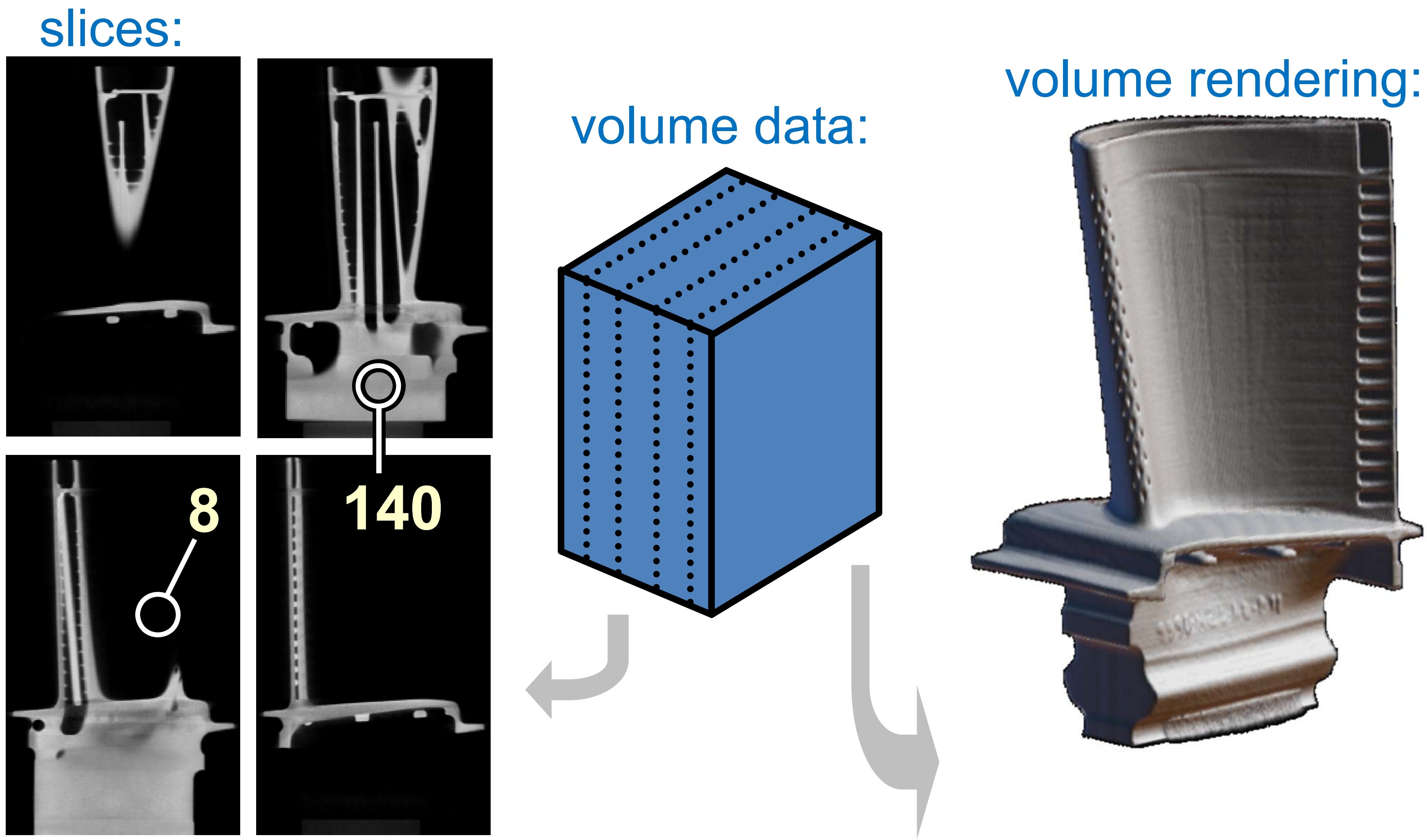
# Transfer Functions for Direct Volume Rendering

SCI INSTITUTE

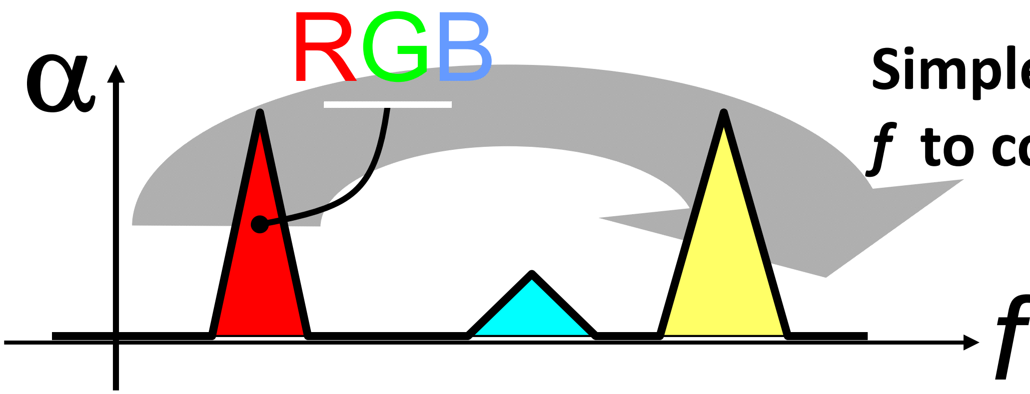
# Introduction

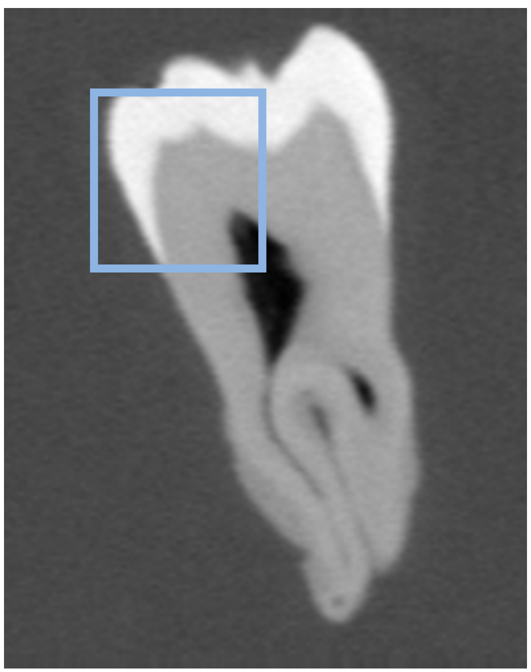
Transfer functions **make volume data visible**
*by mapping data values to optical properties*



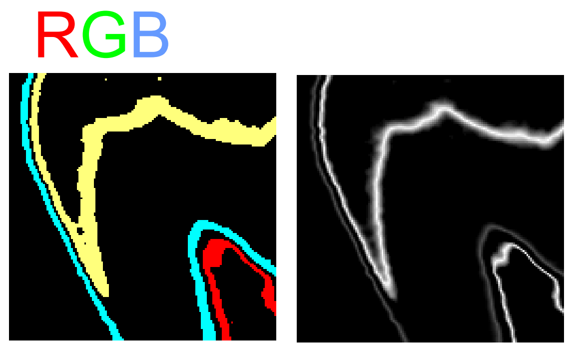
# Transfer Functions (TFs)

