Realtime Computer Graphics on GPUs

Scientific Visualization

Jan Kolomazník

Department of Software and Computer Science Education
Faculty of Mathematics and Physics
Charles University in Prague

Computer Graphics Charles University

Volumetric Data

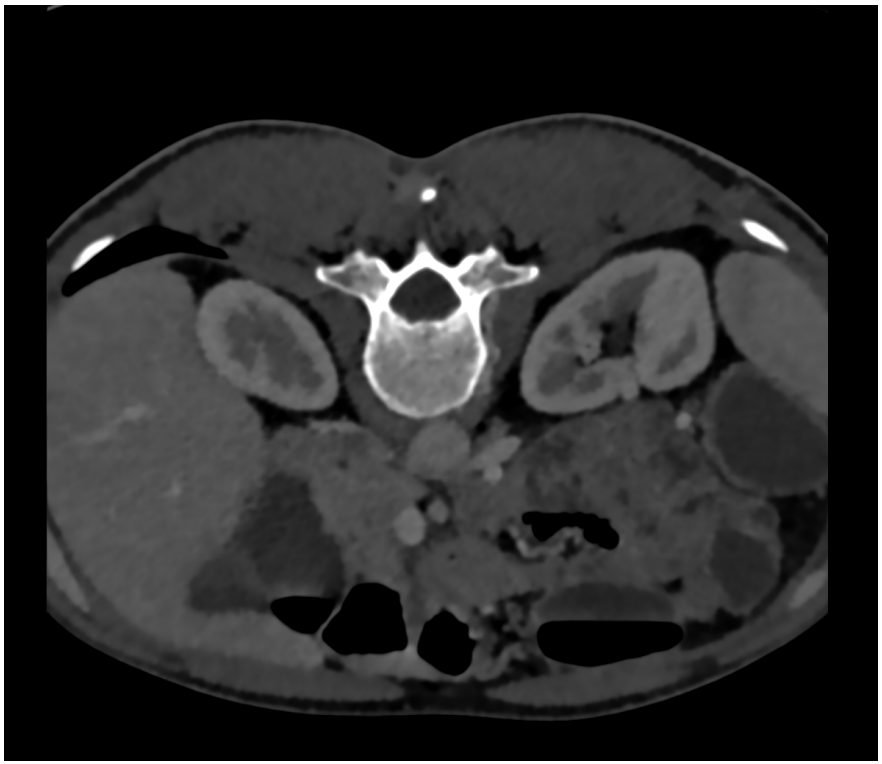
Data Source

- Computed tomography (CT)
- Magnetic resonance (MRI)
- Confocal laser scanning microscopy
- Ultrasonic imaging
- Cryo-electron tomography
- Positron emission tomography (PET)
- ...

Data Representation

- Regular, 3-dimensional grid of samples (voxels)
  - Scalar values – density, absorption coefficients, event counting
  - Vectors
  - Color
- 3D texture:
  - Trilinear filtering
  - Easy *slicing* in general direction
- 2D texture:
  - Set of textures
  - Texture atlas
  - Manual filtering in Z-direction

2D Slices



Direct Volume Rendering

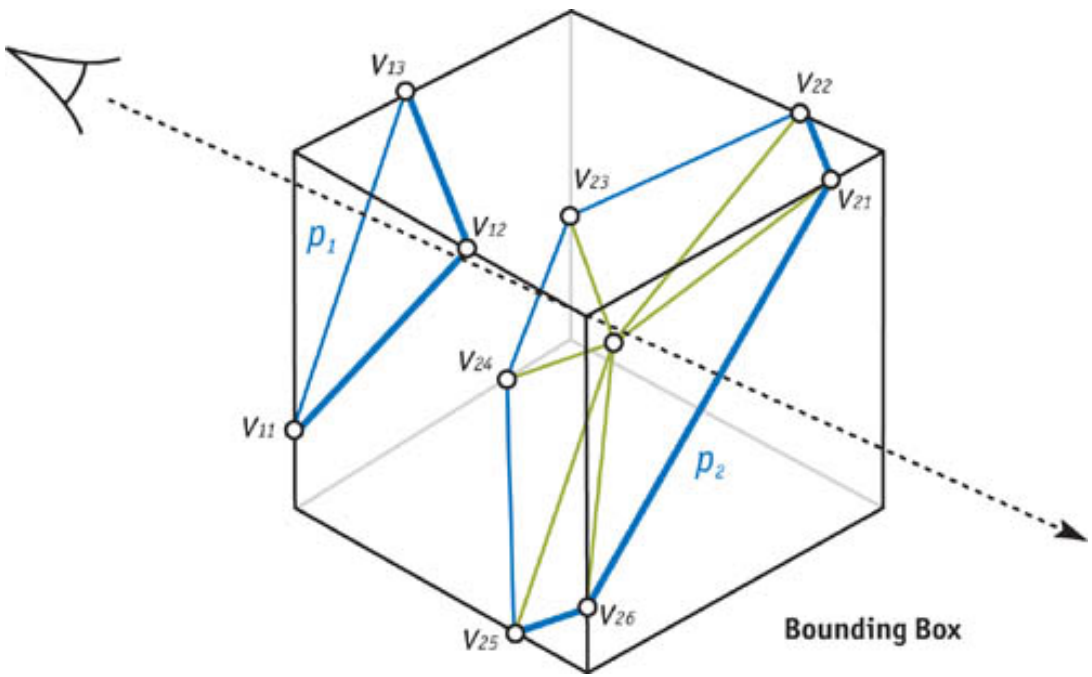
Volume Rendering Integral

$$I(D) = I_0 e^{-\int_{s_0}^{D} \kappa(t)dt} + \int_{s_0}^{D} q(s) e^{-\int_{s}^{D} \kappa(t)dt} ds$$

- entry point s_0
- exit point D (camera position)
- emission at a point q
- I_0 initial intensity s_0 (light emmitance of the background),
- κ is absorption coefficient.

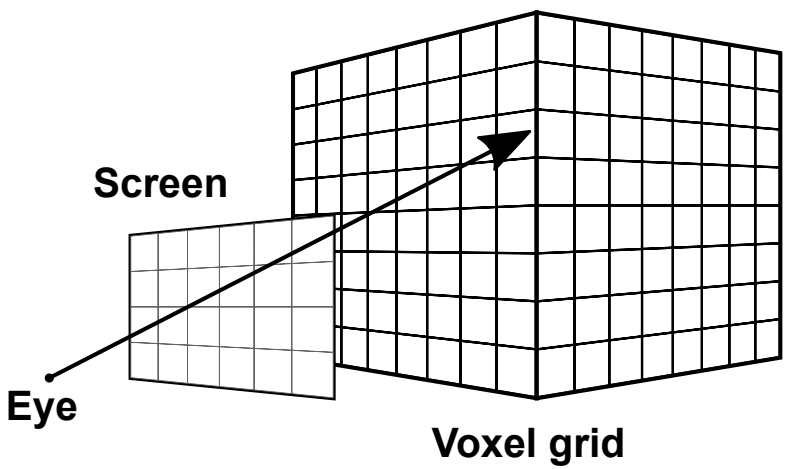
VIEWPORT ALIGNED SLICES

- Generate proxy geometry
  - Viewport aligned slices (billboards)
  - Limited by volume bounding box – limit fragment count
  - Convex – easy to triangulate
- Enable framebuffer blending
  - Color attachment with float precision



RAY-CASTING

- Generate rays from camera through each pixel
  - Fragments generated by rendering bounding volume
- Discrete samples along the ray
- Numerical computation of the rendering integral



Volume Compositing Schemes

- In Direct Volume Rendering, *compositing* accumulates color and opacity along the viewing ray.
- Two main compositing orders:
  - **Front-to-back**: processes samples from the eye toward the volume.
  - **Back-to-front**: processes samples from deep in the volume toward the eye.

Compositing Equations

Color premultiplied by alpha

Front-to-Back Compositing:

$$\begin{aligned} C_{dst} &\leftarrow C_{dst} + (1 - \alpha_{dst})C_{src} \\ \alpha_{dst} &\leftarrow \alpha_{dst} + (1 - \alpha_{dst})\alpha_{src} \end{aligned}$$

- Supports *early ray termination* when $\alpha_{dst} \approx 1$.

Back-to-Front Compositing (painter's algorithm):

$$C_{dst} \leftarrow (1 - \alpha_{src})C_{dst} + C_{src}$$

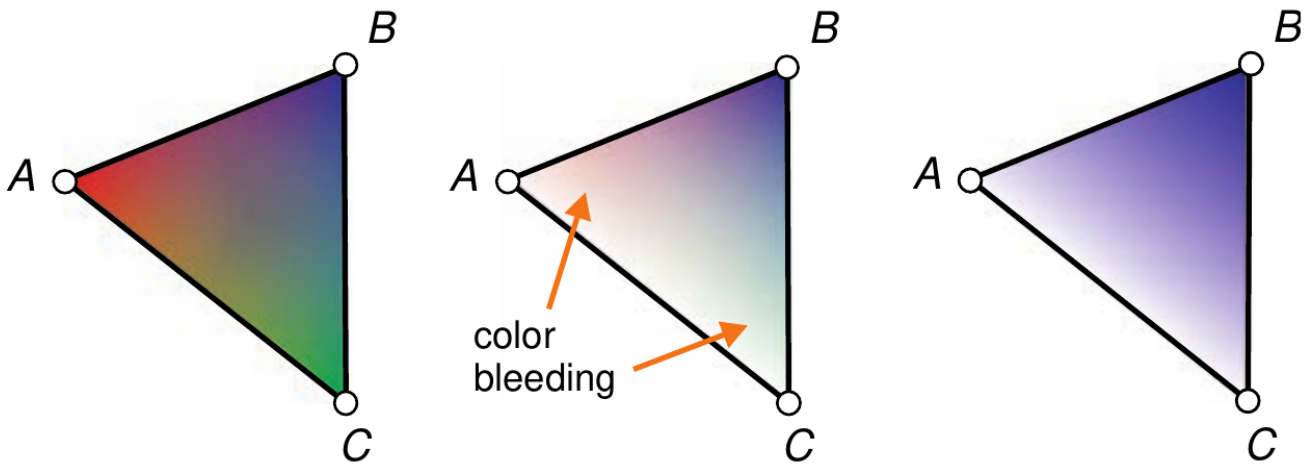
- Doesn't allow early termination.

