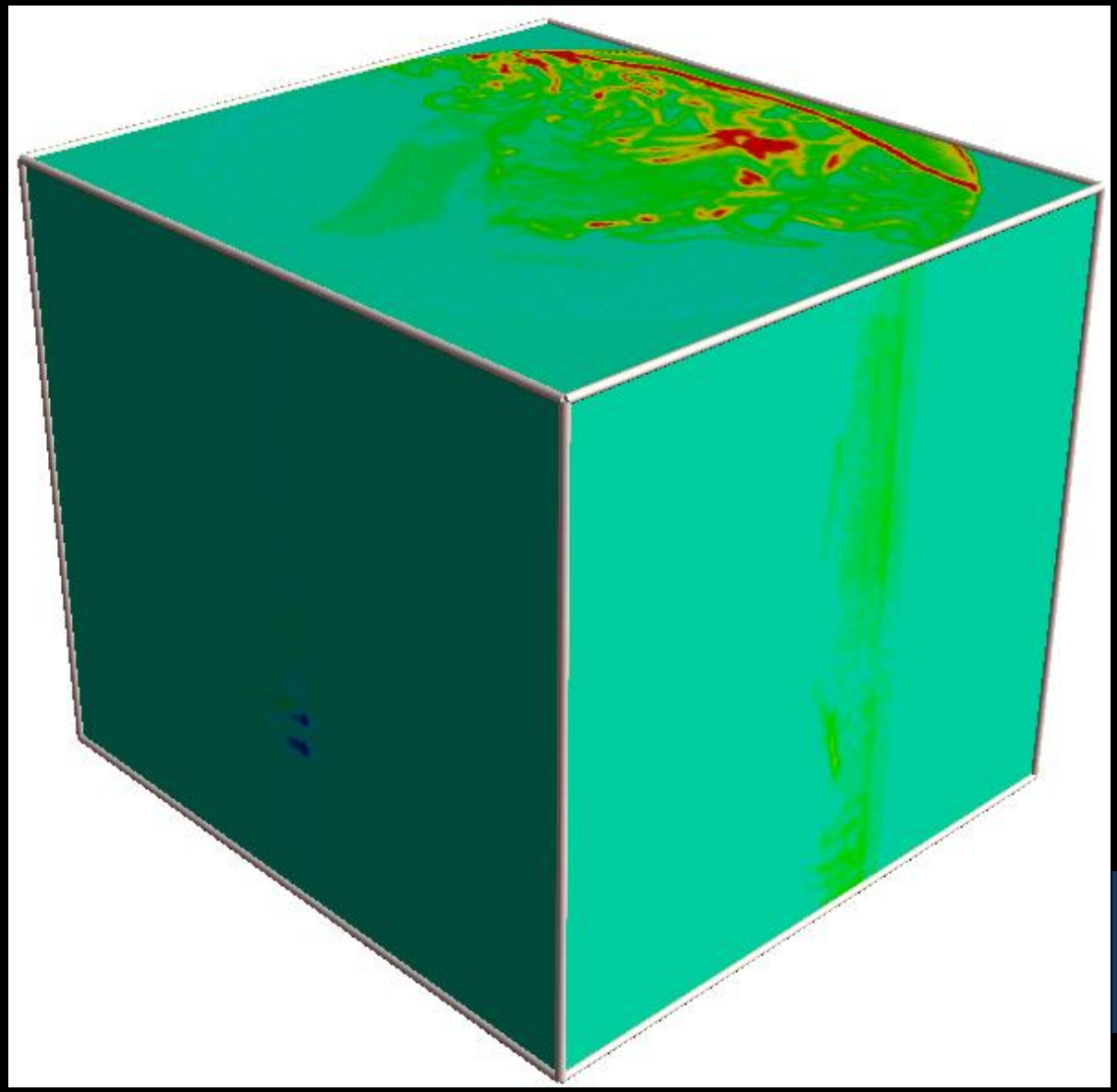
Volume Rendering

Volume Visualization



Visualize a 3D Scalar Dataset

Volume Visualization

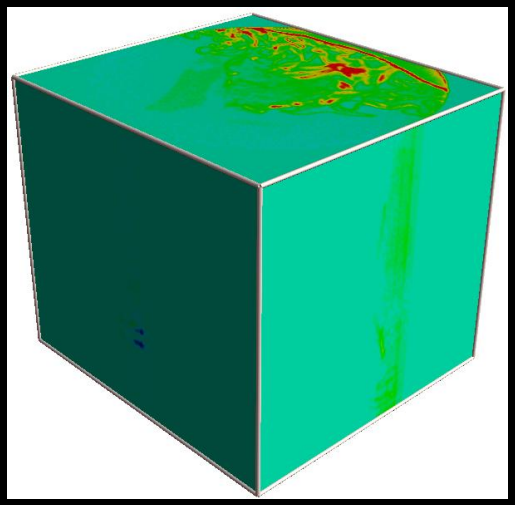
- 3D (volumetric) scalar fields
  - Slice plane and isosurfaces techniques are limited in showing only a subset of the entire scalar volume
- Volume rendering or Volume visualization:
  - Attempt to produce images of an entire 3D scalar volume
  - A separate class of visualization techniques for volumetric scalar fields

Outline

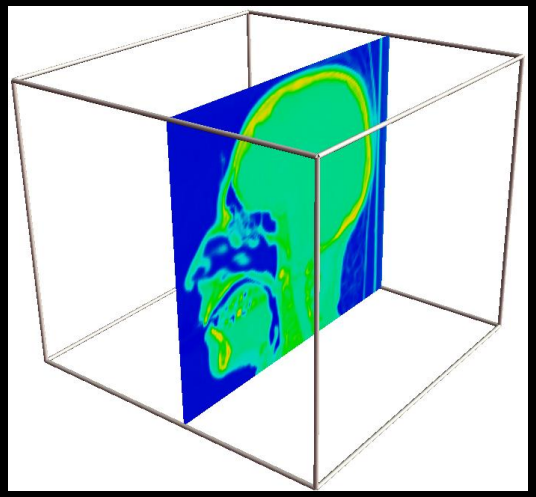
- Motivation
- Volume Visualization Basics
- Image Order Technique
- Object Order Technique
- Volume Rendering vs. Geometric Rendering

Motivation

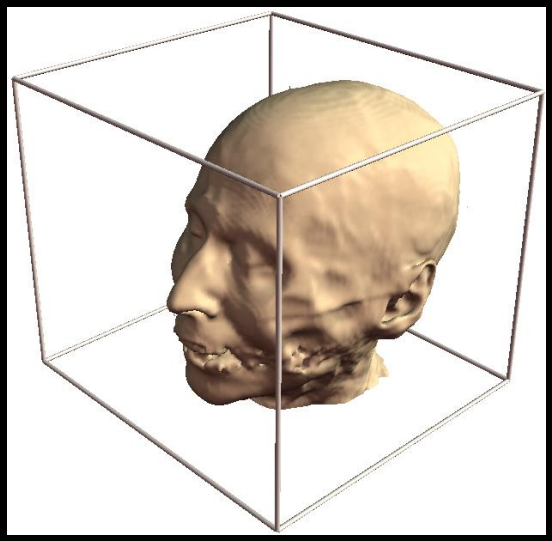
the dataset boundary
Do not reveal inner part