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

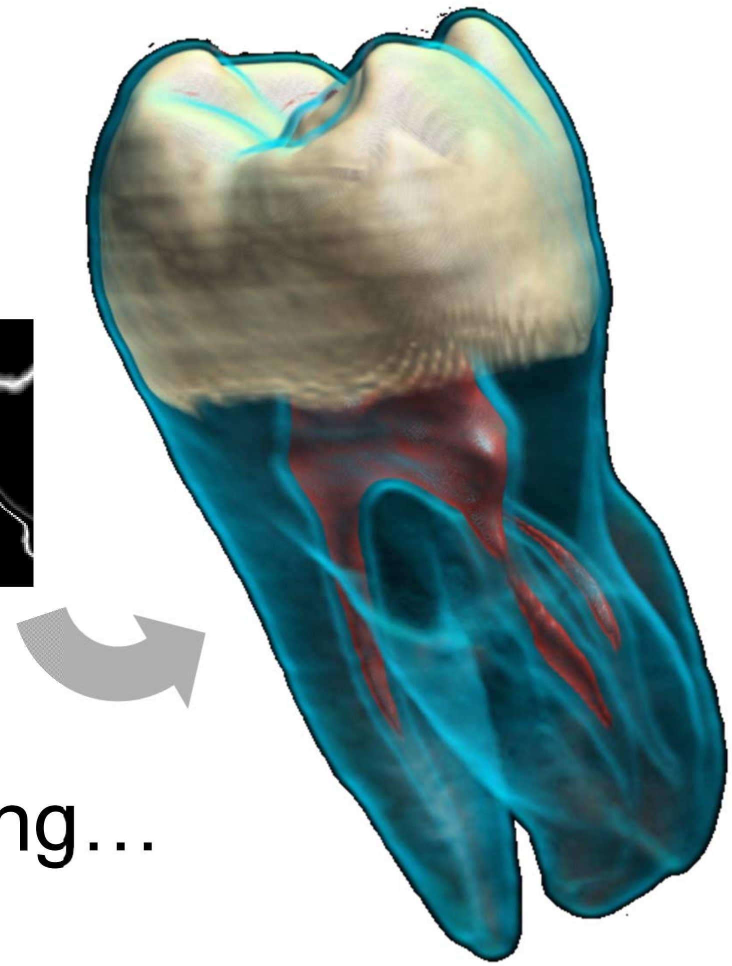


Human Tooth CT

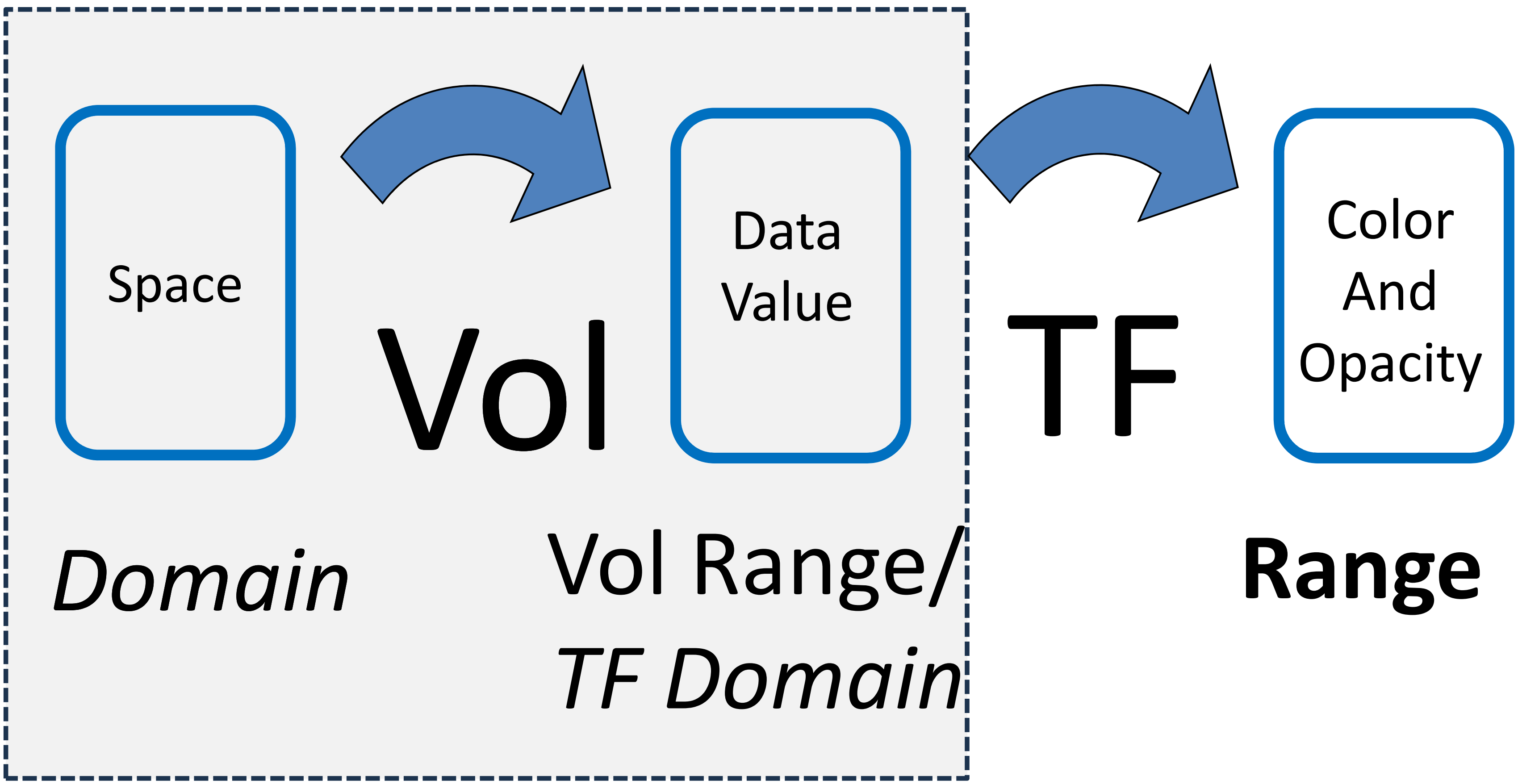



Shading, Compositing…



# Terminology

- Basic Transfer Functions:

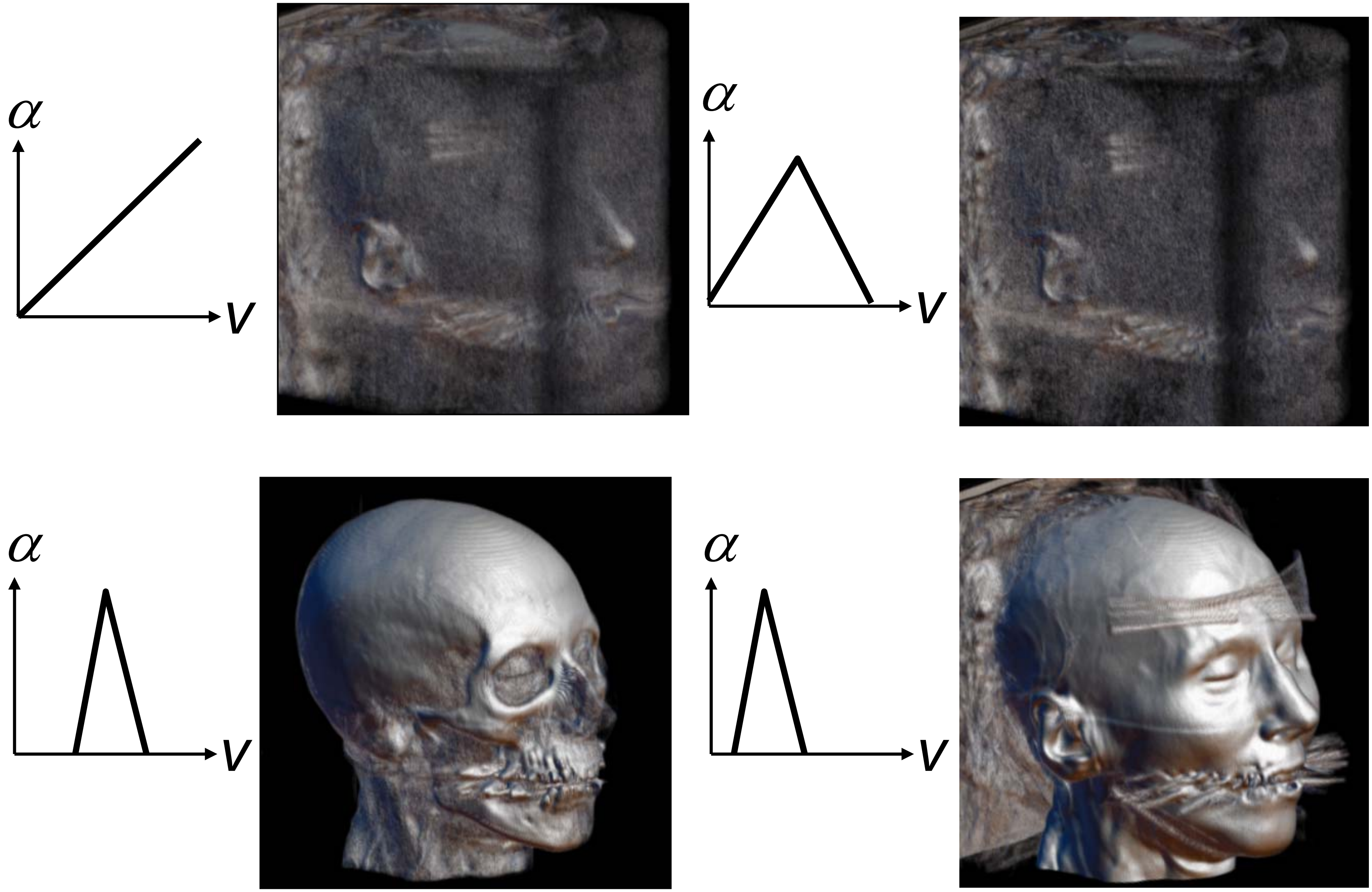


# What else in range?

- "**Optical Properties**": *Anything that can be* ***composited*** with a standard graphics operator ("*over*" / "*superimpose*")
  - **Opacity**: "opacity functions"
    - Most important in 3D!
  - **Color**
    - Can help distinguish features
  - Emittance
  - Phong parameters ($k_a$, $k_d$, $k_s$)
  - Index of refraction

How?

