COSE436

Lecture 23: Volume Rendering II

Dec 3, 2024
Won-Ki Jeong
(wkjeong@korea.ac.kr)

KOREA UNIVERSITY

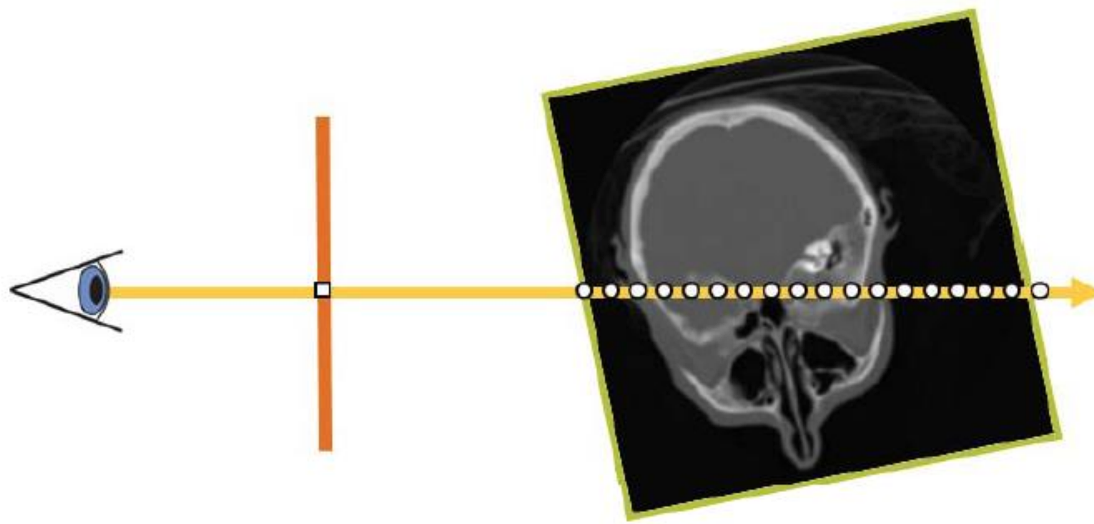
Outline

- Rendering methods
- Classification & transfer function
- Pre-integrated & isosurface volume rendering

Ray Casting Process

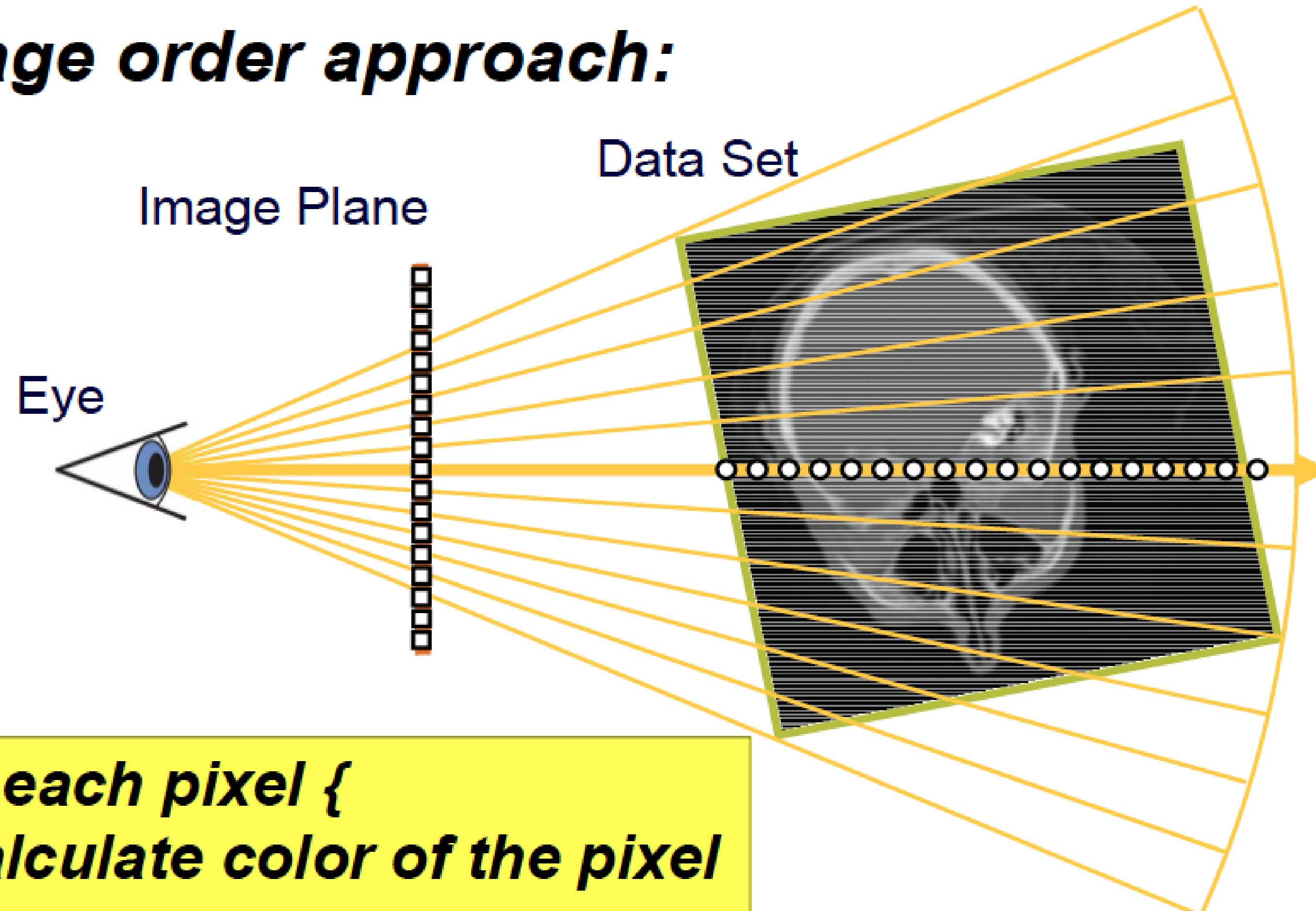
- Ray setup
- For every ray
  - Resample scalar value
  - Classification
  - Shading
  - Compositing

iterative sampling



Volume Rendering

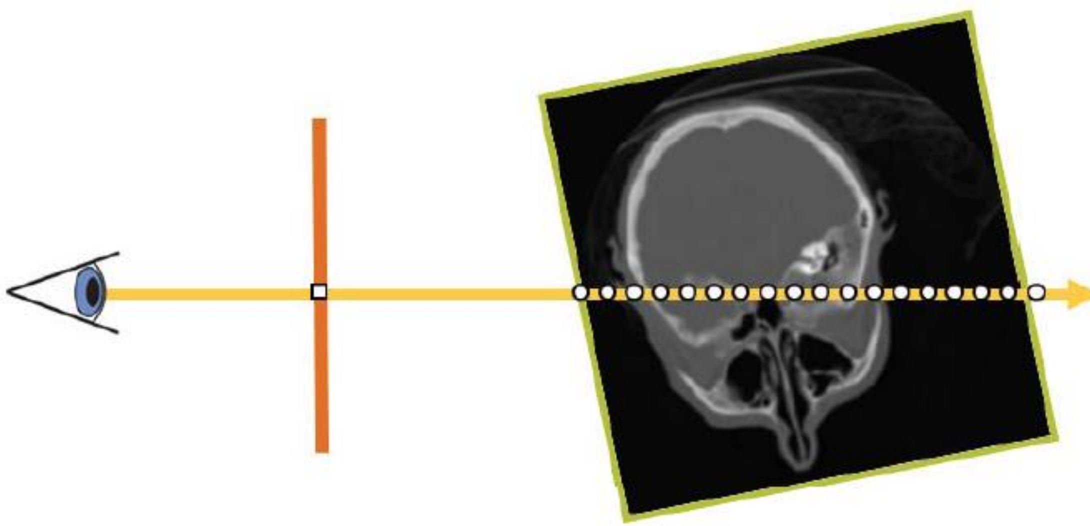
Image order approach:



For each pixel {
calculate color of the pixel
}

Ray Setup

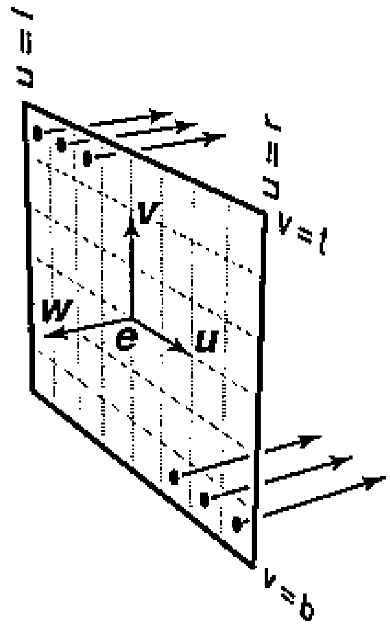
- Two approaches
  - Procedural ray/box intersection
  - Rasterize bounding box

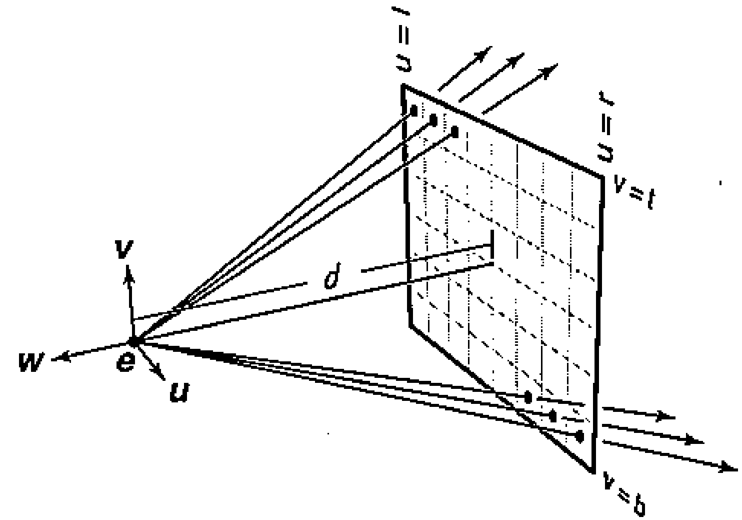


Procedural Ray Setup/Termination

- Fragment shader / CUDA kernel computes ray equation and bounding box intersection per pixel
  - Ray is defined by camera position and pixel location
- Pro : simple and self-contained
- Con : full computational load per pixel/fragment

Procedural Ray Setup/Termination





<u>Orthogonal view</u>

Ray direction : -**w**

Ray origin : **e** + u**u** + v**v**

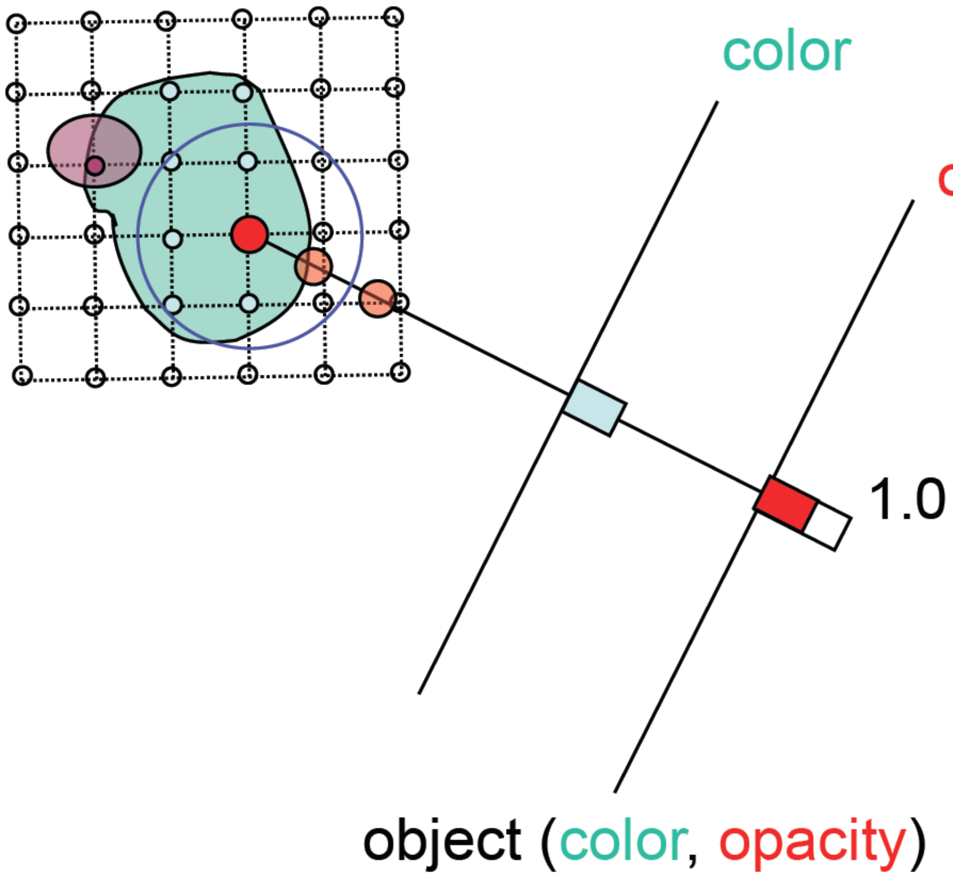
<u>Perspective view</u>

Ray direction : -d**w** + u**u** + v**v**

Ray origin : **e**

Procedural Ray Setup/Termination

- Trilinear interpolation, front-to-back

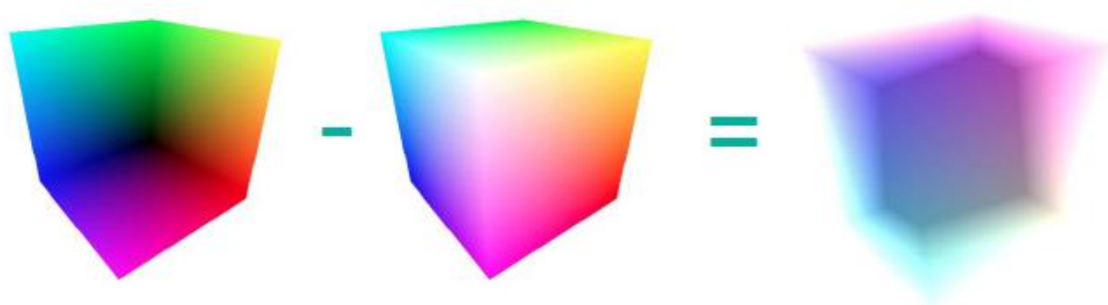


$$C'_{i+1} = C'_i + (1 - A'_i)C_{i+1}$$
$$A'_{i+1} = A'_i + (1 - A'_i)A_{i+1}$$

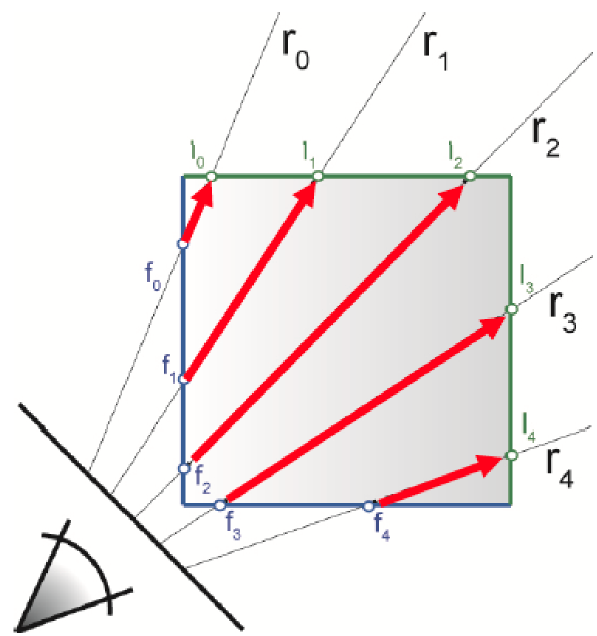
Weiskopf, Machiarju, Moeller

Rasterization-Based Ray Setup

- Rasterize bounding box
  - Fragment == ray
  - Render front / back faces separately
  - Rasterization gives location
  - Subtraction gives direction