the same y-coordinate
Only show 2D

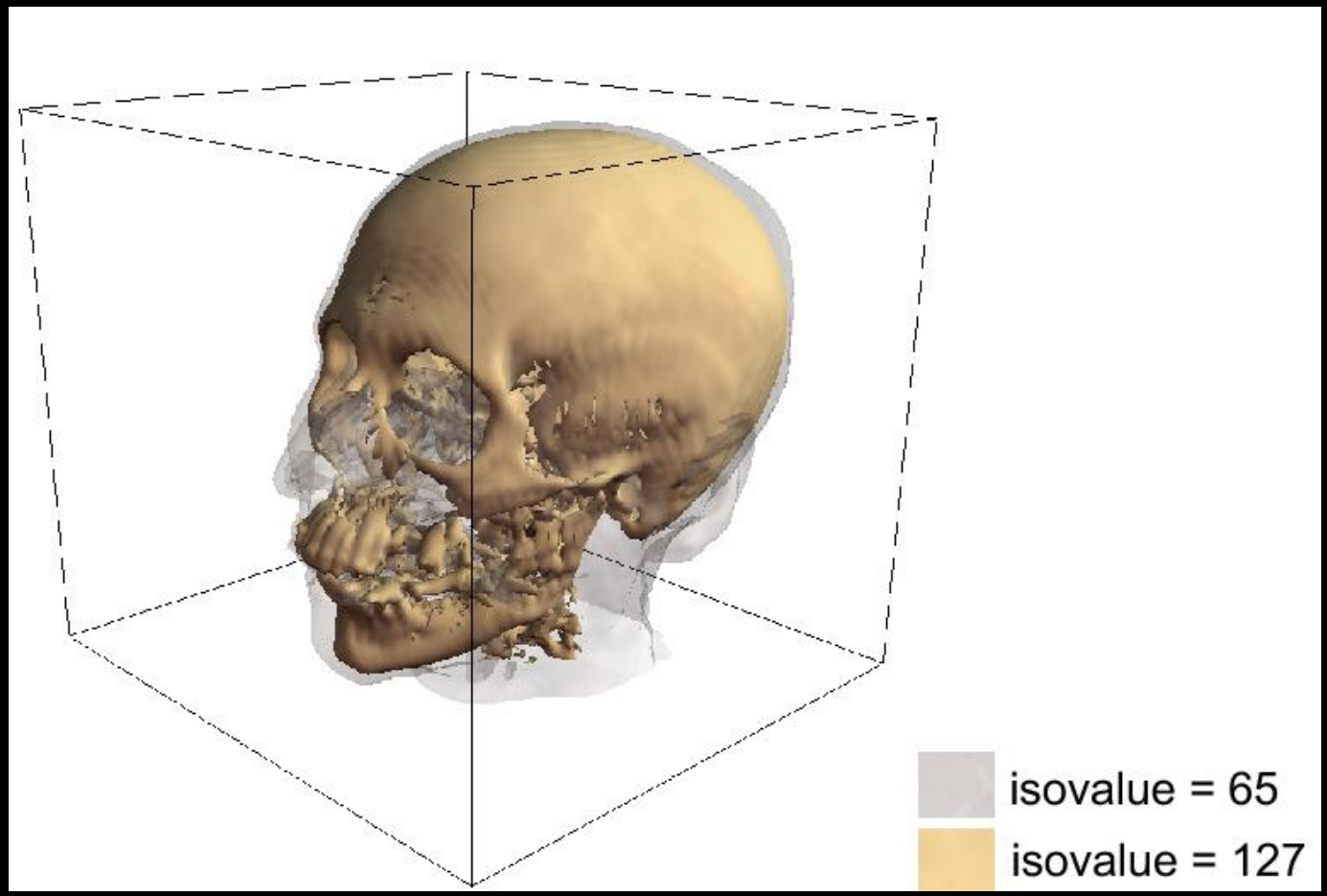


the scalar value 65
Ignores all volume points



Visualizing a 3D scalar dataset (128^3 in 0~255):
(a) Surface plot; (b) Slice plane; (c) Isosurface (Skin).
The methods reduce data dimensionality from 3D to 2D

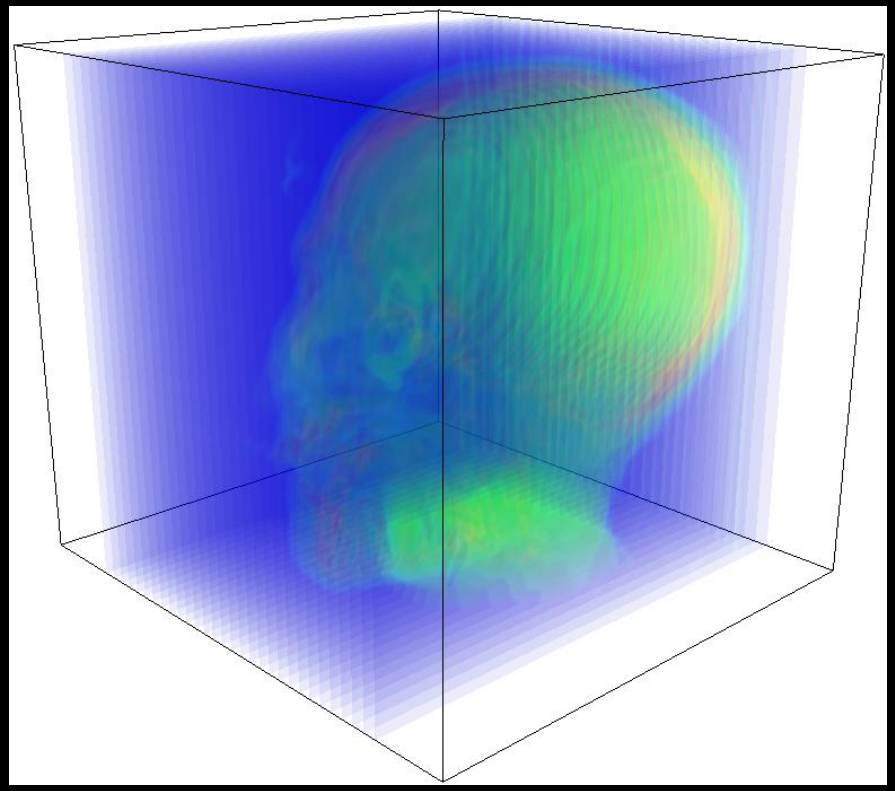
Motivation



Visualization consisting of two isosurfaces:
the skin (isovalue = 65) and bone (isovalue = 127)

Motivation

10 color-mapped slices
Orthogonal to y-axis, with slice transp. 0.1

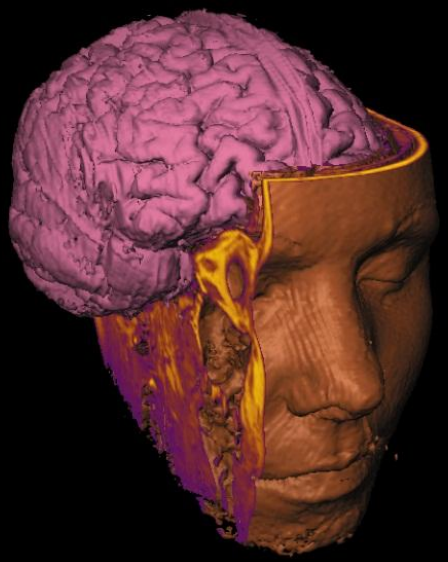


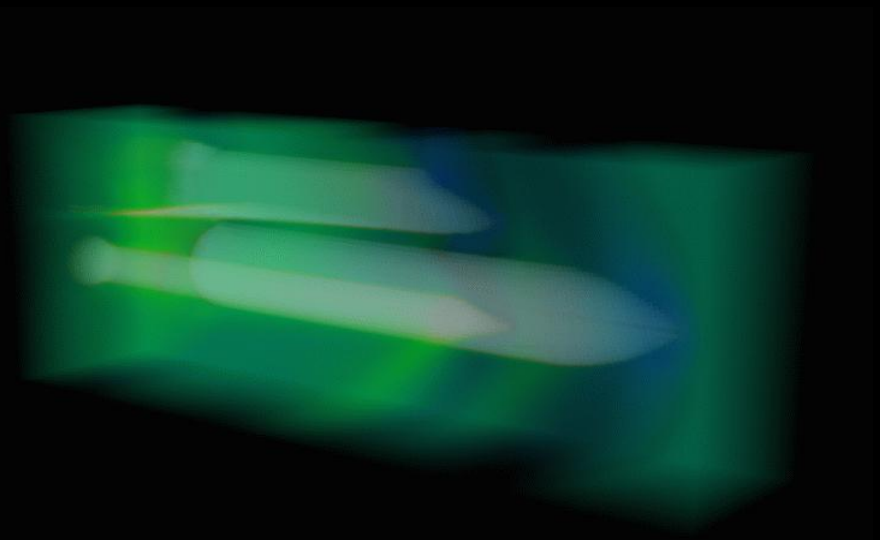
Generalized Slicing: Multiple transparent slices

Visualization of scalar volume using volume-aligned slices

Volume Rendering

- Goal: visualize three-dimensional functions
  - Measurements (medical imaging)
  - Numerical simulation output
  - Analytic functions



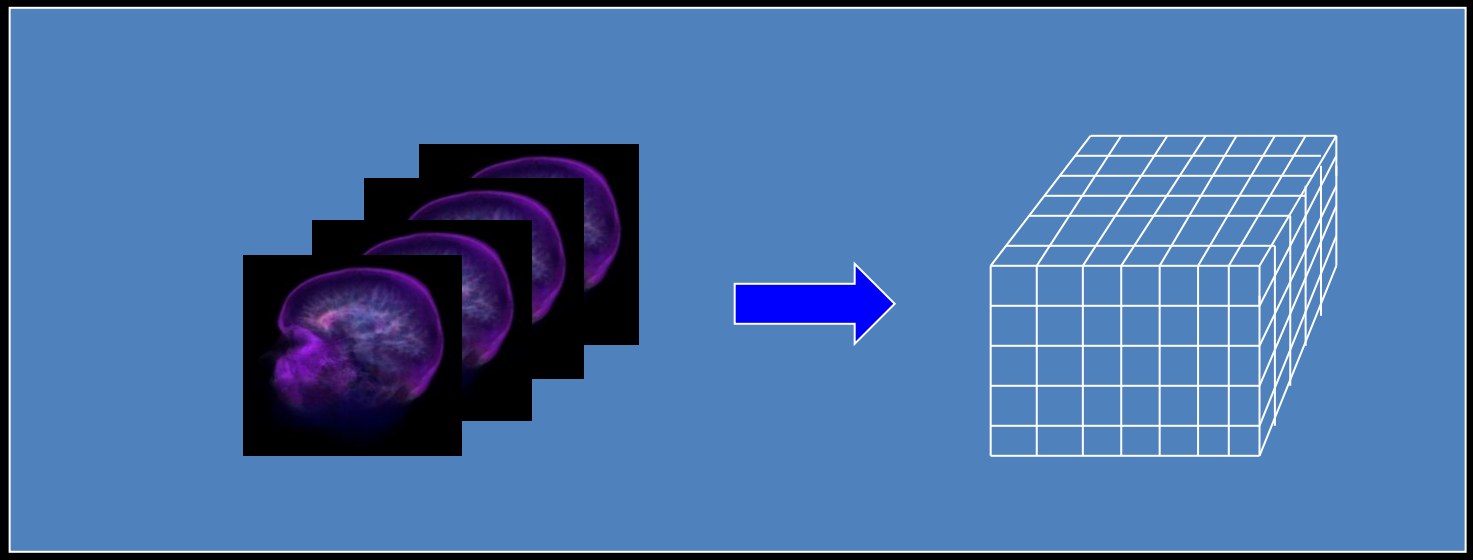


Volume Visualization Basics

- The basic idea behind volume rendering
  - Creating a 2D image that reflects, at every pixel, the scalar data within a given 3D dataset
- Main issue: the choice of the function
  - Mapping an entire set of scalar values, for the voxels along the viewing ray, to a single pixel in the resultant 2D image