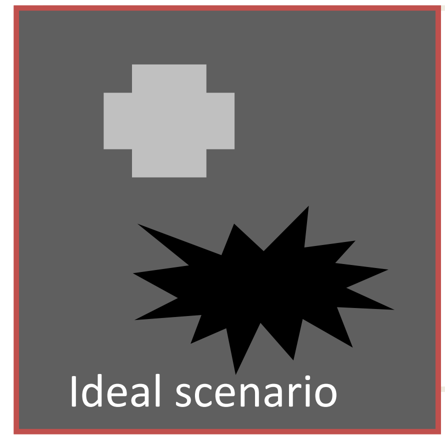
# Setting Transfer Function: Hard

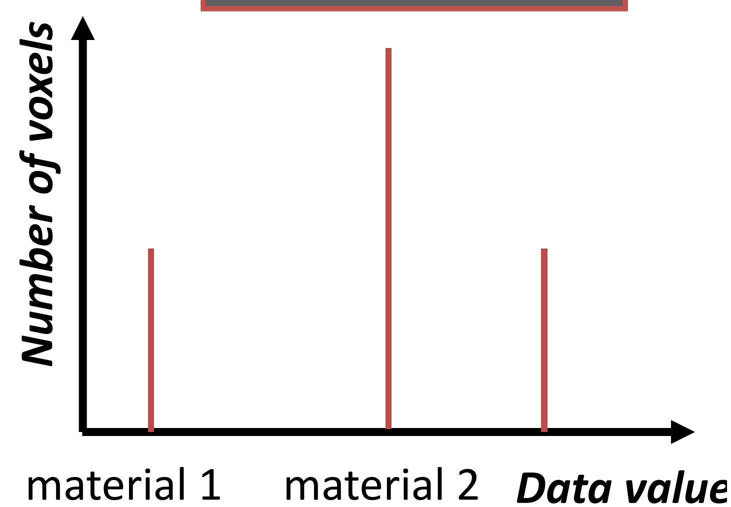


# Volumes Consisting of Materials

So the task is to ***reveal the boundaries of different materials***

***Recall the tooth and the head examples!***



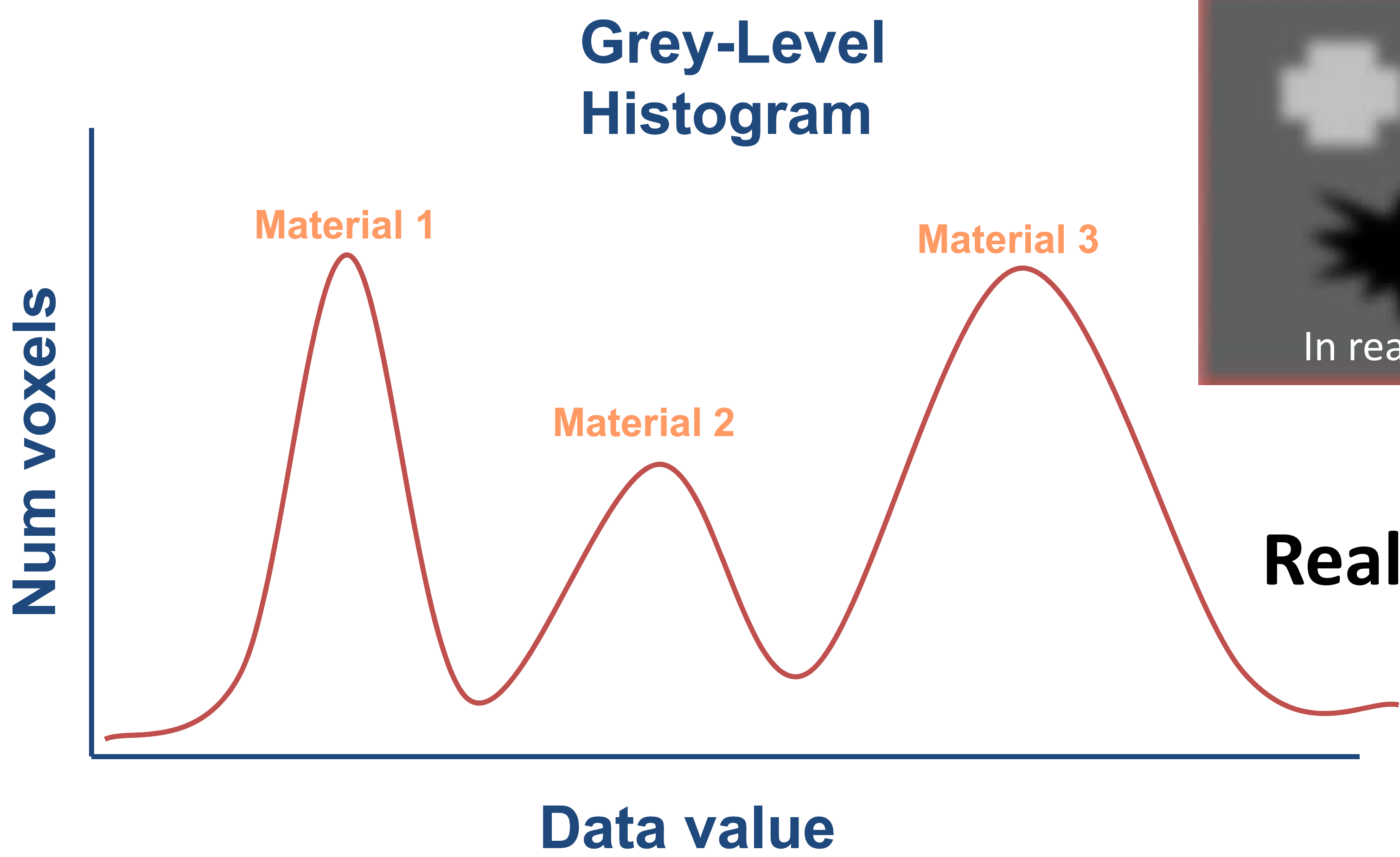



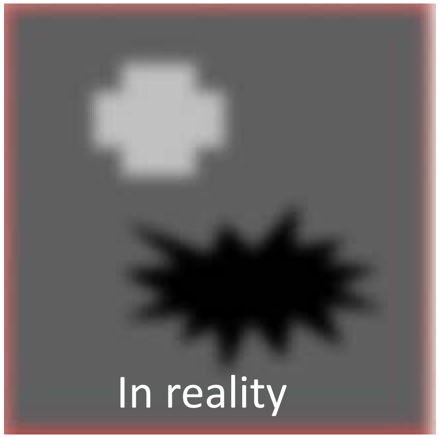

**Ideal situation**

How?

# Volumes Consisting of Materials

***So the task is to reveal the boundaries of different materials***



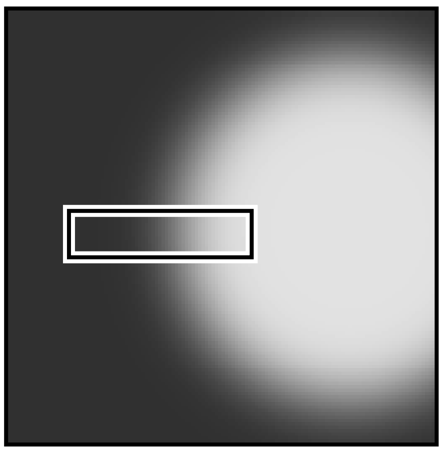



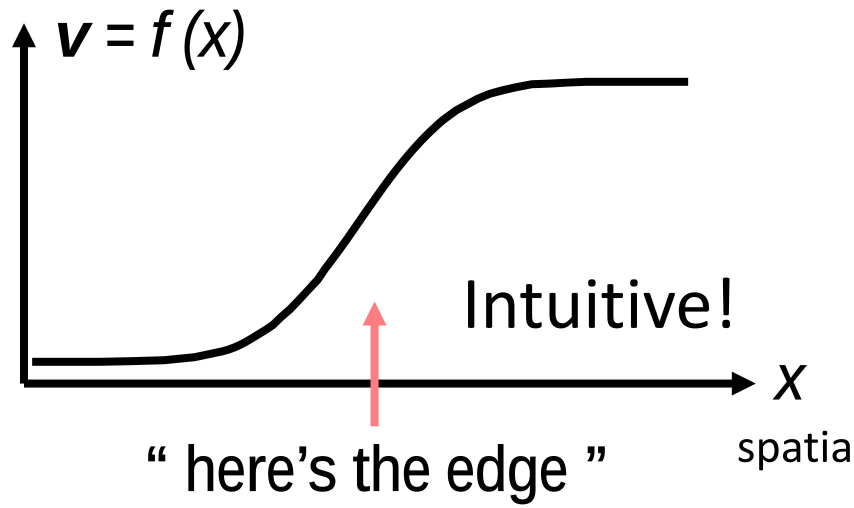

**Reality!!**

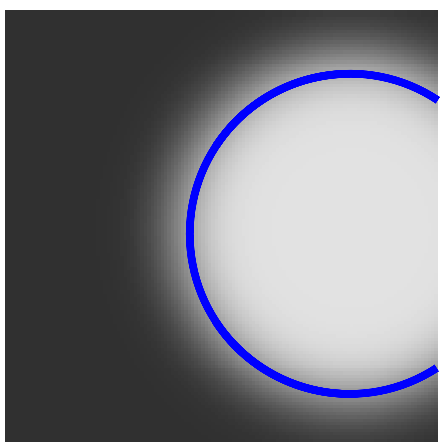
***In reality, the values at material boundaries are blurred due to the precision limit***

# Finding edges: easy in image

**“<u>Where</u>’s the edge?”**



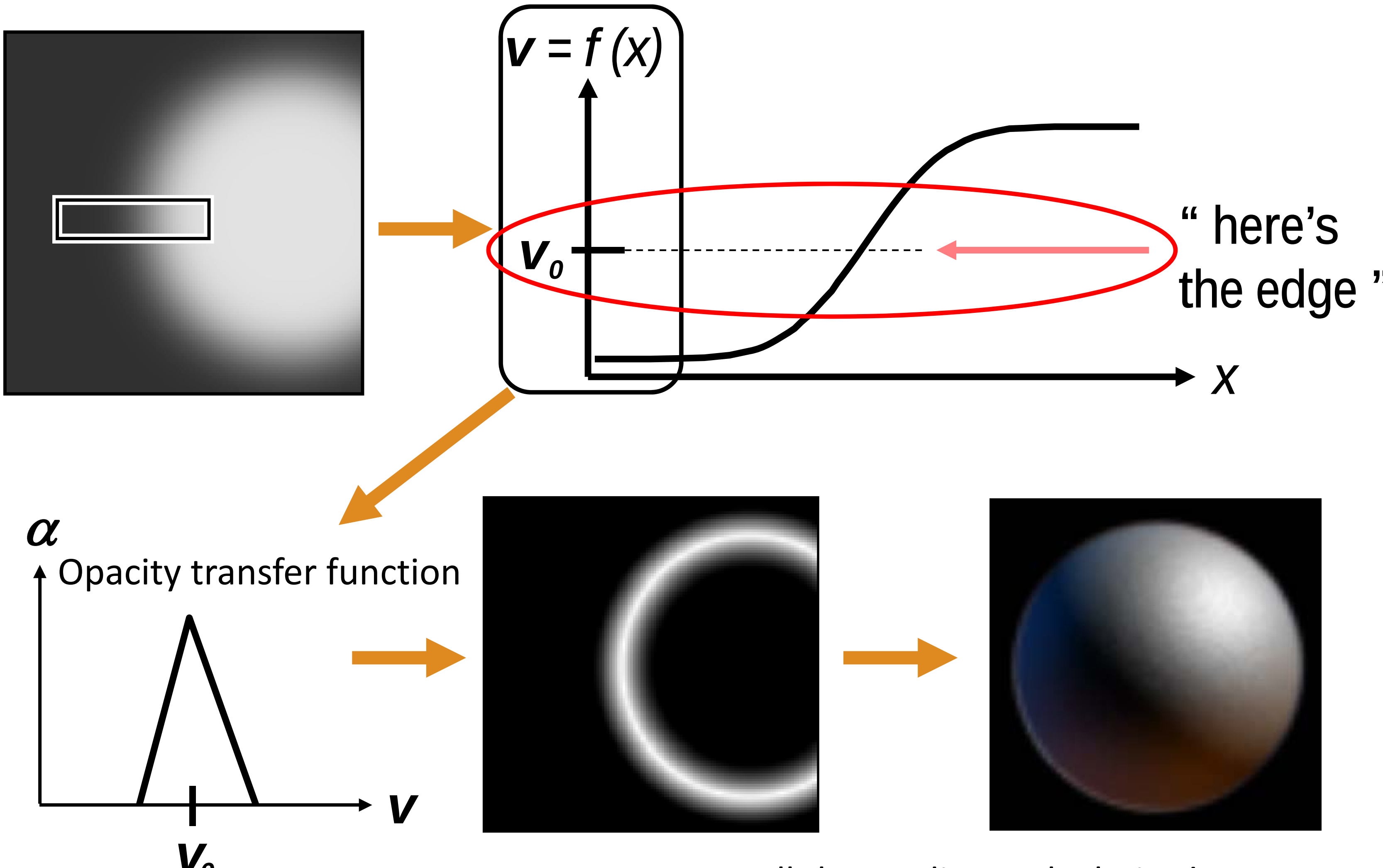


Result: edge pixels

Recall the gradient calculation!

