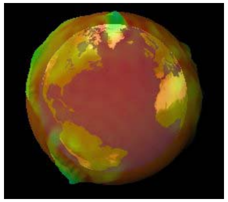
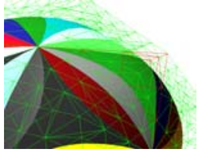
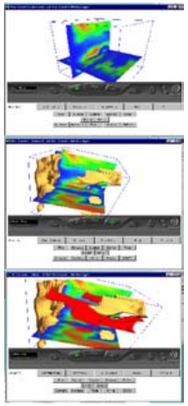


Computational Visualization

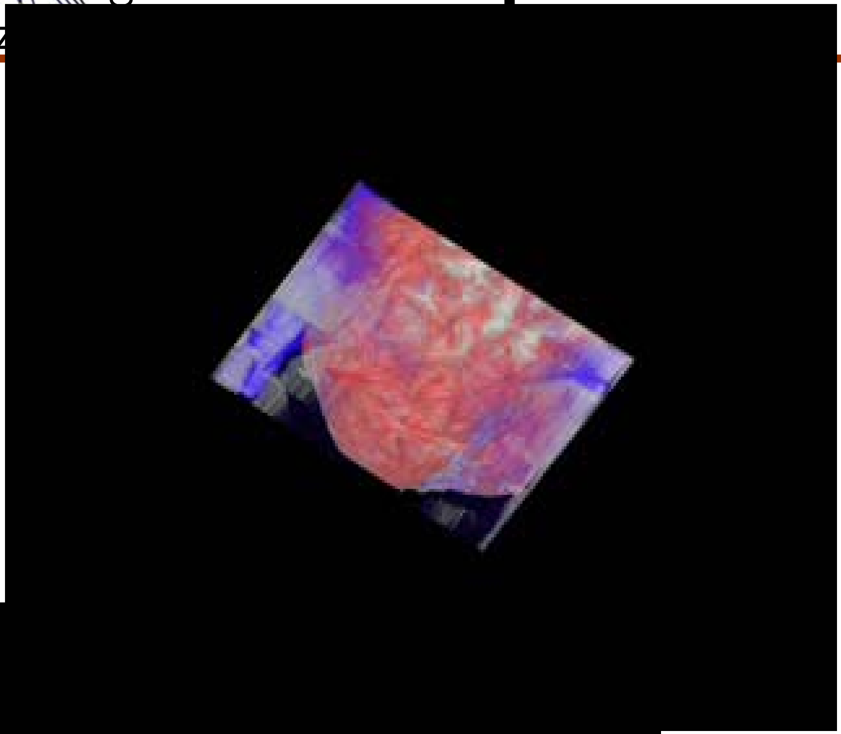
1. Sources, characteristics, representation 
2. Mesh Processing 
3. Contouring 
4. Volume Rendering 
5. Flow, Vector, Tensor Field Visualization 
6. Application Case Studies

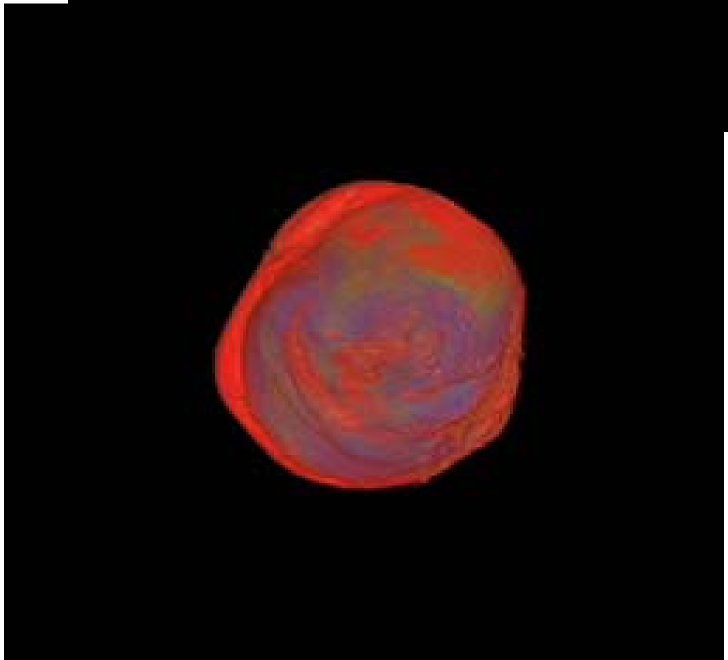
Computational Visualization:

Volume Rendering

Lecture 4

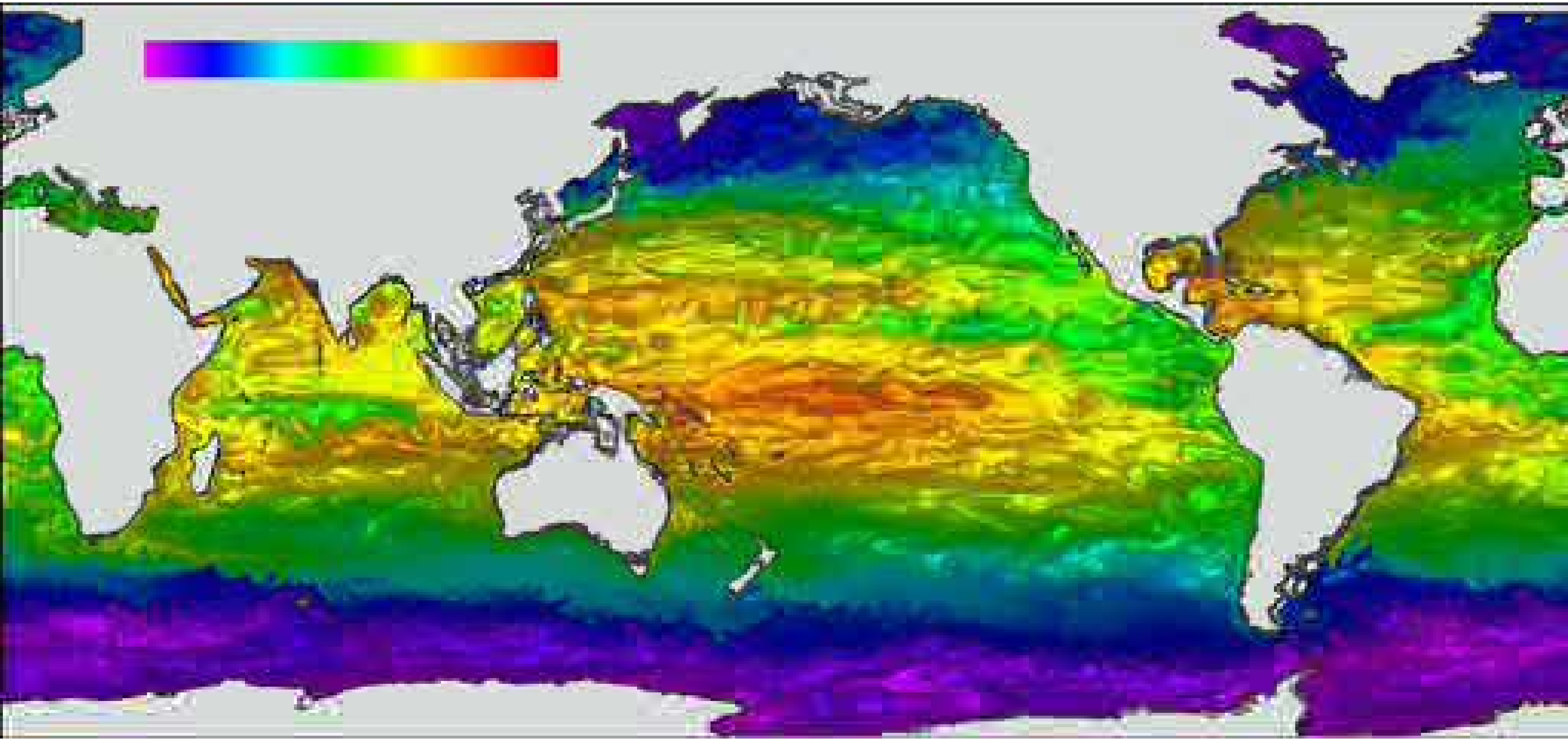
Example Volume Renderings





Oceanographic Simulations

- $2160 \times 960 \times 30 \times 4$(bytes) = 237 MB
- 237(MB)$\times 115$(timesteps) = 27 GB



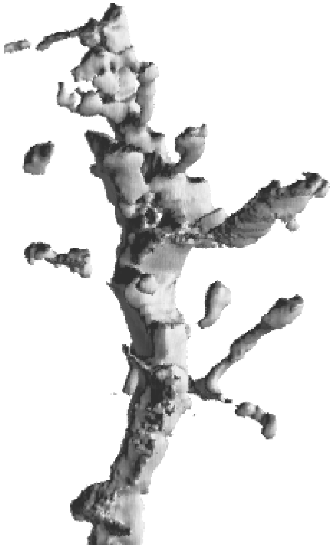
Outline

- Ray Casting/Shading
- Opacity weighted Color Integration
- Volumetric Illustration
- Texture Based Rendering (Hardware Acceleration)
- Optical Models (Gaseous Phenomena)

First Principles

Volume Rendering Algorithm

- Direct volume rendering
  - Ray-casting
  - Splatting
- Indirect volume rendering
  - Fourier
- Texture based volume rendering
  - 3D Texture mapping hardware

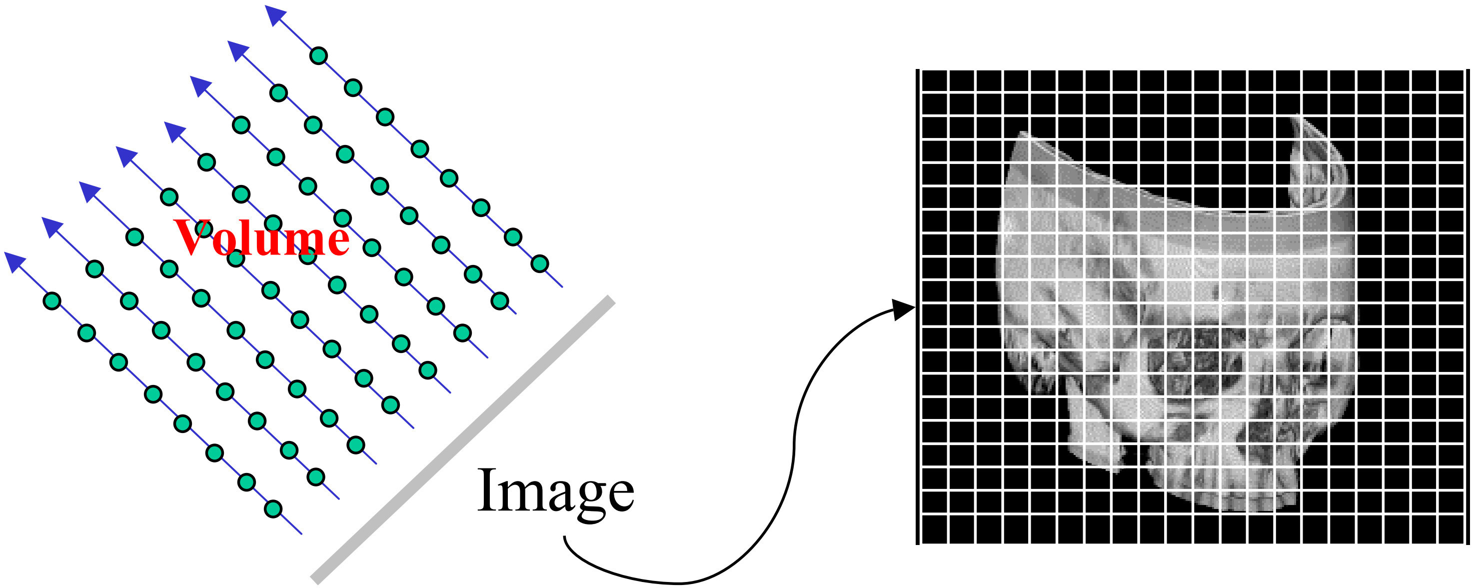


(a) Isosurface Rendering

(b) Direct Volume Rendering

Ray-Casting

View dependent



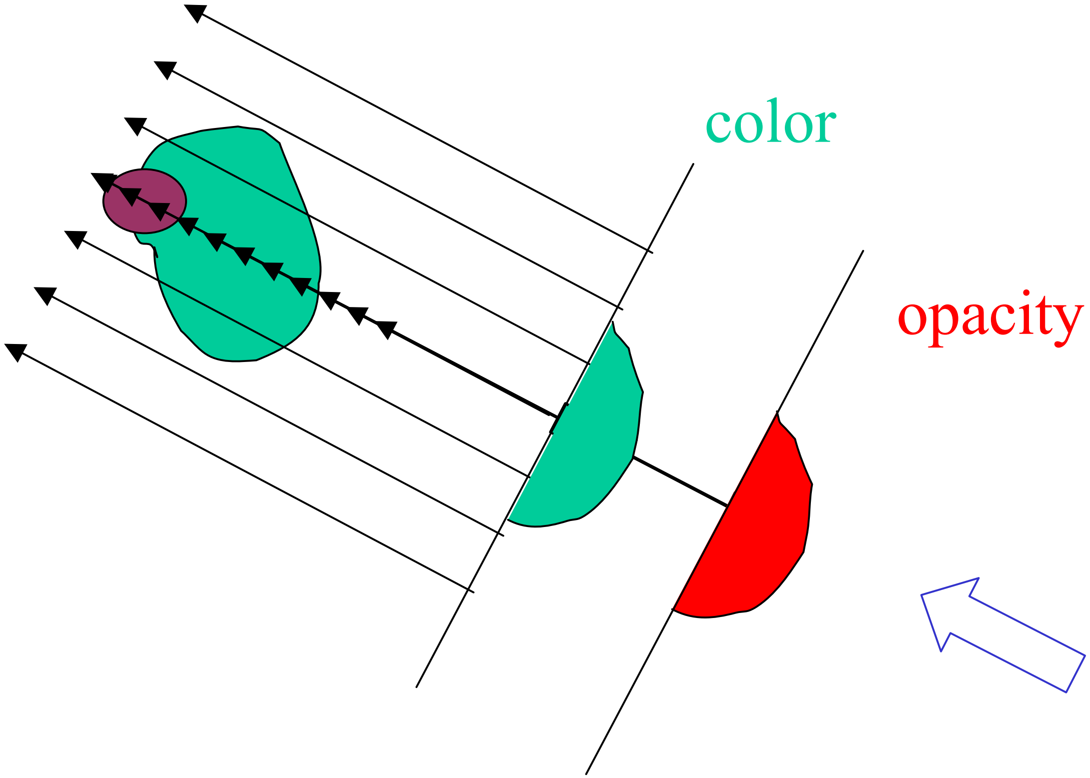
Ray-Casting (cont)

- Advantages
  - Not necessary to explicitly extract surfaces from volume when rendering
  - Can change the transfer functions to make various surfaces stand out within the volume

Ray-Casting (cont)

- Disadvantages
  - Do not have explicit representations for surfaces, therefore not straightforward to compute integral/differential properties
  - Much more computationally intensive to render volume since not dealing directly with the efficient polygon pipeline

Volumetric Ray Integration



object (color, opacity)