Data Representation

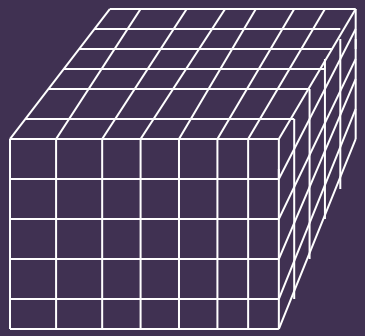
3D volume data are represented by a finite number of cross sectional slices (a stack of images)



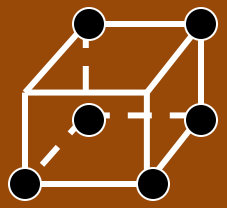
2D arrays x N => 3D array

Data Representation

What is a Voxel? – Two definitions



A voxel is a cubic cell, which has a single value covering the entire cubic region



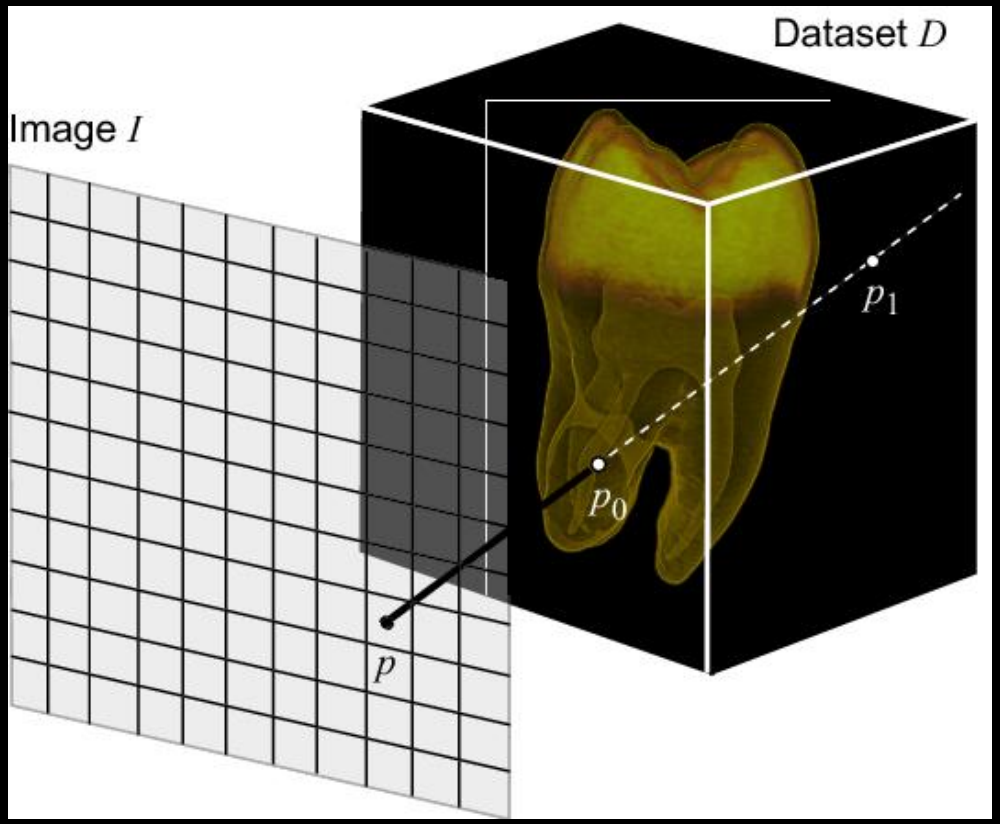
A voxel is a data point at a corner of the cubic cell; The value of a point inside the cell is determined by interpolation

Principle of Volume Vis.

The value of the image pixel p

$$I(p) = F(s(t)), t \in [0,1]$$

F : Ray function



- Create a two dimensional image that reflects, at every pixel, the data along a ray parallel to the viewing direction passing through that pixel

Principle of Volume Vis.

The design is to choose the following two functions:

- Ray function: synthesize the points along the ray

$$I(p) = F(s(t)), t \in [0,1]$$

- Transfer function: map the value of a data point on the ray to a color and opacity (RGBA) value; also called classification

$$f : R \rightarrow [0,1]^4$$

Volume Visualization Basics

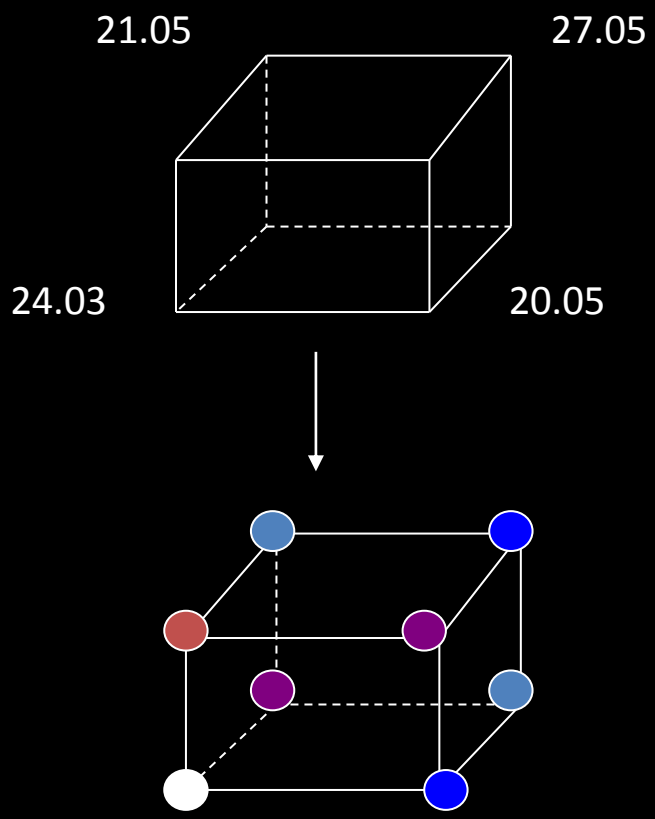
- ***Classification***

  --- The process of designing and applying transfer functions to visually separate different types of materials based on their scalar values
- Create a good classification
  - Choosing the right Transfer function

    & Ray function

Classification

- Map from numerical values to visual attributes
  - Color
  - Transparency
- Transfer functions
  - Color function: c(s)
  - Opacity function: a(s)



Various Ray functions

- ***Maximum Intensity Projection Function***
  - Maximum Intensity (scalar value) Projection (MIP)

$$I(p) = f\left(\max_{t \in [0,T]} s(t)\right)$$ (Maximum scalar value)

  - Maximum Opacity along the ray

$$I(p) = f(s_m) \qquad f_A(s_m) = \max_{t \in [0,T]} f_A(s(t))$$

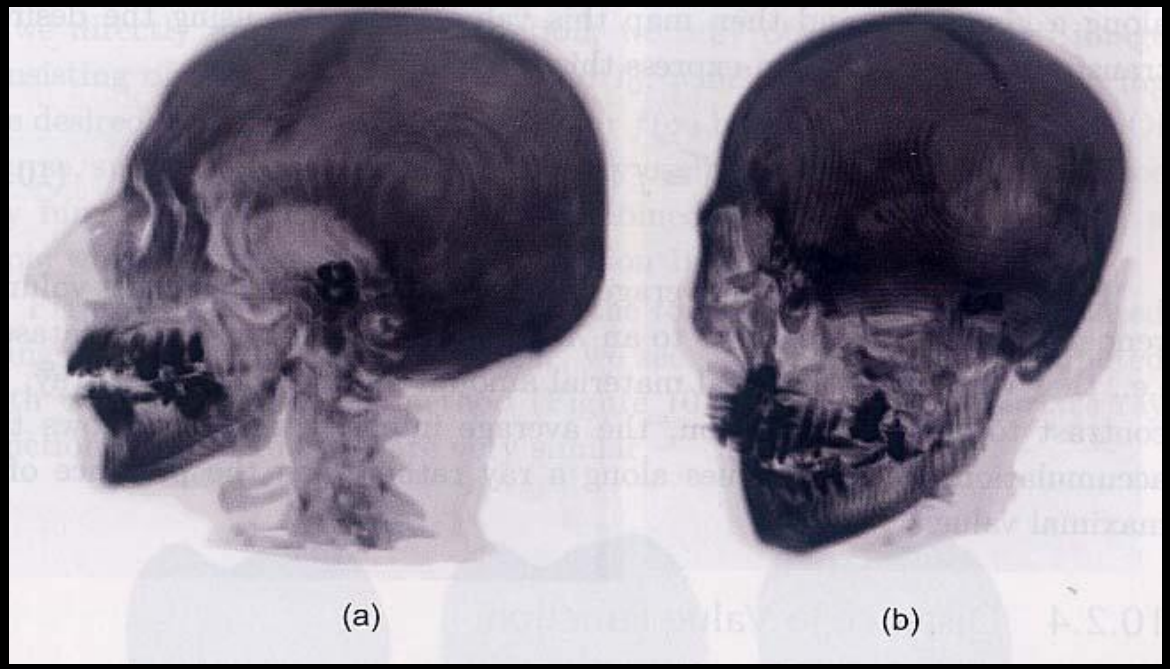
    - Useful if we want to emphasize in the rendering on the presence of a given material
  - The maximum of the intensities (colors) of all pixels computed along the viewing ray

Maximum Intensity Projection

- MIP is useful to extract high-intensity structure from volumetric data
  - e.g.: extract vascular structure from medical MRI datasets
- Disadvantage: failing to convey depth information

Maximum Intensity Projection

- Gray value proportional to the scalar value
  - White: the lowest scalar value Black: the highest value
- The left image is easier to interpret than the right one, since it is taken from an angle where the lack of depth information is not so disturbing



Maximum intensity projection rendering

Various Ray Functions

- ***Average Intensity Function***
  - A second simple ray function

$$I(p) = f\left(\frac{\int_{t=0}^{T} s(t)dt}{T}\right)$$

- Shows the accumulation of scalar values along a ray rather than the presence of a maximal value.
- Produces volume rendering analogous to an X-ray image of the considered dataset.