Opacity from Absorption (based on distance):

$$\alpha_i = 1 - e^{-\kappa_i \Delta s}$$

COLOR BLEEDING

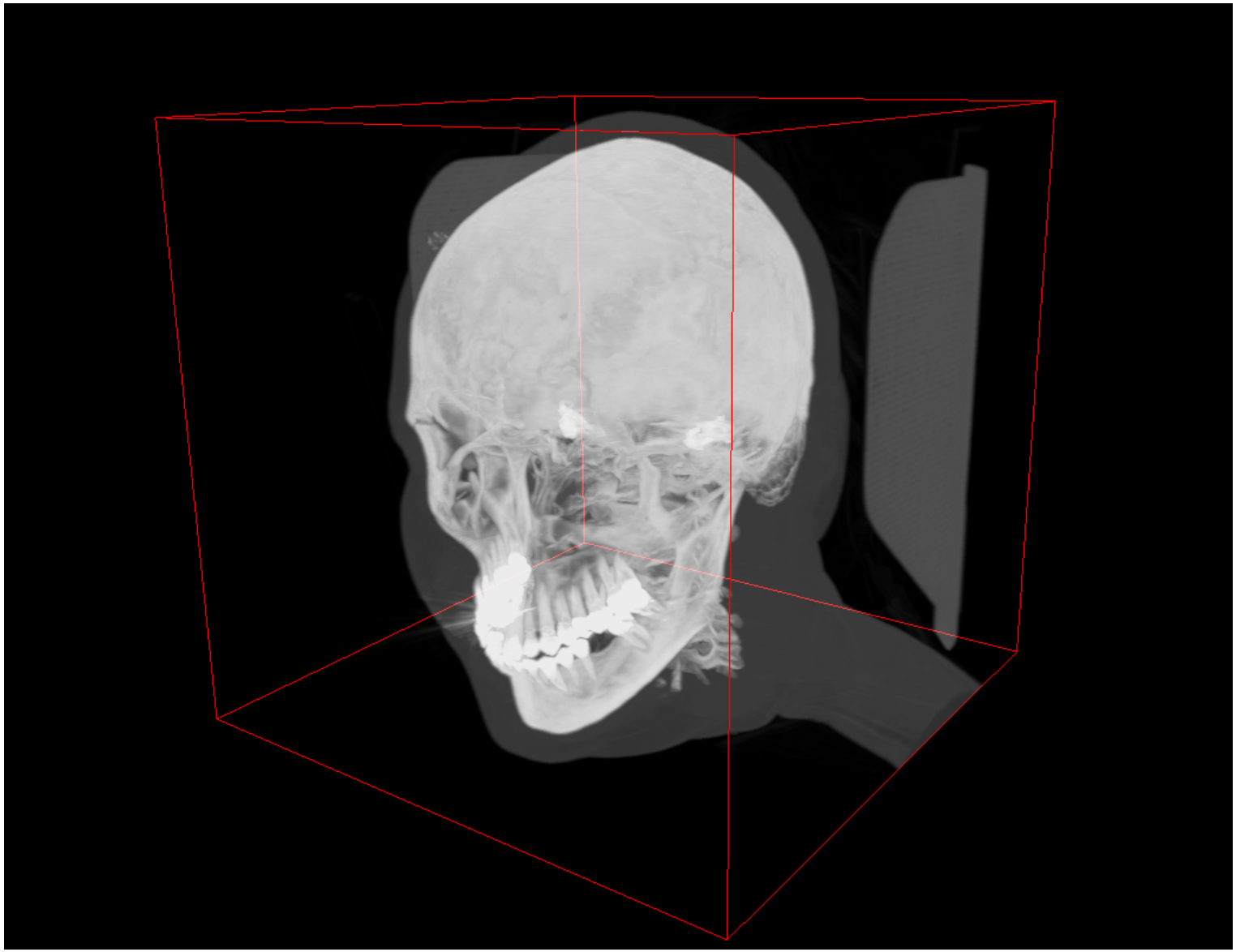
Interpolation with/without premultiplied alpha



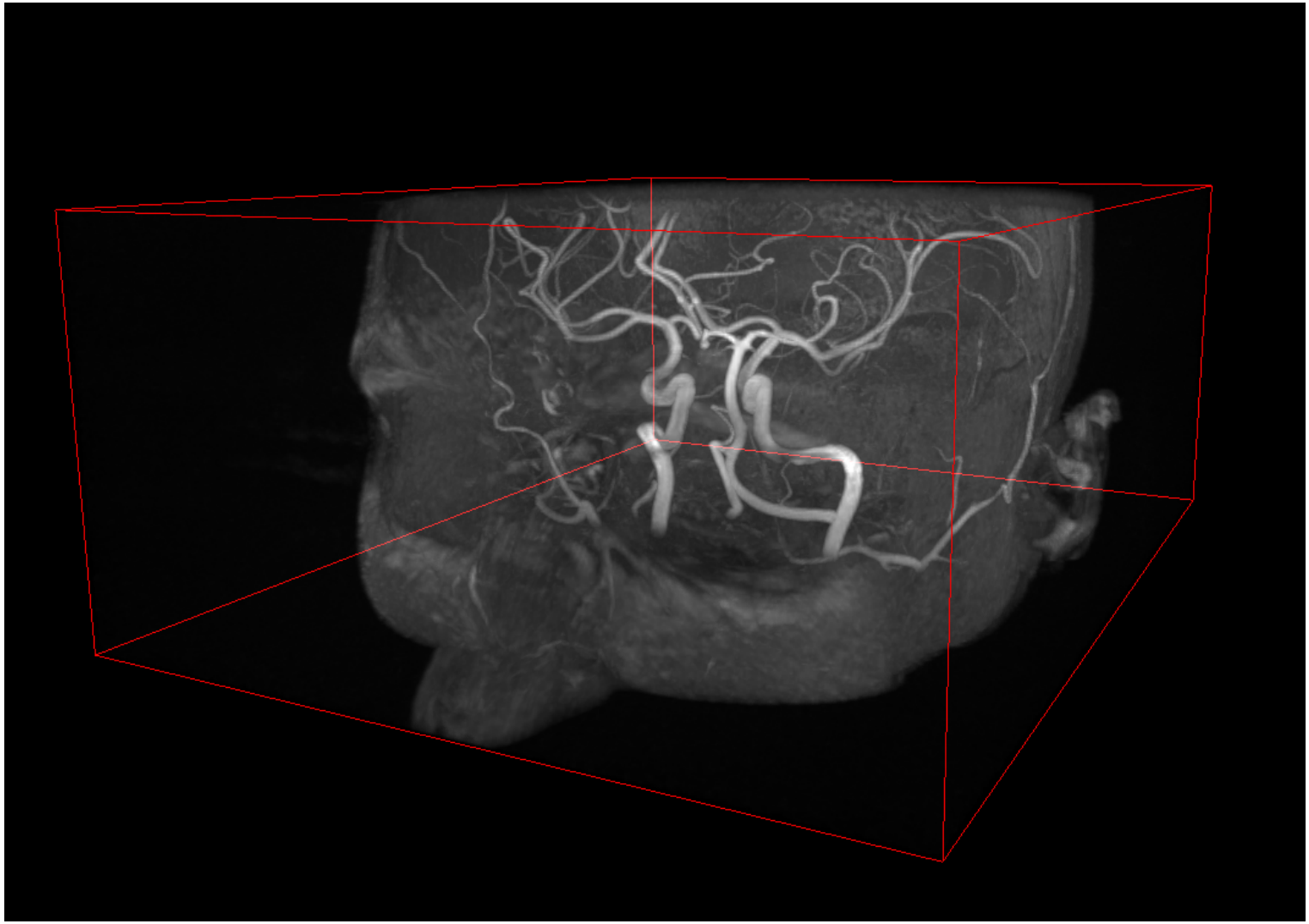
Maximum Intensity Projection

- Uses maximum value found along the ray.
- Bad sense of depth.