Given Colors or Shade Before Resampling

Data comes as color/opacity

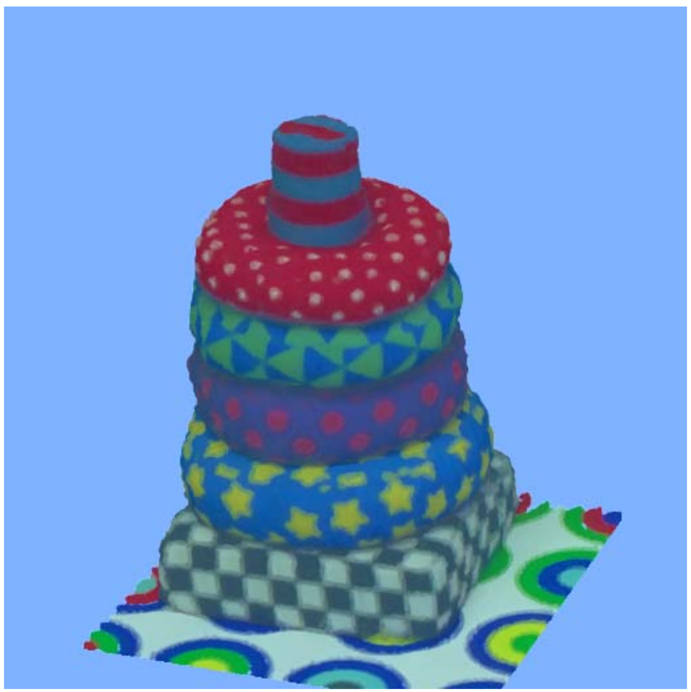
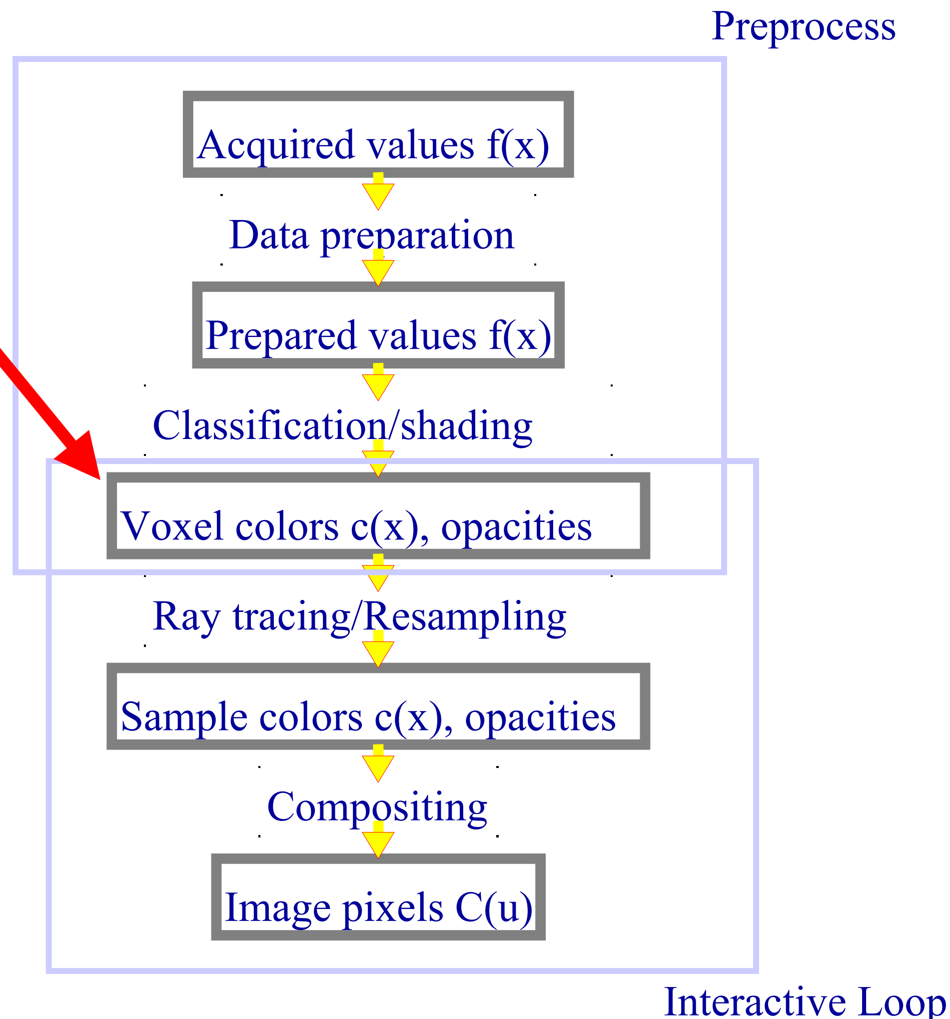
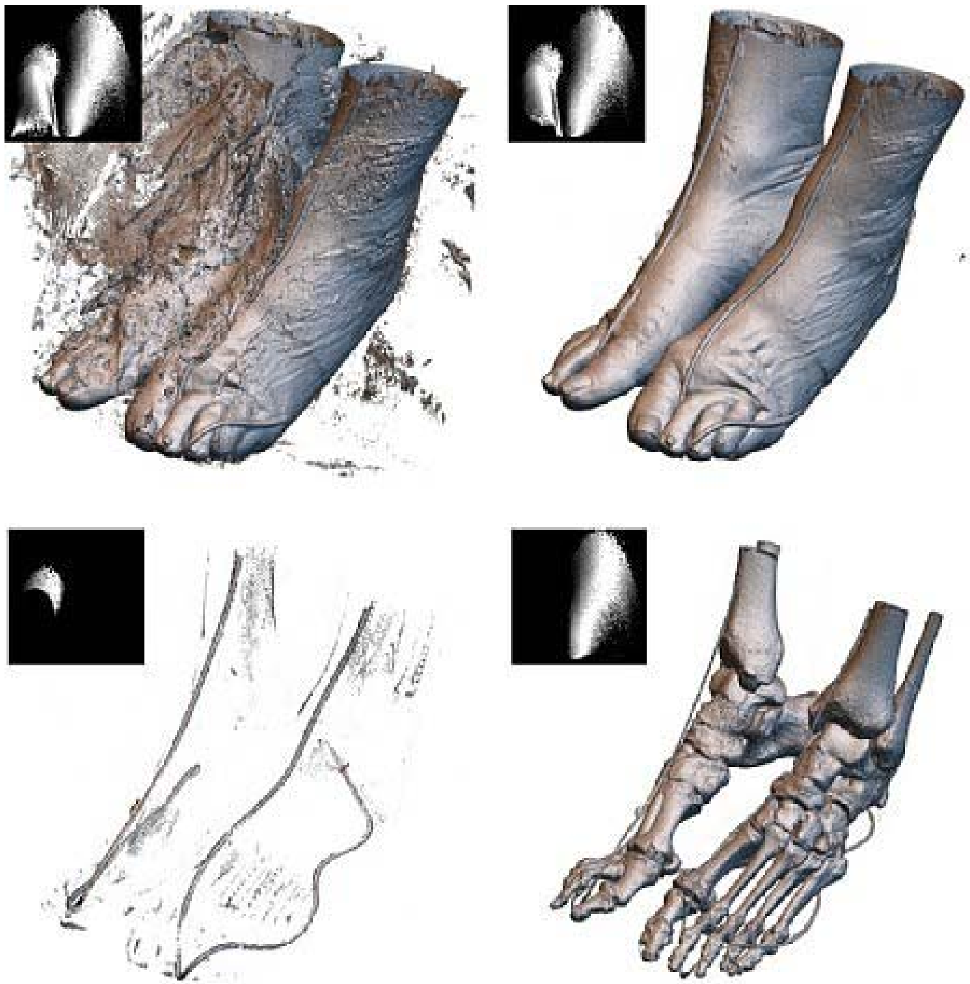


Image based rendering outputs colors

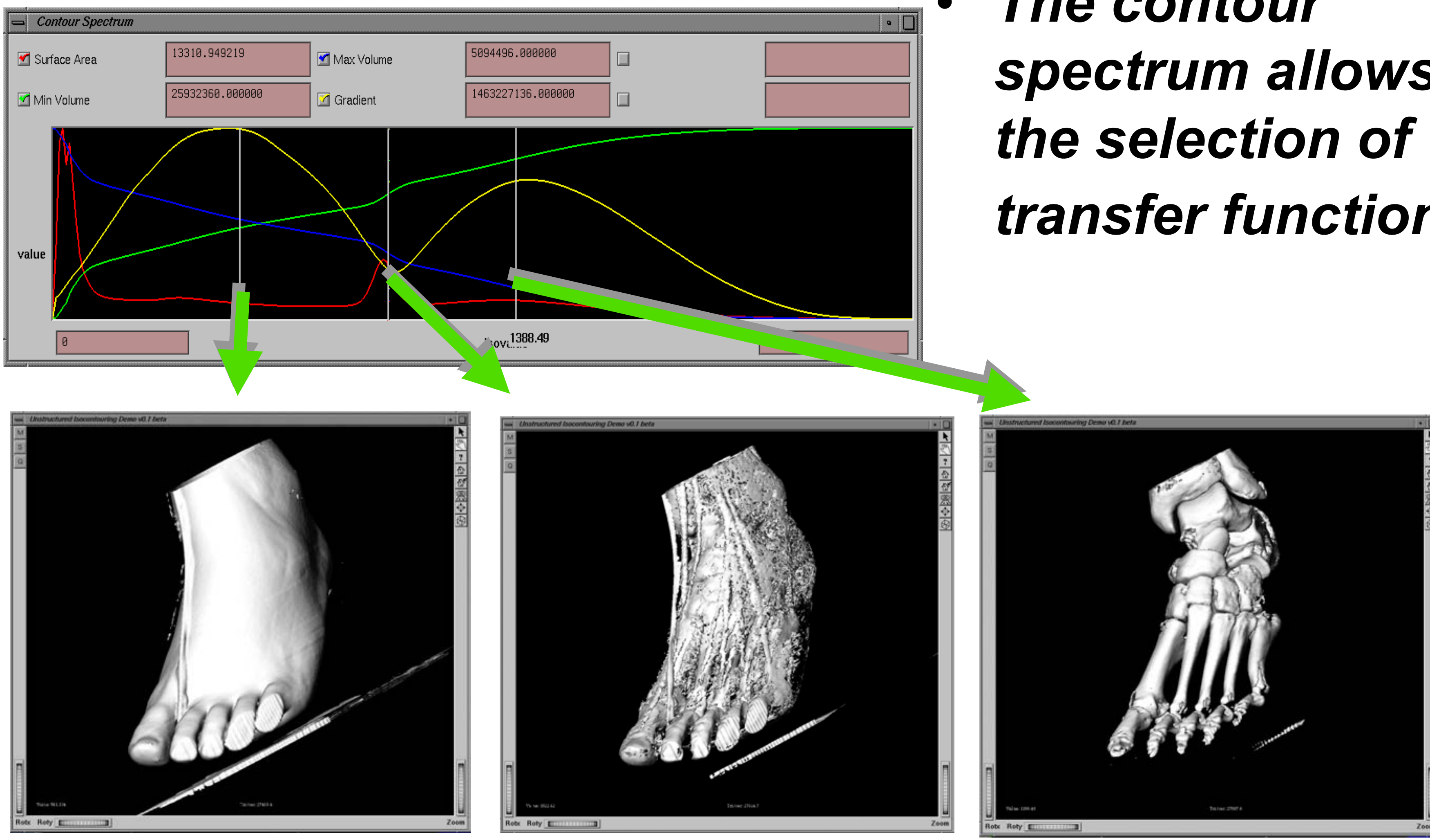


Transfer Functions

- Mapping from data values to renderable optical properties
  – Density
  – Gradient



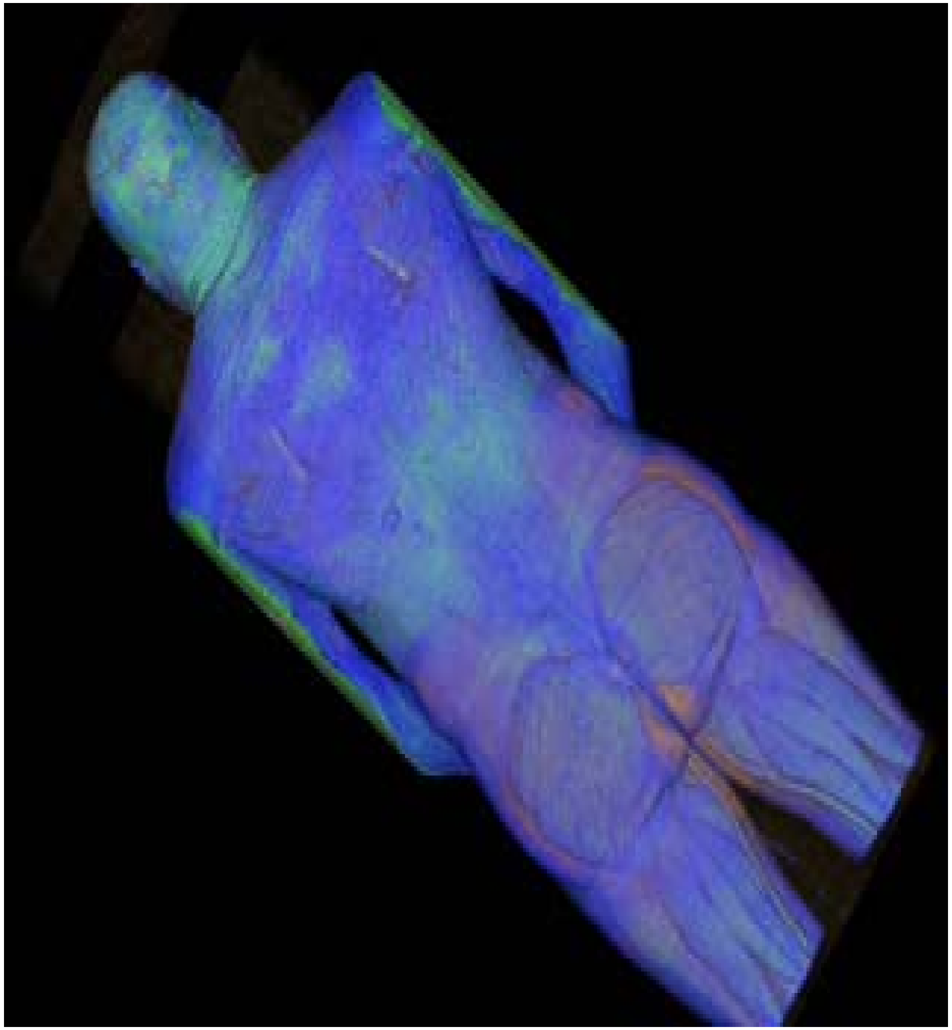
The Contour Spectrum



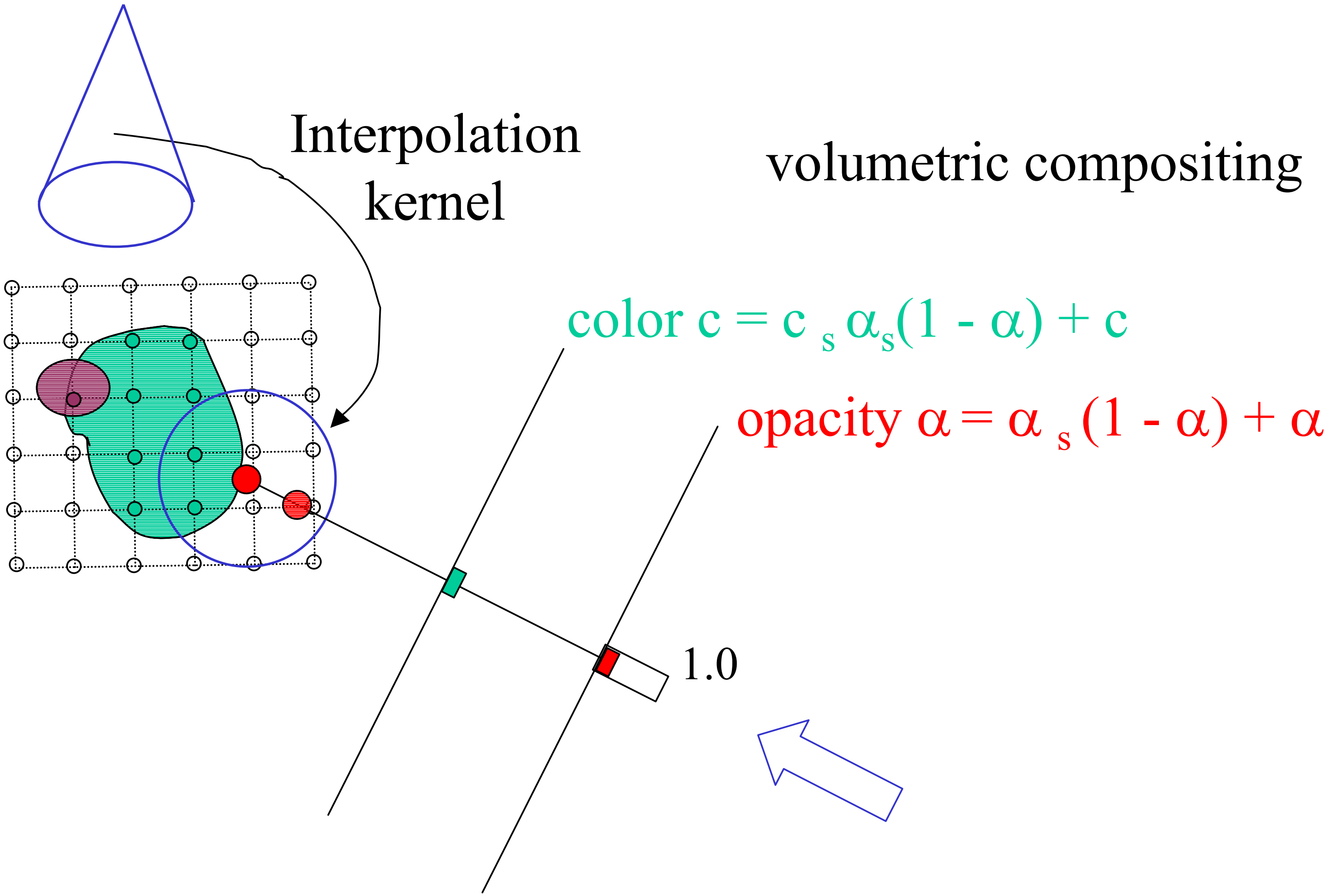
- ***The contour spectrum allows the selection of transfer functions***

Medical Data

(512×512×1871×2(bytes) = 936 MB)



Ray-casting - revisited



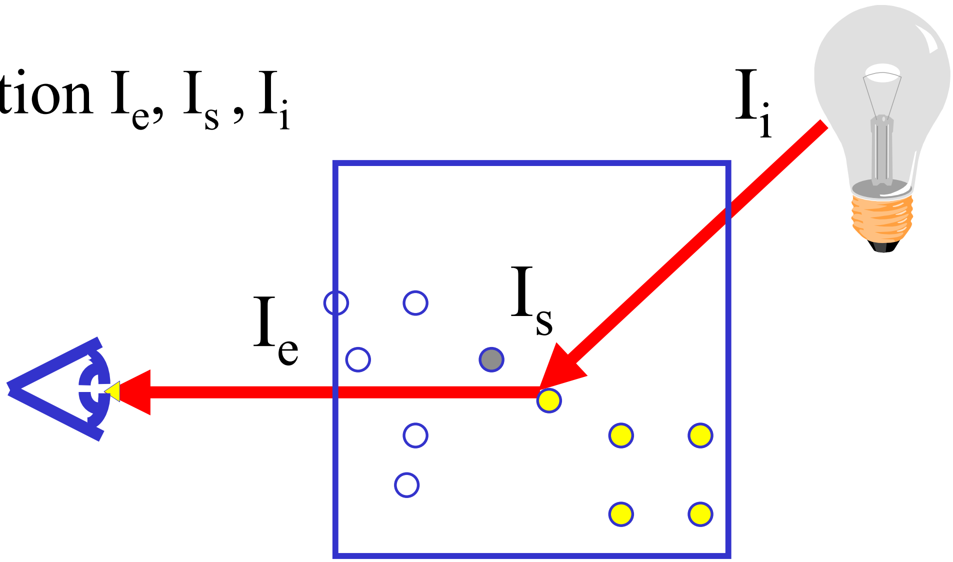
object (color, opacity)