Various Ray Functions

- ***Distance to Value Function***
  - The 3rd ray function

$$I(p) = f\left(\min_{t \in [0,T], s(t) \geq \sigma} t\right)$$

- Useful in revealing the minimal depth
  - Within the volumetric dataset, the nearest one with its scalar value > σ
- Focusing on the *position* (depth) where a certain scalar value is met

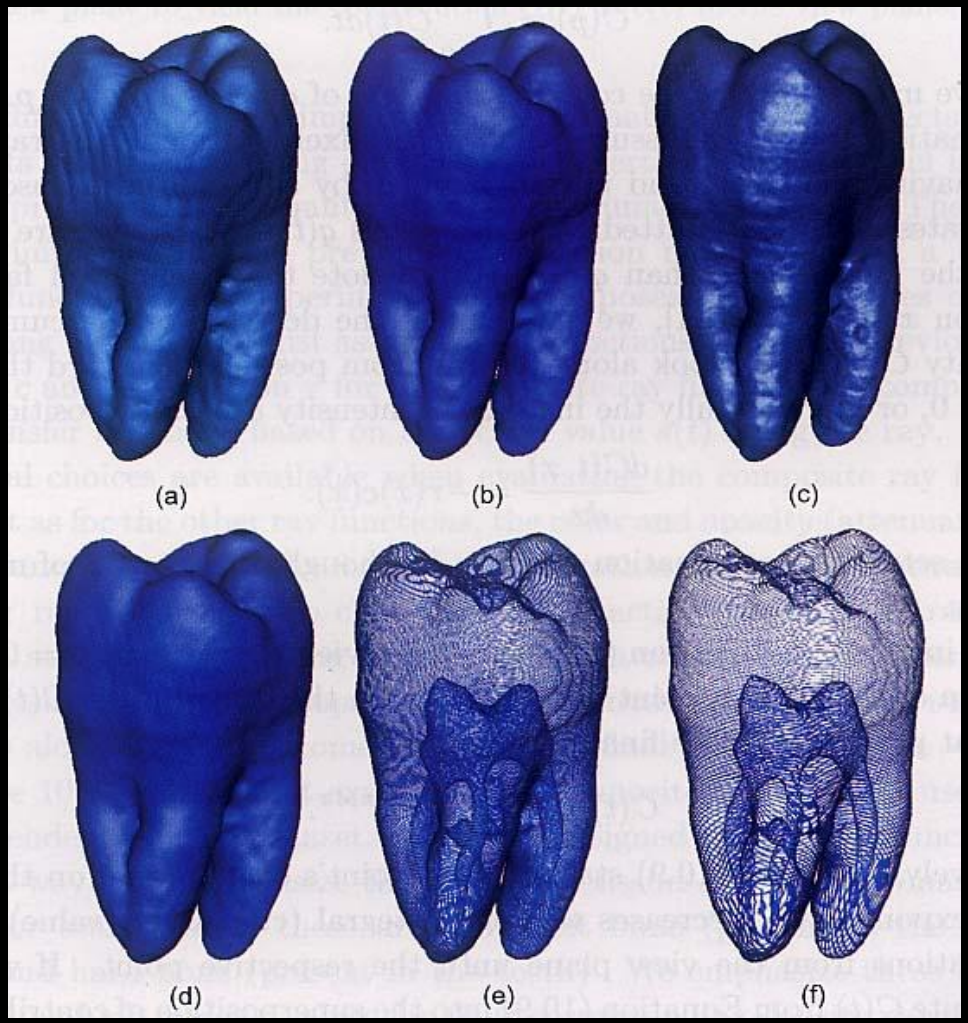
Various Ray Functions

- ***Isosurface Function***
  - Ray functions can also be used to construct familiar isosurface structure
- Ray function

$$I(p) = \begin{cases} f(\sigma) & \exists t \in [0,T], s(t) = \sigma \\ I_0 & \text{otherwise} \end{cases}$$

- In practice, the isosurface ray function becomes useful when combined with volumetric shading

Isosurface Function



Tooth volume dataset computed using different isosurface techniques:

(a) Marching cubes
(b) Isosurface ray function, software ray casting
(c) Graphics hardware ray casting
(d-f) Composition with box opacity function, different integration step sizes.

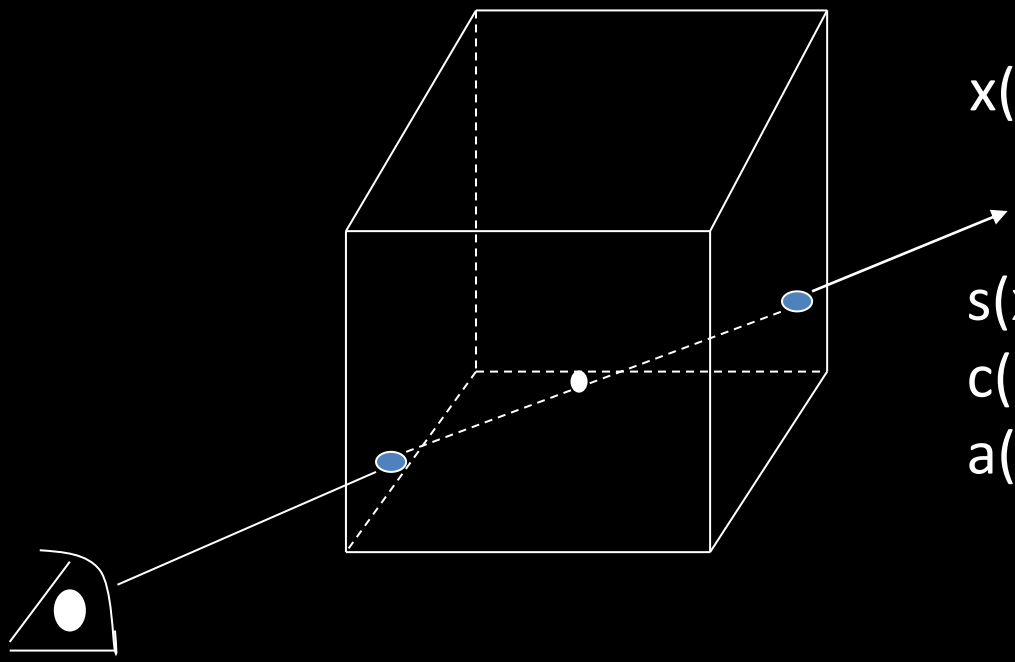
Various Ray Functions

- ***Compositing Function***
  - Previous ray functions can be seen as particular cases of a more general ray function called the *compositing function*
  - The color *C(p)*: *composition* of the contributions of the colors *c(t)* of *all voxels q(t)* along the ray *r(p)* corresponding to the pixel *p*
  - Integral of the contributions of all points along the viewing ray:

$$C(p) = \int_{t=0}^{T} C(t)dt \qquad \frac{dC(t,x)}{dx} = -\tau(x)c(x)$$

Optical Model

- Ray tracing is one method used to construct the final image



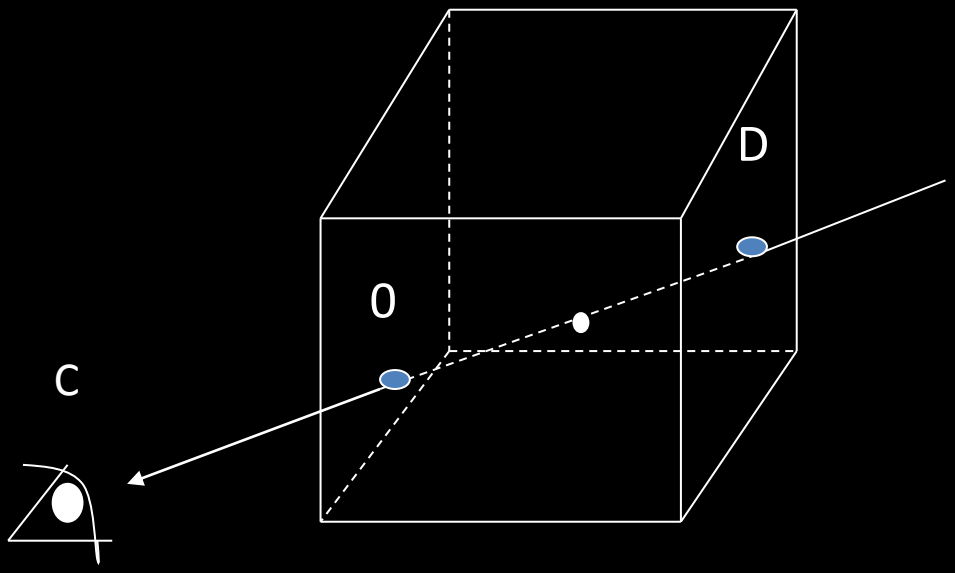
x(t) : ray, parameterized by t

s(x(t)) : Scalar value
c(s(x(t)): Color; emitted light
a(s(x(t)): Absorption coefficient