Maximum Intensity Projection

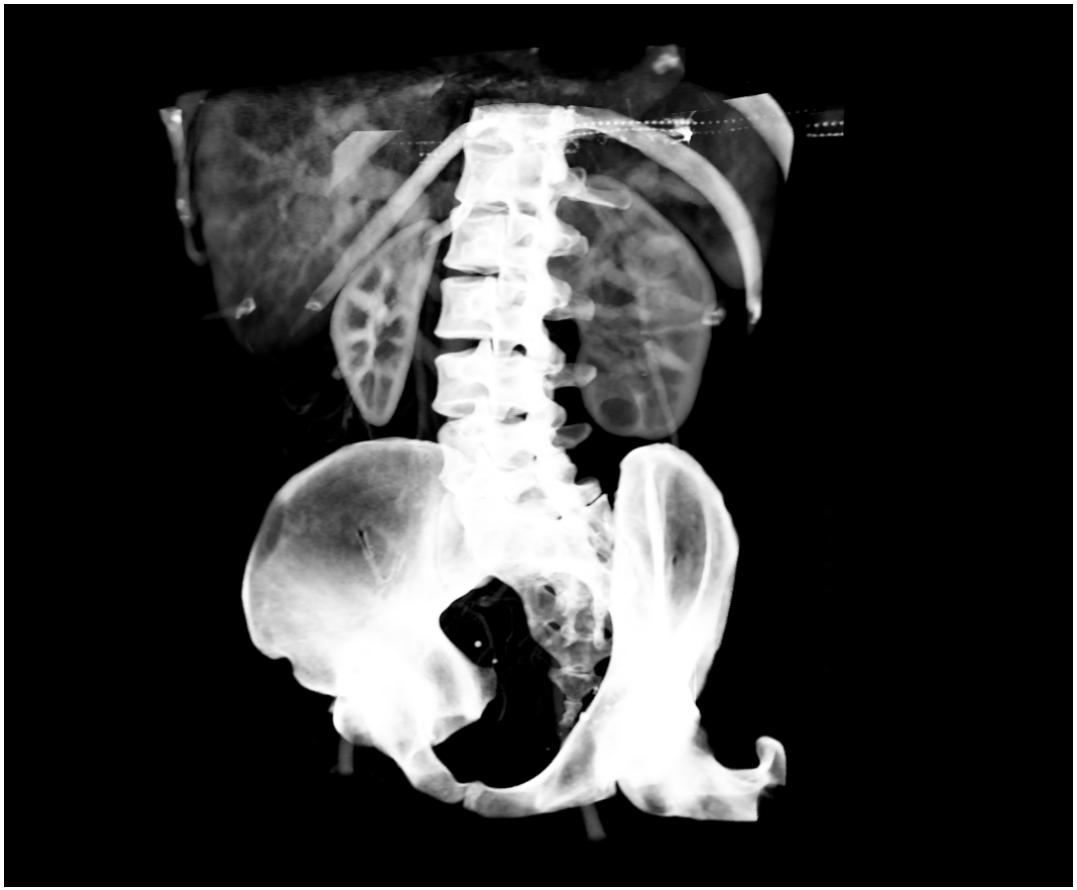


Maximum Intensity Projection



Direct Volume Rendering

- Use values stored in voxels (color, density)
- Simple contrast/lightness adjustments



Combined Geometry Rendering

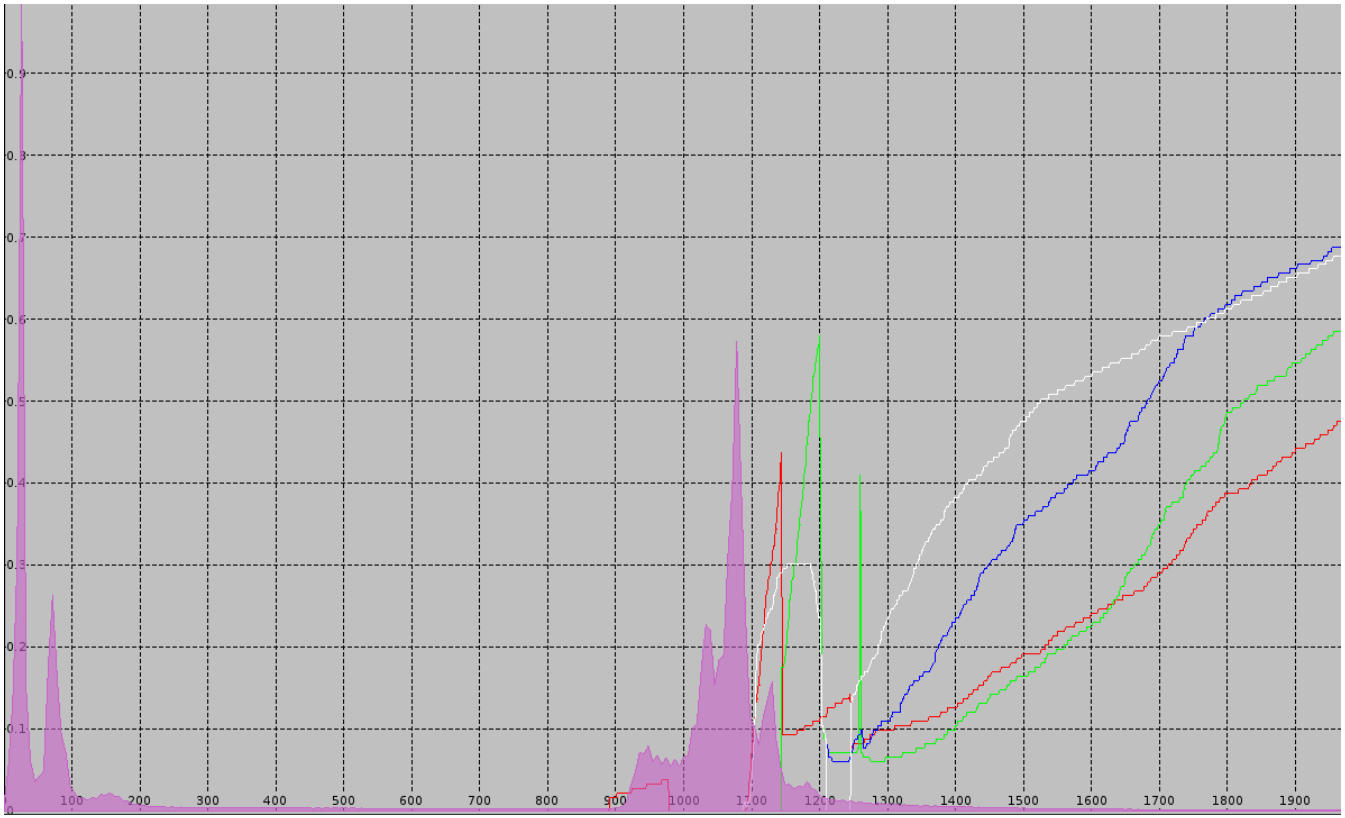
- Opaque geometry
  - Rendered before volume
  - Rays terminated by value in z-buffer
- Transparent geometry
  - Checking for geometry/ray intersections during ray traversal
  - Color computed together with volume sampling

Transfer Functions

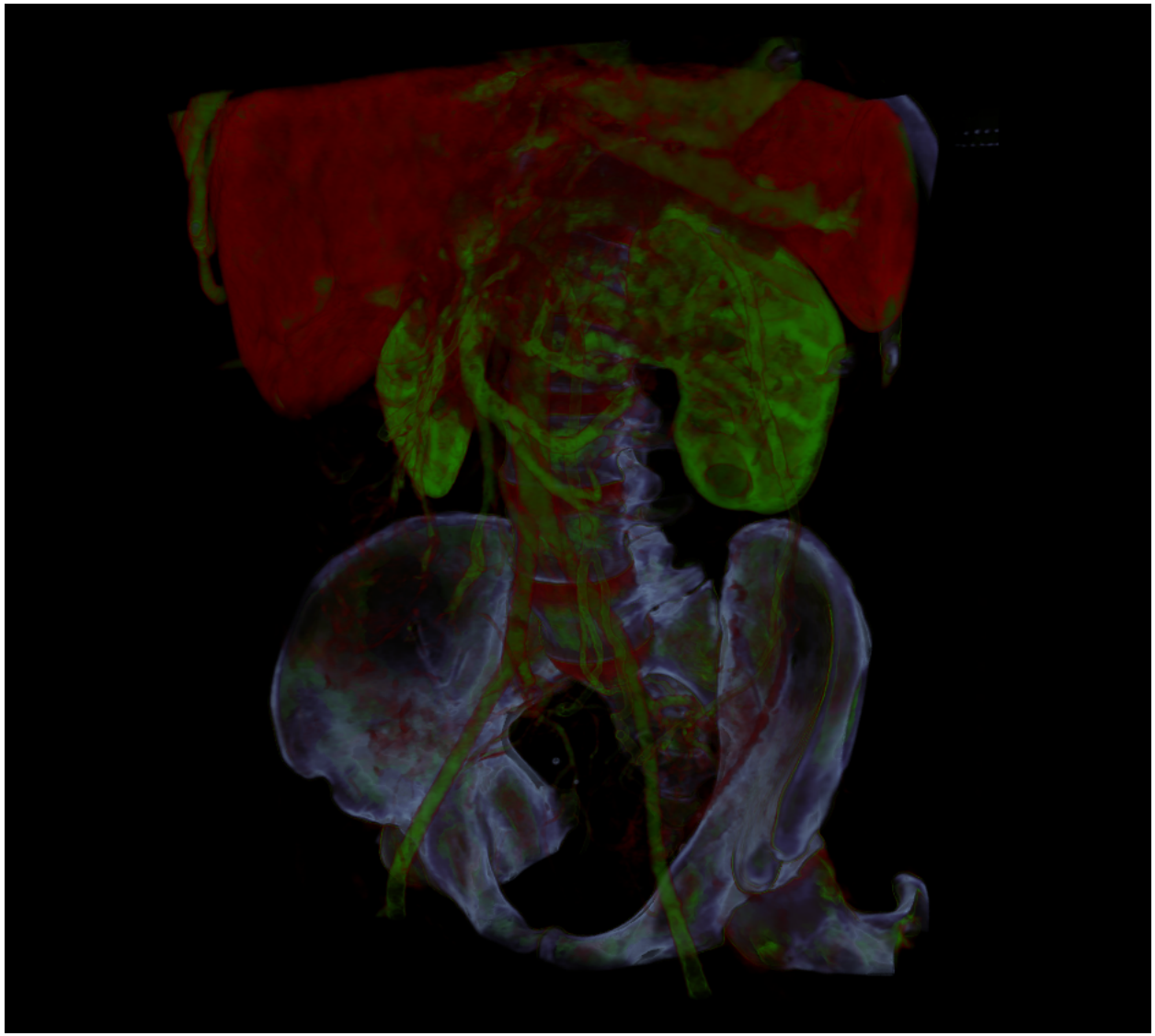
1D Transfer Functions

- Runtime fuzzy classification
- Transfer function $g(v) : R \rightarrow R^4$
- Ray sample: $g(f(\mathbf{x}))$
- Maps scalar value to RGBA color.
- Implementation:
  - 1D RGBA texture with interpolation
  - Final sample color – access TF texture

