Volume Visualization

Leonid I. Dimitrov and Miloš Šrámek
Commission for Scientific Visualization
Austrian Academy of Sciences

Overview

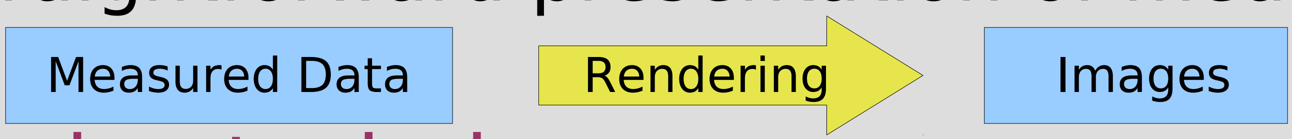
- Volume visualization
  - Volume viewing
  - Mapping
  - Volume rendering
    - Direct volume rendering
    - Isosurfacing

Volume Data

- A spatial sequence of 2D images – slices
- Produced by
  - 3D scanners (tomographs)
    - Different physical background
    - Different and complementary properties
  - Simulation

Volume Visualization

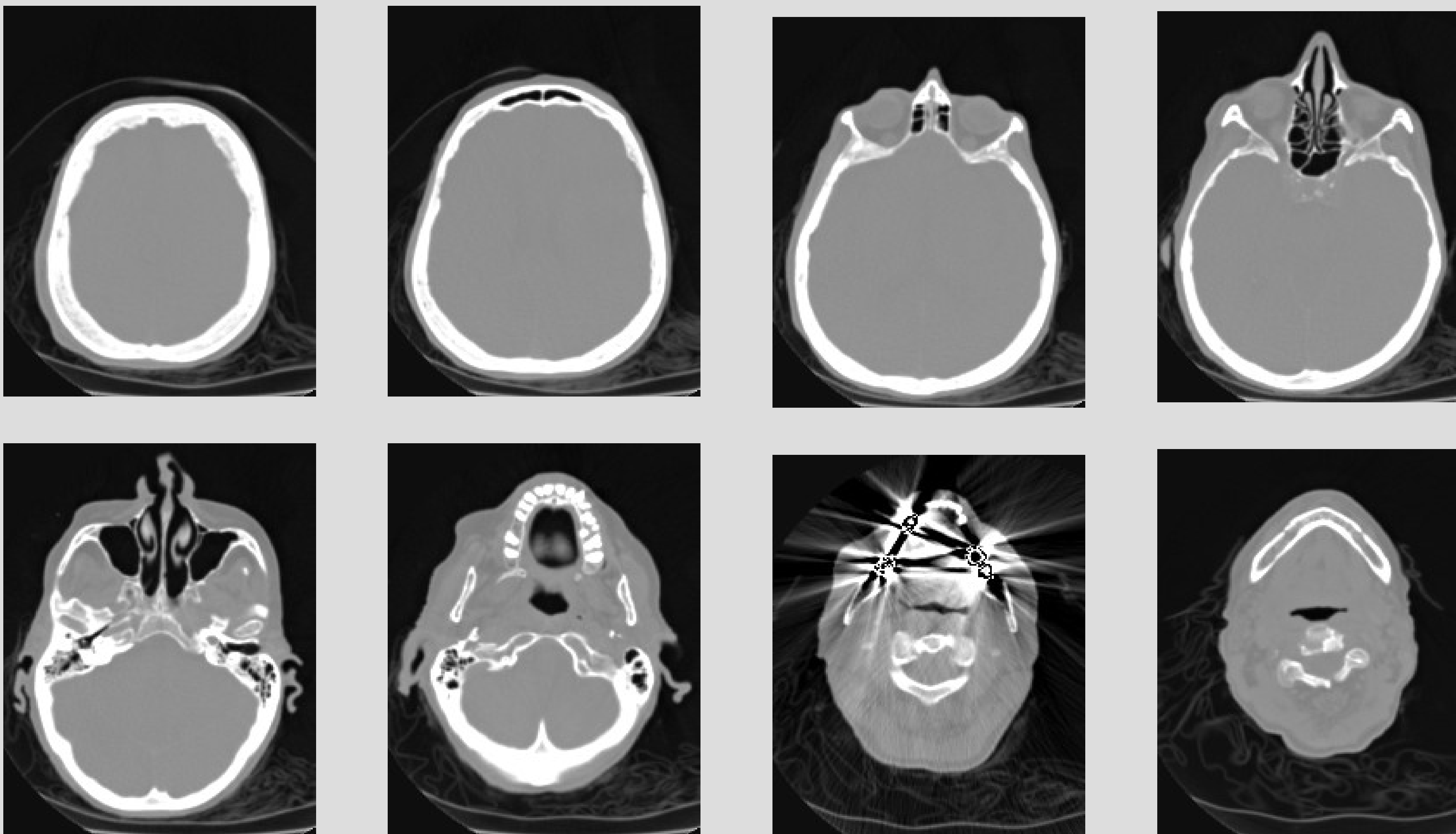
- Visually perceivable data presentation
- Understanding, not photorealism
- Simple volume viewing
  - Straightforward presentation of measured data

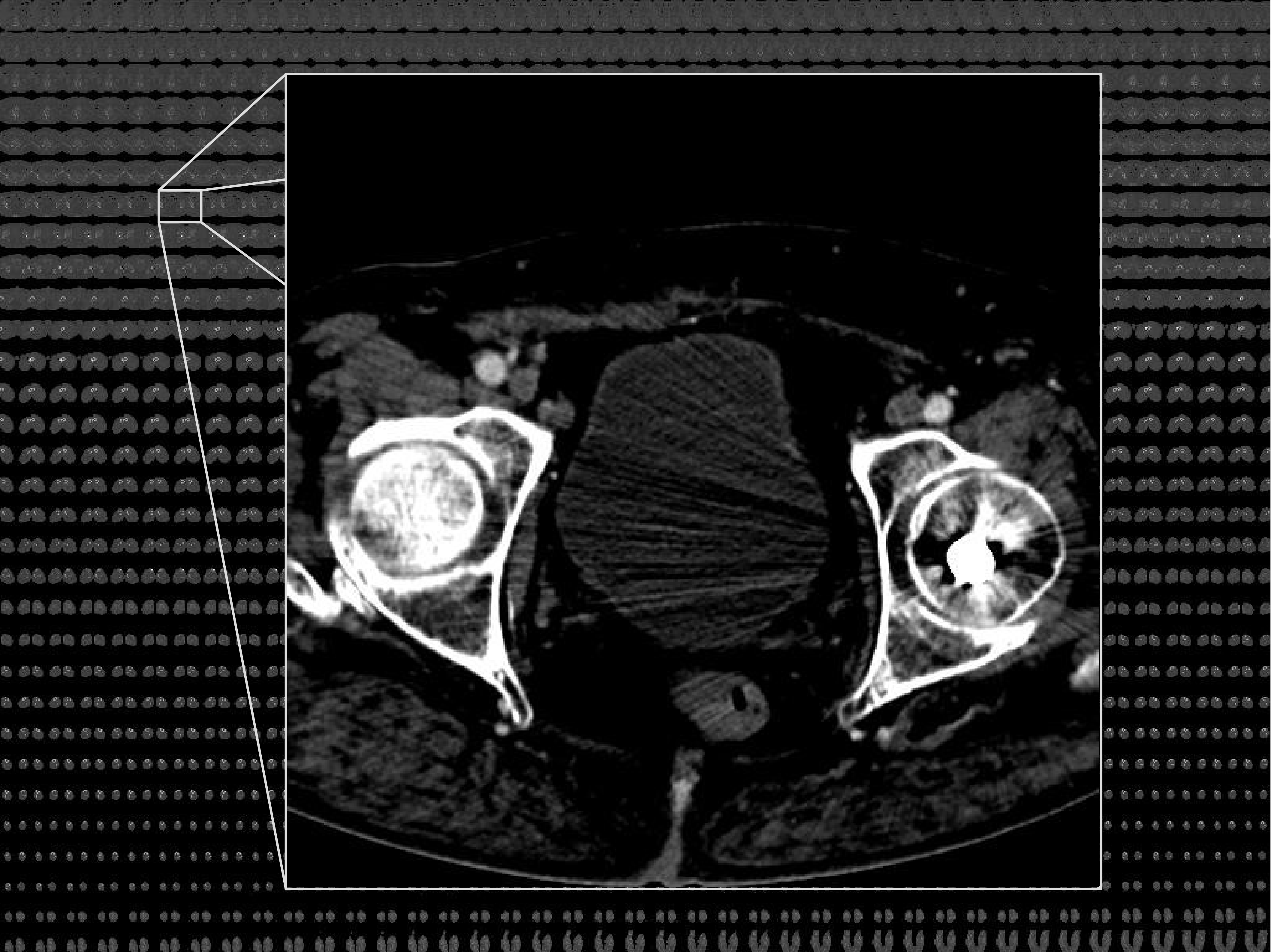


- Mapping techniques
  - Measured densities are mapped to visual attributes (transparency, color)

Volume Viewing (1)

- Slice-by-slice viewing

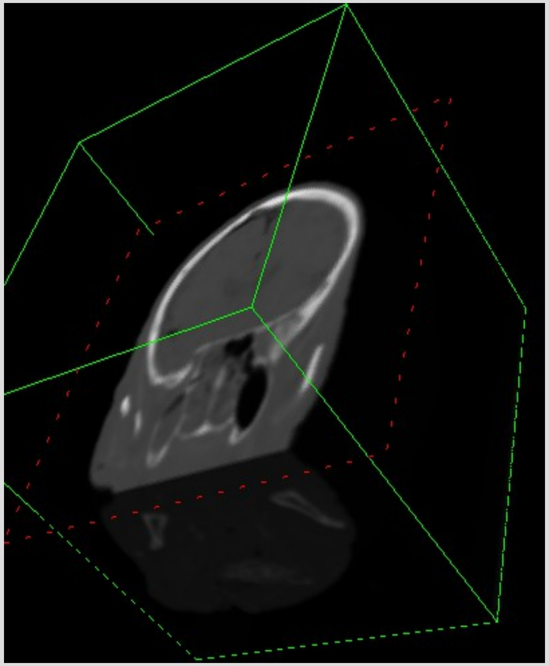




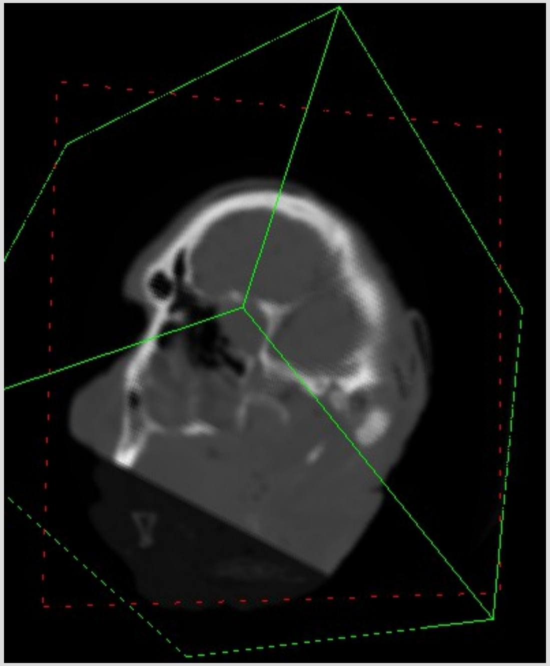
Volume Viewing (2)