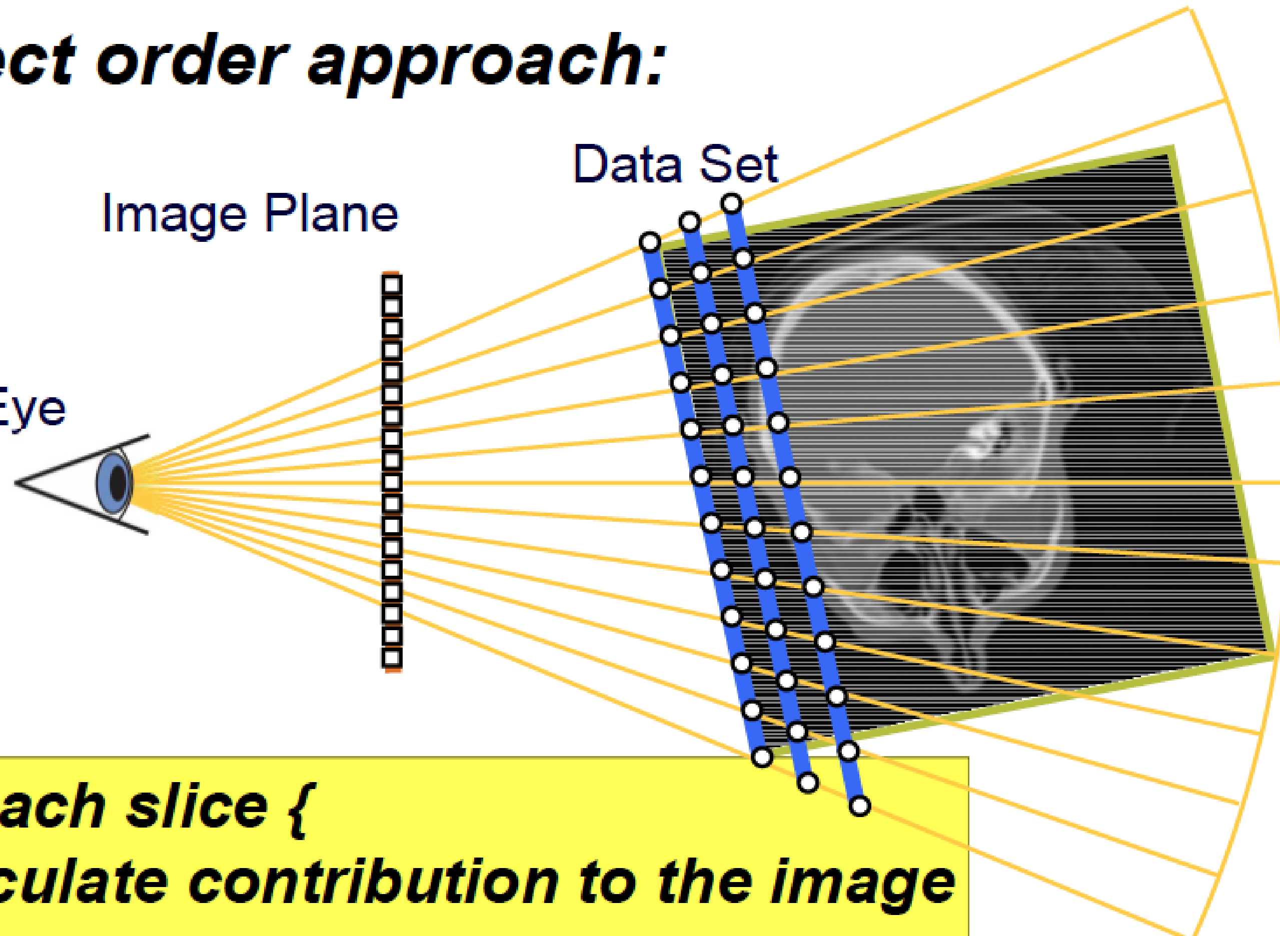
back faces   front faces



Identical for orthogonal & perspective projection!

Volume Rendering

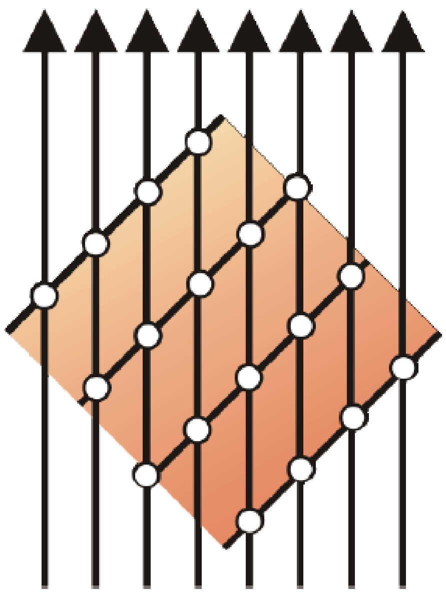
Object order approach:

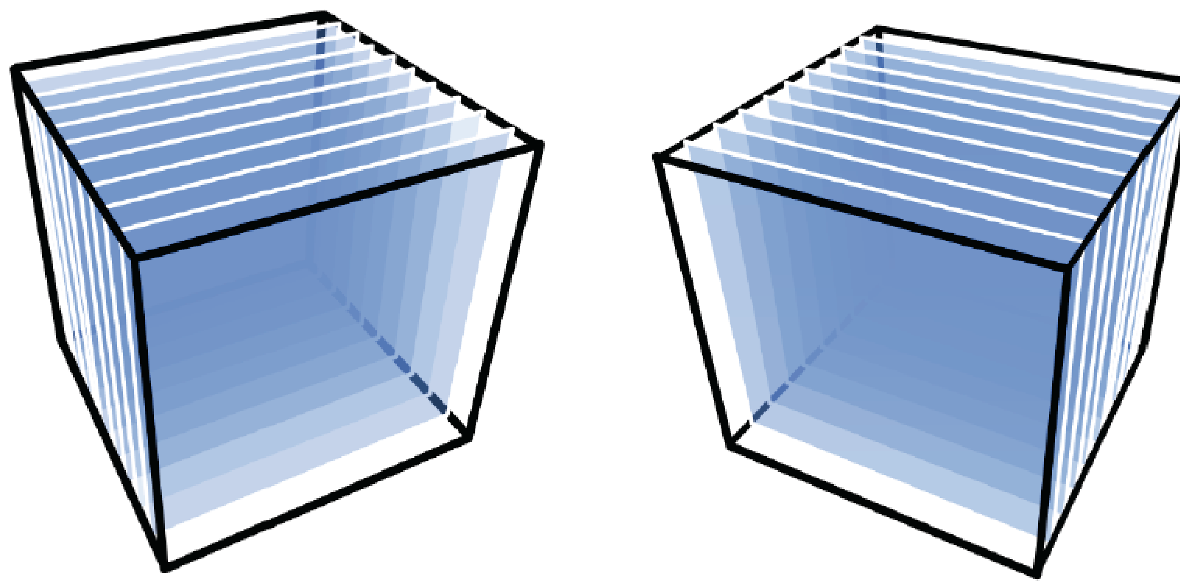


For each slice {
calculate contribution to the image
}

2D Texture Slices

- Object-aligned slices, back-to-front
- Three stacks of 2D textures (x-y, y-z, z-x)
- Bilinear interpolation

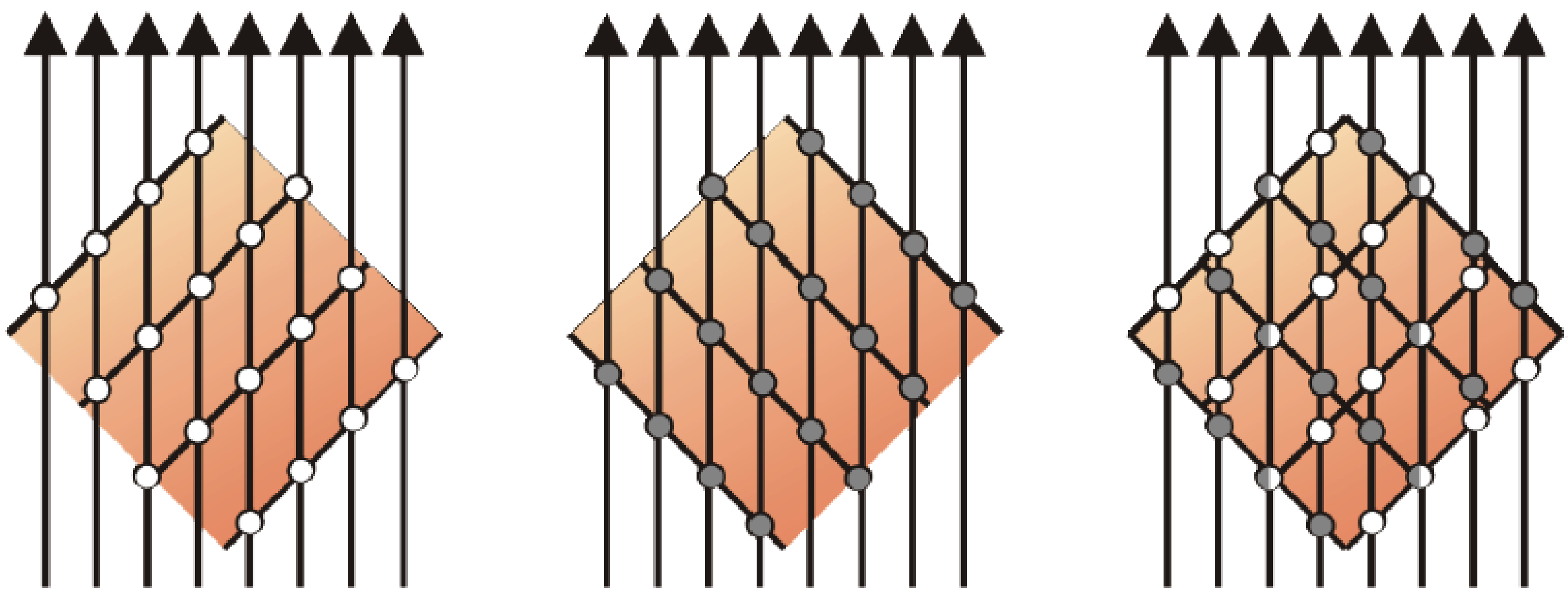




Weiskopf, Machiarju, Moeller

2D Texture Slices

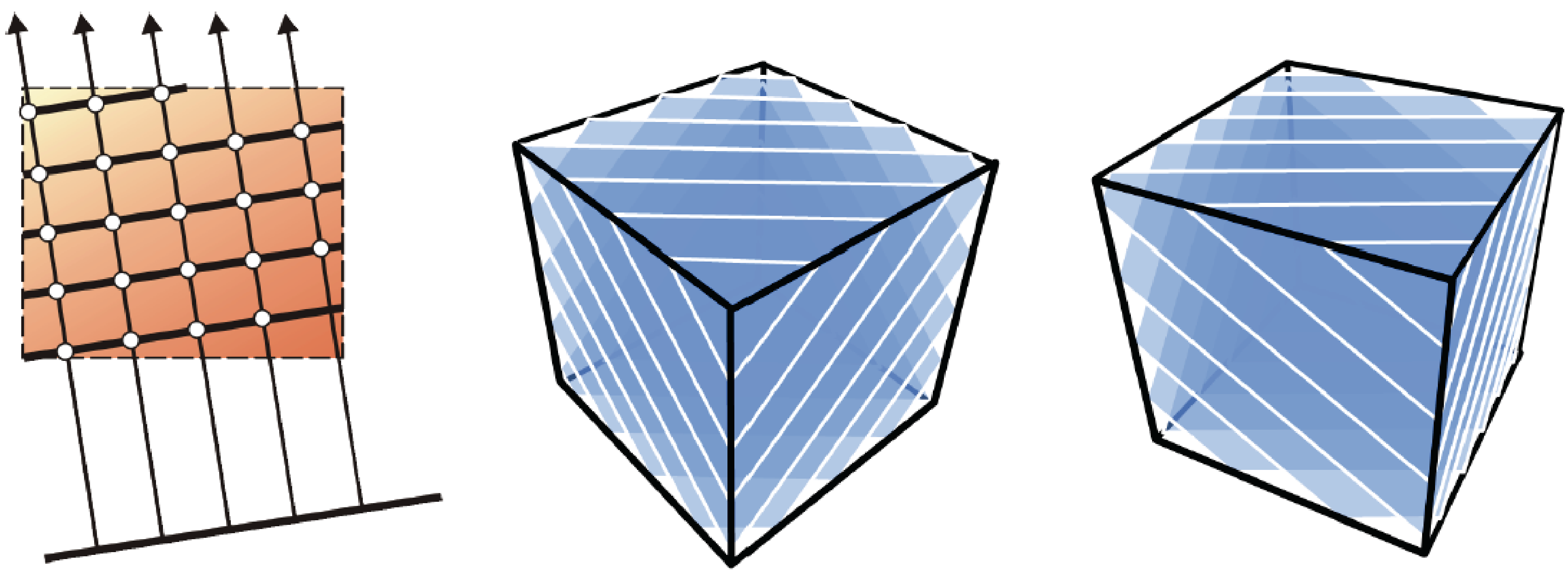
- Artifacts when stack is viewed close to 45 degrees
  - Location of sampling points may change abruptly