Histogram + 1D TF



1D Transfer Functions



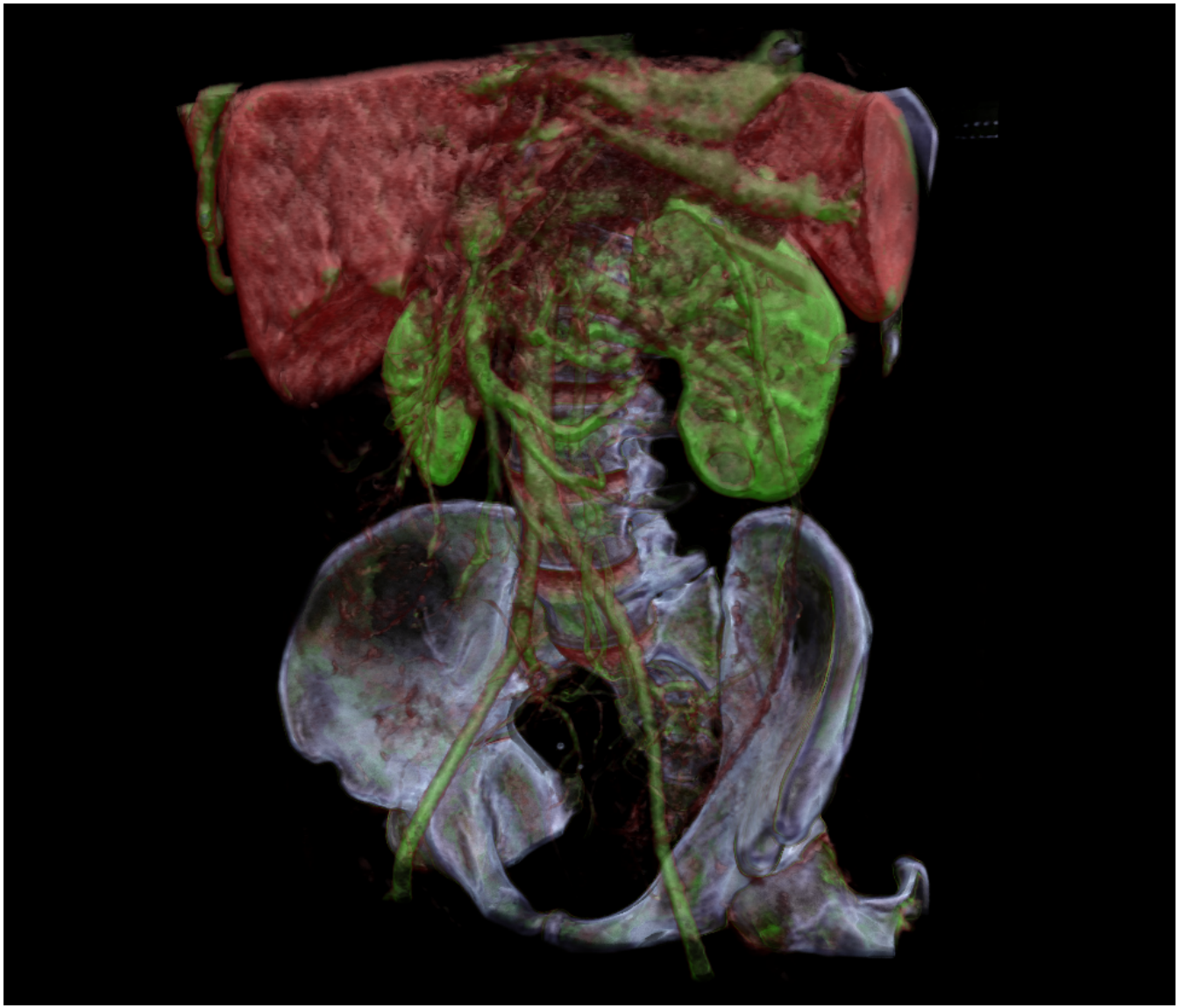
GRADIENT

- $\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \frac{\partial f(\mathbf{x})}{\partial x_3} \right)$
- Direction of the greatest rate of increase.
- Magnitude is the slope of the graph.
- Directional derivative: $\frac{\partial f(\mathbf{x})}{\partial \mathbf{v}} = \nabla f(\mathbf{x}) \cdot \mathbf{v}$
  - Can be computed on the fly
  - Symmetric differences

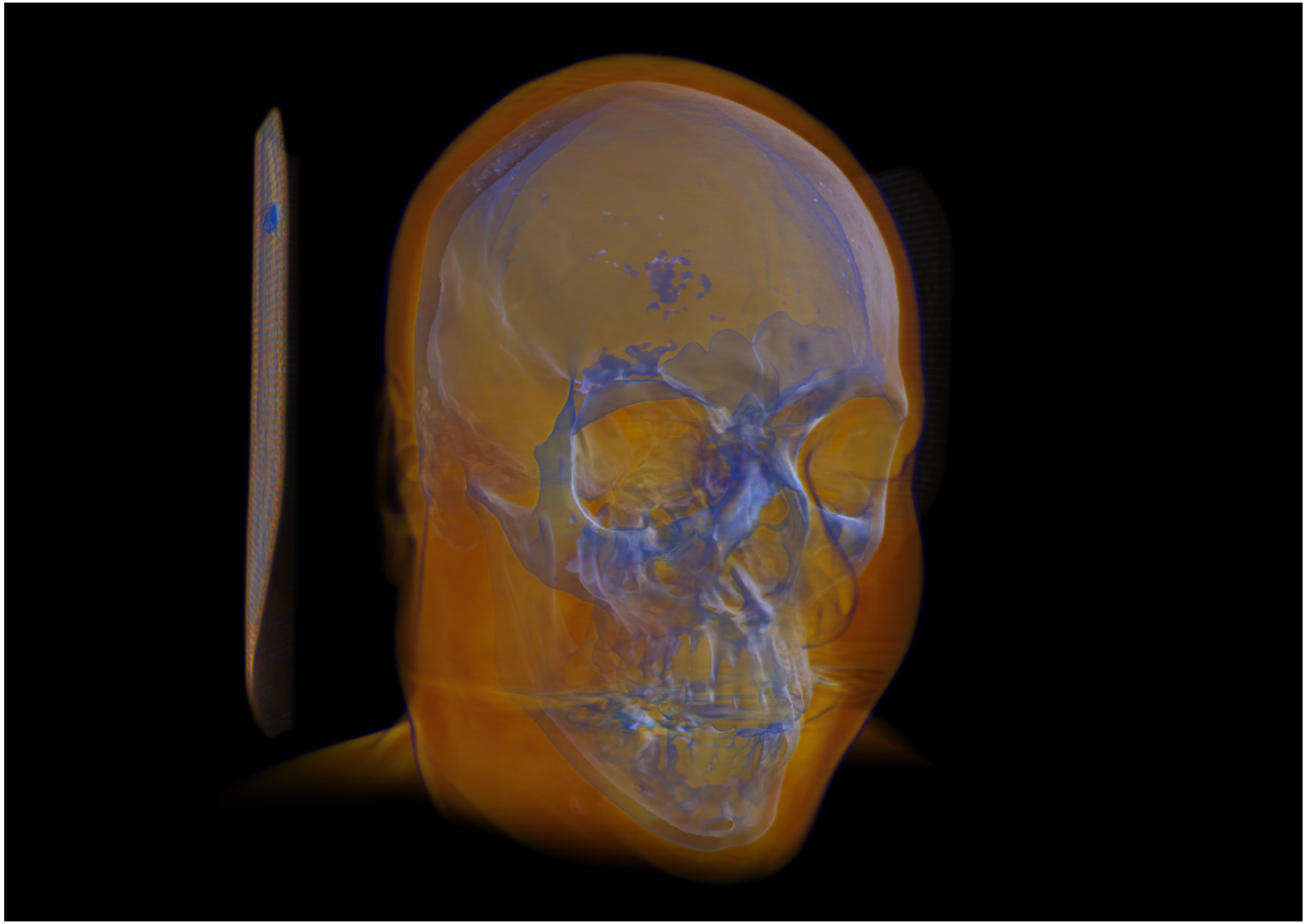
1D Transfer Functions + Light

- Better surface shape perception.
- Compute shading for opaque regions (α channel over some threshold)
- Normalized gradient as surface normal.