- Multiplanar reconstruction
  - Definition of new cutplanes

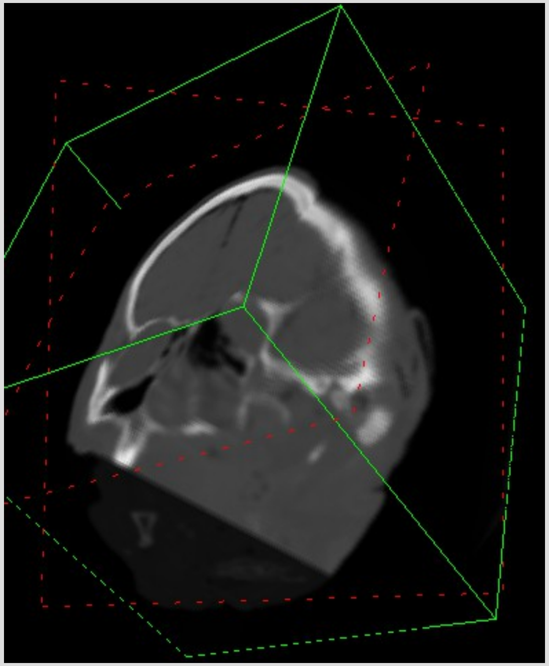
Axis aligned



Oblique

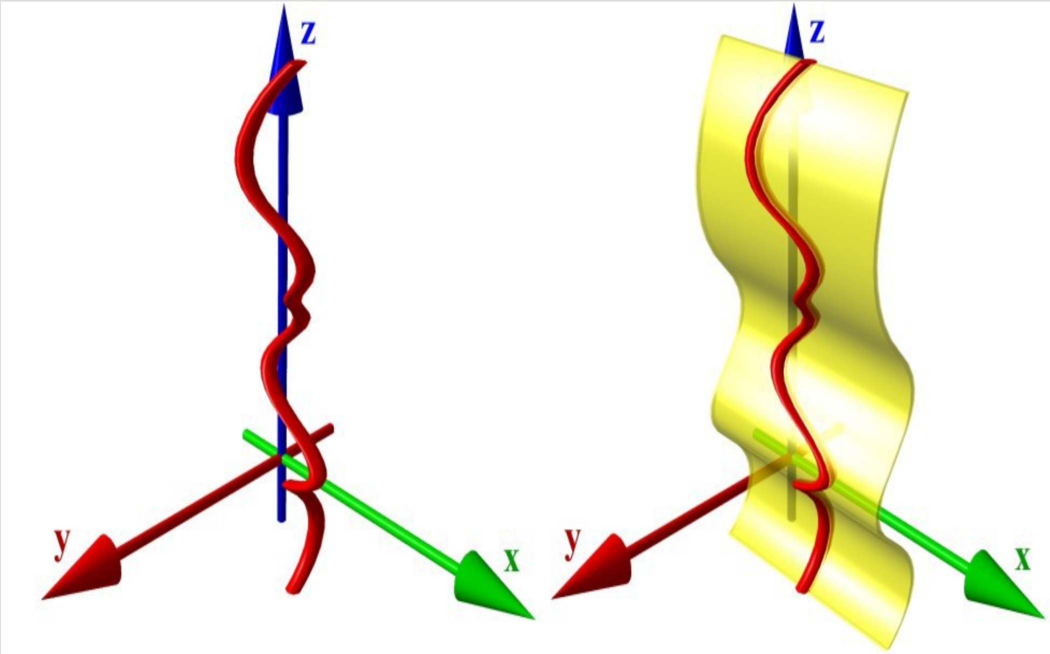


Combined



Volume Viewing (3)

- Curved planar reconstruction
  - Volume cutting along a line



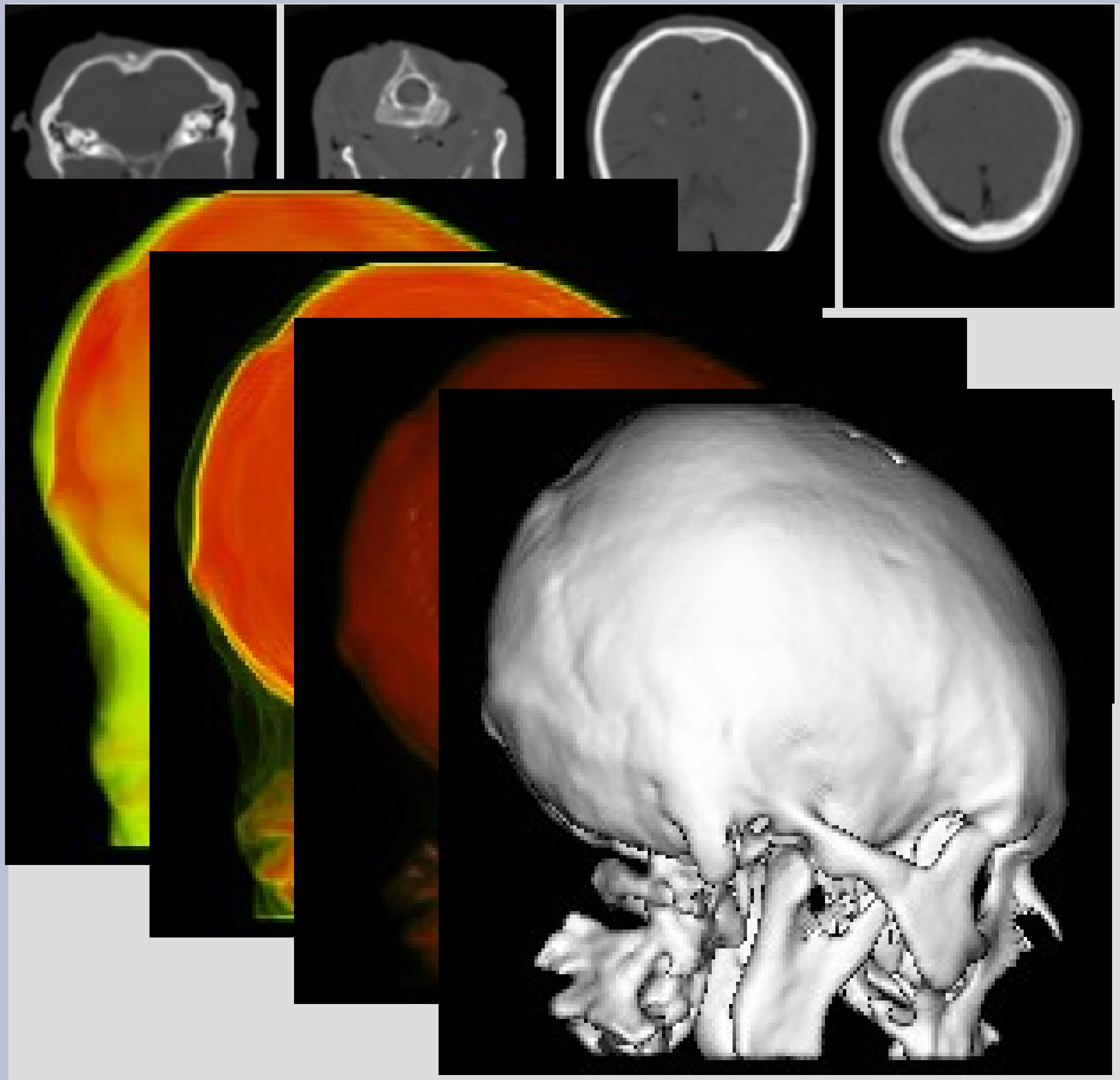
Kanitsar 2002

Volume Viewing (4&5)

- Reproojection
  - Add all values along a viewing ray
  - Simulation of X-ray projection

- Maximum intensity projection
  - Register the brightest value along a viewing ray
  - Suitable for thin structures

Volume Visualization by Mapping



Data Acquisition

CT, MRI, USG, PET, SPECT

Mapping

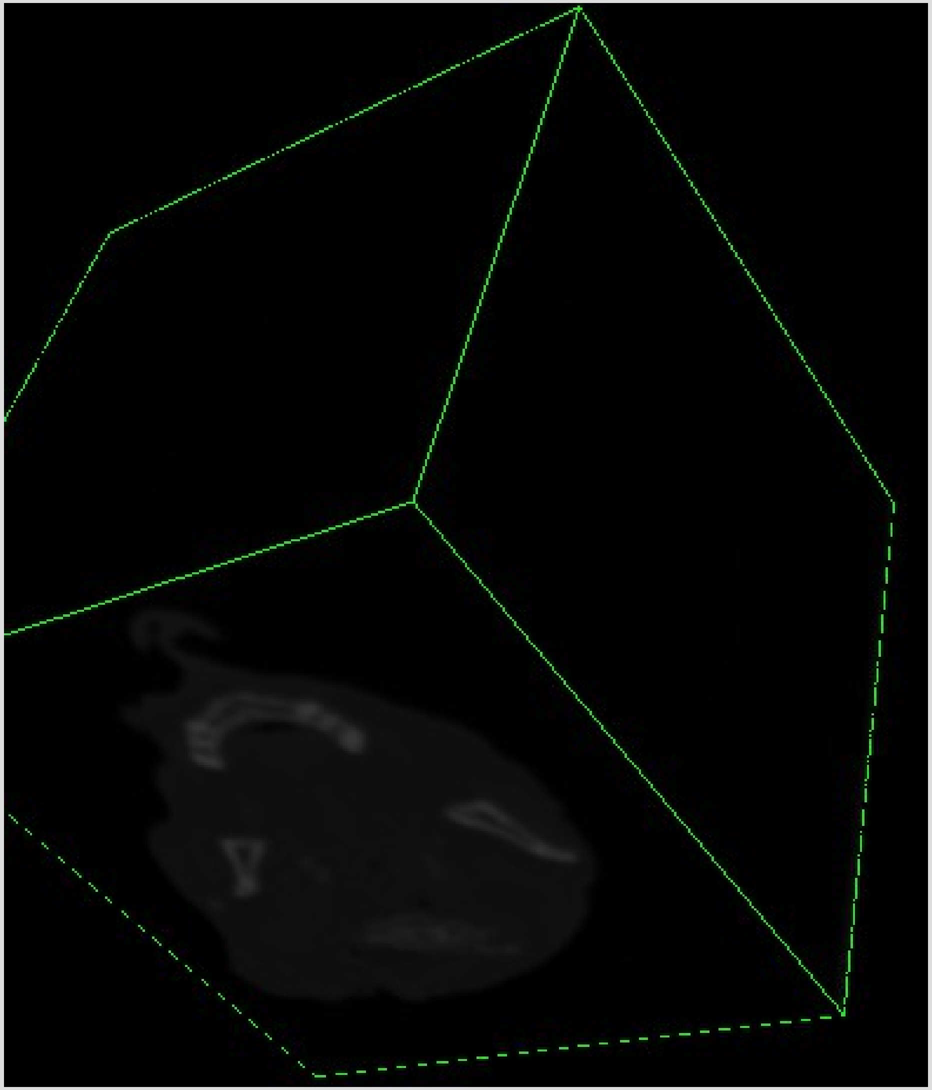
Visual attributes

Rendering

Surface & Volume Techniques

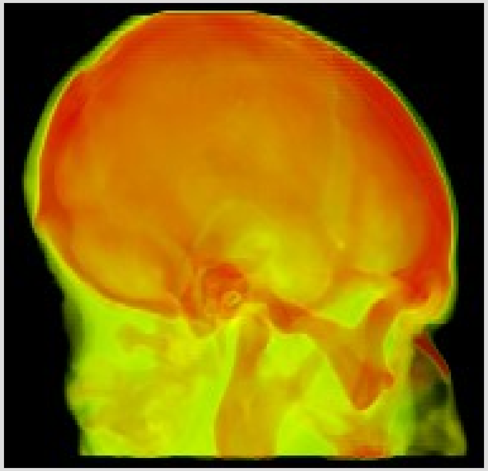
Why Do We Need Mapping?

