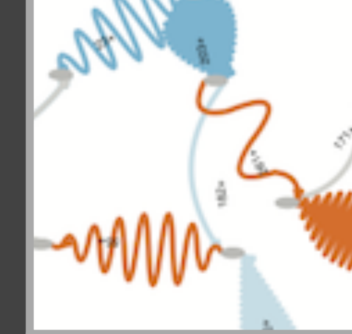
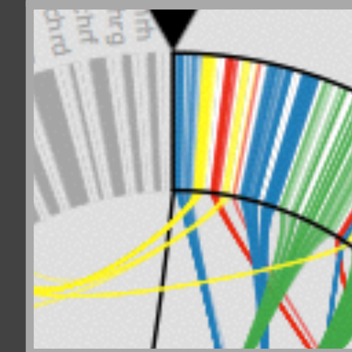
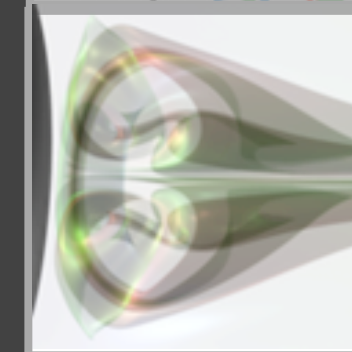
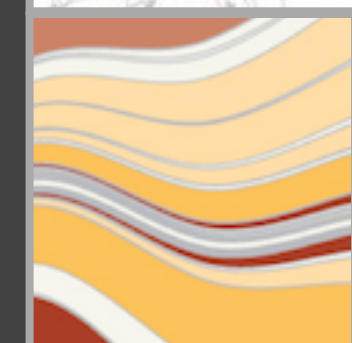
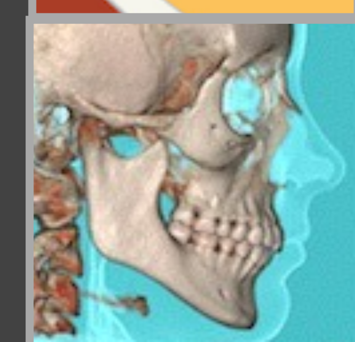
cs6630 | November 14 2014

TRANSFER FUNCTIONS

Alex Bigelow
University of Utah

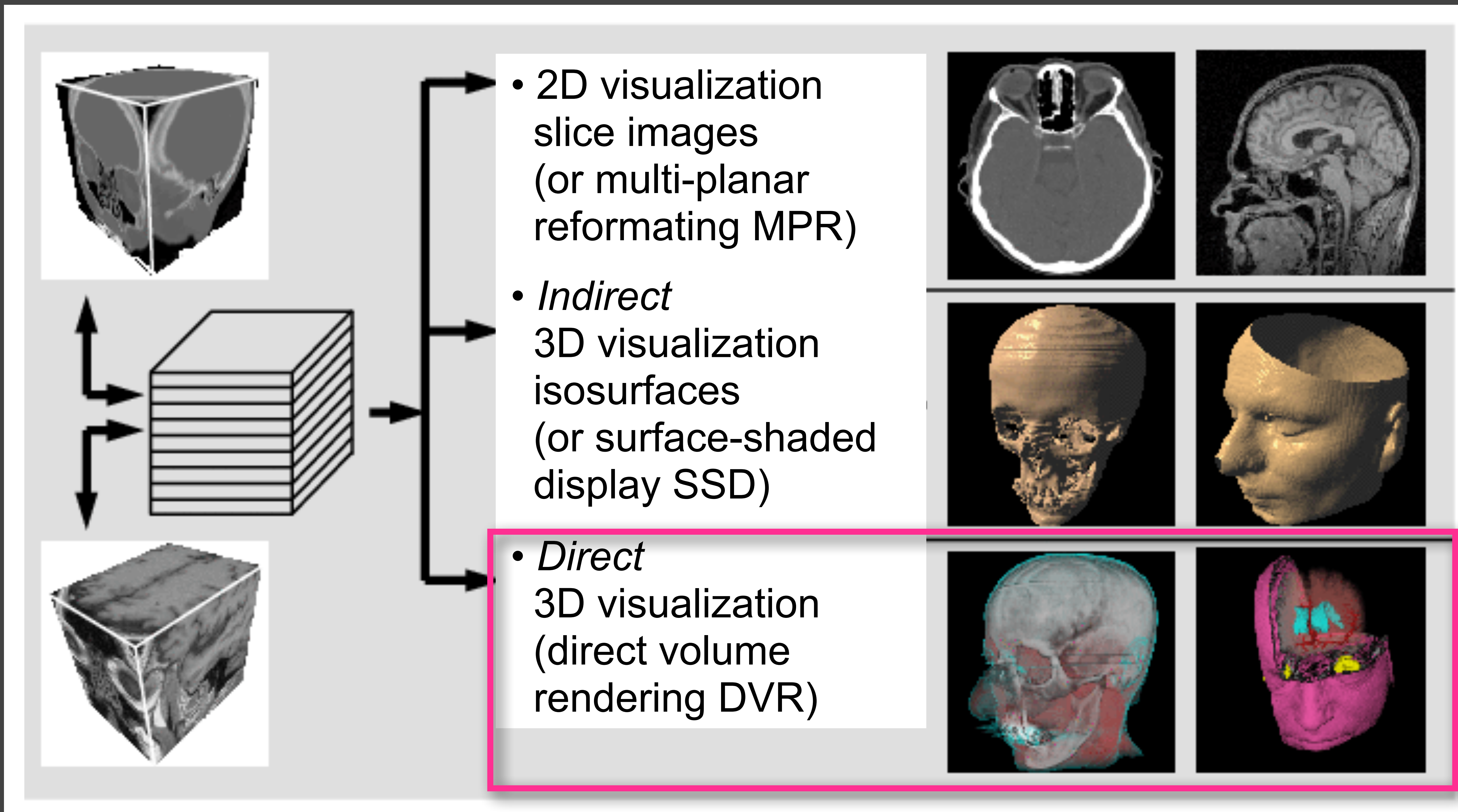
cs6630 | November 13 2014

TRANSFER FUNCTIONS

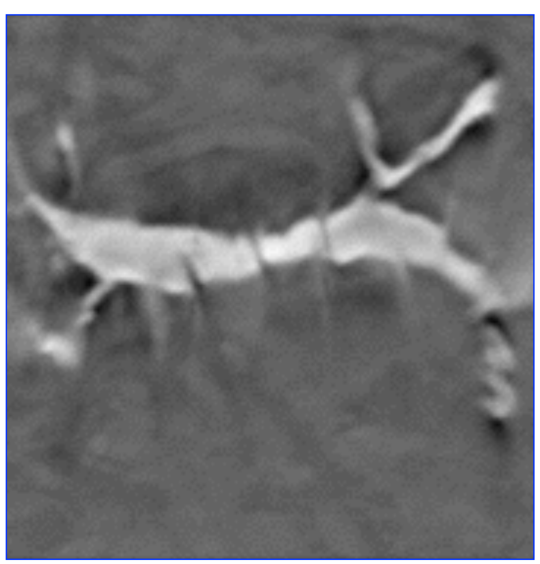
Alex Bigelow
University of Utah

slide acknowledgements:
Miriah Meyer, University of Utah
Torsten Moller, Simon Frasier University
Josh Levine, Clemson
Markus Hadwig, KAUST

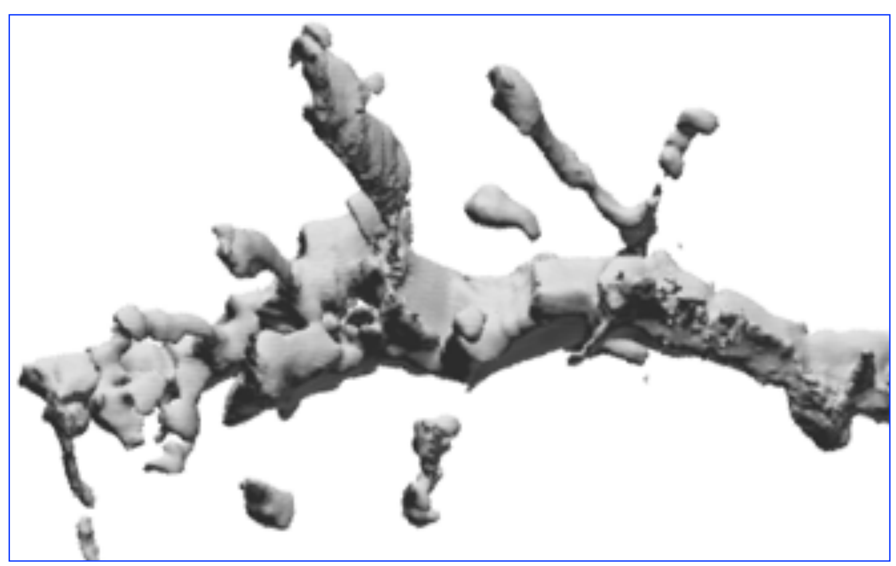
last time . . .


• 2D visualization
slice images
(or multi-planar
reformating MPR)
• *Indirect*
3D visualization
isosurfaces
(or surface-shaded
display SSD)
• *Direct*
3D visualization
(direct volume
rendering DVR)

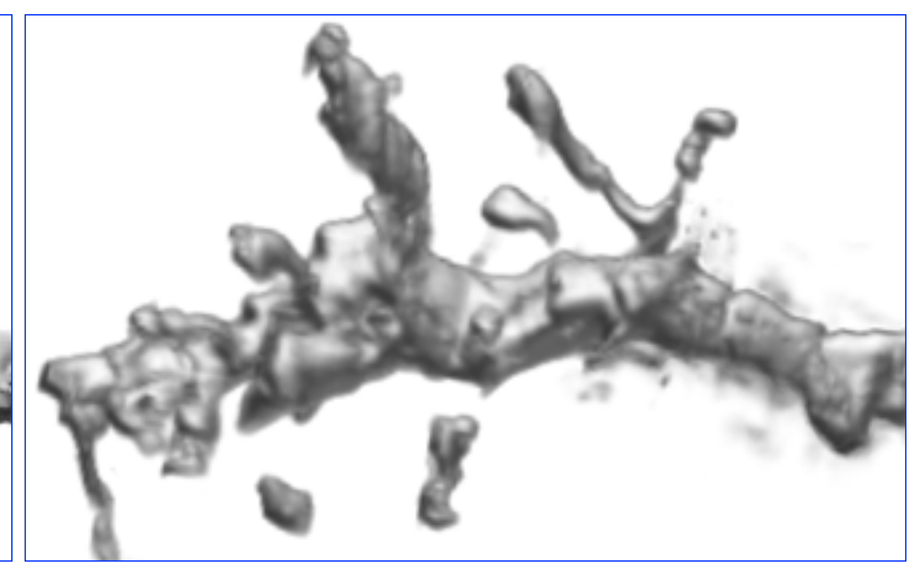
- Isosurfacing is "binary"
  - What about points inside isosurface?
  - How does each voxel contributes to image?
- Is a hard, distinct boundary necessarily appropriate for the visualization task?



Slice

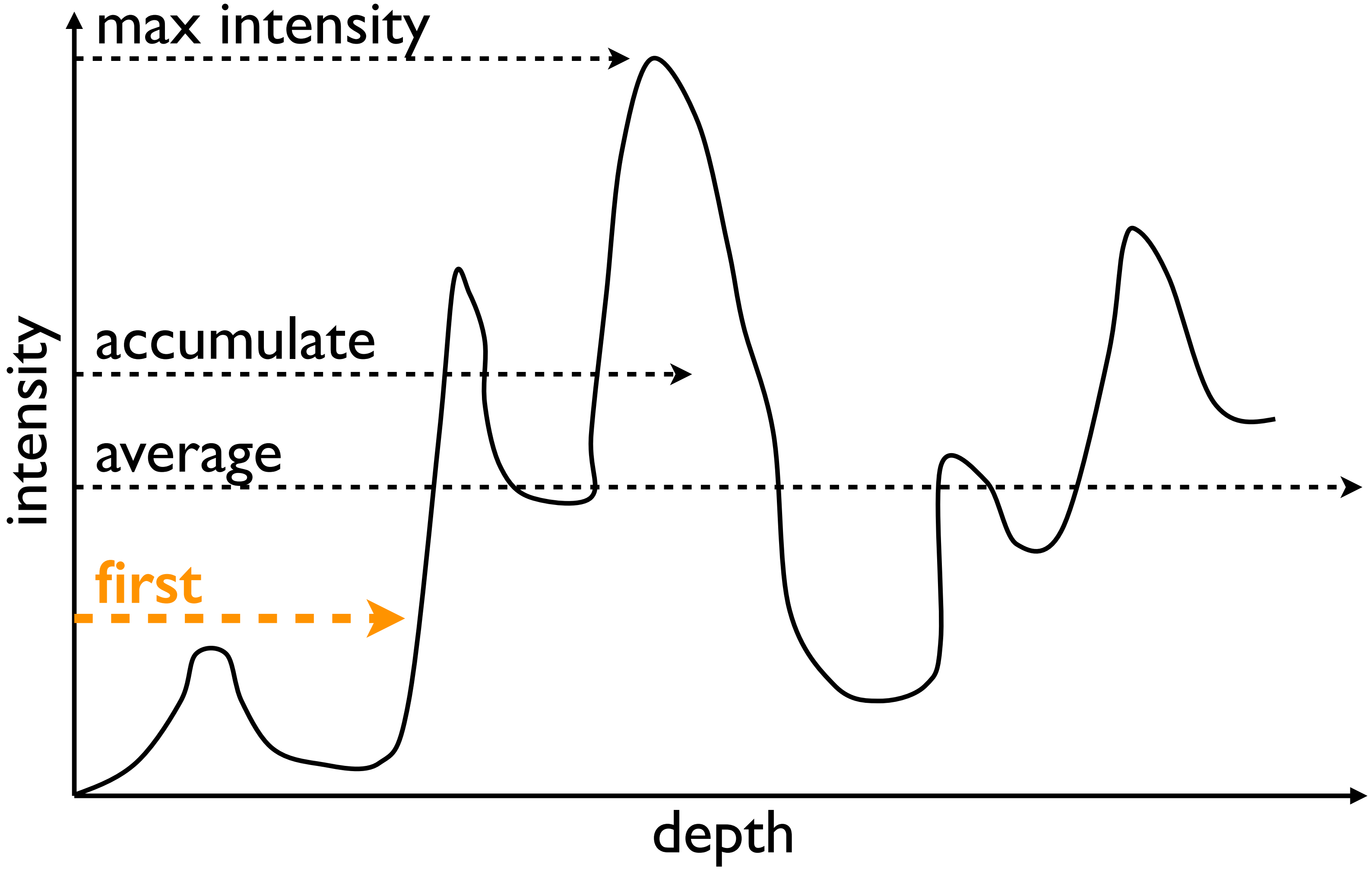


Isosurface



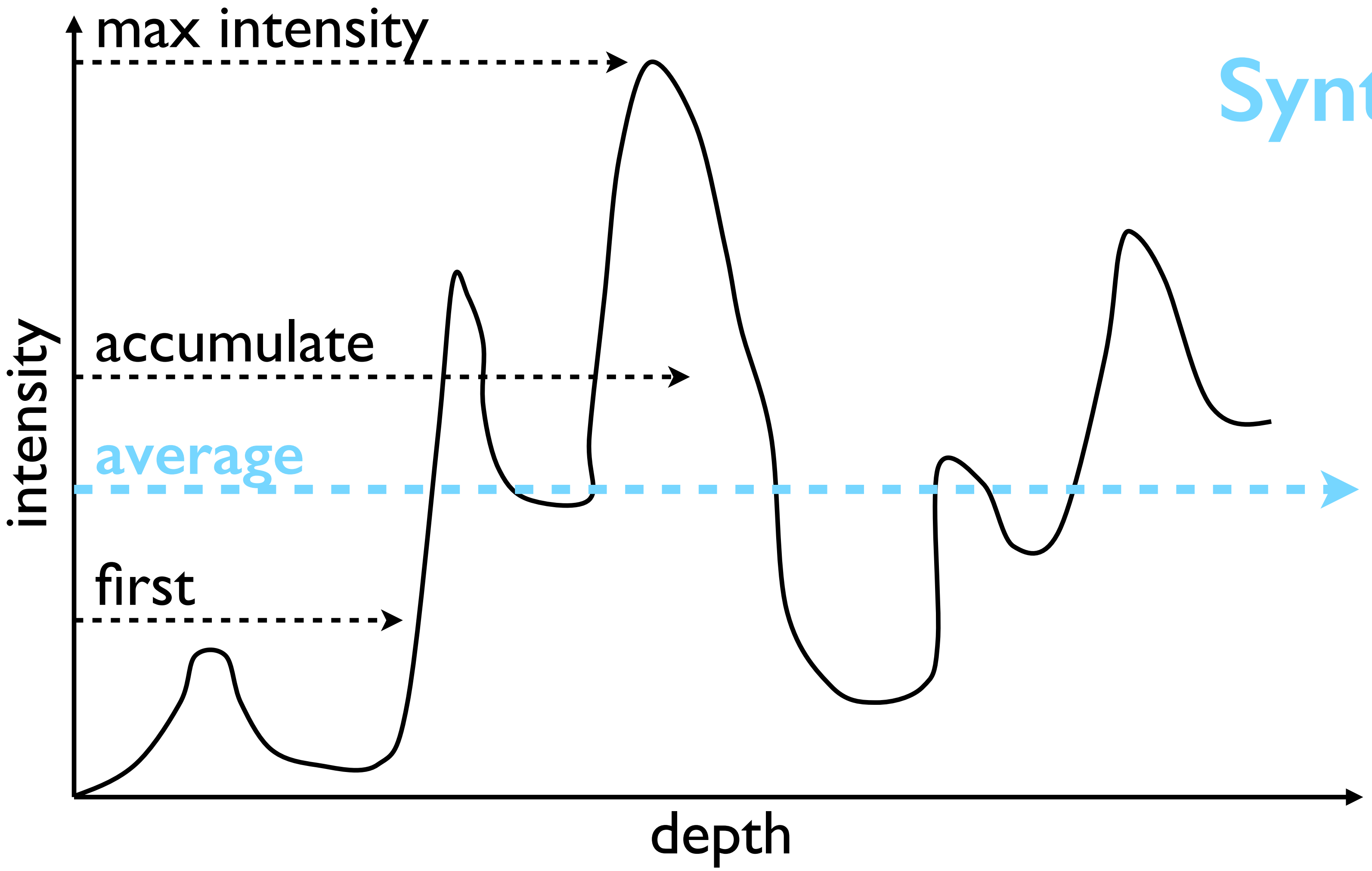
Volume Rendering

Pixel Compositing Schemes

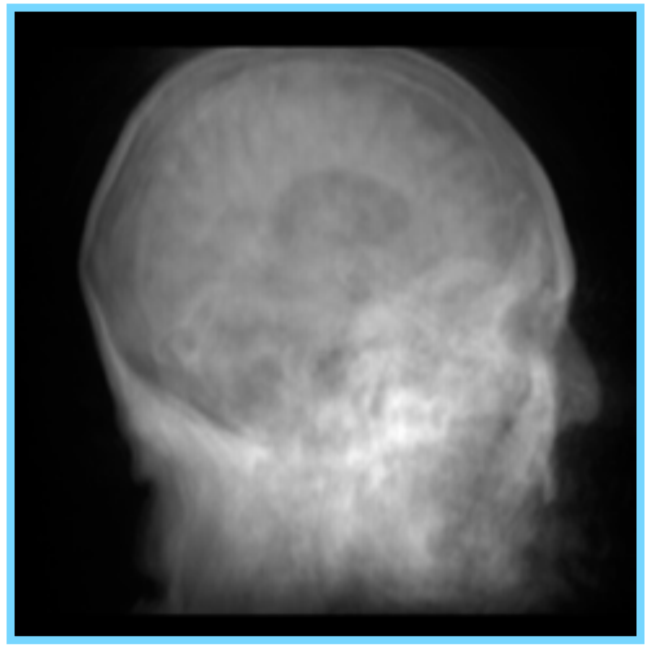


Exact Isosurface

Pixel Compositing Schemes



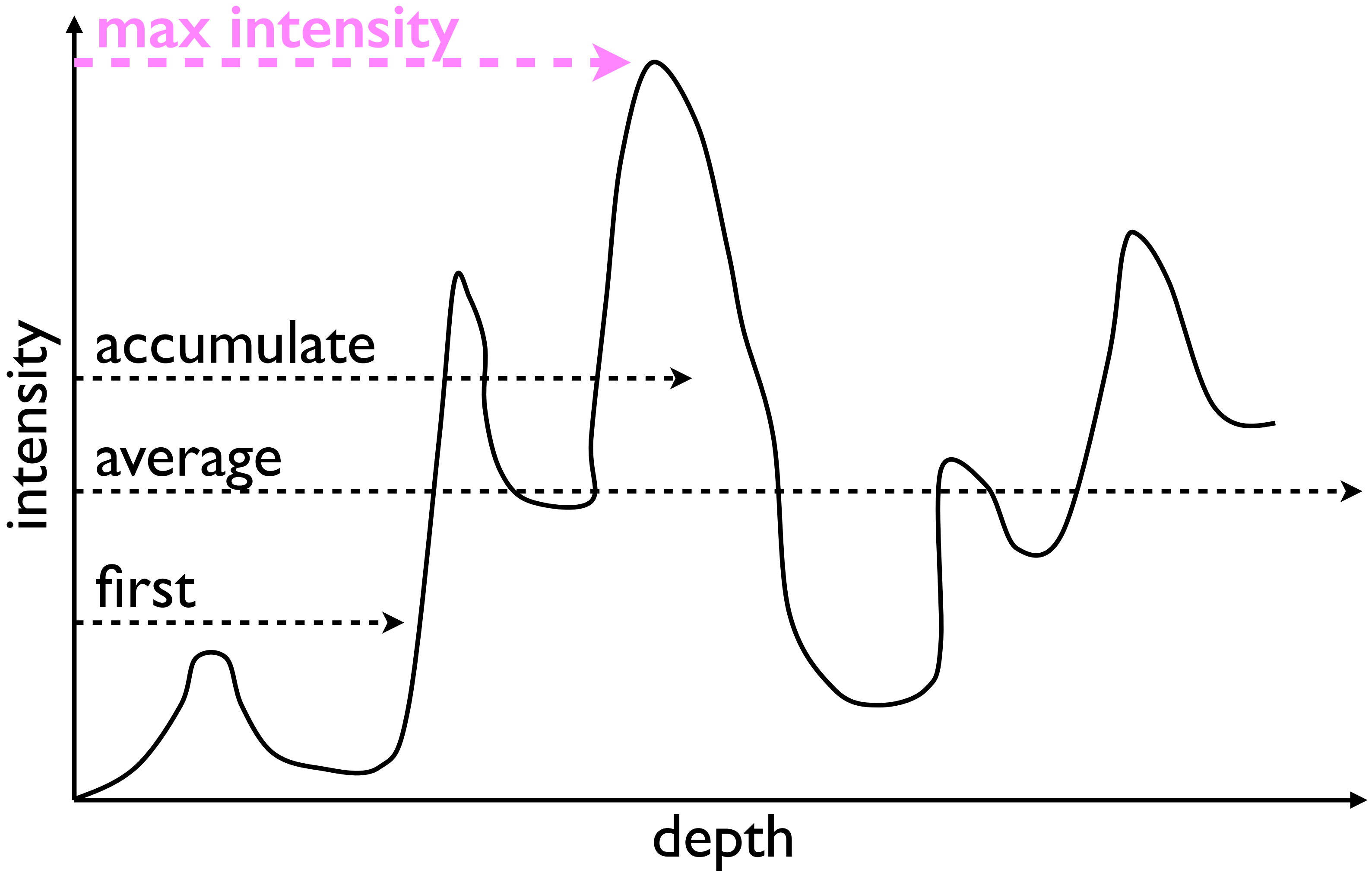
Synthetic Reprojection

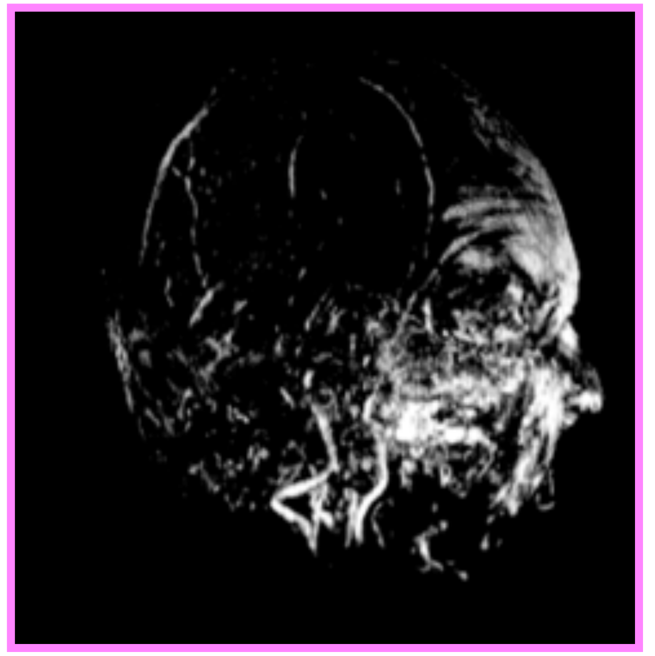


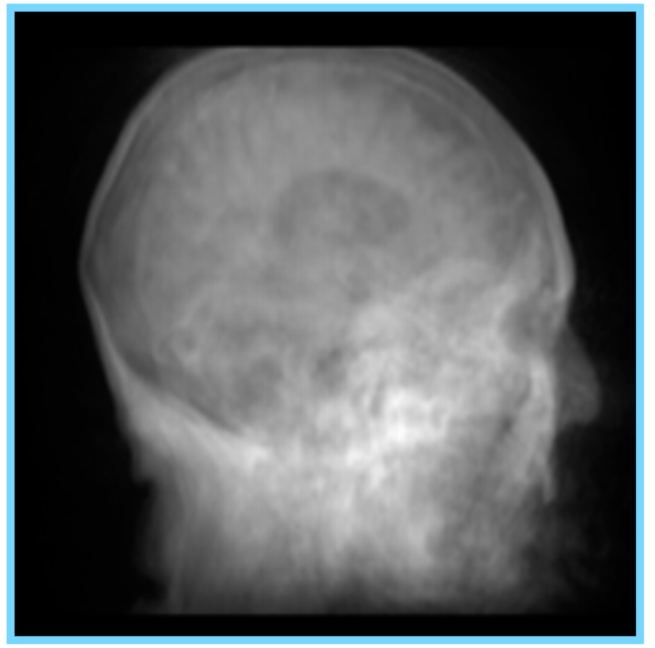
Similar to X-rays

maximum intensity projection (MIP)

Pixel Compositing Schemes

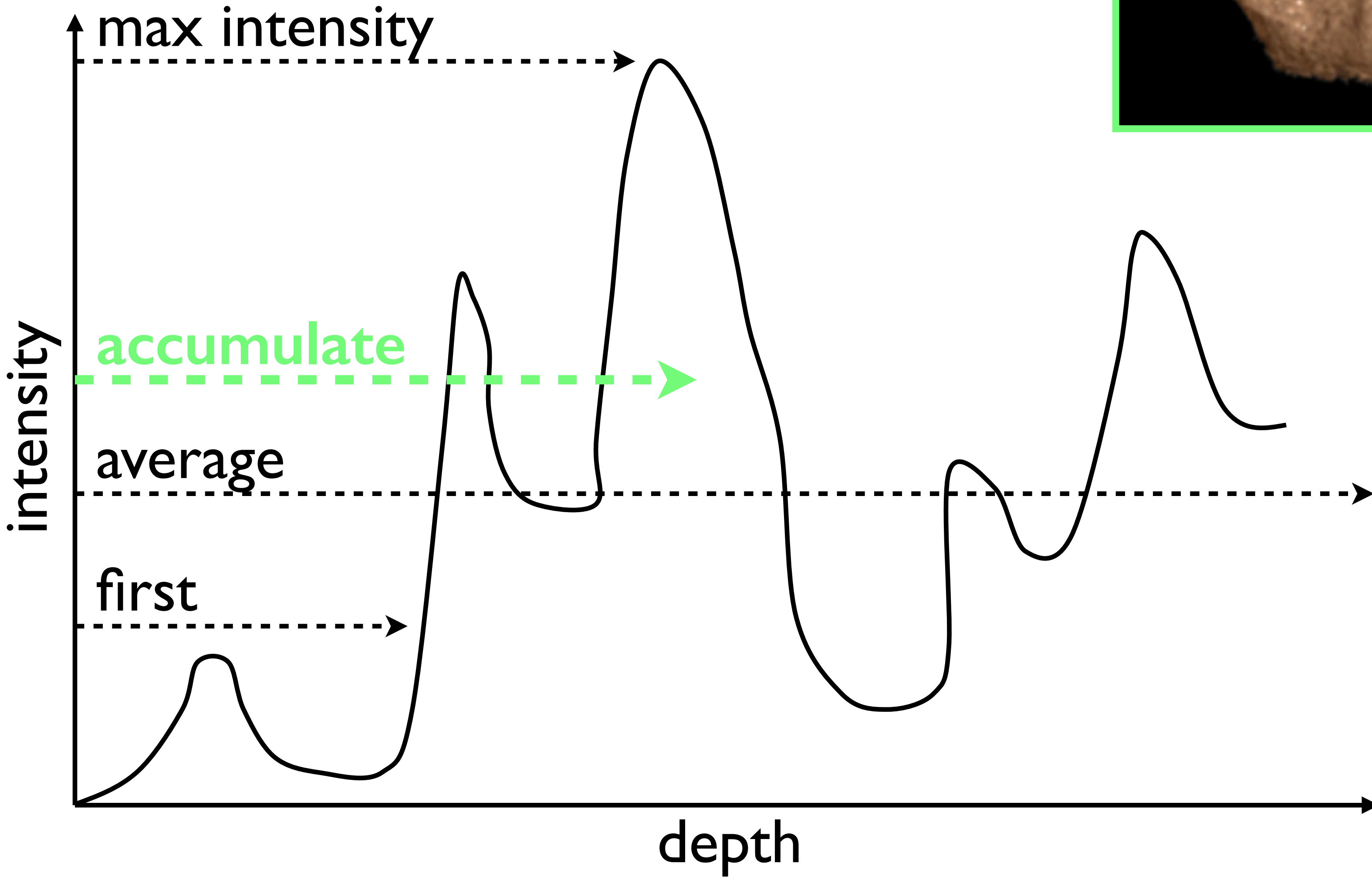


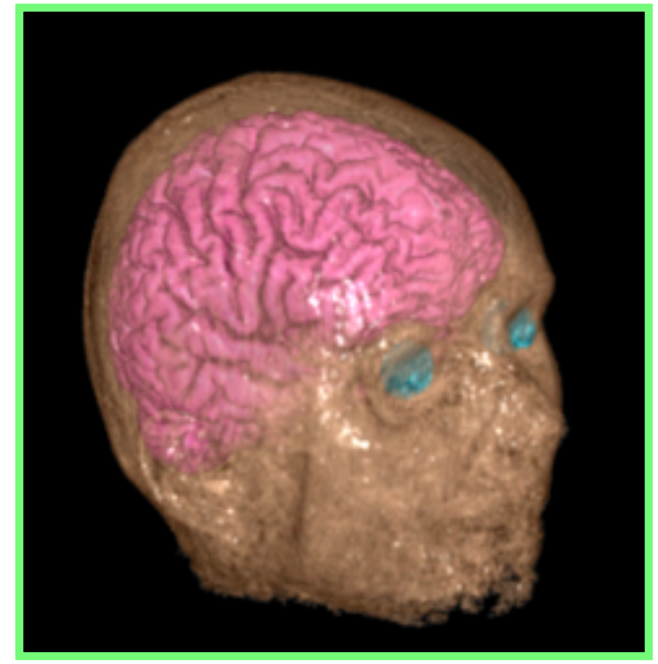


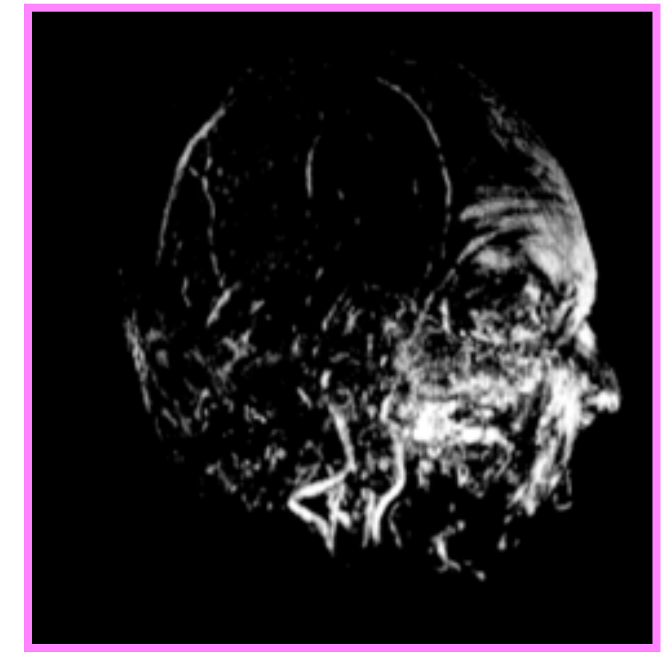


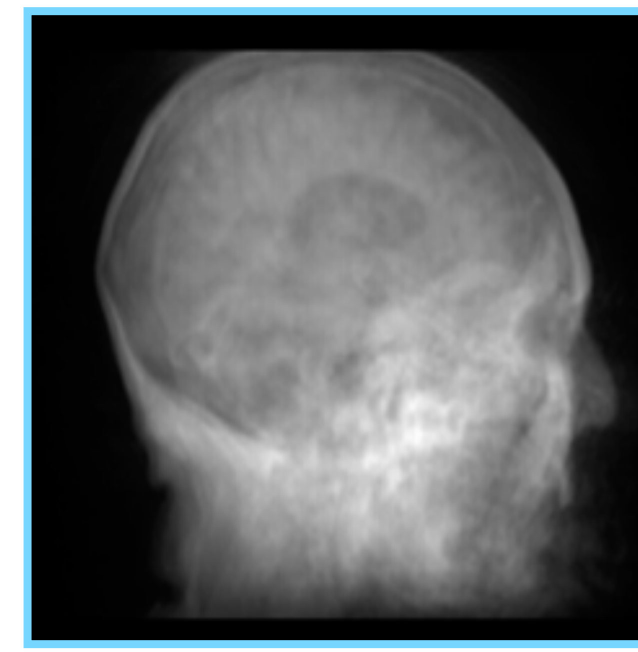
Used in PET and Magnetic Resonance Angiograms