Weiskopf, Machiarju, Moeller

3D Textures

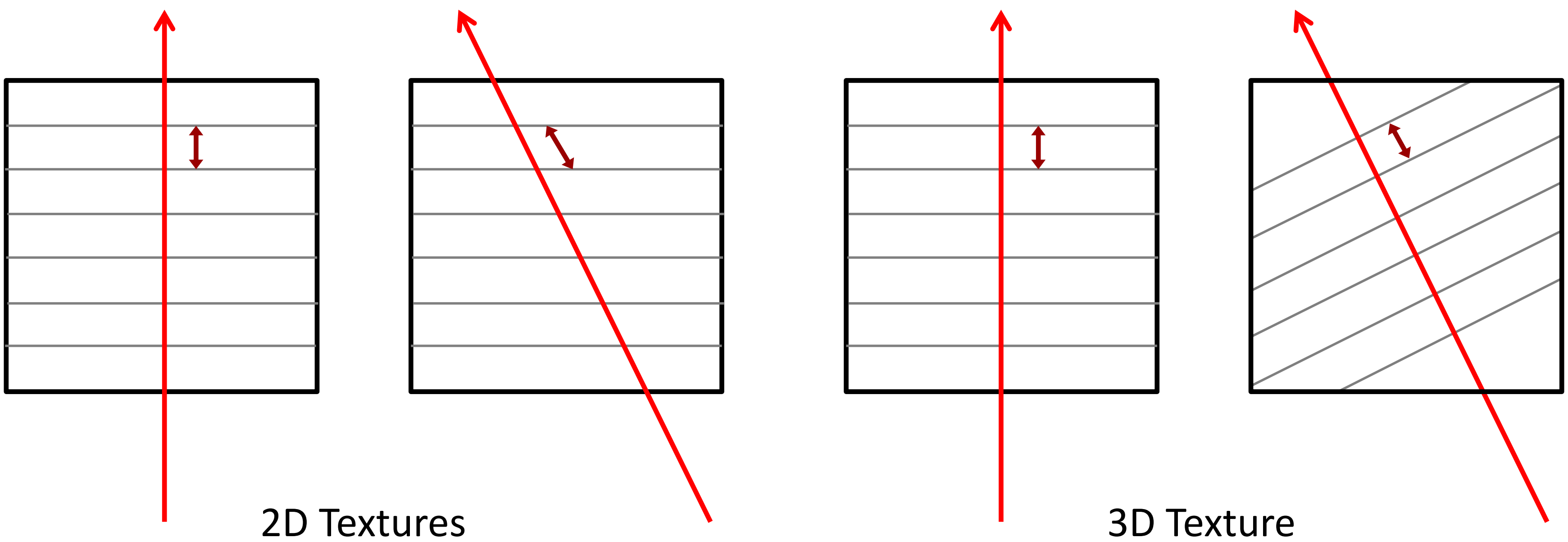
- View-aligned slices, back-to-front
- Single 3D texture
- Trilinear interpolation = better image quality



Weiskopf, Machiarju, Moeller

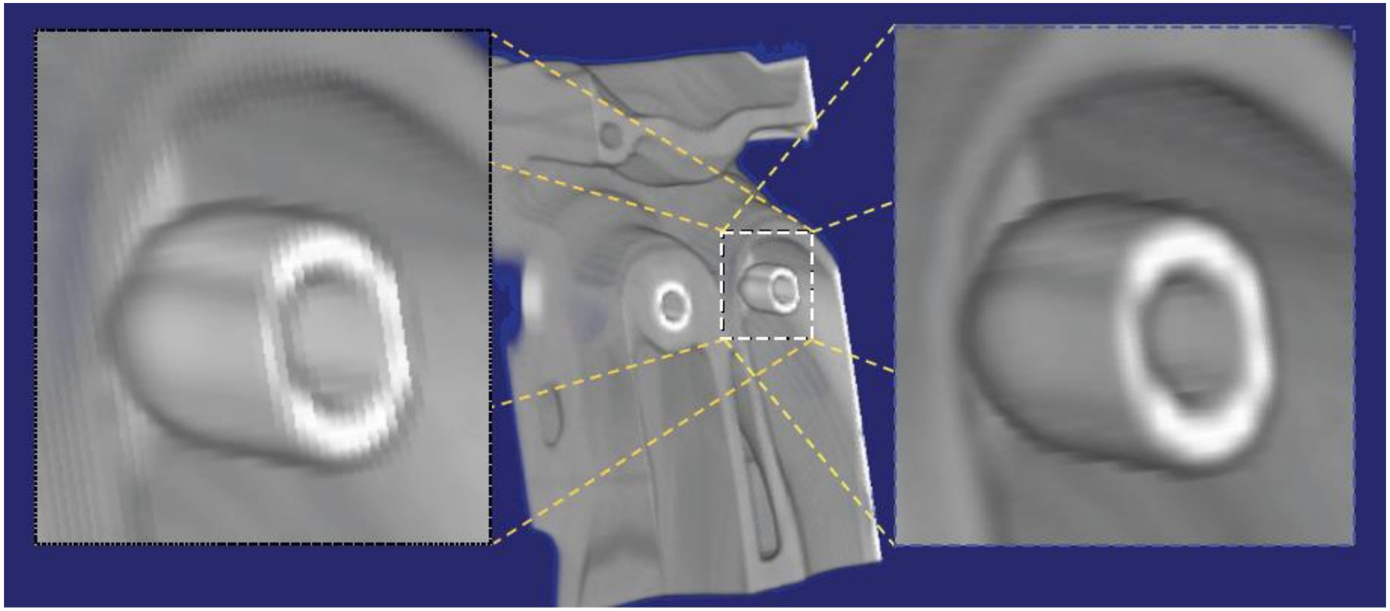
3D Textures

- Constant Euclidean distance between slices along a ray when view direction is changed



3D Texture

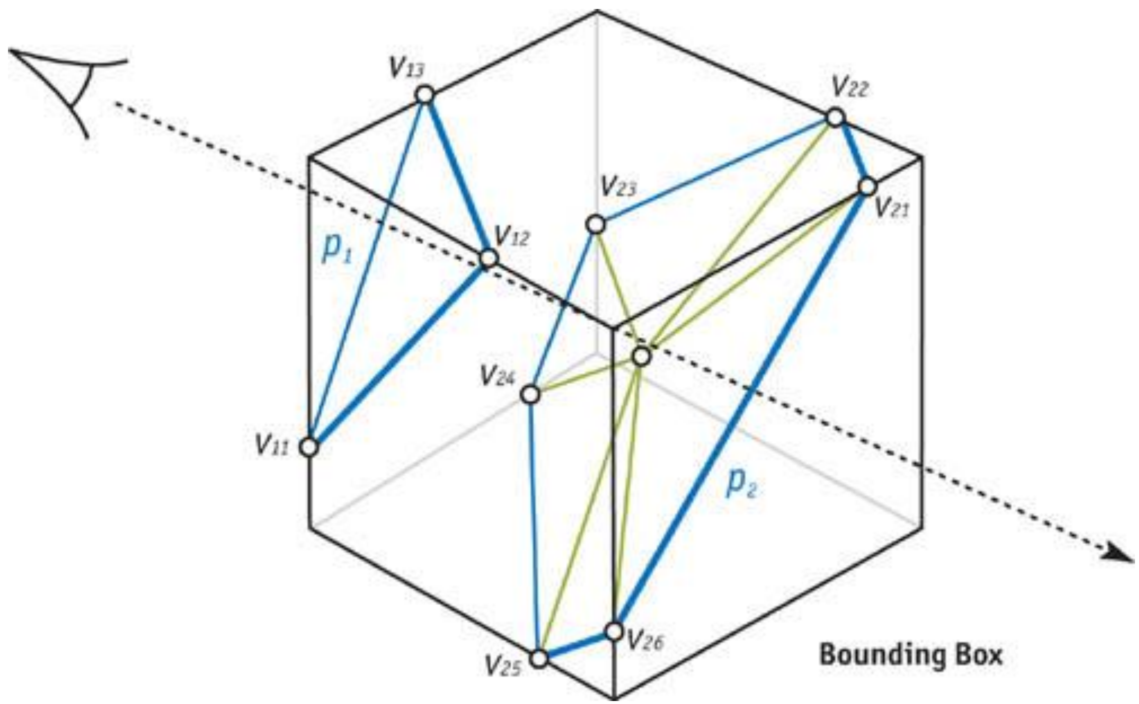
- No artifacts due to inappropriate viewing angles
- Increasing sampling rate is easy with 3D tex.
  - Use more slices



Weiskopf, Machiarju, Moeller

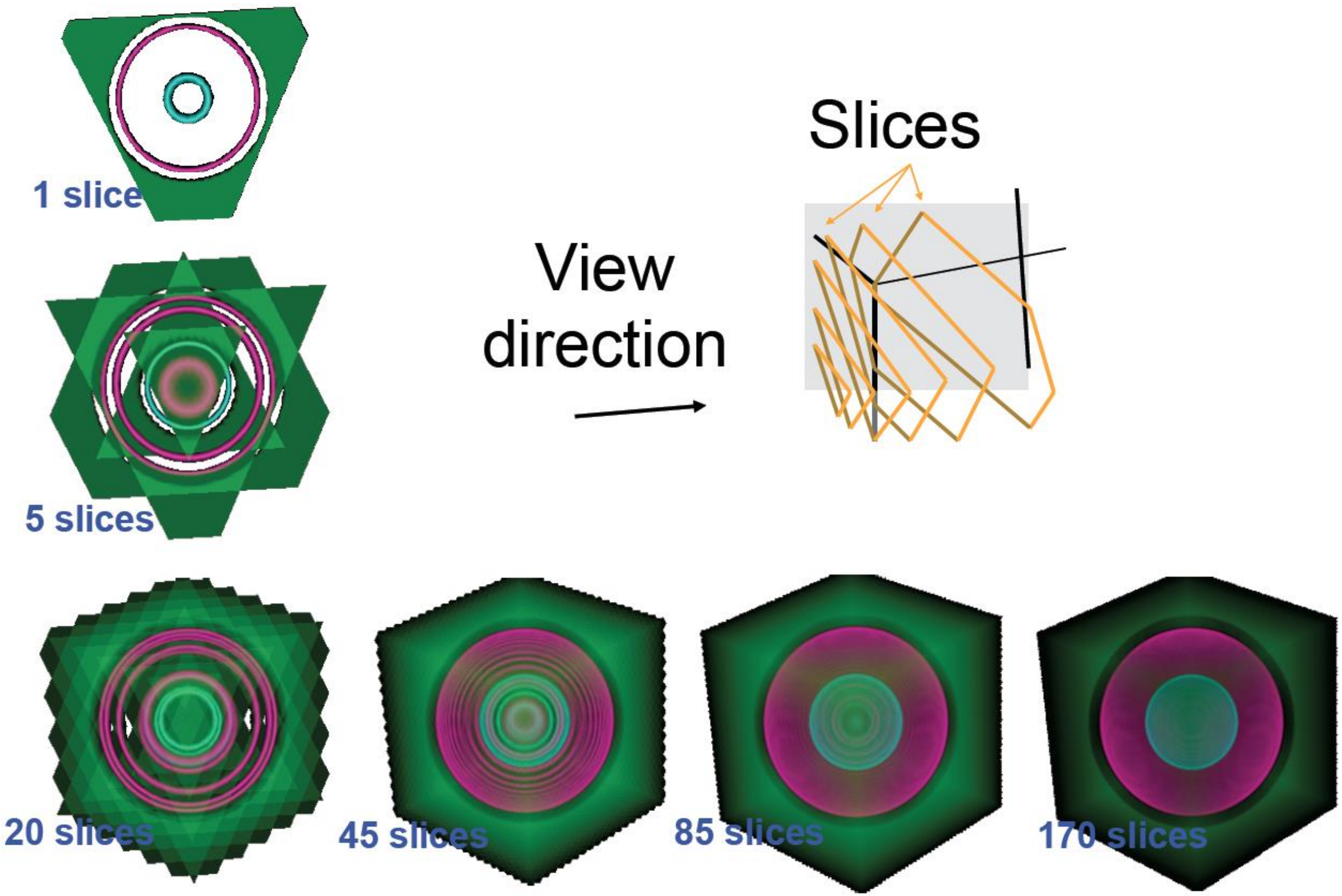
Proxy Geometry

- Plane-bbox intersection & tessellation with center point



GPU Gems

Rendering Quality



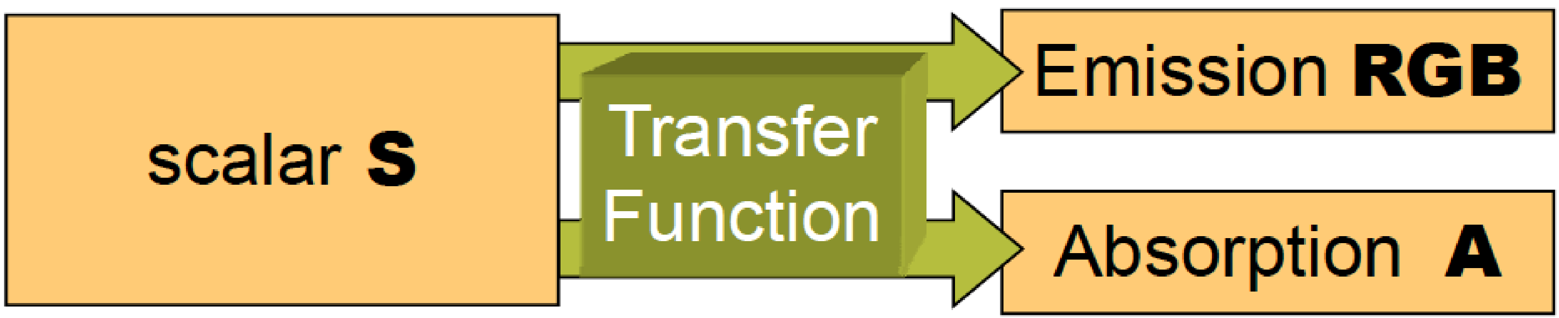
Classification

- User defines look of the data by
  - Change per-voxel color and transparency
  - How? : transfer function