1D Transfer Functions + Light

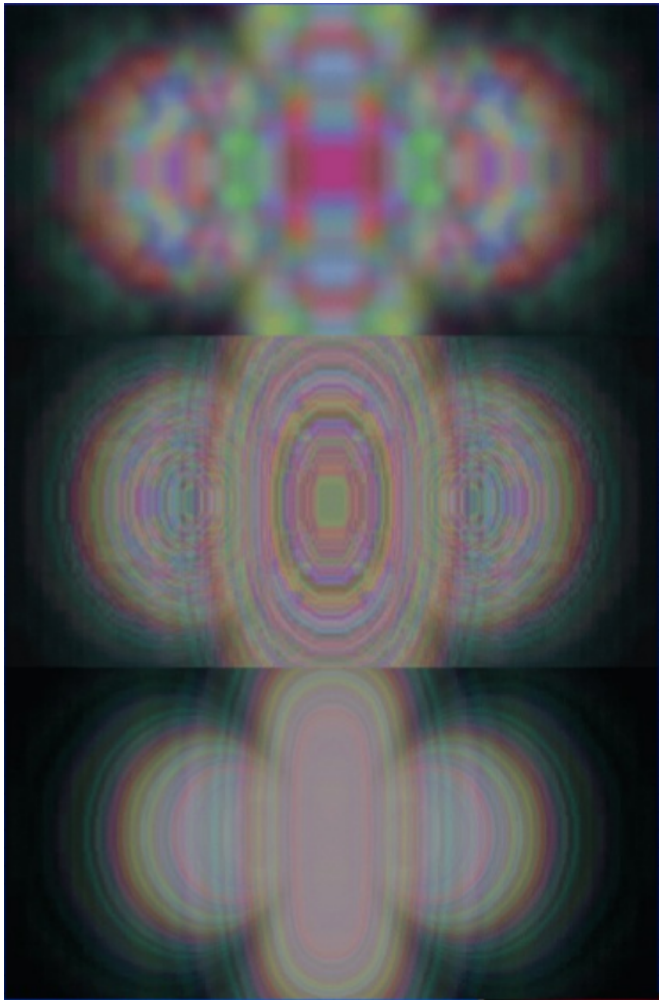


1D Transfer Functions



POST-CLASSIFICATION VS. PRE-CLASSIFICATION

- Pre-classification
  - TF applied before rendering
  - Interpolating already mapped data
- Post-classification
  - TF applied on the fly
  - Mapping interpolated input
- TF pre-integration
  - More precise integratal computation
  - Precompute integrals for each possible segment (start, end values)



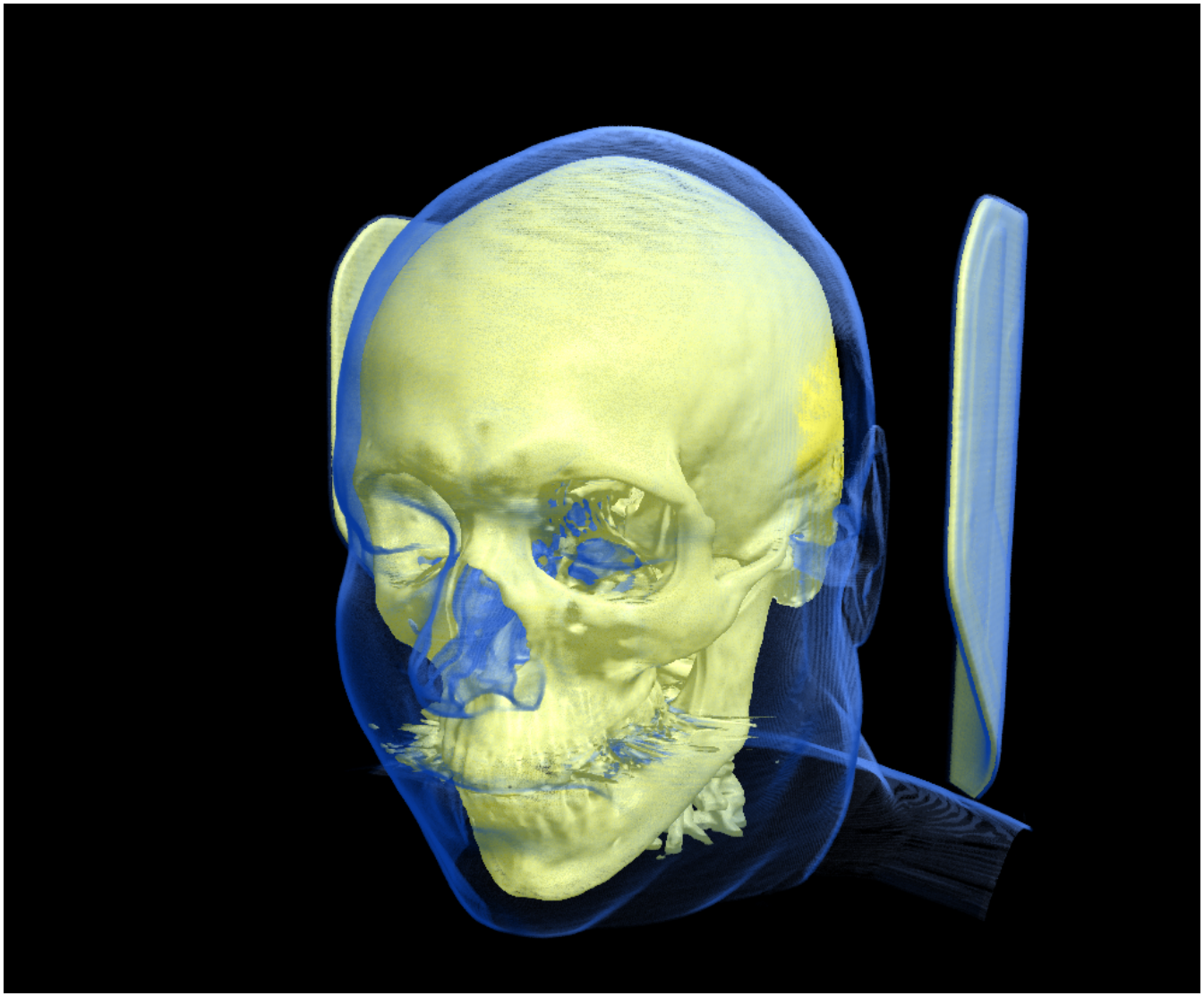
2D Transfer Functions

- Transfer function $h(v_1, v_2) : R^2 \rightarrow R^4$
- Ray sample: $h(f(\mathbf{x}), g(\mathbf{x}))$
- Maps two dimensional vector to RGBA color.
- Where we get the second dimension?.
  - Dual source CTs
  - Gradient magnitude ...

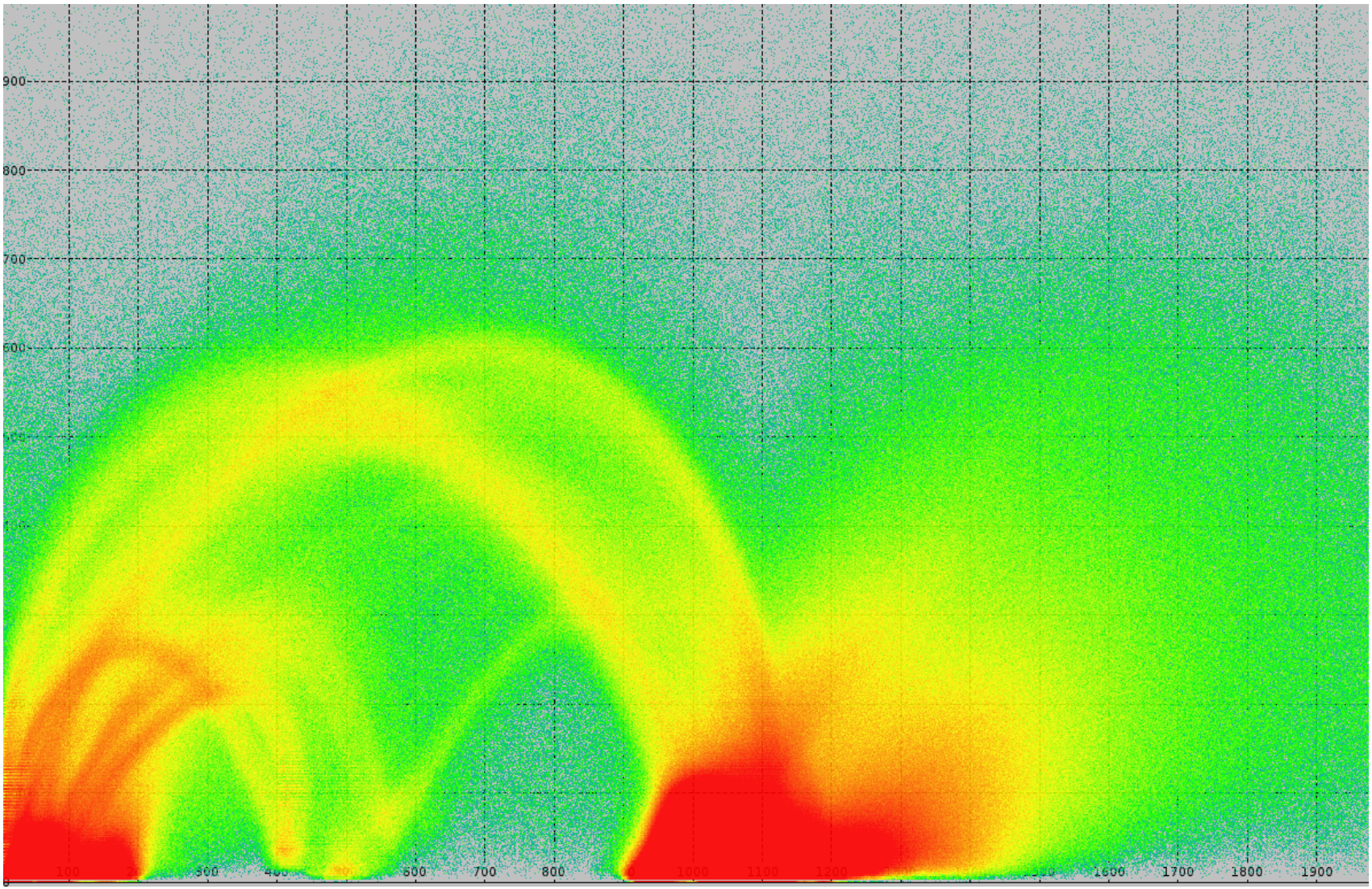
Gradient Magnitude

- Can be computed on the fly.
- Ability to separate borders from homogeneous regions.