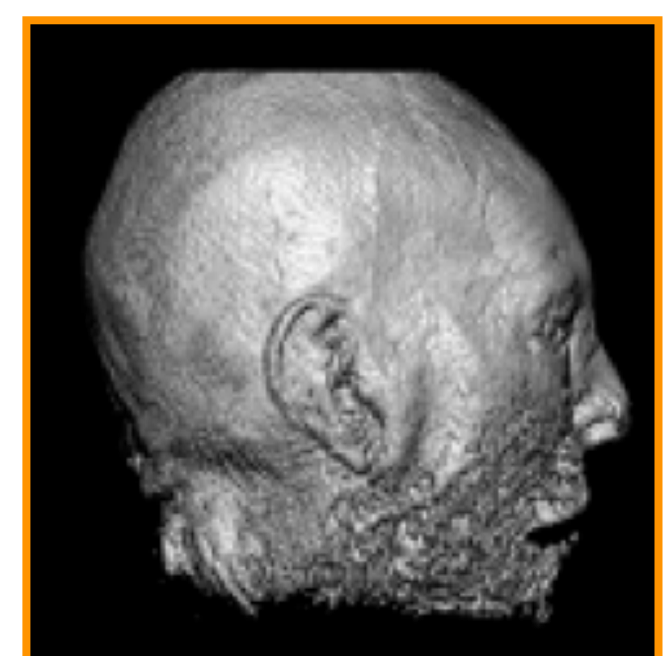
Pixel Compositing Schemes







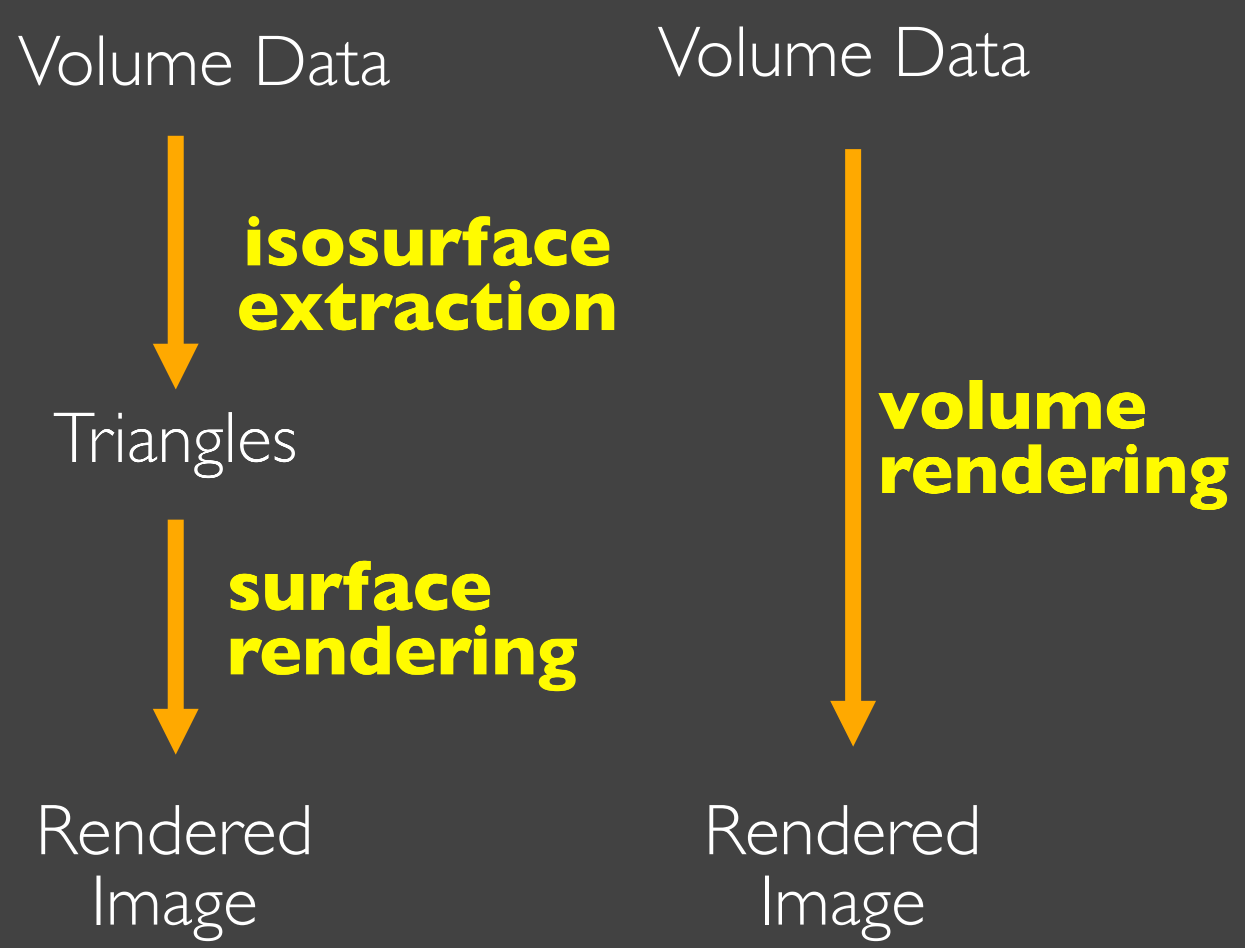




color to distinguish structures
opacity to show inside

pipelines: isosurface vs. volume rendering

no intermediate geometric structures

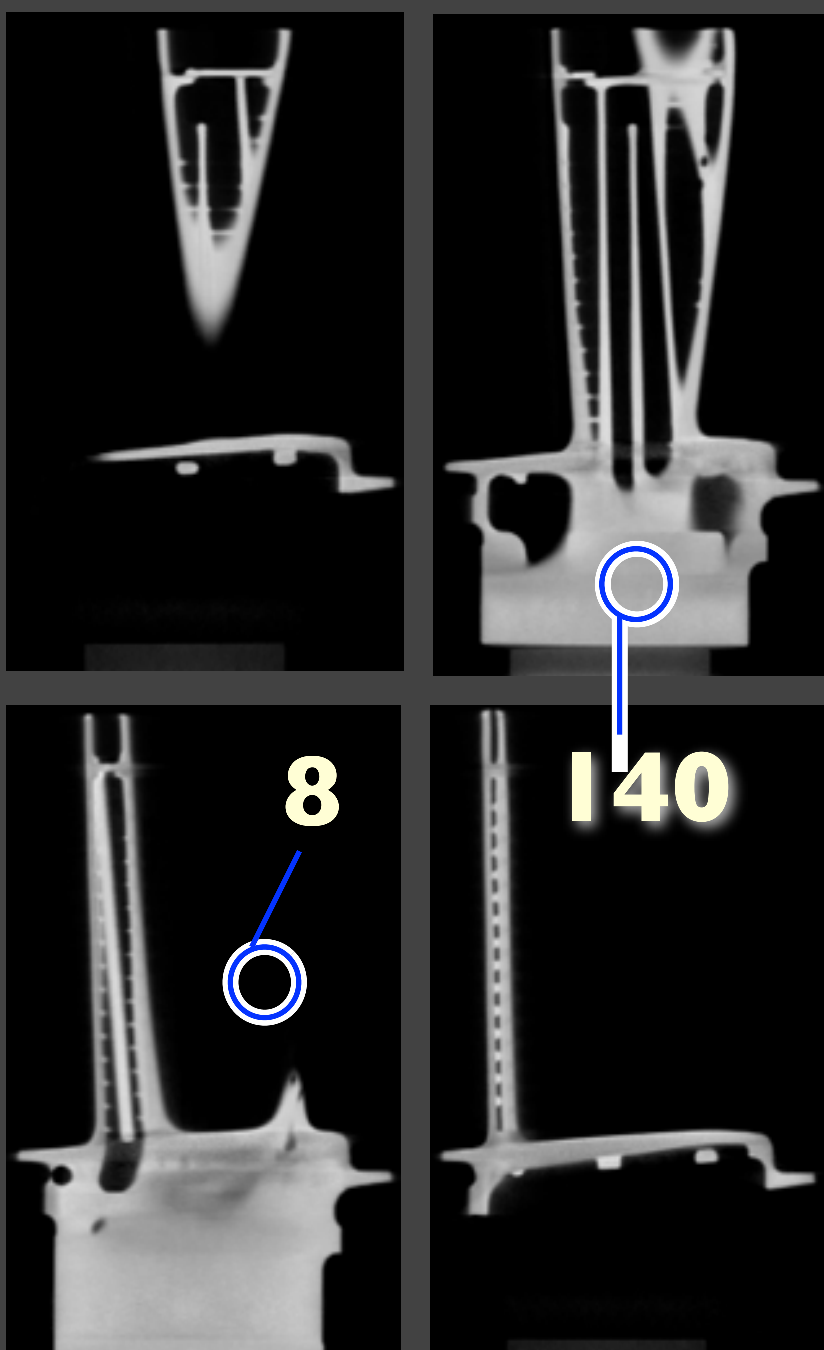


TRANSFER FUNCTIONS

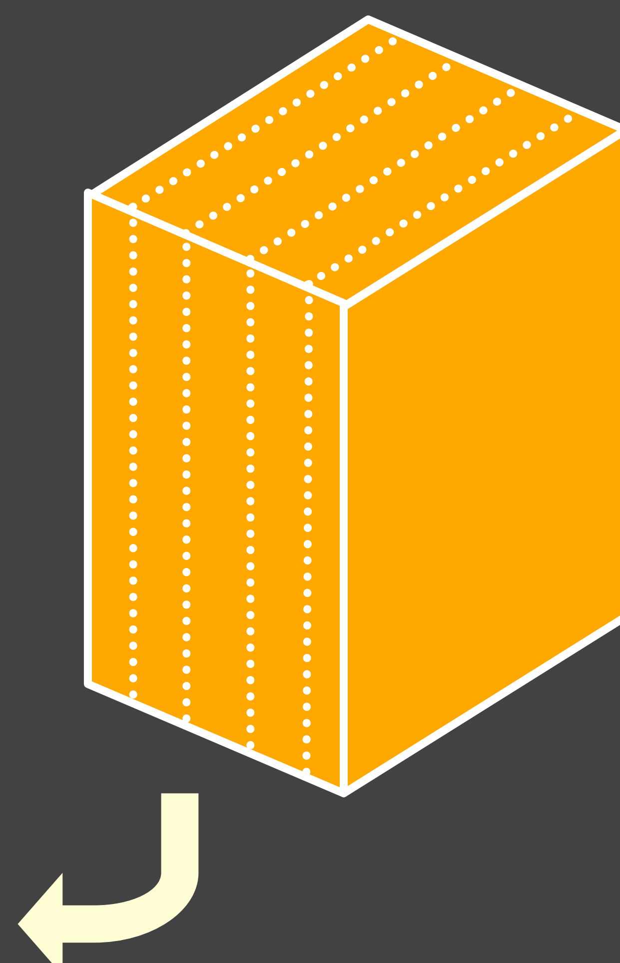
Introduction

Transfer functions make volume data visible by mapping data values to optical properties

slices:



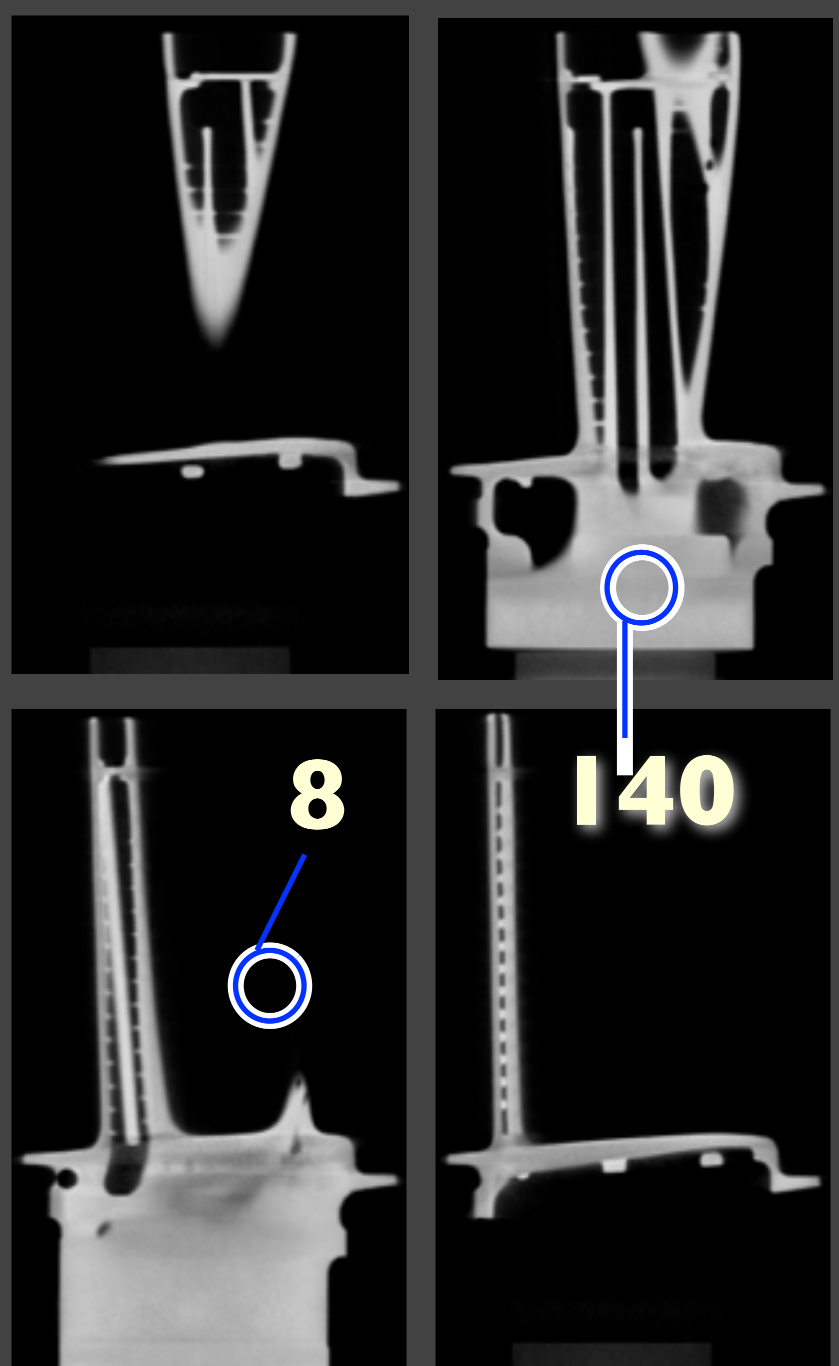
volume data:



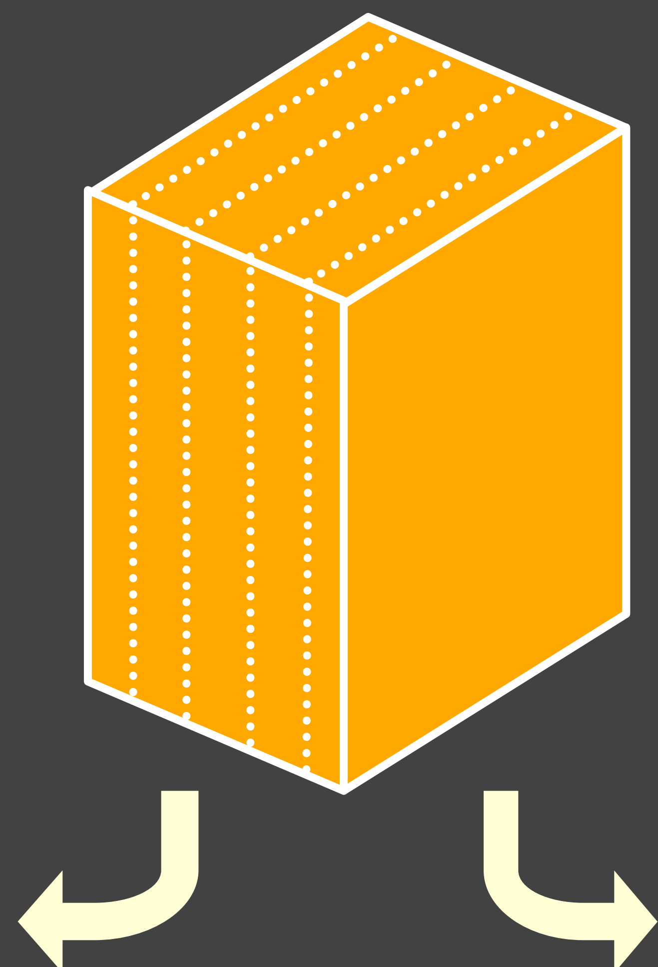
Introduction

Transfer functions make volume data visible by mapping data values to optical properties

slices:



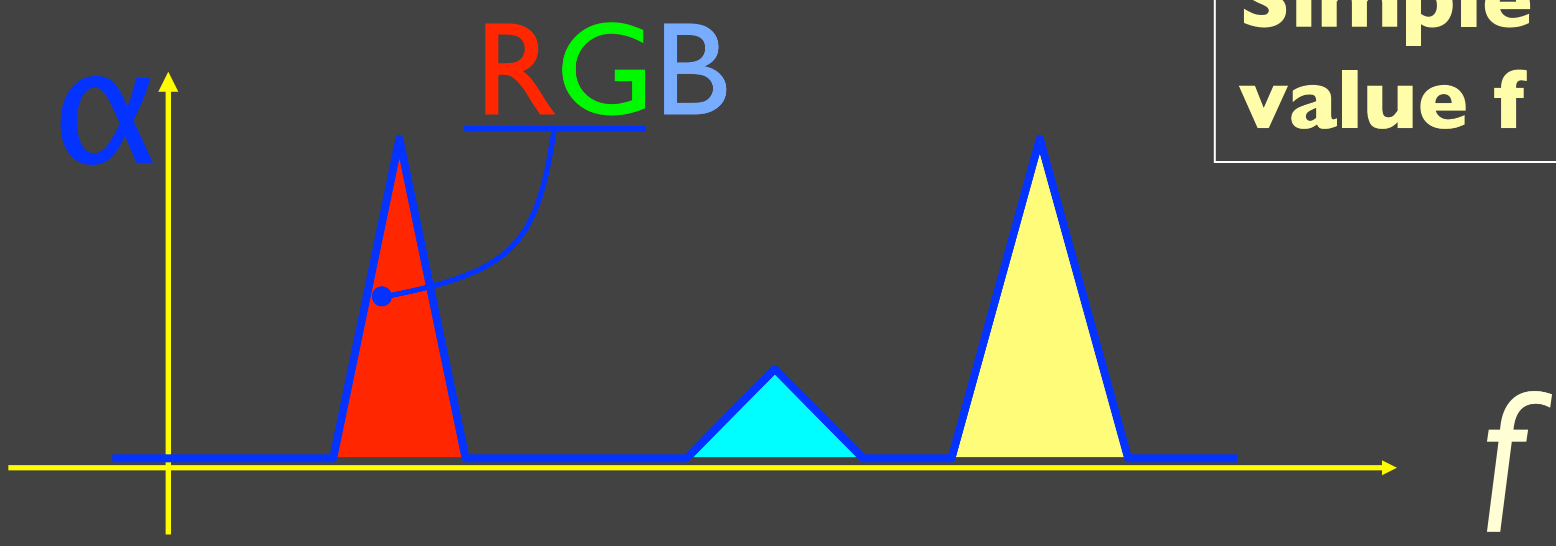
volume data:



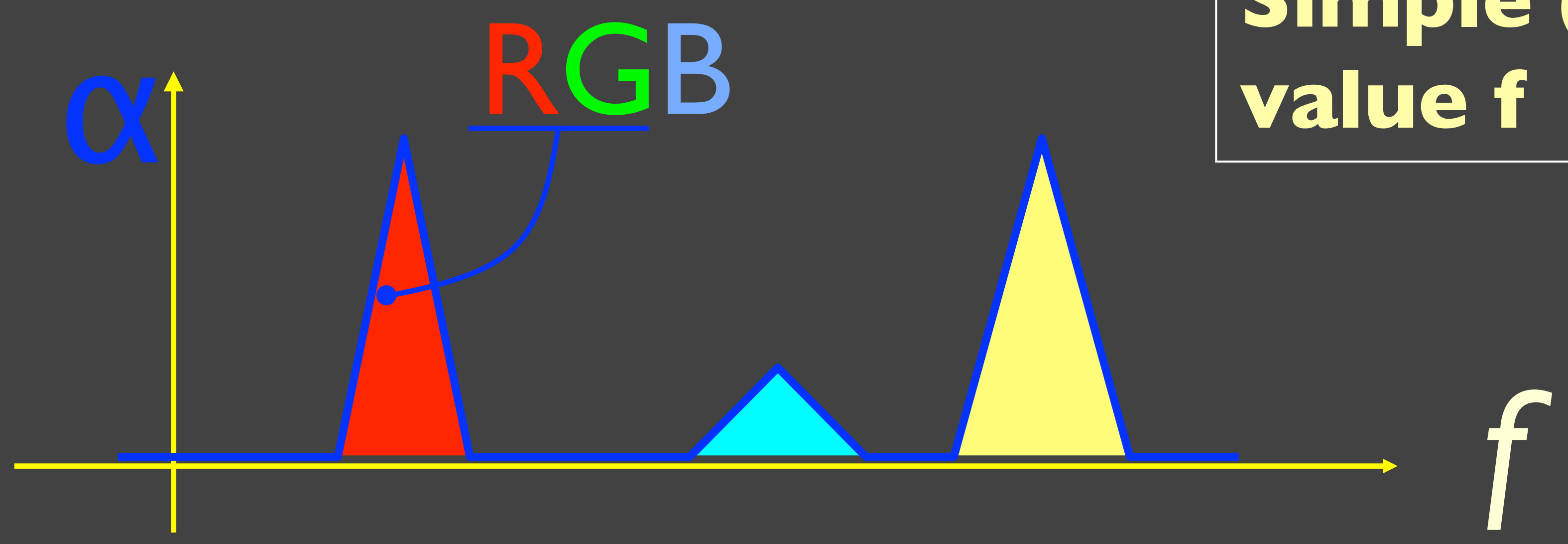
volume rendering:

Transfer Functions (TFs)

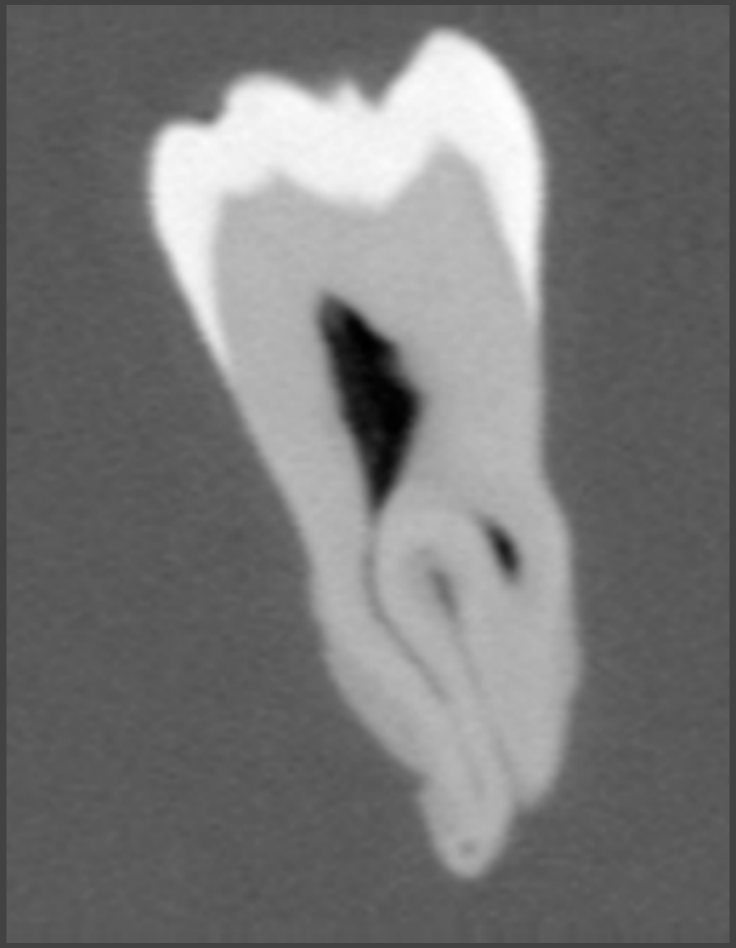
Simple (usual) case: Map data value f to color and opacity



Transfer Functions (TFs)



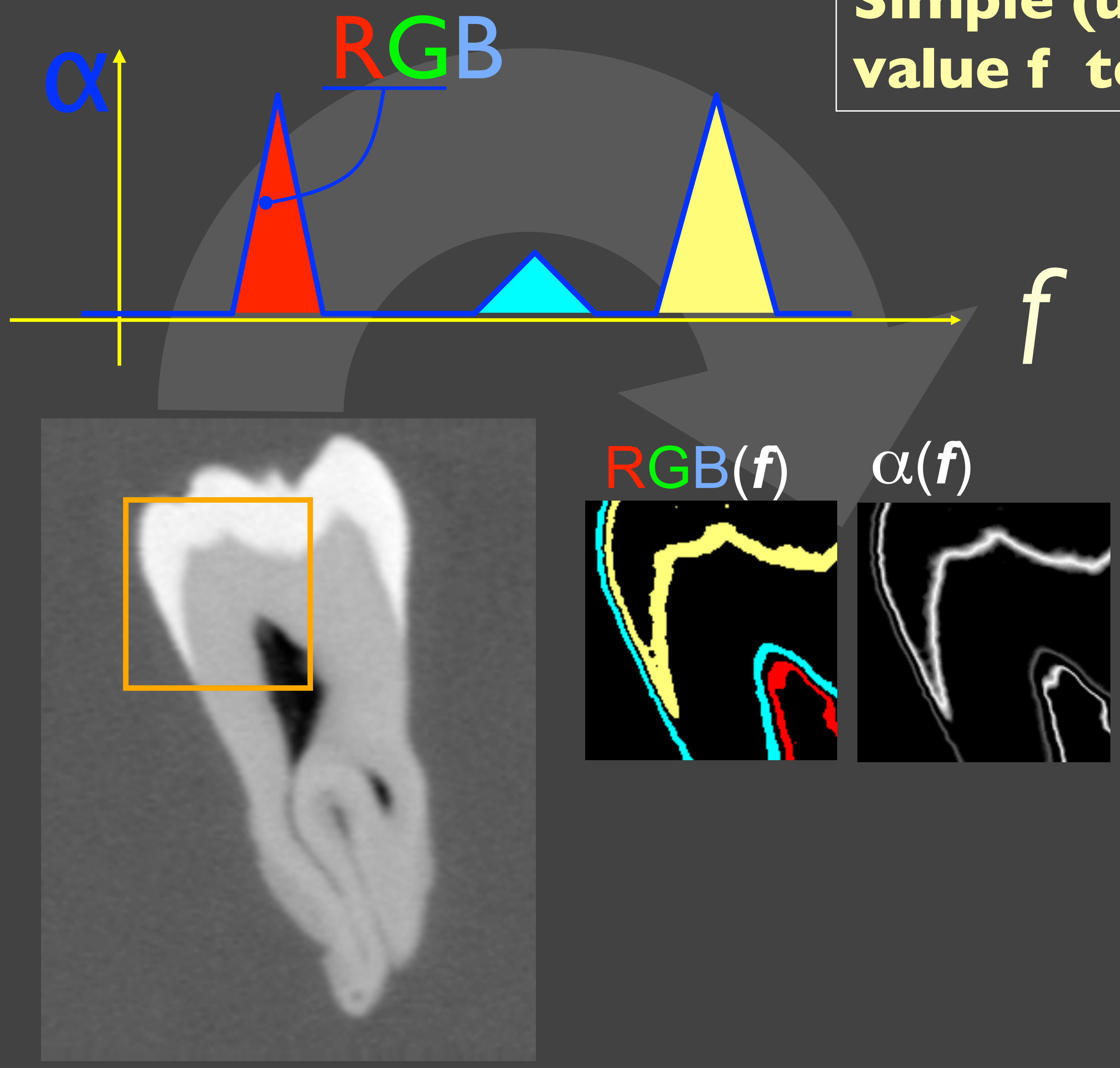
Simple (usual) case: Map data value f to color and opacity



Human Tooth CT

Transfer Functions (TFs)

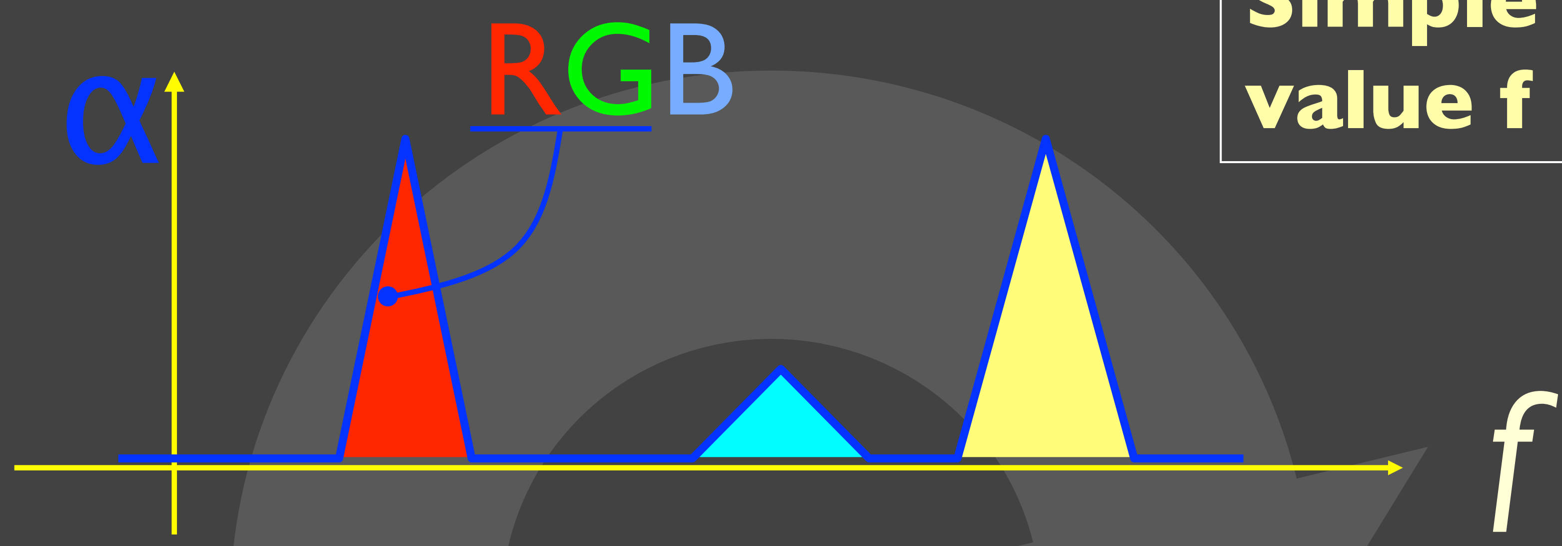
Simple (usual) case: Map data value f to color and opacity