Opacity-Weighted Color

1. From first principles, emitted intensity different from shaded intensity
2. From Blinn, Opacity-Weighting before interpolation helps quality
3. From short cut, cannot do separate interpolation

Derivation from First Principles of Volume Rendering

•Actually change notation I_e, I_s , I_i



1 region in volume

Ray intensity by line integral

$$I_{ray} = \int t(l) I_s(l) \alpha(l) dl$$

$$I_e(l) = I_s(l) \alpha(l)$$

Blinn's Associated Colors

- Associated color, opacity associated or multiplied
- Generalized to Volume Rendering $\tilde{C} = \alpha C$
- Compositing Equations

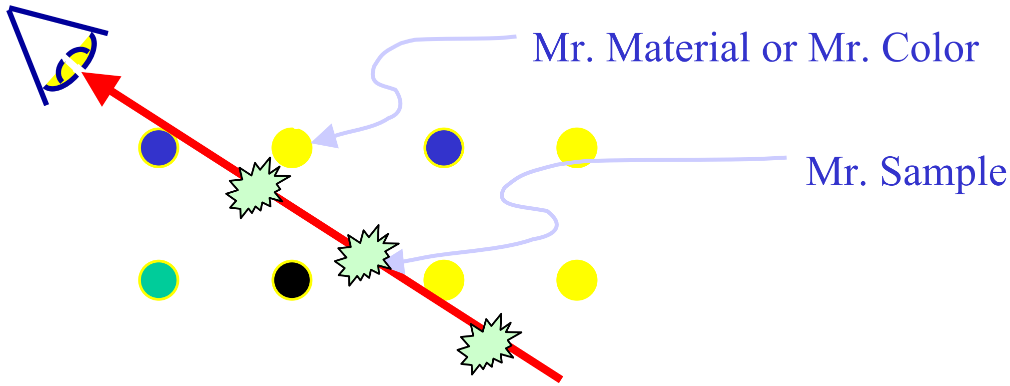
$$\tilde{C}_{\text{new}} = (1-\alpha_{\text{front}})\tilde{C}_{\text{back}} + \tilde{C}_{\text{front}}$$
$$\alpha_{\text{new}} = (1-\alpha_{\text{front}})\alpha_{\text{back}} + \alpha_{\text{front}}$$

See Blinn, SIGGRAPH'82,
Porter and Duff, SIGGRAPH'84
Blinn IEEE CGA, Sep. 1994.
See Drebin et al. SIGGRAPH'88

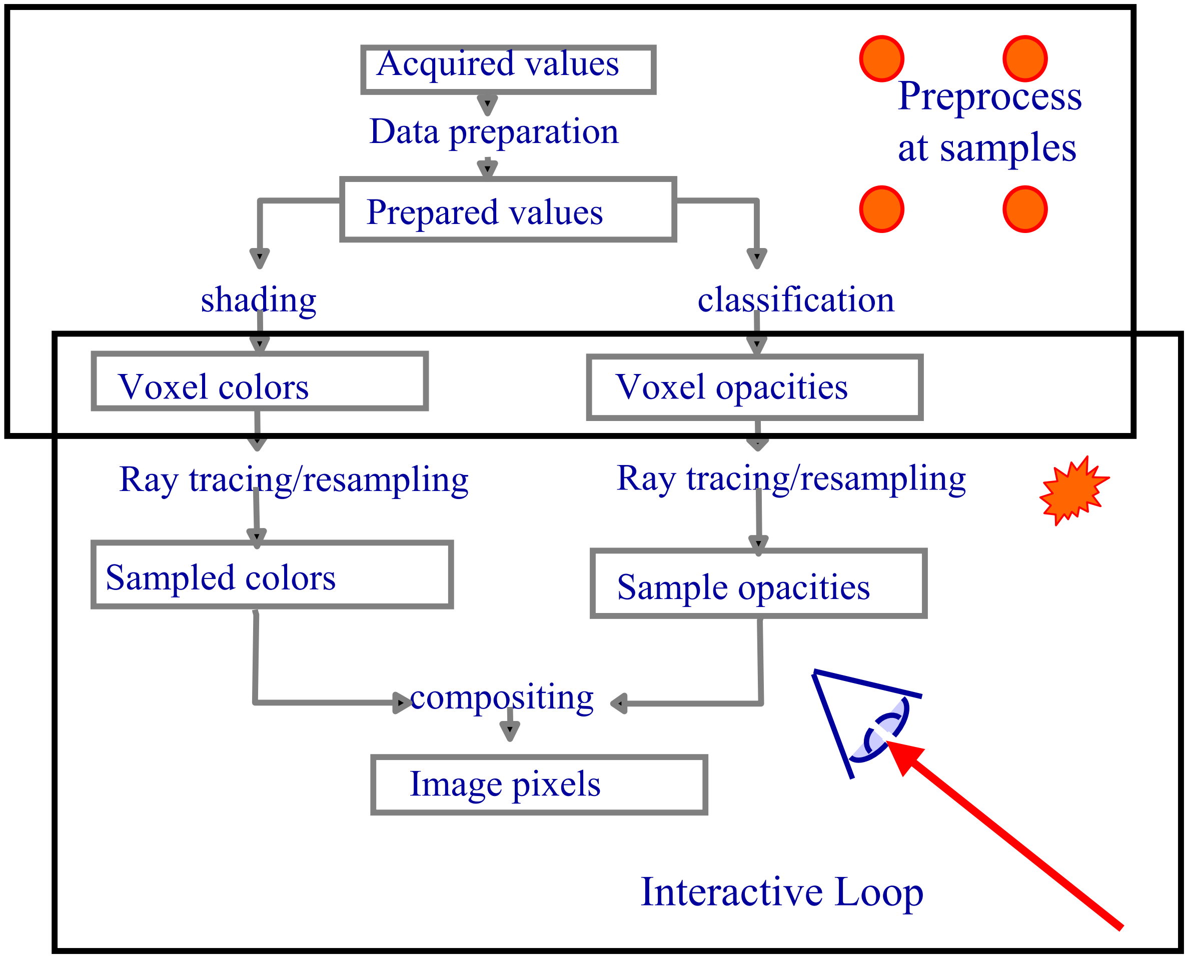
Works for back-to-front, front-to-back, parallel, etc.

A Shortcut to Represent Materials and Shading

- Assume that shading at material samples will give good results
- Levoy: separate interpolation of colors and opacities
- Pre-shade



Separate Interpolation of Colors and Opacities (Levoy '88)



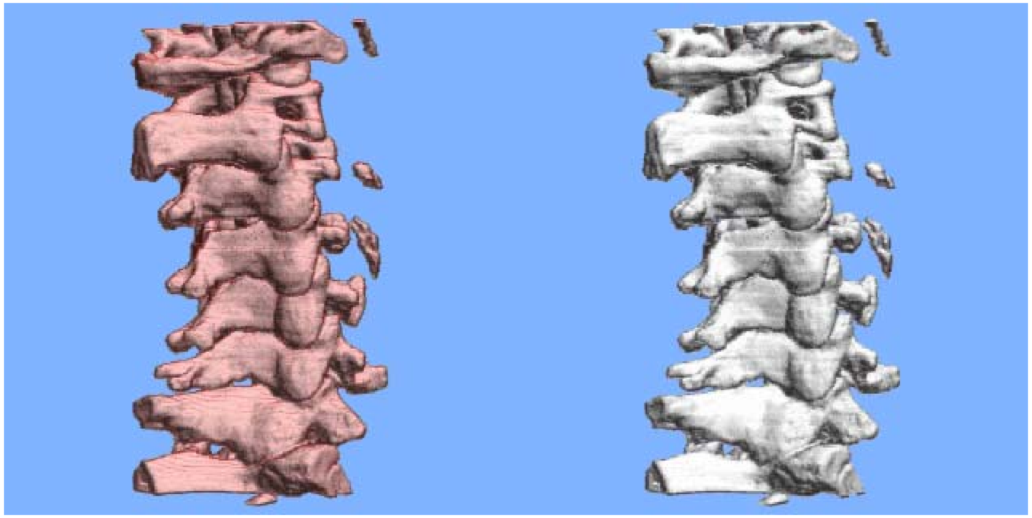
$$C_{out} = C_{in} \cdot \alpha_{in} \cdot (1-\alpha) + C$$
$$\alpha_{out} = \alpha_{in} \cdot (1-\alpha) + \alpha$$

Which one?

$$C_{out} = C_{in} \cdot (1-\alpha) + C \cdot \alpha$$
$$\alpha_{out} = \alpha_{in} \cdot (1-\alpha) + \alpha$$

Opacity-Weighted Color Interpolation

C. M. Wittenbrink, T. Malzbender, and M. E. Goss, Opacity-Weighted Color Interpolation for Volume Sampling, Volume Visualization Symposium ’98, Research Triangle Park, NC, 1998.



Opacity-Weighted Interpolation (Wittenbrink et. al. 98)

FTB color:

$$\tilde{C}_{\text{front(new)}} = (1-\alpha_{\text{front}})C_{\text{back}}\alpha_{\text{back}} + \tilde{C}_{\text{front(old)}}$$

BTF color:

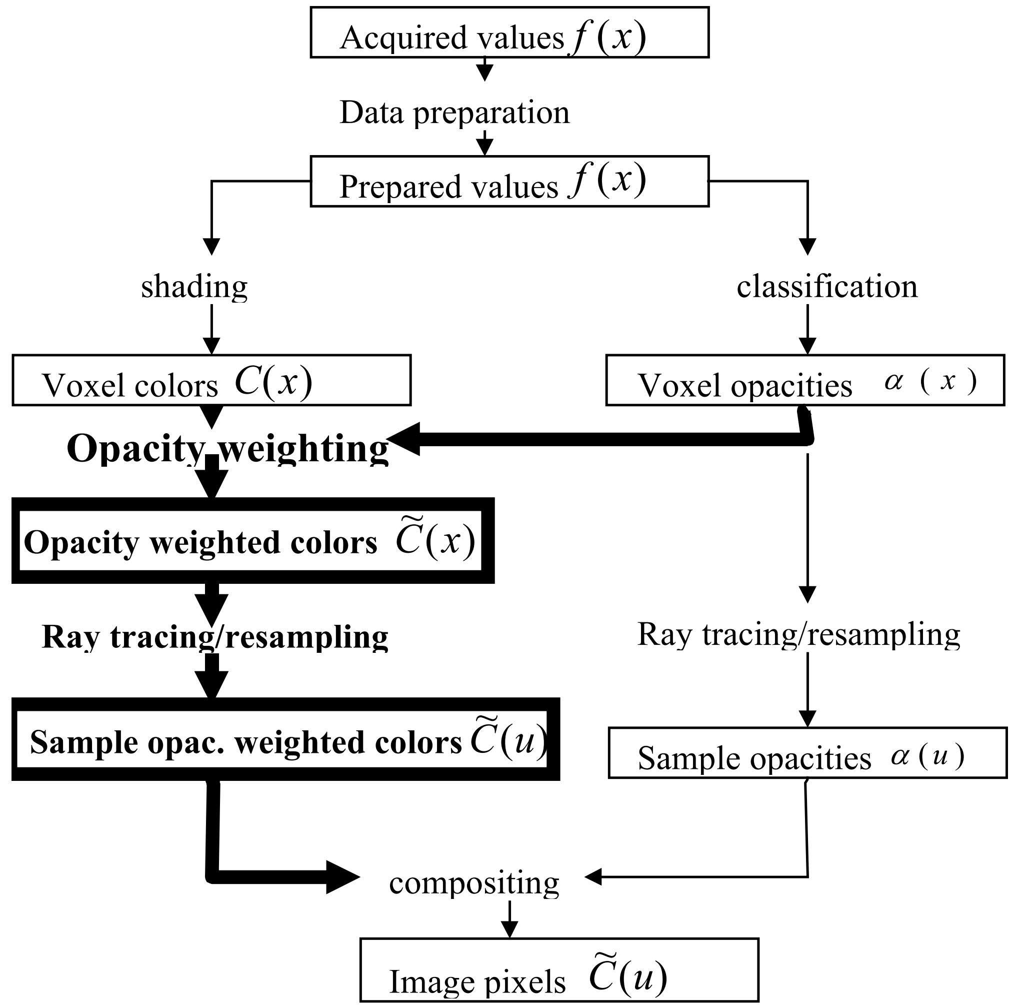
$$\tilde{C}_{\text{back(new)}} = (1-\alpha_{\text{front}})\tilde{C}_{\text{back(old)}} + C_{\text{front}}\alpha_{\text{front}}$$

Opacity:

$$\alpha_{\text{new}} = (1-\alpha_{\text{front}})\alpha_{\text{back}} + \alpha_{\text{front}}$$

The colors that are composited must be pre-weighted with opacity, i.e. associate color:

$$C' = \alpha C$$



Example Calculation

Separate

$$\widetilde{C}_{12} = (1-\alpha_1)C_2\alpha_2 + C_1\alpha_1$$
$$= (1-0)\times 0.5\times 0.5 + 0\times 0 = 0.25$$

$$\alpha_{12} = (1-\alpha_1)\alpha_2 + \alpha_1$$
$$= (1-0)\times 0.5 + 0 = 0.5$$

$$\widetilde{C}_{123} = (1-\alpha_{12})C_3\alpha_3 + \widetilde{C}_{12}$$
$$= (1-0.5)\times 1\times 1 + 0.25 = 0.75$$

Opacity-weighted

$$\widetilde{C}_{12} = (1-\alpha_1)\widetilde{C}_2 + \widetilde{C}_1$$
$$= (1-0)\times 0.5 + 0 = 0.5$$

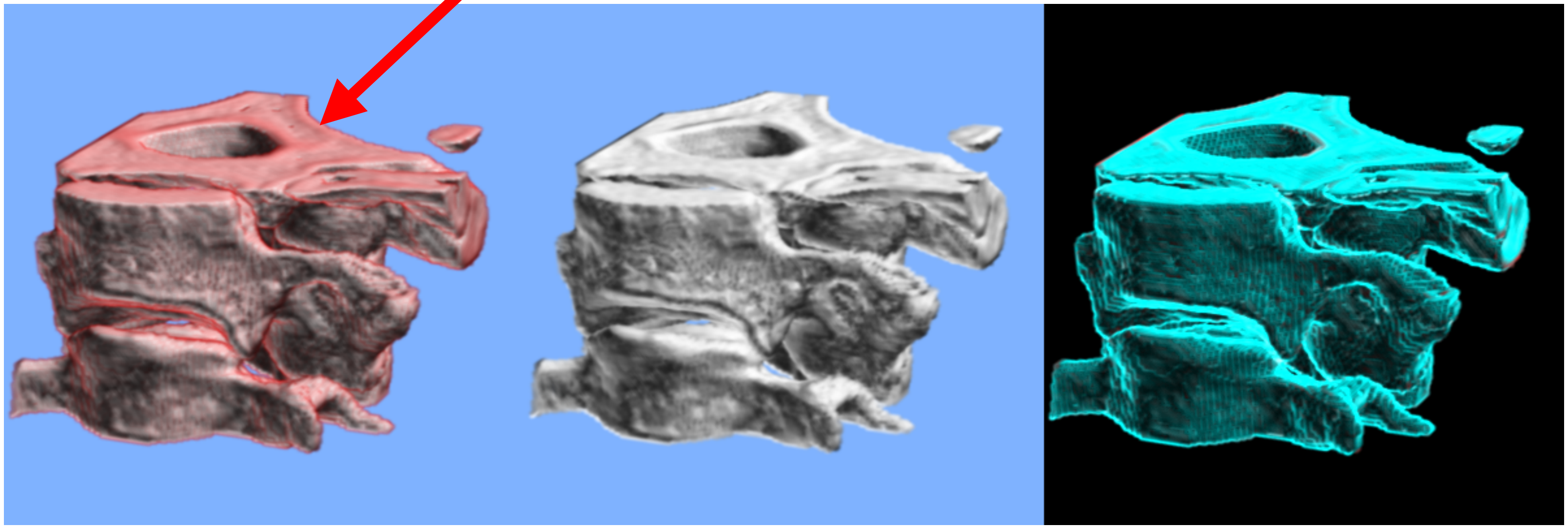
Different color

$$\widetilde{C}_{123} = (1-\alpha_{12})\widetilde{C}_3 + \widetilde{C}_{12}$$
$$= (1-0.5)\times 1 + 0.5 = 1$$