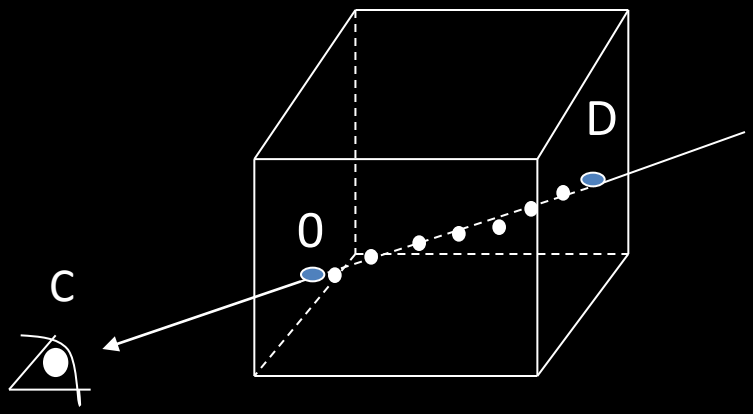
Ray Integration

- Calculate how much light can enter the eye for each ray

$$C = \int_0^D c(s(x(t))e^{-\int_0^t a(s)x(t'))dt'} dt$$



Discrete Ray Integration



$$C = \sum_{0}^{n} C_i \prod_{0}^{i-1} (1 - A_i)$$

$$C_i' = C_i + (1 - A) C_{i+1}'$$

Back to front blending:
step from n-1 to 0

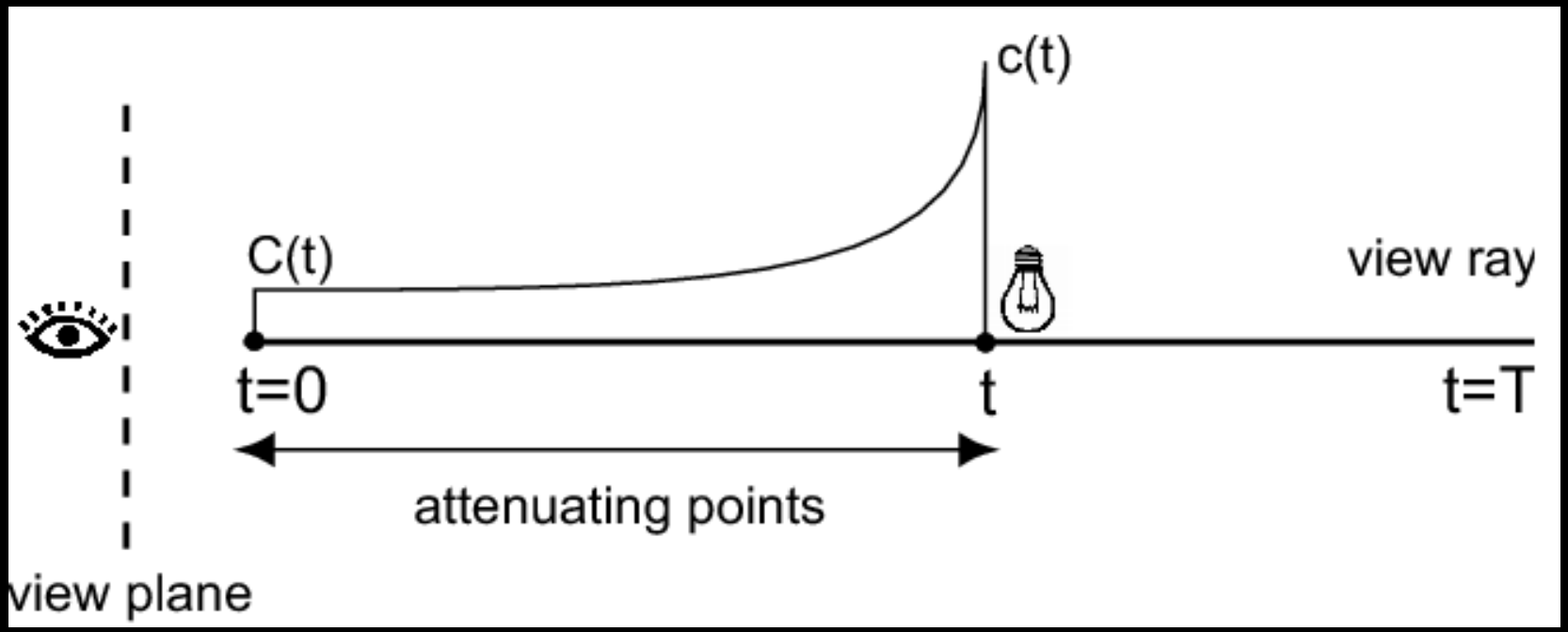
Compositing Function

- The pixel color: $C(t) = c(t)e^{-\int_0^t \tau(x)dx}$
- The above formula states that a point's contribution on the view plane exponentially decreases with the integral of the attenuations from the view plane until the respective point.
- Integral illumination model

$$C(p) = \int_{t=0}^{T} c(t)e^{-\int_0^t \tau(x)dx} dt$$

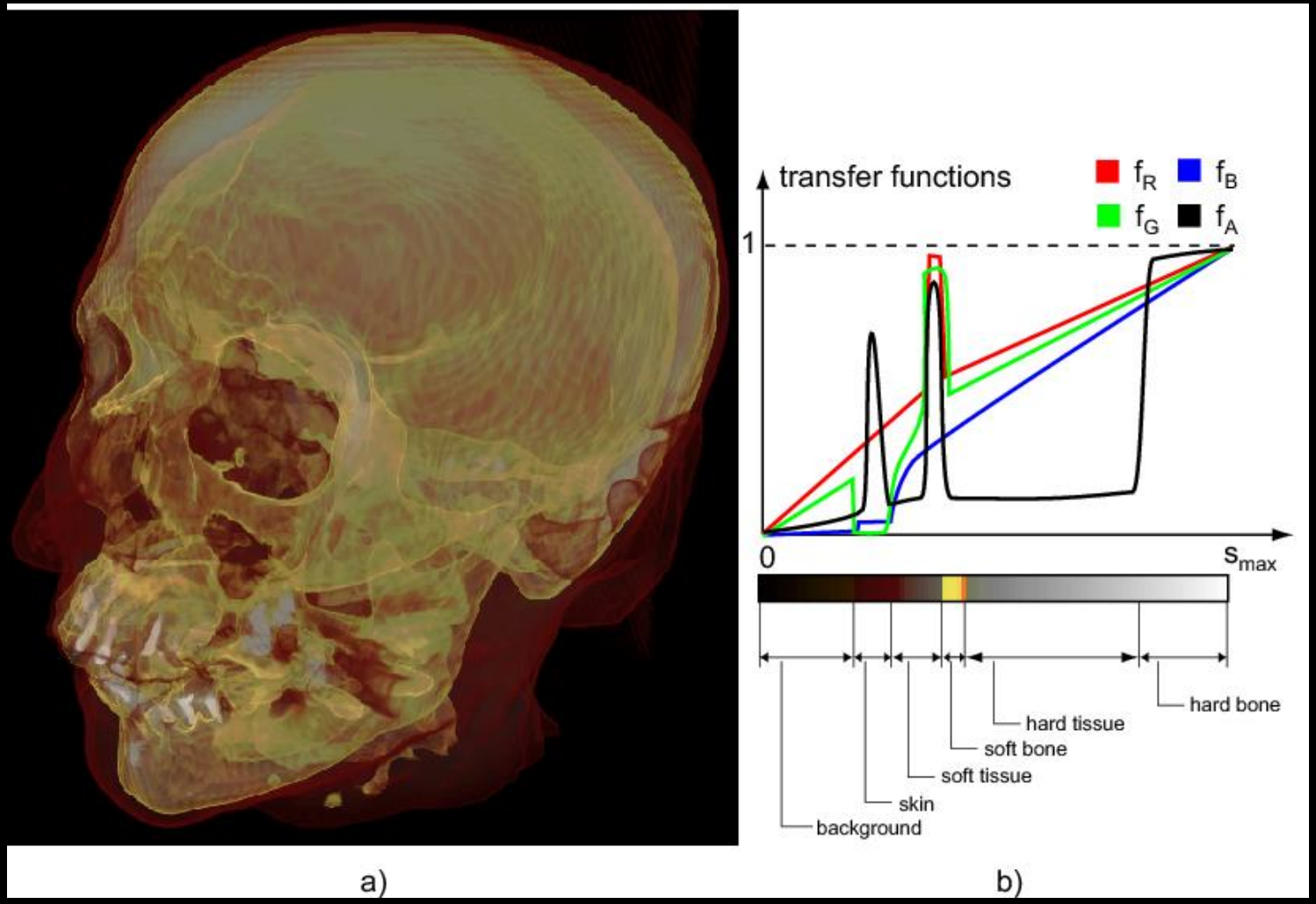
  - Neglects several effects such as scattering or shadows
  - Capable of producing high-quality images of volumetric datasets

Compositing Function



Volumetric illumination model: color c(t) emitted at position t along a view ray gets attenuated by the values Tao(x) of the points x situated between t and the view plane to yield the contribution C(t) of c(t) to the view plane.