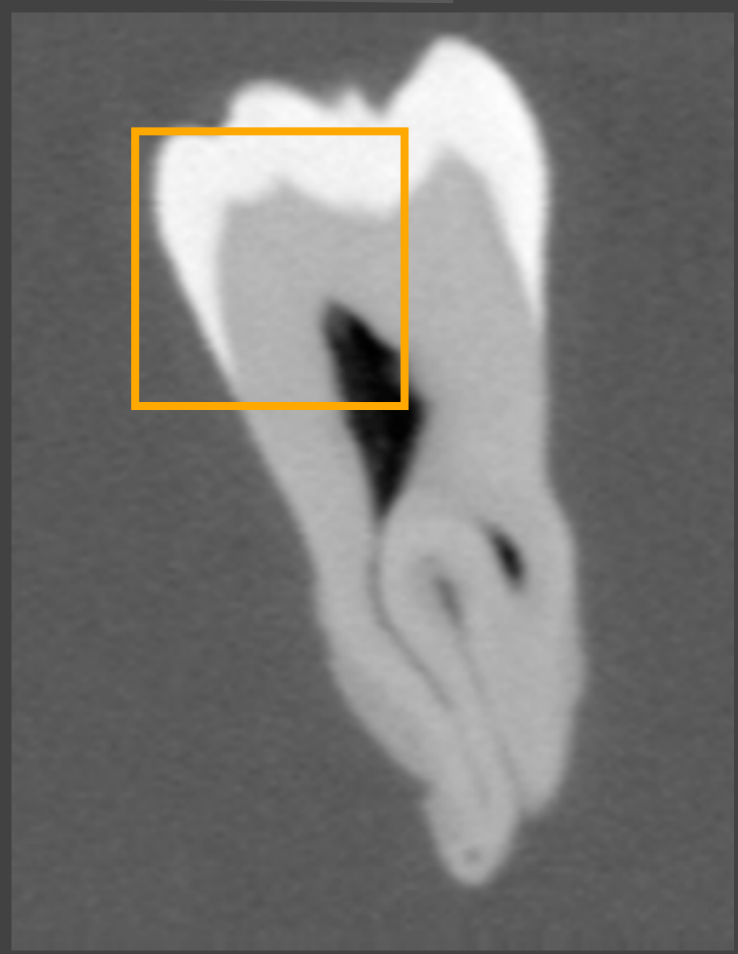


Human Tooth CT

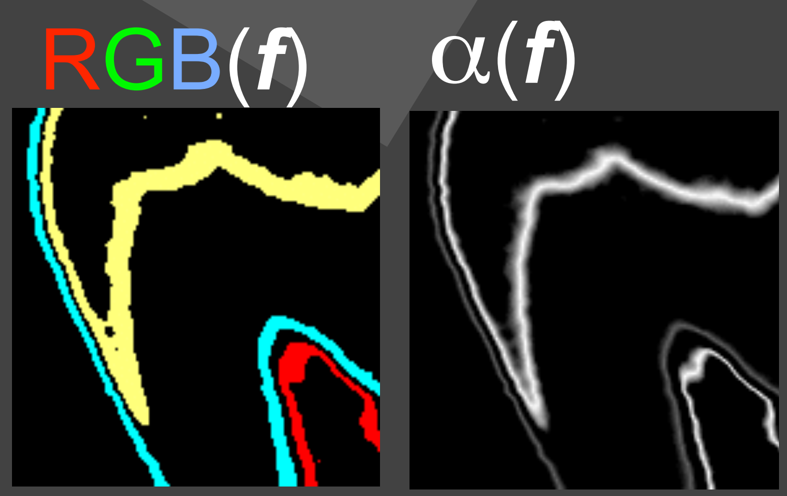
Transfer Functions (TFs)

Simple (usual) case: Map data value f to color and opacity





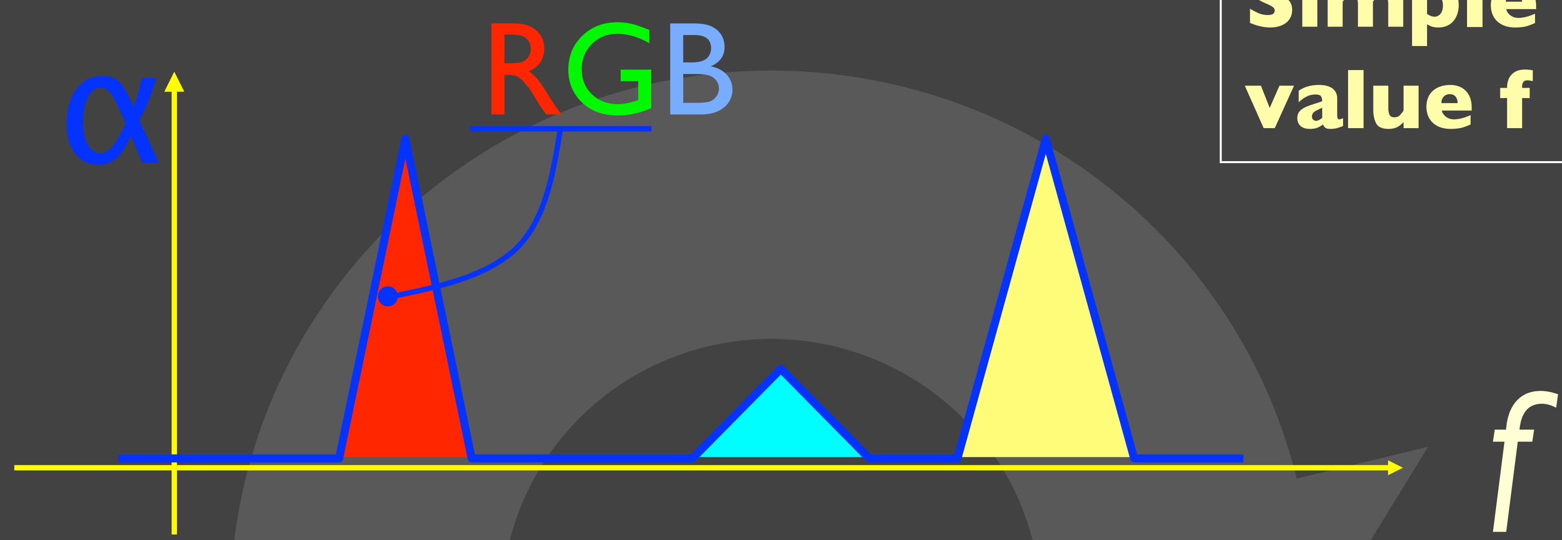
Human Tooth CT

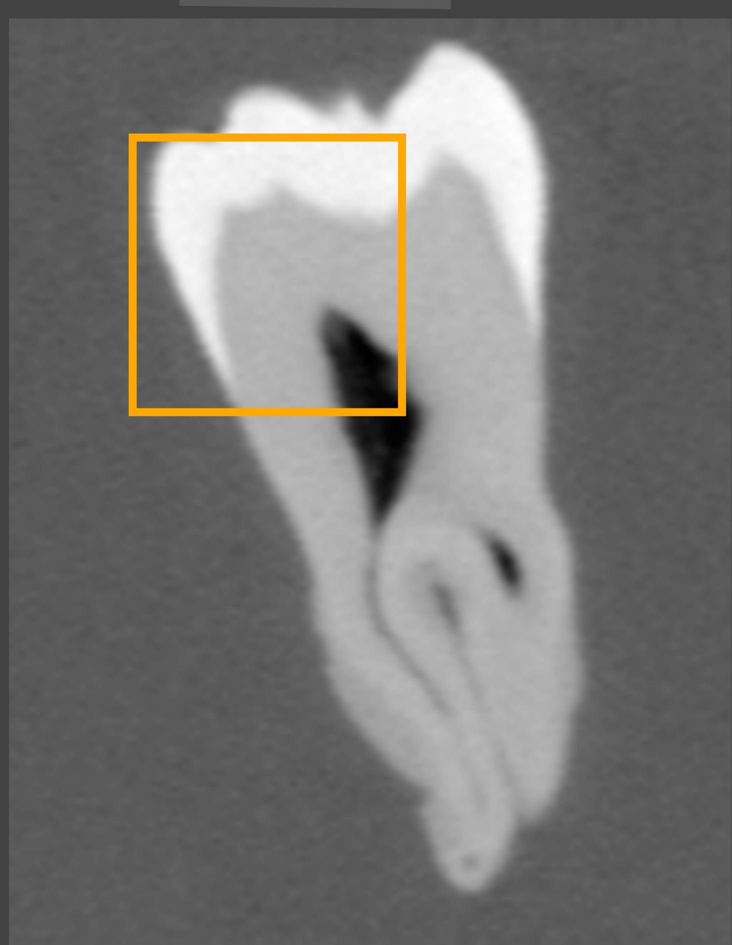


Shading, Compositing…

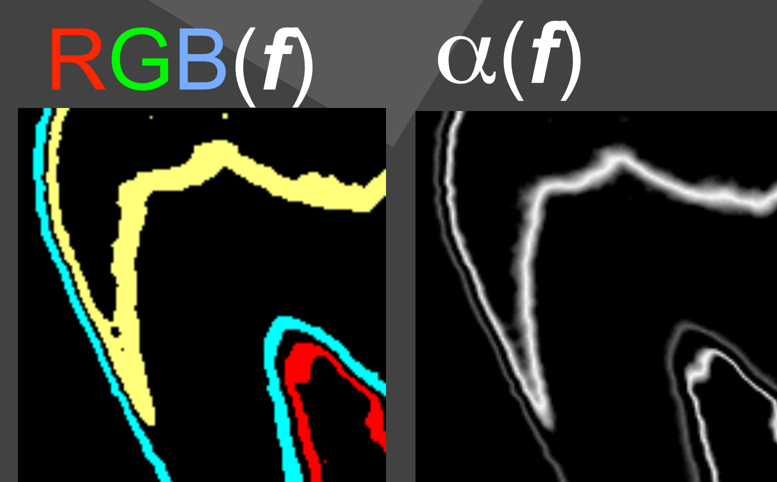
Transfer Functions (TFs)

Simple (usual) case: Map data value f to color and opacity

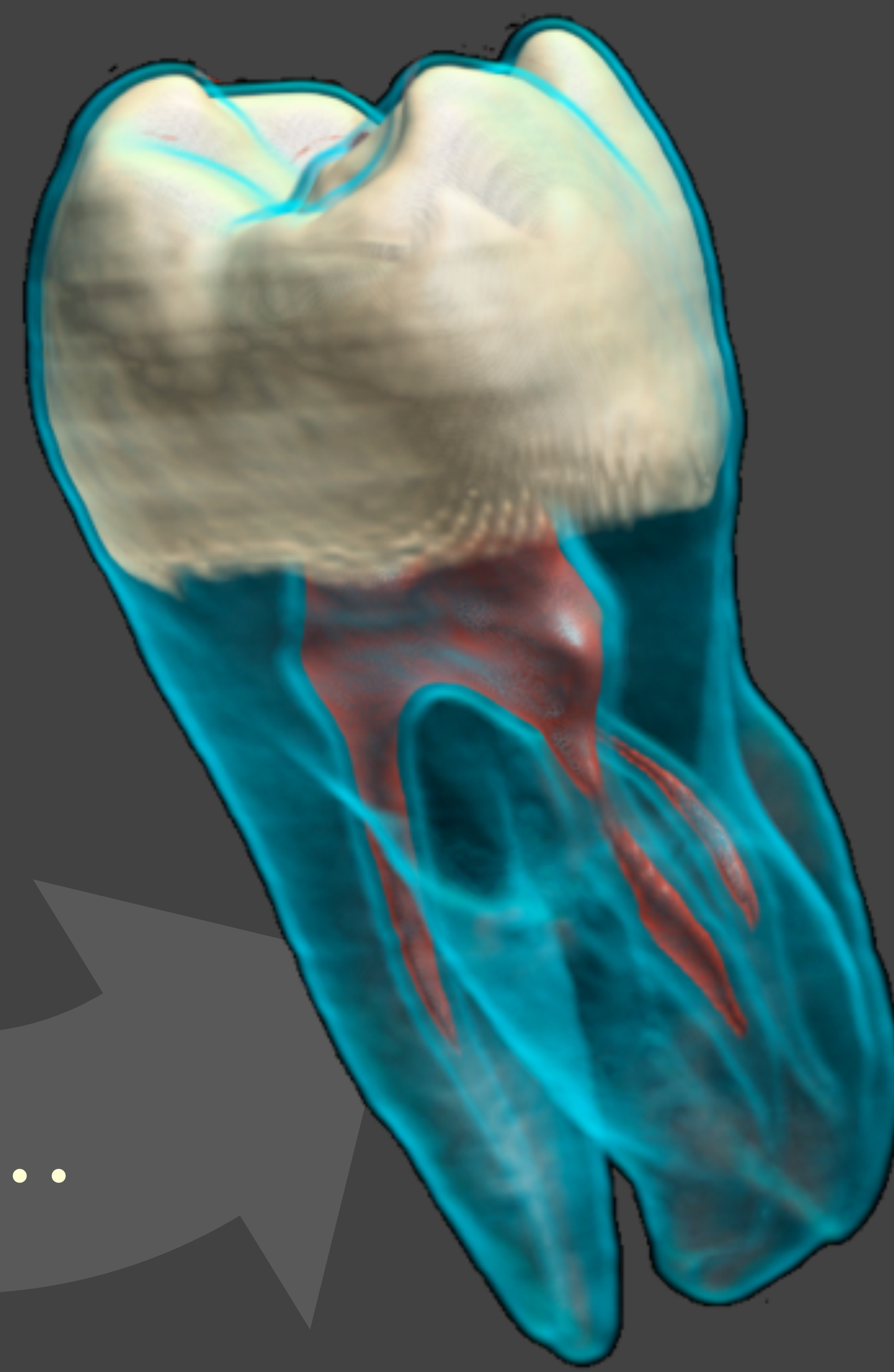




Human Tooth CT

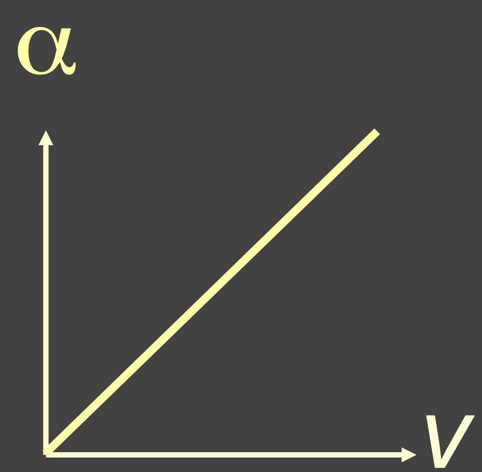


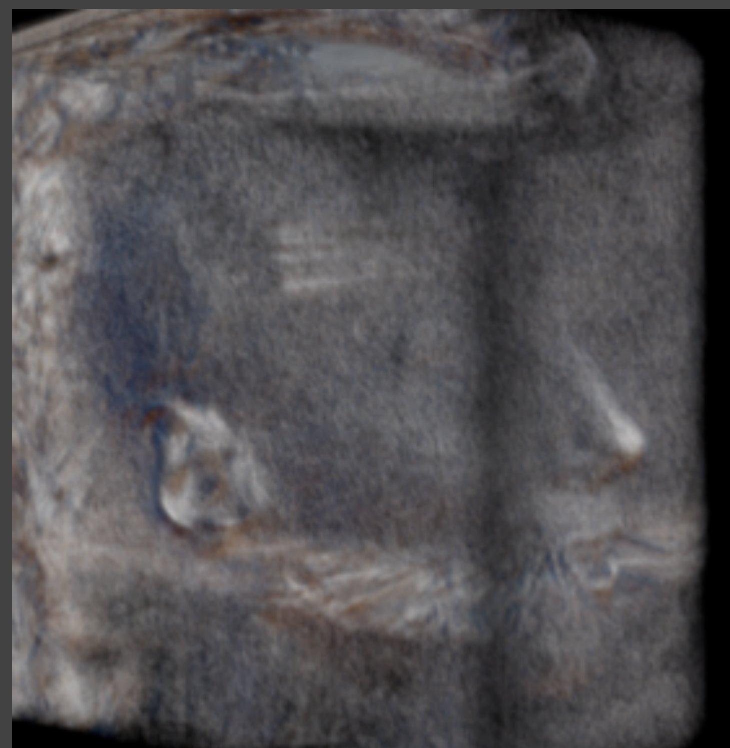
Shading, Compositing…

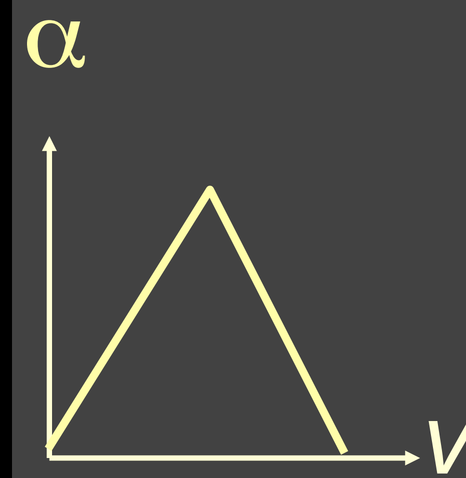


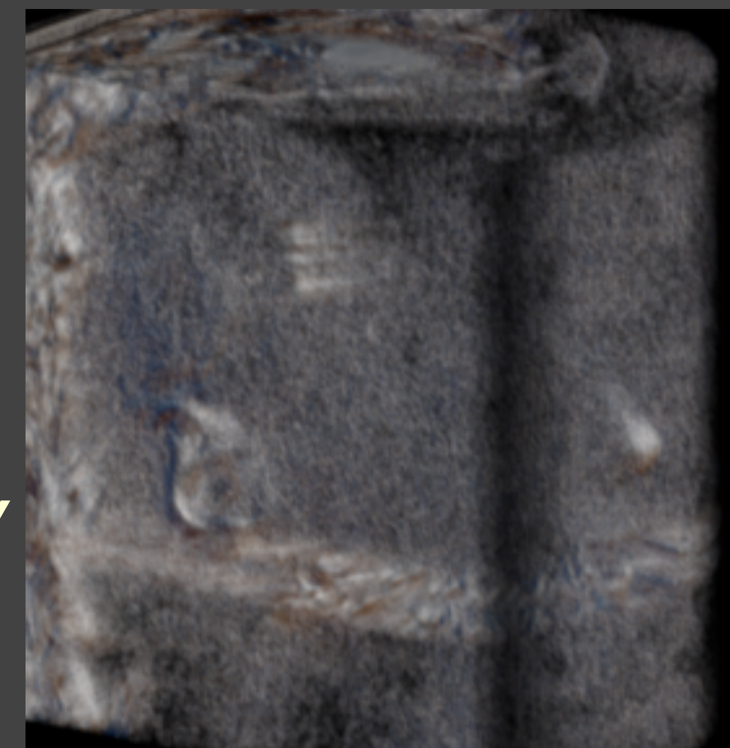
- “Optical Properties”: Anything that can be composited with a standard graphics operator (“over”)
  - Opacity: “opacity functions”
    - *Most important*
  - Color
    - *Can help distinguish features*
  - Phong parameters (k_a, k_d, k_s)
  - Index of refraction

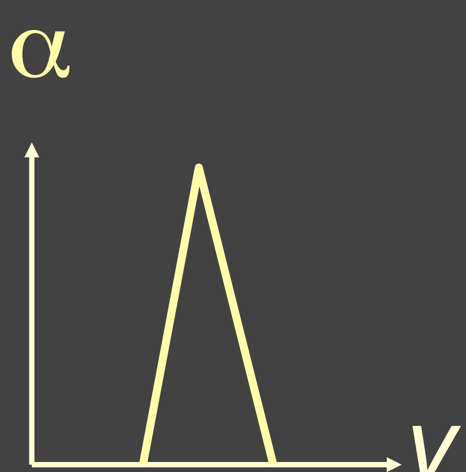
Setting Transfer Function: Hard

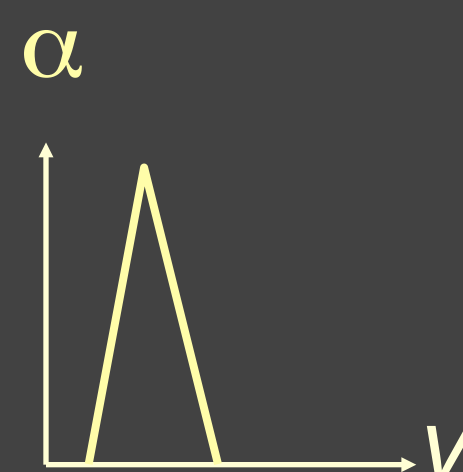


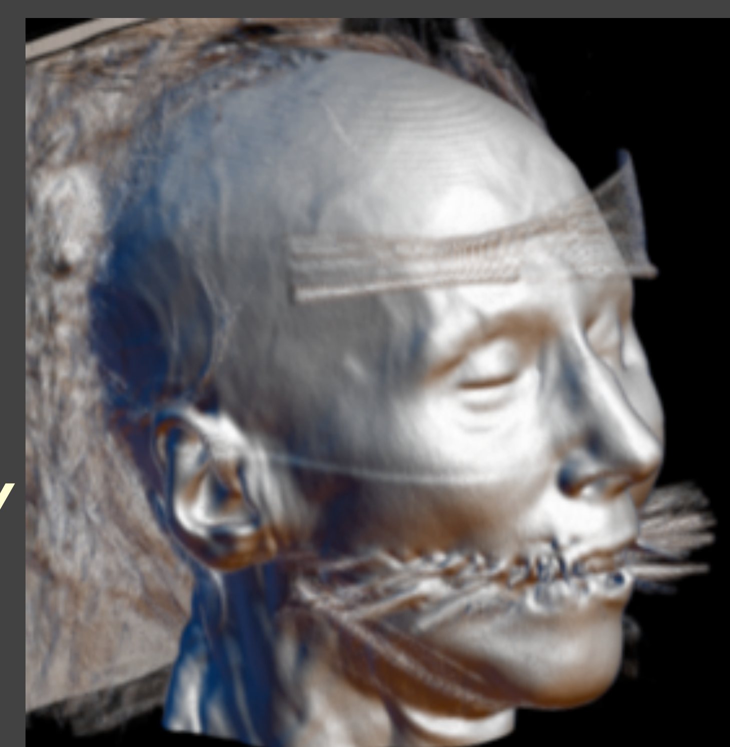




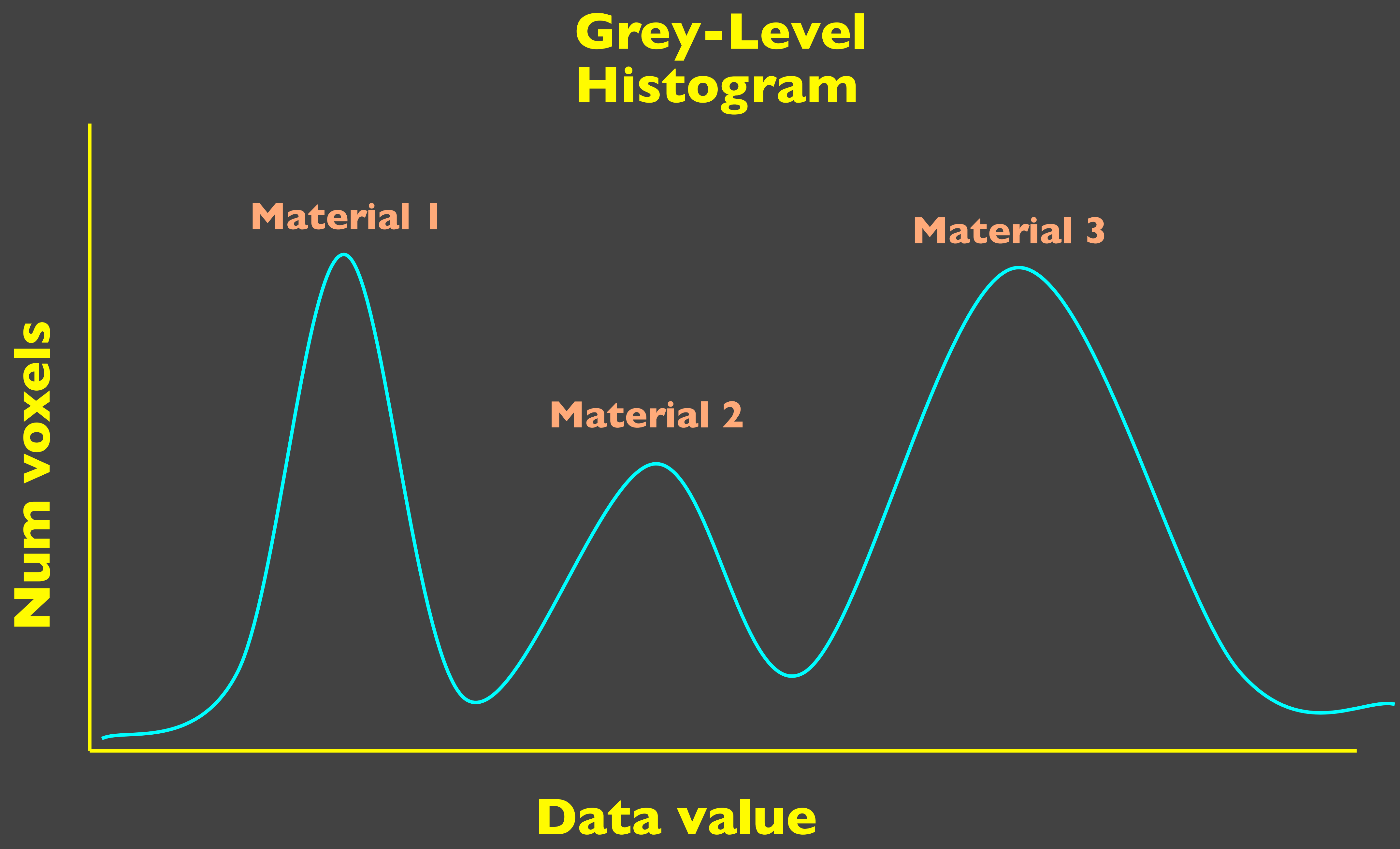








Volumes as Consisting of Materials



What Makes TF's Hard?

1. Non-spatial: spatial isolation doesn't imply data value isolation
2. Many degrees of freedom
3. No constraints or guidance
4. Material uniformity assumption

Goals for TF Design

- Make good renderings easier to come by
- Make space of TFs less confusing
- Remove excess “flexibility”
- Provide one or more of:
  - Information
  - Guidance
  - Semi-automation / Automation

TF Techniques/Tools

1. **Trial and Error (manual)**
2. Image-Centric Approach
3. Data-Centric Approach

DEMOS:
1D TRANSFER FUNCTION

1. Trial and Error

- Manually edit graph of transfer function
- Enforces learning by experience
- Get better with practice
- Can make terrific images

William Schroeder, Lisa Sobierajski Avila, and Ken Martin; Transfer Function Bake-off Vis '00

TF Techniques/Tools

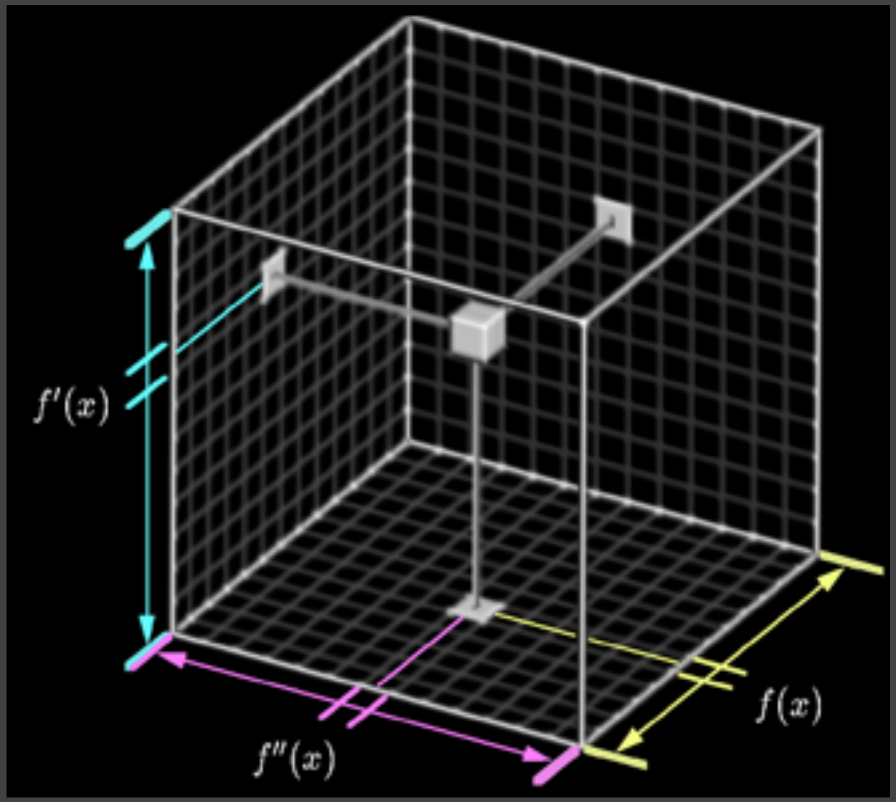
1. Trial and Error (manual)
2. **Image-Centric Approach**
3. Data-Centric Approach

2. Use derivatives

Reasoning:

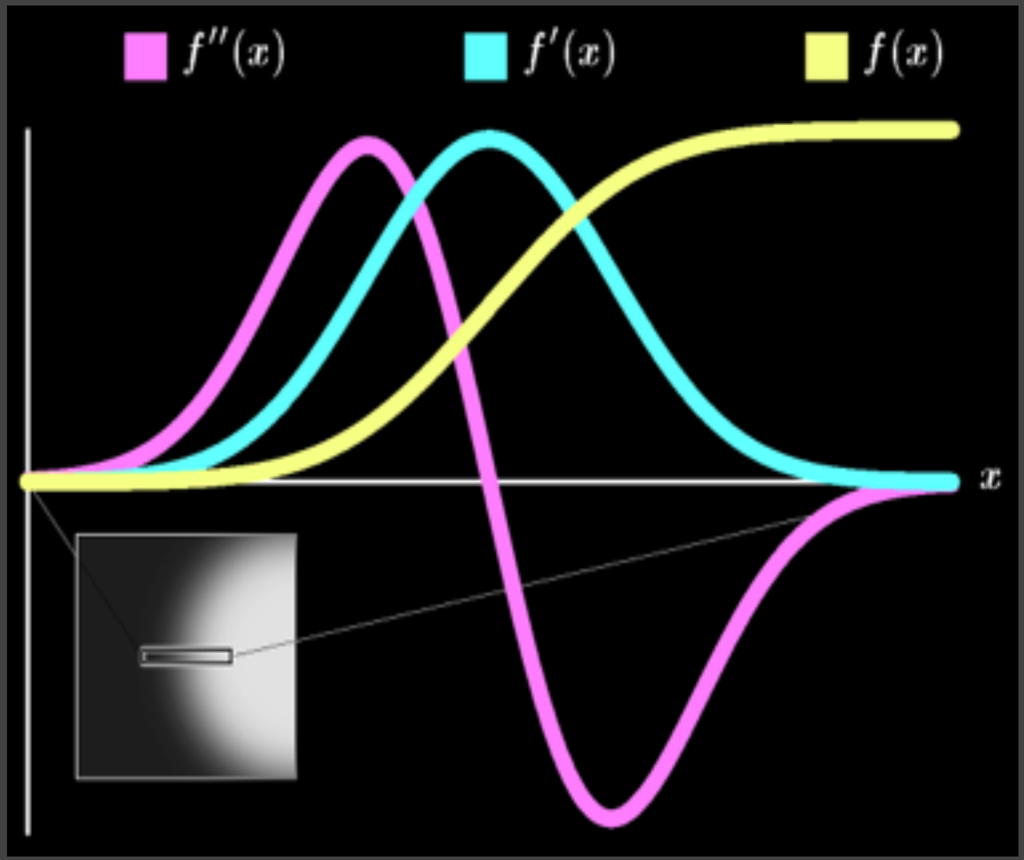
- TFs are volume-position invariant
- Histograms “project out” position
- Interested in boundaries between materials
- Boundaries characterized by derivatives

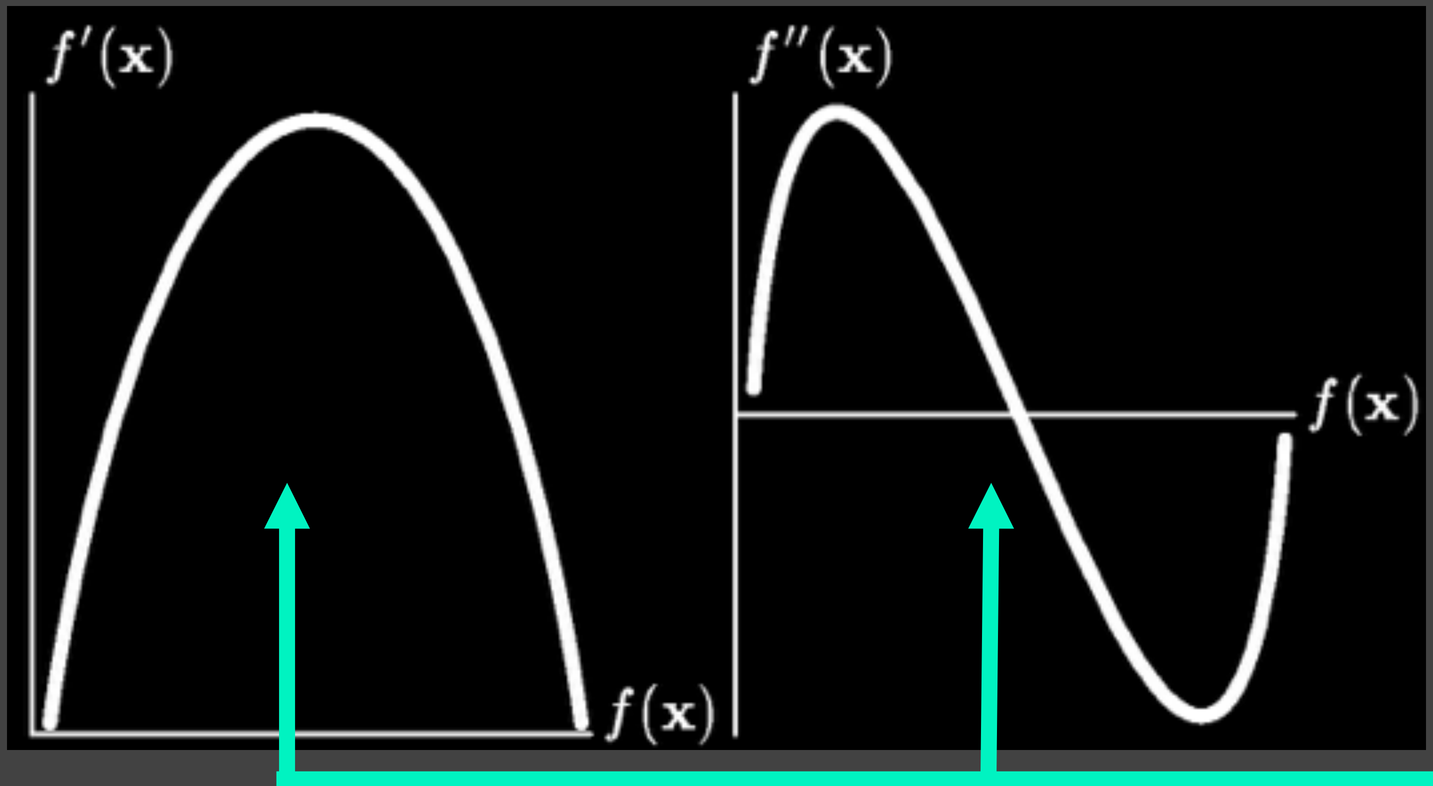
Make 3D histograms of value, 1st, 2nd deriv.



