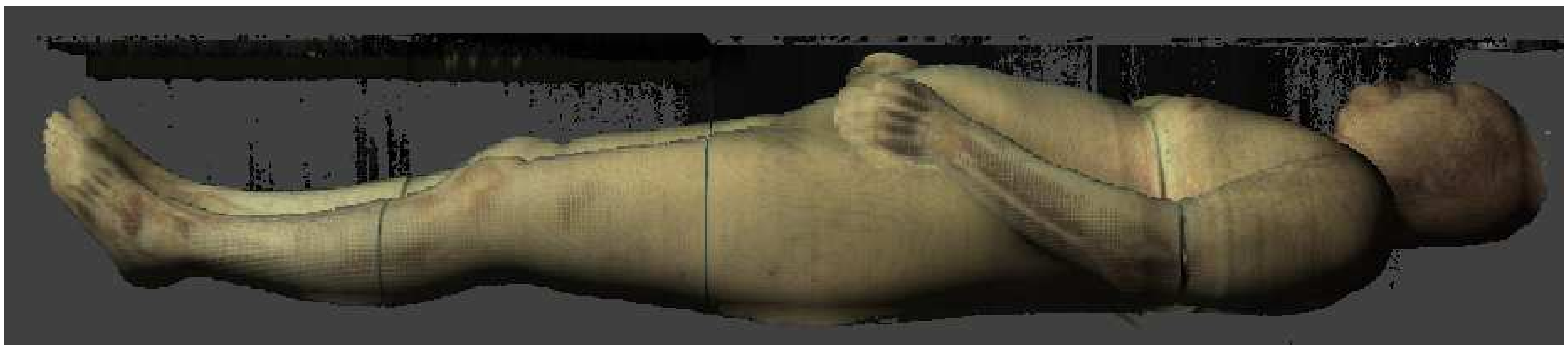
Volume Rendering

The NLM Visible Human Female, using a simple material and light model which averages voxels based on distance from the eye:

The chest cavity.

Volume Rendering

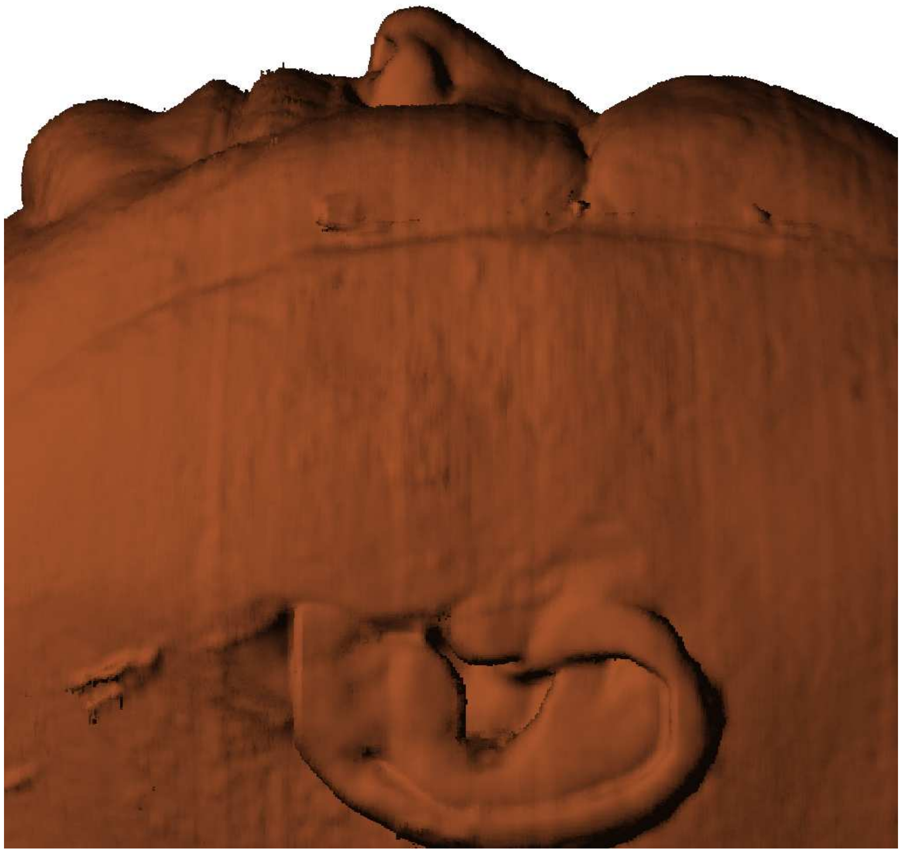
The NLM Visible Human Female, using a more sophisticated material and light model which penetrates the surface of the skin by only a few voxels and computes diffuse surface lighting:



This method is akin to *iso-surfacing*.

Volume Rendering

Non-photorealistic effects can also be achieved.

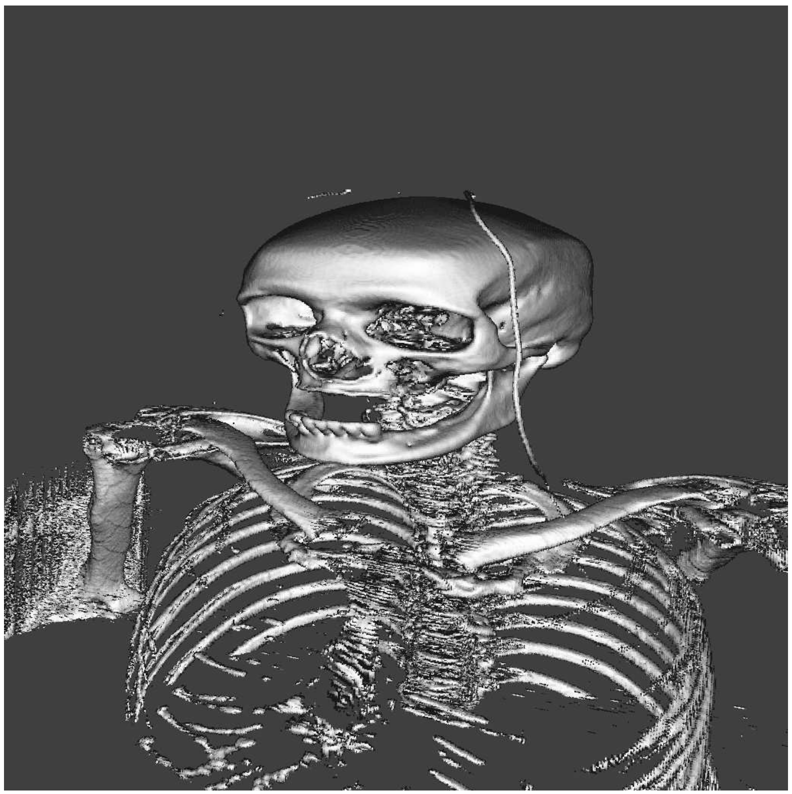


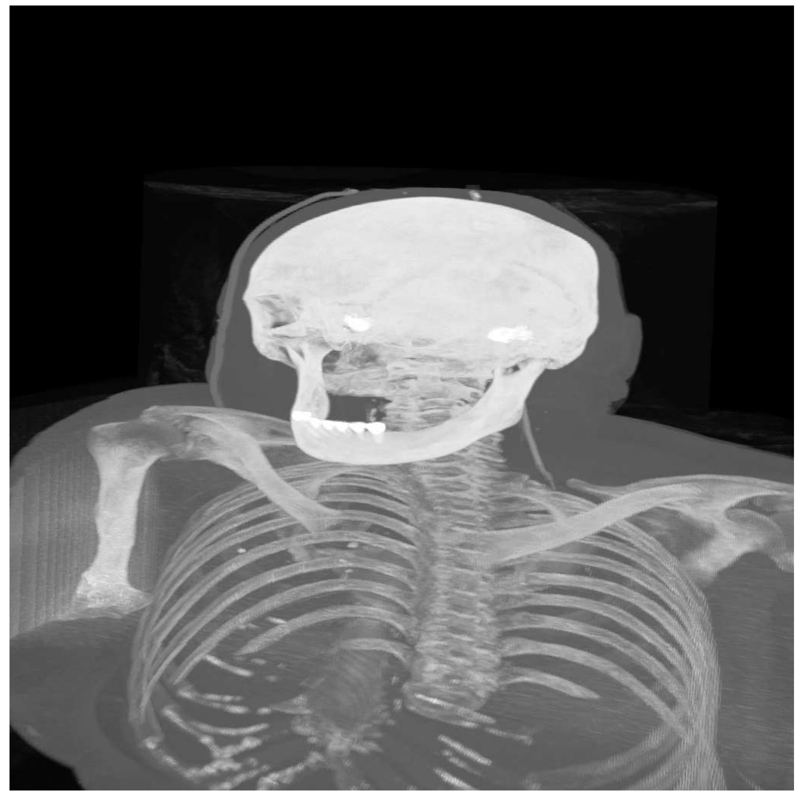
Volume Rendering

Non-photorealistic and realistic, combined.

Volume Rendering

Maximum Intensity Projection: Only the voxel with the largest intensity encountered by the ray projects back to the image.





Volume Rendering

Computation of surface normals:

- Must estimate the surface normal of the object is at every voxel struck by a ray.
- Which voxels represent surface and which represent space is determined by the transfer function.

1. Sample space in a small neighborhood of each voxel which hits the surface.
2. Average the 'inside' voxels and the 'outside' voxels.
3. The estimated normal is the vector from the inside average to the outside average.

Volume Rendering and OpenGL

Volume rendering can be done with OpenGL using *3D Texturing*.

Unfortunately, the numbers are against it.

A high-end graphics card might have as much as 512Mbyte or even a 1Gbyte of RAM.

Consumer cards these days have 128Mbyte or 256Mbyte of RAM.

Volumetric datasets grow very large quickly.

Suppose $x =$ length of the a side of a square image or a cubic volumetric dataset.

Then...

The Power of Powers

x	x^2	x^3
1	1	1
2	4	8
4	16	64
8	64	512
16	256	4096 (4K)