Compositing Function



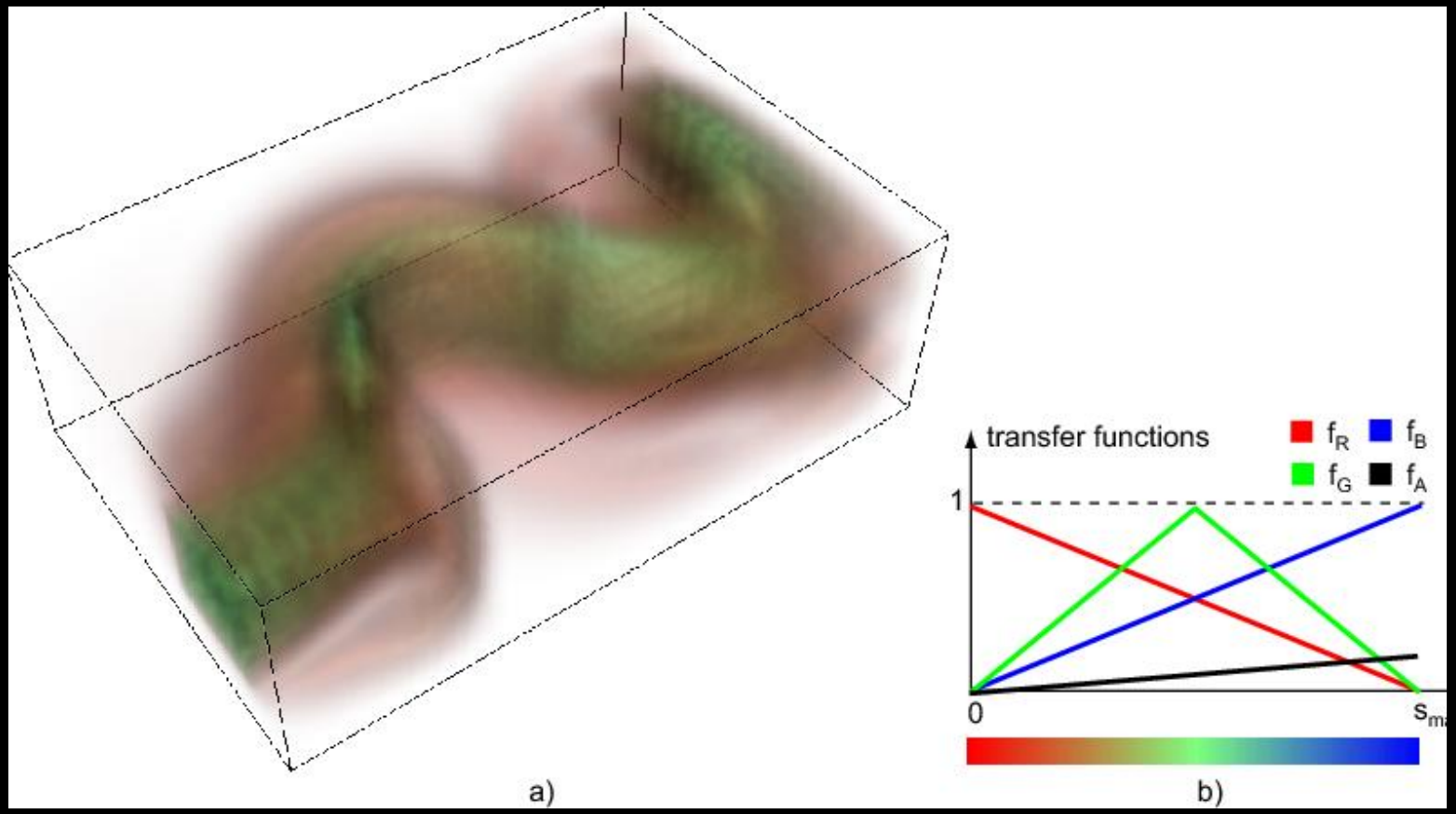
(a) Volume rendering of head dataset. (b) The transfer function used emphasizes soft tissue, soft bone, and hard bone. Using high-opacity values for their corresponding density ranges

Compositing Function

- The design of appropriate color and opacity transfer functions
  - The transfer and opacity functions are used to visually separate different tissues , and also have smooth variations across the transition area rather than abrupt, step-like jumps

Compositing Function

Volume rendering can also be applied to other datasets than scanned datasets containing material density values



(a) Volume rendering of flow field velocity magnitude and **(b)** Corresponding transfer functions.

Compositing Function

- Volume rendering of any scalar fields are possible, the results can sometimes be harder to interpret
  - CT and MRI datasets show structures that often are easier to interpret than arbitrary volumetric scalar fields
  - Some volume datasets exhibit no natural boundaries between regions with different scalar values

Volumetric Shading

- Shading is an important additional cue that can significantly increase the quality of volume rendering
- illumination function (Phong lighting algorithm)

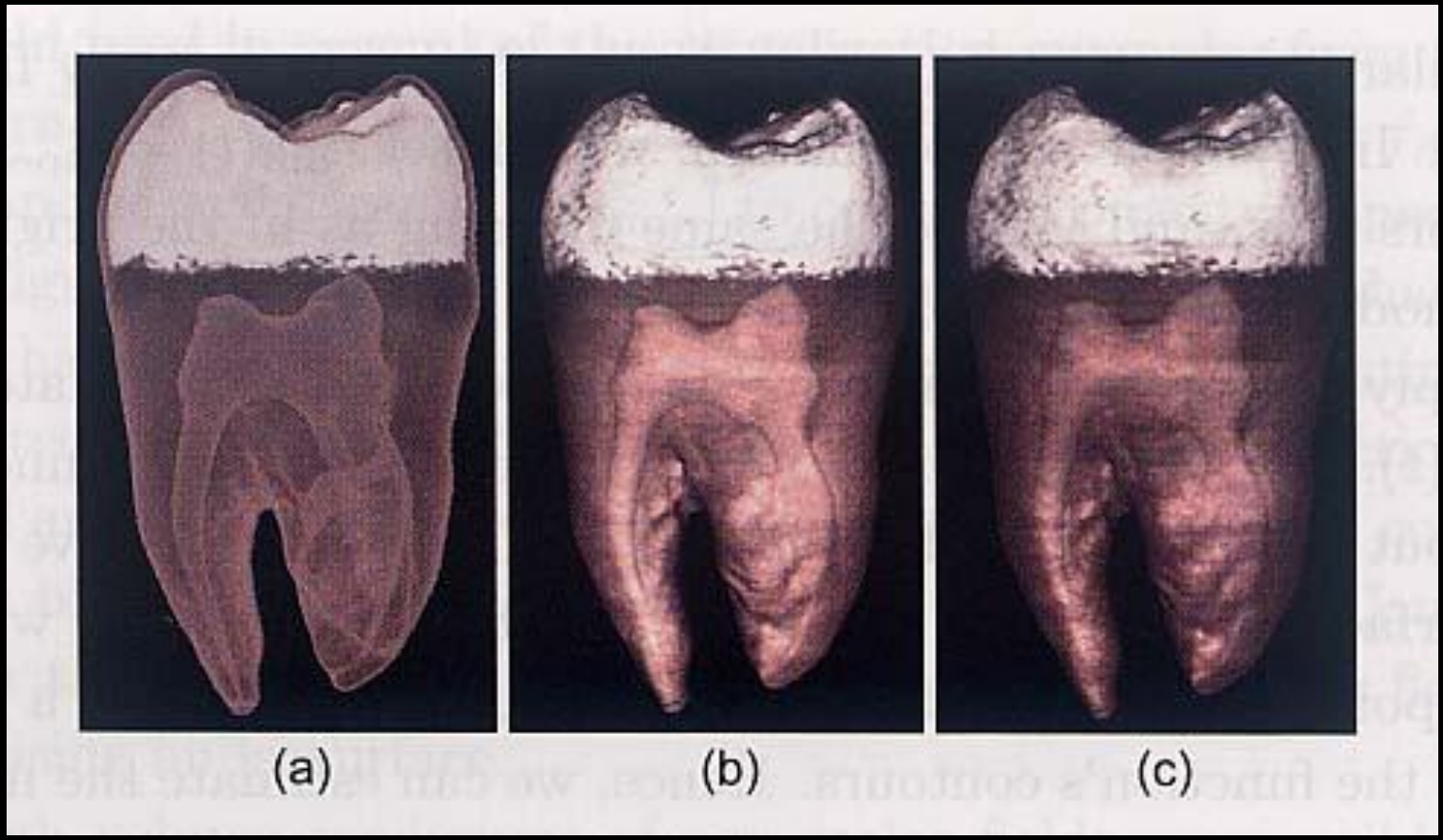
  C = ambient + diffuse + specular

  = constant + Ip Kd (N.L) + Ip Ks (N.H)^n

$$I(t) = c_{amb} + c_{diff}(t)\max(-L \cdot n(t), 0) + c_{spec}(t)\max(-r \cdot v, 0)^{\alpha}$$

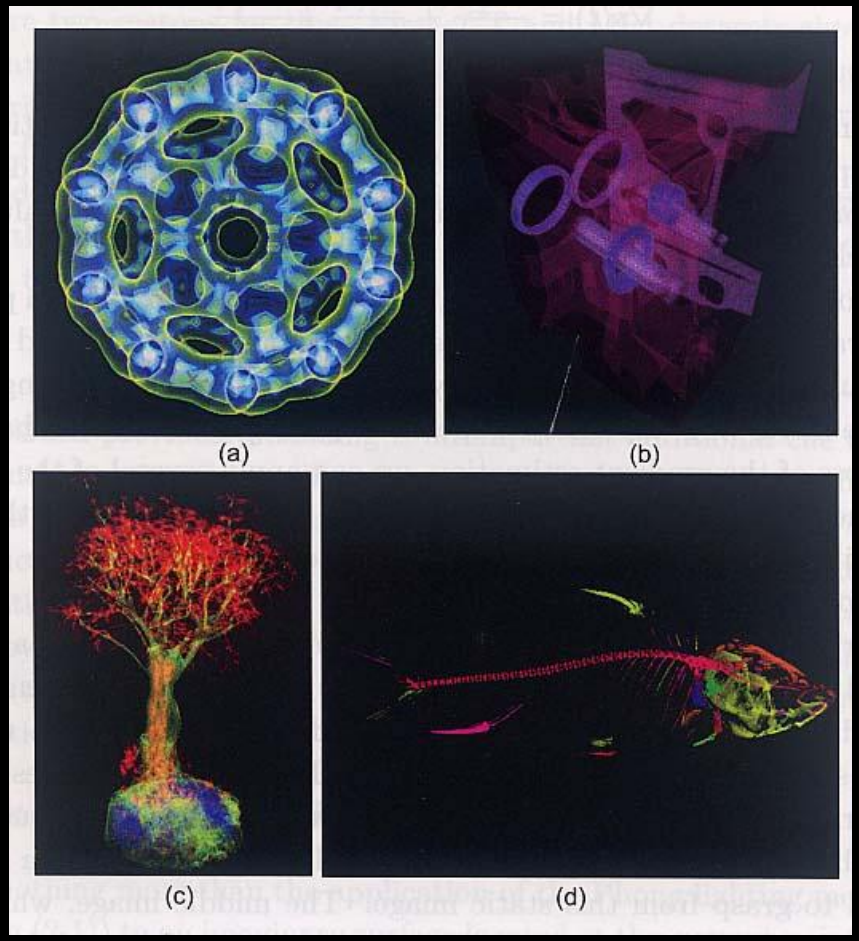
Volumetric Shading

(b) & **(c)** are easier to understand due to the shading cues



Volumetric lighting. **(a)** No lighting. **(b)** Diffuse lighting. **(c)** Specular lighting.