By (1) inspecting and (2) algorithmically analyzing histogram volume, we can create transfer functions

Derivative relationships

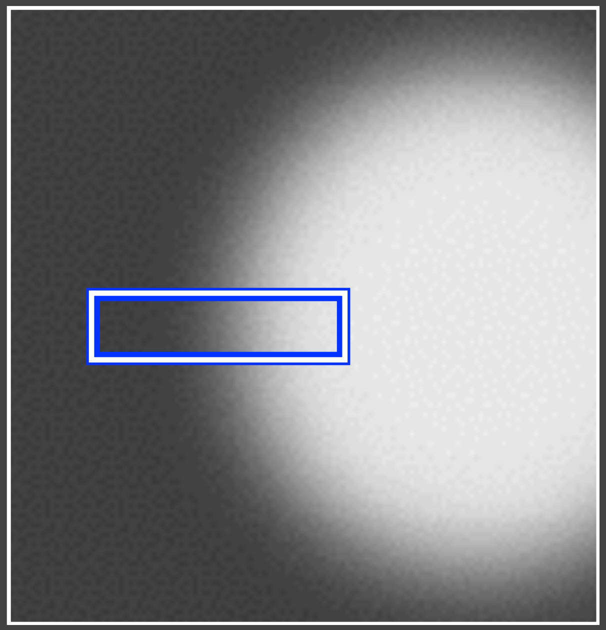


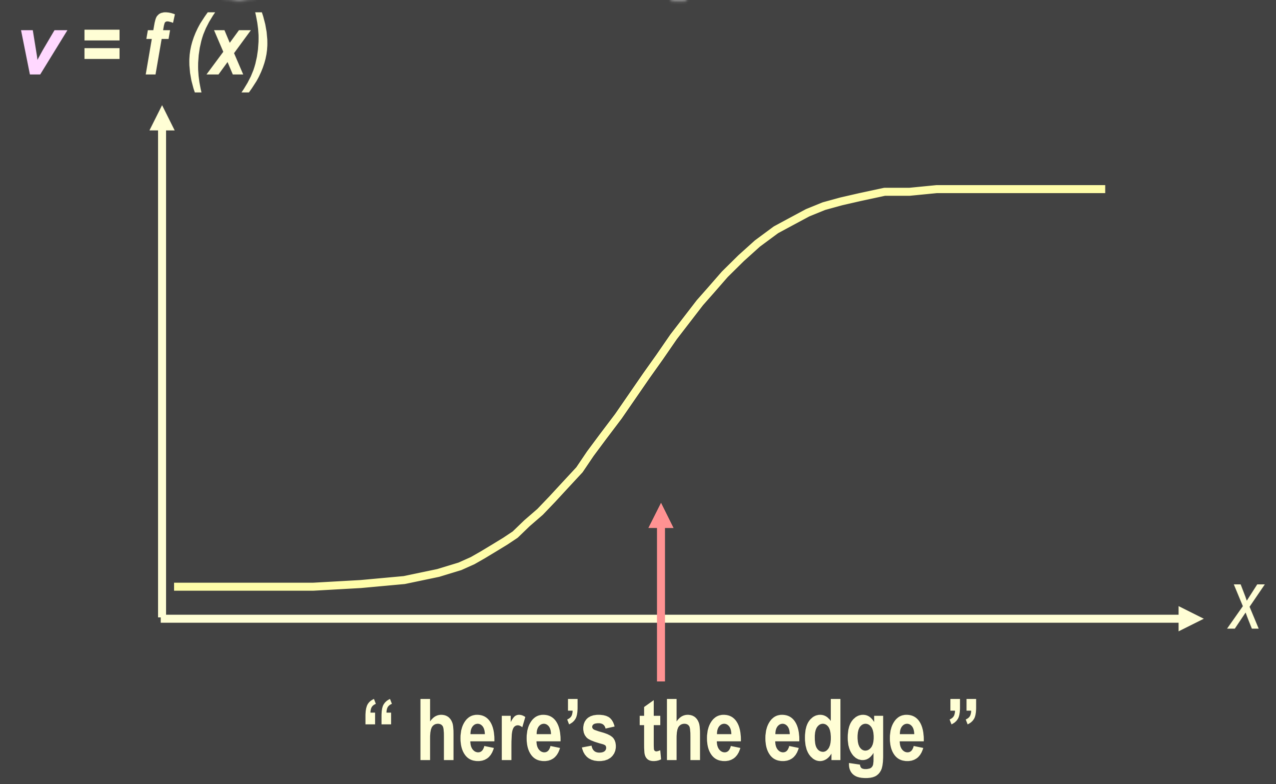


Edges at maximum of 1st derivative or zero-crossing of 2nd

Finding edges: easy

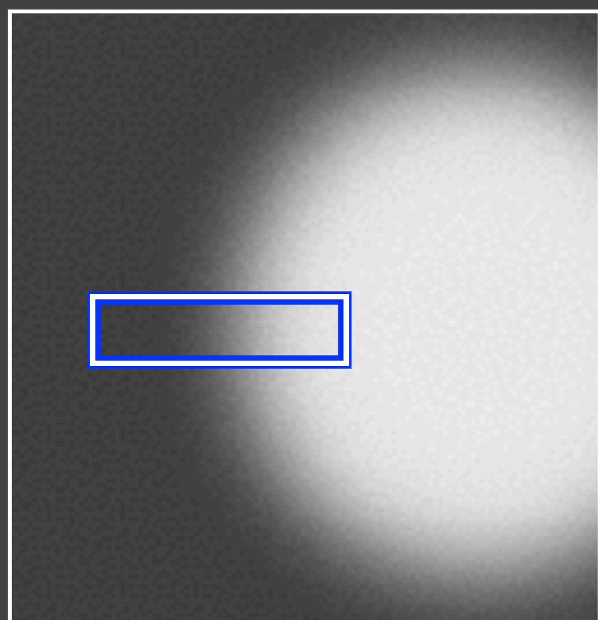
“Where’s the edge?”

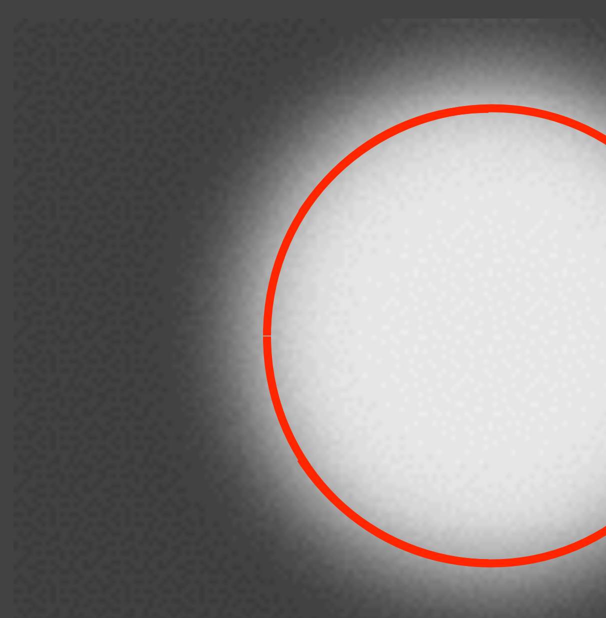




Finding edges: easy

"Where's the edge?"

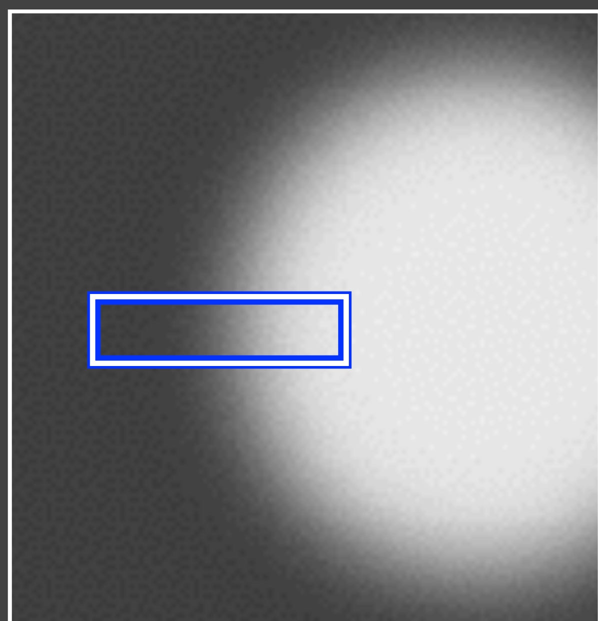


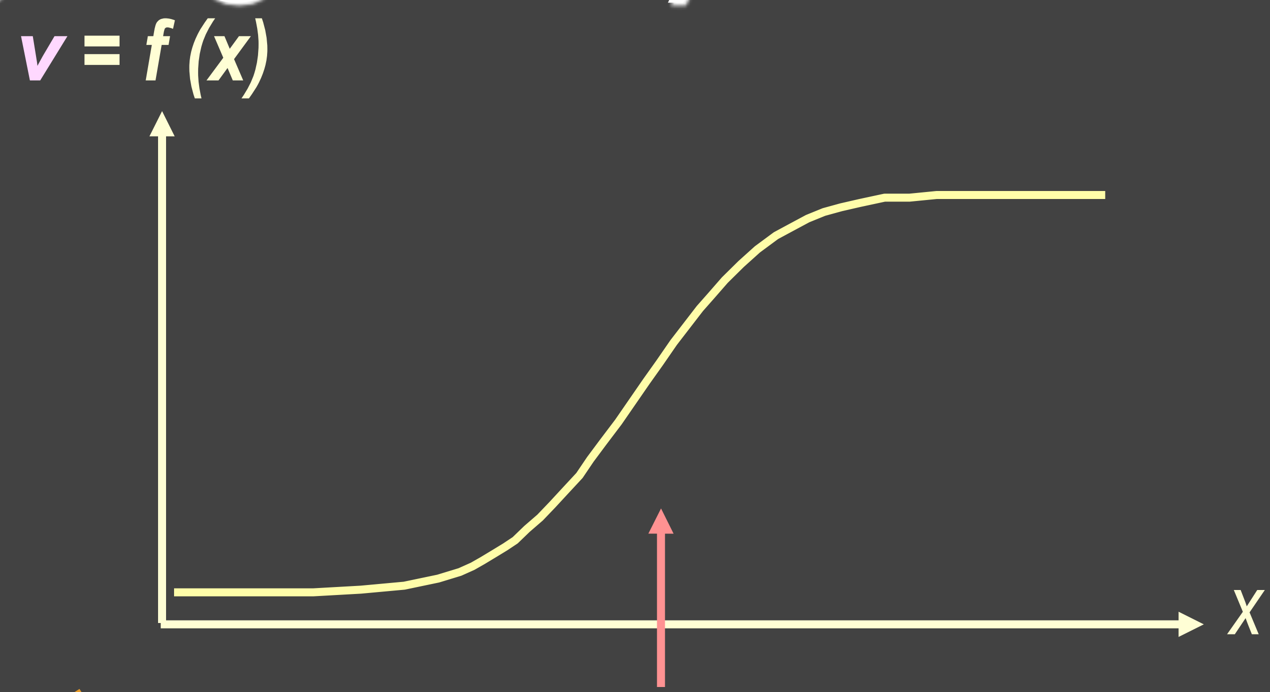


Result: edge pixels

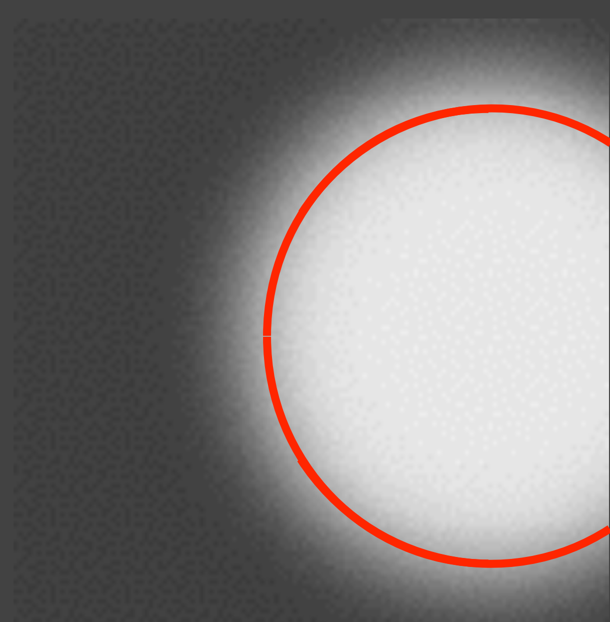
Finding edges: easy

"Where's the edge?"



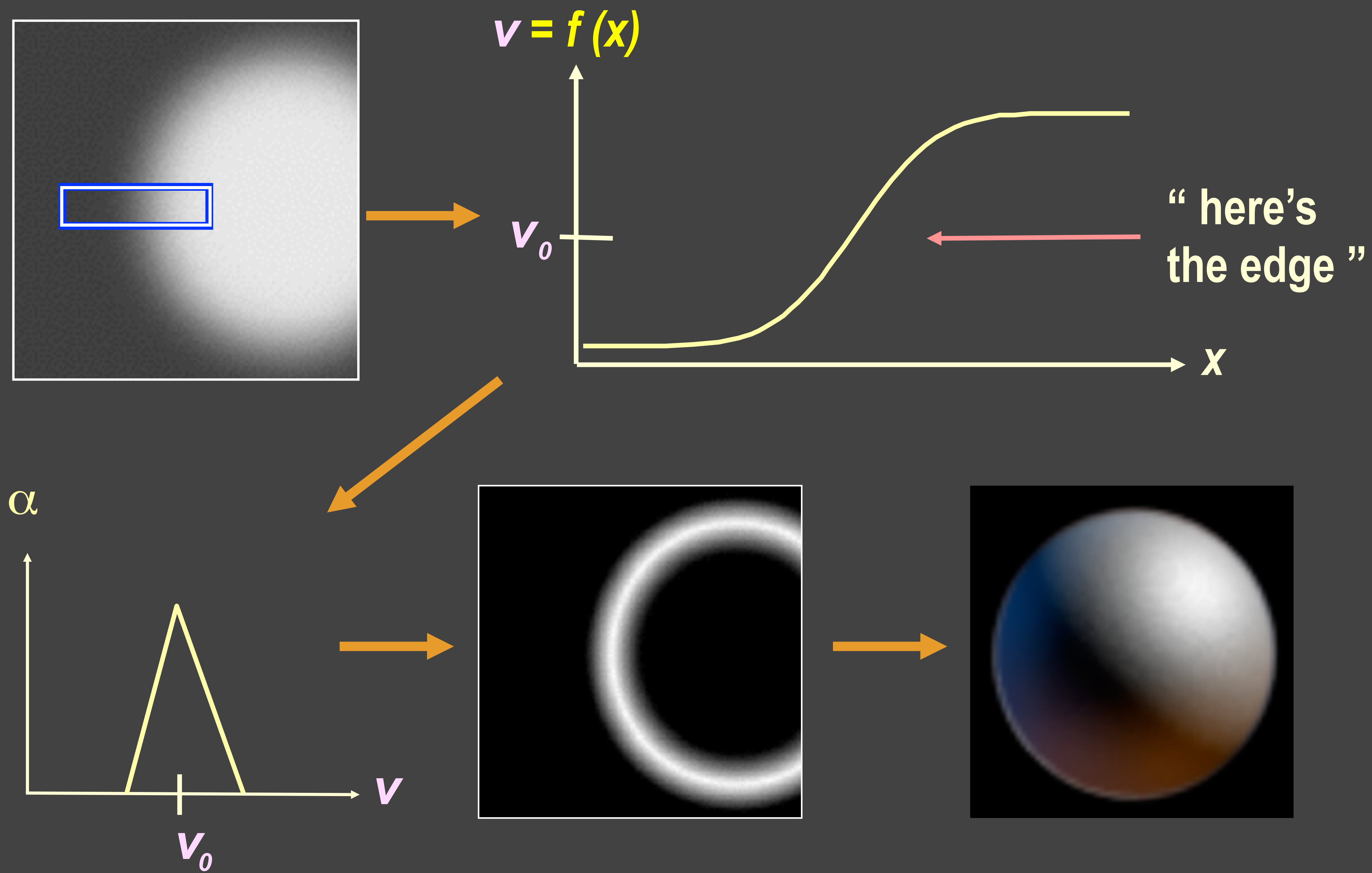


" here's the edge "



Result: edge pixels

Transfer function Unintuitive



Gradient

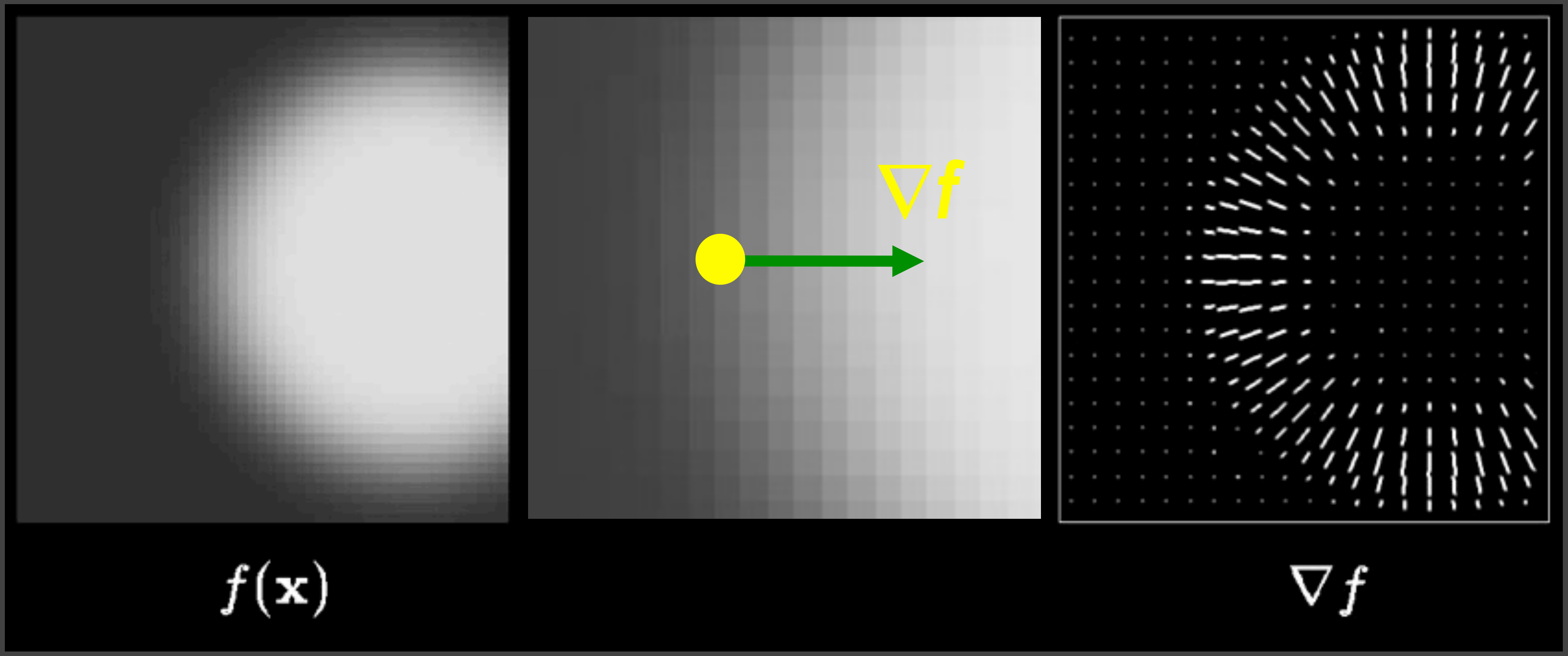
∇f = (dx, dy, dz)

= ((f(1,0,0) - f(-1,0,0))/2,

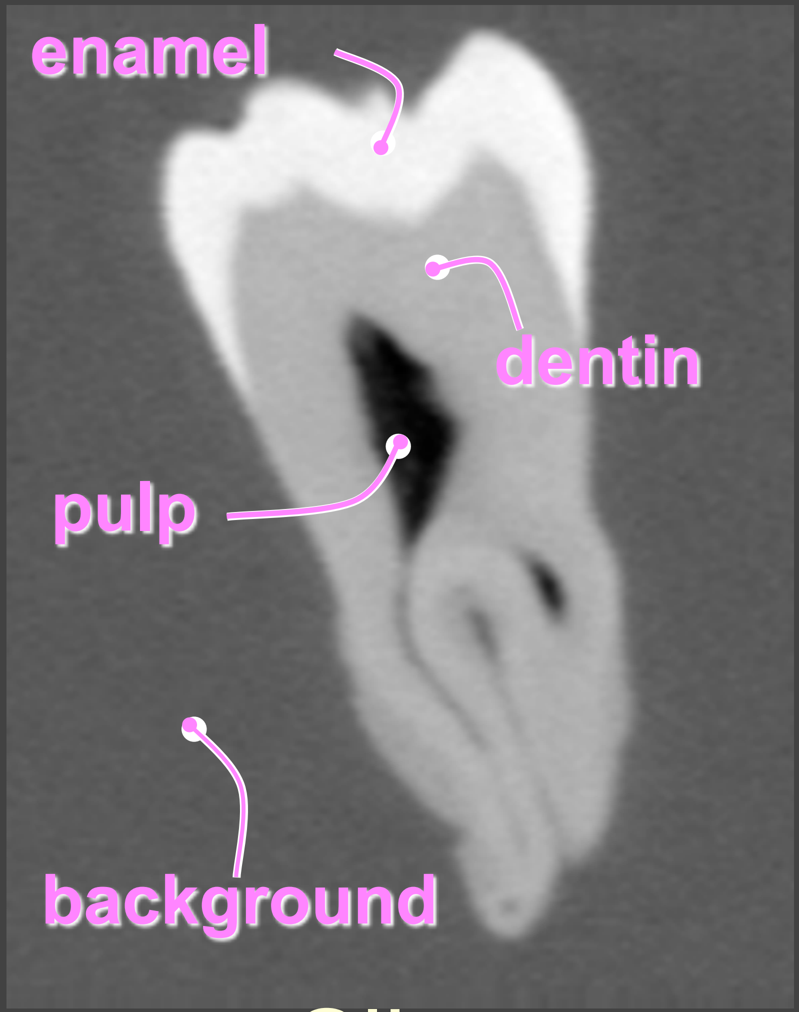
(f(0,1,0) - f(0,-1,0))/2,

(f(0,0,1) - f(0,0,-1))/2)

- Approximates "surface normal" (of isosurface)



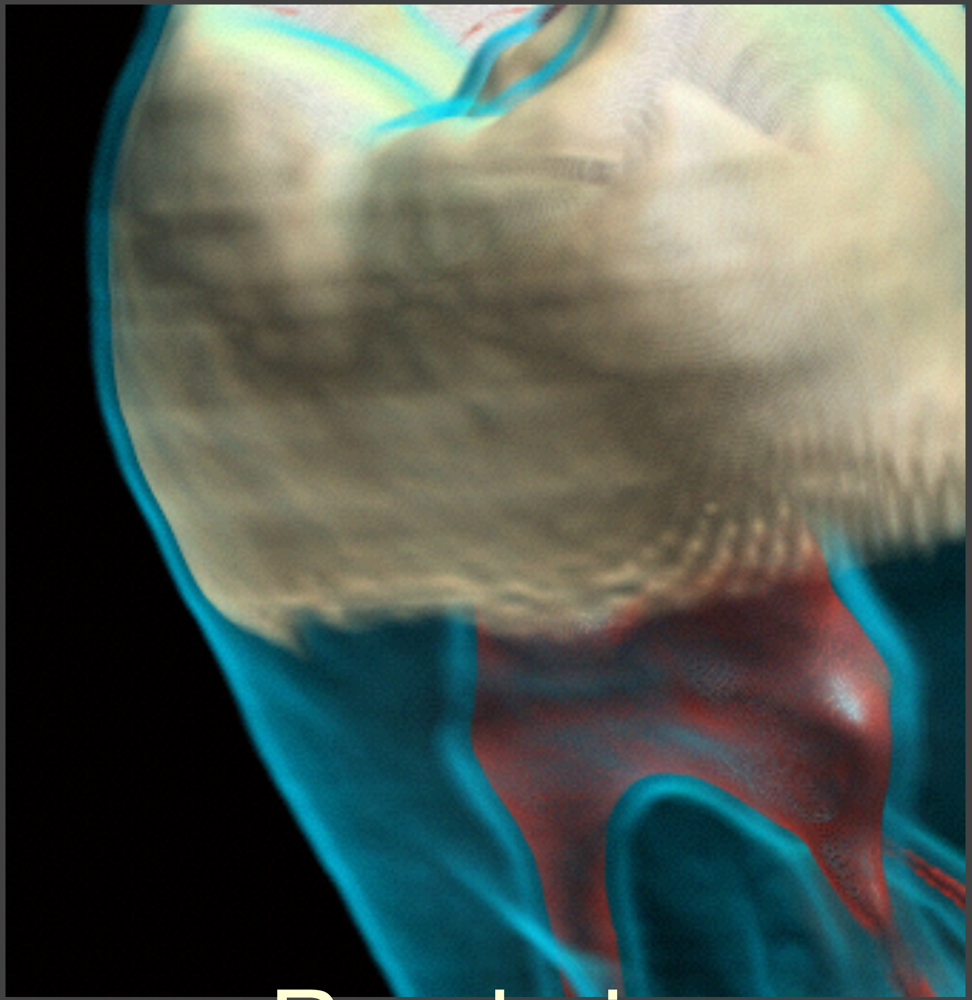
1D TFs: limitation



Slice

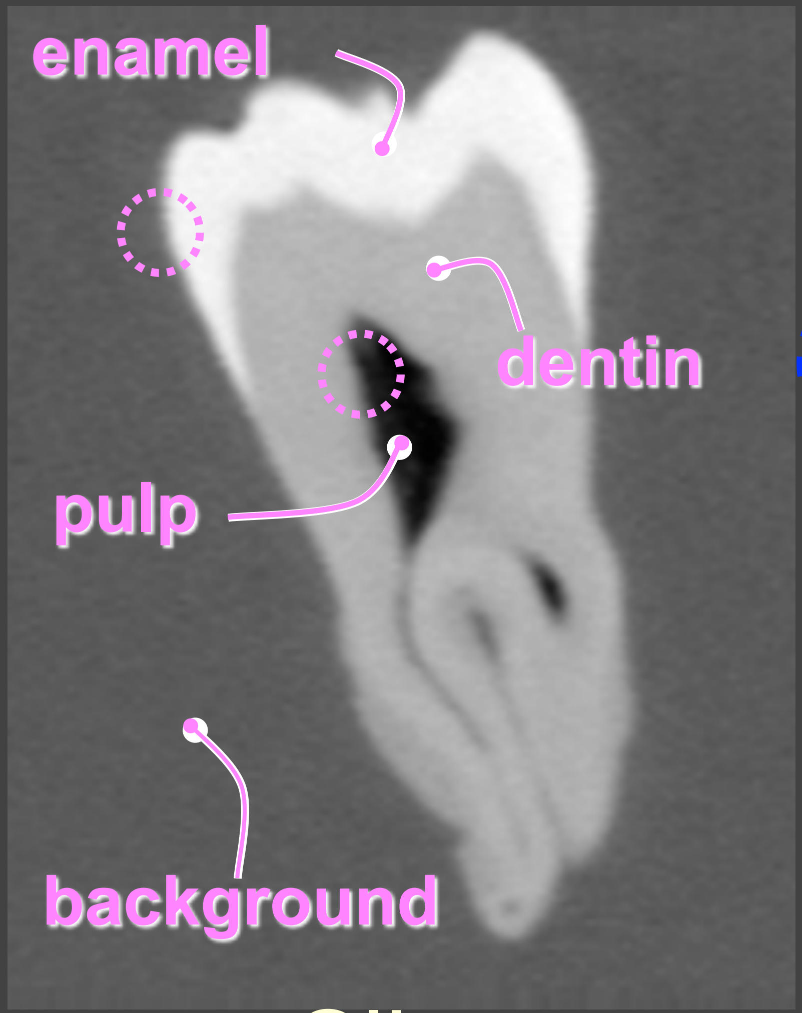
RGB(*f*)


1D TF output


Rendering

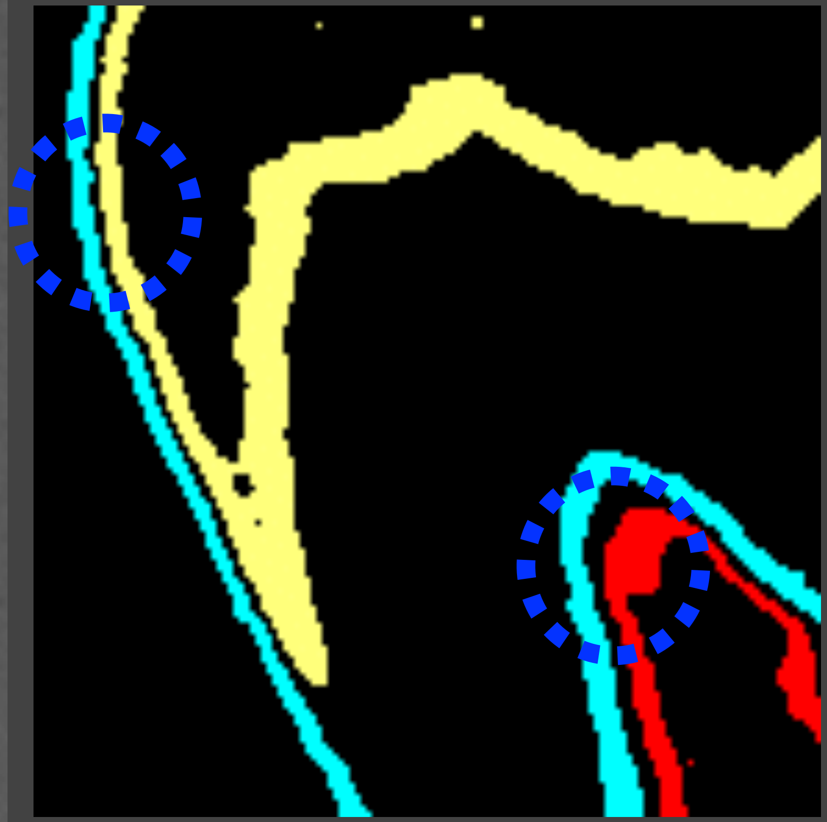
1D transfer functions can not accurately capture all material boundaries