- No visual representation readily exists for 3D data
- Area of interest is occluded by the black background
- We need something to make the background transparent



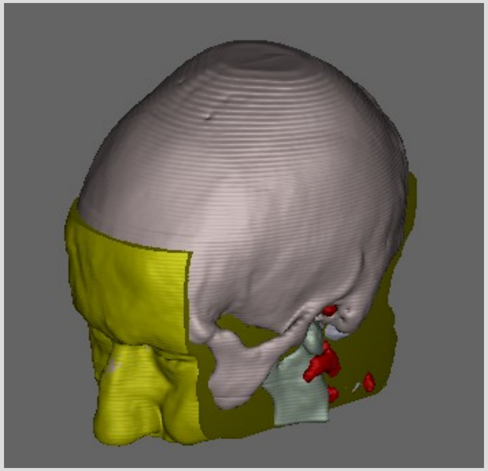
Assignment of Visual Attributes

- Mapping: Assignment of visual attributes to data:
  - transparency, color, reflectance, surface strength...
- “Area of interest” specification achieved:



Density-based
classification

>>> ...... >>>



Space-based
segmentation

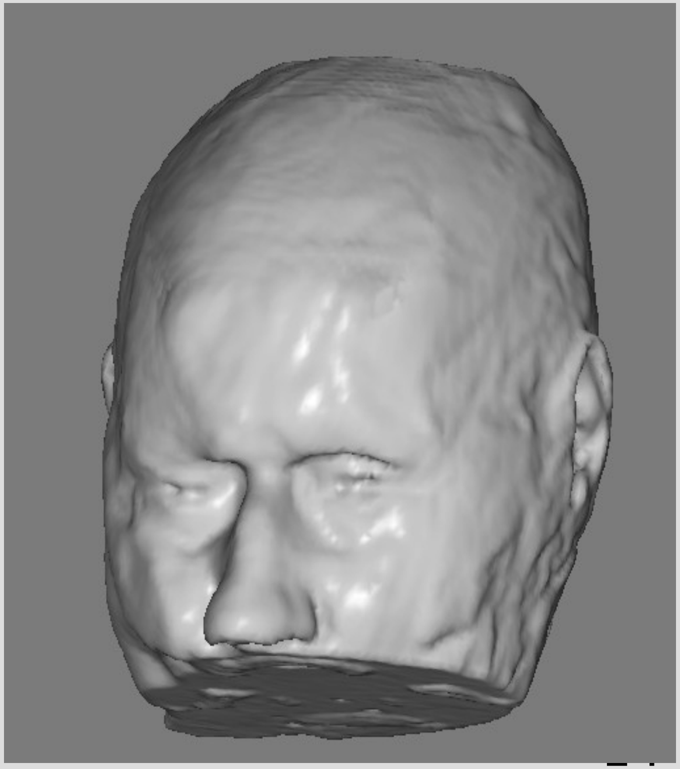
Mapping: Summary

- **Transfer function based:**
  - Color & transparency assigned to voxels
  - Semitransparent volumes
  - Display of volumes
- **Segmentation-based**
  - Unambiguous object definition
  - Color & transparency assigned to objects
  - Display of surfaces

(not used for classification of rendering techniques)

Classification of Rendering Techniques (1)

- Based on the basic rendering primitive
- Surface rendering
  - Basic primitive: 2D patches (polygons)
  - Extra data structure: a surface model
  - Decoupling of the model and the data
  - Rendered by standard CG approaches



Triangulation by the Marching Cubes technique, approx. 200000 polygons

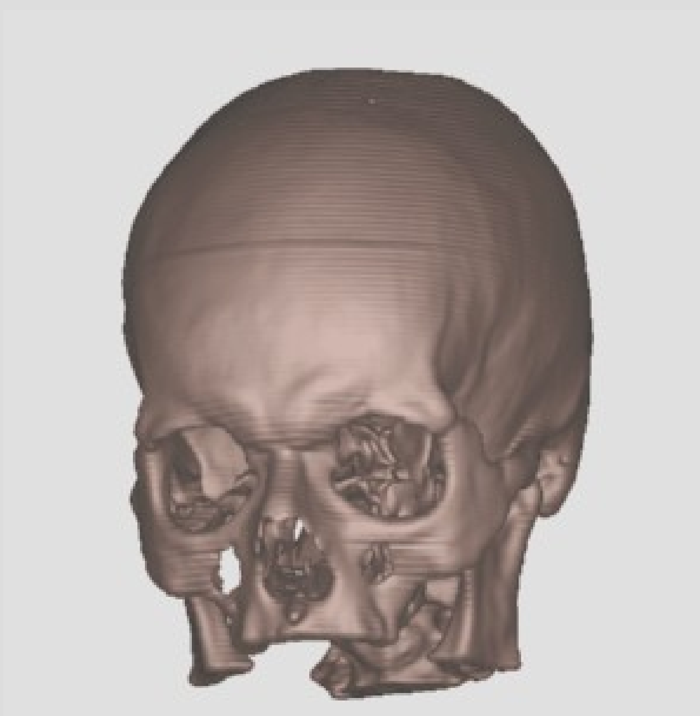
Classification of Rendering Techniques (2)

- Volume rendering
- Basic primitive: the voxel itself
- Rendering directly from volume data:
- Two flavors:
  - TF based: *Direct volume rendering* (DVR)
    - All (semitransparent) volume samples potentially contribute to the image
  - Segmentation (object) based: *Isosurfacing*
    - Only visible surfaces are displayed

DVR vs. Isosurfacing

- Rendering algorithms are similar
- Isosurfacing is a limit case of DVRs with special TF and parameter setting

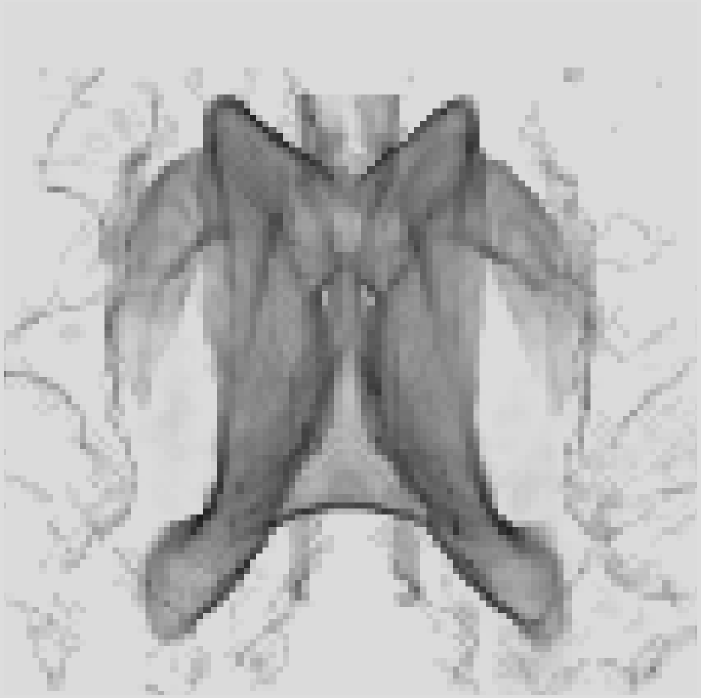
Isosurfacing



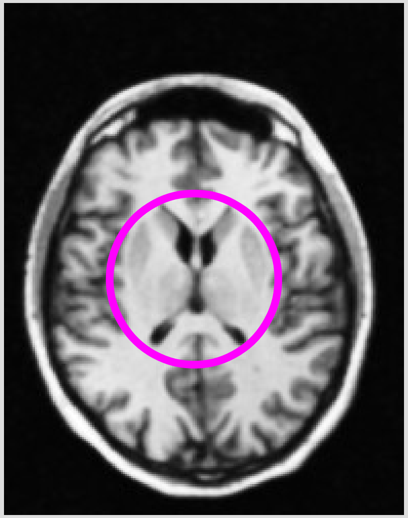
DVR

When to Prefer DVR?

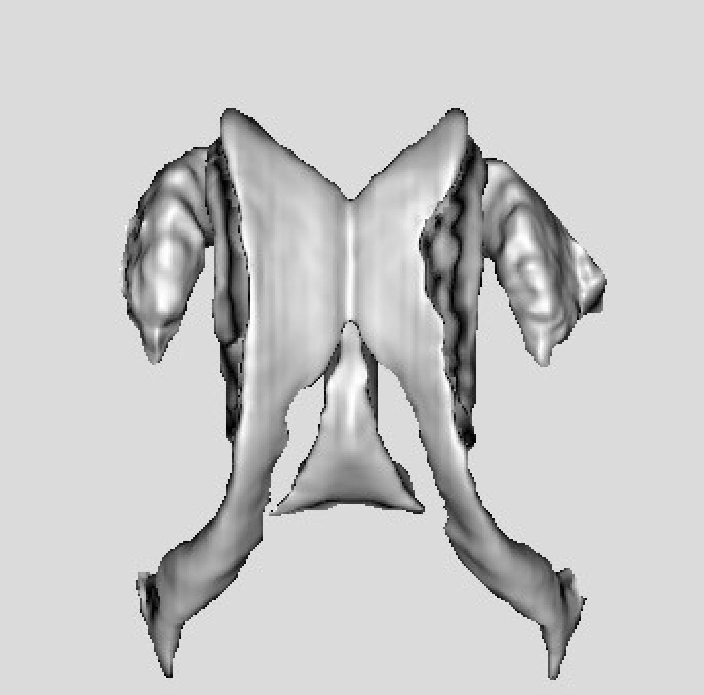
- Low data contrast, weak edges, thin objects



DVR



MRI head data: Ventricles & deep brain structures

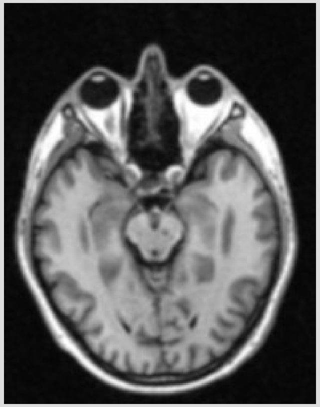


Isosurfacing

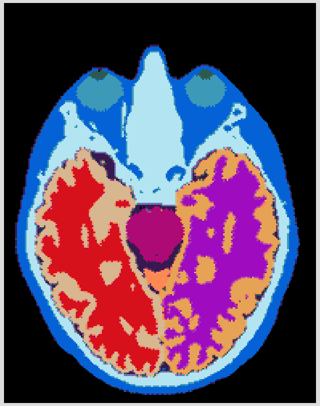
When to Prefer Isosurfacing?