Volumetric Shading



Examples of volume rendering:
(a) Electron density. (b) Engine block.
(c) Bonsai tree. (d) Carp fish.

Volume rendering allows us to create insightful, but also aesthetically pleasing renderings of volumetric datasets

Image Order Techniques

- Volumetric ray casting
  - The most straightforward way to implement compositing function
  - Evaluate the rendering integral by taking samples along the viewing rays
- Pseudocode

```
for(all pixels p in the image plane I)
{
    v = ray perpendicular to I passing through p;
    q0, q1 = intersections of v with the volume;
    C(p) = 0; //color of pixel p
    for(float t=0;t<1;t+=Δt)
    {
        q = (1 − t)q0 + tq1;
        C(p)+ = c(t)e^(−∫0^t τ(x)dx) Δt;
    }
}
```

Image Order Techniques

- **Computation strategies**

* Integral illumination model

$$C(p) = \int_{t=0}^{T} c(t) e^{-\int_0^t \tau(x)dx} dt$$

* Approximate the exponential term of the inner sum using Taylor expansion. In simple format

$$C(p) = \sum_{i=0}^{N} c_i \left(\prod_{j=0}^{i-1} (1 - \tau_j) \right)$$

Image Order Techniques

- Evaluate the above formula in back-to-front order

$$C_N = c_N$$
$$C_{N-1} = c_{N-1} + (1-\tau_{N-1})c_N$$
$$C(p) = C_0 = c_0 + (1-\tau_0)c_1 + (1-\tau_0)(1-\tau_1)c_2 + \cdots$$

- Evaluate the composite ray function

$$C_i = c_i + (1-\tau_i)C_{i+1}$$

Computation:

$C_{out} = C_{in} + C(x)*(1- \alpha_{in})$

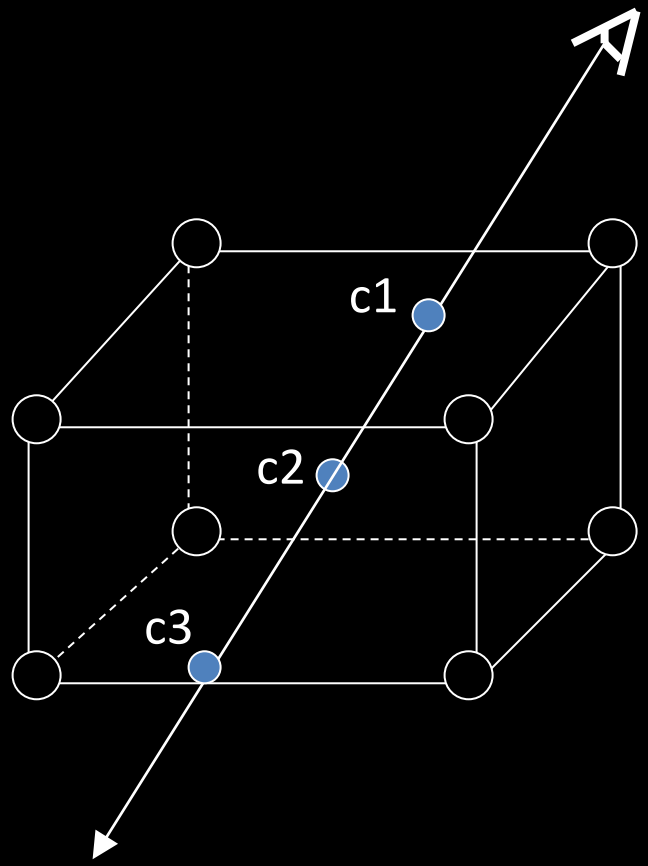
$\alpha_{out} = \alpha_{in} + \alpha(x) *(1- \alpha_{in})$

Compositing method

Or you can use ‘Front-to-Back’ Compositing formula

Front-to-Back compositing: use ‘over’ operator

C = background ‘over’ C1
C = C ‘over’ C2
C = C ‘over’ C3
…



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

Sampling and Interpolation Issues

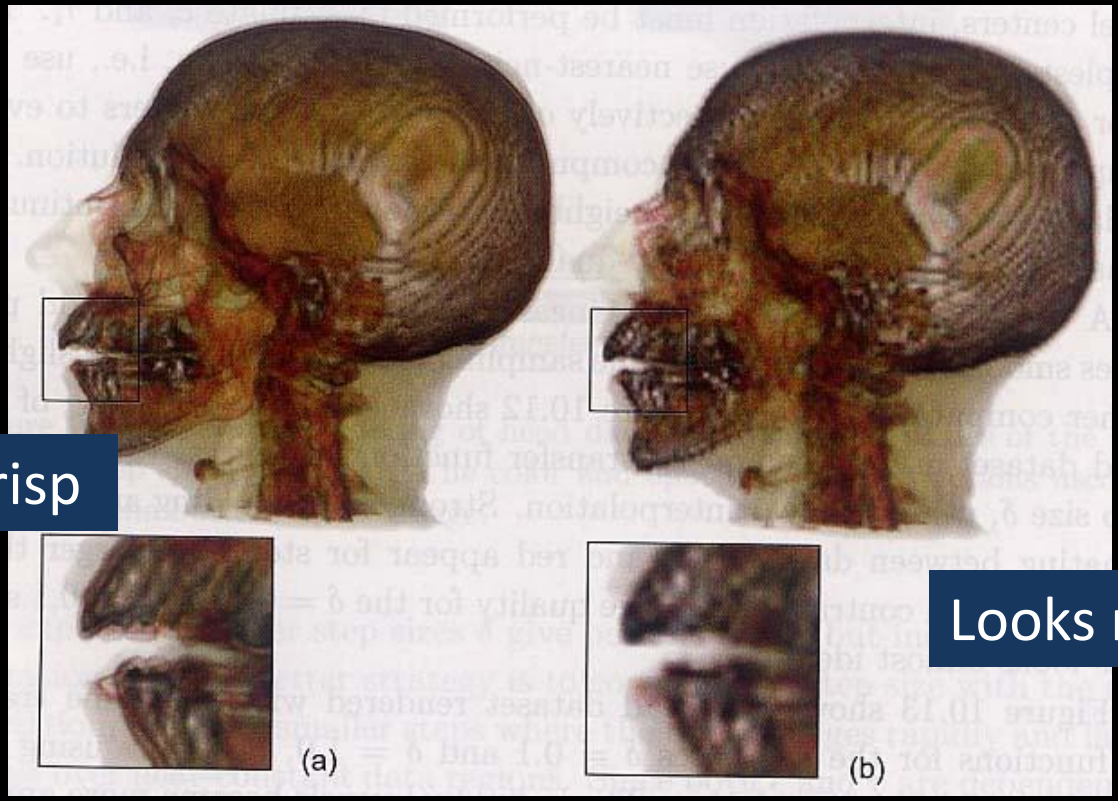
- The quality of a volume-rendered image depends on the accuracy of evaluating the discretized, two main issues:
  - * The choice of the step size δ
  - – The interpolation of color c and opacity τ along the ray
  - – Smaller step size δ gives better results, but increases the computation time
    - A better strategy is to correlate the step size with the data variation
  - * Since the sample point i along a ray will not coincide with voxel center, interpolation must be performed to evaluate c_i and τ_i
    - Better solution: trilinear interpolation

Classification and Interpolation Order

- Two choices with respect to the order of classification
  - *Pre-classification*: first classify, then interpolate
    - Generally produces coarser-looking images
    - Color interpolation can sometimes produce wrong results
  - *Post-classification*; first interpolate, then classify
    - Produces smoother images that only contain valid colors from the corresponding colormap
    - Scalar interpolate
      - Disadvantage: may yield values that correspond to nonexistent materials at points where the sampled dataset exhibits inherent discontinuities
  - The results of the two methods look very similar for
    - Smoothly varying datasets and transfer function

Classification and Interpolation Order

Looks quite crisp



Looks more blurred

Comparison of (a) post-classification and (b) pre-classification techniques. The insets show a zoomed-in detail region from the large image.