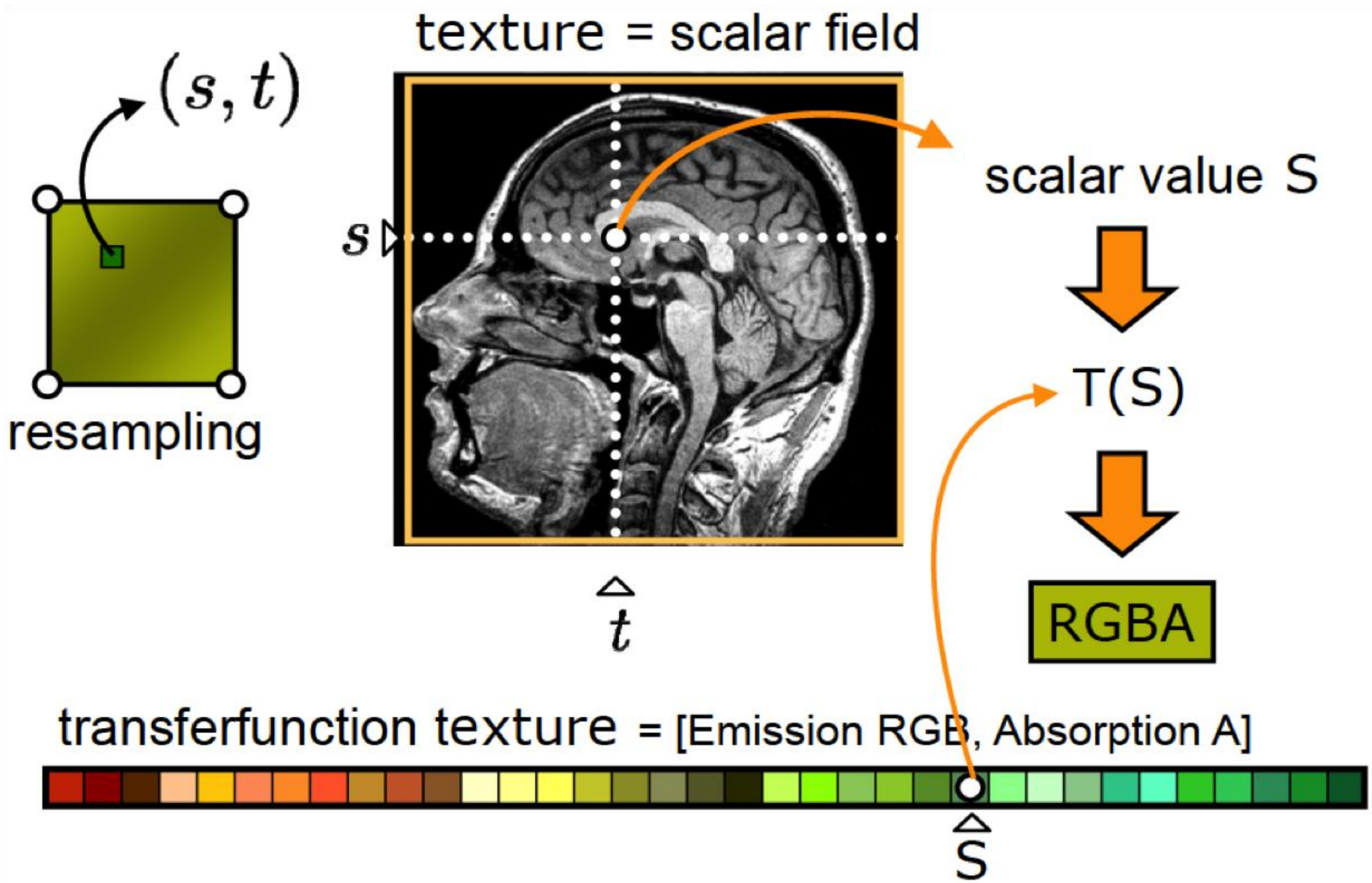
Transfer Function

- Input volume contains only scalar value
- Transfer function T maps scalar to vector
  - (R,G,B) : color or emission
  - A : absorption

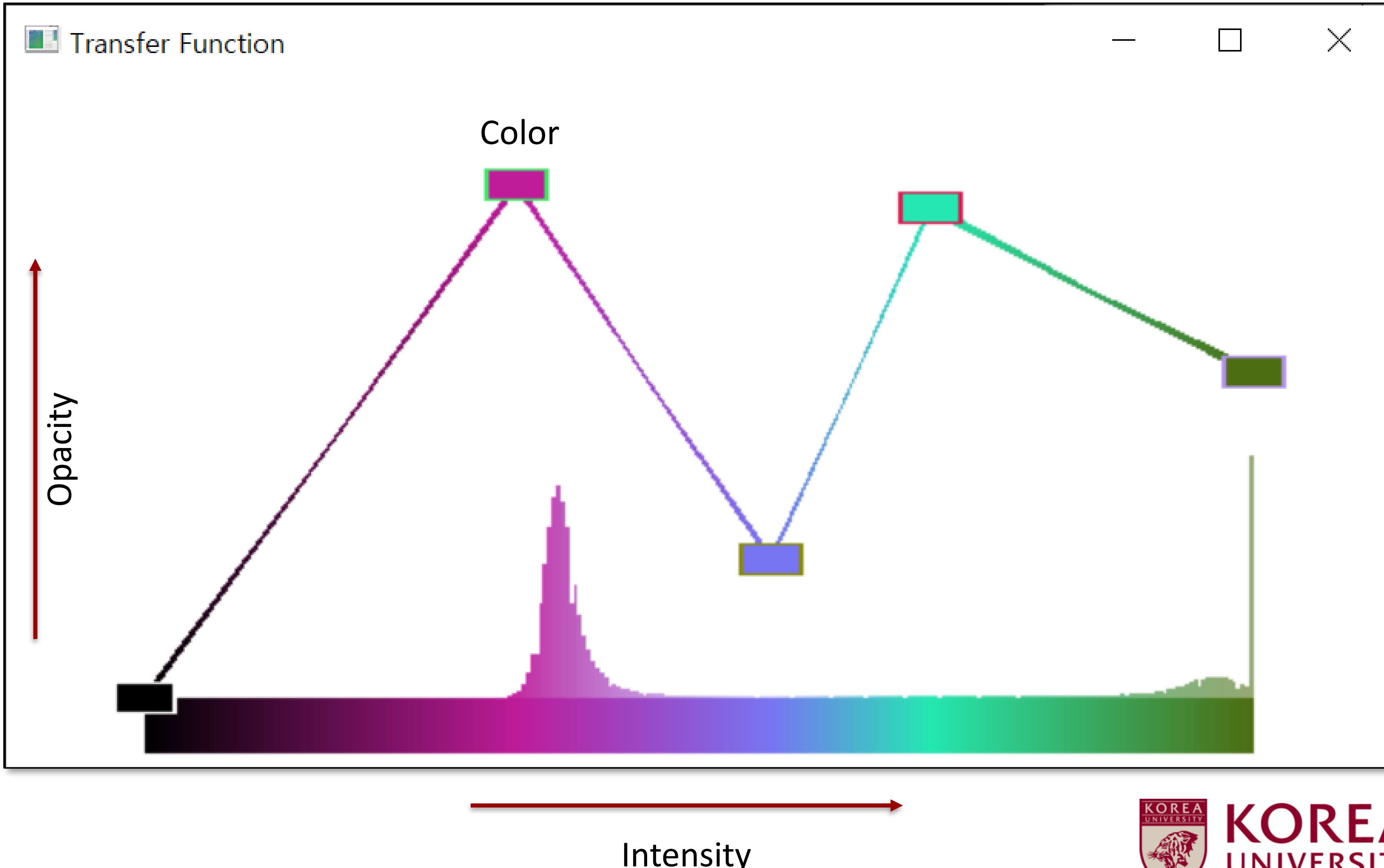
$$T(S) = (R, G, B, A)$$



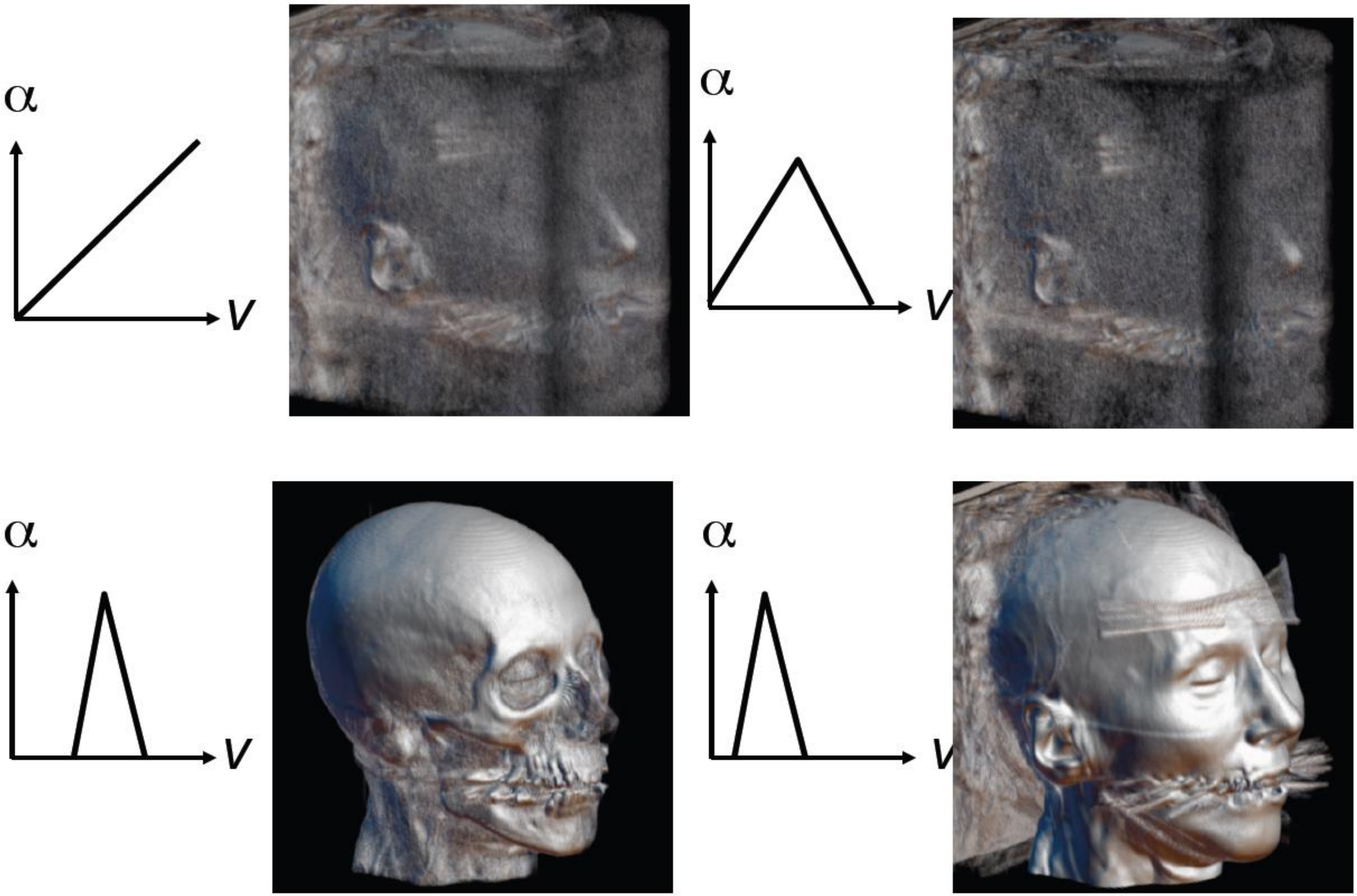
Transfer Function Application



1D Transfer Function Example



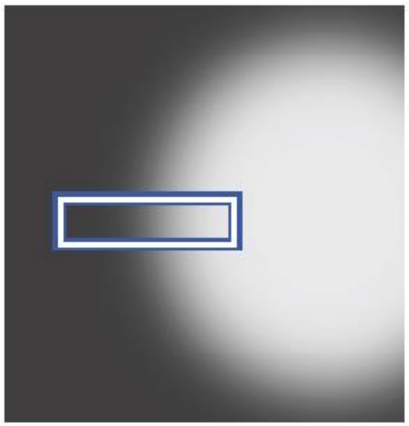
Setting Transfer Function is Hard!