- Numerous & complex objects, TFs make no sense

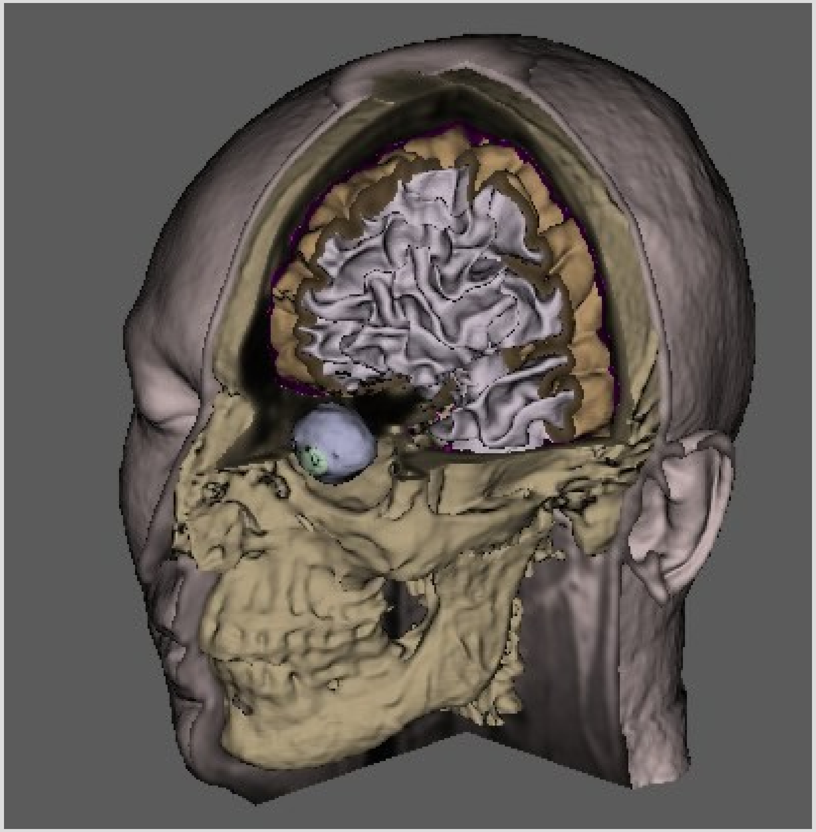
DVR



MRI data



Object labels



Isosurfacing with cutplanes

DVR Basics

- Simplified light interaction with semi-transparent material
- Light attenuation and emission along a ray

$$\frac{dI(t)}{dt} = \rho(t)I(t) - k(t)\rho(t),$$

I(t): Light intensity at the point t
ρ(t): Optical density (attenuation)
ρ(t)I(t): Light attenuation at t
k(t): Chromacity
k(t)ρ(t): Light emission rate at t

- No shadows, no reflections
- Numerical evaluation:
  - Per-segment compositing by Porter&Duff's operators
    - Front-to-back order
    - Back-to-front order

DVR Techniques (1)

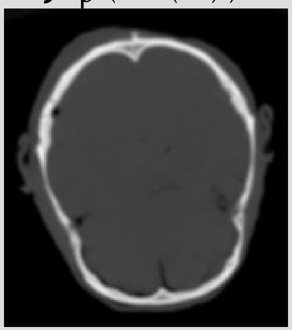
Free/adjustable parameters (mapping):

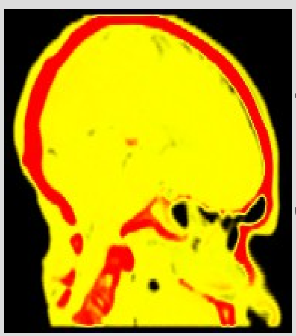
- Transfer functions: $\rho(t) = f_\rho(d(t))$

  $k(t) = f_k(d(t))$
- Edge accentuation: $\rho(t) \sim |\nabla d(t)|$
- Shading: $k(t) \sim \nabla d(t) . \vec{p} \quad f_k(d(t))$
- Depth cueing
- Others: shape, size, ...

DVR Techniques (2)

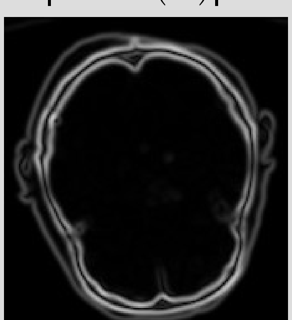
$\rho(t) = f_\rho(d(t))$



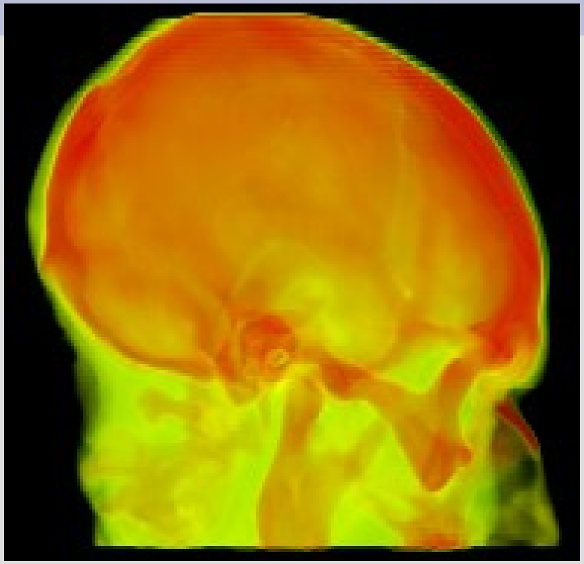


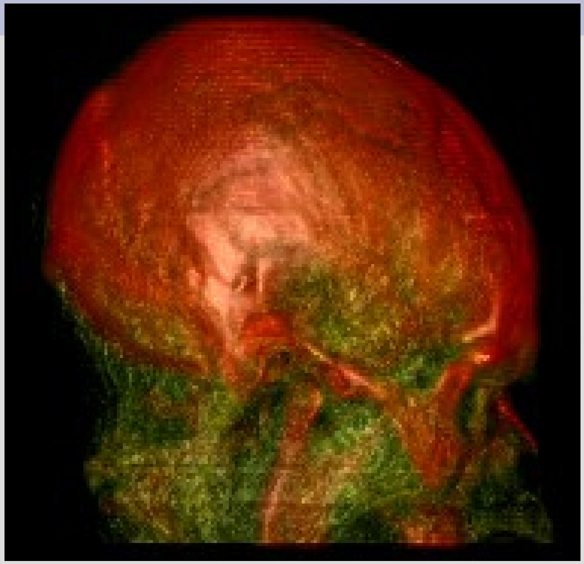
Tr. function applied

$\rho(t) \sim |\nabla d(t)|$



No edge enhancement

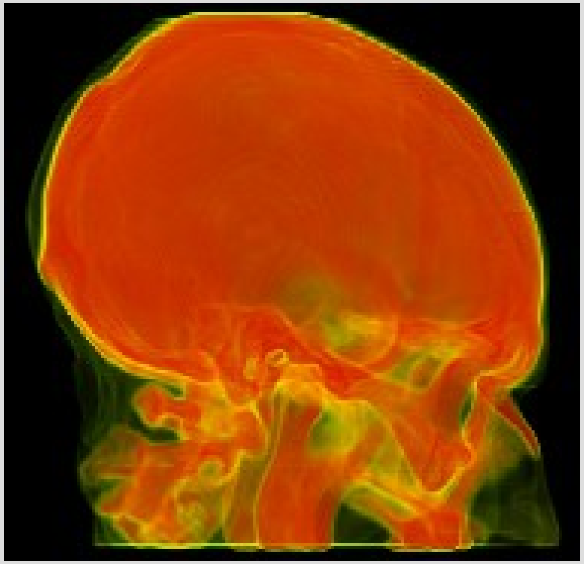


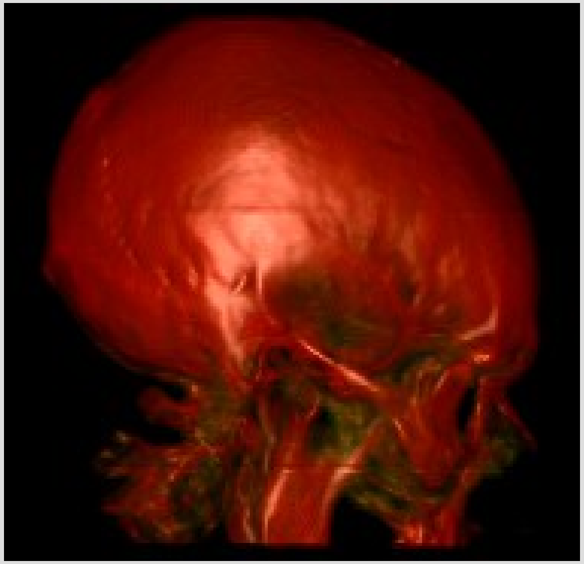


Unshaded

Shaded

Edge enhancement





Rendering by Compositing

- Interaction with matter results in absorption
  - Beer-Lambert law:

$$\frac{dI(t)}{dt} = \rho(t)I(t) - k(t)\rho(t),$$

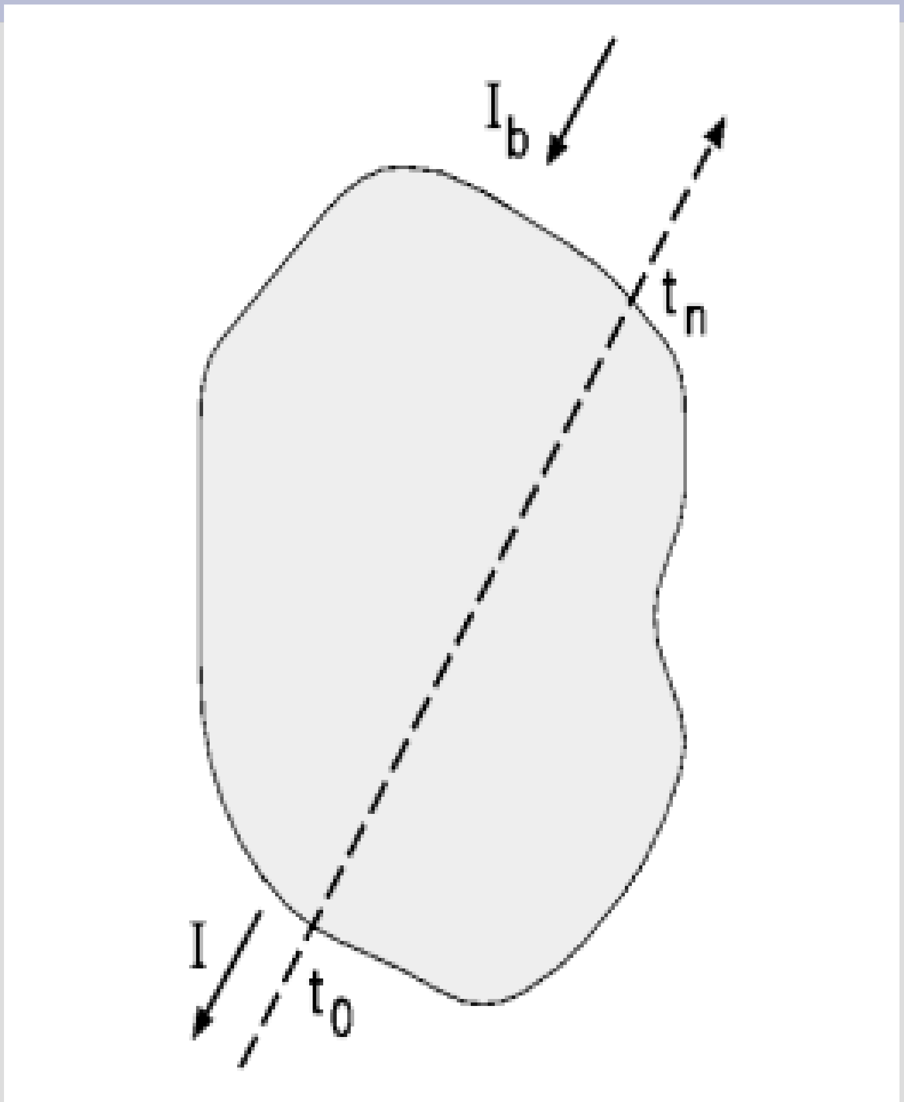
- Evaluation along a projection ray

Volume Rendering Integral

- Integral form of the Beer's law

$$I = \int_{t_0}^{t_n} k(t)\rho(t) e^{-\int_{t_0}^{t} \rho(u)\,du}\,dt + I_b e^{-\int_{t_0}^{t_n} \rho(t)\,dt}$$

transparency



Per Segment Evaluation of the VRI (1)

- **Transparency:** $e^{-\int_{t_i}^{t_{i+1}} \rho(u)\, du}$
  - **1:** fully transparent
  - **0:** fully opaque
- **Opacity**:
  - amount of stopped light
  - opacity = 1 – transparency
- **Segment opacity**: $\alpha_i = 1 - e^{-\int_{t_i}^{t_{i+1}} \rho(u)\, du}$
  - The amount of light stopped along a segment

Per Segment Evaluation of the VRI (2)

- Front-to-back compositing

$$I_m = I_{m-1} + (1 - \beta_{m-1})C_m$$
$$\beta_m = \beta_{m-1} + (1 - \beta_{m-1})\alpha_m$$

'*under*' operator

- Back-to-front compositing

$$I_m = C_m + (1 - \alpha_m)I_{m-1}$$

'*over*' operator

I – accumulated color
C- sample color
β- accumulated opacity
α- sample opacity

Approximations

- Approximation of *ρ* and *k* by a constant:

$$\alpha_i = 1 - e^{-\rho_i \Delta t_i}$$

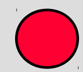
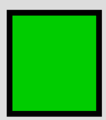
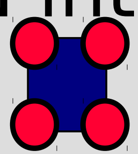
$$C_i = k_i \alpha_i$$

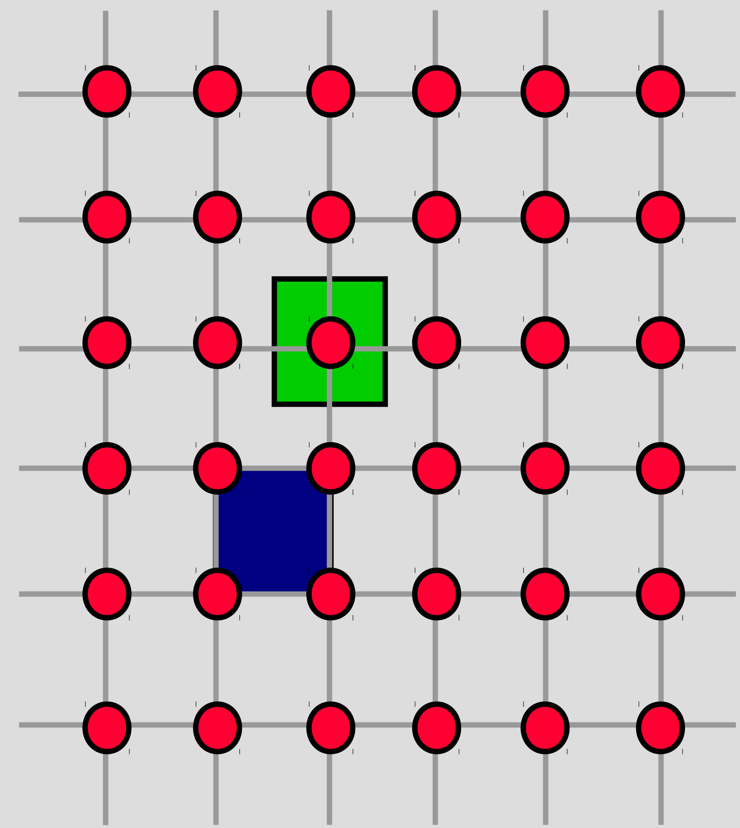
- Chromacity premultiplied by opacity
- Scaling required with real data

DVR Algorithms

- Object-order algorithms (splatting)
  - Projection of samples from volume to image
  - Compositing in image plane
- Image-order algorithms
  - Ray casting based
  - Sequence of samples along the ray
  - Compositing along the ray

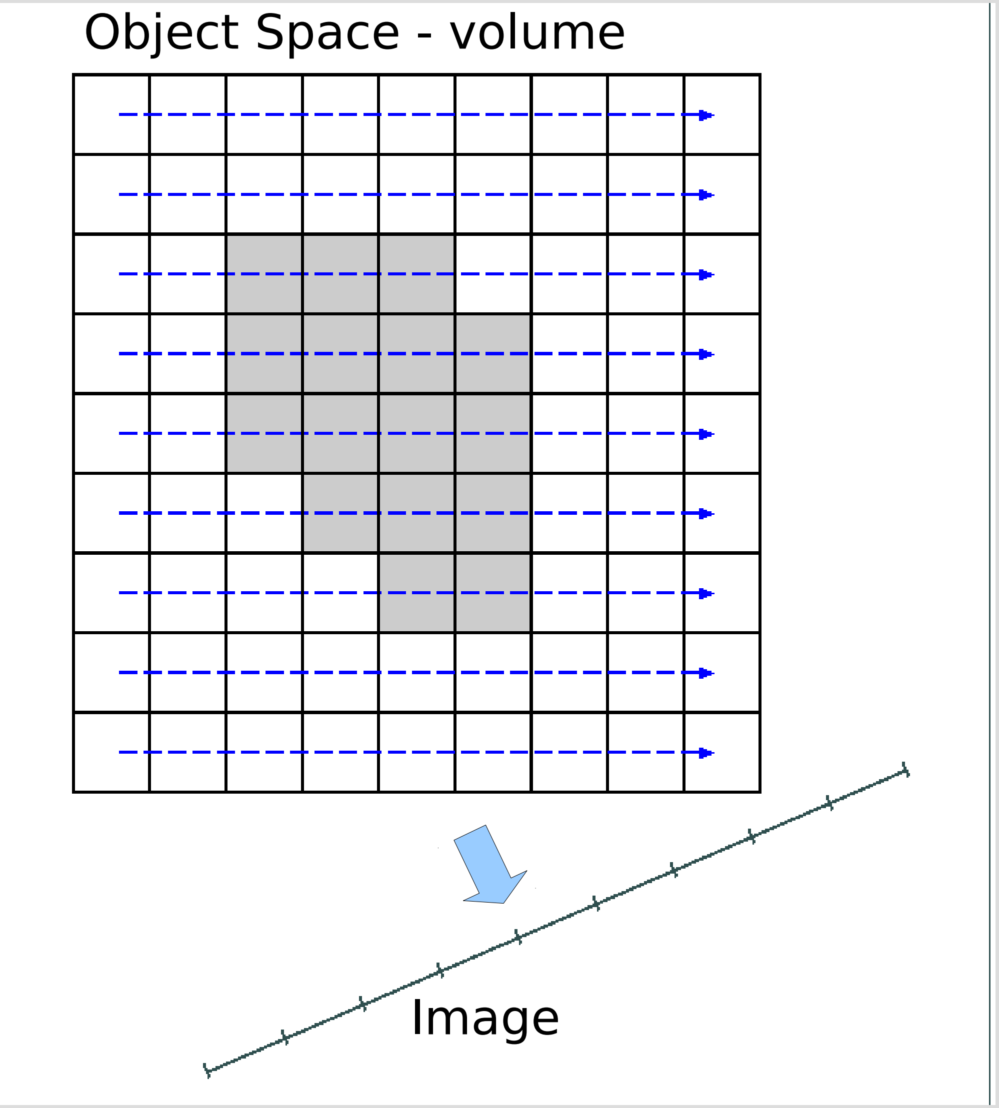
3D Discrete Space

- 3D grid point (sample): 
- P = [x, y, z], x, y, z ∈ Z
- Value at sample P: density
- Voxel: 
  - Voronoi neighborhood of P
  - NN interpolation
- Cell: 
  - 8 samples
  - higher order interpolation



Object-Order VR: Splatting

- Samples/voxels are projected (splatted) onto the viewing plane
- Back-to-front of front-to-back order
- One sample projects onto several pixels