1D TFs: limitation

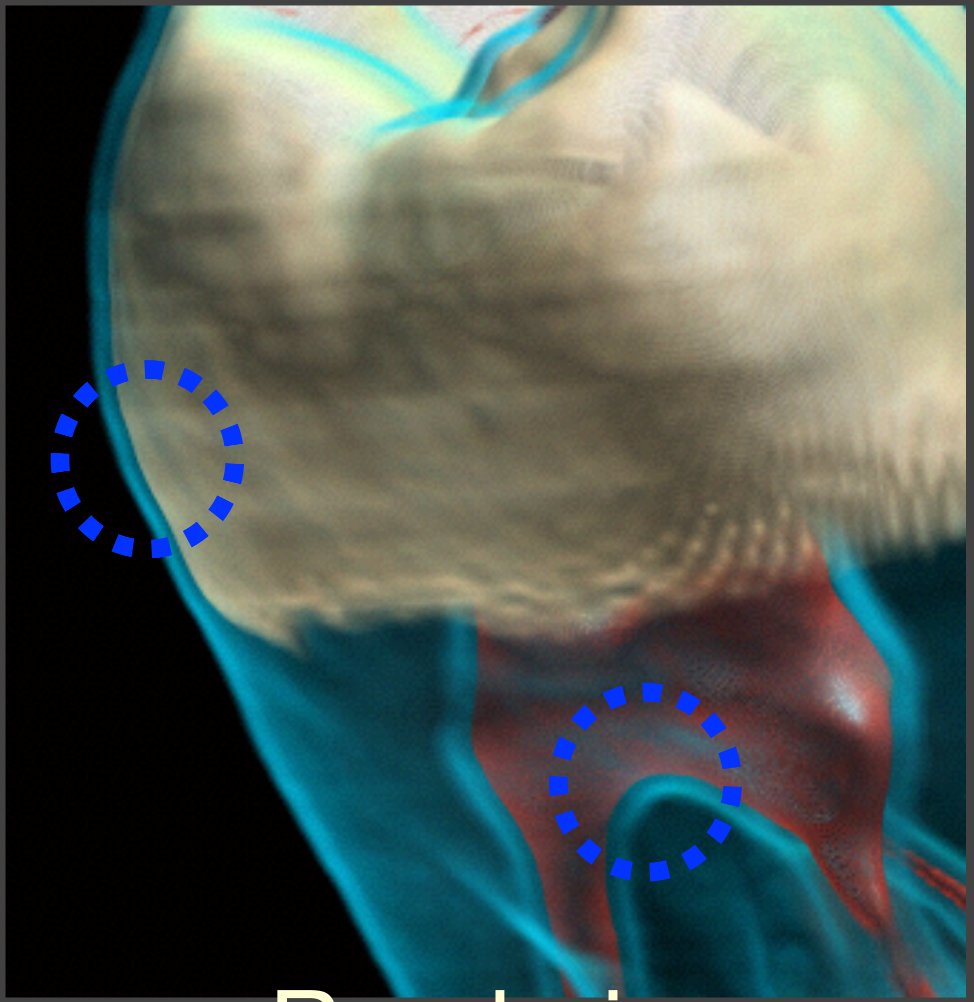


Slice

RGB(*f*)



1D TF output



Rendering

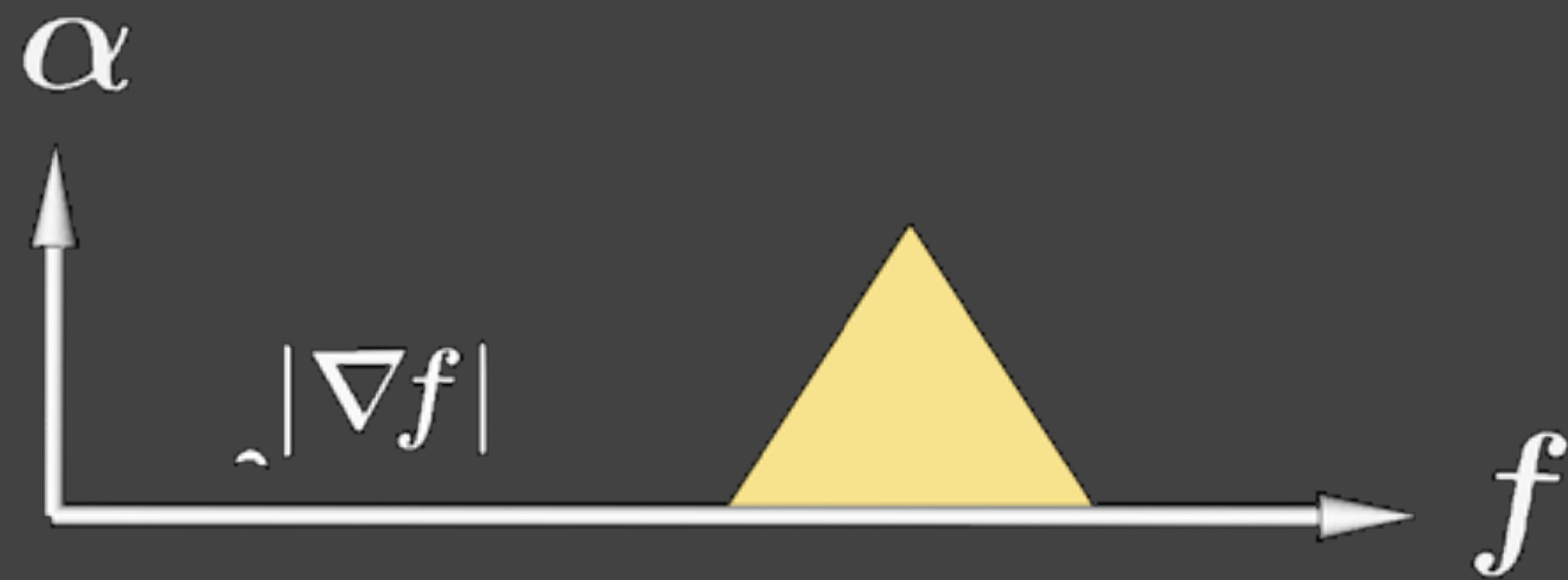
1D transfer functions can not accurately capture all material boundaries

1D → 2D Transfer Function

$\text{RGB}(f)$
$\alpha(f)$
} Generalize…

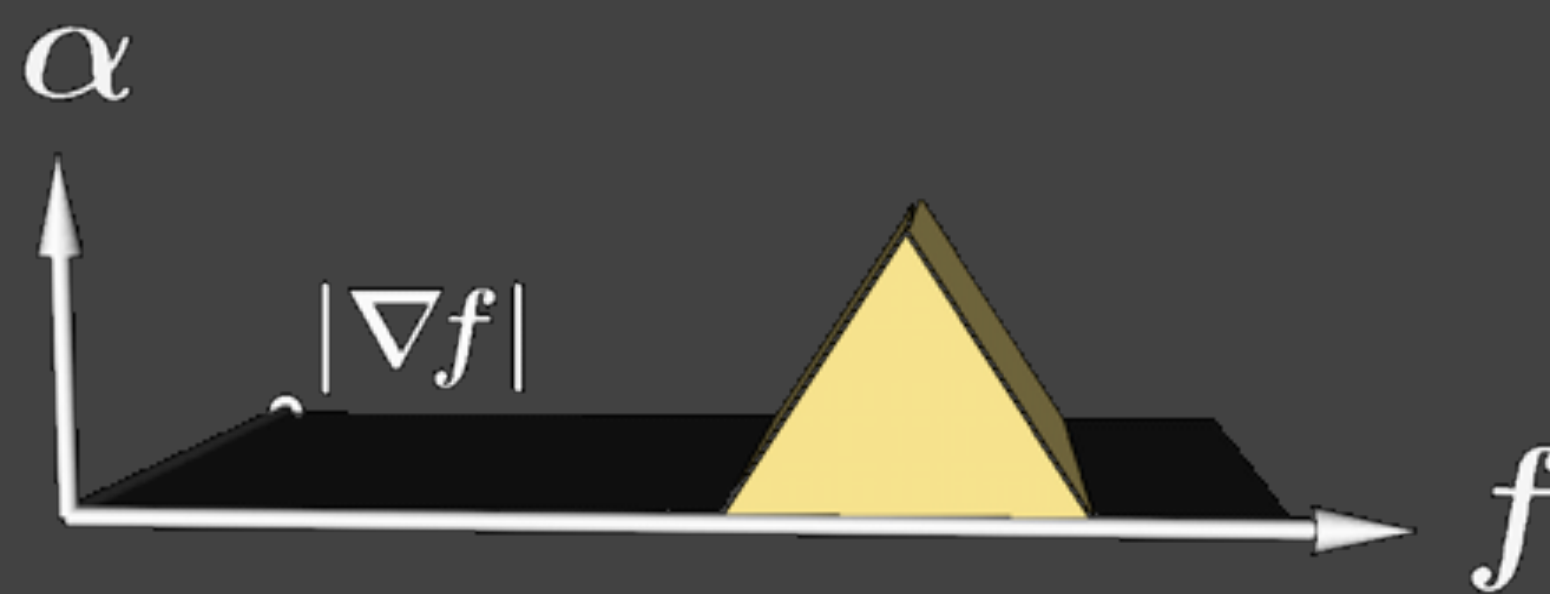
1D → 2D Transfer Function

$RGB(f)$
$\alpha(f)$
} Generalize…



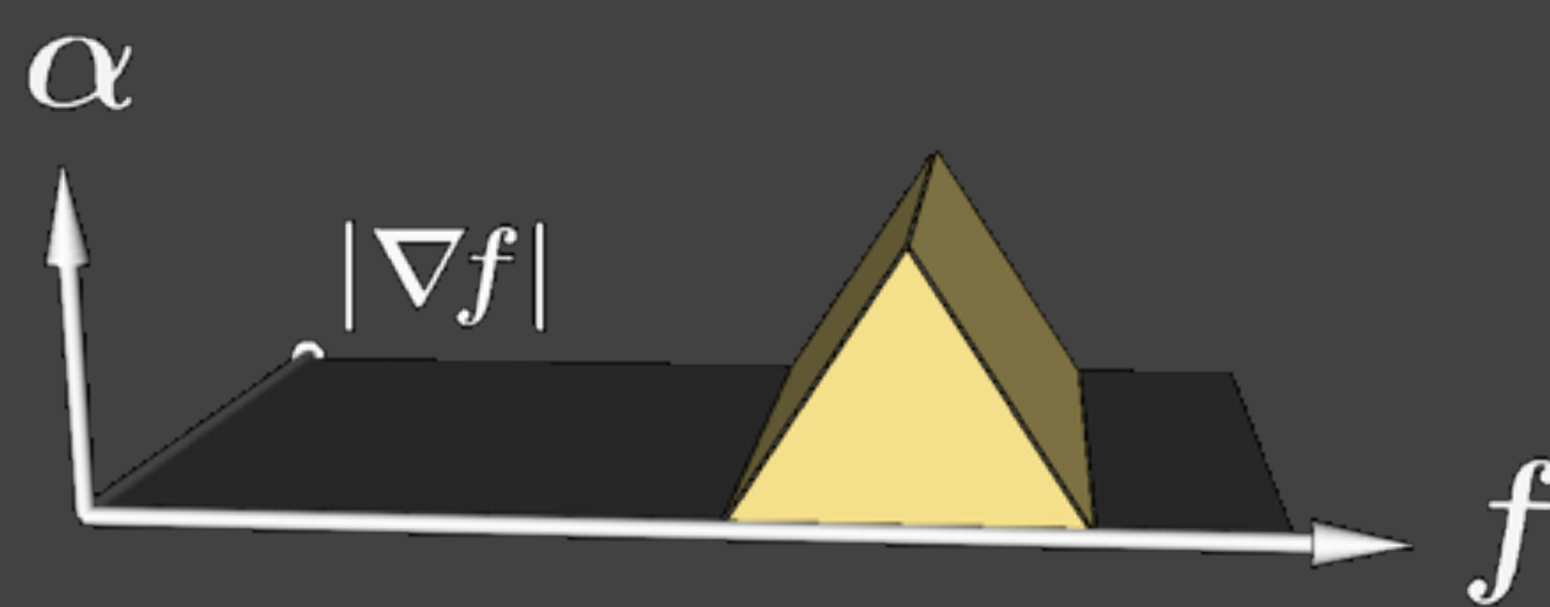
1D → 2D Transfer Function

$\text{RGB}(f)$, $\alpha(f)$ } Generalize…



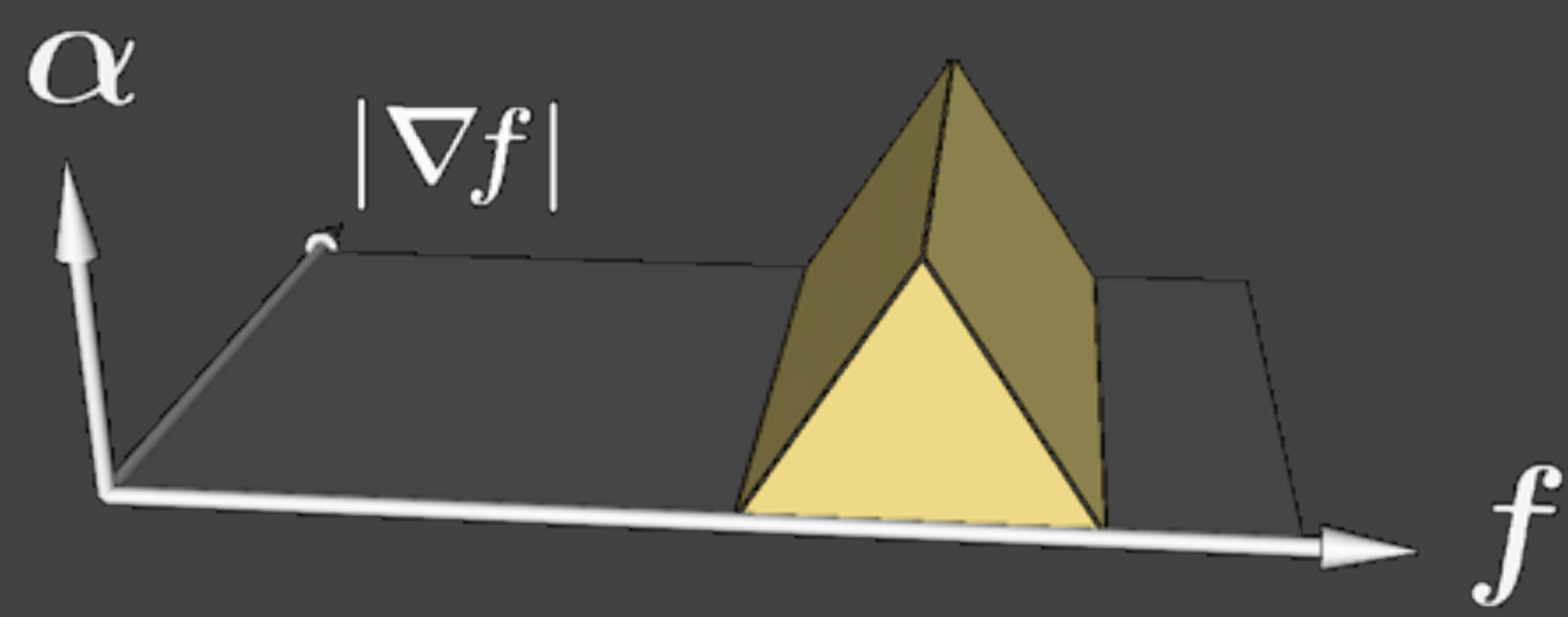
1D → 2D Transfer Function

$RGB(f)$
$\alpha(f)$ } Generalize…



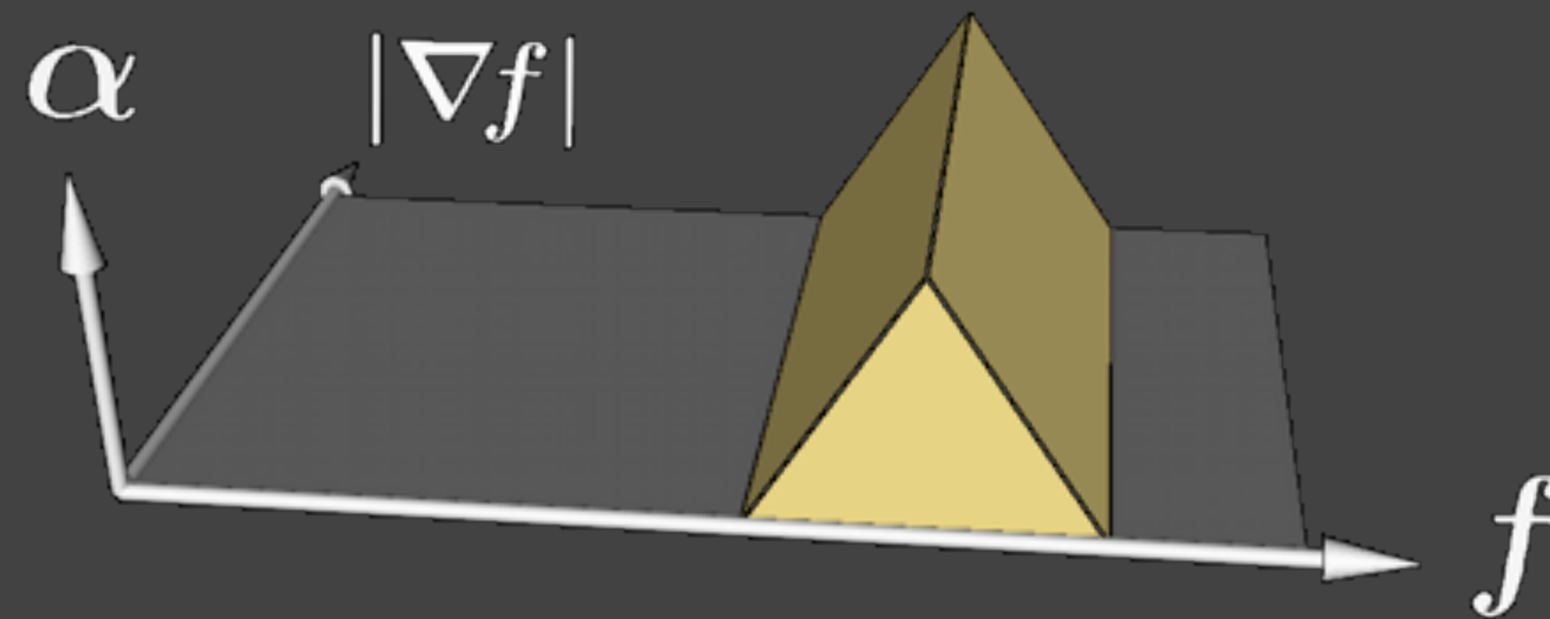
1D → 2D Transfer Function

$\text{RGB}(f)$
$\alpha(f)$
} Generalize…



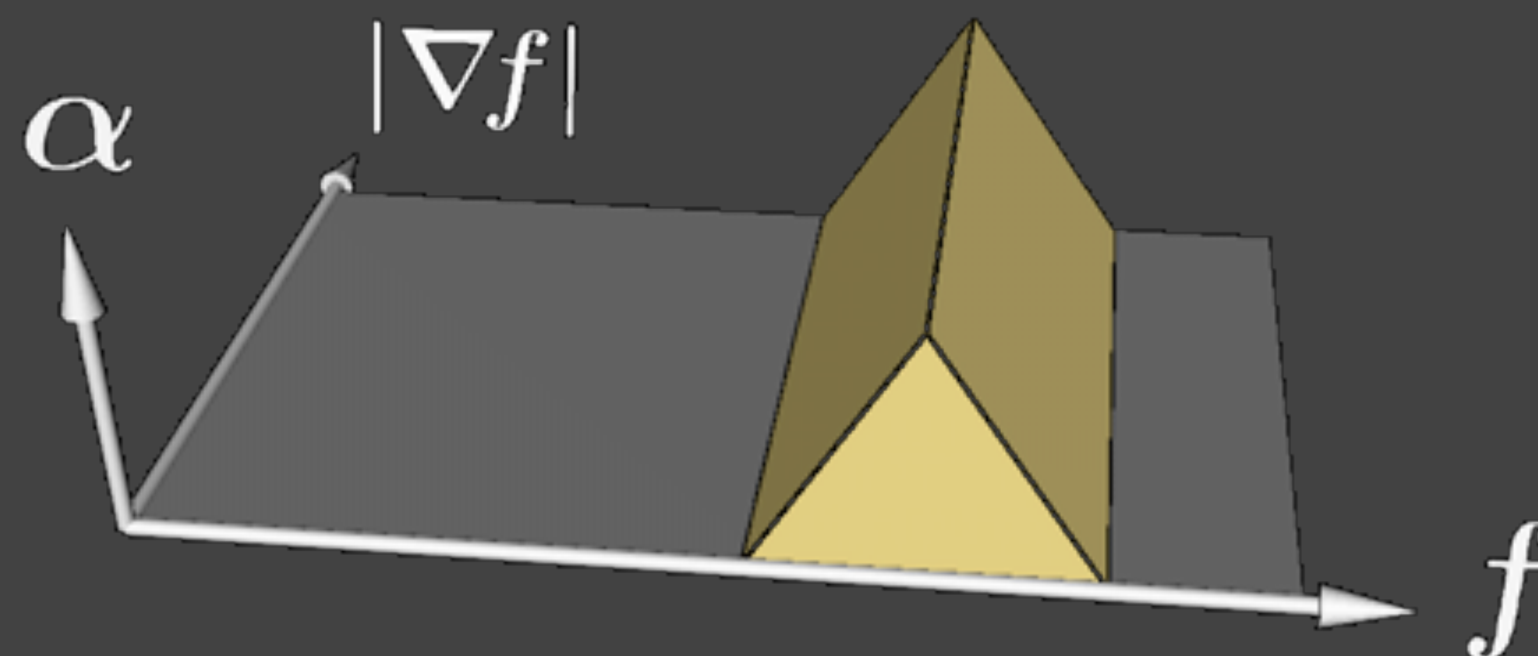
1D → 2D Transfer Function

$\mathrm{RGB}(f)$
$\alpha(f)$
} Generalize…



2D Transfer Function

$$\left.\begin{array}{r}\mathrm{RGB}(f)\\ \alpha(f)\end{array}\right\}$$ Generalize…

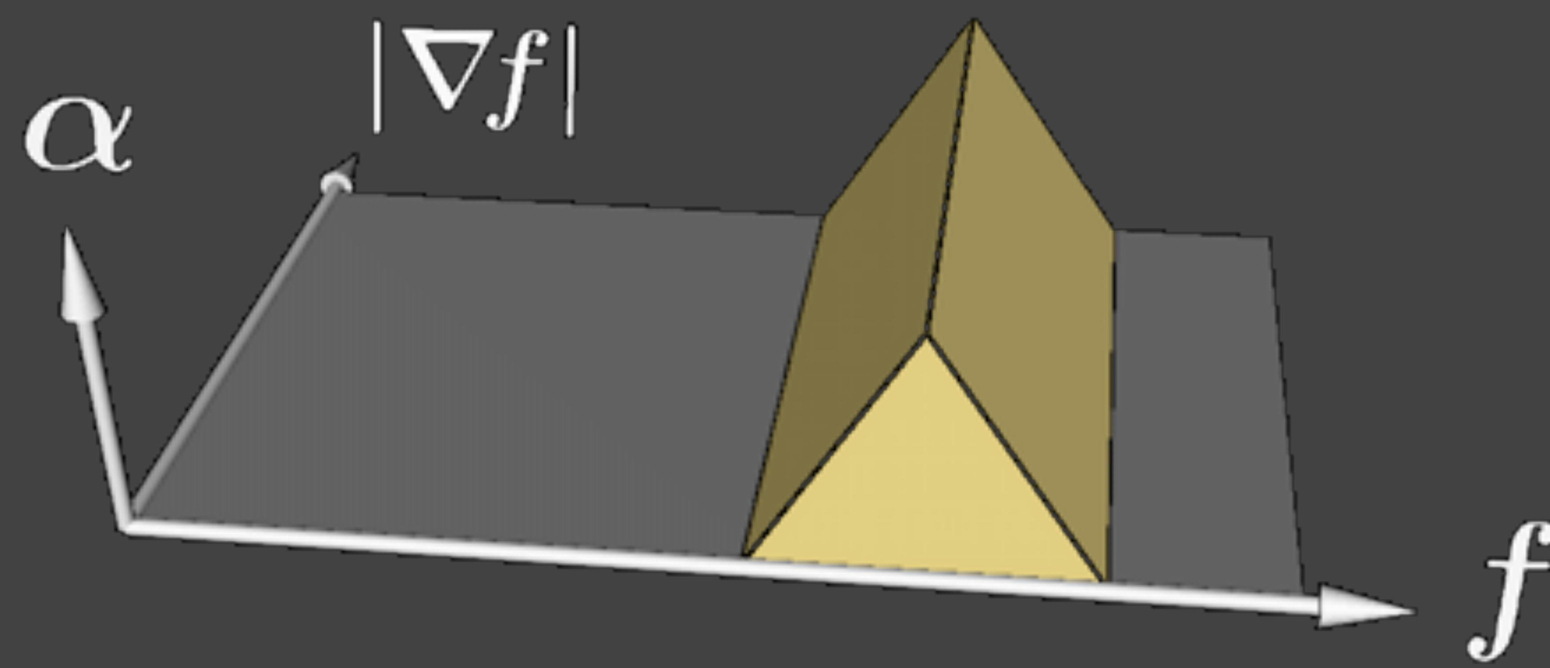


$$\rightarrow \begin{array}{r}\mathrm{RGB}(f,|\nabla f|)\\ \alpha(f,|\nabla f|)\end{array}$$

2D Transfer Function

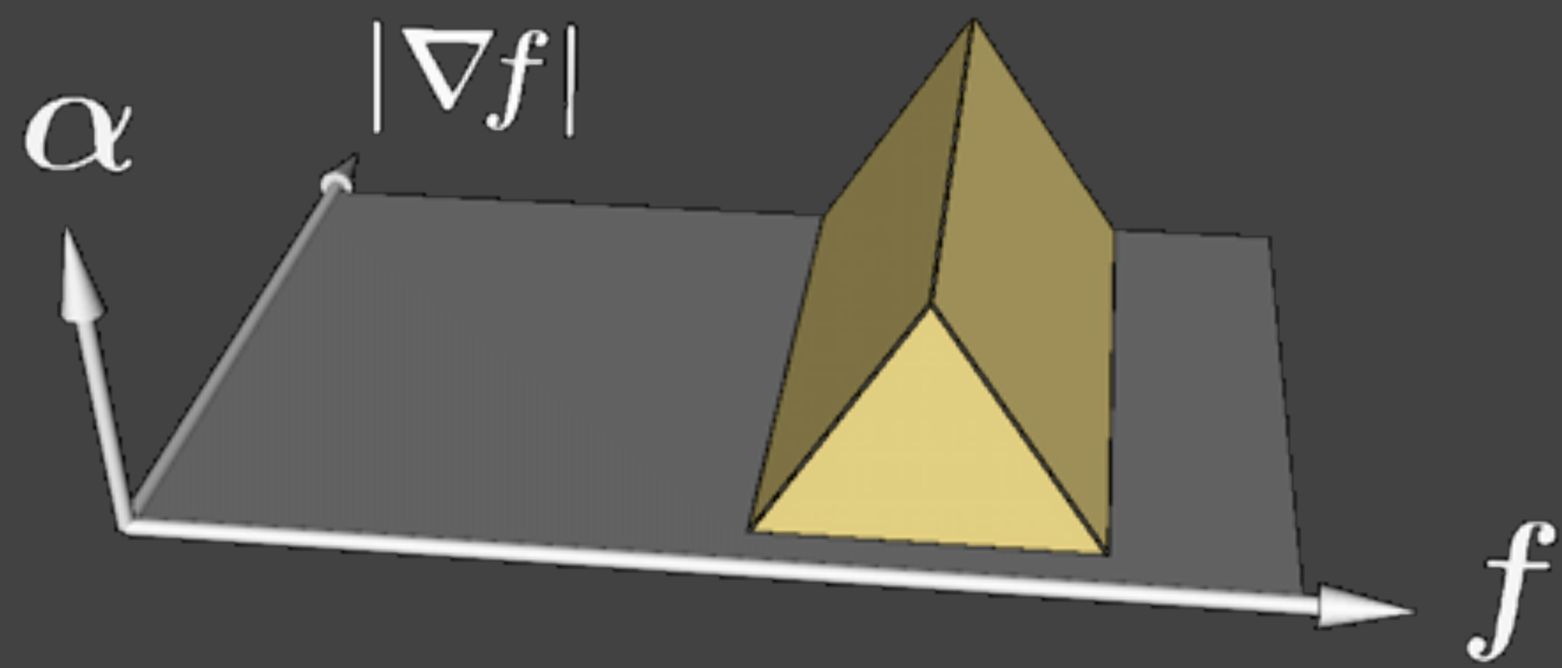
$$\text{RGB}(f, |\nabla f|)$$
$$\alpha(f, |\nabla f|)$$

Modify…



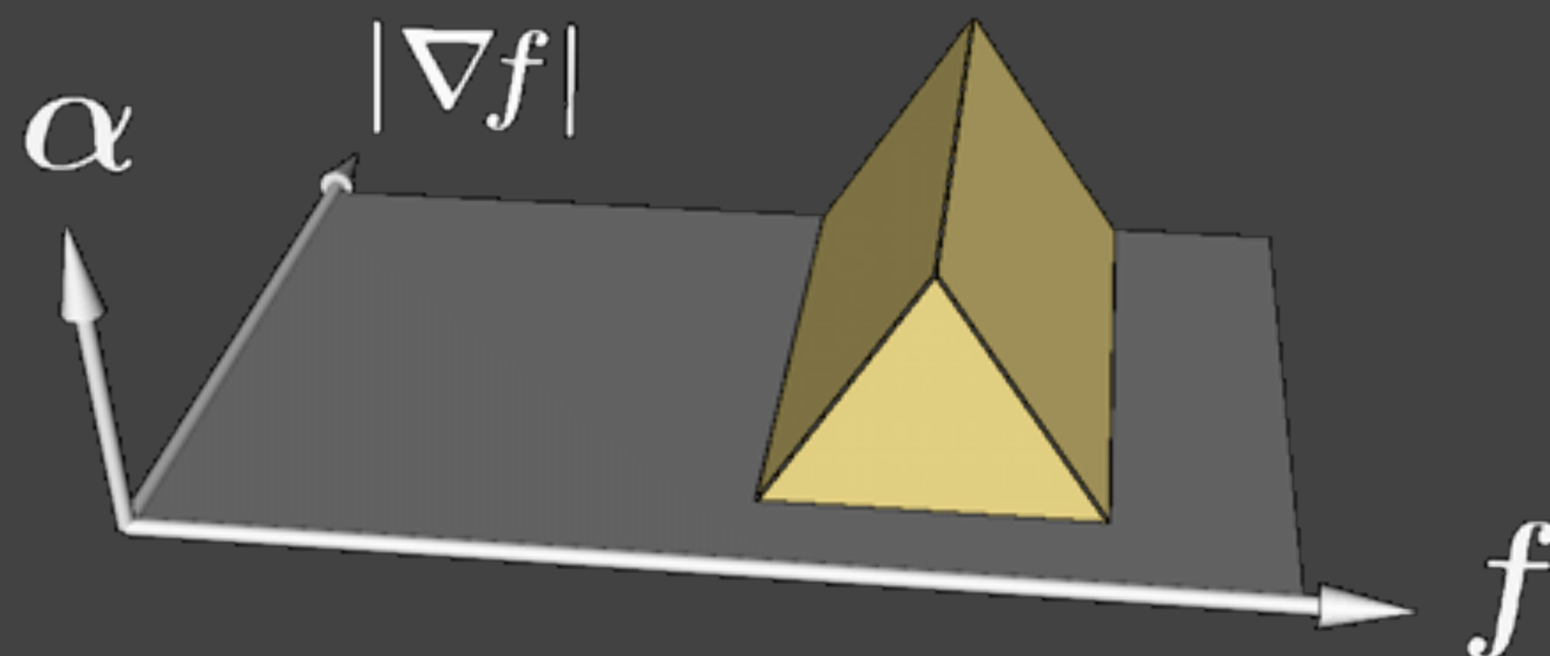
2D Transfer Function

$$\left.\begin{aligned}\mathrm{RGB}(f, |\nabla f|)\\ \alpha(f, |\nabla f|)\end{aligned}\right\}\ \text{Modify...}$$



2D Transfer Function

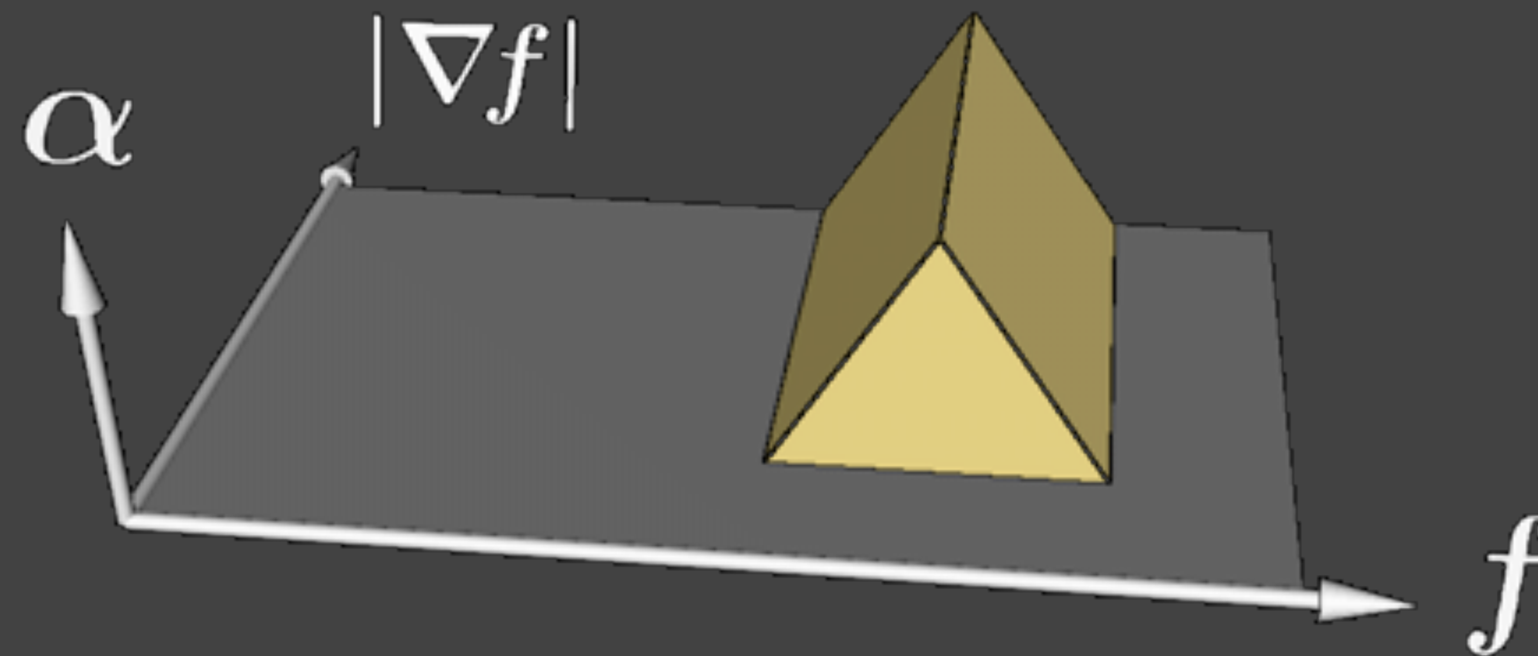
$$\left.\begin{array}{r} \mathrm{RGB}(f, |\nabla f|) \\ \alpha(f, |\nabla f|) \end{array}\right\} \text{Modify...}$$