32	1024 (1K)	32767 (32K)
64	4096 (4K)	262144 (256K)
128	16384 (16K)	2097152 (2M)
256	65536 (64K)	16777216 (16M)

The Power of Powers

x	x^2	x^3
256	65536 (64K)	16777216 (16M)
512	262144 (256K)	134217728 (128M)
1024	1048576 (1M)	1073741824 (1G)
2048	4194304 (4M)	8589934592 (8G)
4096	16777216 (16M)	68719476736 (64T)
8192	67108864 (64M)	549755813888 (512T)
16384	268435456 (256M)	4398046511104 (4P)
32768	1073741824 (1G)	35184372088832 (32P)
65536	4294967296 (4G)	281474976710656 (256P)

The Power of Powers

- A 2048^3 volume is 8 Gigavoxels, far larger that graphics card memory. Still larger that (typical) workstation RAM.
- The Visible Human Female dataset (created *10 years ago*) is roughly 8 Gigavoxels, but each voxel is an RGB vector, so the dataset is 24 Gigabytes.
- Large dataset sizes also have a substantial impact on run time.

How to overcome these limits ?

Overcoming the Power of Powers

Some strategies to accelerate volume rendering:

- Not all rays touch all voxels. Take advantage of spacial coherence to load only the data you need.
- The ray casting calculation is "pleasantly parallel" so write your volume render to run in parallel.
- In some cases, only a relatively small proportion of the voxels are important (based on the *transfer function*). Use bounding boxes to do quick tests to cull rays that miss the important data.