Rendering Comparison

Red tissue bleeds onto white bone

Color errors



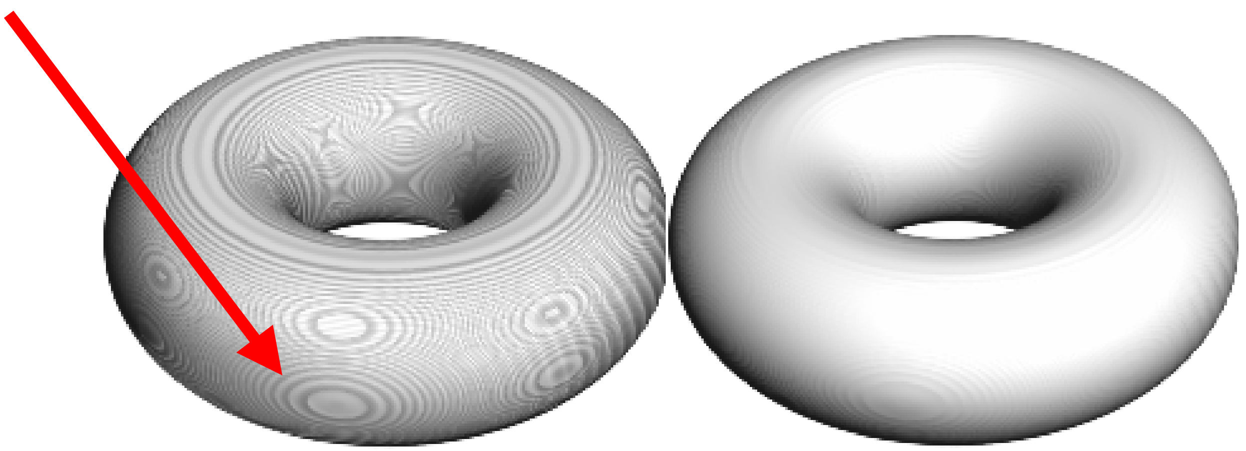
Separate   Opacity-weighted   Difference

100x96x249 spiral CT dataset, classified to 8 bit

Rendering Comparison (cont)

Intensity errors

Banding results from
black air marking surface

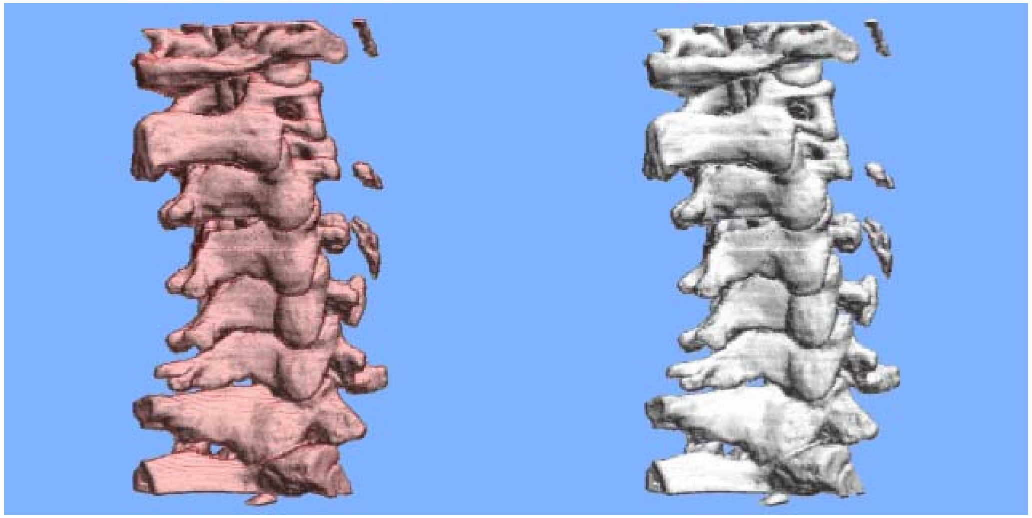


Separate Opacity-weighted

Torus volume, pre-antialiased

Spiral CT Rendering Comparison

Artifact appears to be aliasing   Color & intensity errors



Separate   Opacity-weighted

Summary: Opacity-Weighted Color Interpolation

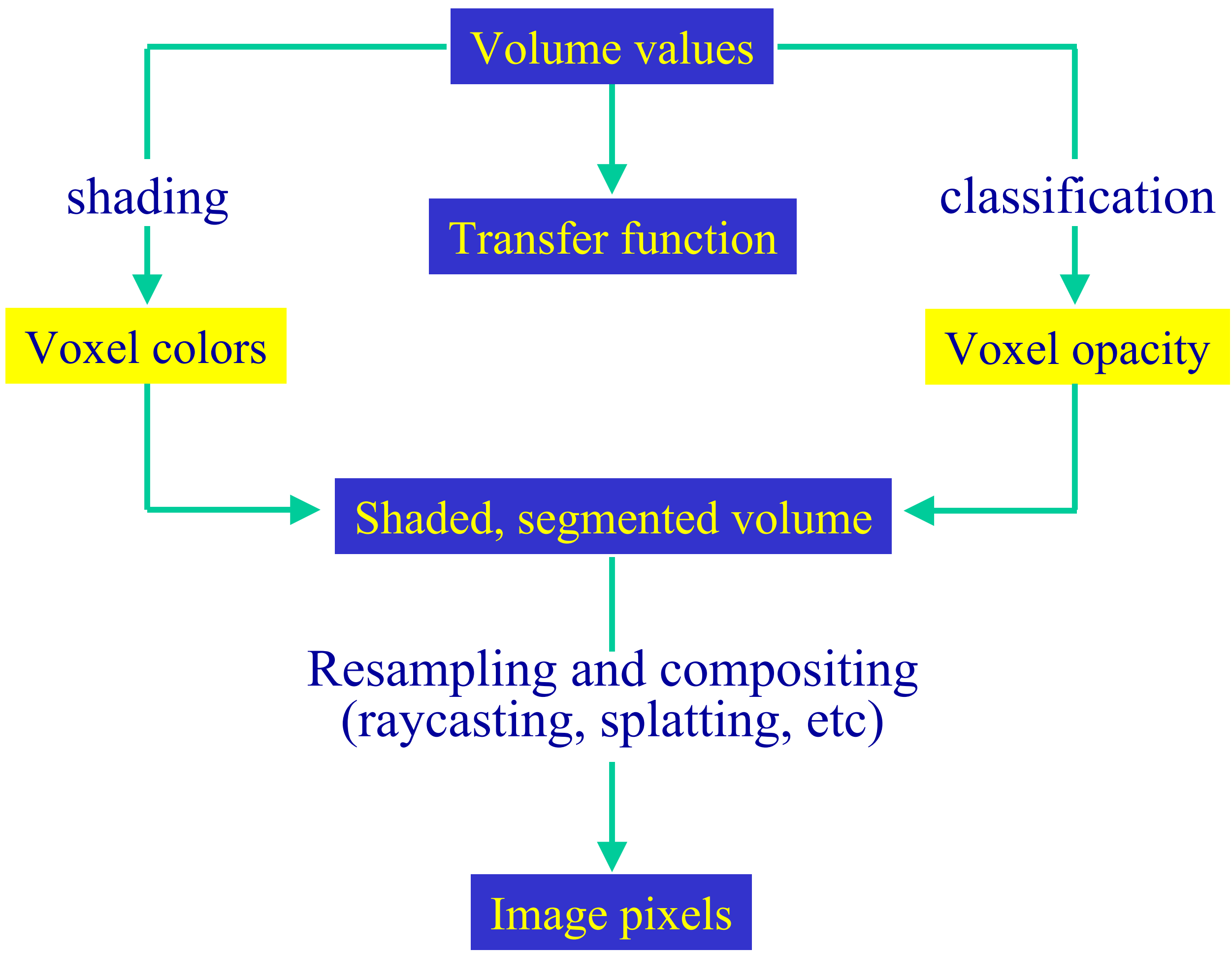
- Opacity-weight $\omega_i = w_i \alpha_i$
- Compute ray sample opacity $\alpha = \sum_i \omega_i$
- Compute ray sample color $\tilde{C} = \sum_i \omega_i C_i$
- Composite

$$\tilde{C}_{\text{new}} = (1 - \alpha_{\text{front}})\tilde{C}_{\text{back}} + \tilde{C}_{\text{front}}$$

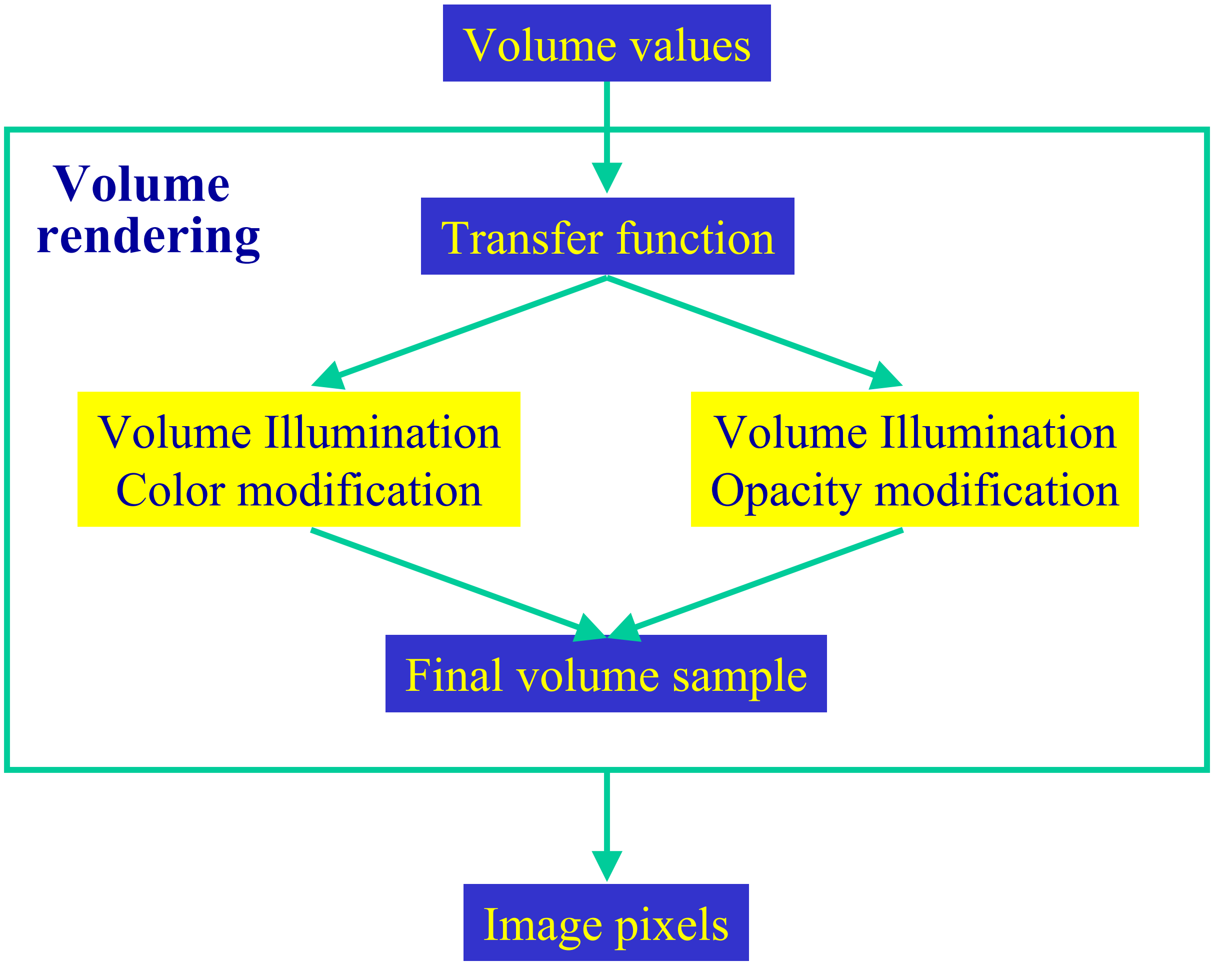
Volume Illustration

- Non-photorealistic rendering of volume models
- Properties
  - Volume sample location and value
  - Local volumetric properties, such as gradient and minimal change direction
  - View direction
  - Light information

Traditional Volume Rendering Pipeline



Volume Illustration Rendering Pipeline



Feature Enhancement

- Boundary enhancement
  - Gradient-based opacity

$$o_g = o_v (k_{gc} + k_{gs} (\|\nabla_f\|)^{k_{ge}})$$

Original opacity

Value gradient of the volume at the sample

Feature Enhancement (cont)

- Boundary enhancement example

Original volume rendering

Boundary enhancement

$k_{gc} = 0.7, k_{gs} = 10, k_{ge} = 2.0$

Feature Enhancement (cont)

- Oriented feature enhancement
  - Silhouette enhancement

$$o_s = o_v(k_{sc} + k_{ss}(1 - abs(\nabla_{fn} \cdot V))^{k_{se}})$$

gradient View direction

Feature Enhancement (cont)

- Silhouettes enhancement example

Original volume rendering

Silhouette and boundary enhancement

$$k_{gc} = 0.8, k_{gs} = 5.0, k_{ge} = 1.0;$$
$$k_{sc} = 0.9, k_{ss} = 50, k_{se} = 0.25$$

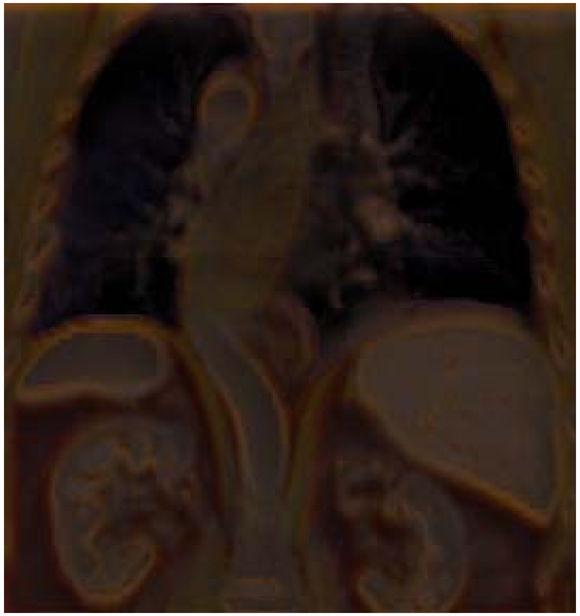
Feature Enhancement (cont)

Original volume rendering

Boundary enhancement

Silhouette and boundary enhancement

Feature Enhancement (cont)



Boundary saturation increased and value also increased

Boundary saturation increased and value decreased

Volumetric colored sketch of data

Depth and Orientation Cues

- Distance color blending
  - Depth-cued color

$$c_d = (1 - k_{ds} d_v^{k_{de}}) c_v + k_{ds} d_v^{k_{de}} c_b$$

controls the size of the color blending effect

controls the rate of application of color blending

The fraction of distance through the volume

Voxel color

Background color

Depth and Orientation Cues (cont)

- Distance color blending example

Original volume rendering

Distance coloring, boundary, and silhouette enhancement

$c_b = (0,0,0.15), k_{ds} = 1.0, k_{de} = 0.5$

Depth and Orientation Cues (cont)

- Feature halos
  - The size of halo effect

$$h_i = \left(\sum_{n}^{neighbors} \frac{h_n}{\left\| P_i - P_n \right\|^2} \right) \left(1 - \left\| \nabla_f (P_i) \right\| \right)$$

The maximum potential halo contribution of a neighbor

location

$$h_n = \left(\nabla_{fn}(P_n) \cdot \left(\frac{(P_i - P_n)}{\left\| P_i - P_n \right\|} \right) \right)^{k_{hpe}} \left(1 - \nabla_{fn}(P_n) \cdot V \right)^{k_{hse}}$$

Depth and Orientation Cues (cont)

- Feature halos example

Original volume rendering

Halos, boundary, and silhouette enhancement

$k_{hpe} = 1.0, k_{hse} = 2.0$

Depth and Orientation Cues (cont)

- Tone shading

number of lights

illuminated object color contribution

$$c = k_{ta} I_G + \sum_{i}^{N_L} \left(I_t + k_{td} I_o \right)$$

controls the amount of gaseous illumination

tone contribution to volume sample color

controls the amount of directed illumination

$$I_t = ((1.0 + \nabla_{fn} \cdot L)/2)c_w + (1 - (1.0 + \nabla_{fn} \cdot L)/2)c_c$$

warm tone color (k_{ty}, k_{ty}, 0)

cool tone color (0, 0, k_{tb})

$$I_o = \begin{cases} k_{td} I_i (\nabla_{fn} \cdot L) : \nabla_{fn} \cdot L > 0 \\ 0 : \nabla_{fn} \cdot L \le 0 \end{cases}$$

Depth and Orientation Cues (cont)

- Tone shading example

Original volume rendering

Tone shading, boundary, and silhouette enhancement

$k_{ty} = 0.3, k_{tb} = 0.3, k_{ta} = 1.0, k_{td} = 0.6$

Depth and Orientation Cues (cont)