Object Order Techniques

- A second class of volume rendering : *object-order techniques*
  - Traverse each object voxel once
  - Evaluate its contribution to the image pixel where ray intersects that voxel
  - Image-order vs. object-order

    (visit every pixel once vs. multiple times)

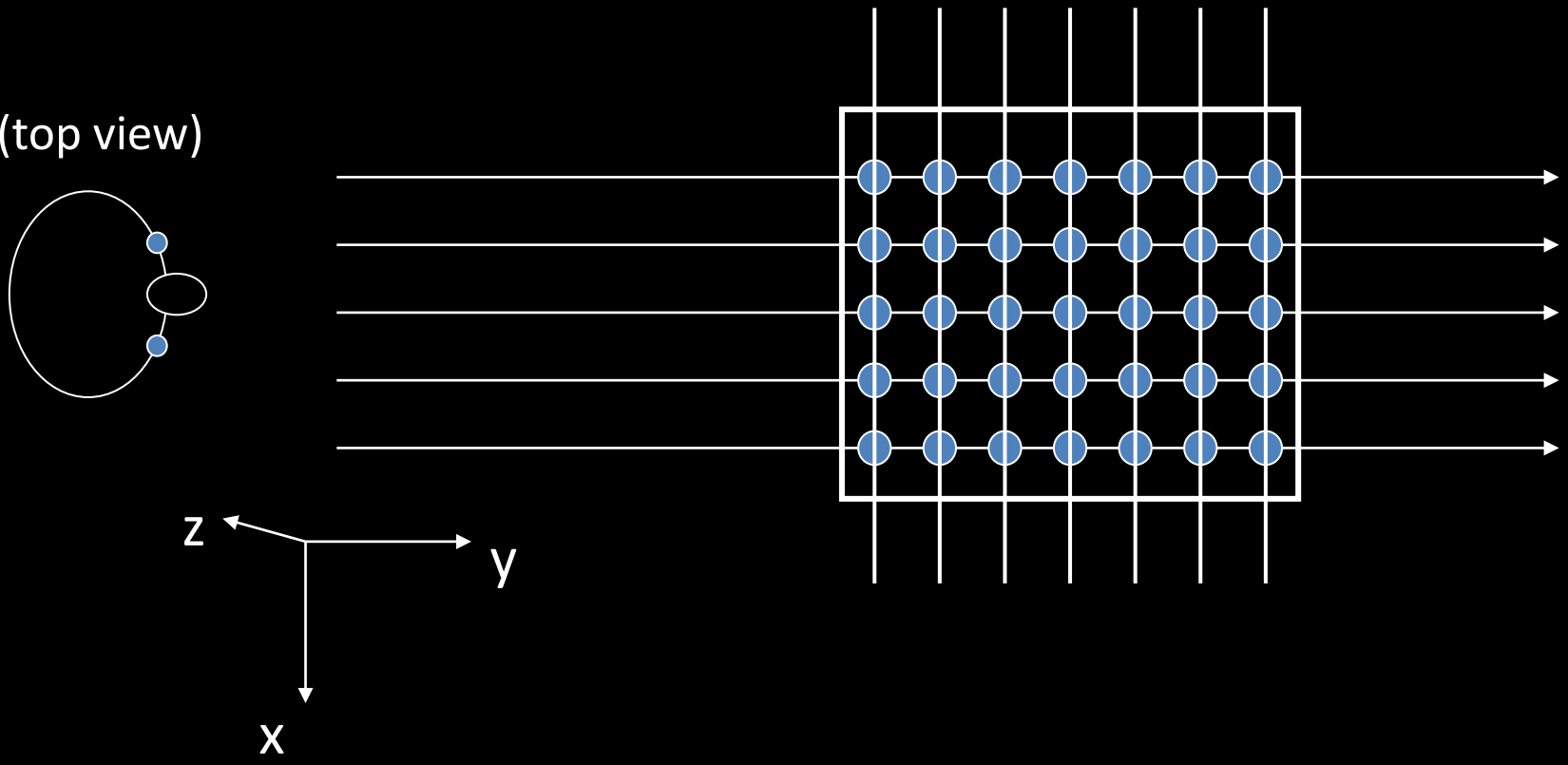
Object Order Techniques

- One most popular method is *volume rendering using textures* (possibly accelerated by graphics hardware)
  - Two subclass:
    - 2D texture supported: slice the 3D volume with a set of planes orthogonal to the volume axis, parallel to the viewing direction
      -- Simple to implement; but image quality influenced by the viewing angle
    - 3D texture supported: loaded with the color and opacity transfer functions applied on the entire dataset
      -- The result is functionally the same as 2D, but of a higher quality

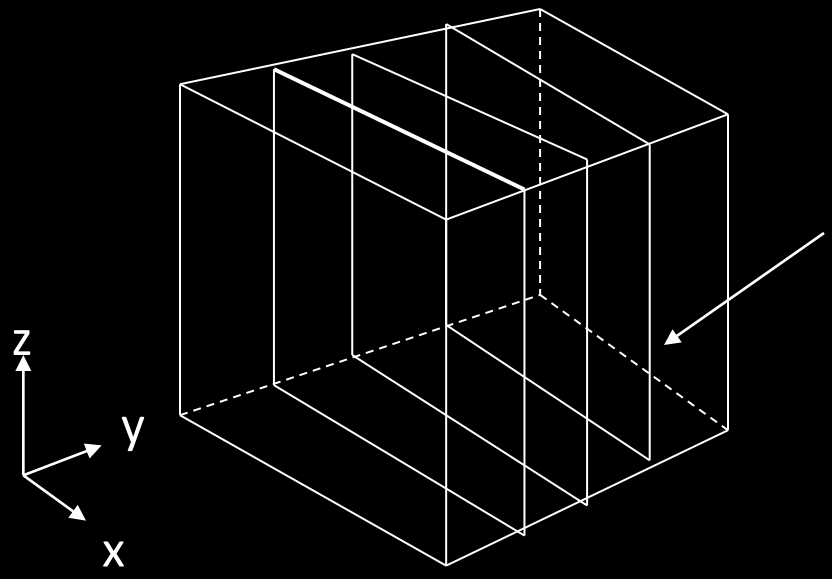
Tex. Mapping for Volume Rendering

Consider ray casting …

(top view)



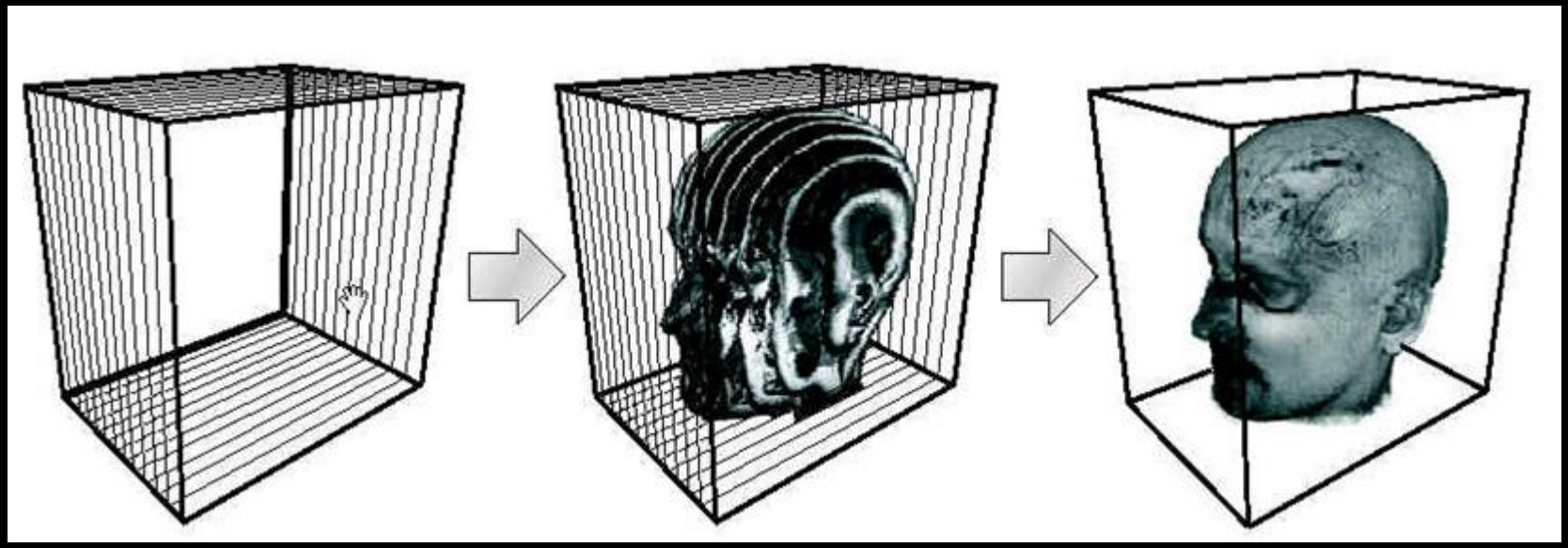
Texture based volume rendering



Use pProxy geometry for sampling

- Render every xz slice in the volume as a texture-mapped polygon
- The proxy polygon will sample the volume data
- Per-fragment RGBA (color and opacity) as classification results
- The polygons are blended from back to front

Texture based volume rendering



Polygon Slices 2D Textures Final Image

Volume Rendering vs Geometric Rendering

Volume rendering vs. Geometric rendering

- Similar aim: producing an image of volumetric dataset that gives insight into the scalar values within
- The *complexity* of the two types of techniques
  - Marching cubes vs. ray-casting techniques

    influenced by the window size

Conclusion

- Volume Visualization (volume graphics and volumetric rendering)
  - Encompasses the set of techniques aimed at visualizing 3D datasets stored at uniform (voxel) grids
  - Mainly used to visualize scalar datasets
  - Frequently in medical practice (CT and Magnetic Resonance Images)
- The key element of volume visualization:
  - By rendering a 3D dataset using appropriate per-voxel transfer function -- Mapping data attributes to opacity and color
- In practice, volume rendering is typically combined in application with slicing, probing, glyphs, and isosurfaces