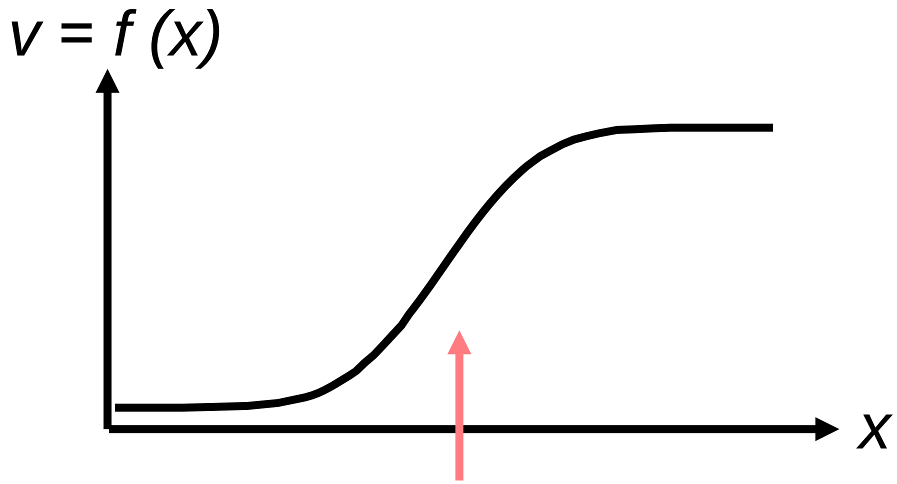
# Transfer function **Unintuitive**



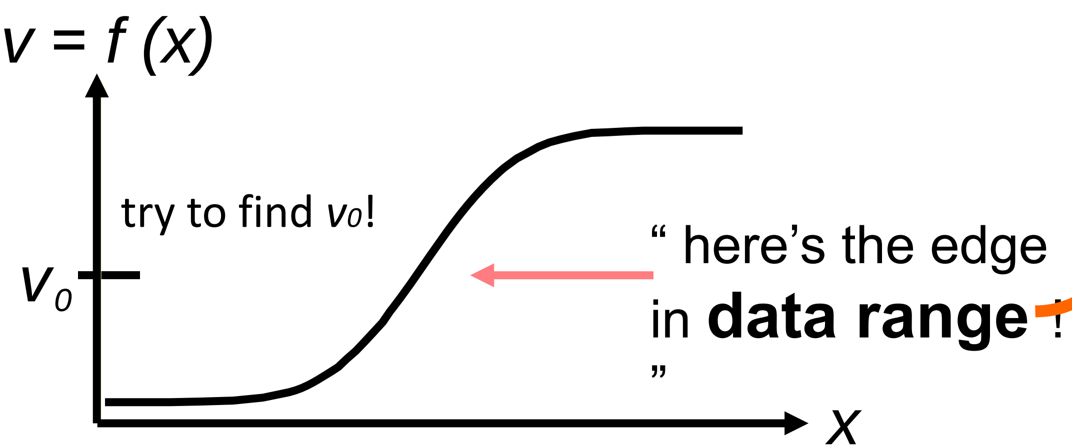

***No spatial content!!!!***

Recall the gradient calculation!

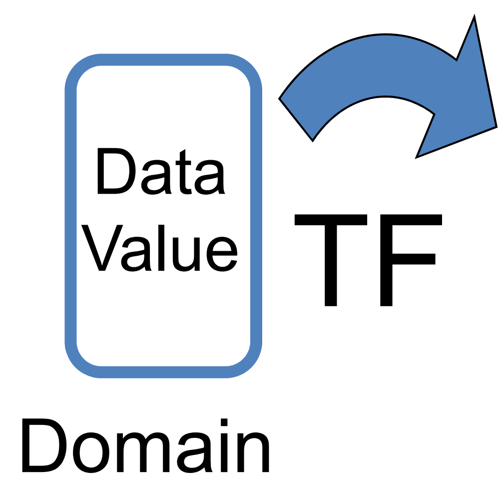
# TFs as feature detection



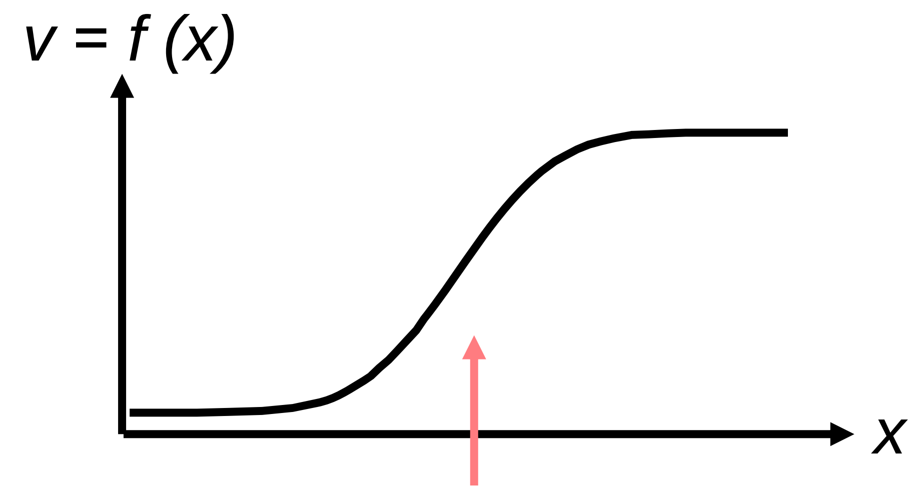

" here's the edge (**in space**)! "



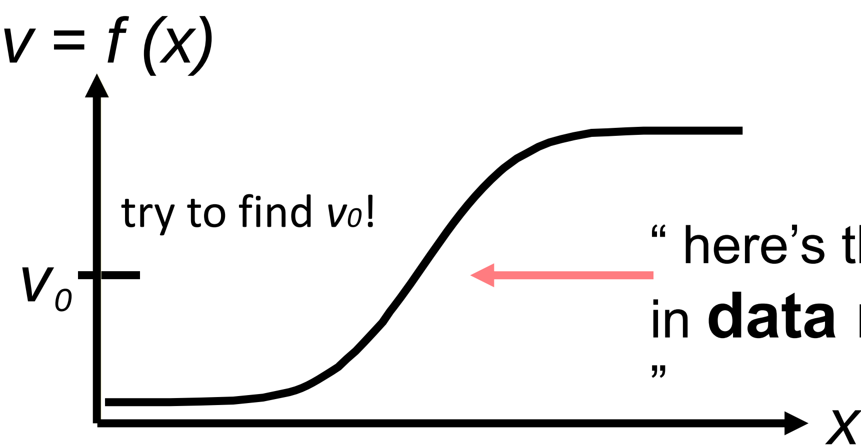

**Domain of the transfer function does not include position**



# TFs as feature detection

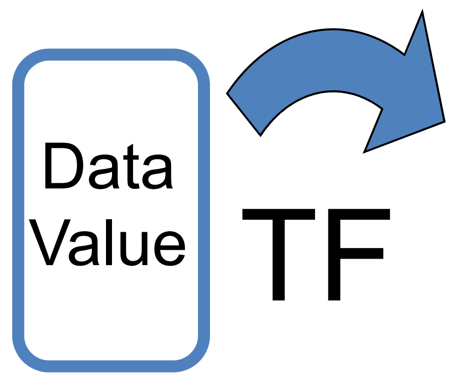



" here's the edge (in space)! "




" here's the edge in **data range** ! "

**Domain of the transfer function does not include position**




Domain

**Note that this $v_0$ may correspond to places other than edges!!!!**

**Now we have seen the challenge of transfer function design, let us see some strategies of addressing this challenge.**

**But before that, let us see some general guideline for transfer function design.**

# Good TFs Should

- 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

# Good TFs Should

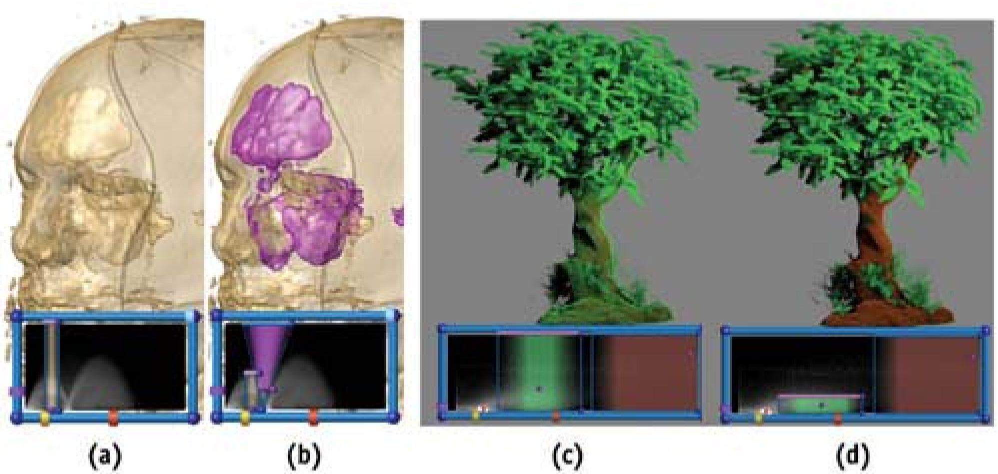
- Make good renderings easier to come by
- Make space of TFs less confusing
- Remove excess "flexibility"
- Provide one or more of:
  - Information
  - Guidance (for Semi-automation)
  - Semi-automation (more popular)
  - Automation (Machine learning!)

# TF Methods

1. Trial and Error (manual)
2. Spatial Feature Detection
3. Image-Centric
4. Data-Centric

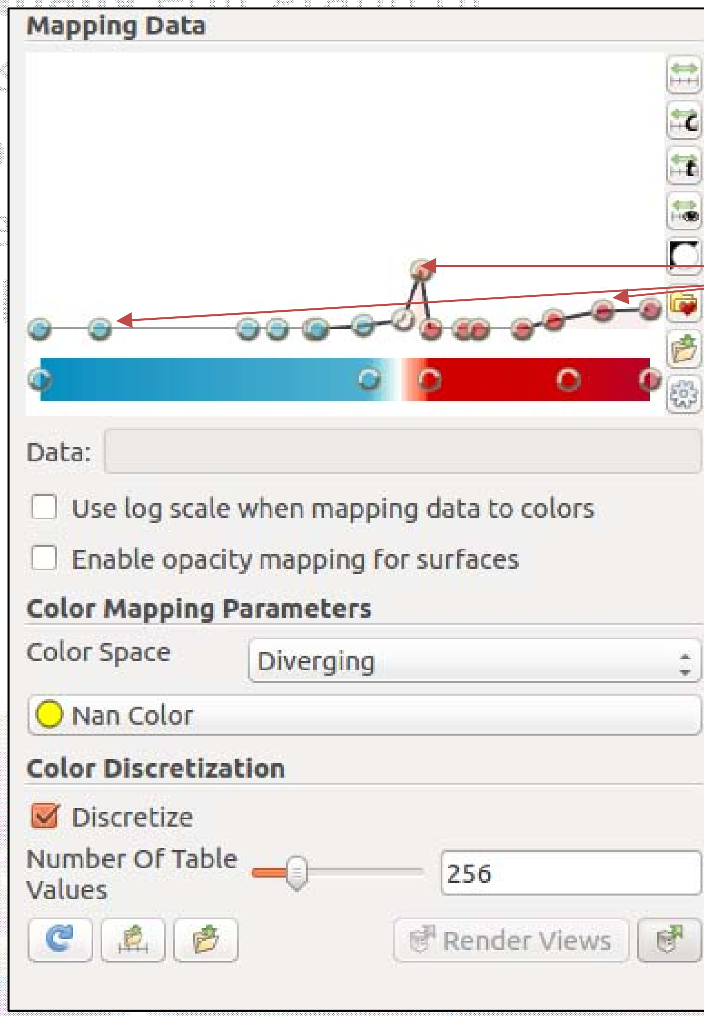
# 1. Trial and Error

1. **Manually** edit graph of transfer function
2. Enforces learning by experience
3. Get better with practice
4. Can make terrific images