2D Transfer Function

$$\mathrm{RGB}(f, |\nabla f|)$$
$$\alpha(f, |\nabla f|)$$

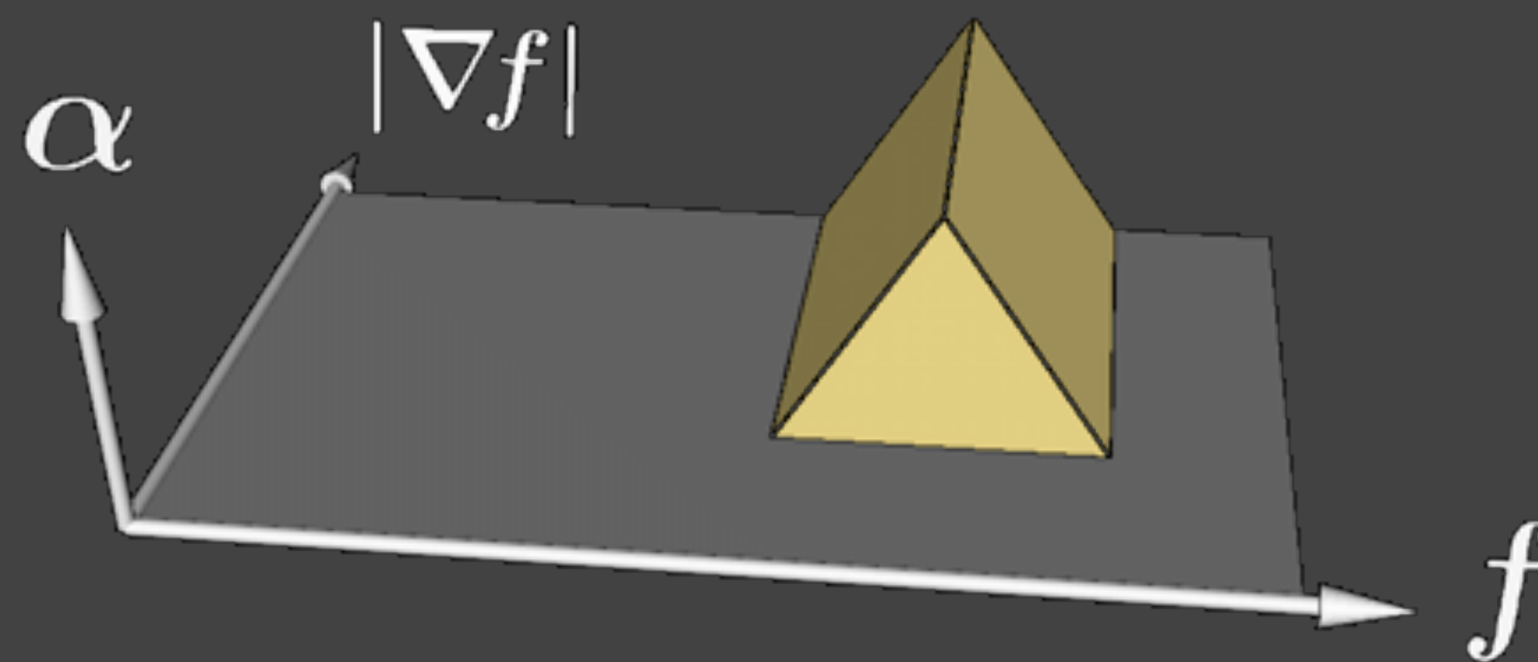
} Modify…



2D Transfer Function

$$\mathrm{RGB}(f, |\nabla f|)$$
$$\alpha(f, |\nabla f|)$$

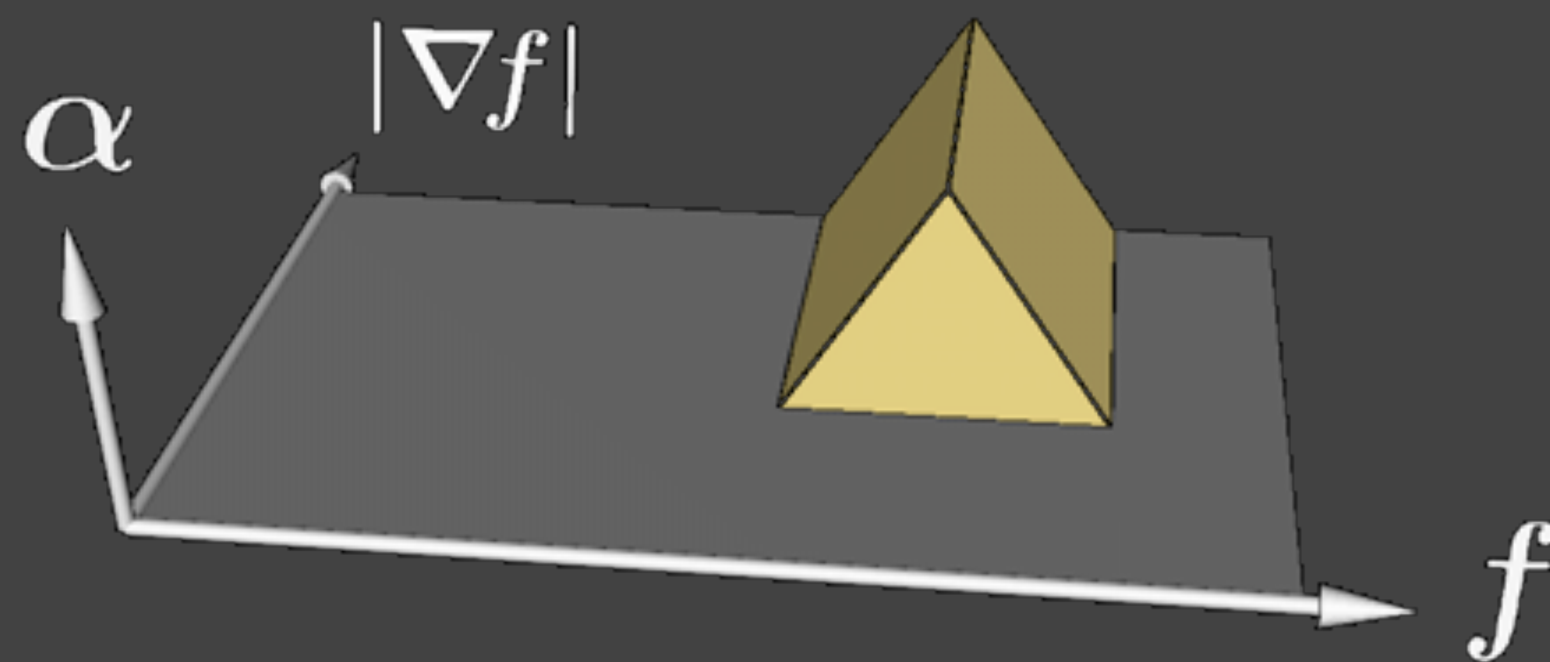
Modify…



2D Transfer Function

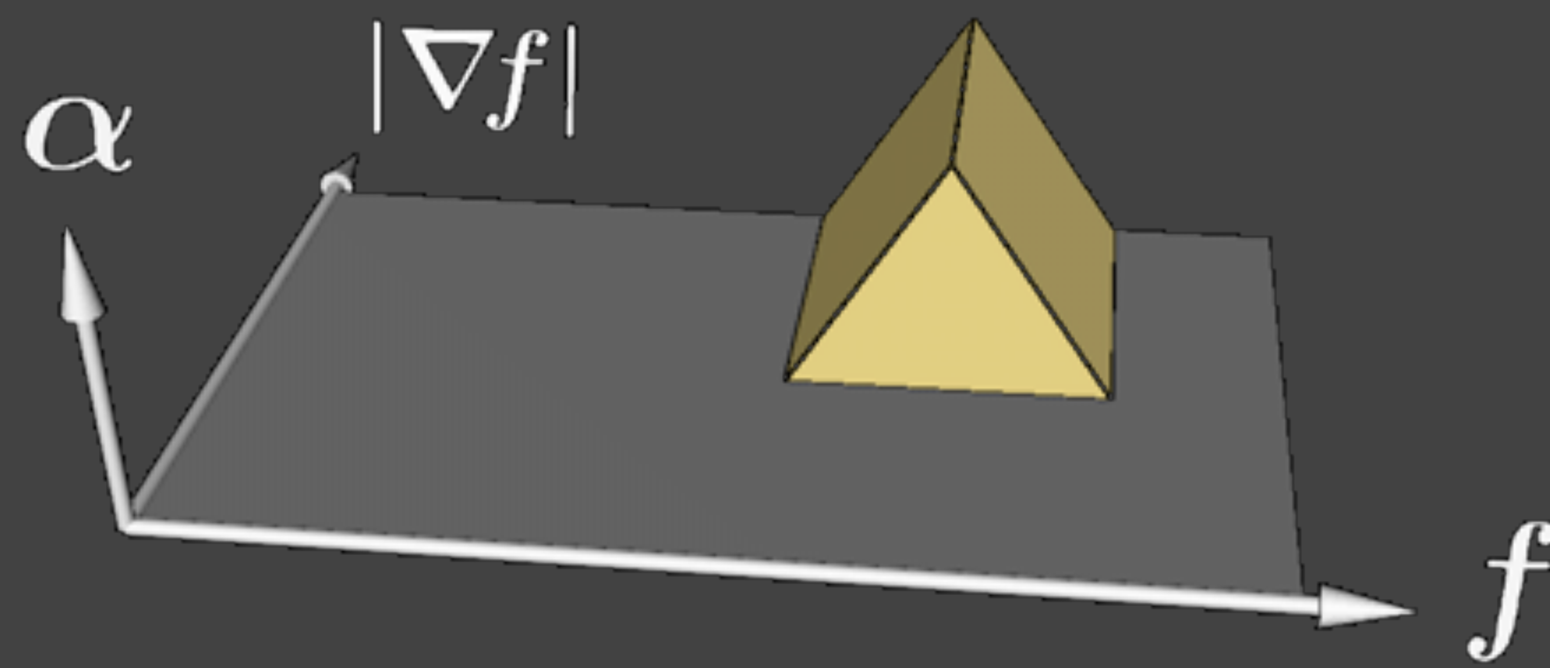
$$RGB(f, |\nabla f|)$$
$$\alpha(f, |\nabla f|)$$

Modify…

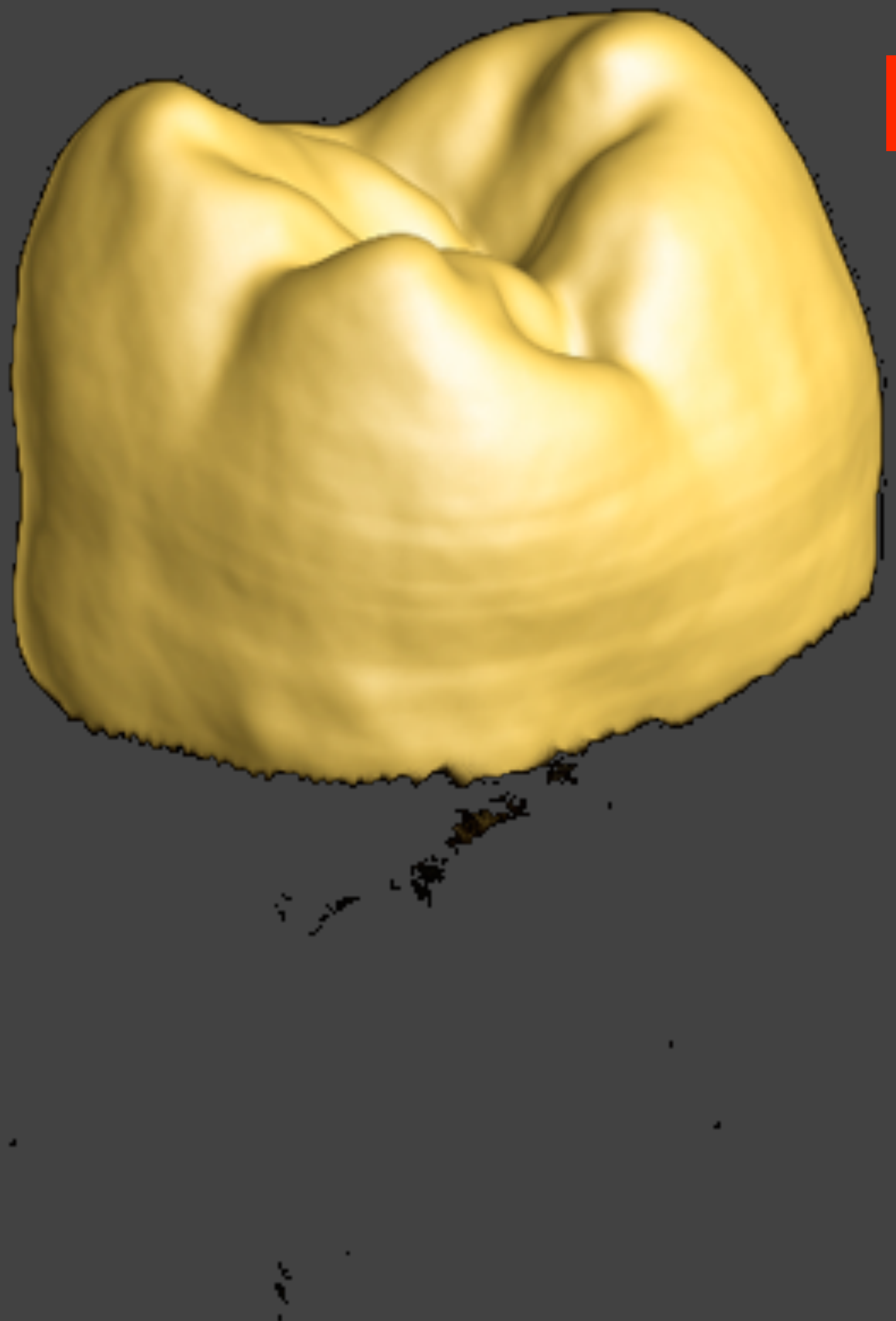


2D Transfer Function

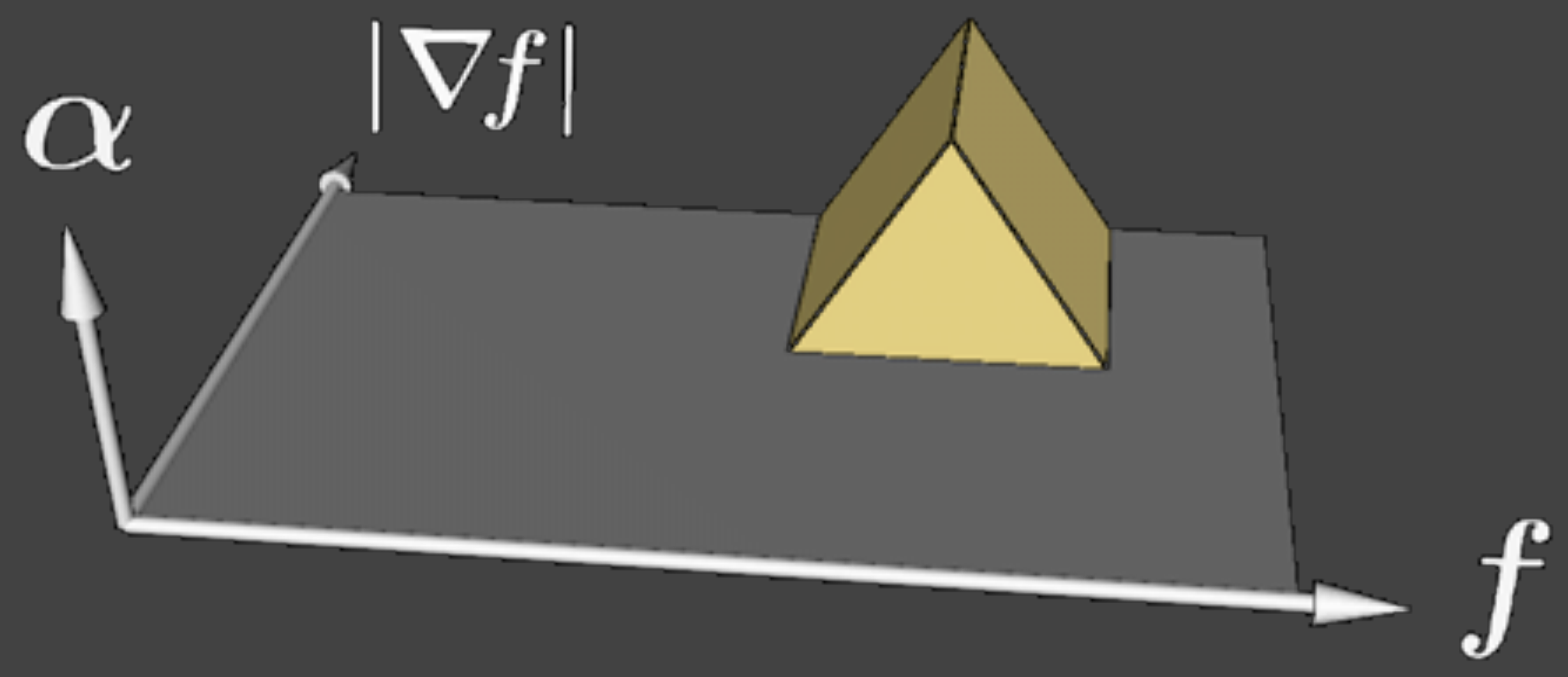
$$\left.\begin{array}{r}\mathrm{RGB}(f,|\nabla f|)\\ \alpha(f,|\nabla f|)\end{array}\right\}\text{ Modify…}$$



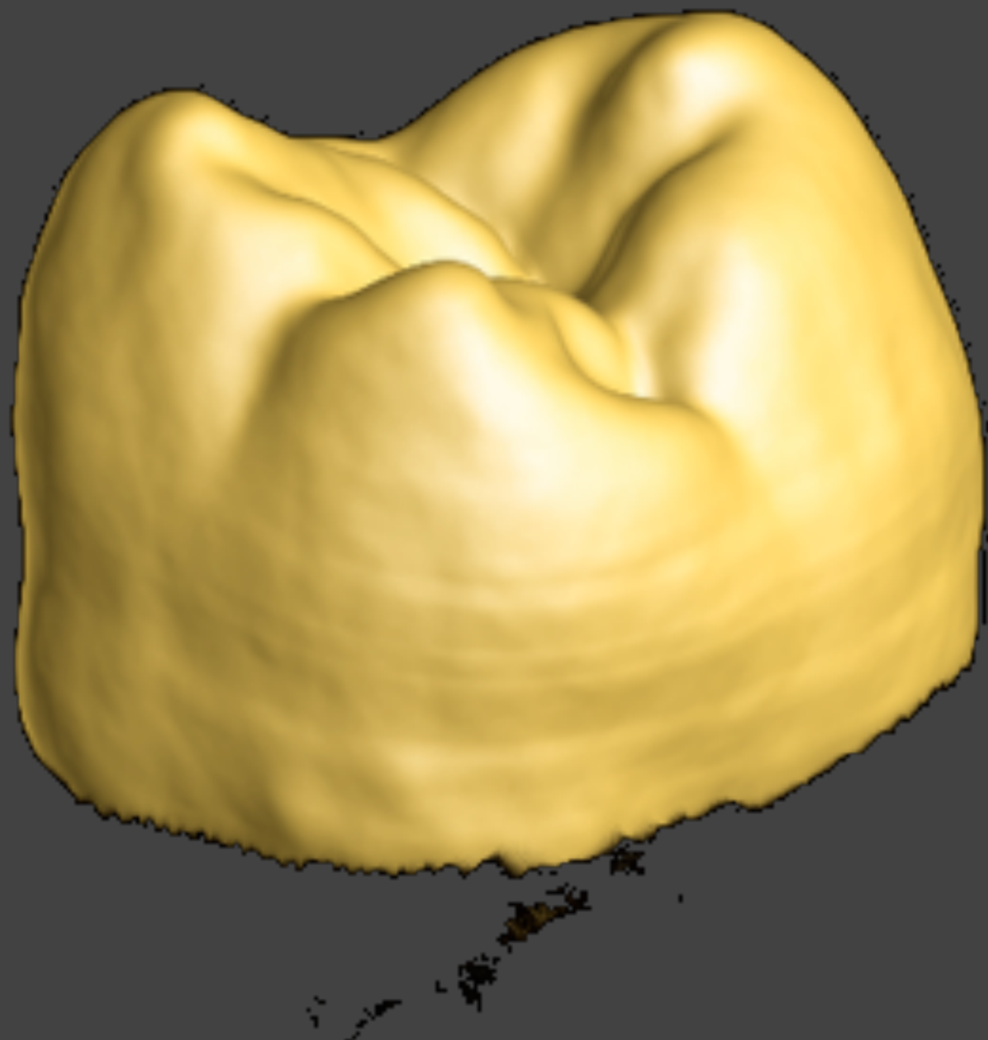
2D Transfer Function



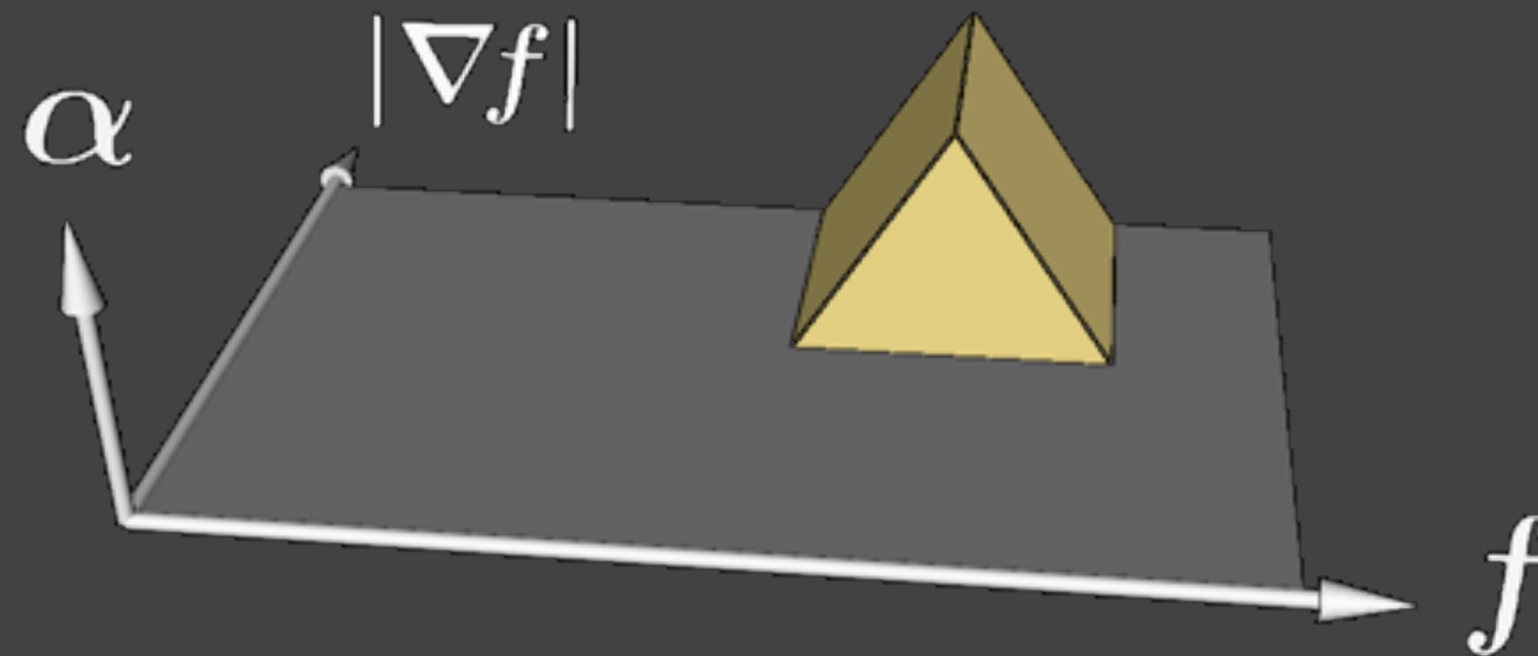
$$\left.\begin{array}{r} RGB(f, |\nabla f|) \\ \alpha(f, |\nabla f|) \end{array}\right\} \text{Modify...}$$



2D Transfer Function

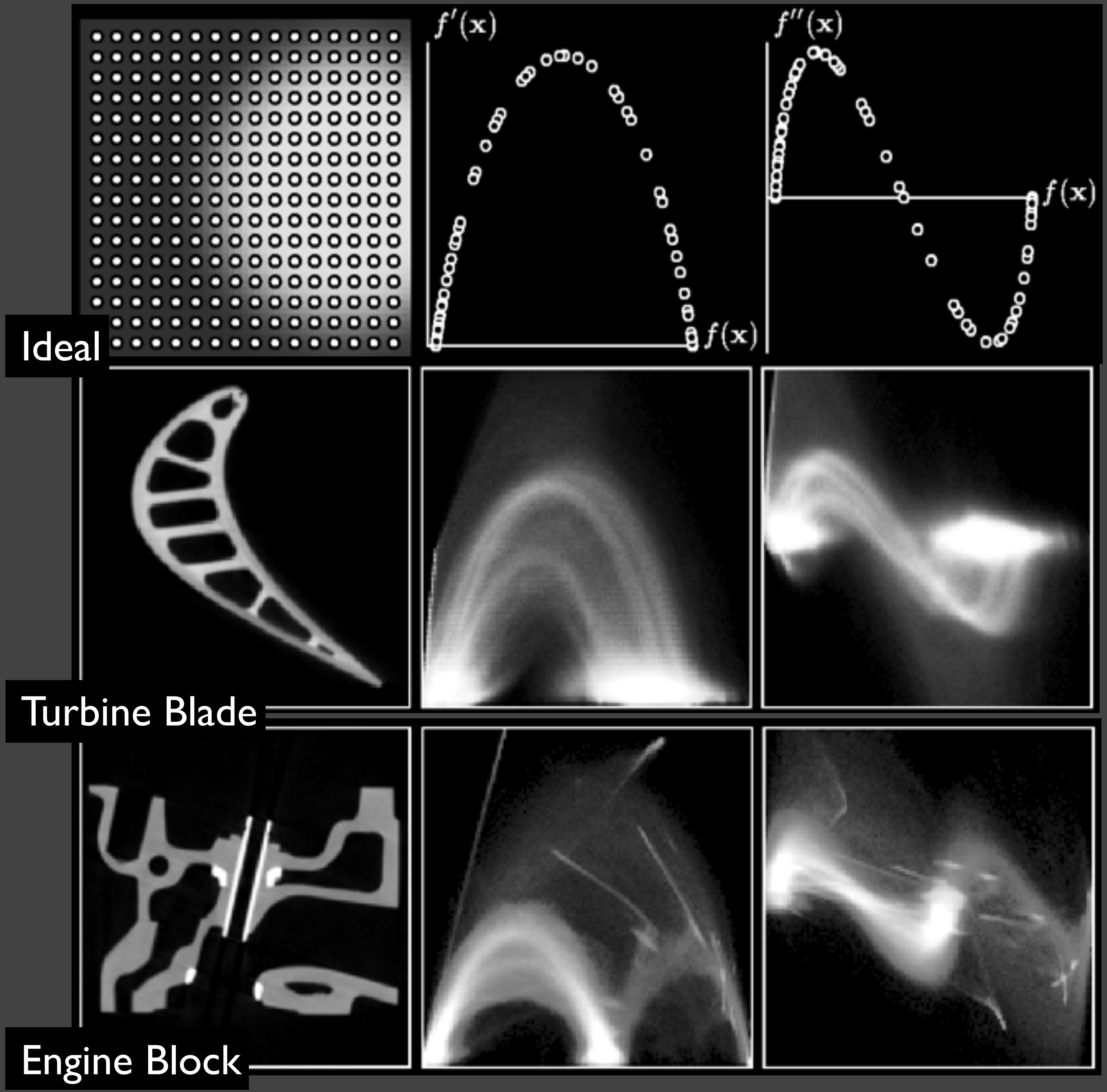


$$\left.\begin{aligned} \mathrm{RGB}(f, |\nabla f|) \\ \alpha(f, |\nabla f|) \end{aligned}\right\}$$ Modify…



2D transfer functions give greater flexibility in boundary visualization

Display of Surfaces from Volume Data, Levoy 1988

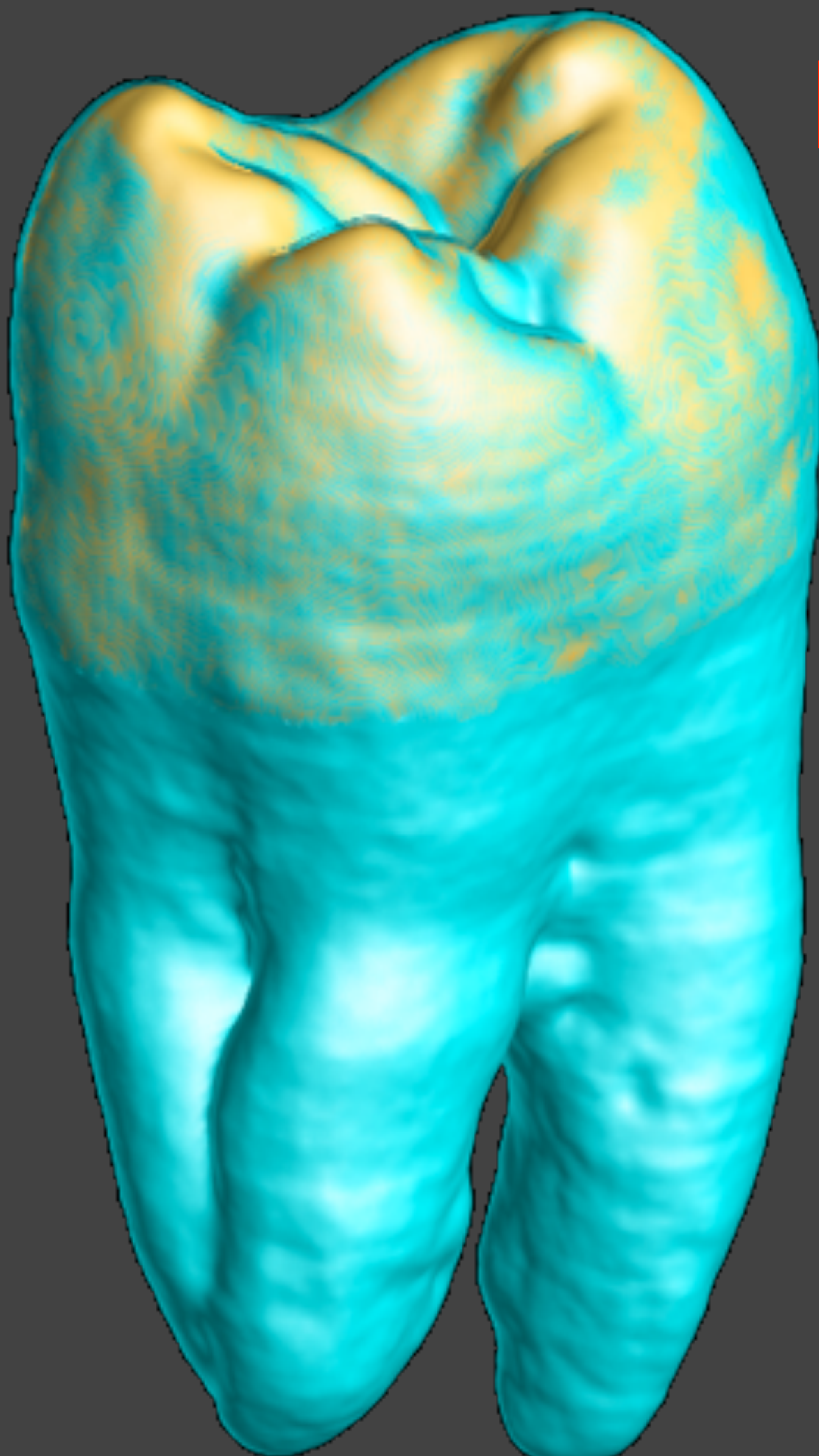


Project histogram volume to 2D scatterplots

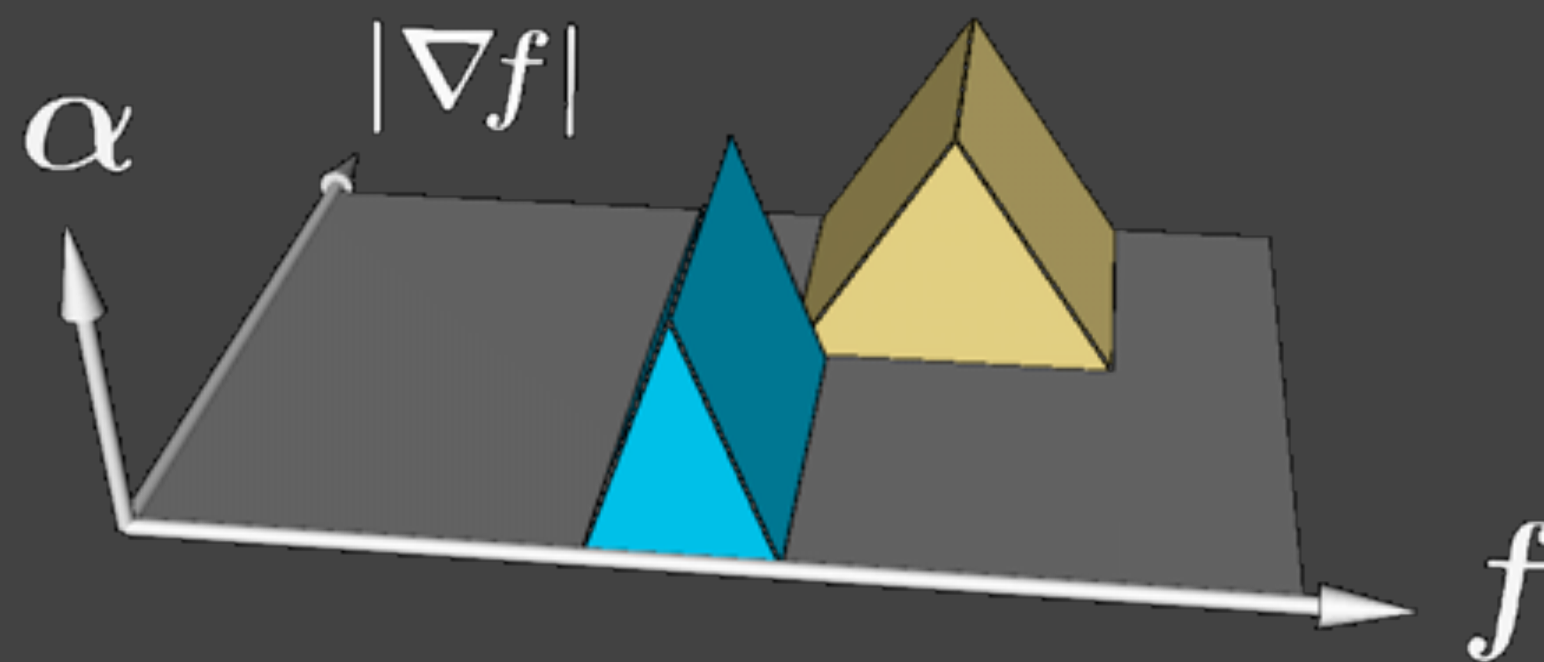
- Visual summary
- Interpreted for TF guidance
- No reliance on boundary model at this stage