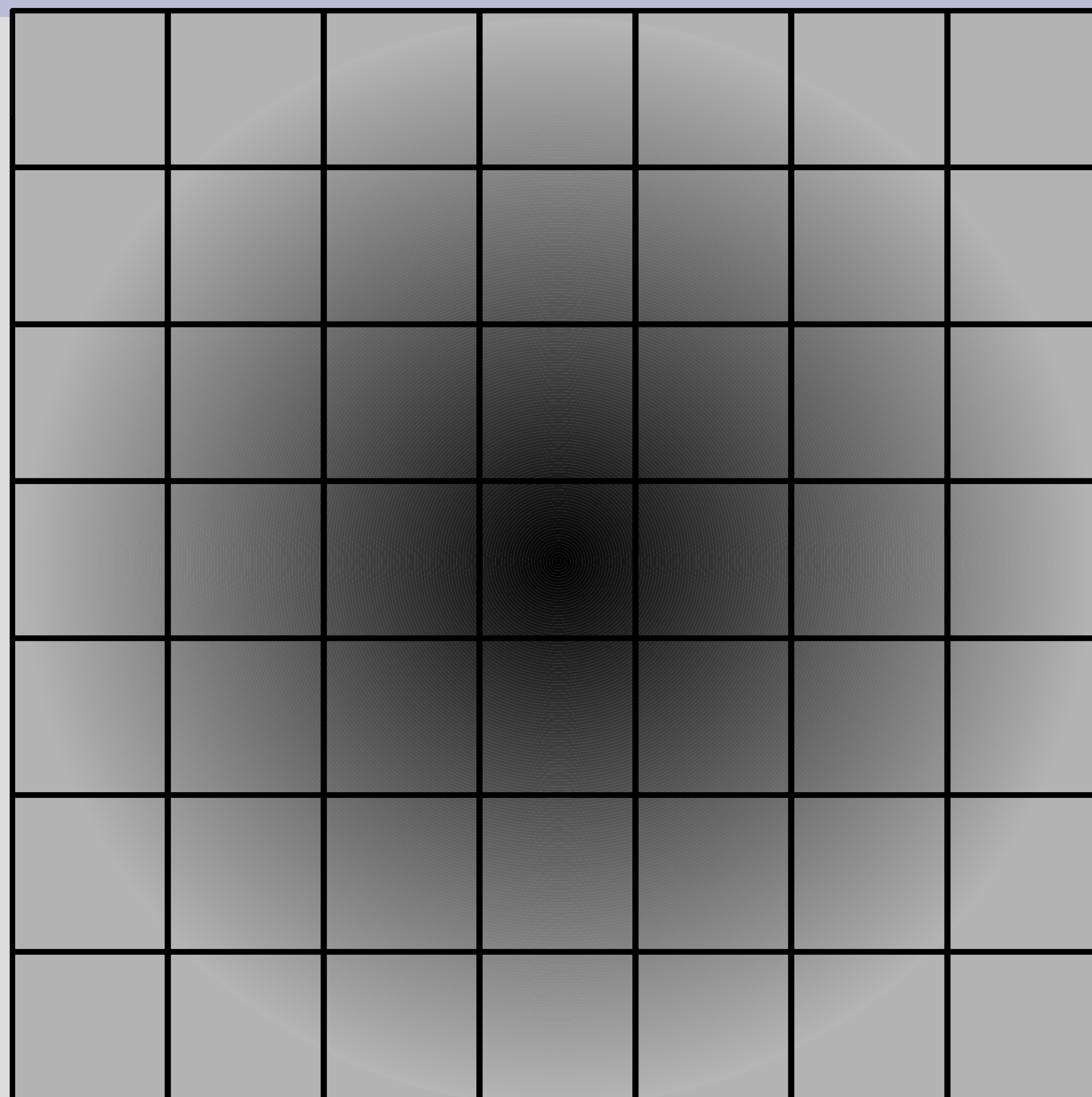
Splatting Footprint (Westover)

- A sample is represented as a cloud of particles
- High resolution footprint table: computed only once

$$\rho(s)\, h_v(x-x_s, y-y_s, z-z_s) \rightarrow \rho(s)\, footprint(x-x_s, y-y_s)$$

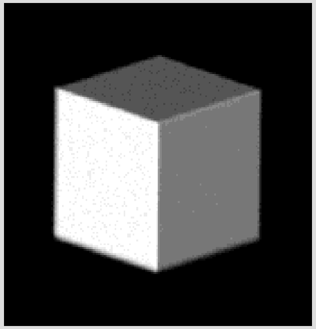
- BTF, FTB compositing

Footprint table

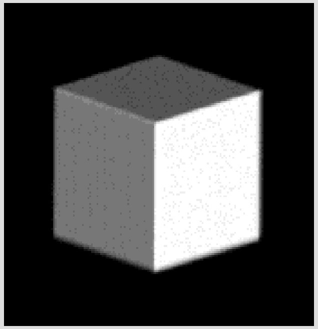


Splatting artifacts (1)

- Traversal order depends on viewing angles
- The most parallel scanline to image is chosen
- Popping artifacts:

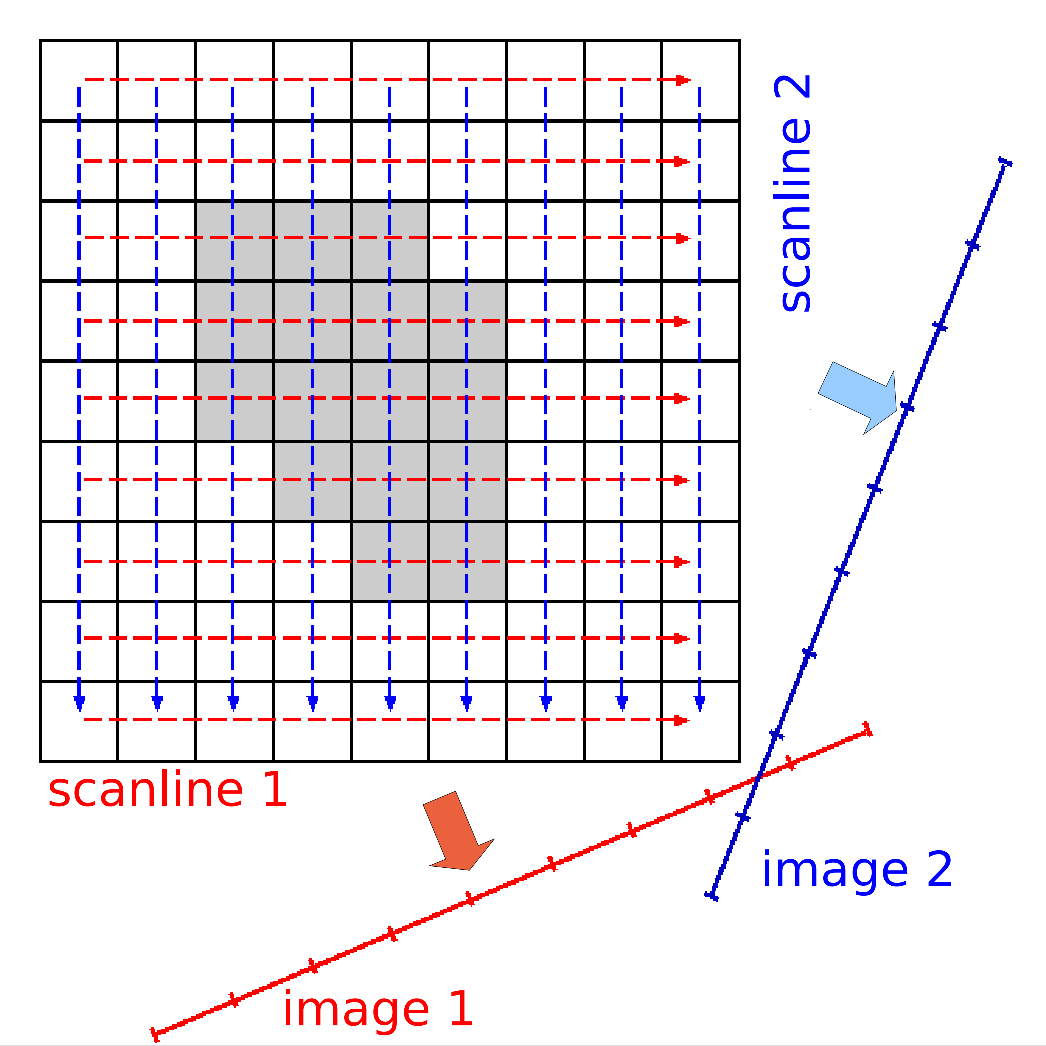


rot=45°



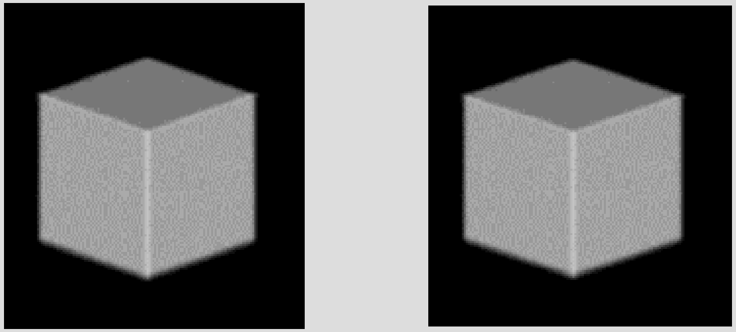
rot=45.1°

[Mueller 1998]



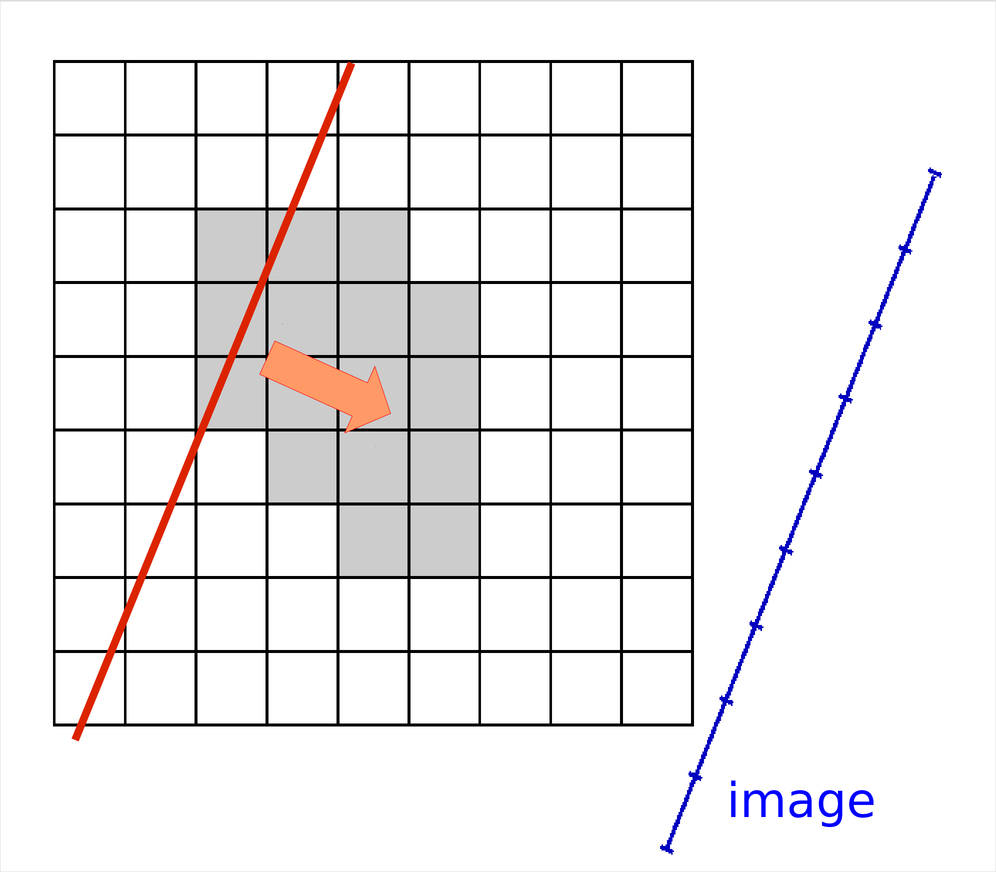
Splatting artifacts (2)