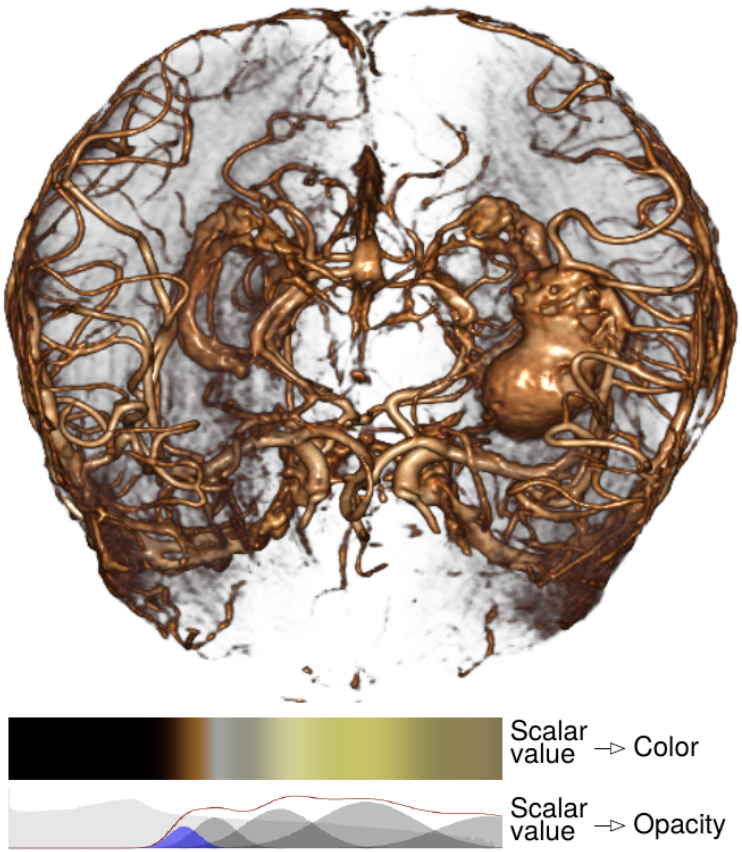
Gradient Magnitude

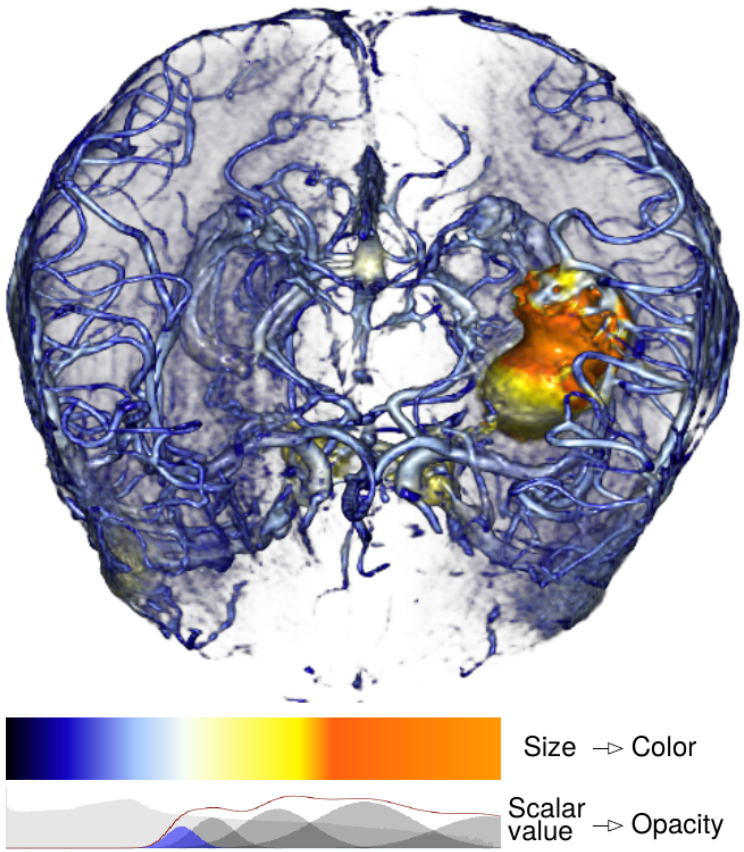


Scatter Plot

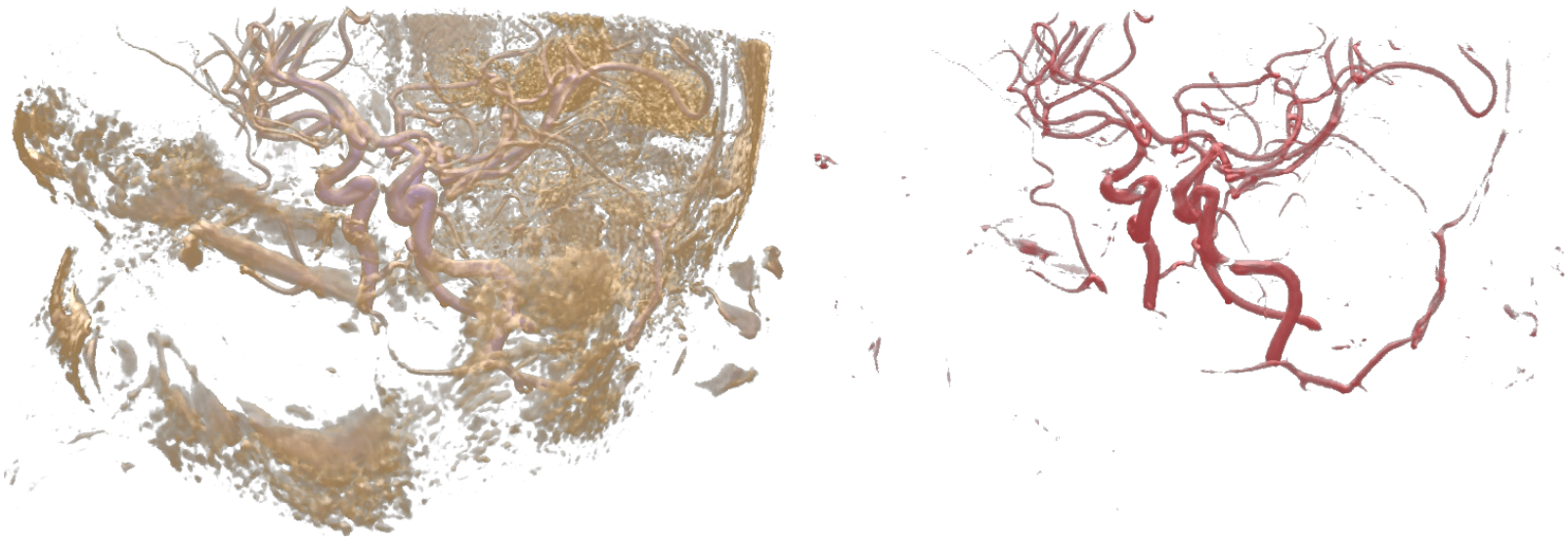


Size-based Transfer Functions





Shape Oriented Transfer Functions



Improvements

LIGHTING

- Self shadowing
- Material modeling
- Light scattering
- ...