DEMO:
2D TRANSFER FUNCTION

2D Transfer Function

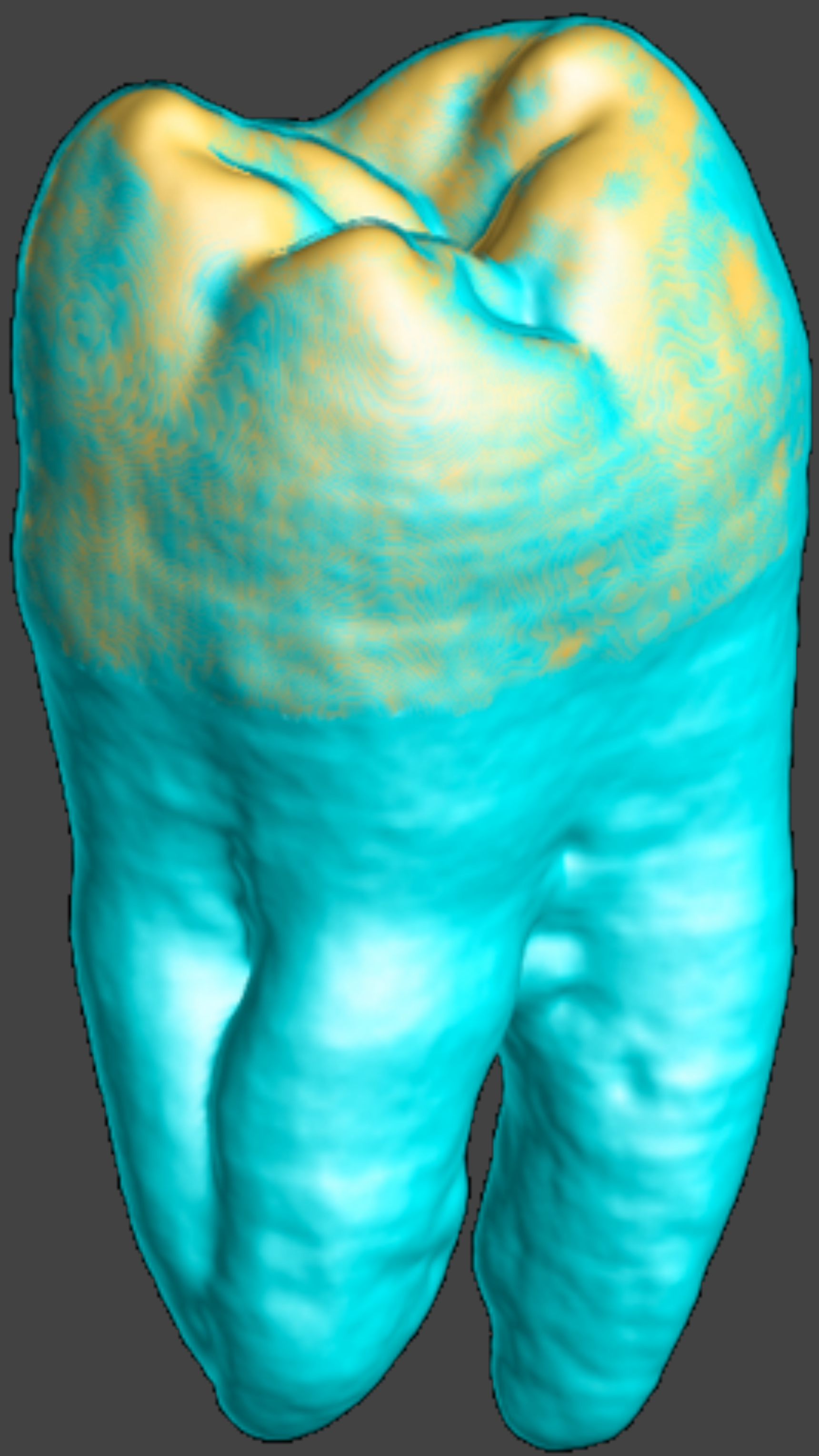


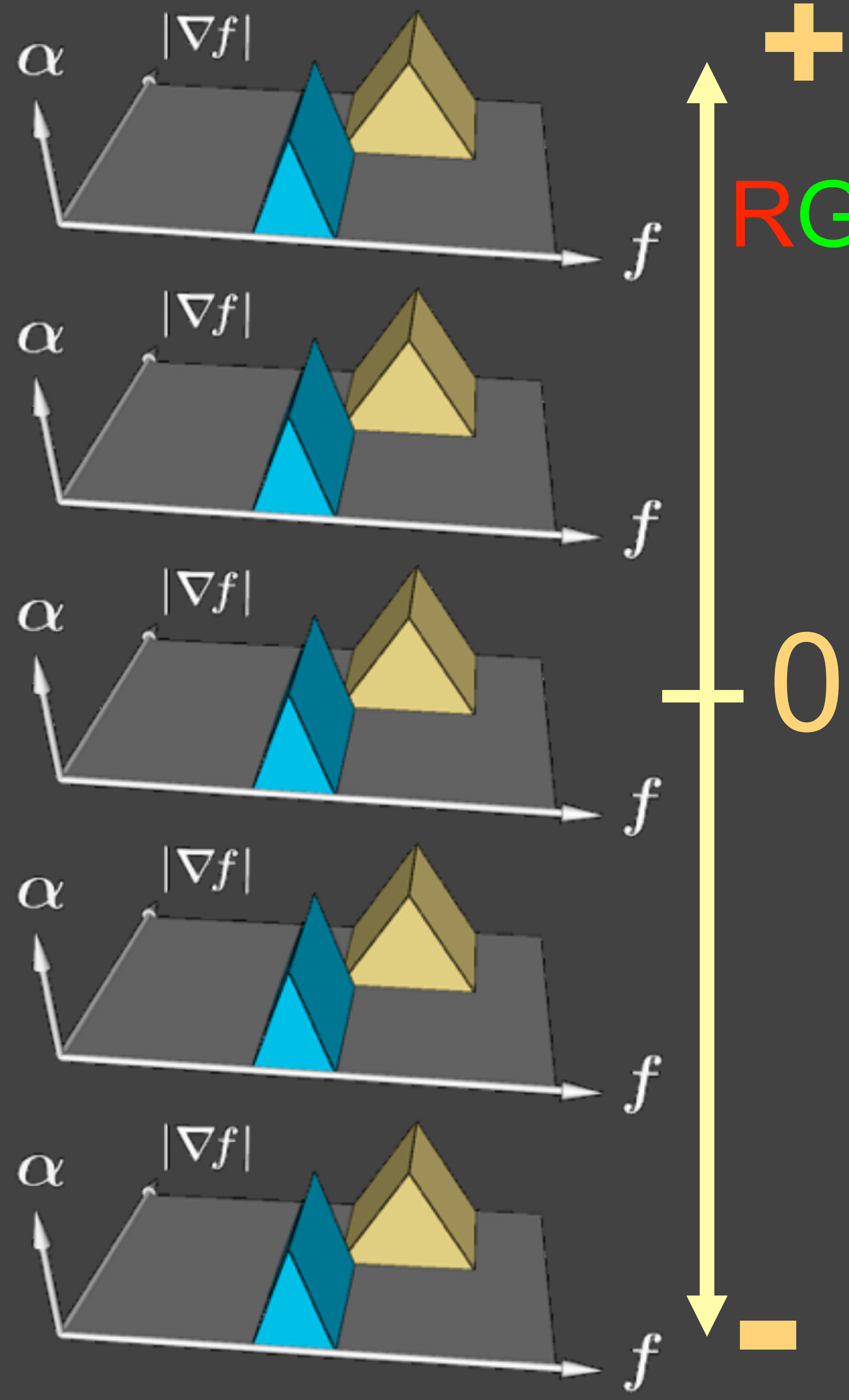
$$\left.\begin{matrix} RGB(f, |\nabla f|) \\ \alpha(f, |\nabla f|) \end{matrix}\right\} \text{Modify…}$$



Trying to reintroduce dentin / background boundary …

2D → 3D Transfer Function





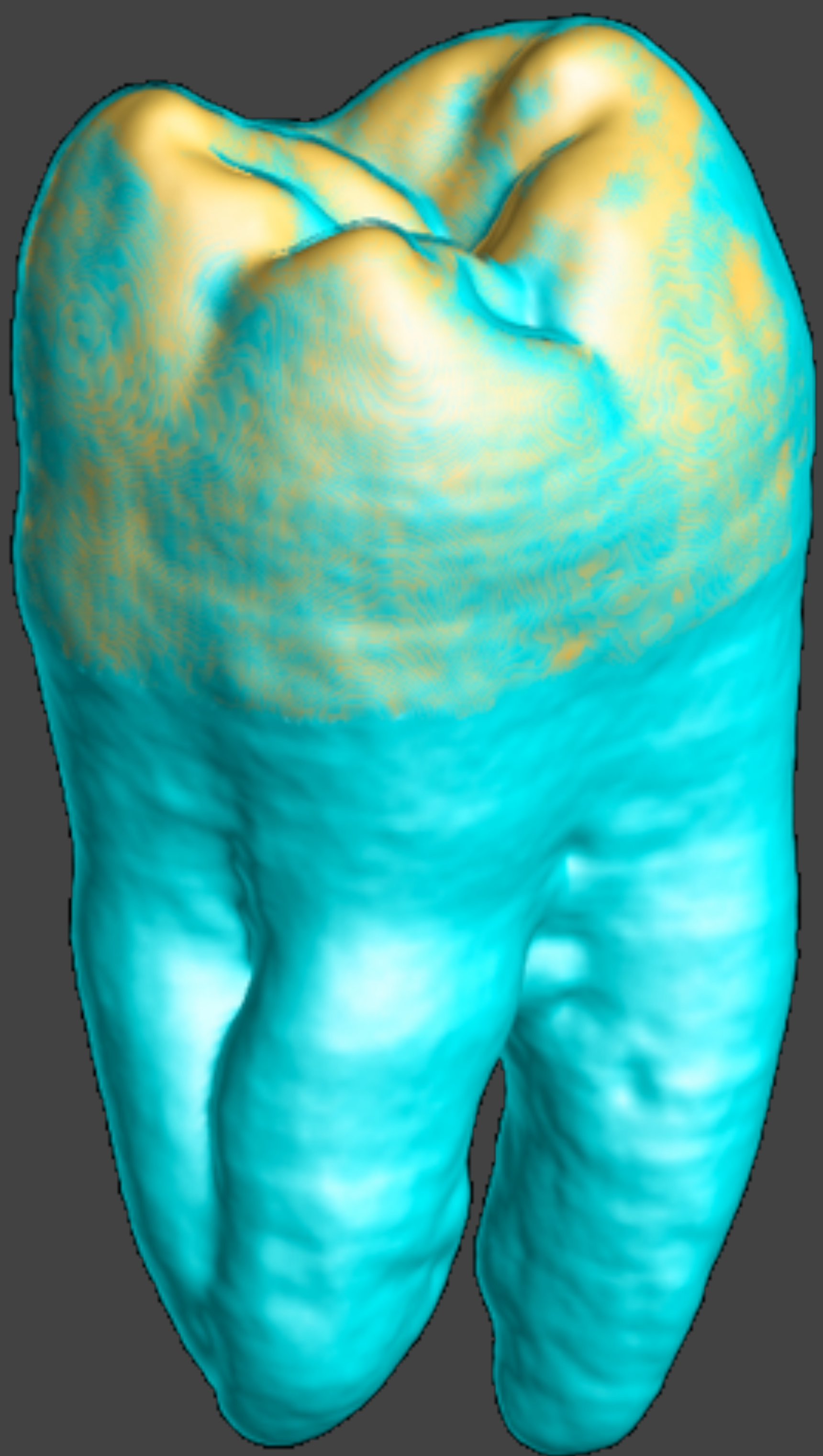
RGB $\alpha(f, |\nabla f|, D^2_{\widehat{\nabla f}} f)$

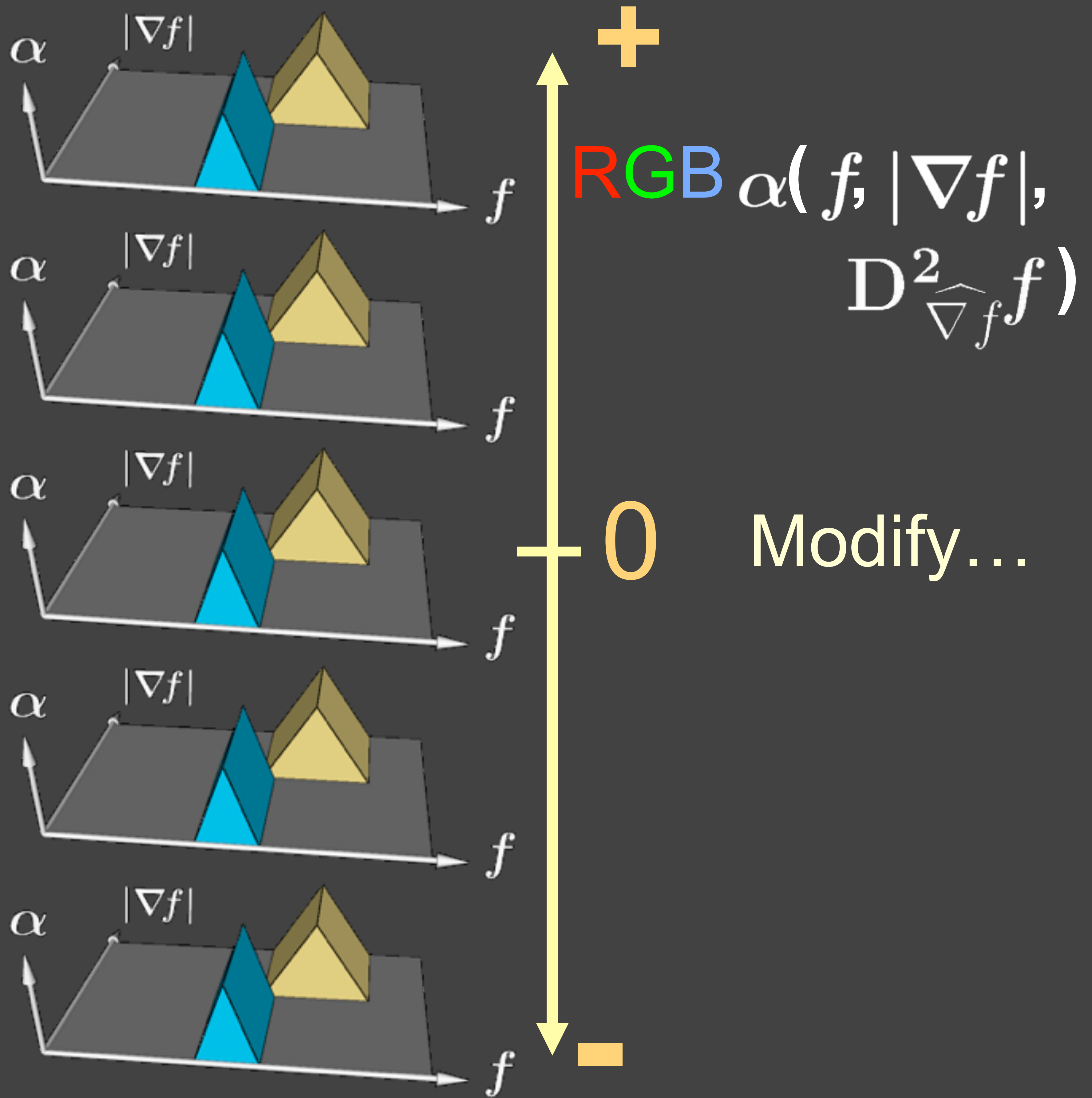
Second directional derivative

$D^2_{\widehat{\nabla f}} f$

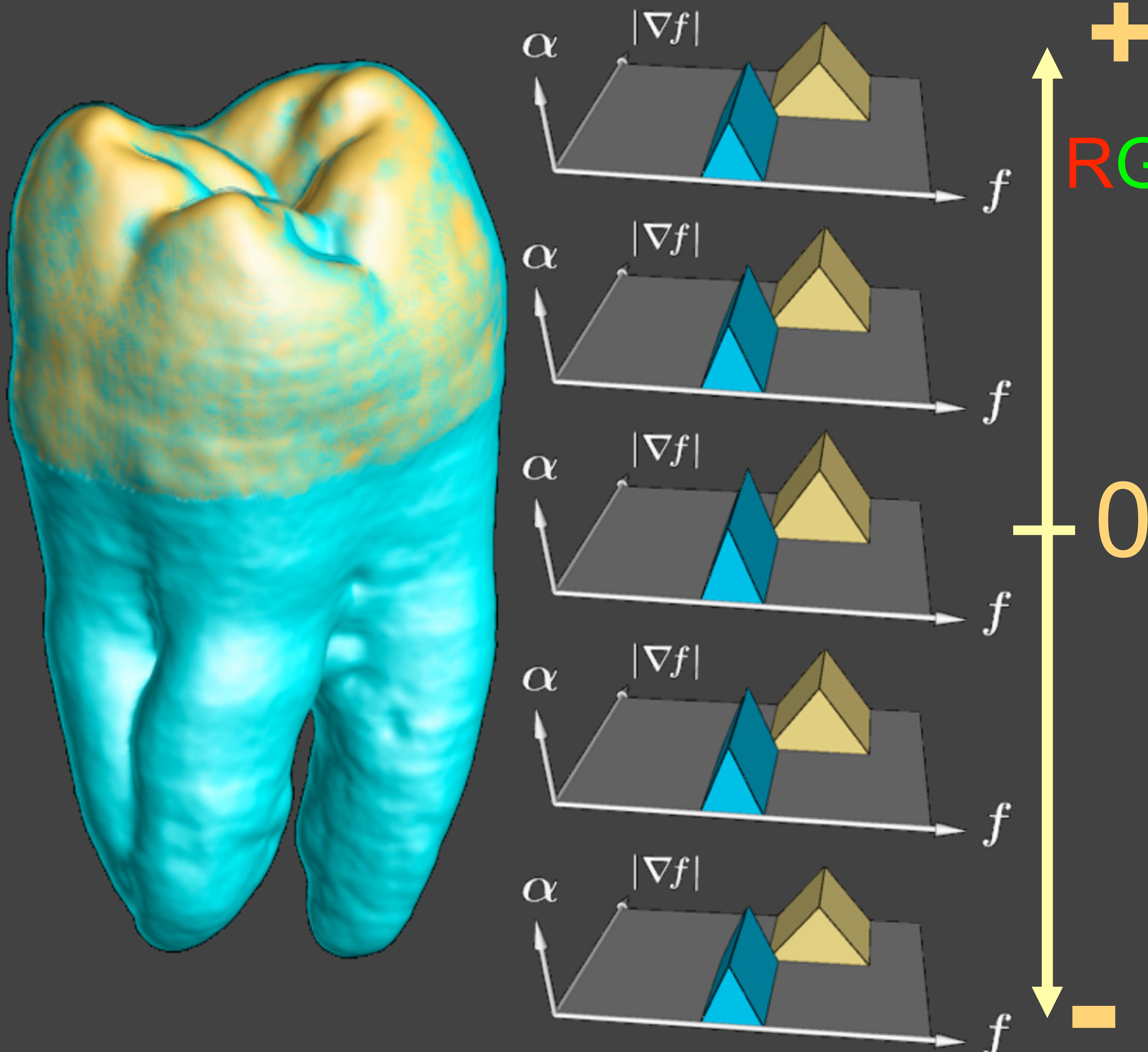
measured with Hessian

3D Transfer Function





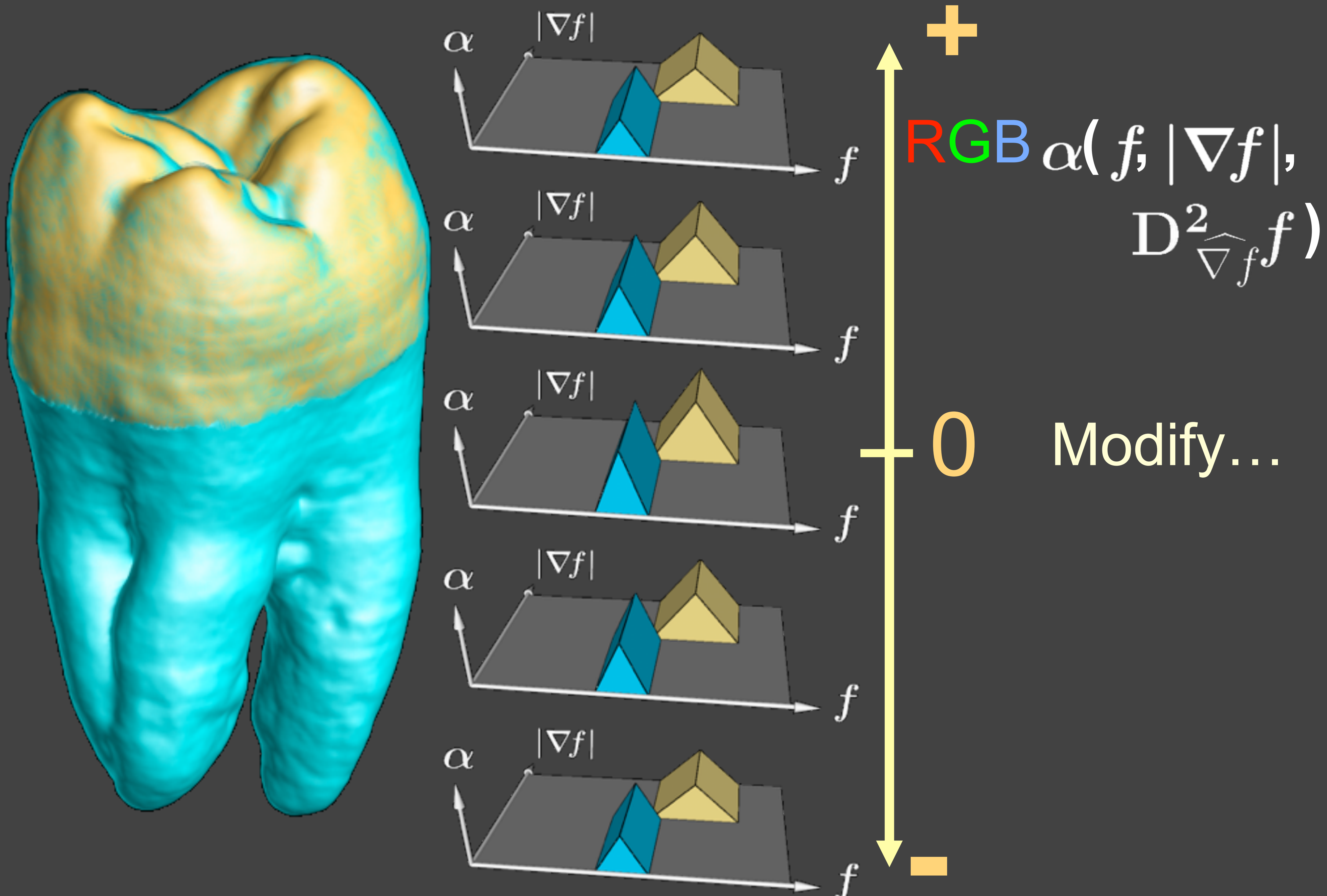
3D Transfer Function



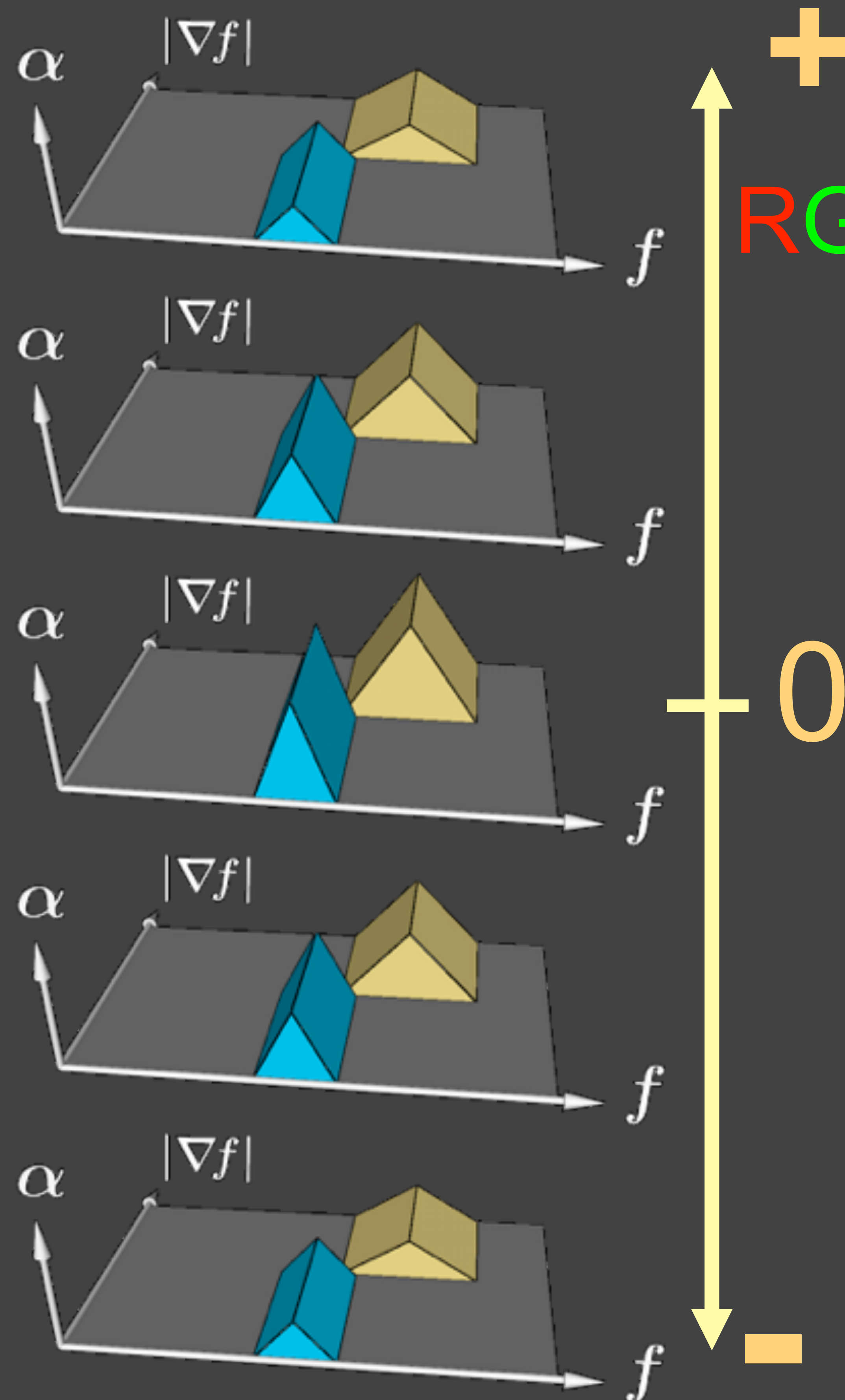
RGB $\alpha(f, |\nabla f|, D^2_{\widehat{\nabla f}} f)$