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

# 1. Trial and Error




**Control points selected by the user!**

In VTK

An example color transfer function (for the head dataset)

```
volumeColor = vtk.vtkColorTransferFunction()
volumeColor.AddRGBPoint(0, 0.0, 0.0, 0.0)
volumeColor.AddRGBPoint(500, 1.0, 0.5, 0.3)
volumeColor.AddRGBPoint(1000, 1.0, 0.5, 0.3)
volumeColor.AddRGBPoint(1150, 1.0, 1.0, 0.9)
```

An example opacity transfer function (for the head dataset)

```
volumeScalarOpacity = vtk.vtkPiecewiseFunction()
volumeScalarOpacity.AddPoint(0, 0.00)
volumeScalarOpacity.AddPoint(500, 0.15) #Skin is more transparent
volumeScalarOpacity.AddPoint(1000, 0.15)
volumeScalarOpacity.AddPoint(1150, 0.85) #Bone get the highest opacity
```

**Note that you can have either more or fewer control points depending on the need!**

# TF Methods

1. Trial and Error (manual)
2. Spatial Feature Detection
3. Image-Centric
4. Data-Centric

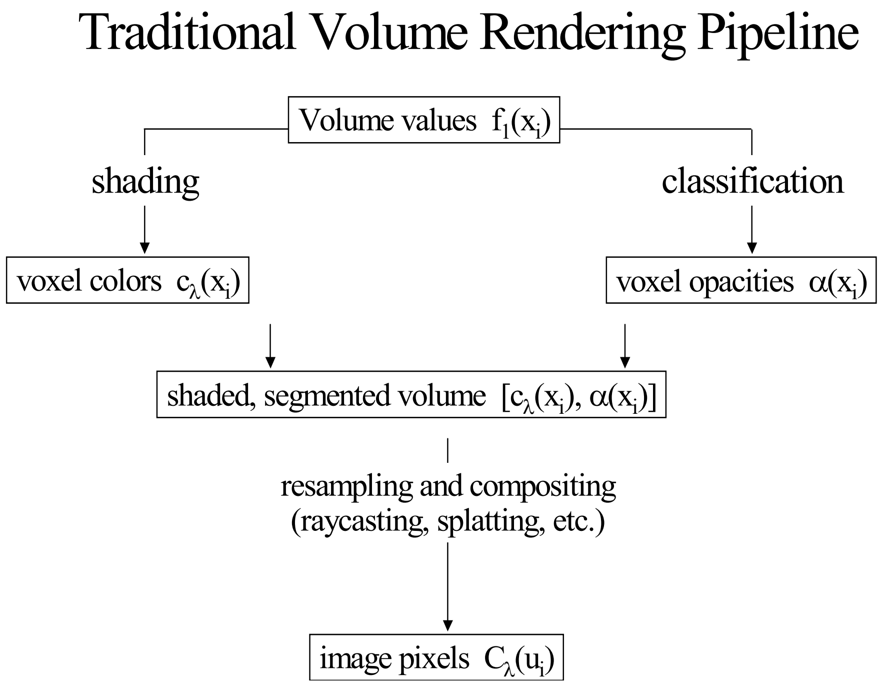
# 2. Spatial Feature Detection

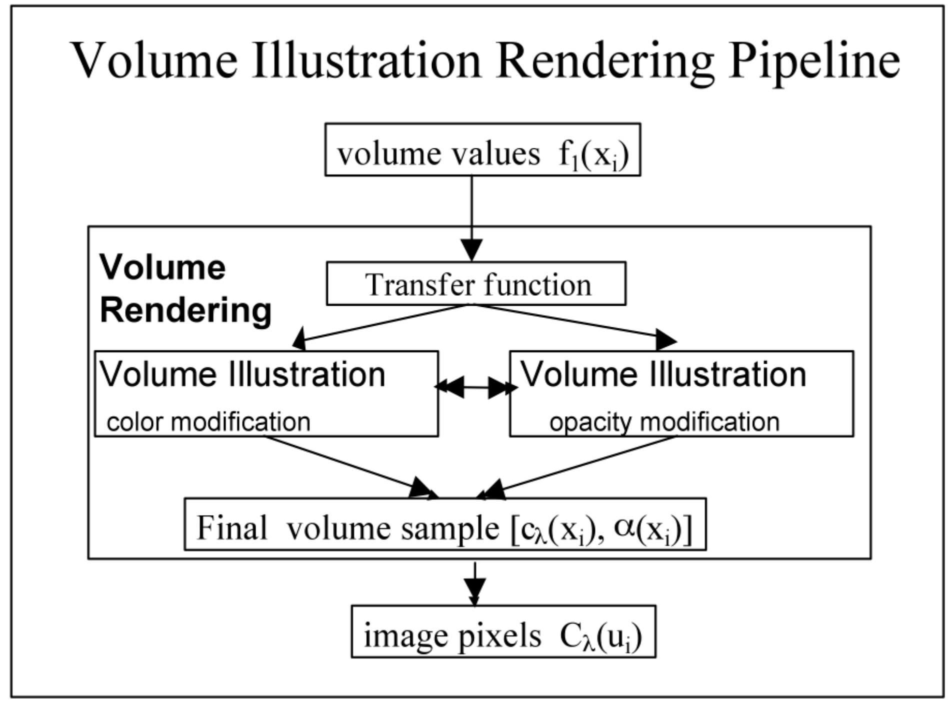
Transform TF specification to **feature detection in the spatial domain**

- extremely flexible
- different parameter space
- ***not exactly transfer functions*** …

1. Fang, Biddlecome, Tuceryan (Vis ‘98)
   “Image-based Transfer Function Design…”
2. Rheingans, Ebert (Vis ’00, TVCG July ’01)
   “Volume Illustration: Non-photorealistic…”
3. Hladuvka, Gröller (VisSym ’01) “Salient Representation of Volume Data”