Distance coloring, boundary, and silhouette enhancement

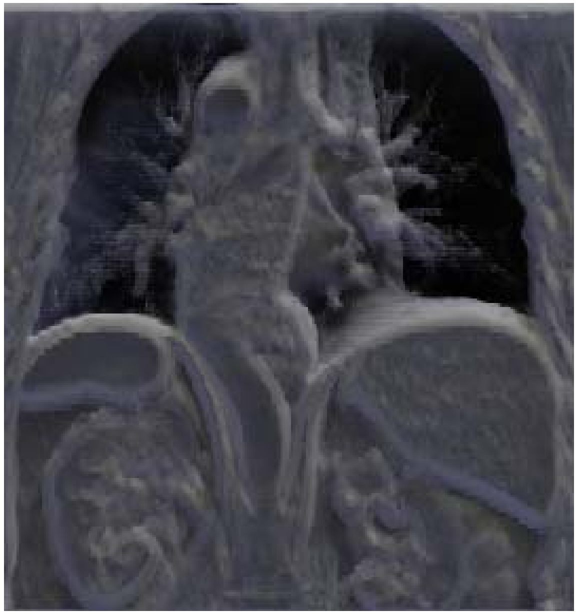
Halos, boundary, and silhouette enhancement

Tone shading, boundary, and silhouette enhancement

Depth and Orientation Cues (cont)

- Gray scale data

Original volume rendered image

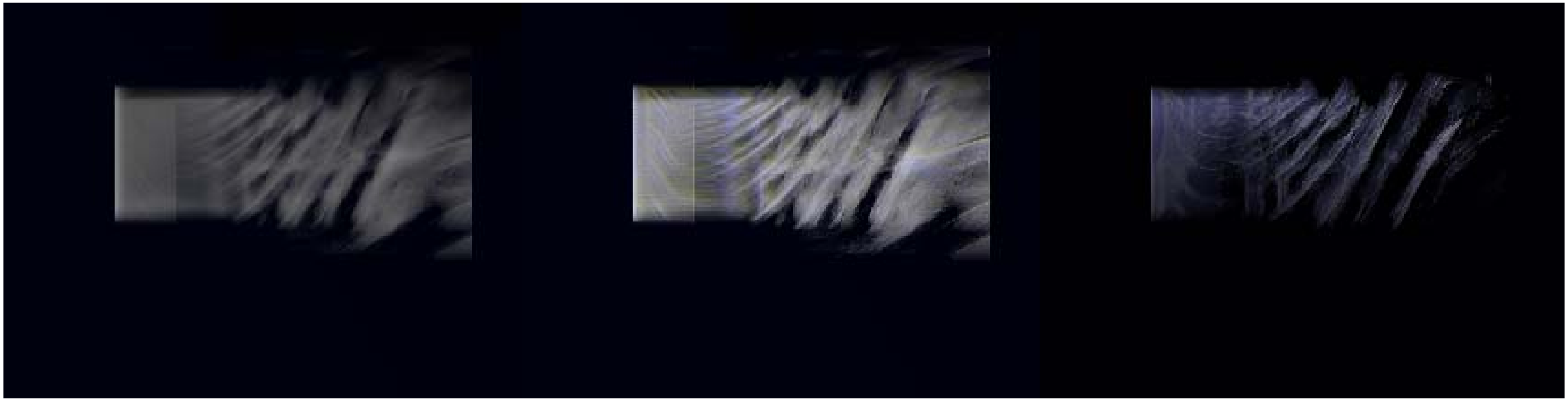


Tone enhancement of image data

Boundary volumetric sketch of data

Depth and Orientation Cues (cont)

- 2D square vortex results



Original gaseous rendering of jet

Tone shading, boundary, silhouette enhancement added

White silhouette color fading added to blue gaseous volume

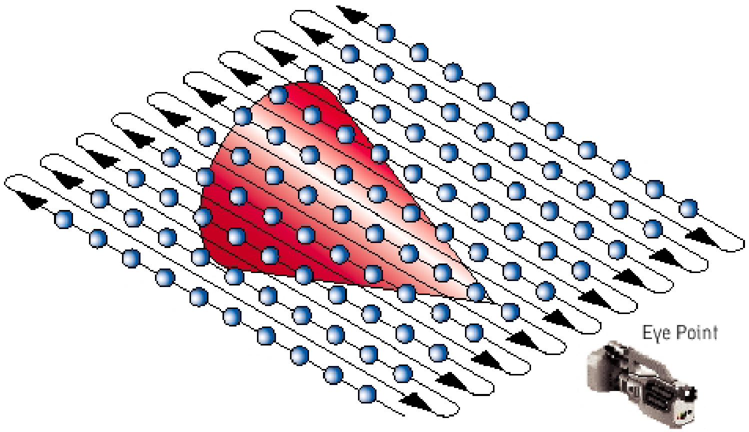
Center
Computational
Visualization
CCV

To wake up with coffee!

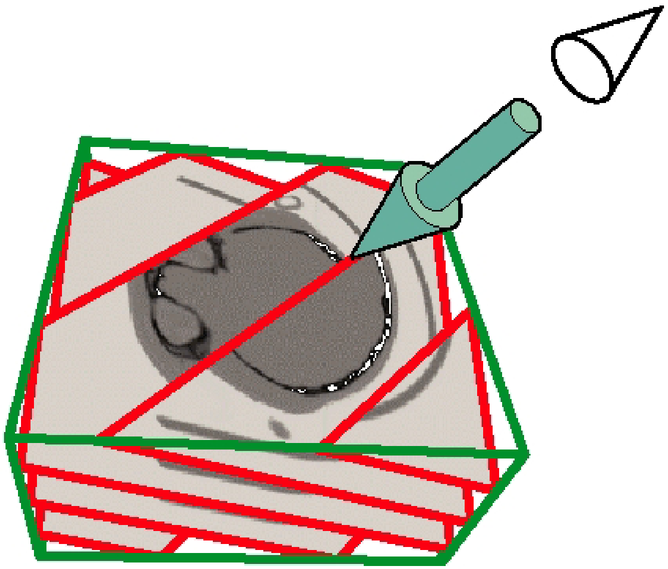
Or Mineralwasser !!

Texture Based Volume Rendering

- 3D Texture mapping hardware

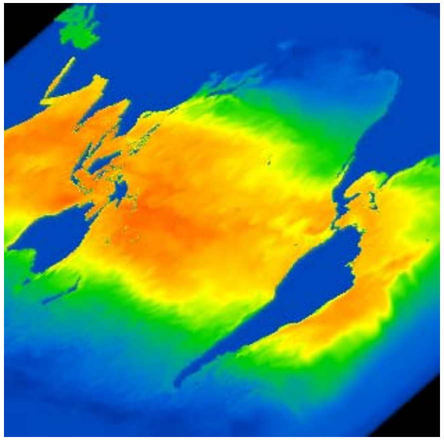


Ray Casting



3D Texture Mapping

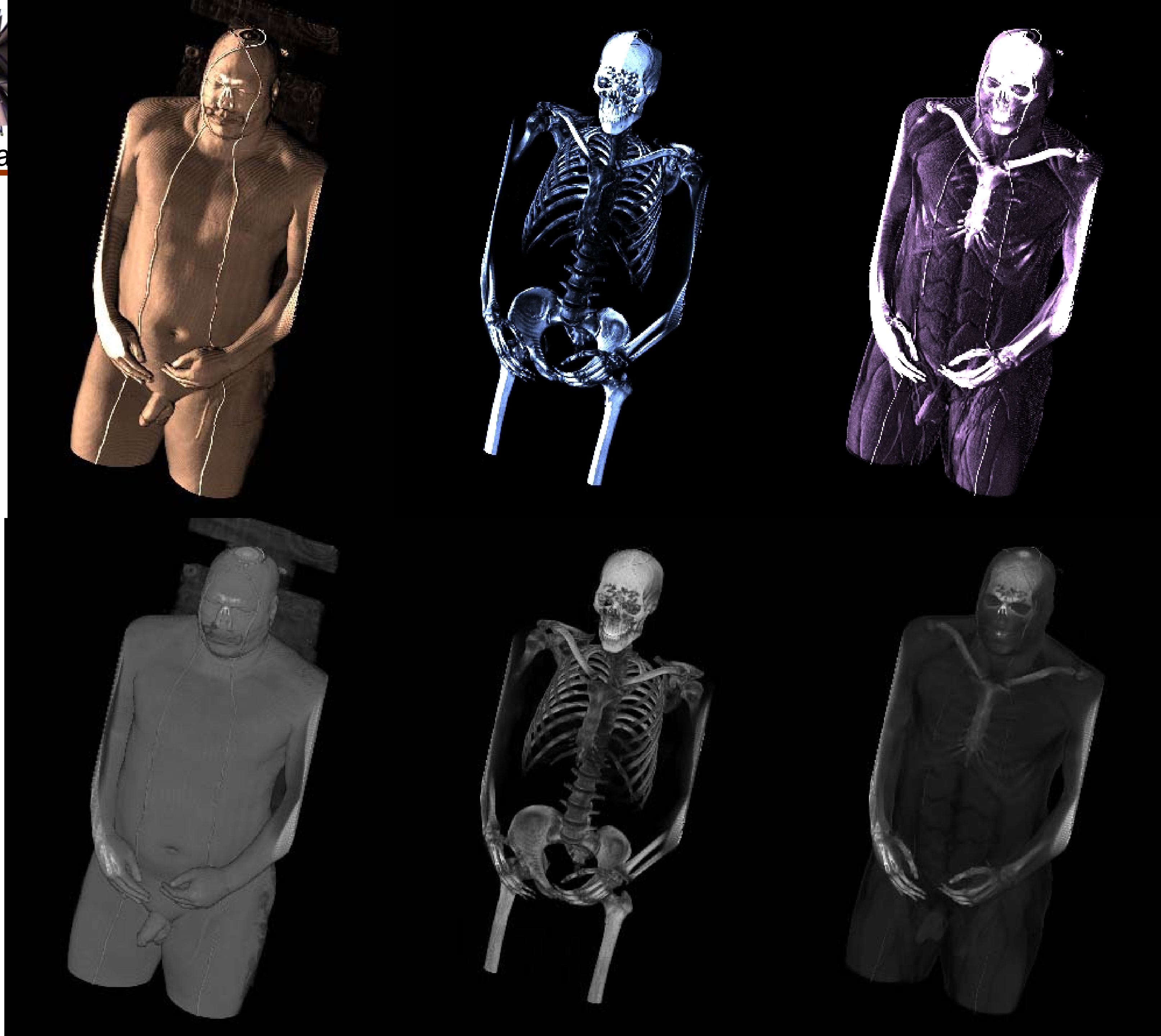
Parallel Texture Based Volume Rendering



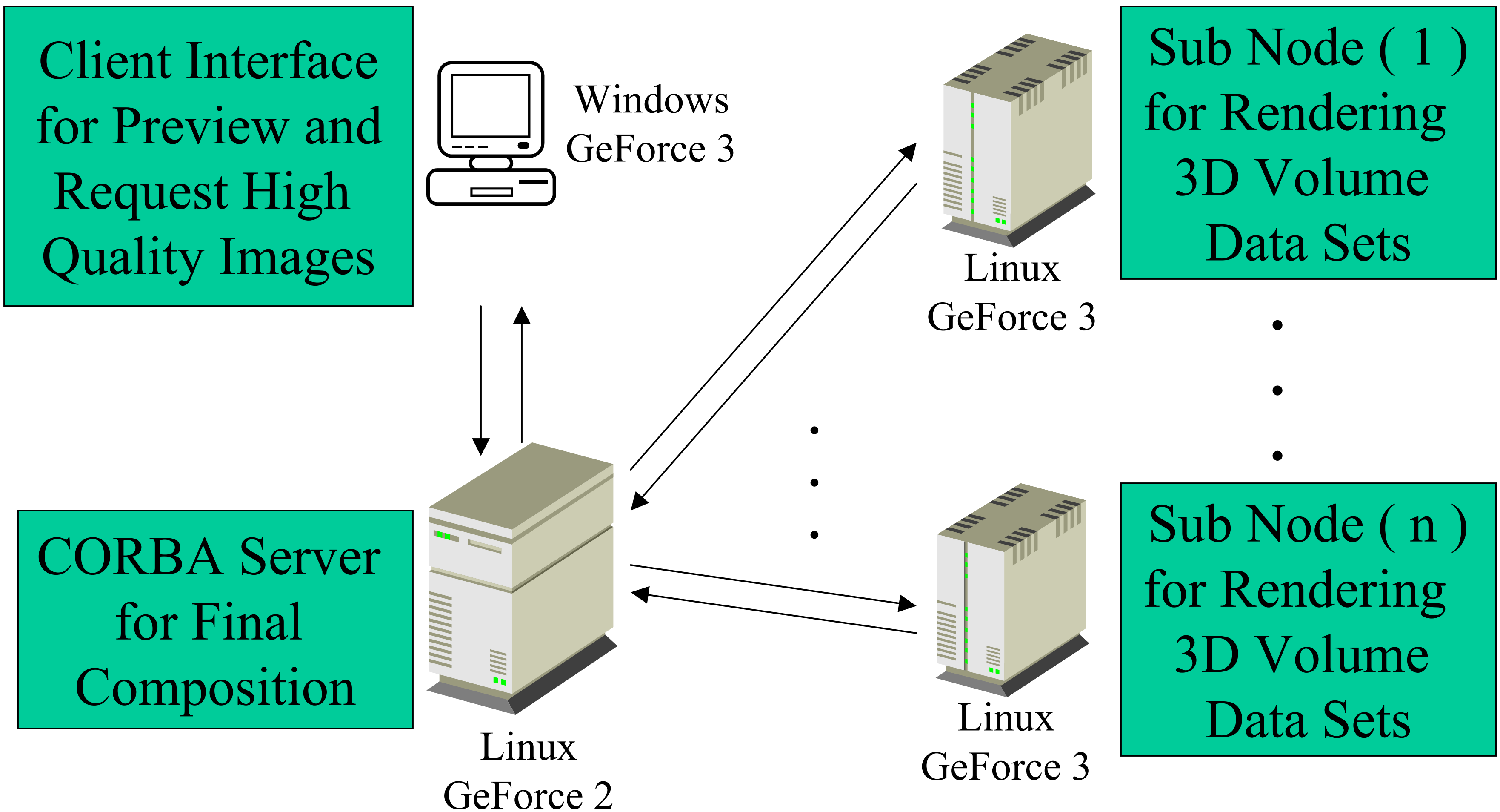
Real-time multipipe texture based volume rendering of the time-varying oceanography temperature data.

Visualization of seismic simulation data on the CCV Visualization Lab's front multi projection system.

Shaded image of the Visible Human female data using texture hardware.



System Diagram



Center ccv Visualization Computational

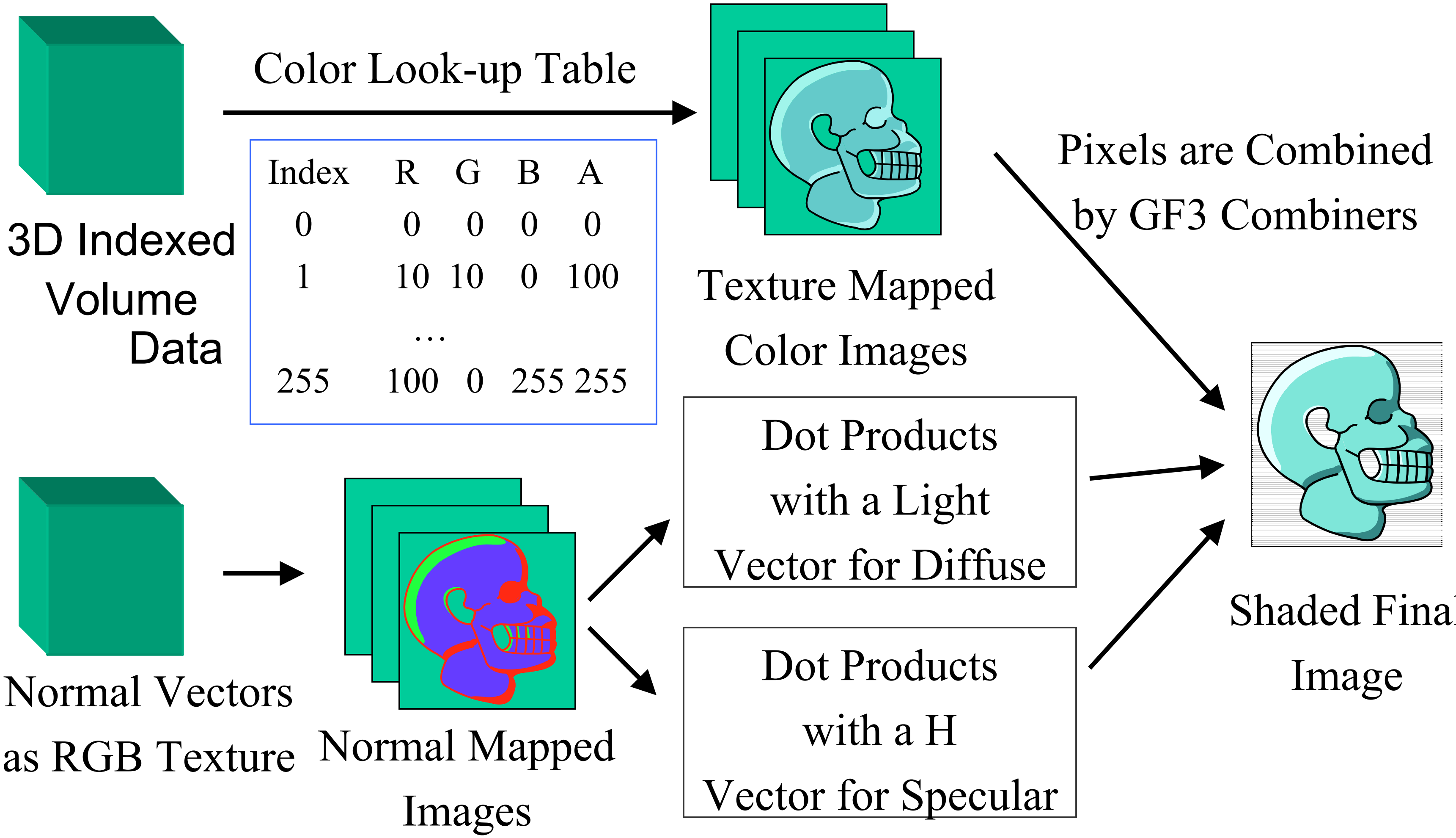
Client-server Algorithm

1. Adjust color table & transfer function using Windows interface.
2. Send a request to CORBA server.
3. The CORBA server distributes work to each node using MPI.
4. Each node renders each part of data using back-to-front composition.
5. The CORBA server takes the image pieces from each node and composites them into an image.
6. The Windows interface takes the final image.