Modify…

3D Transfer Function



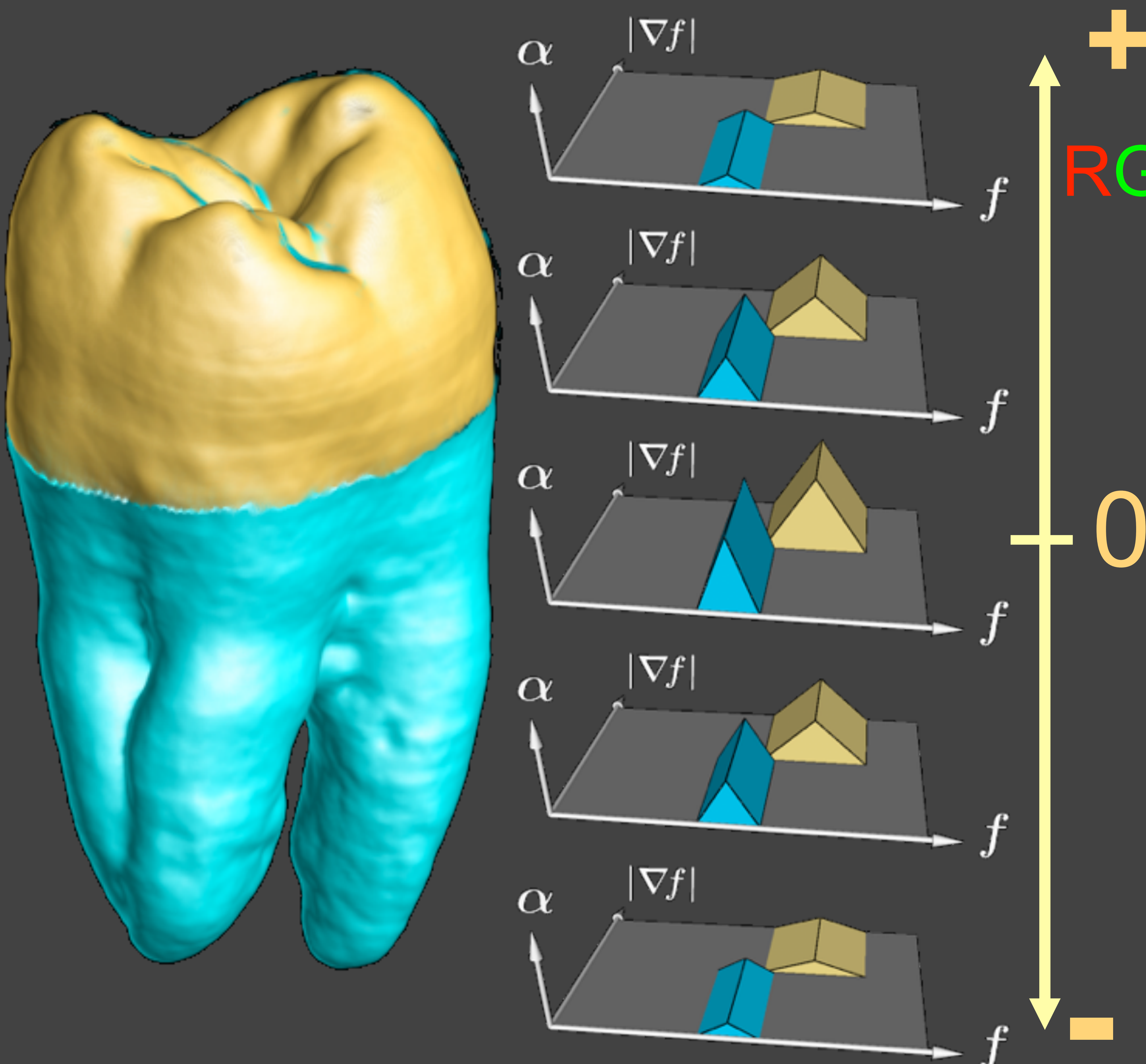
3D Transfer Function



RGB $\alpha(f, |\nabla f|, D^2_{\widehat{\nabla} f} f)$

Modify…

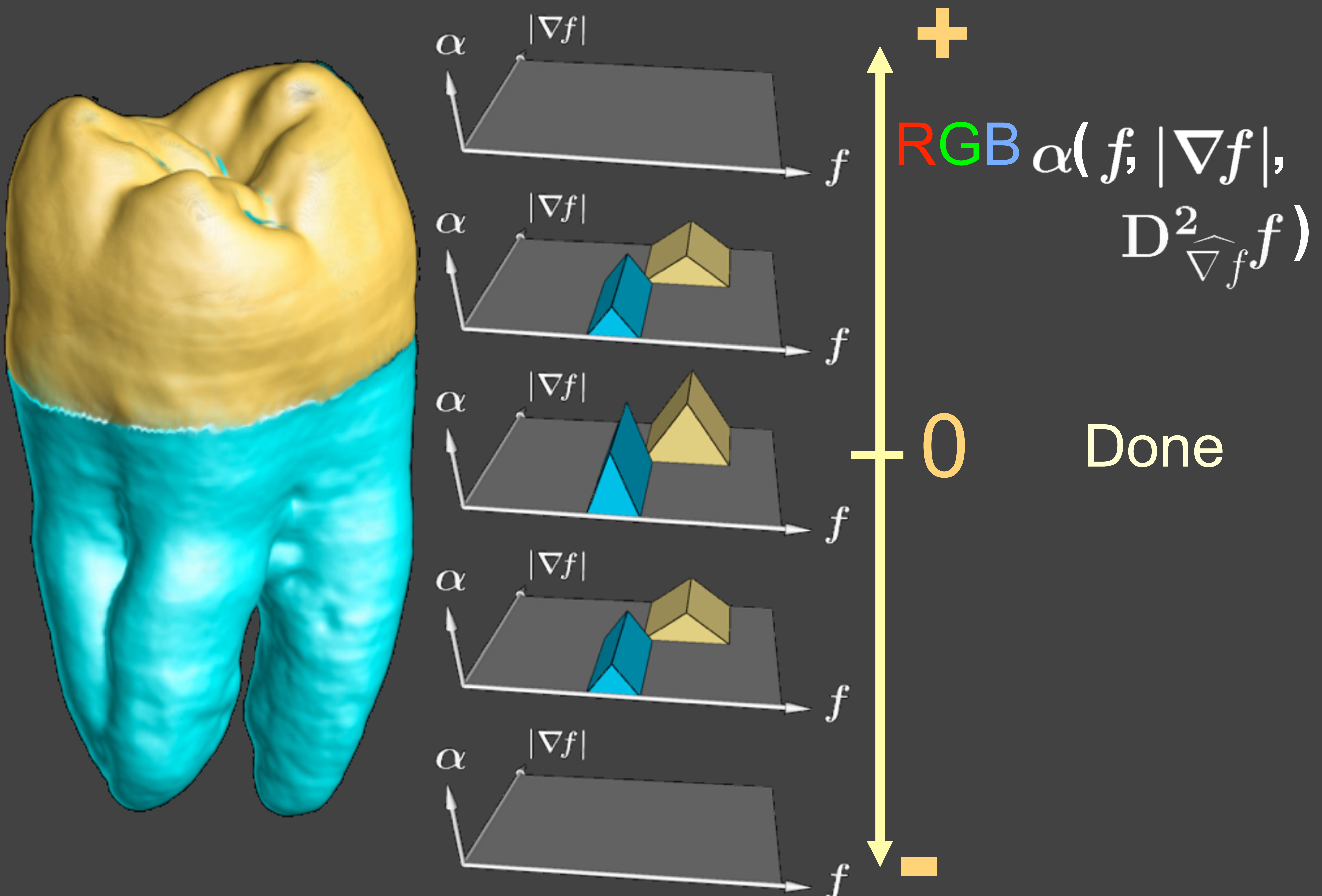
3D Transfer Function



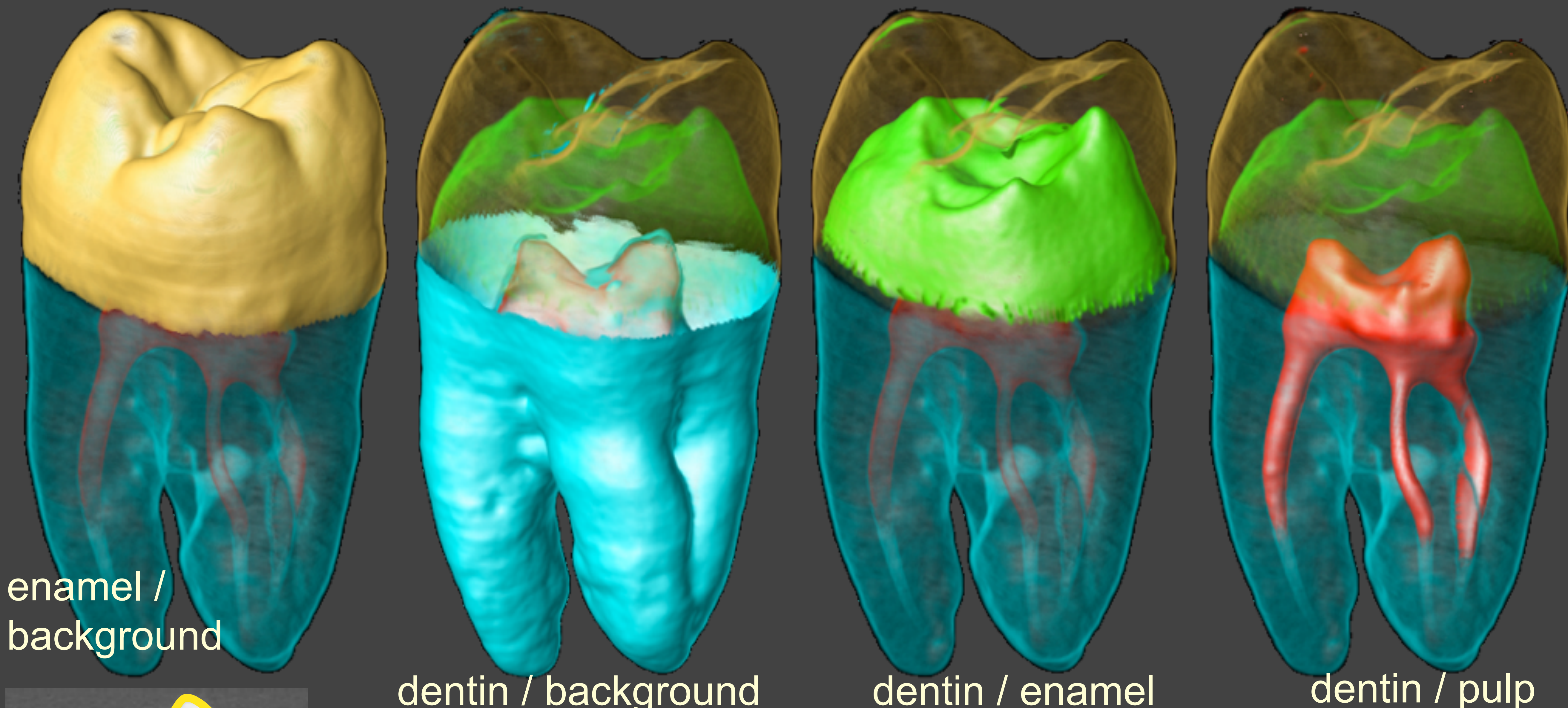
RGB $\alpha(f, |\nabla f|, D^2_{\widehat{\nabla} f} f)$

Modify…

3D Transfer Function



3D Transfer Function



1D: not possible

2D: specificity not as good

Multi-Dimensional TFs

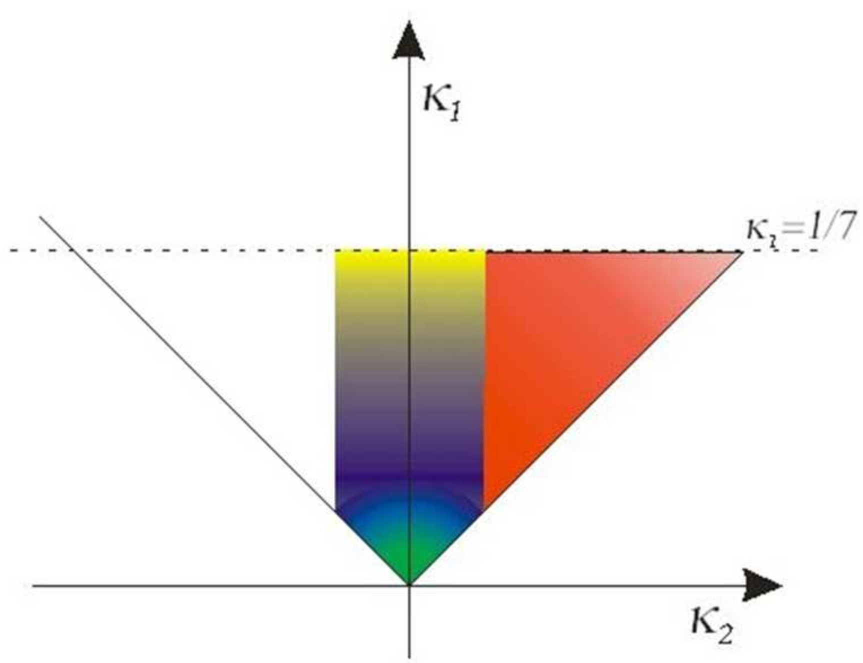
- Strengths:
  - Better flexibility, specificity
  - Higher quality visualizations
- Weaknesses:
  - Even harder to specify
  - Unintuitive relationship with boundaries
  - Greater demands on user interface

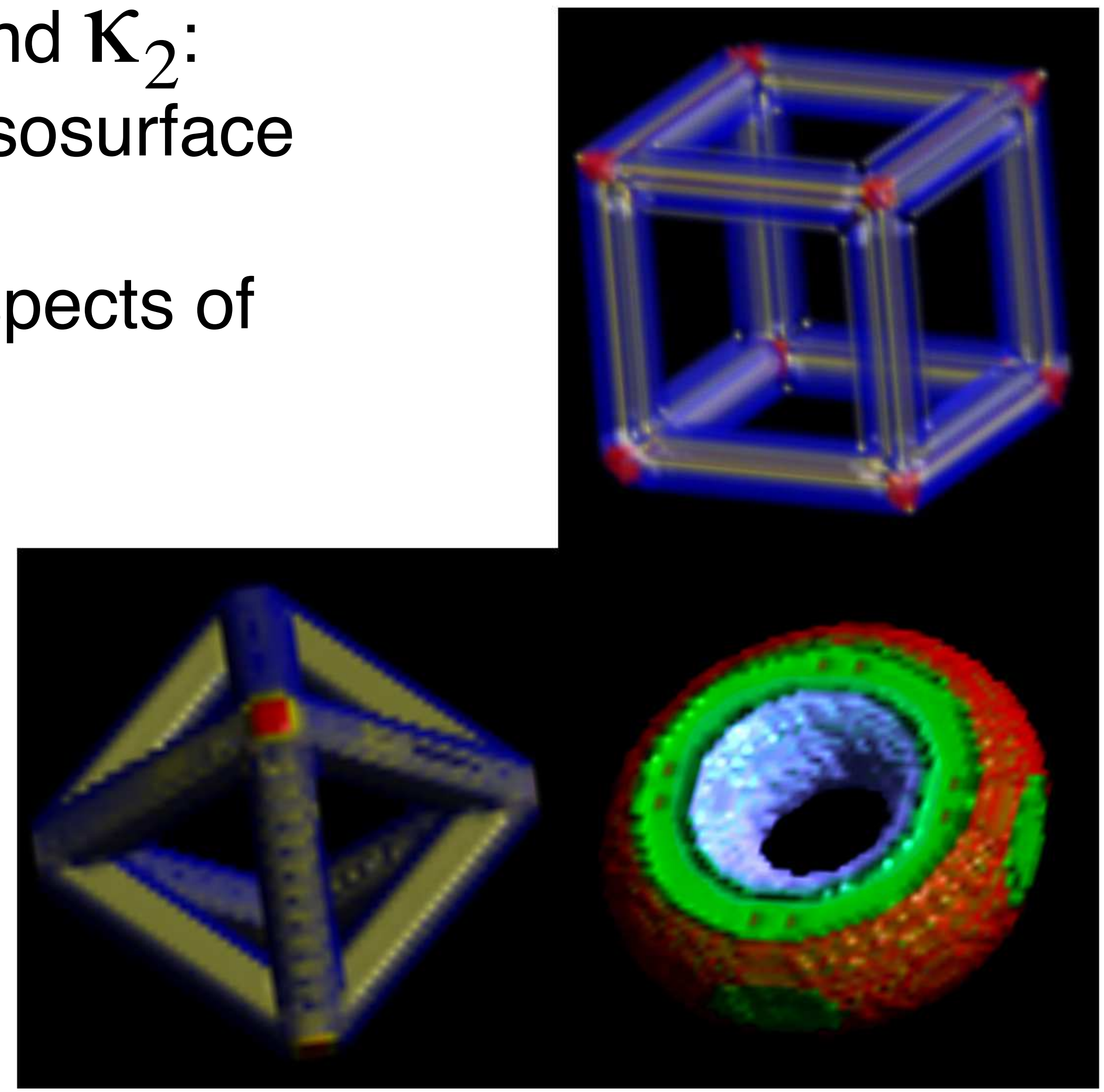
Some Other Methods

Curvature

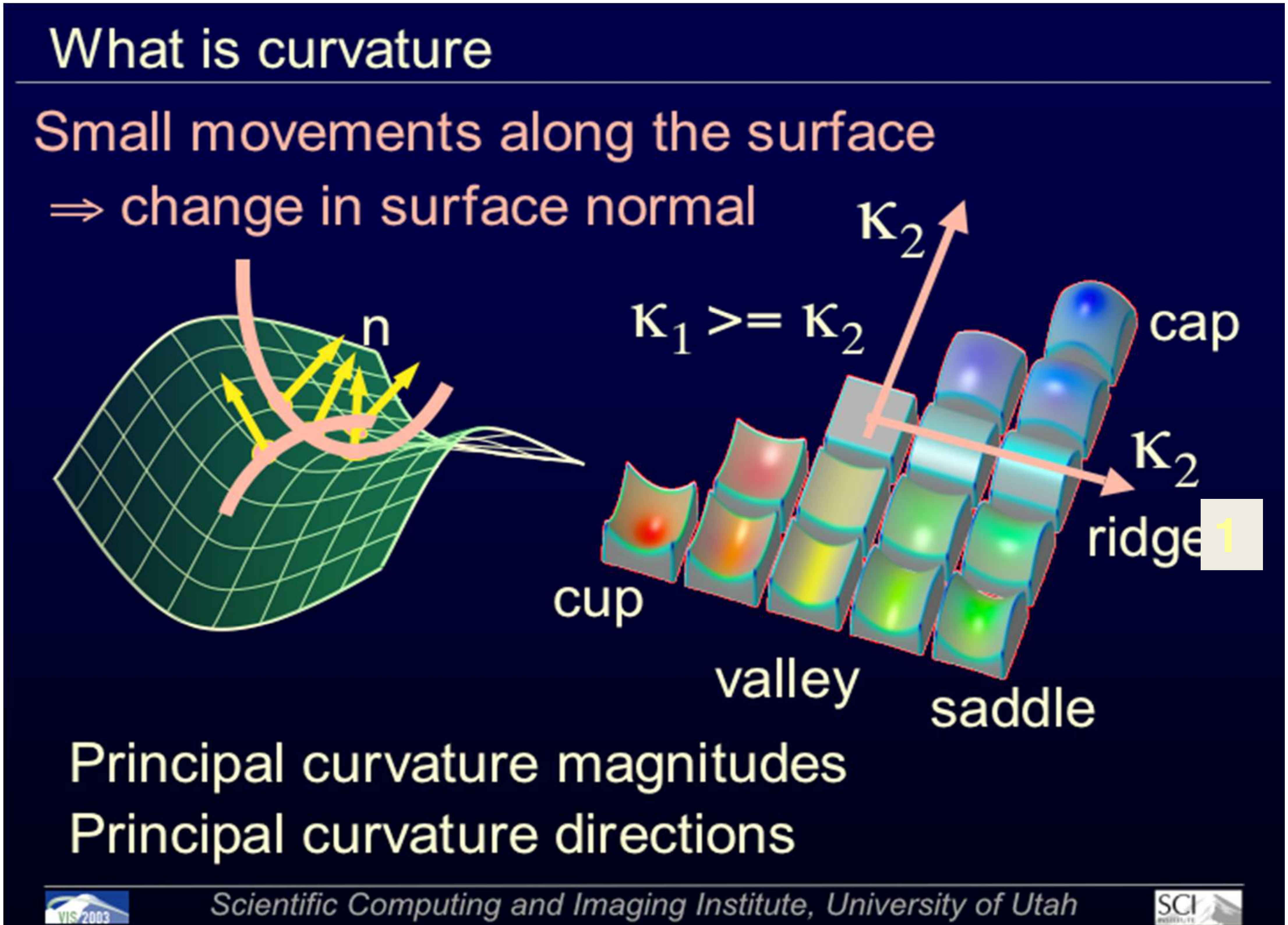
"Curvature-Based Transfer Functions for Direct Volume Rendering", Hladůvka, König, Gröller: SCCG '00

- Uses 2D space of κ_1 and κ_2: principal curvatures of isosurface at a given point
- Graphically indicates aspects of local shape
- Specification is simple

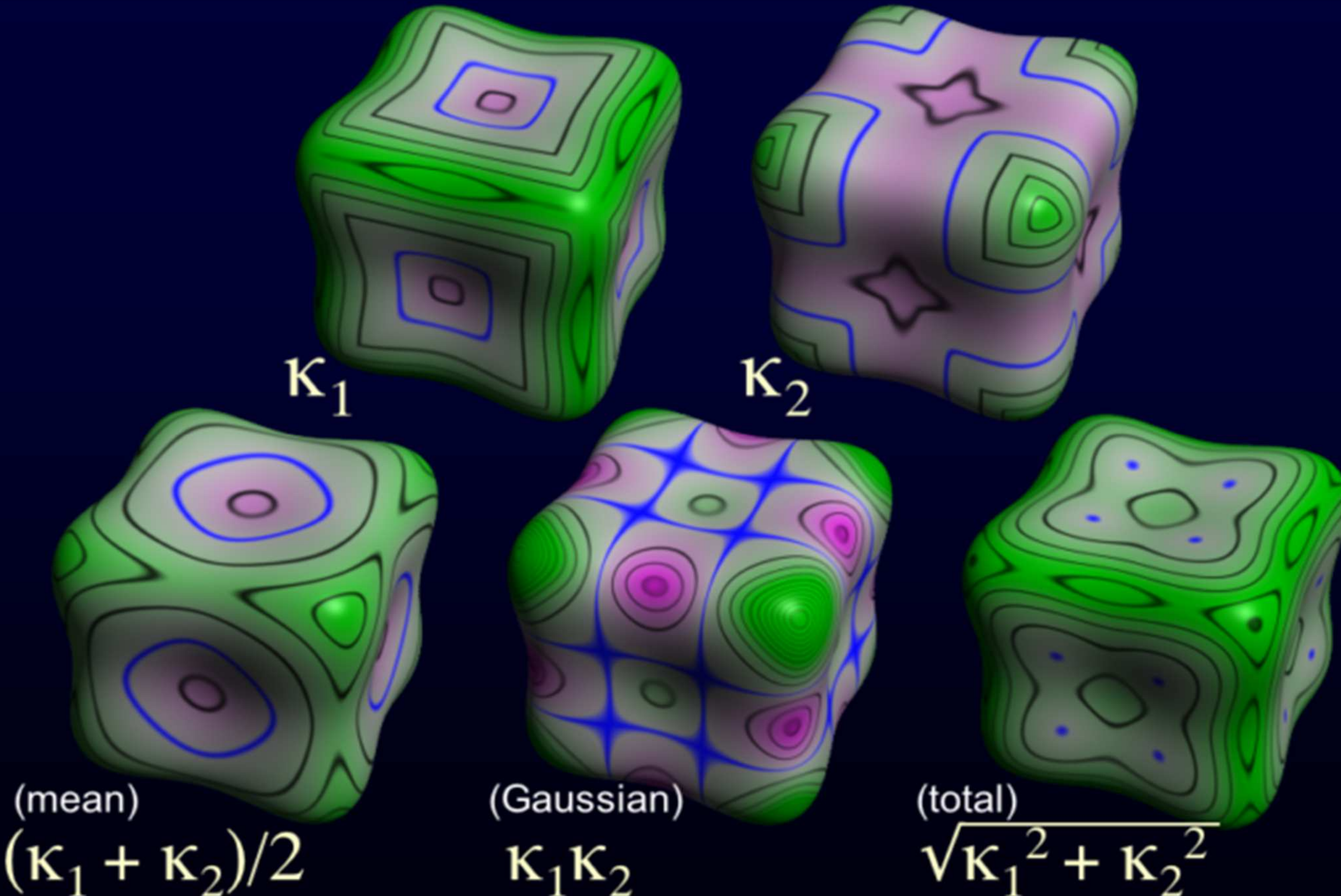




Curvature


What is curvature
Small movements along the surface
⇒ change in surface normal
n
κ1 >= κ2
κ2
κ2
cap
ridge
cup
valley
saddle
Principal curvature magnitudes
Principal curvature directions
VIS 2003
Scientific Computing and Imaging Institute, University of Utah
SCI

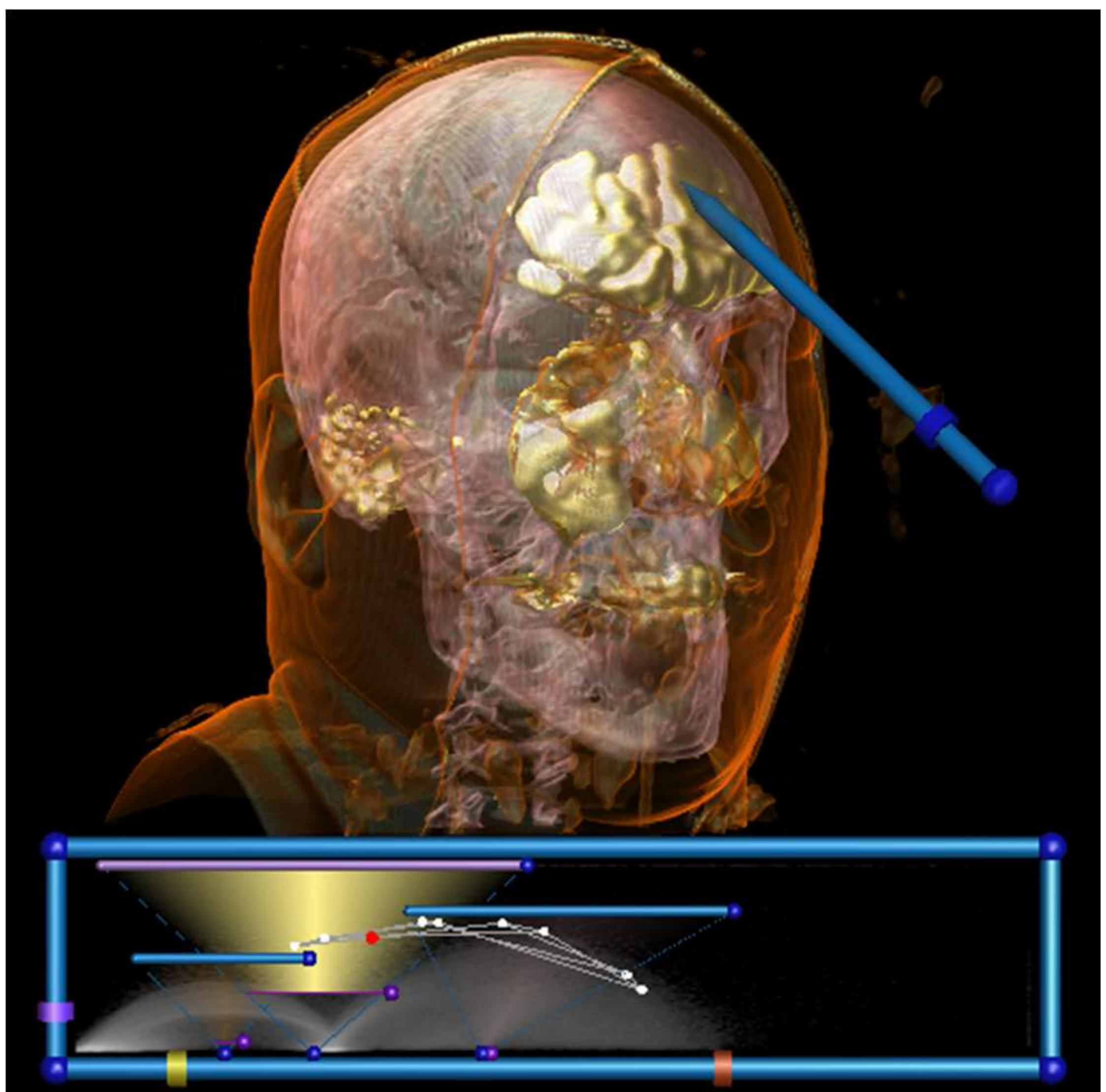
Curvature measures



Different Interaction

"Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets" Kniss, Kindlmann, Hansen: Vis '01

- Make things opaque by pointing at them
- Uses 3D transfer functions (value, 1st, 2nd derivative)
- "Paint" into the transfer function domain



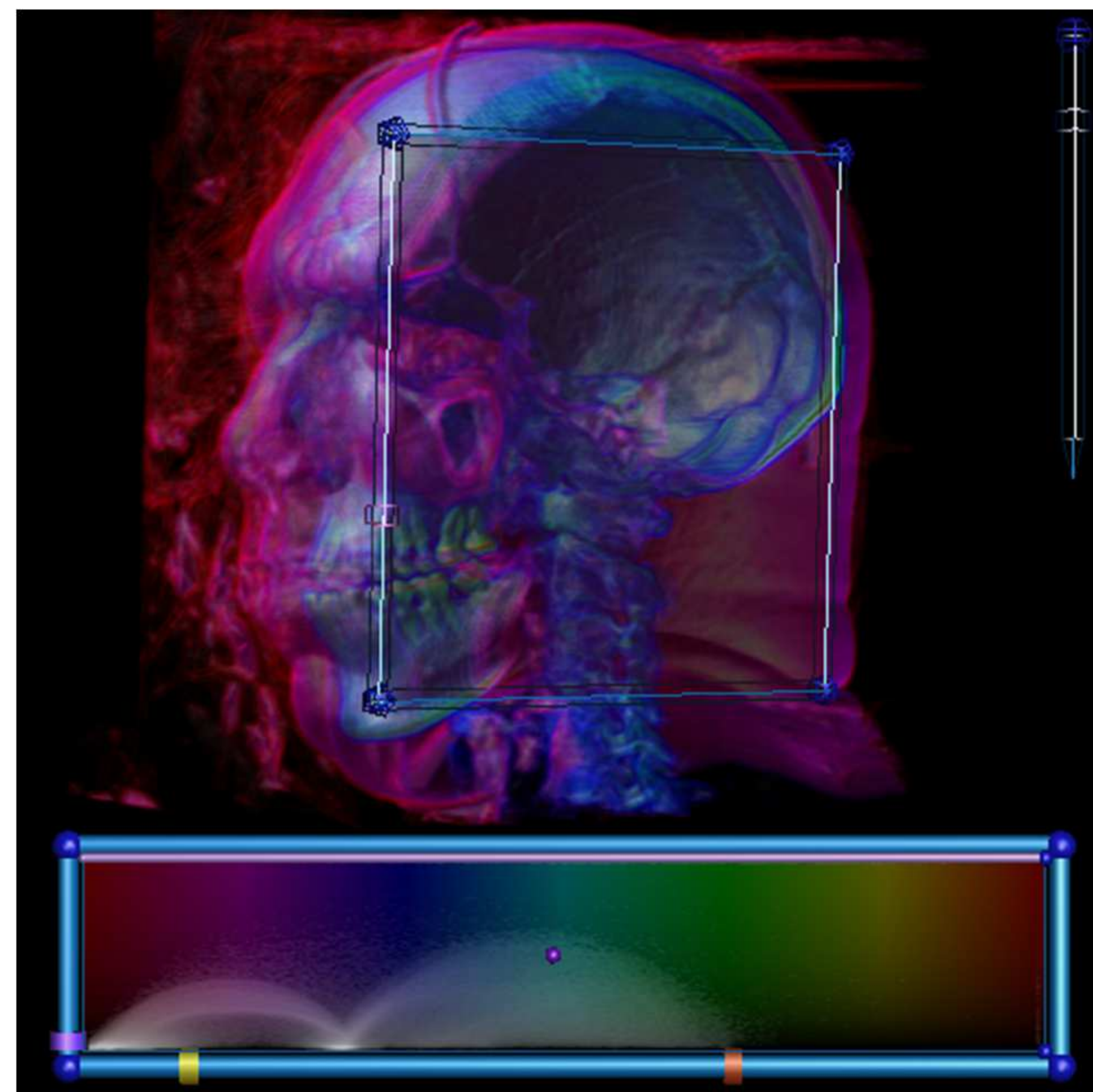


Image-centric

Specify TFs via the resulting renderings

- Genetic Algorithms ("Generation of Transfer Functions with Stochastic Search Techniques", He, Hong, *et al.*: Vis '96)
- Design Galleries (Marks, Andalman, Beardsley, *et al.*: SIGGRAPH '97; Pfister: Transfer Function Bake-off Vis '00)
- Thumbnail Graphs + Spreadsheets ("A Graph Based Interface…", Patten, Ma: Graphics Interface '98; "Image Graphs…", Ma: Vis '99; Spreadsheets for Vis: Vis '00, TVCG July '01)
- Thumbnail Parameterization ("Mastering Transfer Function Specification Using VolumePro Technology", König, Gröller: Spring Conference on Computer Graphics '01)

TF Techniques/Tools

1. Trial and Error (manual)
2. Image-Centric Approach
3. **Data-Centric Approach**

Data-centric

Specify TF by analyzing volume data itself

1. Salient Isovalues:
   - Contour Spectrum (Bajaj, Pascucci, Schikore: Vis '97)
   - Statistical Signatures ("Salient Iso-Surface Detection Through Model-Independent Statistical Signatures", Tenginaki, Lee, Machiraju: Vis '01)
   - Other computational methods ("Fast Detection of Meaningful Isosurfaces for Volume Data Visualization", Pekar, Wiemker, Hempel: Vis '01)
2. "Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering" (Kindlmann, Durkin: VolVis '98;

Data-centric

- If you have more than one value per location…

L21: Vector Visualization

required reading

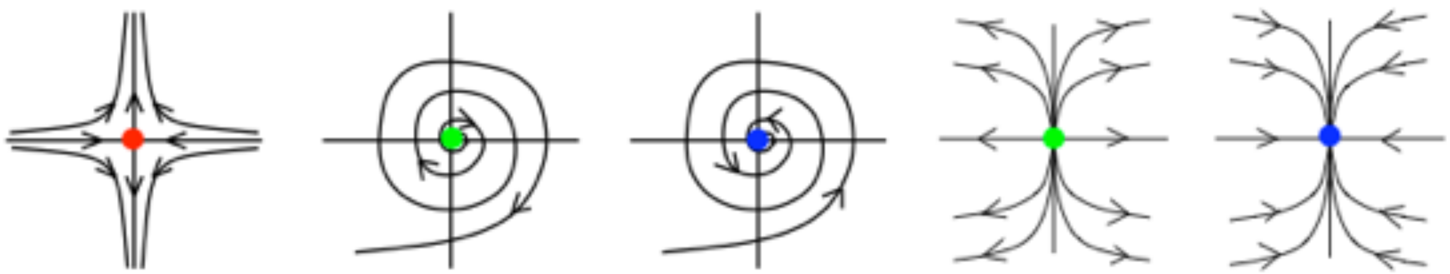


Figure 8.7. The main types of critical points in a flow field: saddle, circulating sinks, circulating sources, noncirculating sinks, and noncirculating sources. From [Tricoche et al. 02, Figure 1].

flow simulations or measurements. Flow vis in particular deals with a specific kind of vector field, a velocity field, that contains information about both direction and magnitude at each cell. The three common cases are purely 2D spatial fields, purely 3D spatial fields, and the intermediate case of flow on a 2D surface embedded within 3D space. Time-varying flow datasets are called **unsteady**, as opposed to **steady** flows where the behavior does not change over time.

One of the features of interest in flows are the **critical points**, the points in a flow field where the velocity vanishes. They are classified by the behavior of the flow in their neighborhoods: the


5. Flow Visualization
11:17 / 35:00