JITTERING

- Uniform ray sampling – alias
- Hide behind noise:
  - Randomly shift ray origins, along view direction
  - Pregenerated random texture

SPEEDUP TECHNIQUES

- Early termination
  - Do not sample data behind opaque sections
- Empty space skipping
  - Large sparse data
  - Multiresolution
  - Skip sections without important data
  - Must be recomputed when TF changes

Isosurfaces

Polygonal Mesh

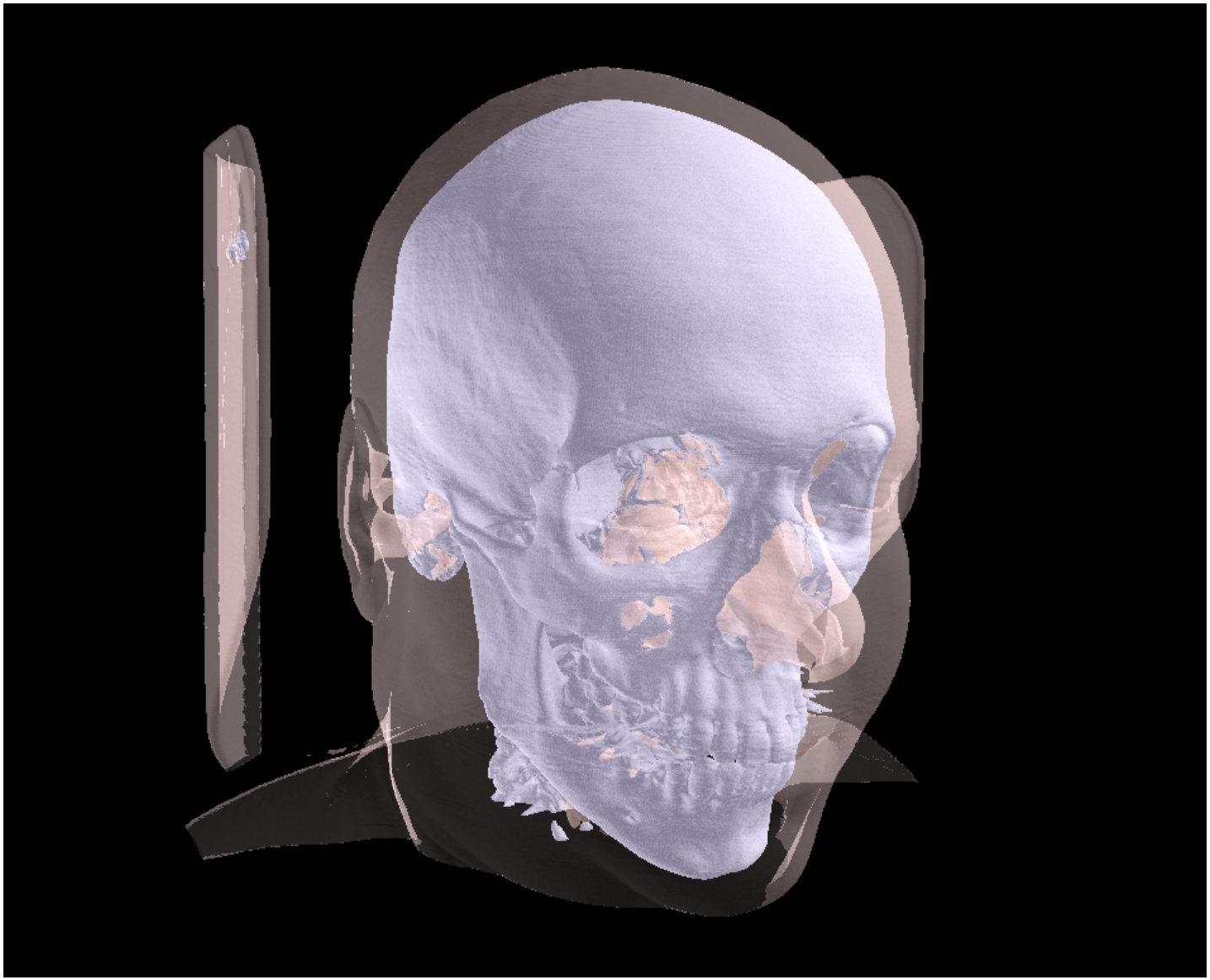
- Polygonal mesh representing level set
- Volume preprocessing:
  - Cuberille (+filtering)
  - Marching cubes, tetrahedra, ...
- Use normal rasterization pipeline for rendering

Realtime Isosurface Construction

- Ray-casting
  - Search for isovalue crossings
  - Fine search in subintervals for intersection point
  - Gradient for surface normal

ISO-SURFACES

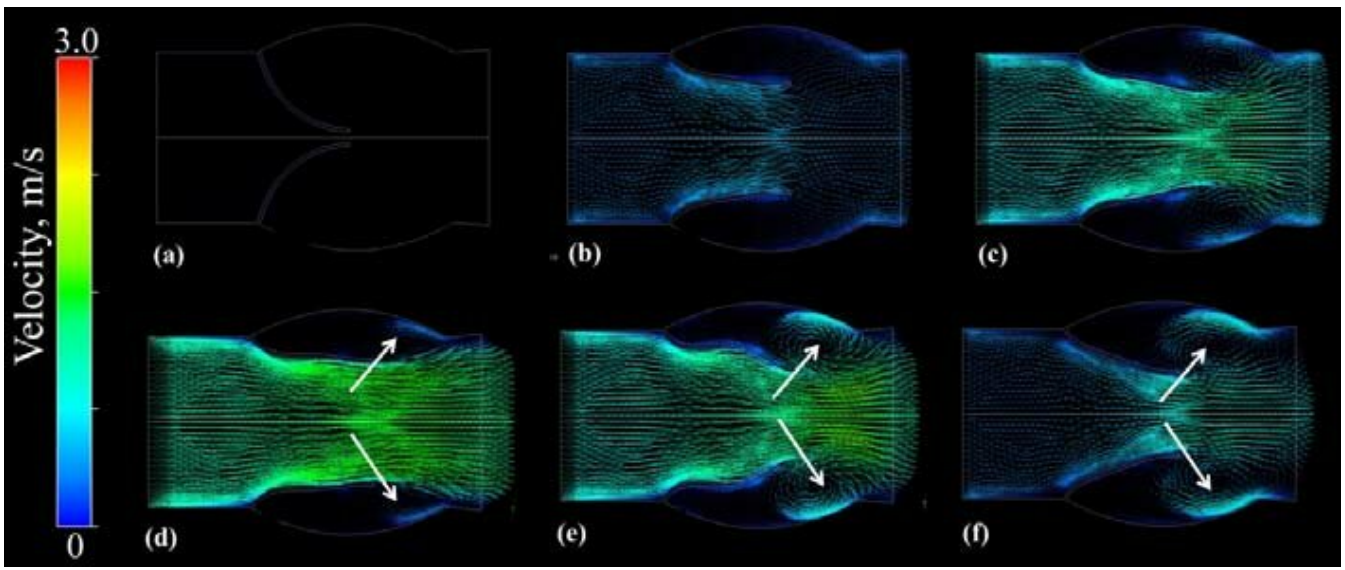
ISO-SURFACES



Vector Fields

Data Source

- Physical simulations:
  - Fluid dynamics
  - Particle simulations
  - Electromagnetic fields (Maxwell)



Numerical Integration

- Simulate motion under vector field influence
- Numerical integration
  - Euler method – low numerical stability, fast
  - Higher order Runge-Kutta methods

Differential Operators

- $\nabla = \frac{\partial}{\partial x}\frac{\partial}{\partial y}\frac{\partial}{\partial z}$
- Gradient
  - $\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \frac{\partial f(\mathbf{x})}{\partial x_3}\right)$
- Divergence
  - $\nabla \cdot \mathbf{F} = \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} + \frac{\partial \mathbf{F}_z}{\partial z}$
- Curl
  - $\nabla \times \mathbf{F} = \left(\frac{\partial \mathbf{F}_z}{\partial y} - \frac{\partial \mathbf{F}_y}{\partial z}, \frac{\partial \mathbf{F}_x}{\partial z} - \frac{\partial \mathbf{F}_z}{\partial x}, \frac{\partial \mathbf{F}_y}{\partial x} - \frac{\partial \mathbf{F}_x}{\partial y}\right)$

Glyphs, Icons, Probes

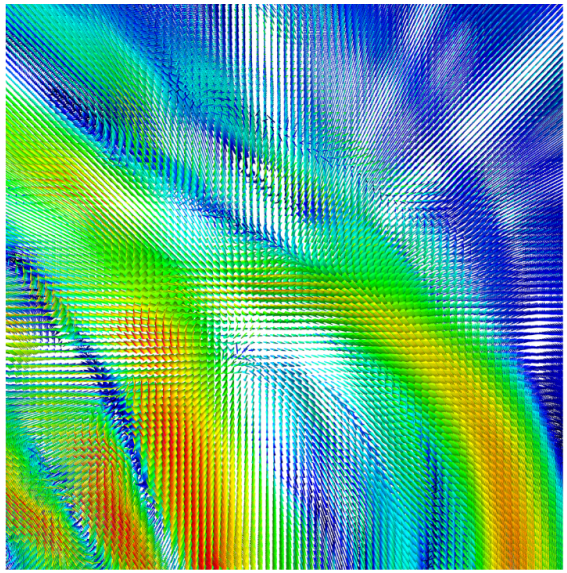
- Sample vector field:
  - Arrows
  - Lines
  - Balls, ellipsoids
  - Ribons
- Other characteristics represented by shape, color
- To prevent clutter:
  - Importance sampling
  - Slice-probe through vector field

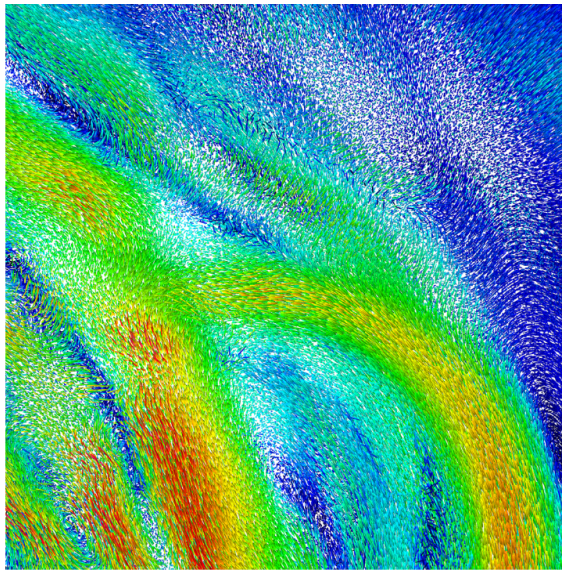
Rendering Glyphs

- Large number of similar geometries
- Instanced rendering
  - Impostors for complicated geometries
- Geometry shader:
  - From point samples generate glyph geometry