Hardware Accelerated Rendering Algorithm

1. Load a 3D indexed volume data and normal vectors as RGB to texture memory on GF3
2. Set up a color look-up table
3. Set up combiners of GF3 for shading for color of texture , diffuse and specular
4. Calculate intersection between texture cube and texture mapped planes parallel with view planes
5. Composite the texture mapped planes using back-to-front composition

Hardware Accelerated Rendering



Index	R	G	B	A
0	0	0	0	0
1	10	10	0	100
…				
255	100	0	255	255

Front-to-back Composition

- Texture-mapped planes blending

$$C_d = C_d + (1-\alpha_d)\alpha_s C_s \ , \quad \alpha_d = \alpha_d + (1-\alpha_d)\alpha_s$$

C_d : Destination color $\quad$ C_s: Source color

α_d : Destination alpha $\quad$ α_s: Source alpha

- Final composition of sub-images

$$C_d = C_d + (1-\alpha_d)C_s \ , \quad \alpha_d = \alpha_d + (1-\alpha_d)\alpha_s$$

- OpenGL Commands and notes
  - glBlendFunc (GL_ONE_MINUS_DST_ALPHA , GL_ONE)
  - $\alpha_s C_s$ should be pre-multiplied using a color table or register combiners of GeForce3

Image Enhancement

- Bilateral Filter

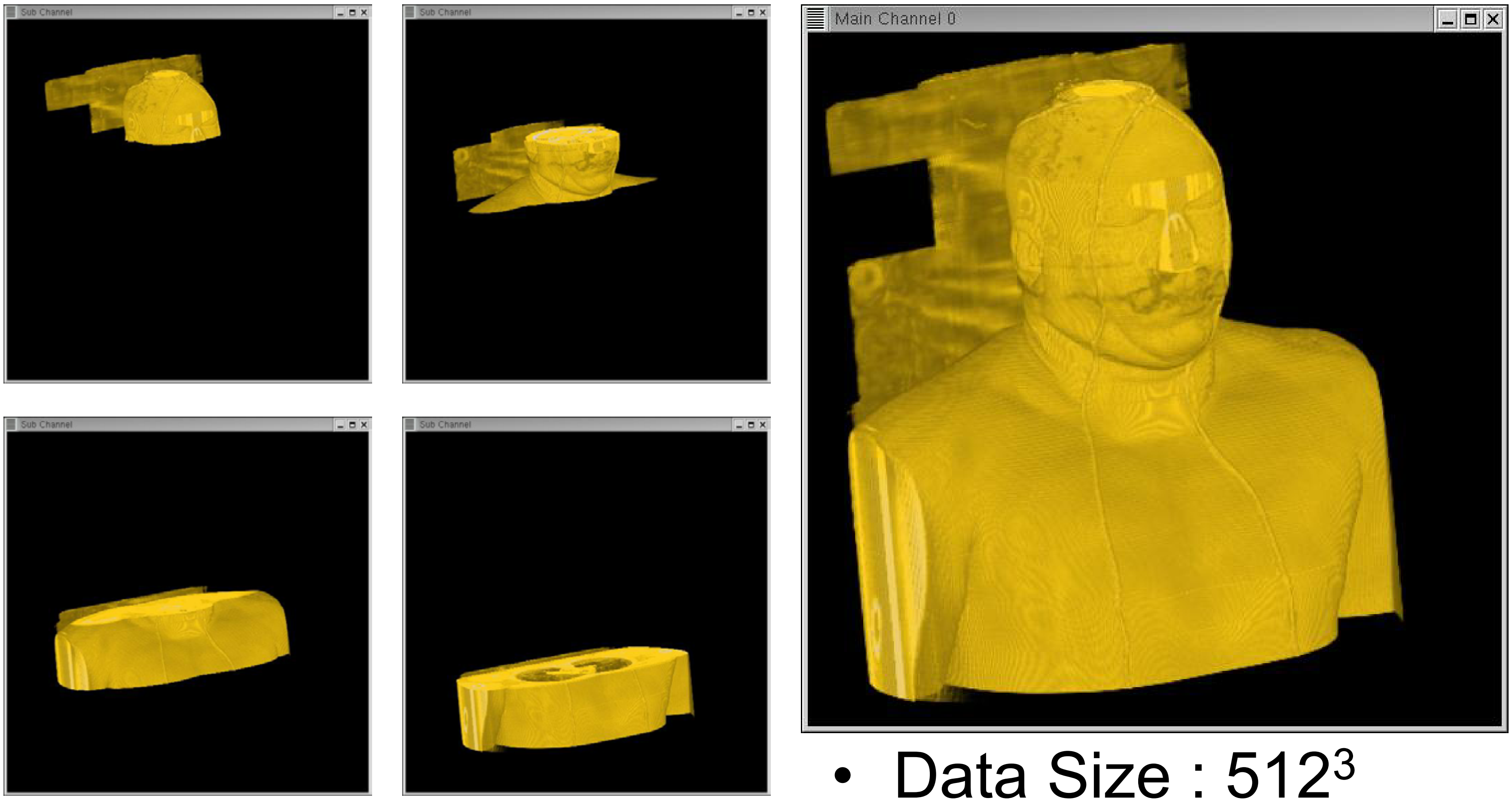
$$w_{ijk} = \exp\left[-\frac{\left(f(x,y,z) - f(i,j,k)\right)^2}{2\delta^2} \right]$$

$$f_{new}(x,y,z) = \sum\sum\sum w_{ijk} \cdot f(i,j,k)$$

$f(i,j,k)$: Original image, $f_{new}(x,y,z)$: New image

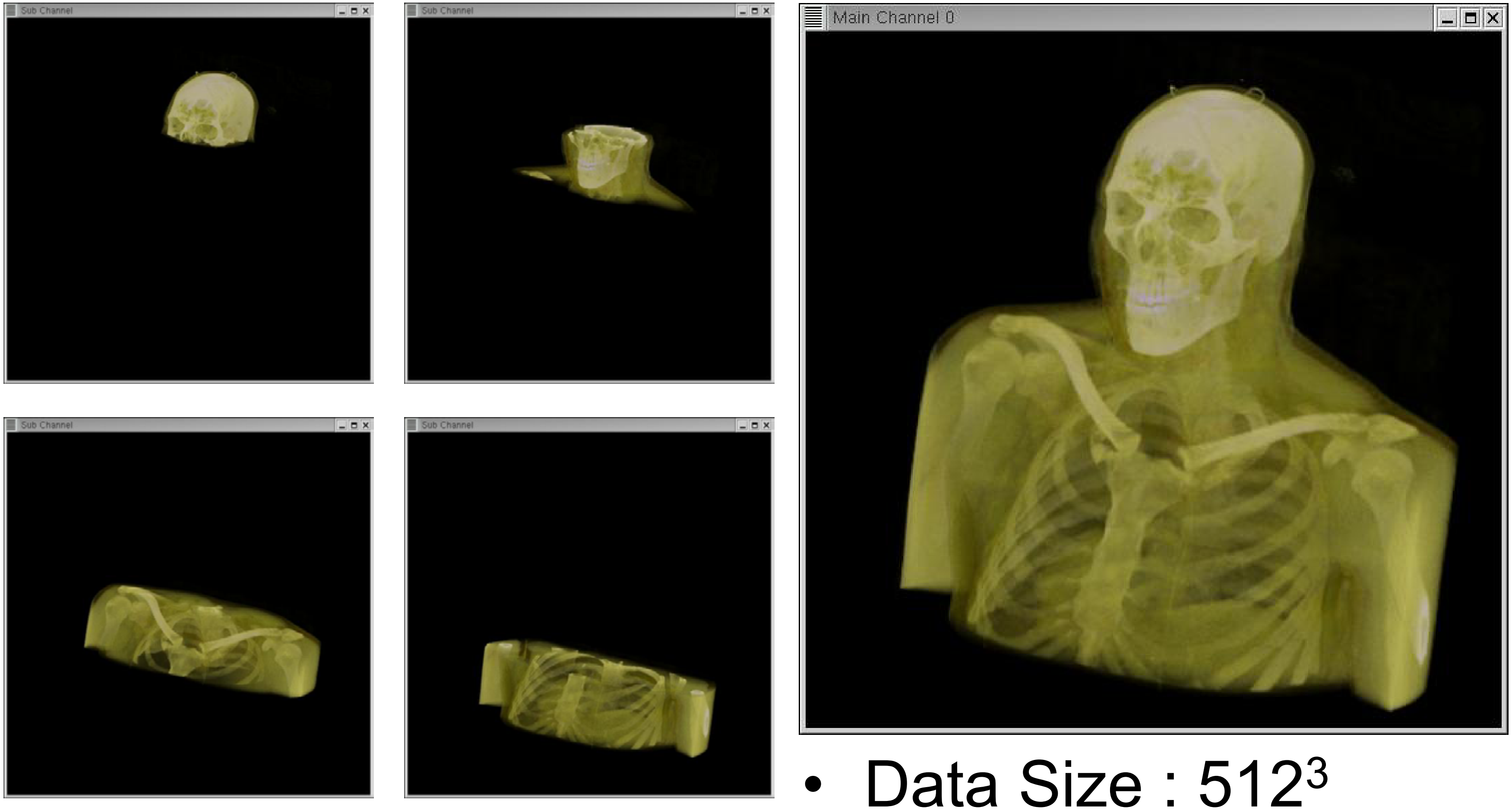
- Normal Calculation
  - (Multi-Linear Centroid Averaging) MLCA

Unshaded Images of Each Node and A Final Image - Skin



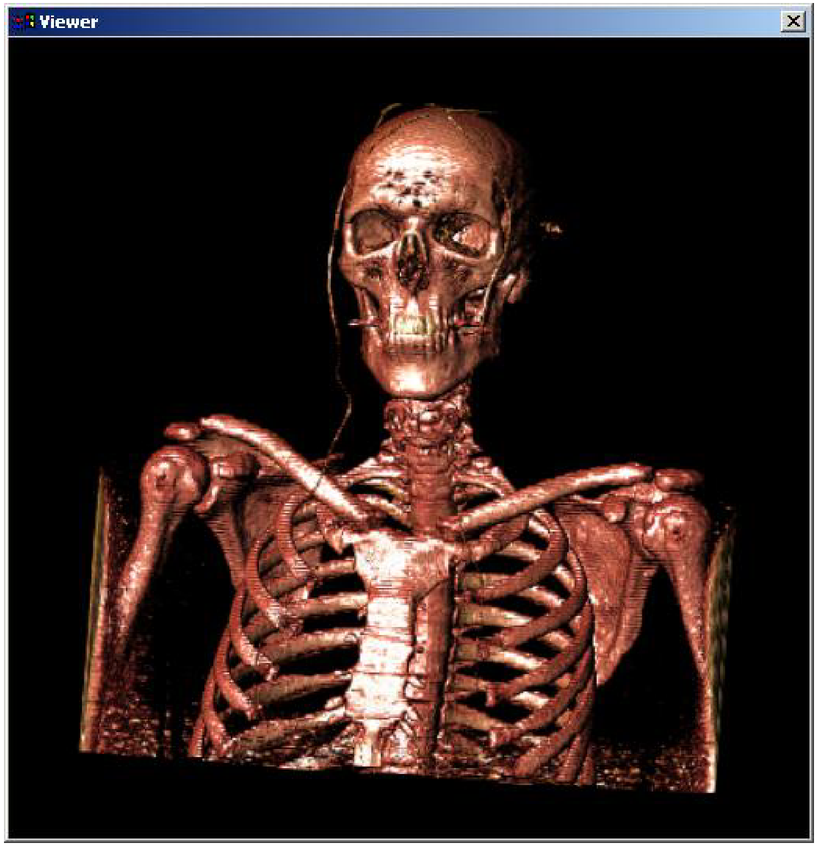
- Data Size : 512^3
- Perfomance : 4.01fps

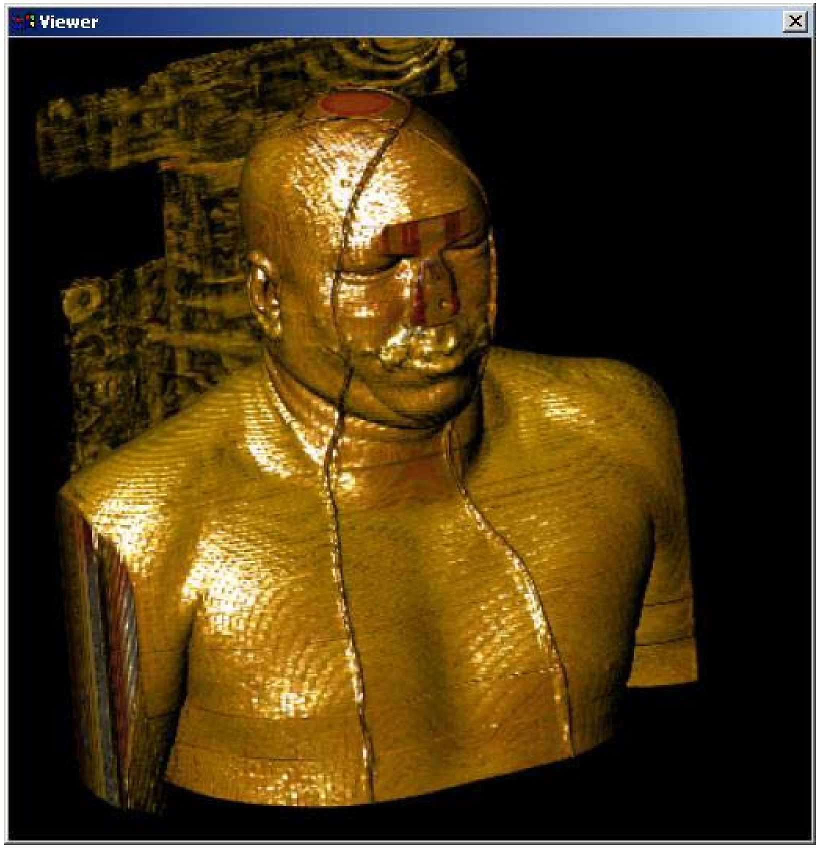
Unshaded Images of Each Node and A Final Image - Bones