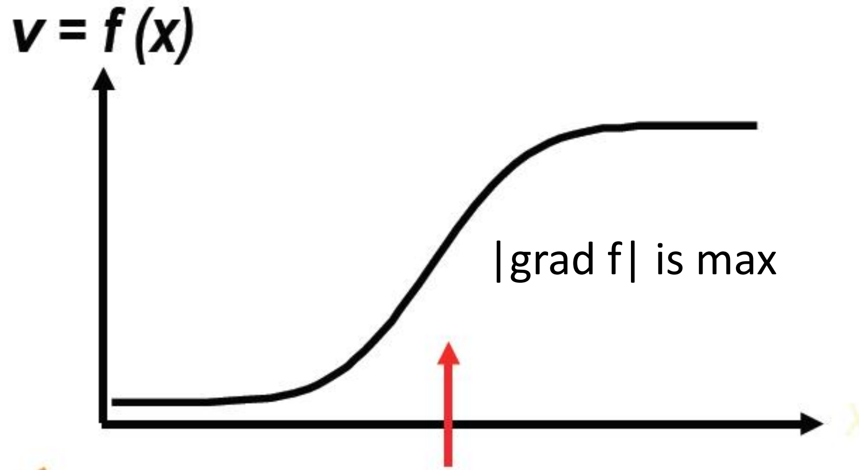


Gordon Kindlmann

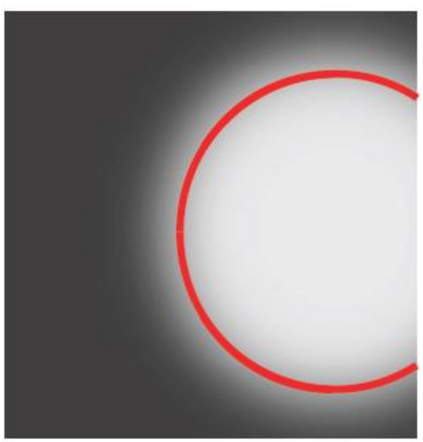
Finding Edges

"Where's the edge?"





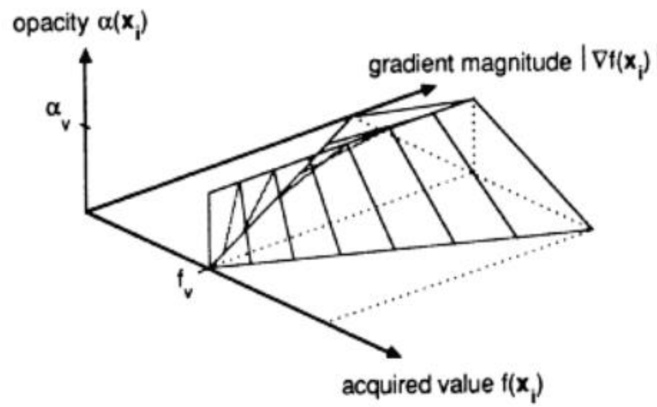
" here's the edge "

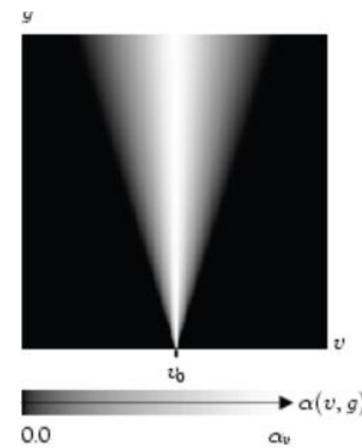


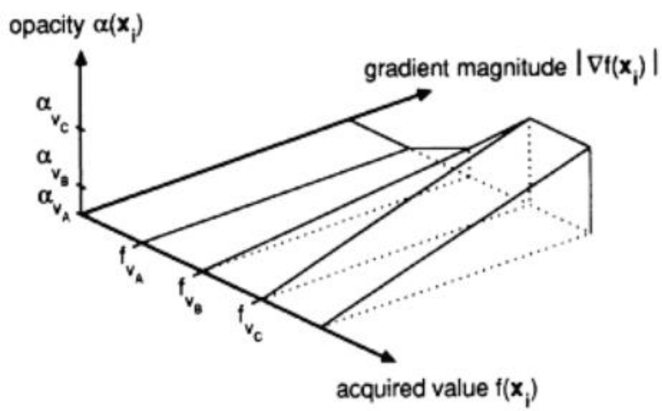
Result: edge pixels

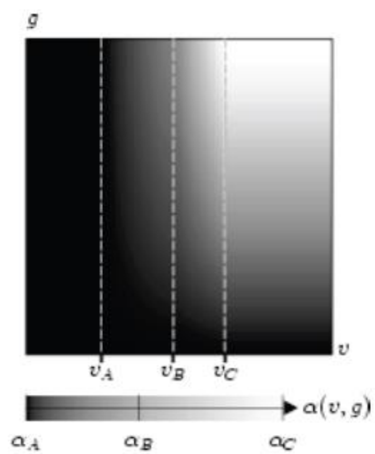
2D Transfer Function

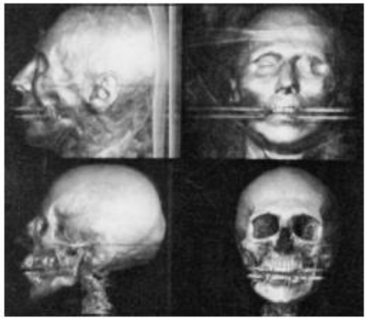
- Assign opacity from data value and |grad f|







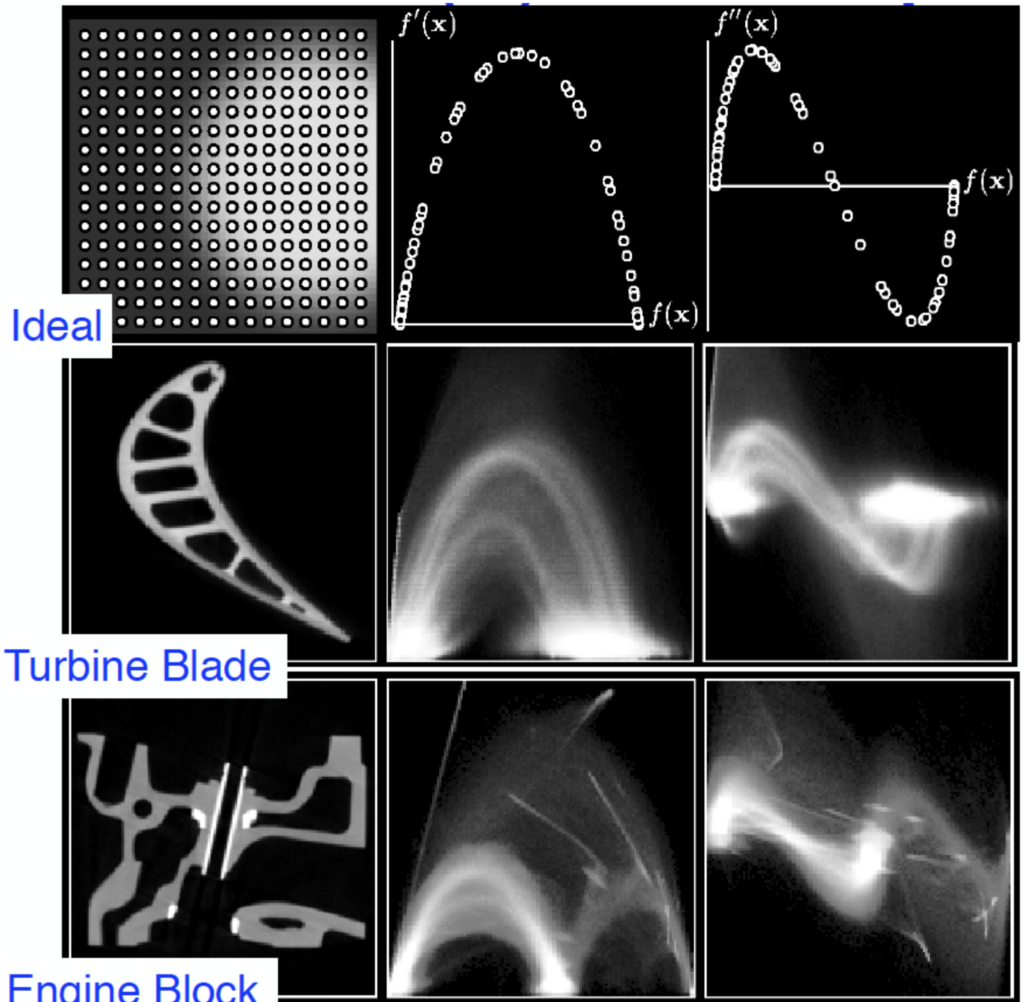




Gordon Kindlmann

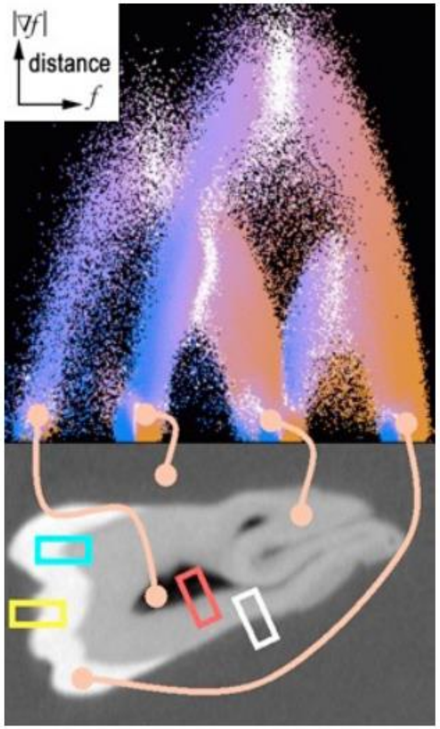
2D Scatter Plot

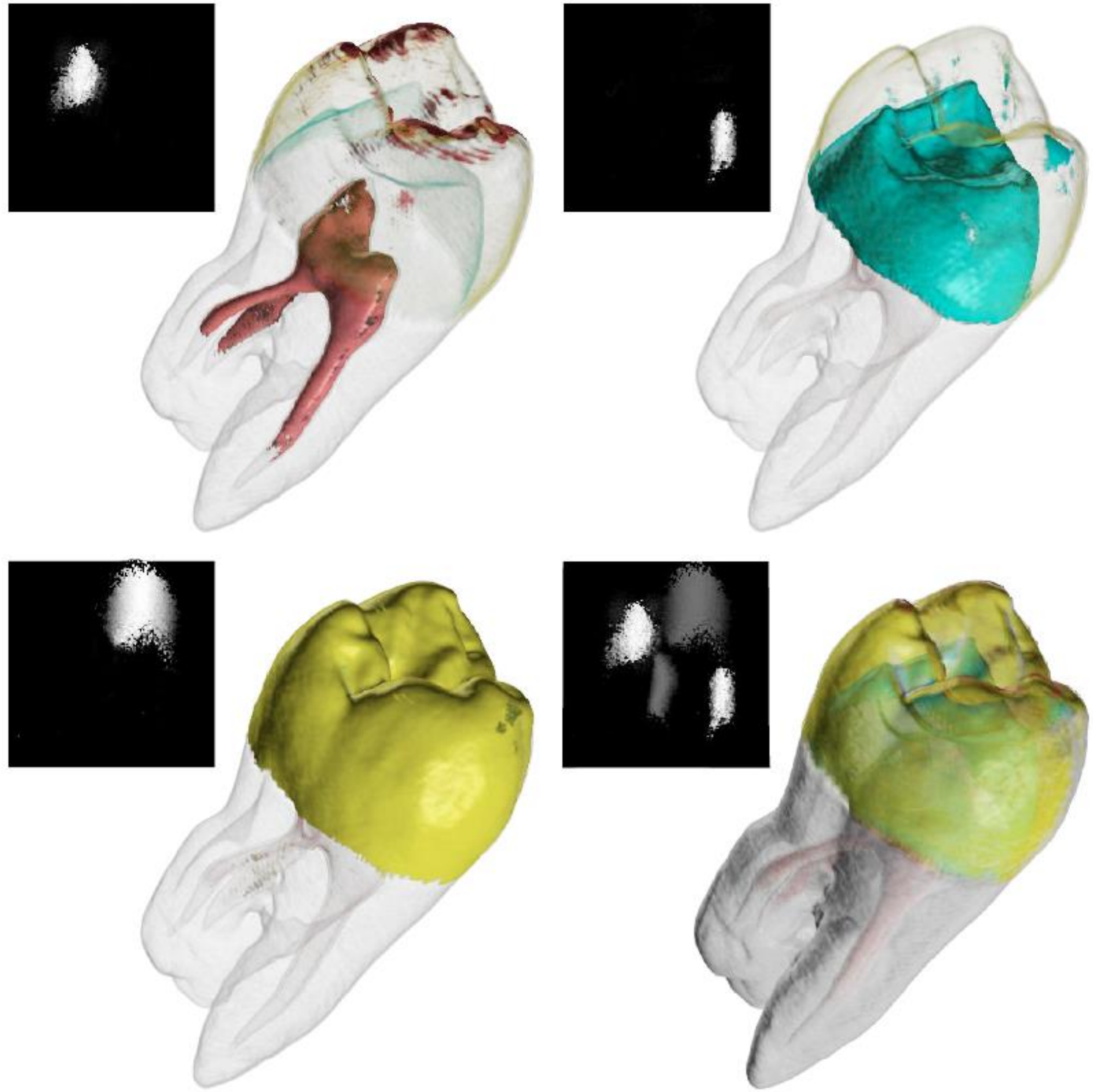
- Project histogram volume to 2D scatterplots