- Image aligned sheet-buffer
- No popping



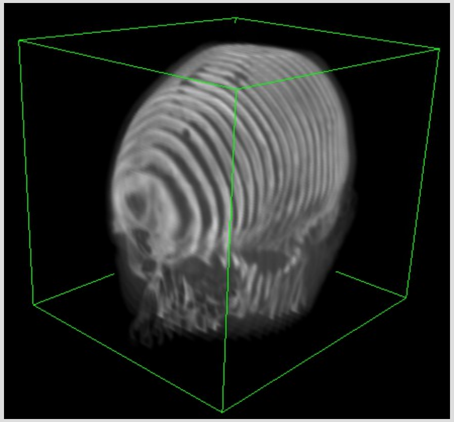
rot=45° rot=45.1°

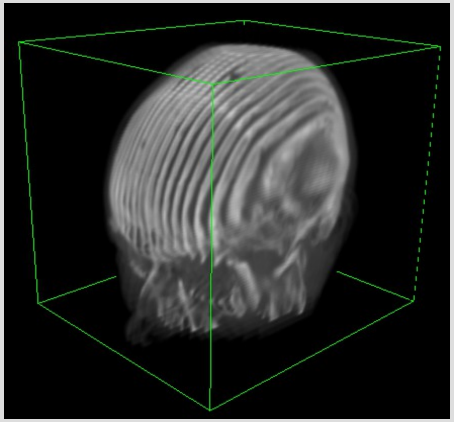
[Mueller 1998]



HW Acceleration by Texture Mapping

- Do the costly part by hardware
- 2D Textures
  - compositing only
  - volume aligned slices
  - three copies of the volume required
- 3D Textures
  - interpolation & compositing
  - image aligned slices
- Shading possible in fragment programs





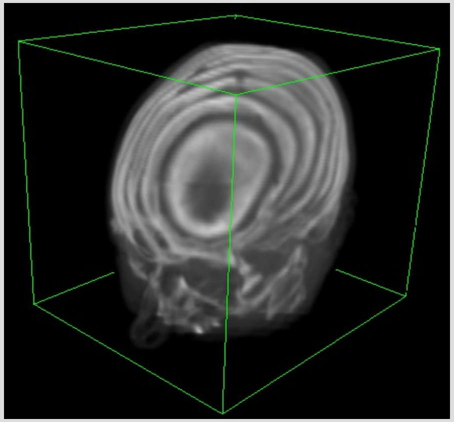


Image Order VR by Ray-Casting (1)

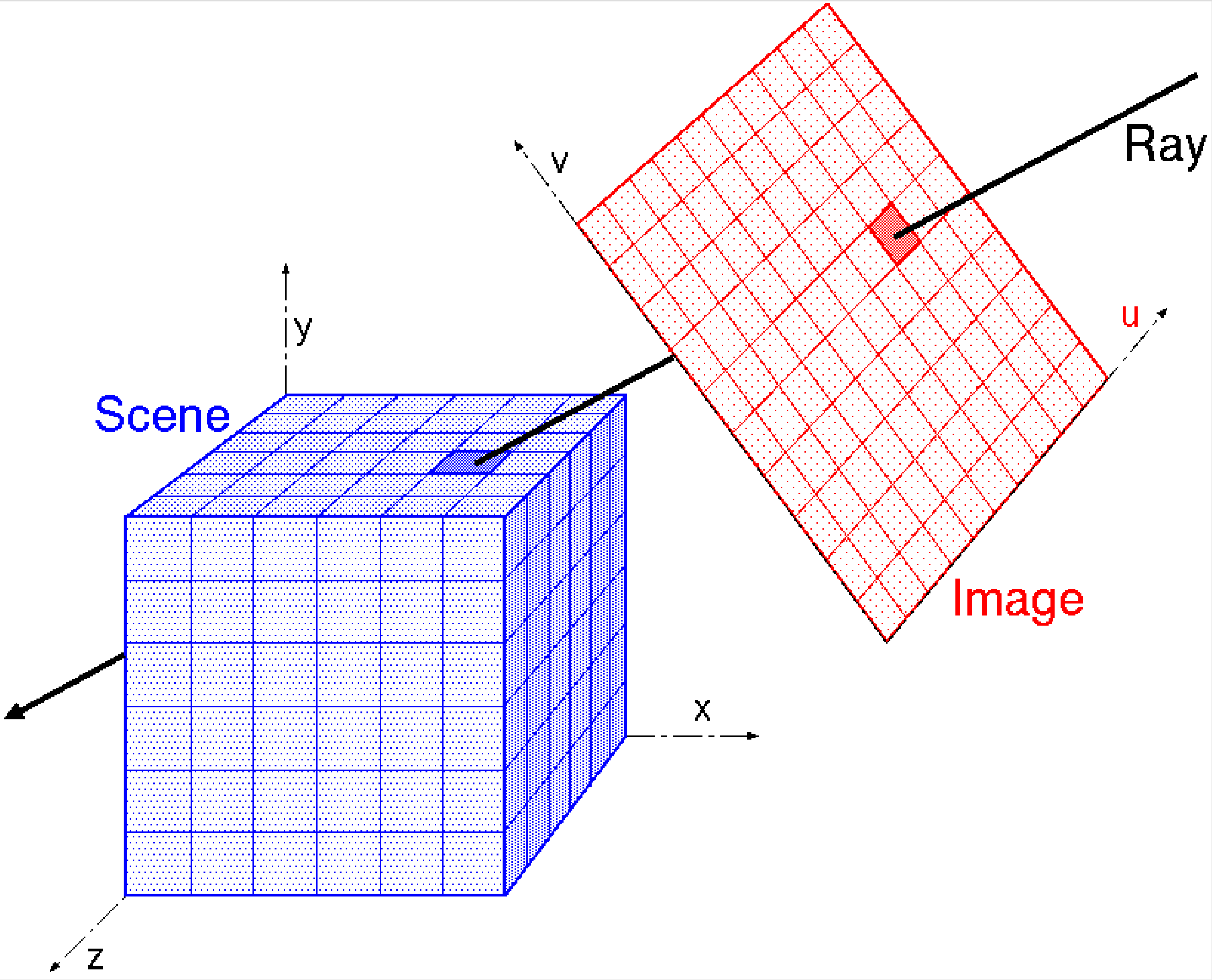
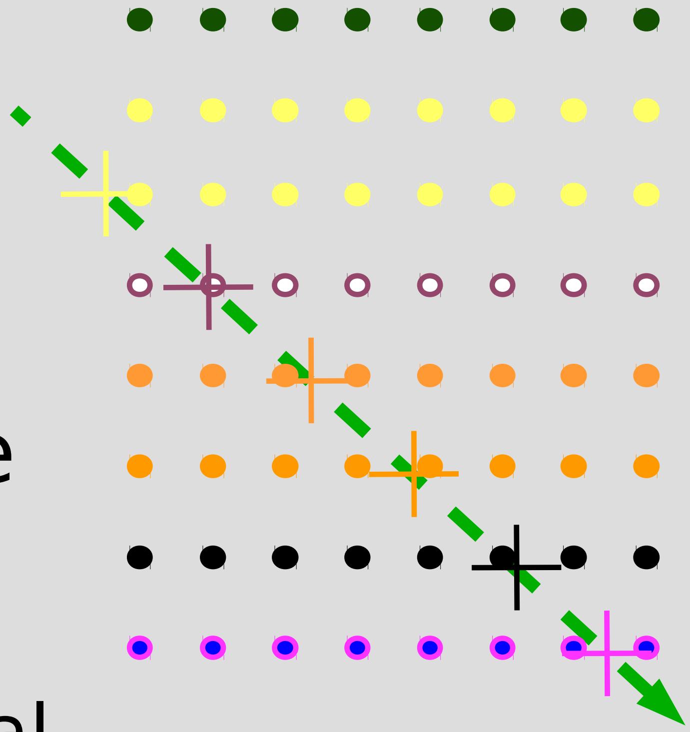


Image Order VR by Ray-Casting (2)

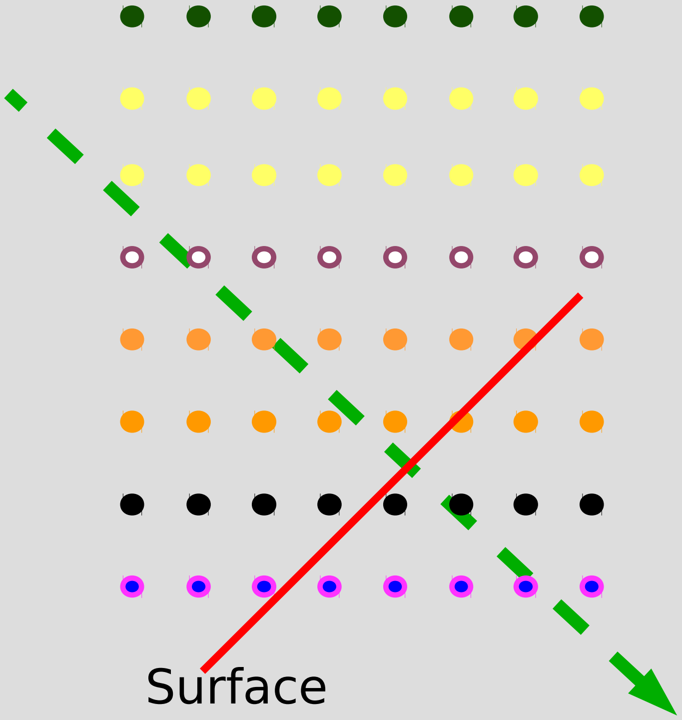
- Shoot rays from each pixel
- Define a sequence of samples
- Accumulate color and opacity along each ray

- CPU and GPU implementations possible
- Acceleration required:
  – Adaptive sampling, empty space skipping, hierarchical subdivision, early termination