- Data Size : 512^3
- Performance 4.01fps

Shaded Images of Visible Human Male Data Set

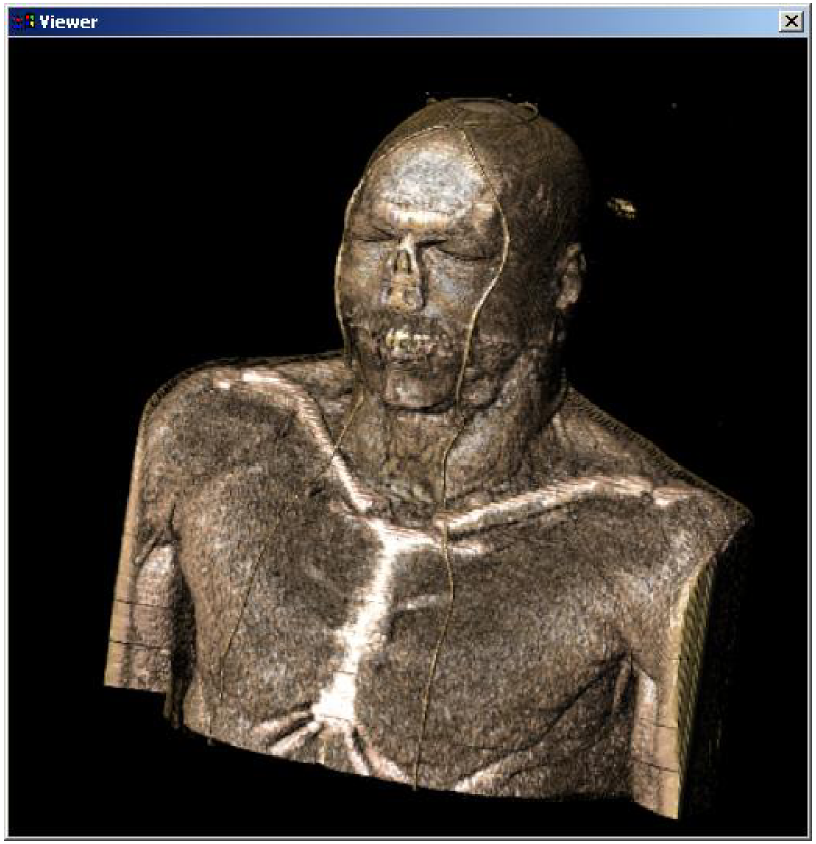


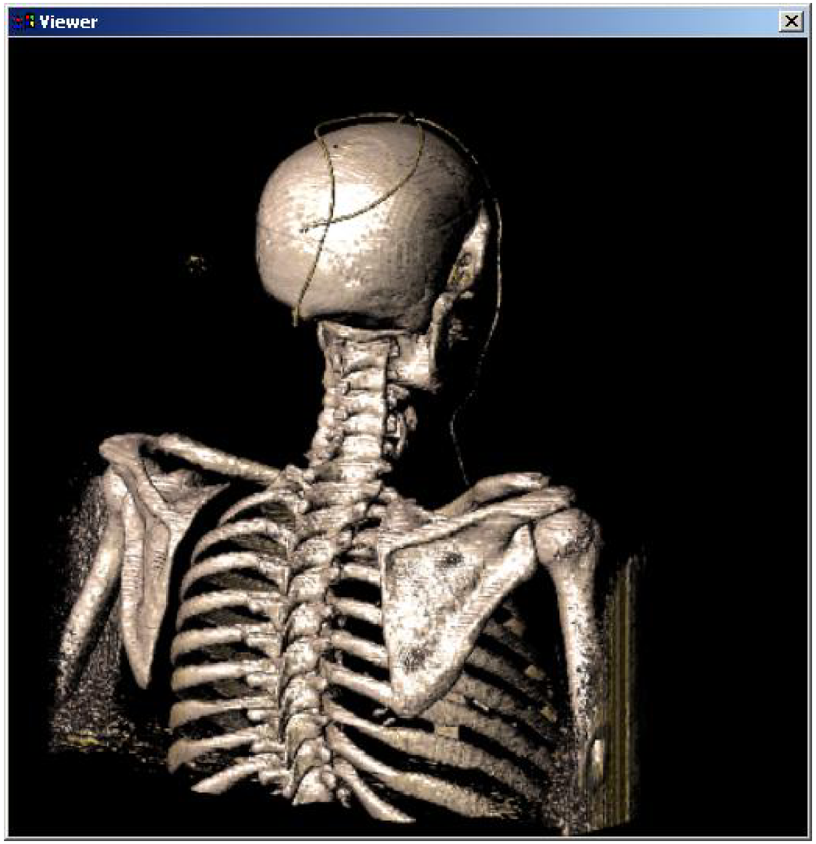


Visualization of bones and skin

Data size : 512^3

Shaded Images of Visible Human Male Data Set

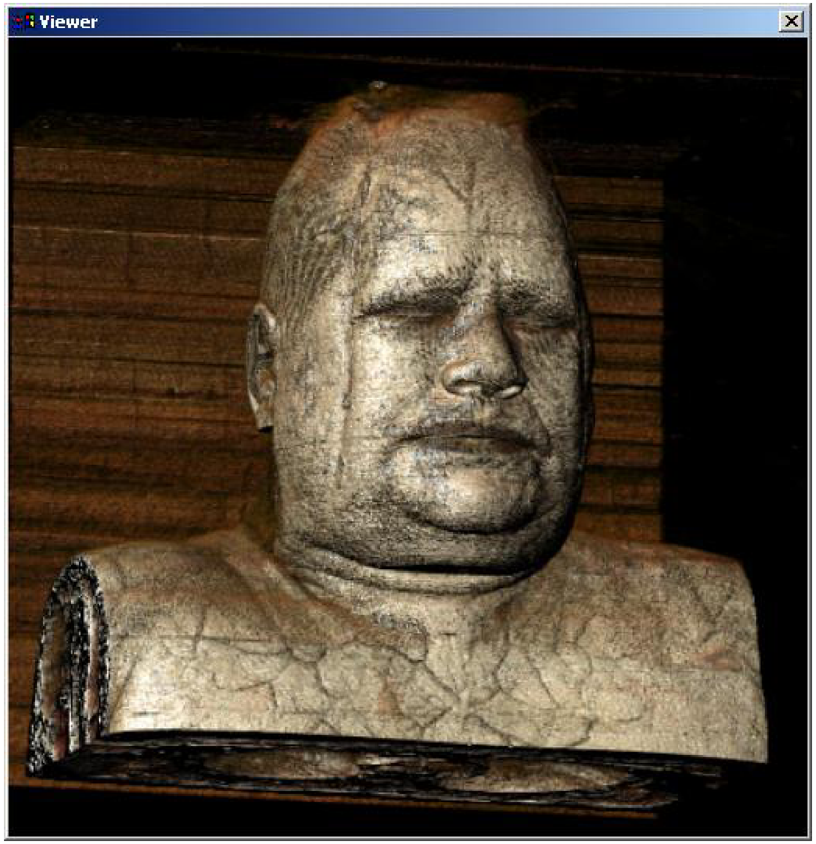


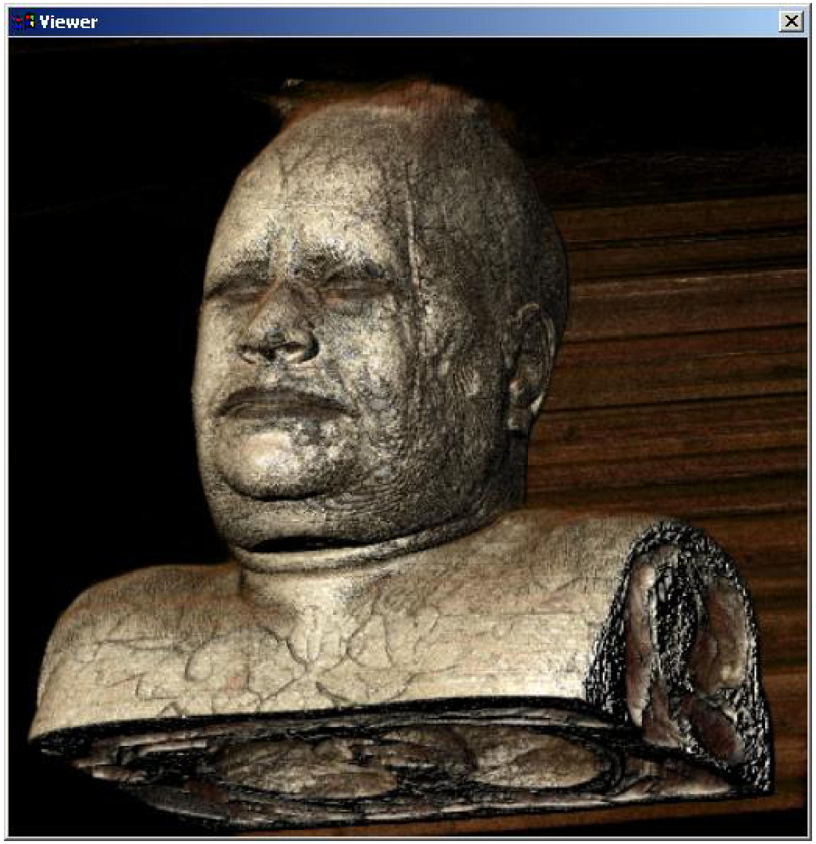


Visualization of muscles and bones

Data size : 512^3

Shaded Images of Visible Human Female Data Set



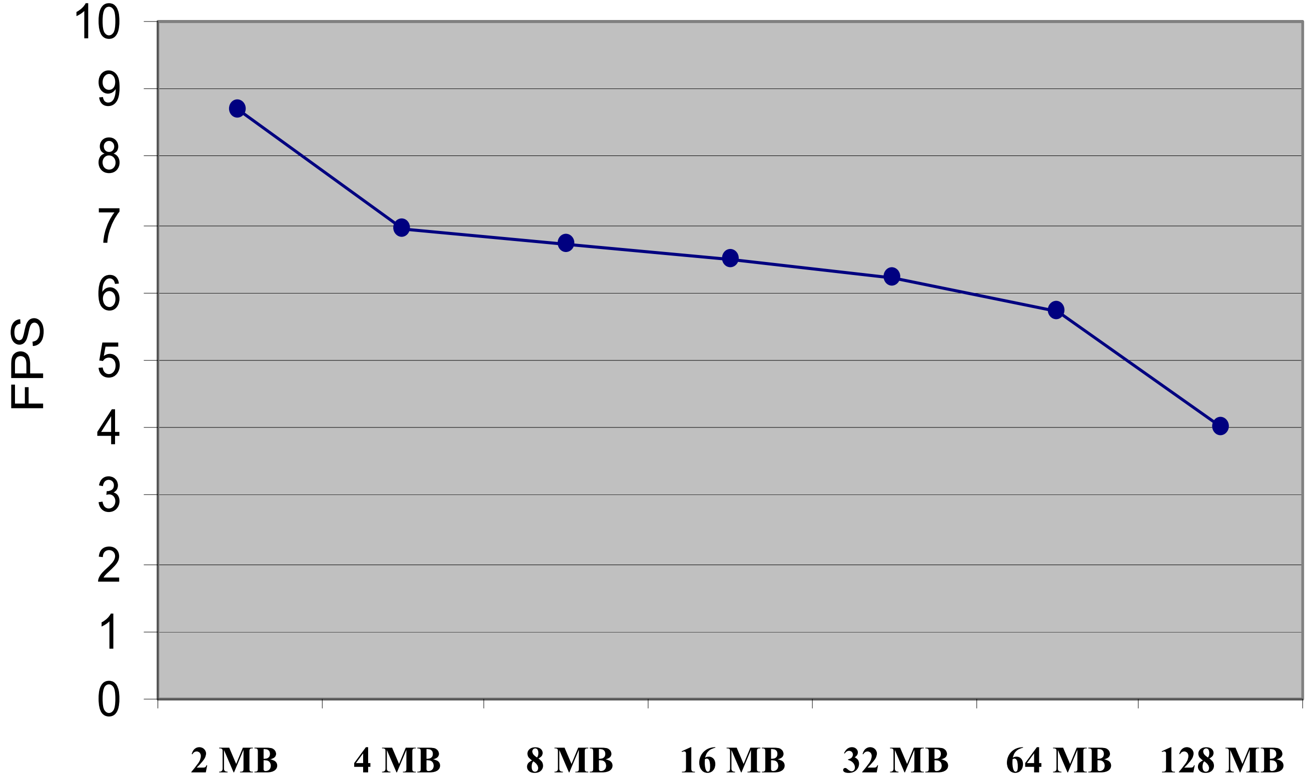


Visualization of skin

Data size : 512^3

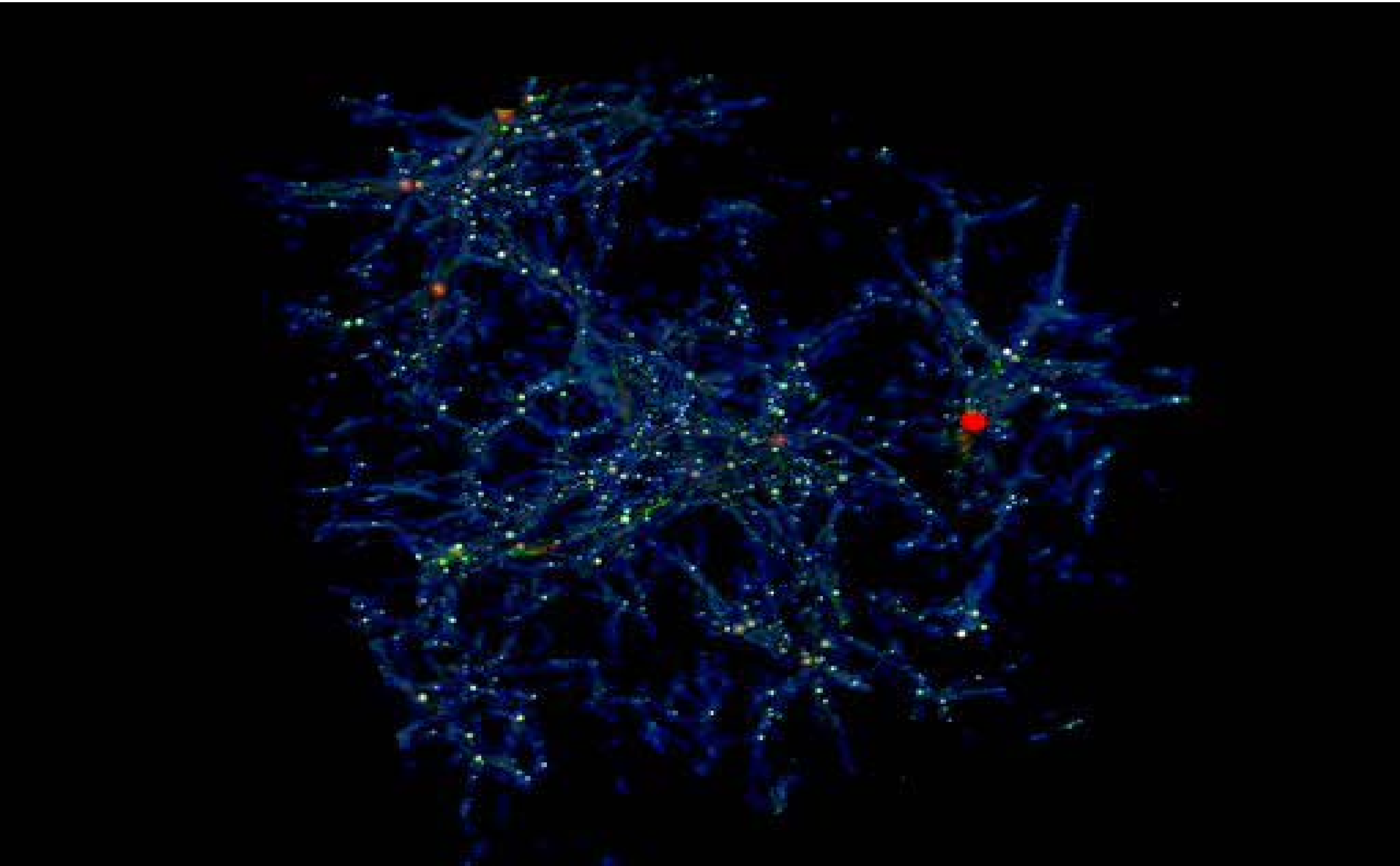
Performance

Parallel Unshaded Rendering with 4 GeForce3 and a GeForce2 Cards



X	*Data Size (MB)*	*FPS*
1	2	8.68
2	4	6.92
3	8	6.73
4	16	6.47
5	32	6.22
6	64	5.74
7	128	4.01

Mini-Halos Simulation



Optical Models

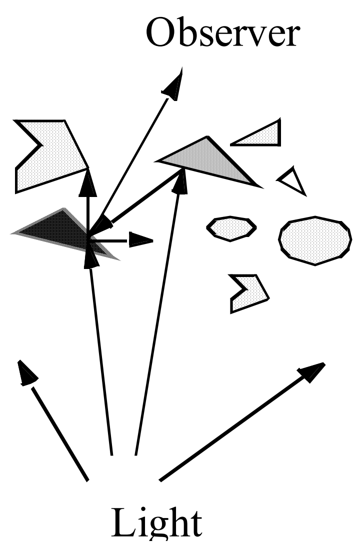
- Jim Blinn's 1982 SIGGRAPH paper on light scattering
- Nelson Max, "Optical Models", IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 2, 1995.

- The mathematical framework for light transport in volume rendering based on

  S. Chandrasekhar "Radiative Transfer", Oxford Universtiy Press, 1950

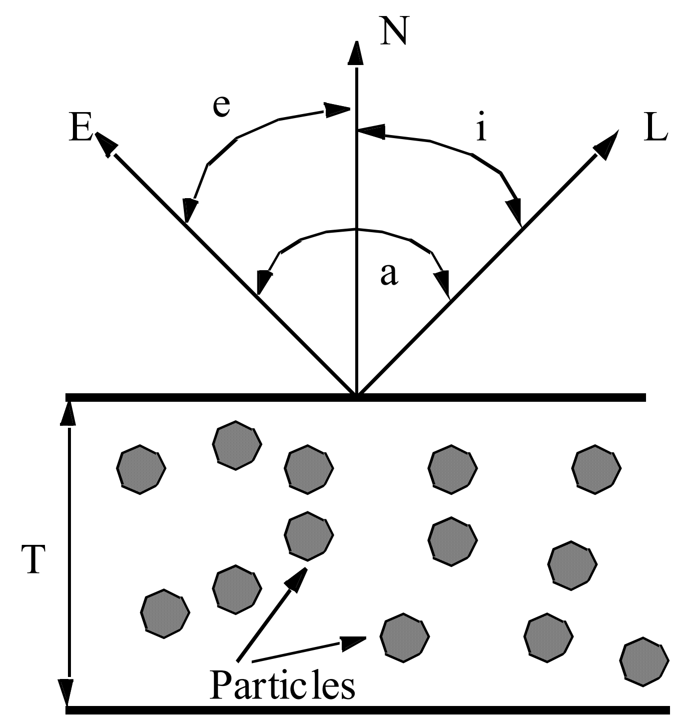
Transport of Light

- Determination of Intensity
- Local - Diffuse and Specular
- Global - Radiosity, Ray Tracing
- Mechanisms in Ultimate Model
  - Emittance
  - Absorption
  - Scattering (single vs. multiple)



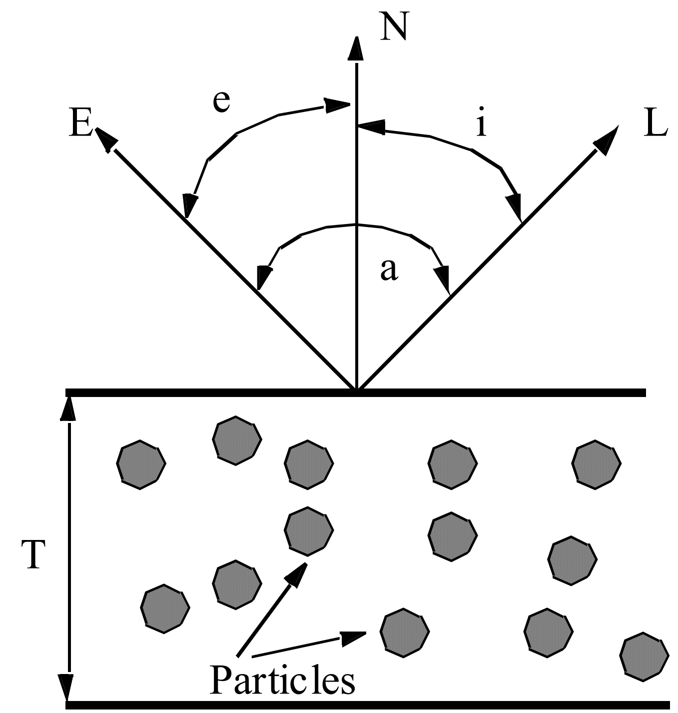
Blinn gaseous model- 1982

- <u>Assumptions</u>:
  - *N* - surface normal
  - *E* - eye vector
  - *L* - light vector
  - *T* - surface thickness
  - *e* - angle btw. *E* and *N*
  - *a* - angle btw. *E* and *L* aka phase angle
  - i - angle btw. *N* and *L*



Blinn model (contd.)