# Volume Illustration




Thanks to Penny Rheingans and David Ebert

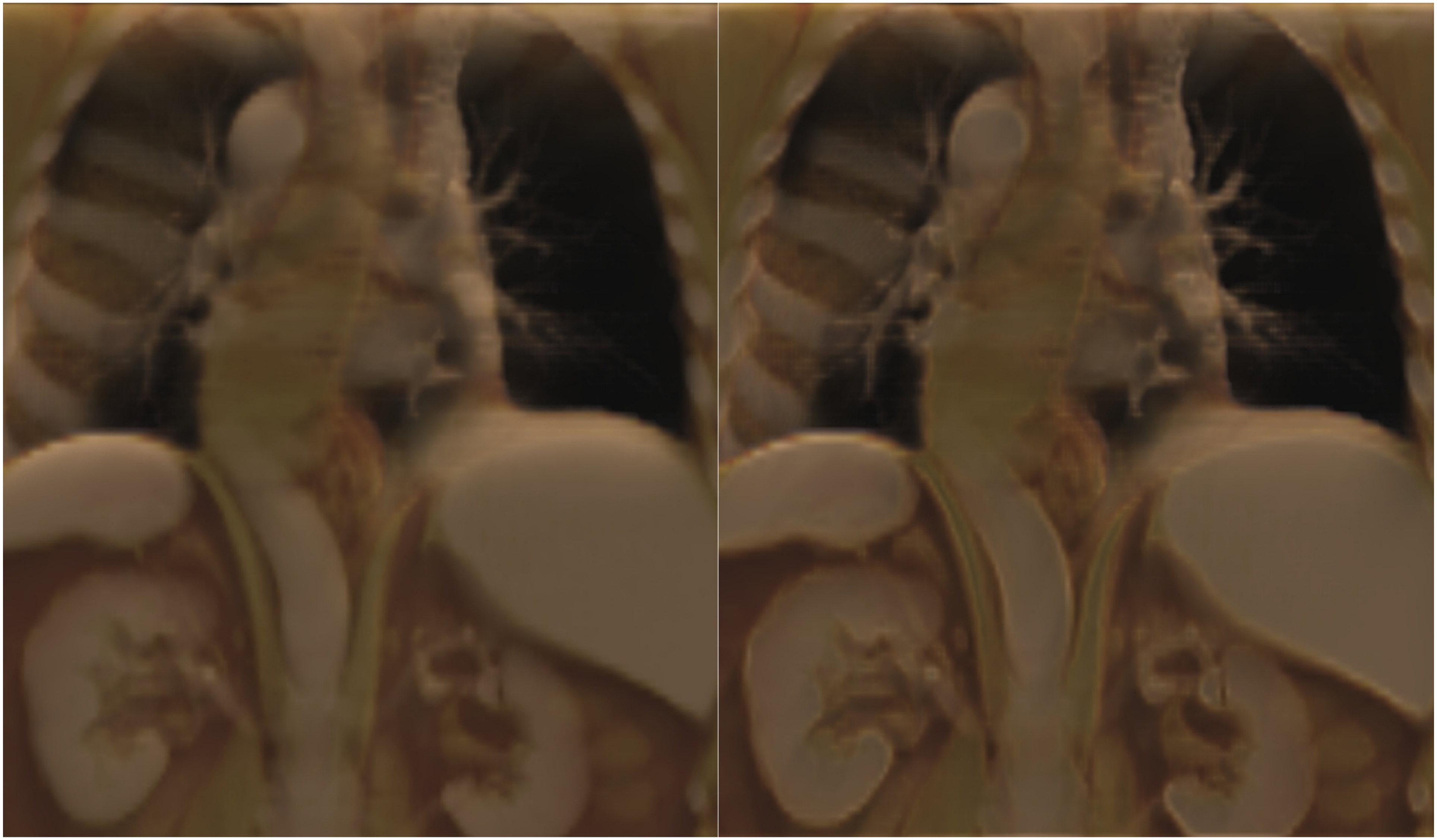
## Feature Enhancement

- Boundary, silhouette enhancement

## Depth and Orientation Cues

- Halos, depth cueing

# Volume Illustration



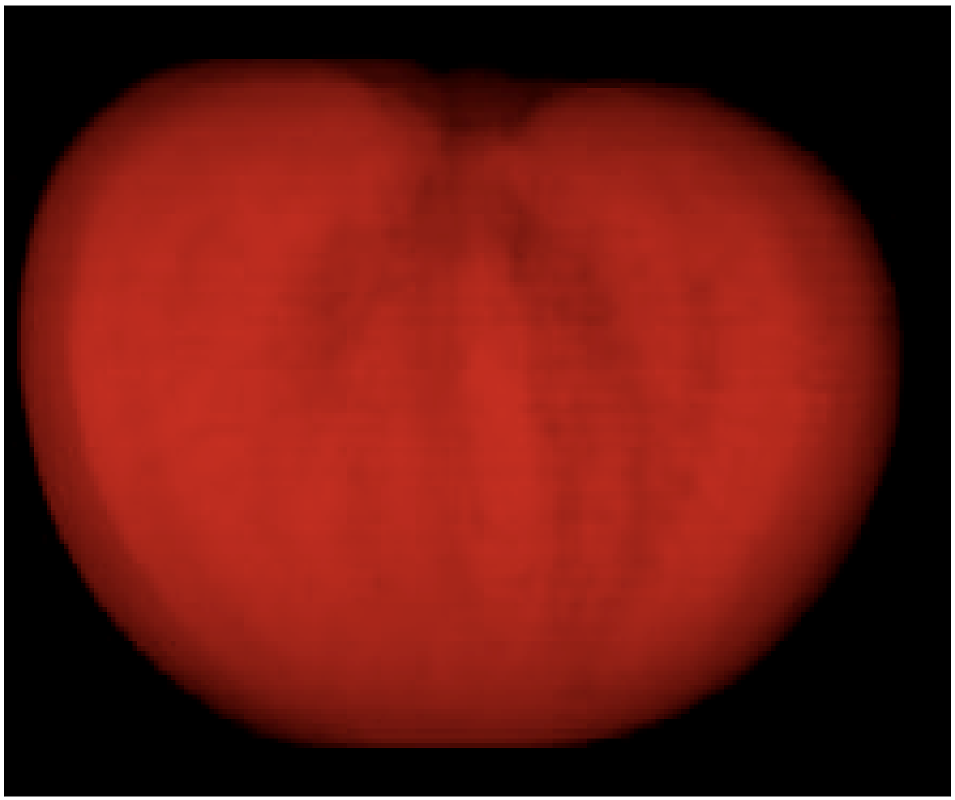
Original TF    Boundaries (gradient)

# Volume Illustration

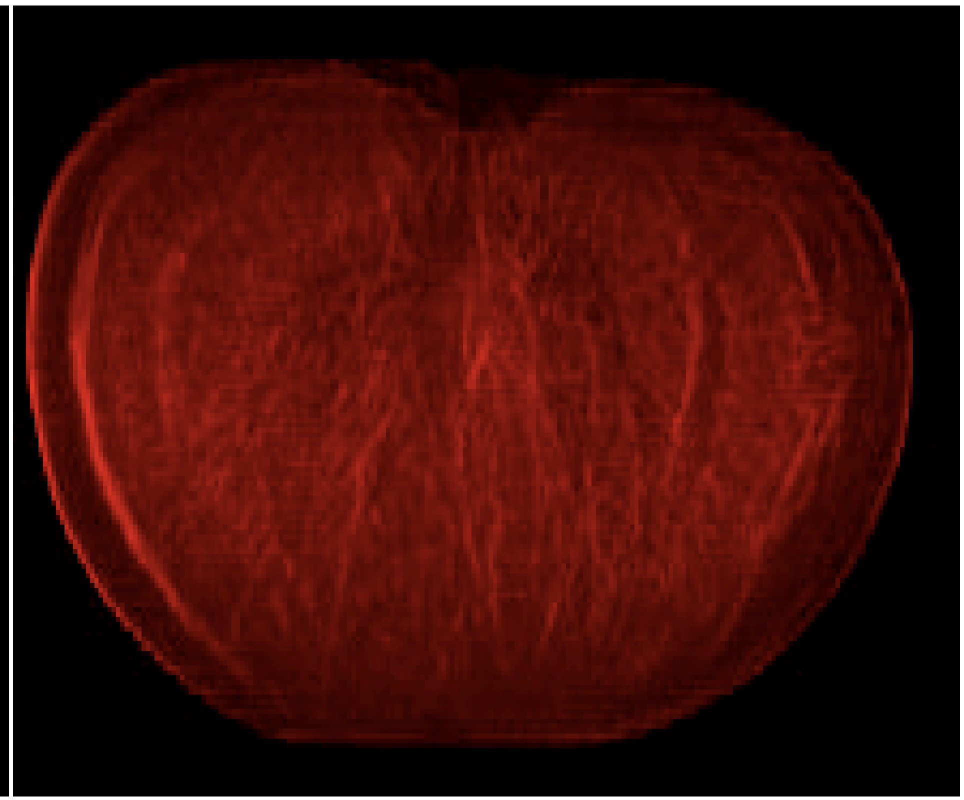
Silhouettes

Halos

# Volume Illustration



Traditional



Boundary and silhouette

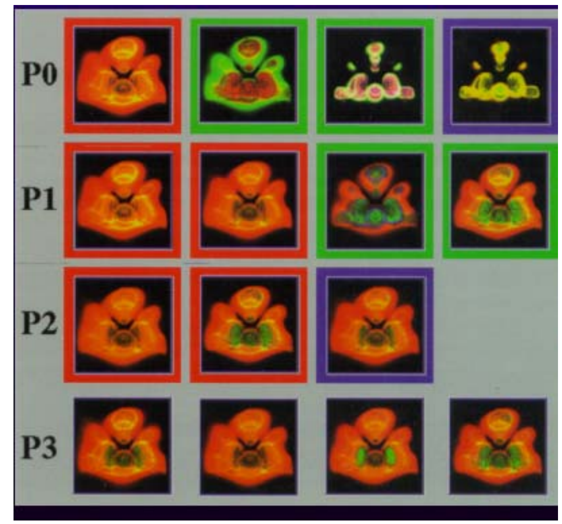
# TF Methods

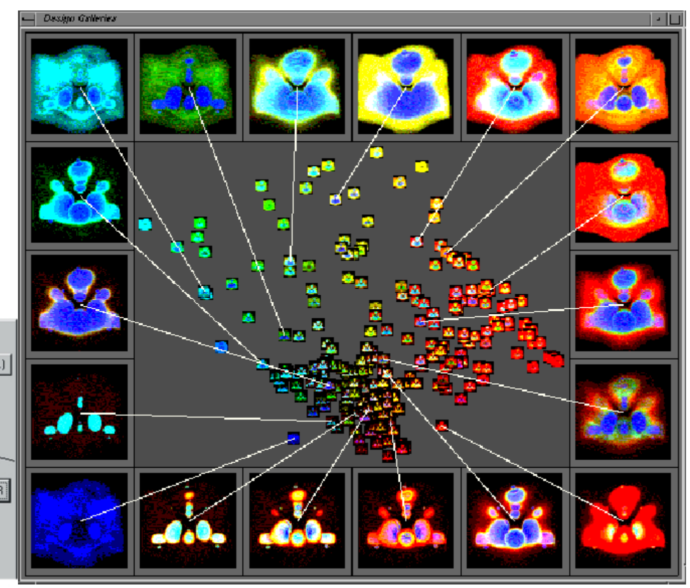
1. Trial and Error (manual)
2. Spatial Feature Detection
3. Image-Centric
4. Data-Centric