Gordon Kindlmann

2D Transfer Function

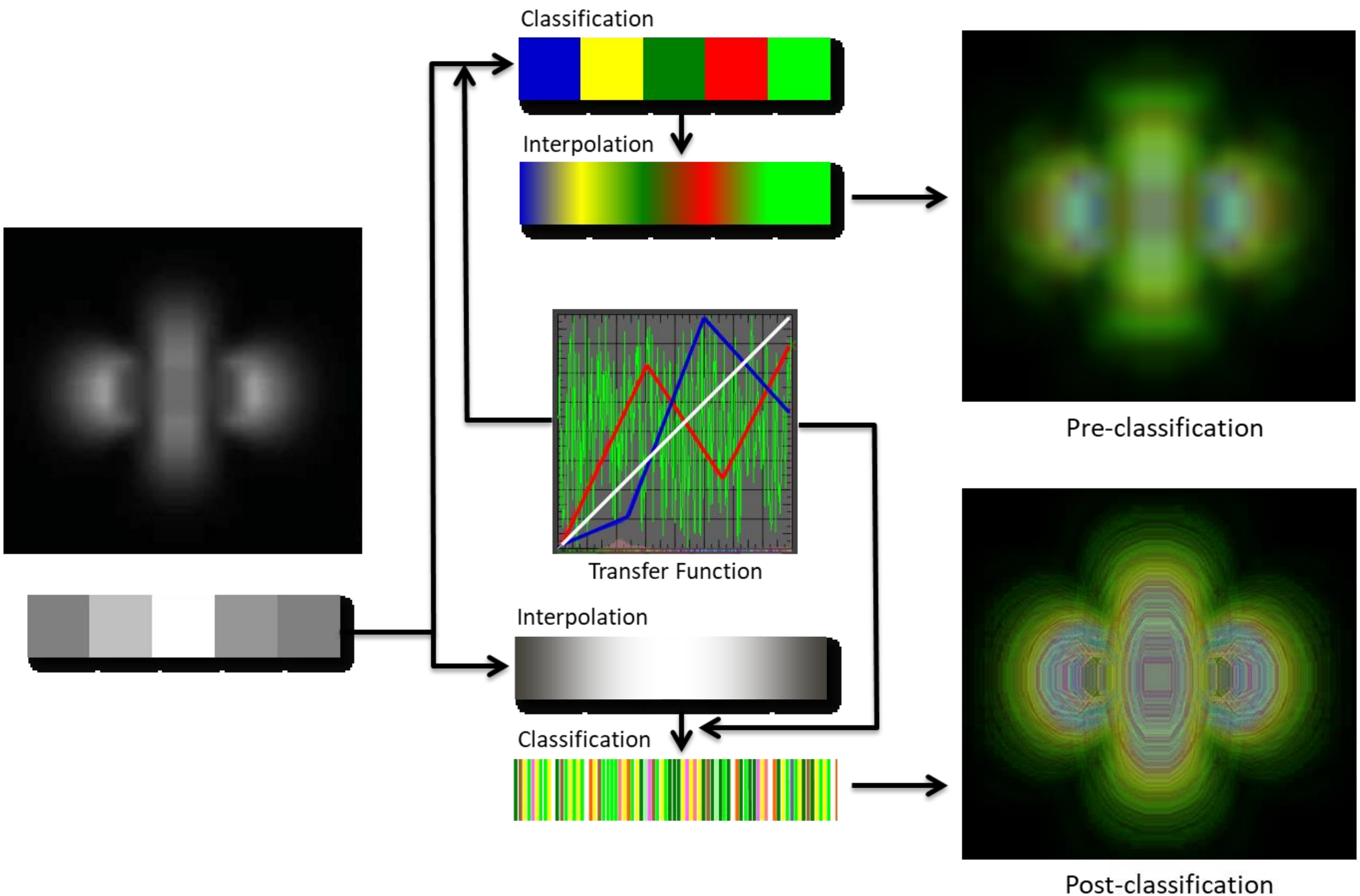




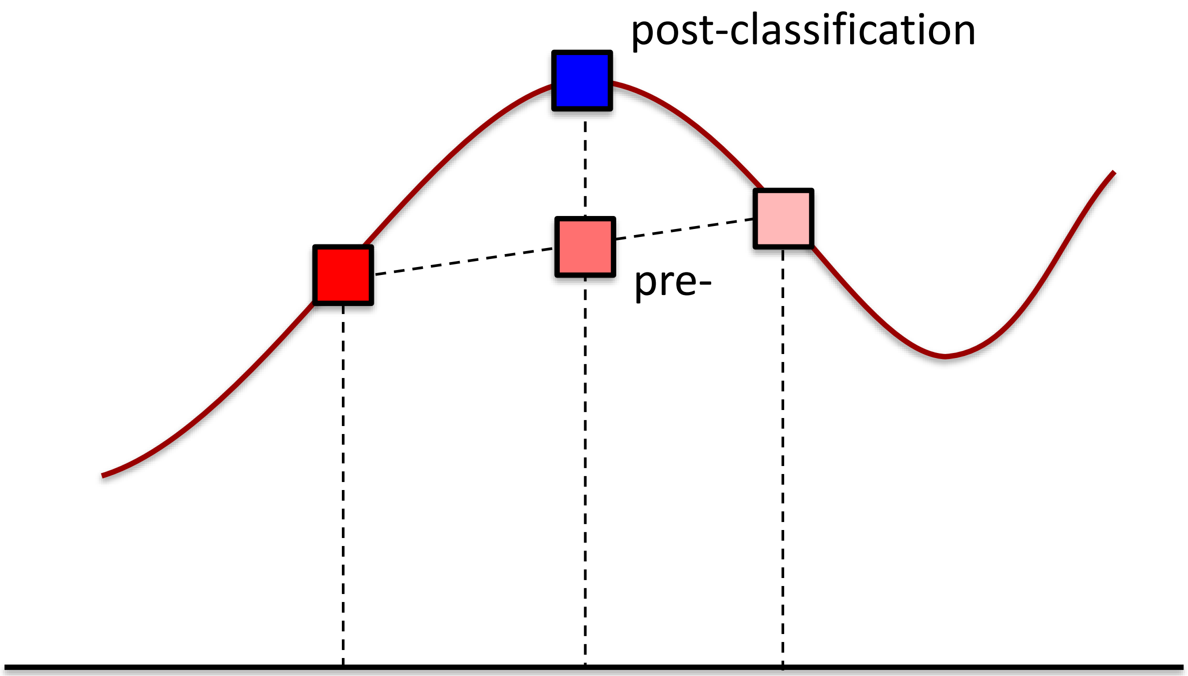
Mostly accurate isolation of all material boundaries

Gordon Kindlmann

Type of Classification



Pre- vs. Post-classification

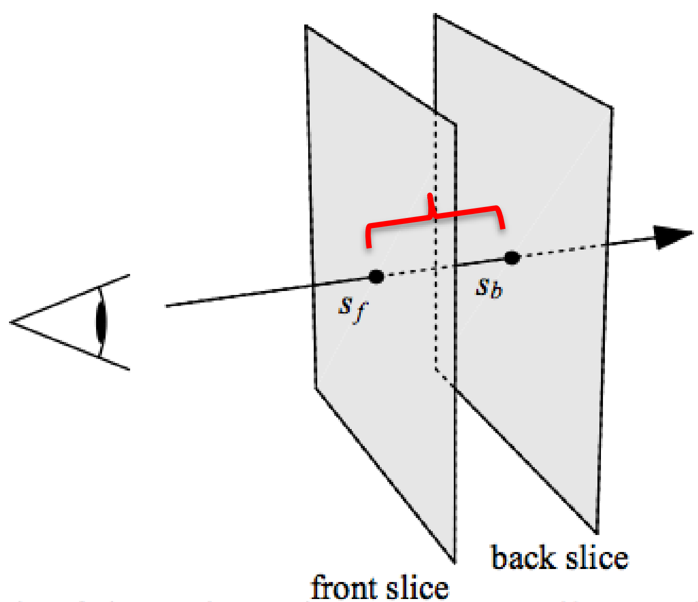


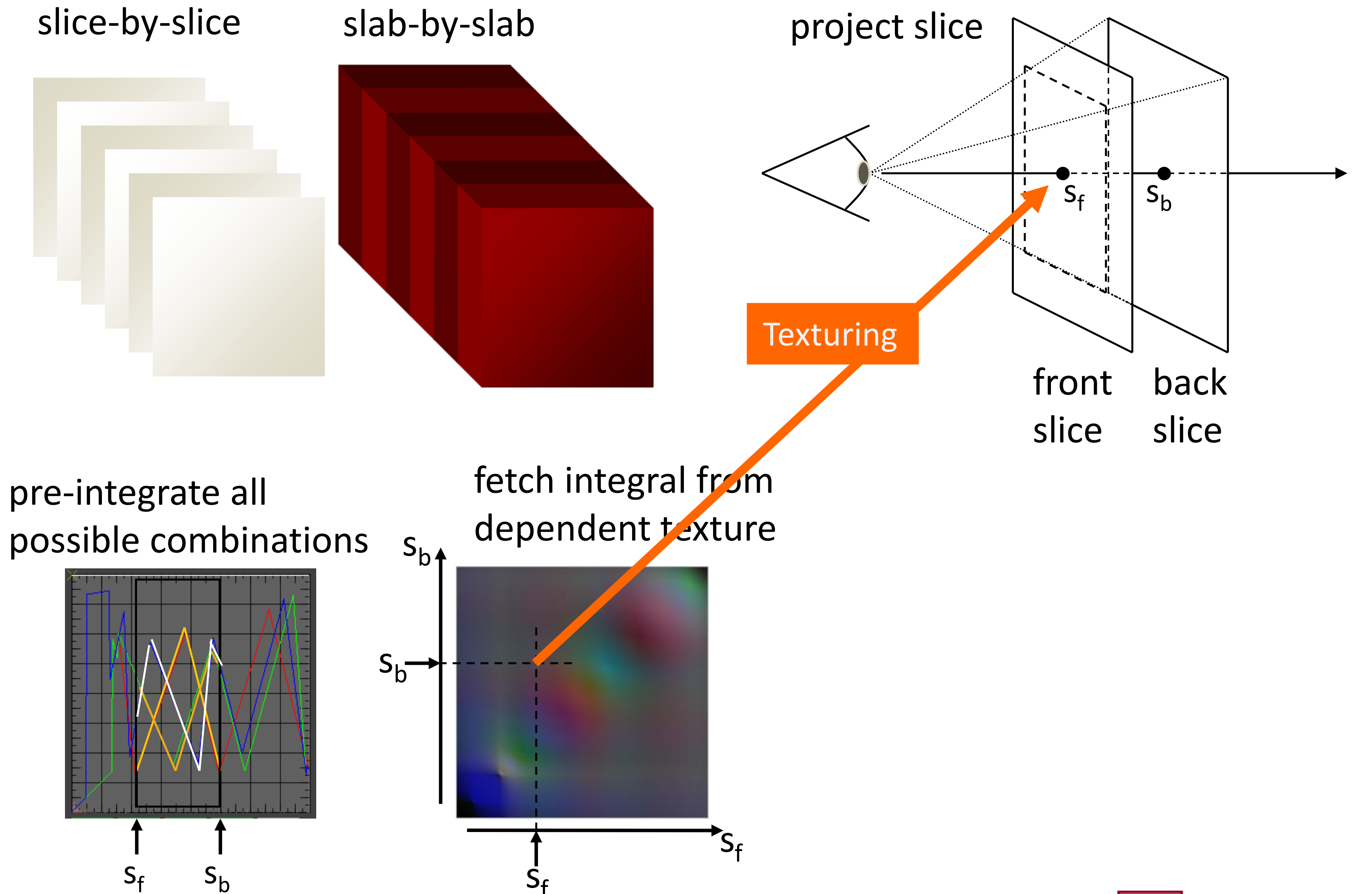
Pre- vs. Post-classification

- Pre-
  - Classification can be done only once = fast
  - Low quality
- Post-
  - Classification must be done per sample = slow
  - High-quality
- Similar to Gouraud vs. Phong shading!

Pre-integrated Volume Rendering

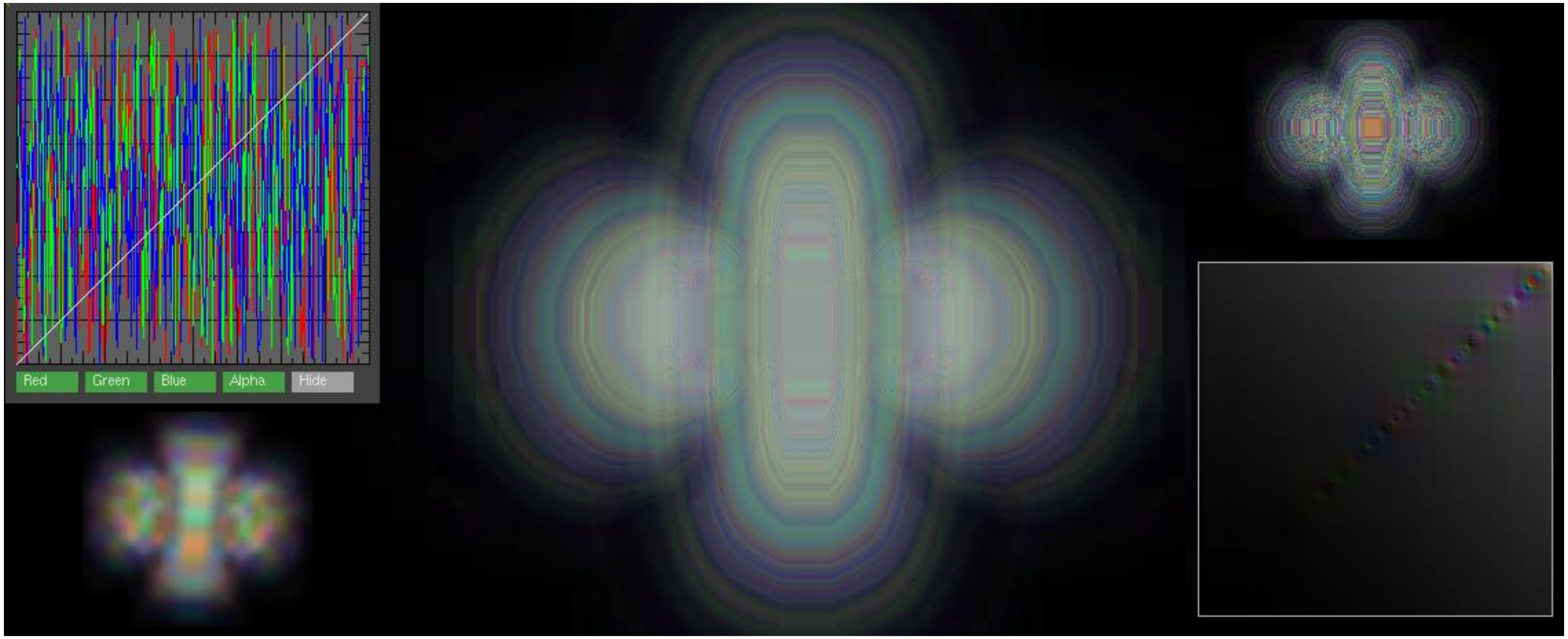
- Idea
  - Linear interpolation of scalar values within a slap
  - Constant length of a slap : L
  - Pre-compute integral along a ray in a slab