- <u>Assumptions (contd.):</u>
  - particles are little spheres with radius *p*
  - *n* - number density (number of particles per unit volume)
  - μ - cosine of angle e, *(N.E)*
  - *D* - proportional volume of the object occupied by particles



$$D = n\frac{4}{3}\pi p^3$$

Blinn model – transparency (1)

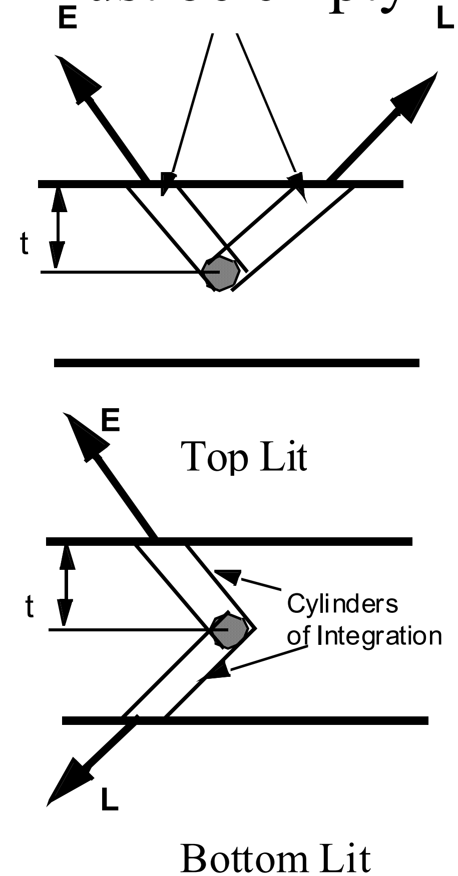
- Expected particles in a volume will be *nV*
- Probability that there are no particles in the way can be modeled as a Poisson process:

$$P(0,V) = e^{-nV}$$

- Hence the probability that the light is making it through those tubes is:

$$P(0,V) = e^{-n\pi p^2 T'/\mu_0} e^{-n\pi p^2 T'/\mu}$$

Cylinders must be empty



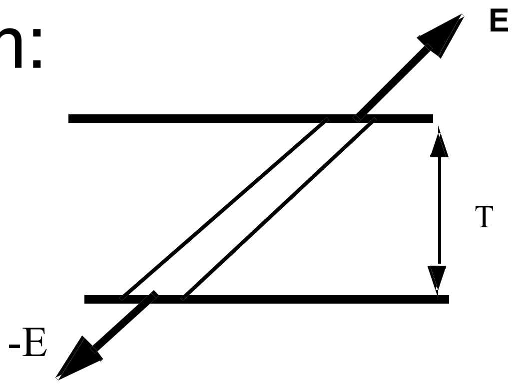
Blinn model – transparency (2)

- Transparency through the medium:

$$Tr = e^{-\tau/\mu}$$

- τ is called the optical depth:

$$\tau = n\pi p^2 T$$



Max model - 1995

- Several cases:
  - Completely opaque or transparent voxels
  - Variable opacity correction
  - Self-emitting glow
  - Self-emitting glow with opacity along viewing ray
  - Single scattering of external illumination
  - Multiple scattering

Max model - absorption only

- $I(s)$ = intensity at distance s along a ray
- $\tau(s)$ = extinction coefficient

$$\frac{dI}{ds} = -\tau(s) I(s)$$

$$I(s) = I(0)\exp\left(-\int_0^s \tau(t)\,dt\right)$$

$$= I_0 T(s)$$

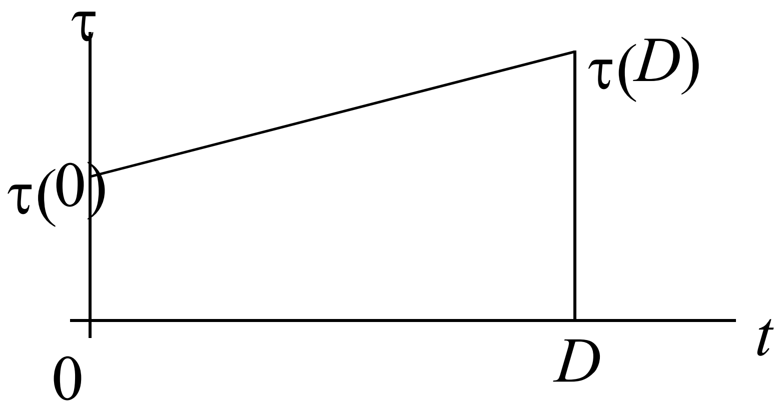
- $T(s)$ = transparency between 0 and s

Max - absorption only

- Linear variation of τ:

$$T(s) = \exp\left(-\int_0^D \tau(t)\,dt\right)$$

$$= \exp\left(-D\,\frac{\tau(0)+\tau(D)}{2}\right)$$



Max model - absorption only

- On the opacity α:

$$\alpha = 1 - T(s) = 1 - \exp\left(- \int_0^D \tau(t)\,dt \right)$$
$$= 1 - \exp(-\tau D)$$
$$= \tau D - (\tau D)^2 / 2 + \ldots$$

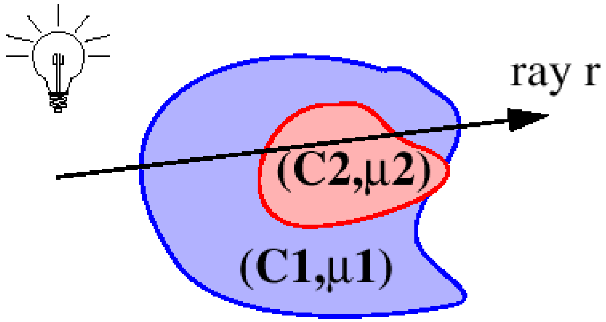
- assuming τ to be constant in the interval

Volume Ray Integration (1)

- The continuous form:

$$I(D) = I_0 \exp\left(-\int_0^D \tau(t)\,dt \right) + \int_0^D g(s) \exp\left(-\int_s^D \tau(t)\,dt \right) ds$$

- In general , cannot compute analytically



Volume Ray Integration (2)

- Practical Computation Method:

$$I(D) = I_0 \exp\left(-\int_0^D \tau(t)\,dt\right) + \int_0^D g(s)\exp\left(-\int_s^D \tau(t)\,dt\right)ds$$

$$t_i = \exp(-\tau(i\Delta x)\Delta x) \approx 1 - \tau(i\Delta x)\Delta x$$

$$I(D) = I_0 \prod_{i=1}^{n} t_i + \sum_{i=1}^{n}\left(\prod_{j=i+1}^{n} t_j\right) g_i$$
$$= g_n + t_n\left(g_{n-1} + t_{n-1}\left(g_{n-2} + \ldots\left(g_1 + t_1 I_0\right)\ldots\right)\right)$$

which leads to the familiar BTF or FTB compositing

g(*s*)

- *g*(*s*) could be:
  - Self-emitting particle glow
  - Reflected color, obtained via illumination

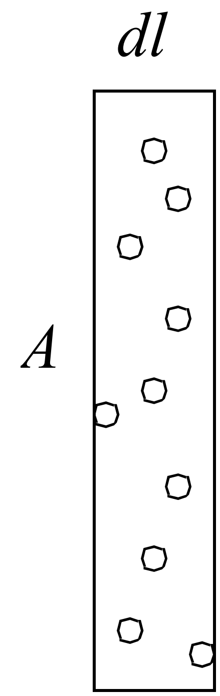
- The color is usually the sum of emitted color *E* and reflected color *R*

Max - self-emitting glow

- Identical glowing spherical particles:
- projected area $a = \pi r^2$
- surface glow color = C
- number per unit volume = N

$$\frac{\text{occluded area}}{\text{total area}} = \frac{aNAdl}{A}$$

- extinction coefficient $\tau = aN$
- added glow intensity per unit length
  $g = CaN = C\tau$



Max - self-emitting glow

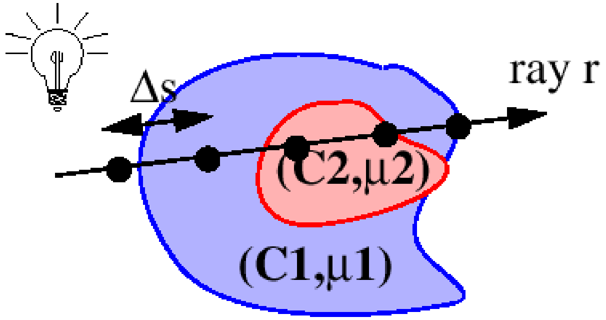
- Special Case g=$C\tau$: (and C constant)

$$\int_0^D g(s)\exp\left(-\int_s^D \tau(t)dt\right)ds = \int_0^D C\tau(s)\exp\left(-\int_s^D \tau(t)dt\right)ds$$

$$= C\left(1-\exp\left(-\int_0^D \tau(t)dt\right)\right)$$

$$I(D) = I_0 T(D) + C(1-T(D))$$

- This is compositing color C on top of background I_0

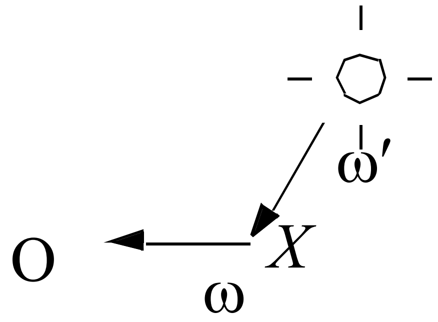


Max - self-emitting glow

- For I_0=0 and τ: varying according to f :

Max - reflection

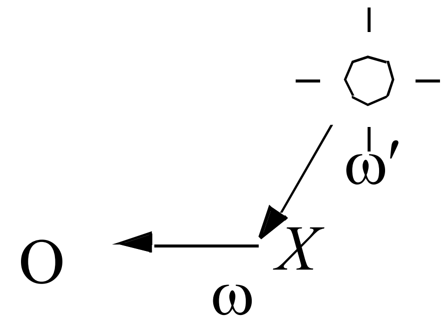
$$g(x) = r(x, \omega, \omega')i(x)$$



- $i(x)$ = illumination reaching point x
- ω = unit reflection direction vector
- ω' = unit illumination direction vector
- $r(x,\omega,\omega')$: BRDF

  $|\nabla f(x) \cdot \omega'|$ for conventional surface shading effects

Max - reflection



- For particle densities:

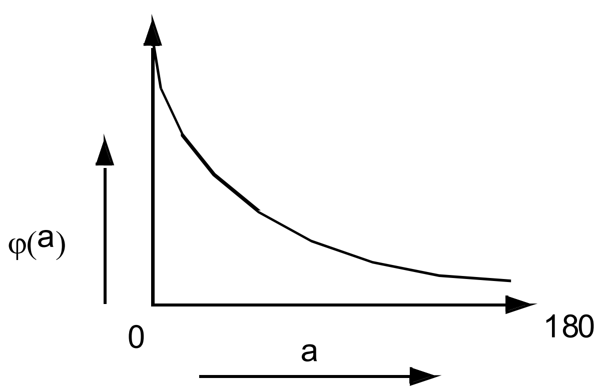
$$r(x,\omega,\omega') = w(x)\tau(x)p(\omega,\omega')$$

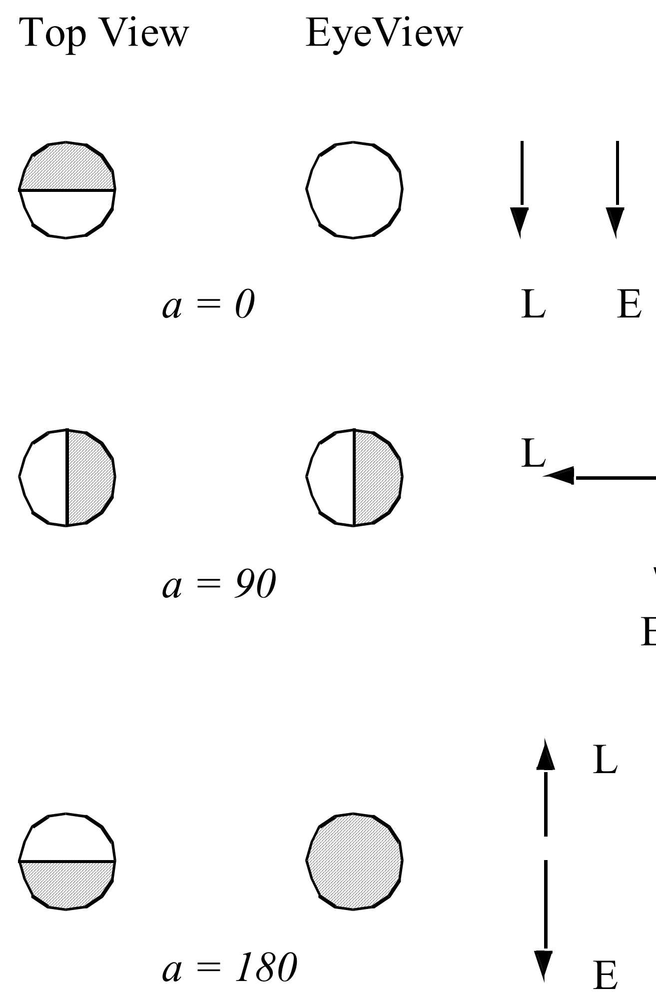
  - $w(x)$ = albedo
    - Blinn: assumes that the primary effect is from interaction of light with one **single** particle
    - albedo - proportion of light reflected from a particle: in the range of 0..1
  - $p(\omega,\omega')$ = phase function

- still unrealistic external reflection of outside illumination

Blinn - Phase Function

- “how” we see the particles
- depends on the angle of eye *E* and light vector *L*
- smooth drop off …





Blinn - Phase Function

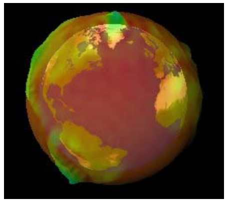
- Many different models possible
- Constant function $\varphi(a) = 1$
  - size of particles much less than wavelength of visible light
- Anisotropic $\varphi(a) = 1 + x\cos(a)$
  - more light forward then backward - essentially our diffuse shading
- Lambert surfaces $\varphi(a) = \frac{8\pi}{3}\left(\sin(a) + (\pi - a)\cos(a)\right)$
  - spheres reflect according to Lamberts law
  - physically based

Blinn - Phase Function

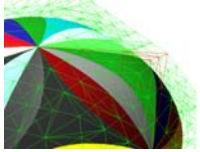
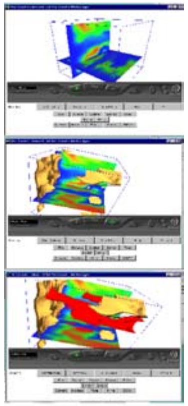
- Rayleigh Scattering $\varphi(a) = \frac{3}{4}\left(1 + \cos^2(a)\right)$
  - diffraction effects dominate
- Henyey-Greenstein $\varphi(a) = \left(1 - g^2\right) / \left(1 + g^2 - 2g\cos(a)\right)^{3/2}$
  - general model with good fit to empirical data
- Empirical Measurments
  - tabulated phase function
- sums of functions
  - weighted sum of functions - model different effects in parallel

Further reading

- **3D RGB Image Compression for Interactive Applications,***ACM Transactions on Graphics, Vol.20, No.1, pages 10-38, 2001*
- **Compression-Based 3D Texture Mapping for Real-Time Rendering** *Graphical Models, Vol. 62, No. 6, pp. 391-410*
- **Compression-based Ray Casting of Very Large Volume Data in Distributed Environments** *HPC-Asia 2000, pages 720-725, Beijing, China, May 2000*
- **Parallel Ray Casting of Visible Human on Distributed Memory Architectures** Proceedings of *Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization* May 26-28, 1999 Vienna, Austria. pp. 269-276



Computational Visualization

1. Sources, characteristics, representation 
2. Mesh Processing 
3. Contouring 
4. Volume Rendering 
5. Flow, Vector, Tensor Field Visualization 
6. Application Case Studies