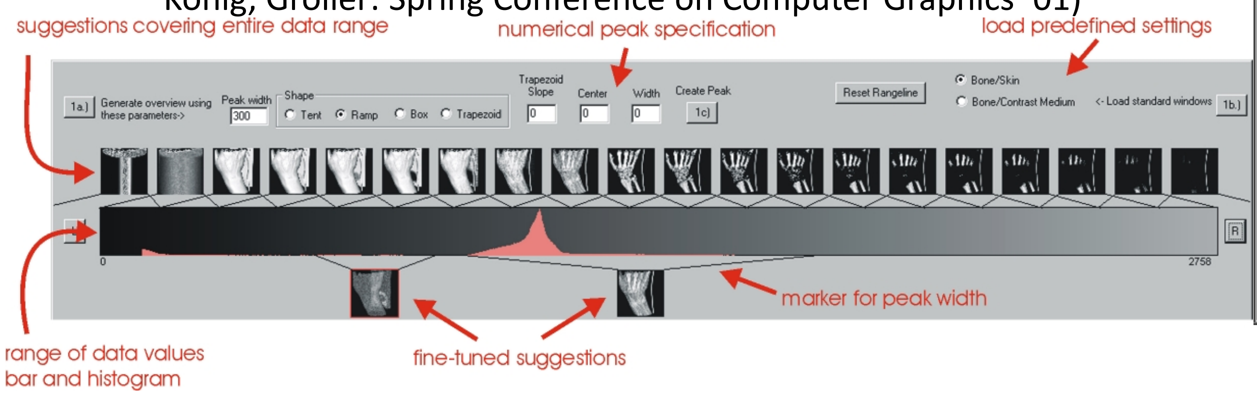
# 3. Image-centric

## 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 Methods

1. Trial and Error (manual)
2. Spatial Feature Detection
3. Image-Centric
4. Data-Centric
5. Others

# 4. Data-centric

Specify TF by **analyzing** volume data itself

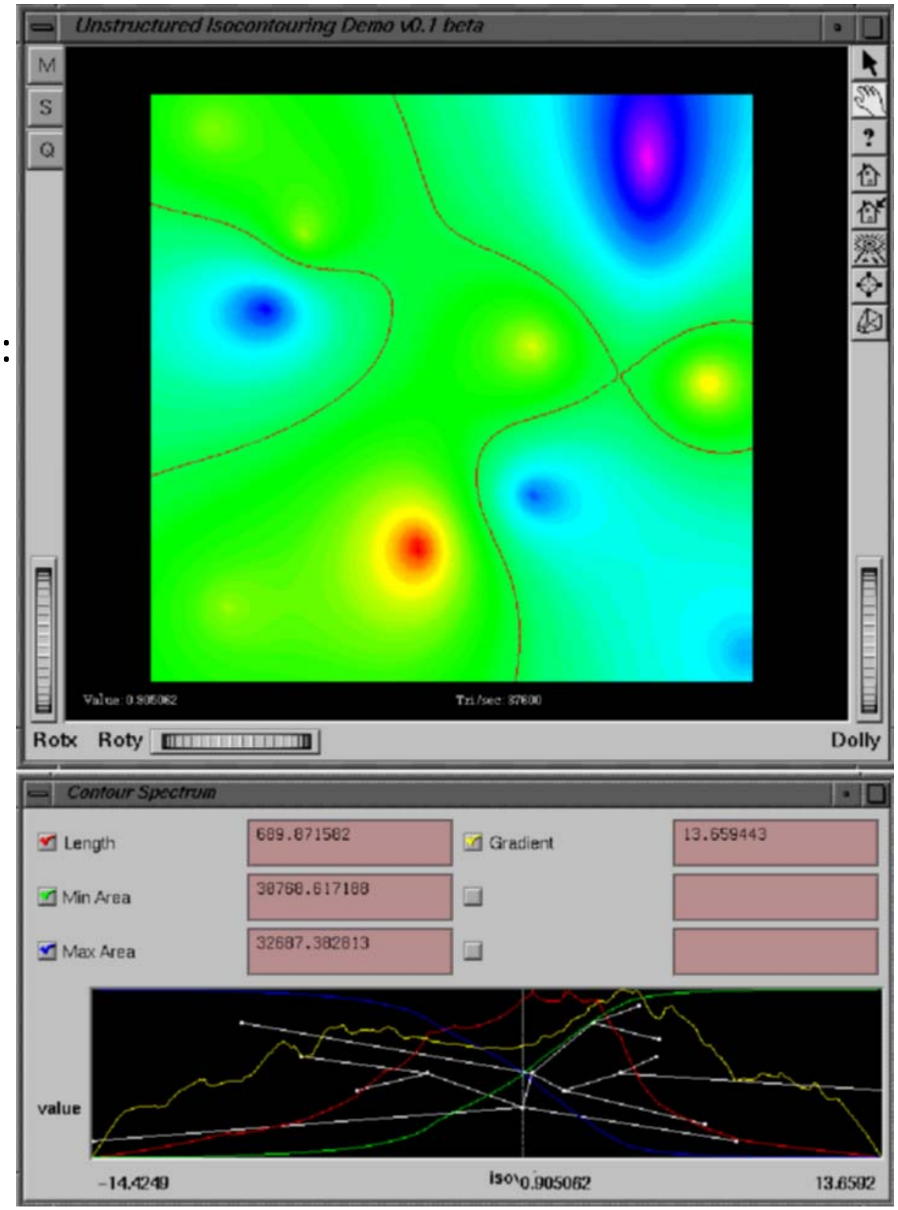
1. Salient Isovalues:
   - Contour Spectrum (Bajaj, Pascucci, Schikore:  Vis ’97)
   - Statistical Signatures (“Salient Iso-Surface Detection Through Model-Independent Statistical Signatures”, Tenginaki, Lee, Machiraju: Vis ’01)
   - Other computational methods (“Fast Detection of Meaningful Isosurfaces for Volume Data Visualization”, Pekar, Wiemker, Hempel: Vis ’01)
2. “Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering” (Kindlmann, Durkin: VolVis ’98; Kindlmann MS Thesis ’99; Transfer Function Bake-Off Panel: Vis ‘00)

# Salient Isovalues

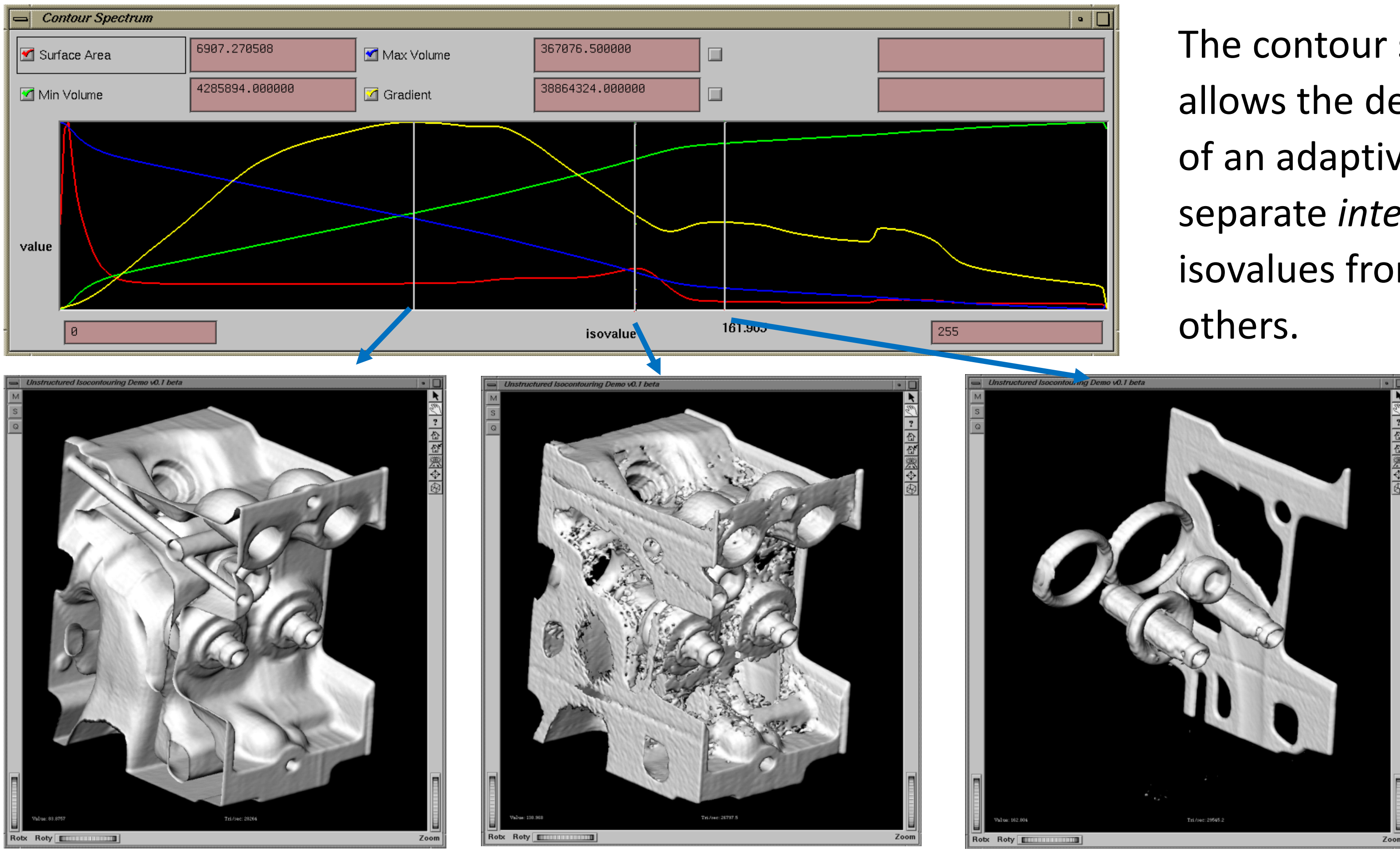
What are the “best” isovalues for extracting the **main structures** in a volume dataset?

Contour Spectrum (Bajaj, Pascucci, Schikore: Vis ’97; Transfer Function Bake-Off: Vis ’00)

- Efficient computation of isosurface metrics
  - Area, enclosed volume, gradient surface integral, etc.
- Efficient connected-component topological analysis
- Interface itself concisely summarizes data