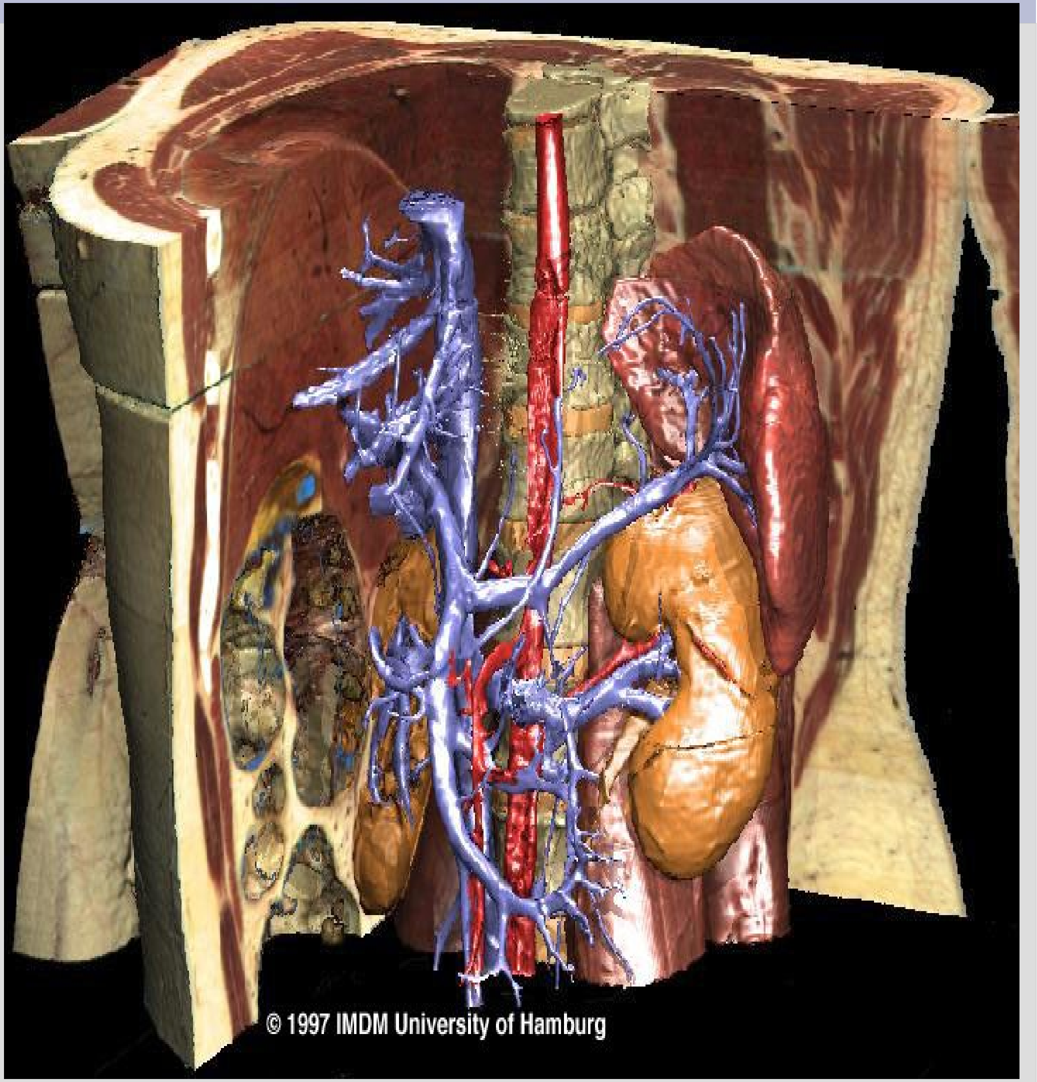
Isosurfacing Basics

- Interpolation domain:
  - Original densities
  - Segmentation labels
- Algorithms:
  - (First hit) ray tracing
  - Ray/surface intersections by numerical root finding
- CPU and GPU implementations possible
- Acceleration required



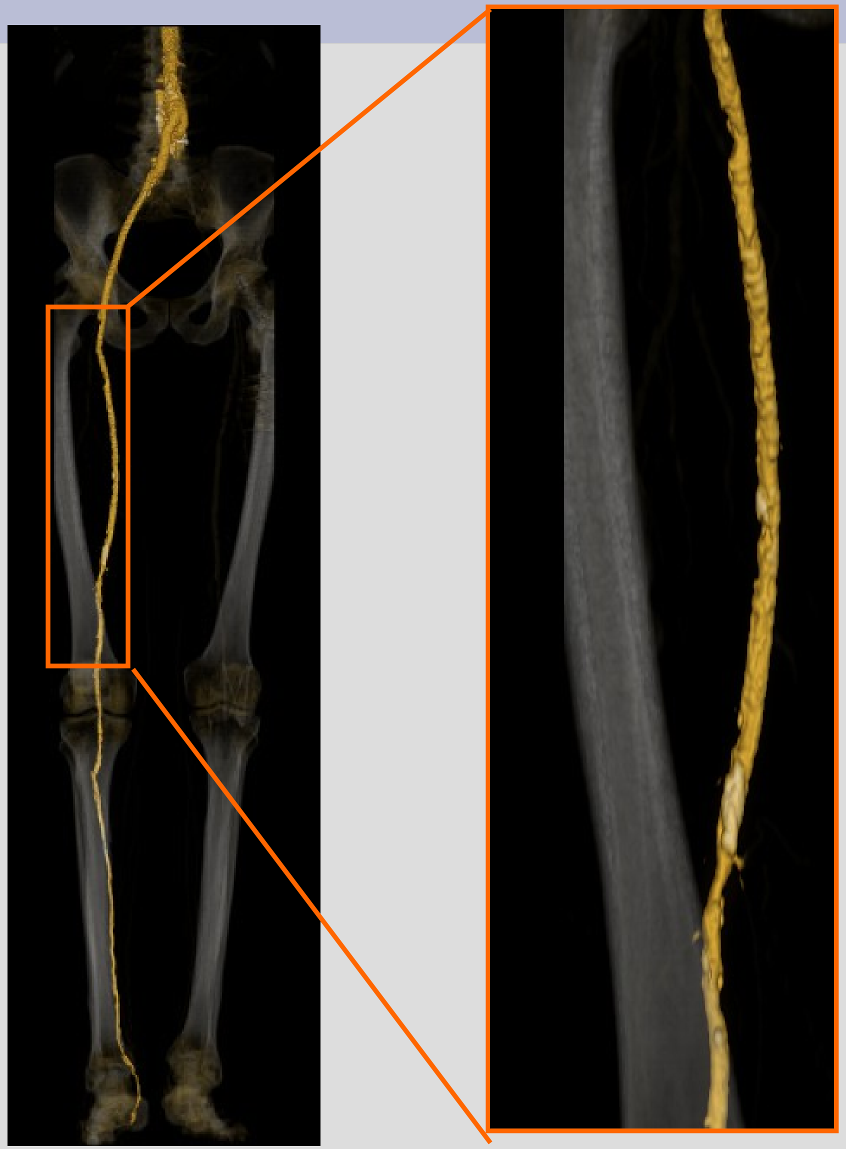
Ray Casting Advantages

- Simplicity and flexibility
- Combination of techniques possible:
  - DVR, isosurfacing, MIP, CPR, cutplanes
  - per object definition of techniques and parameters



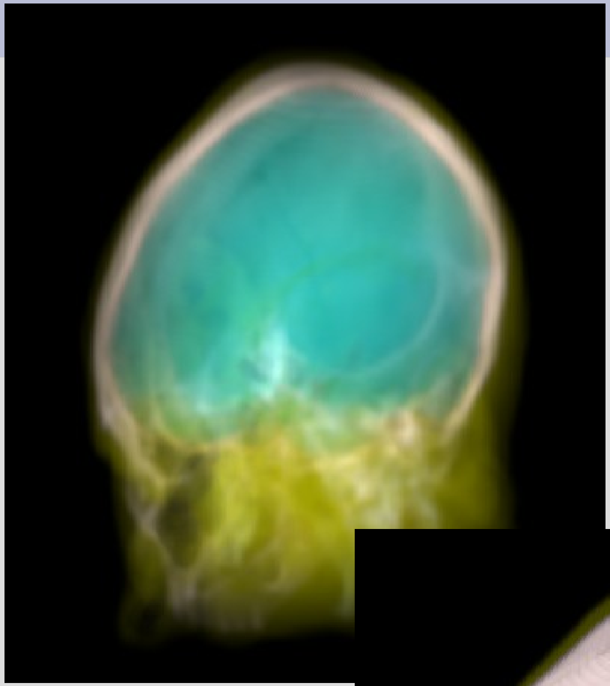
Ray Casting Advantages

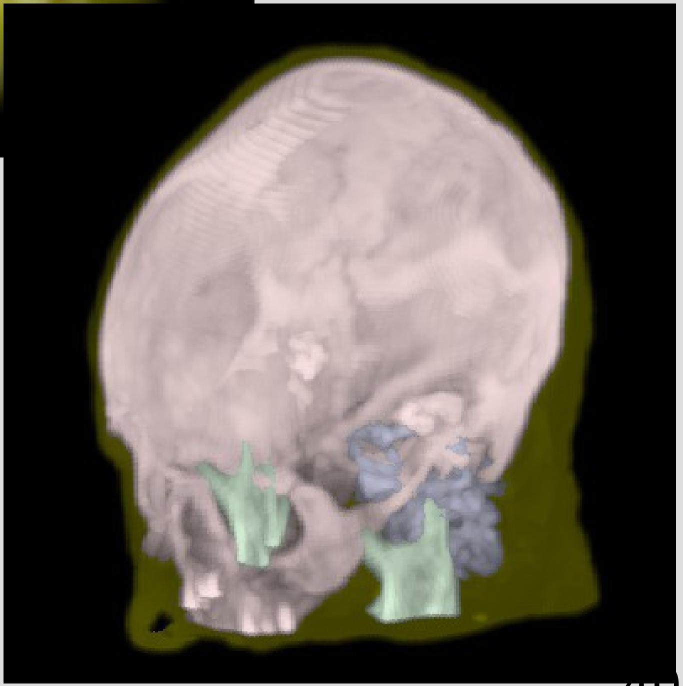
- Simplicity and flexibility
- Combination of techniques possible:
  - DVR, isosurfacing, MIP, CPR, cutplanes
  - per object definition of techniques and parameters



Ray Casting Advantages

- Simplicity and flexibility
- Combination of techniques possible:
  - DVR, isosurfacing, MIP, CPR, cutplanes
  - per object definition of techniques and parameters





Conclusion

- We have plentiful volume rendering techniques
- We do not have enough memory
- The computers are not fast enough
- The doctors still prefer slice-by-slice viewing (but getting better recently...)

Thank you!

milos.sramek@oeaw.ac.at
www.viskom.oeaw.ac.at/~milos