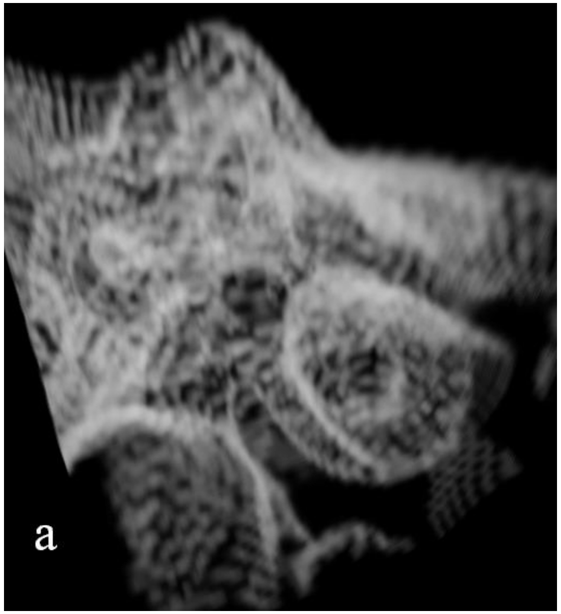

slice-by-slice
slab-by-slab
project slice
s_f
s_b
front slice
back slice
Texturing
pre-integrate all possible combinations
fetch integral from dependent texture
s_b
s_b
s_f
s_f
s_f
s_b

Pre-integrated Volume Rendering

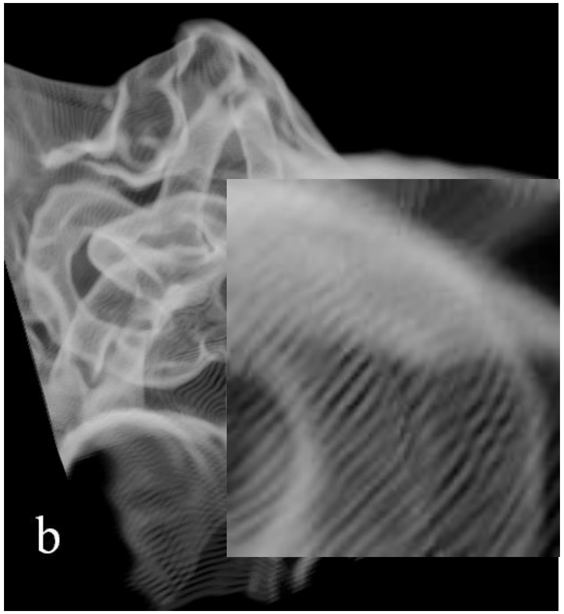
- High-quality as post-classification, fast rendering as pre-classification
- Cons : pre-integrated table must be updated when transfer function is changed = use GPU for speed up



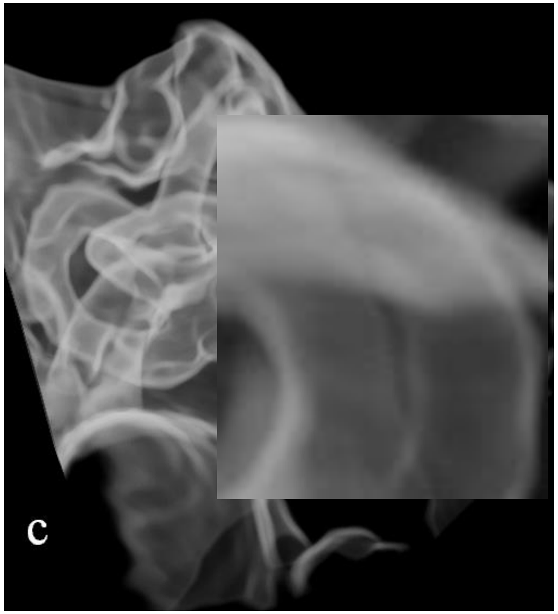
Klaus Engel



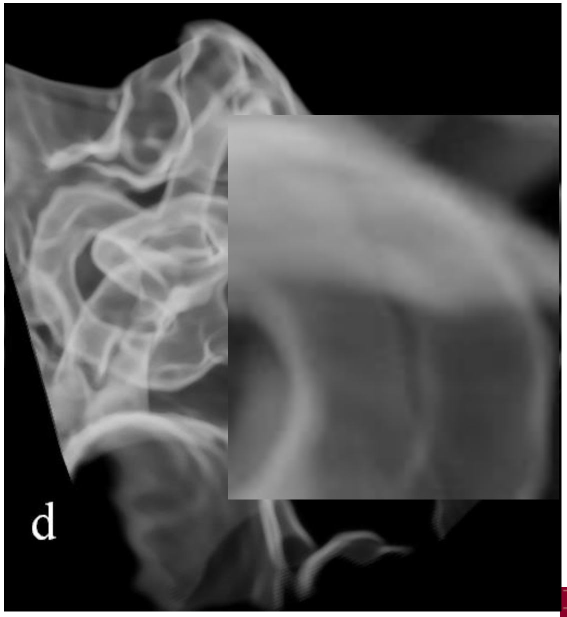
128 slices
pre-
classification



128 slices
post-
classification



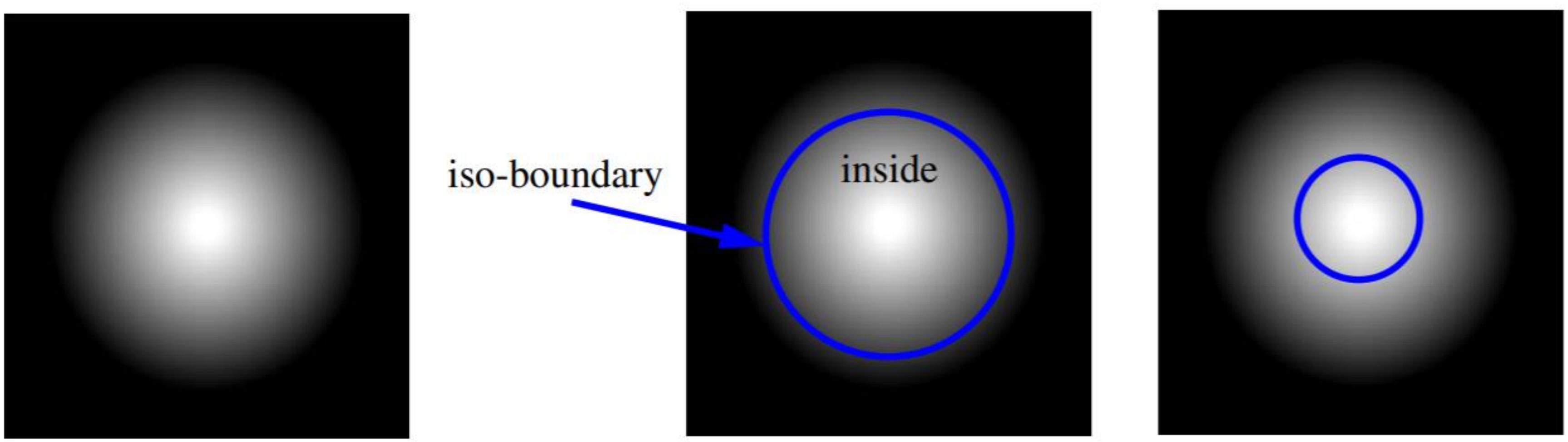
284 slices
post-
classification



128 slices
pre-integrated

Isosurface Volume Rendering

- Voxels are separated as inside or outside
  - Iso-value is where the contour/boundary located



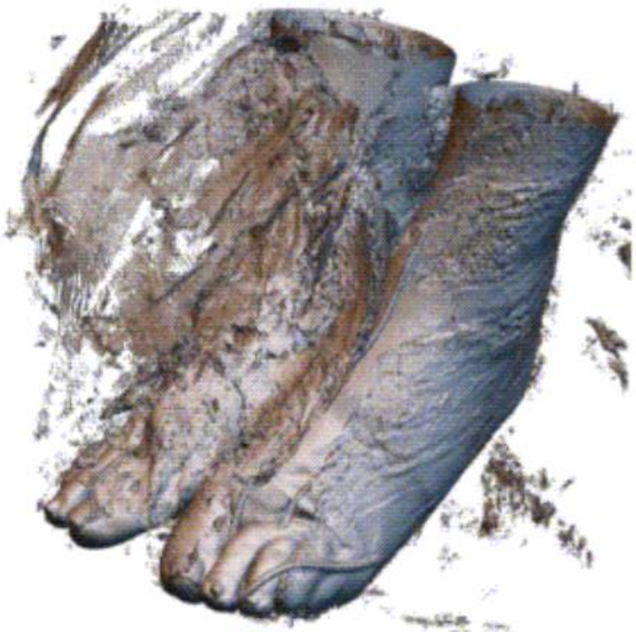
cross-section of a smooth sphere

iso-value = 50
will render a large sphere

iso-value = 200
will render a small sphere

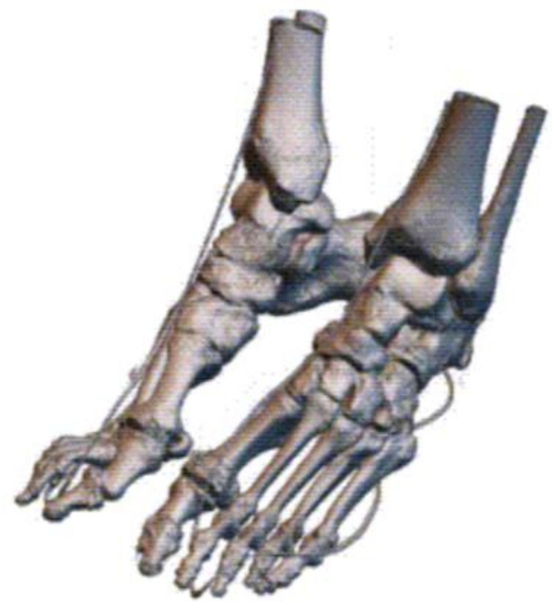
Image Credit: Klaus Mueller

Example of Isosurface Volren



iso-value = 30

iso-value = 80



iso-value = 200

Image Credit: Klaus Mueller

Algorithm

- Marching along ray until reach zero-crossing

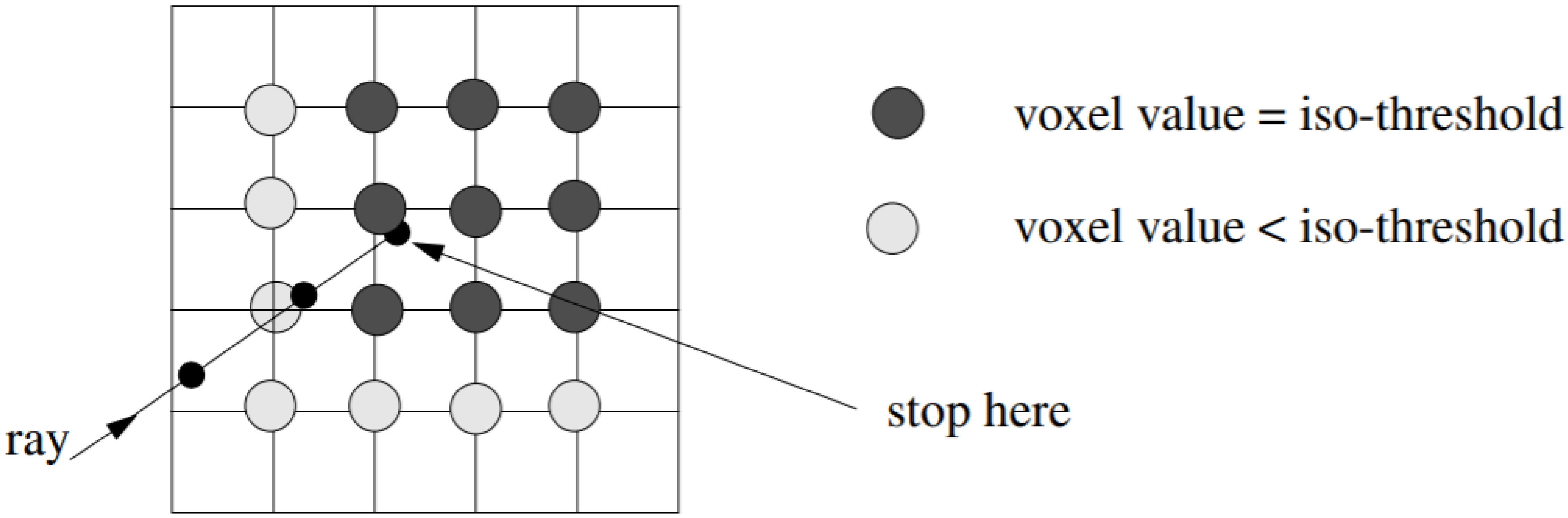
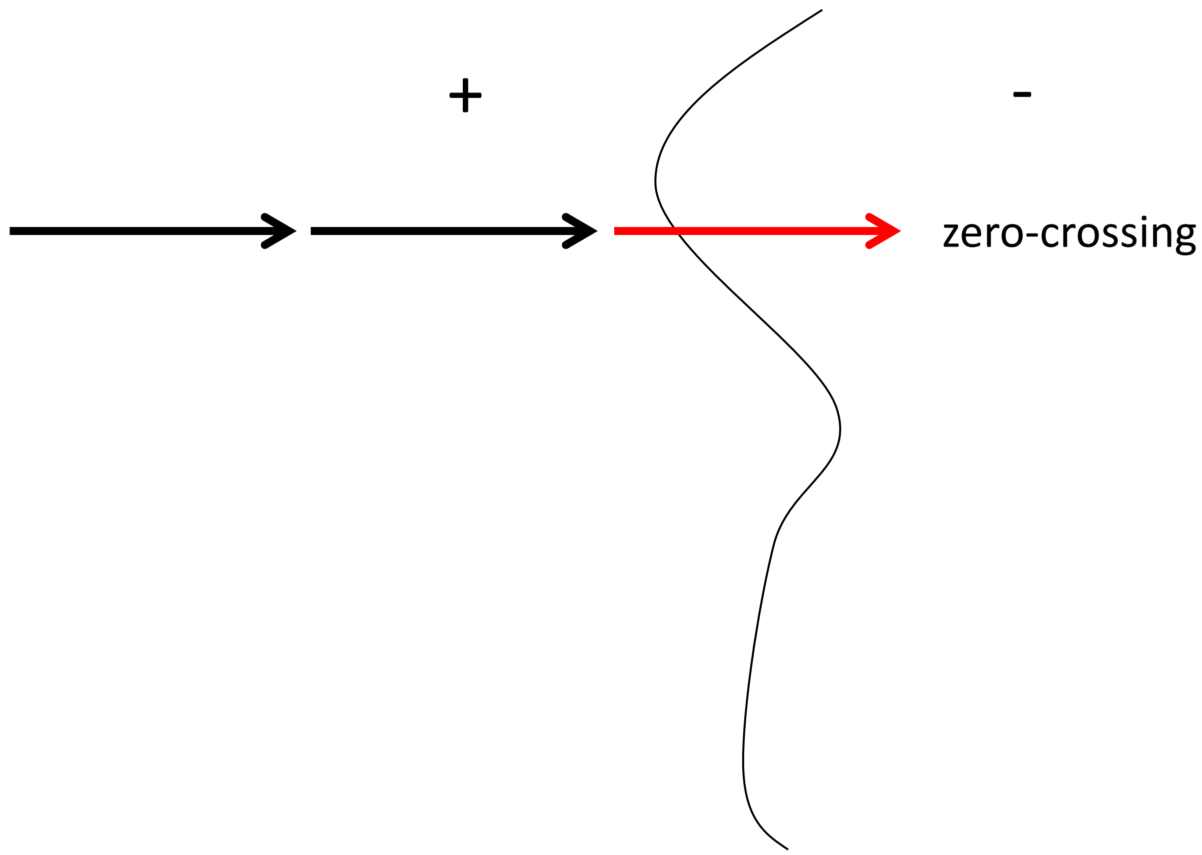