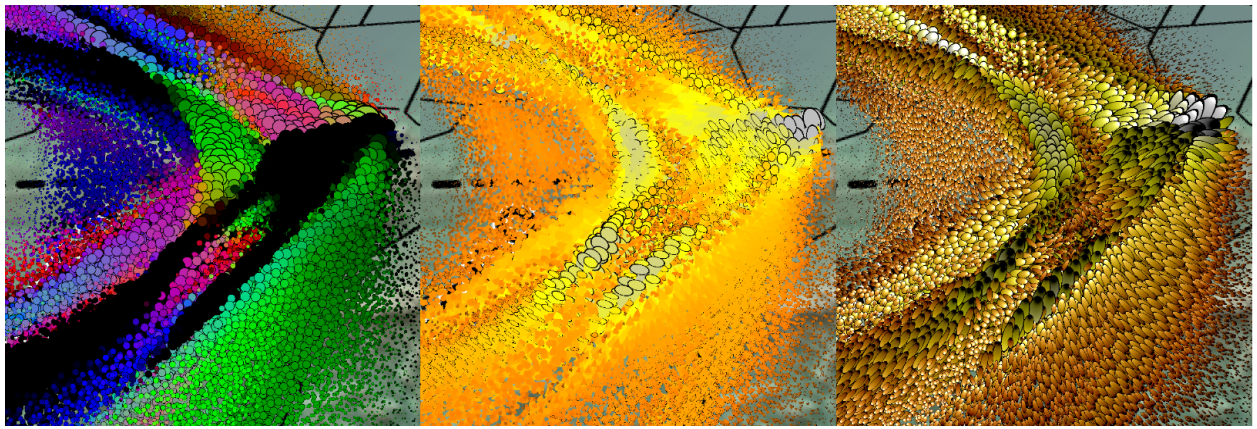
Glyph Examples

- Sampling jitter



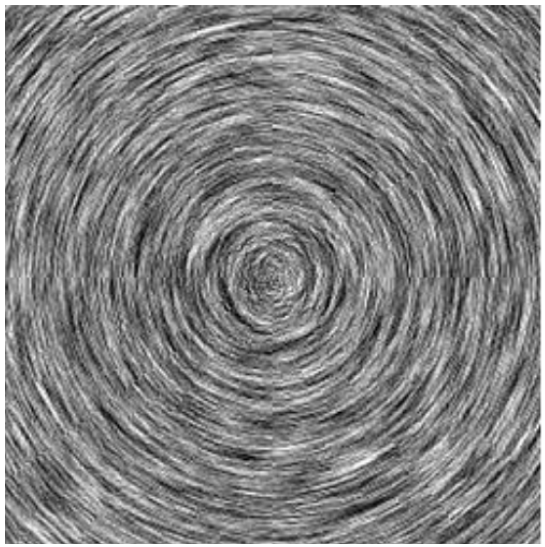


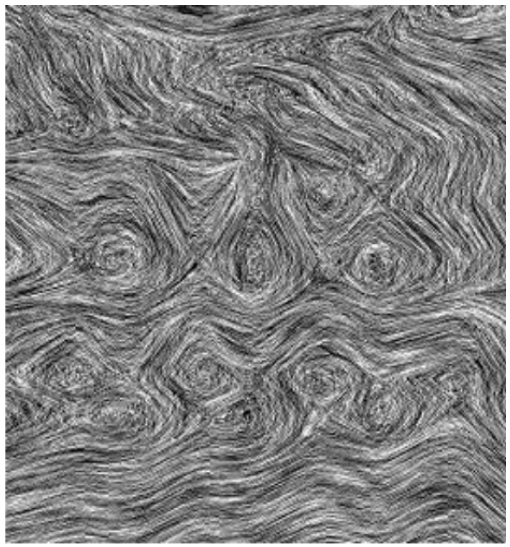
- Different shading



Line Integral Convolution

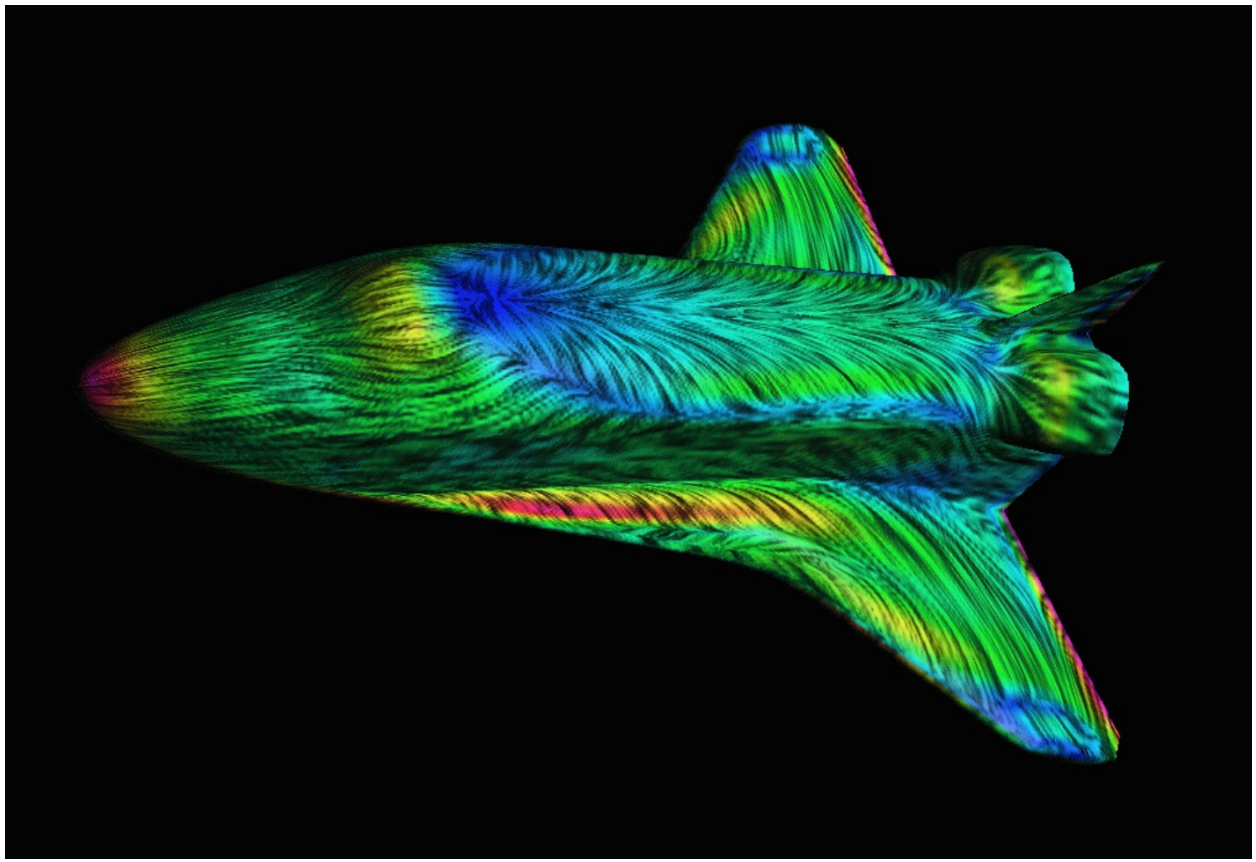
- Underlying texture blurred along vector directions
  - Multiple texture accesses in fragment shader – integration





LIC on Surface

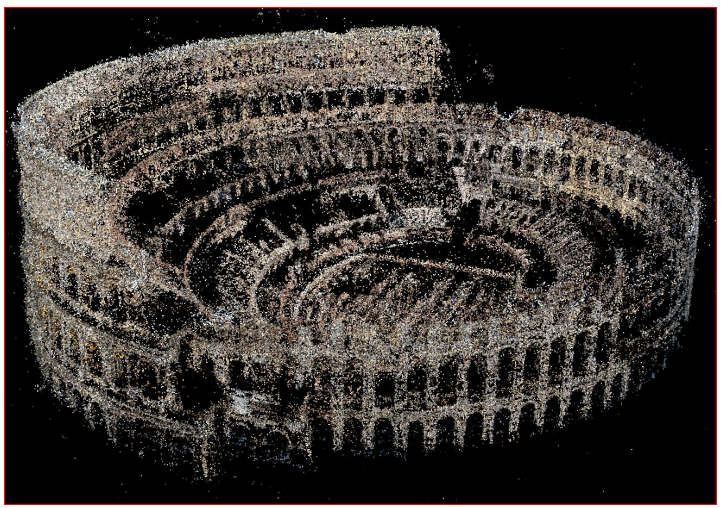
- Compute in object fragment shader



Point Clouds

Data Source

- Surface points:
  - 3D scanner output
  - Scene reconstruction:
    - Stereo cameras
    - Camera + depth sensor (Kinect)
    - Single moving camera
- Random spatial samples:
  - Unstructured vector field
  - Unstructured volume



Point Cloud Rendering

- Glyph for each point
  - Colored/textured facets
- Glyph for group of points
  - Size, shape – properties of point group

Volume Rendering

- Unstructured volume samples:
  - Datastructure for fast queries (octree, . . .)
  - Ray sample – weighted average of points in certain radius
- Surface reconstruction:
  - Distance field
  - Isosurface rendering