Image Credit: Klaus Mueller

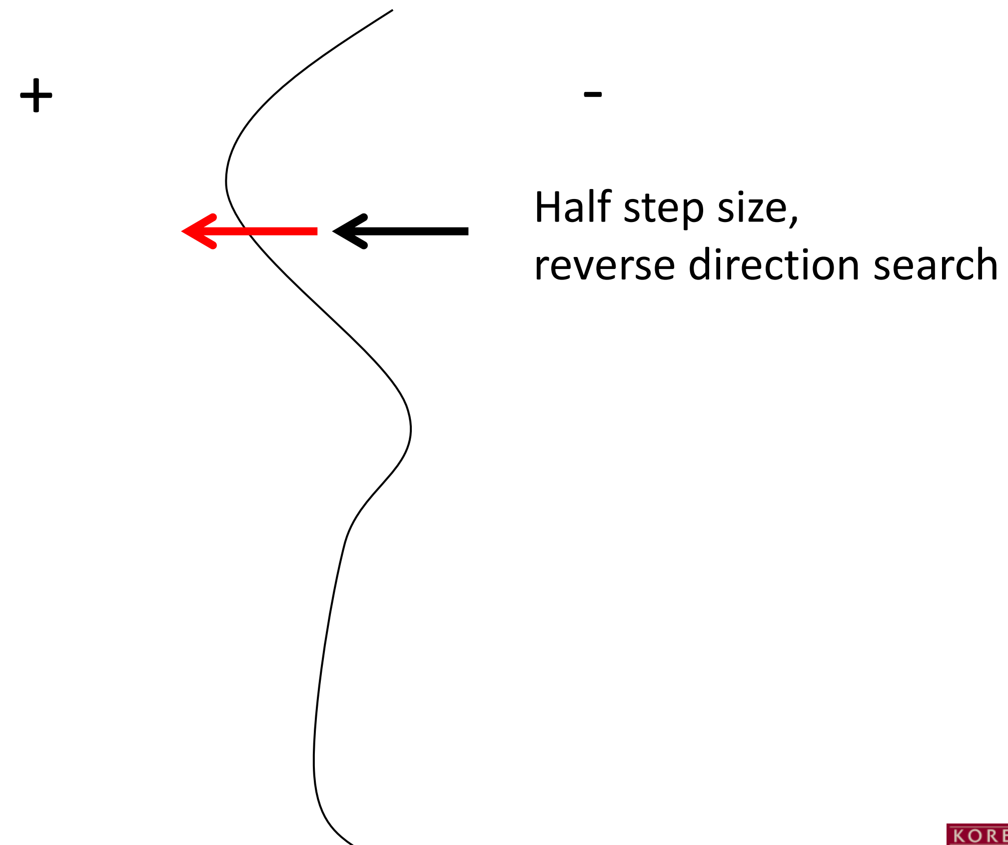
Algorithm

- Iterative root finding



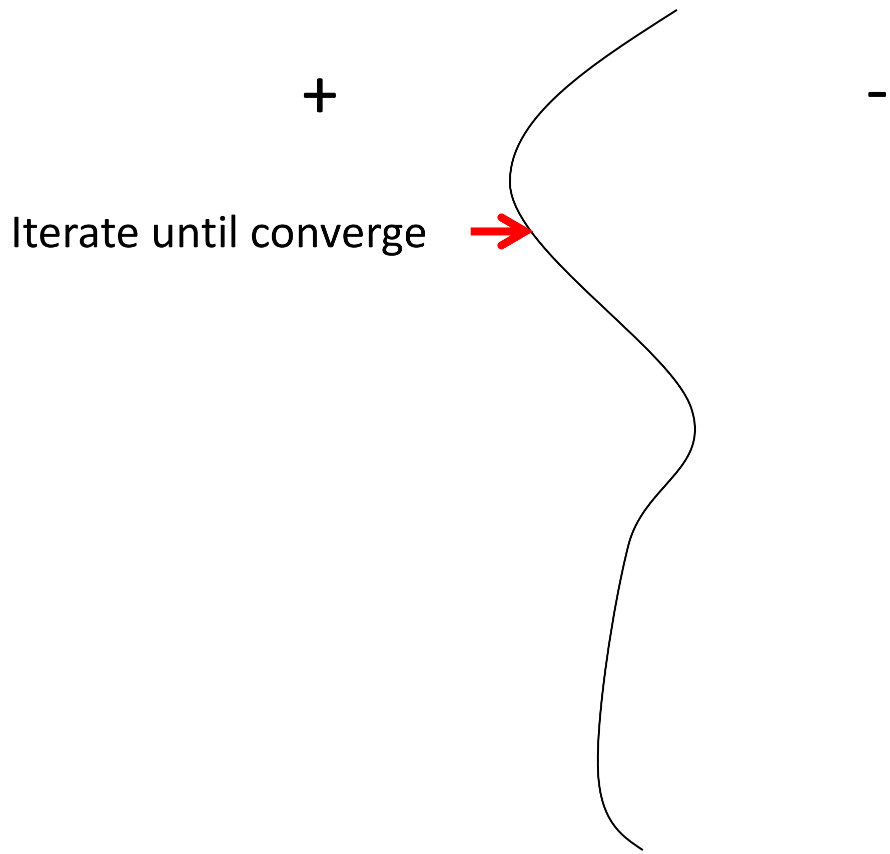
Algorithm

- Iterative root finding



Algorithm

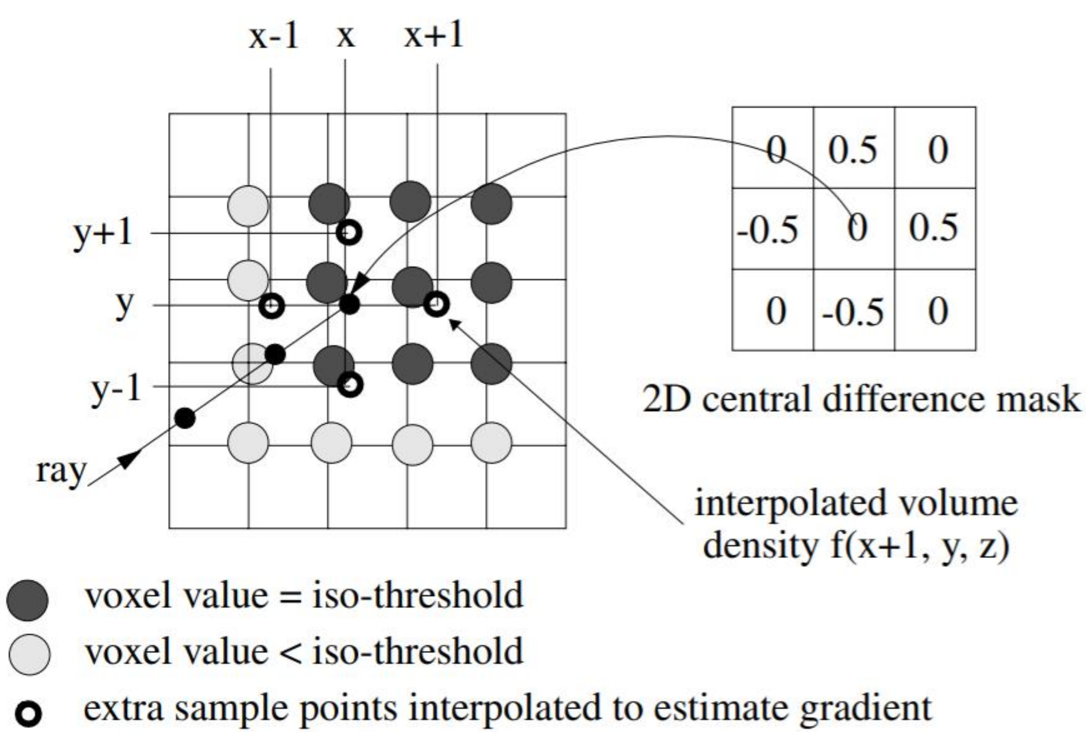
- Iterative root finding



Algorithm

- Compute gradient vector

$$g_x = \frac{f(x-1, y, z) - f(x+1, y, z)}{2} \qquad g_y = \frac{f(x, y-1, z) - f(x, y+1, z)}{2} \qquad g_z = \frac{f(x, y, z-1) - f(x, y, z+1)}{2}$$



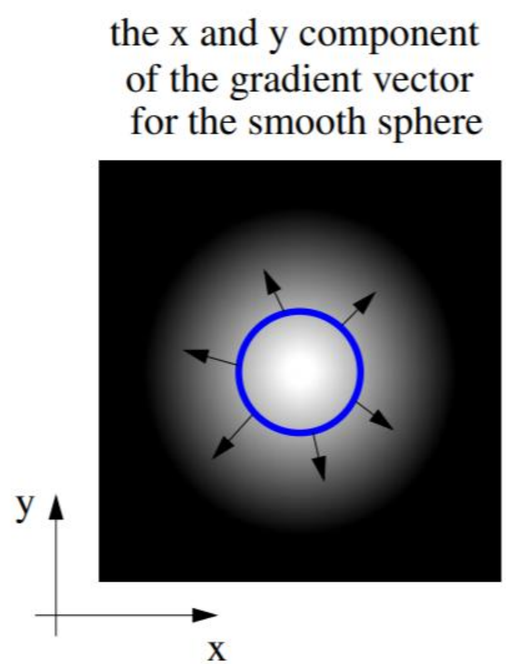
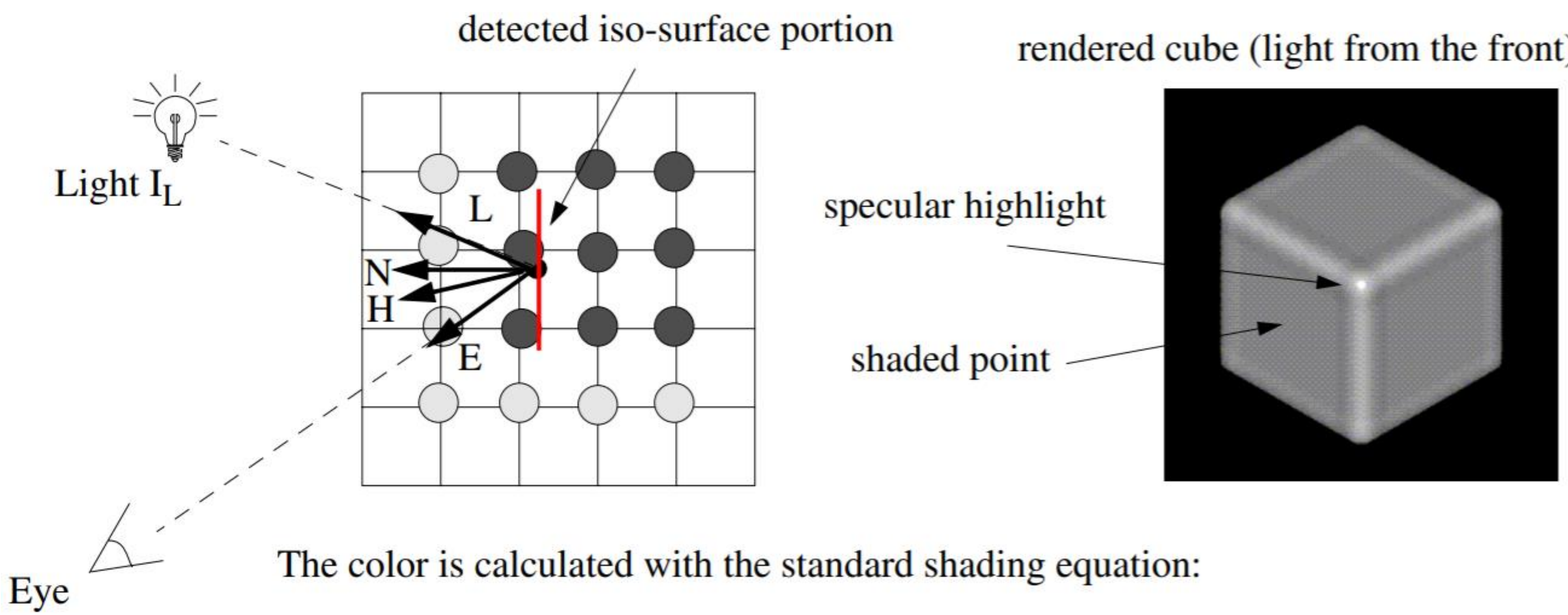


Image Credit: Klaus Mueller

Algorithm

- Use any illumination model (e.g., Phong)



The color is calculated with the standard shading equation:

$$C = C_{obj}\,(k_a\, I_A + k_d\, I_L\, N \cdot L) + k_s\, I_L\, (H \cdot N)^{ns}$$

Image Credit: Klaus Mueller

KOREA UNIVERSITY

Questions?

KOREA UNIVERSITY