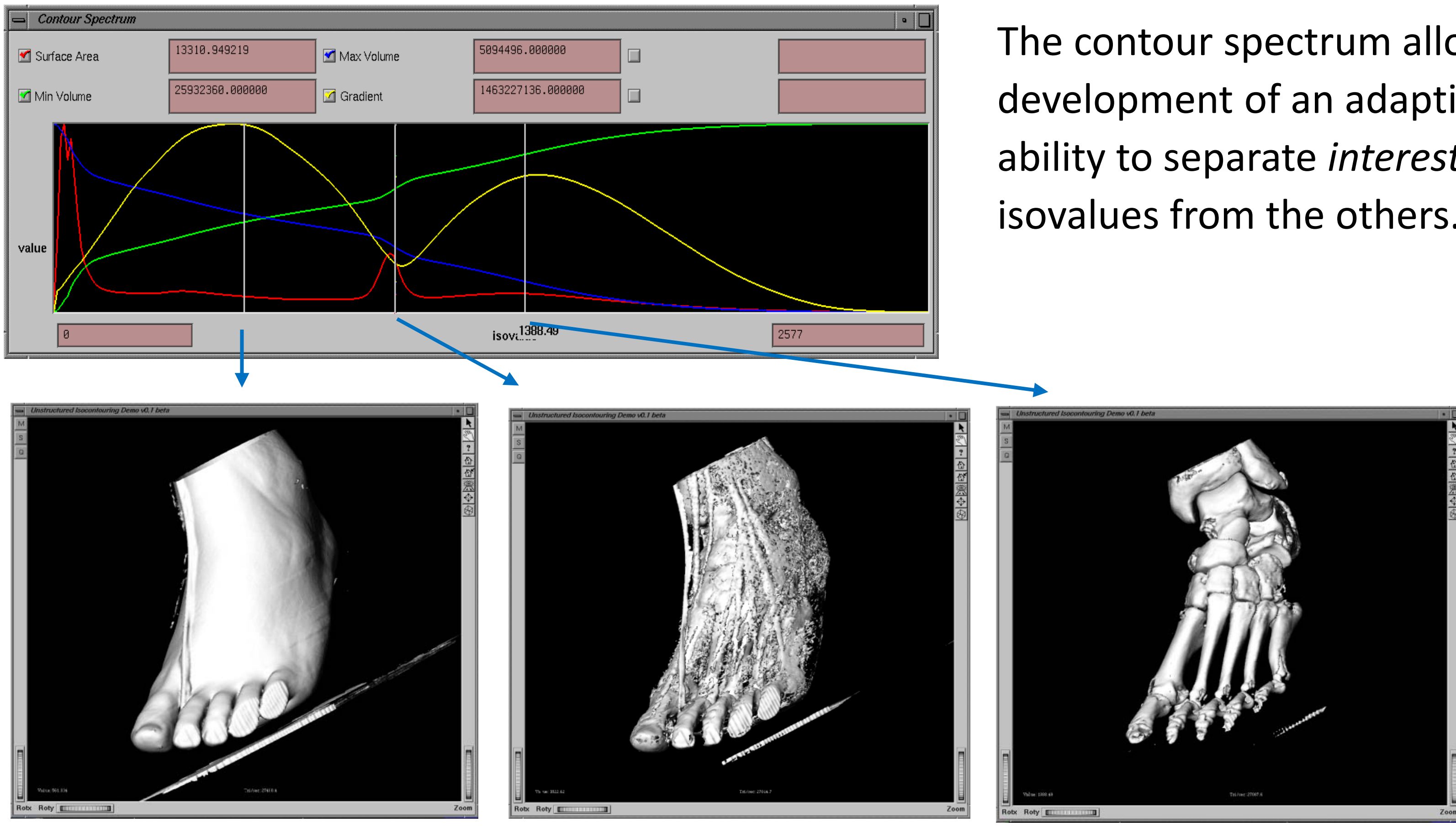


# Contour Spectrum



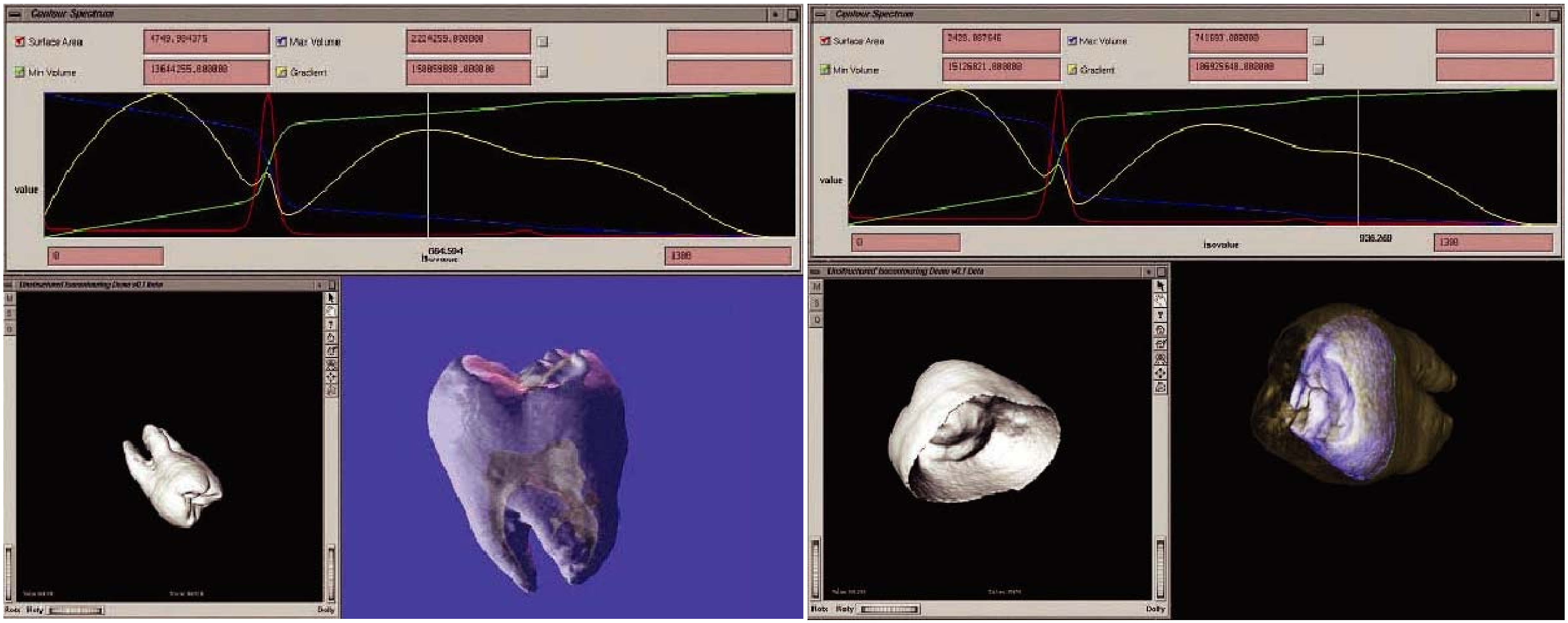

The contour spectrum allows the development of an adaptive ability to separate *interesting* isovalues from the others.

# Contour Spectrum




The contour spectrum allows the development of an adaptive ability to separate *interesting* isovalues from the others.

# Contour Spectrum




**Any issue with the contour spectrum???**

# "Semi-Automatic …"

Reasoning:

- ***TFs are volume-position invariant***
- Histograms "**project out (lose)**" position
- *Interested in boundaries between materials*

# "Semi-Automatic …"

Reasoning:

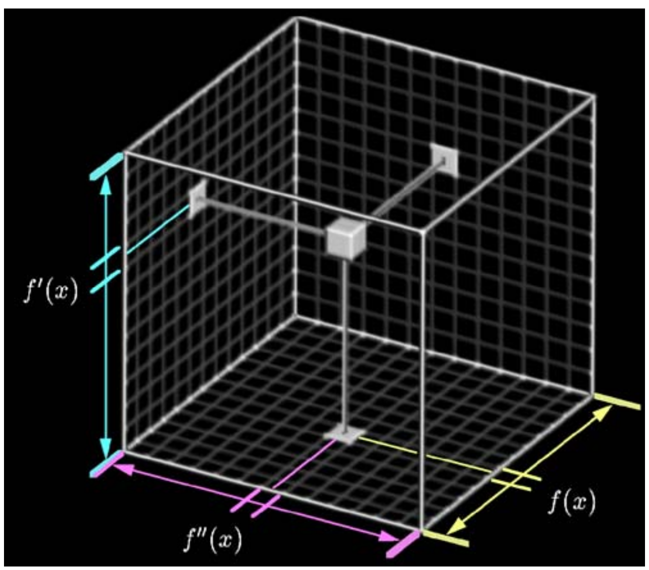
- ***TFs are volume-position invariant***
- Histograms "**project out (lose)**" position
- *Interested in boundaries between materials*
- Boundaries characterized by derivatives

# "Semi-Automatic …"

Reasoning:

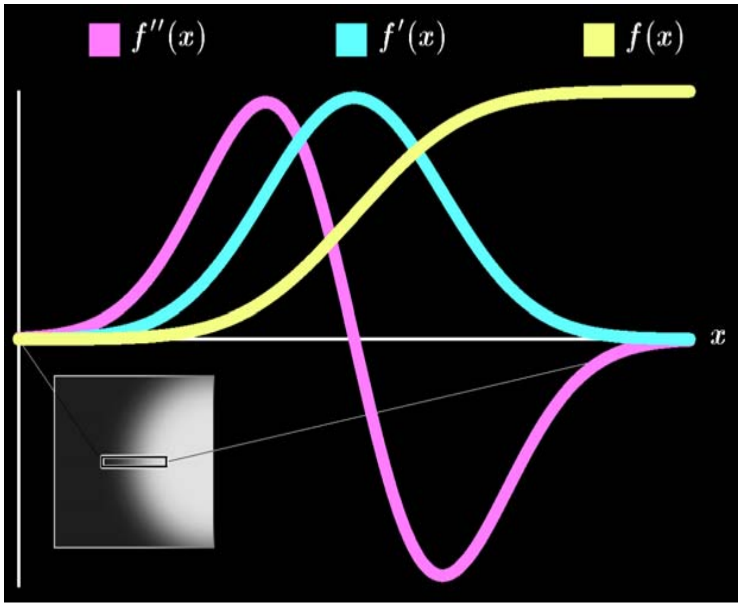
- ***TFs are volume-position invariant***
- Histograms "**project out (lose)**" position
- *Interested in boundaries between materials*
- Boundaries characterized by derivatives

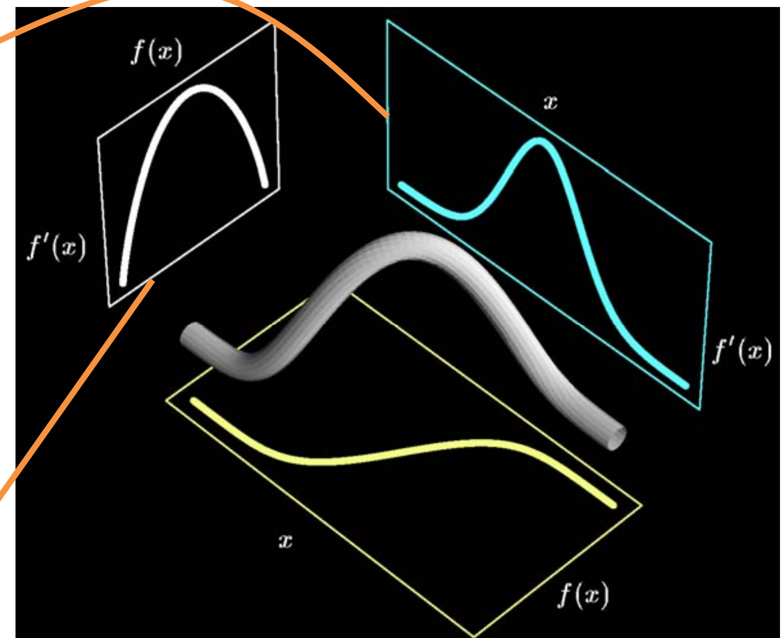
➔ **Make 3D histograms of value, 1st, 2nd deriv**.

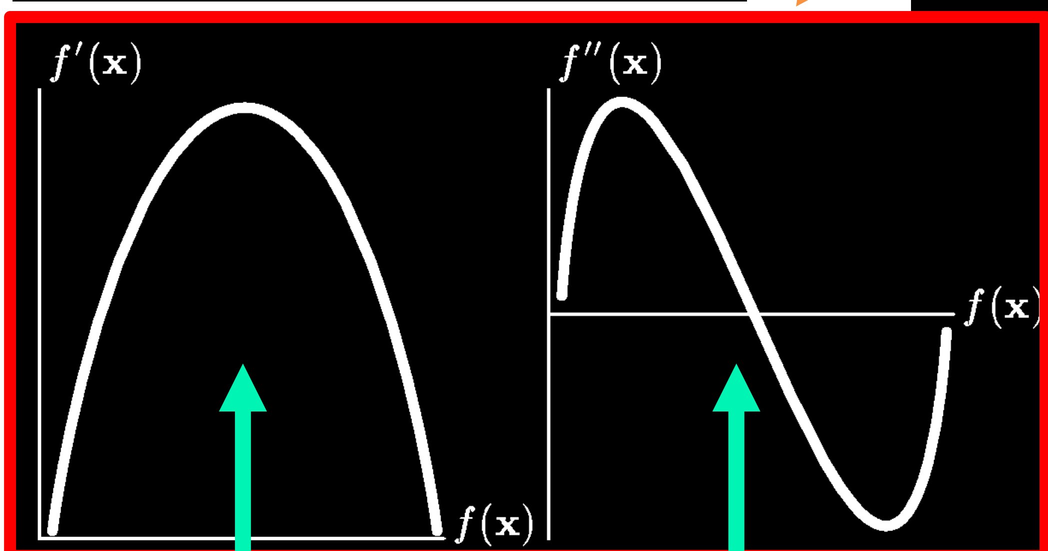



By (1) inspecting and (2) algorithmically analyzing histogram volume of the function and its derivatives, we can create transfer functions

# Derivative relationships

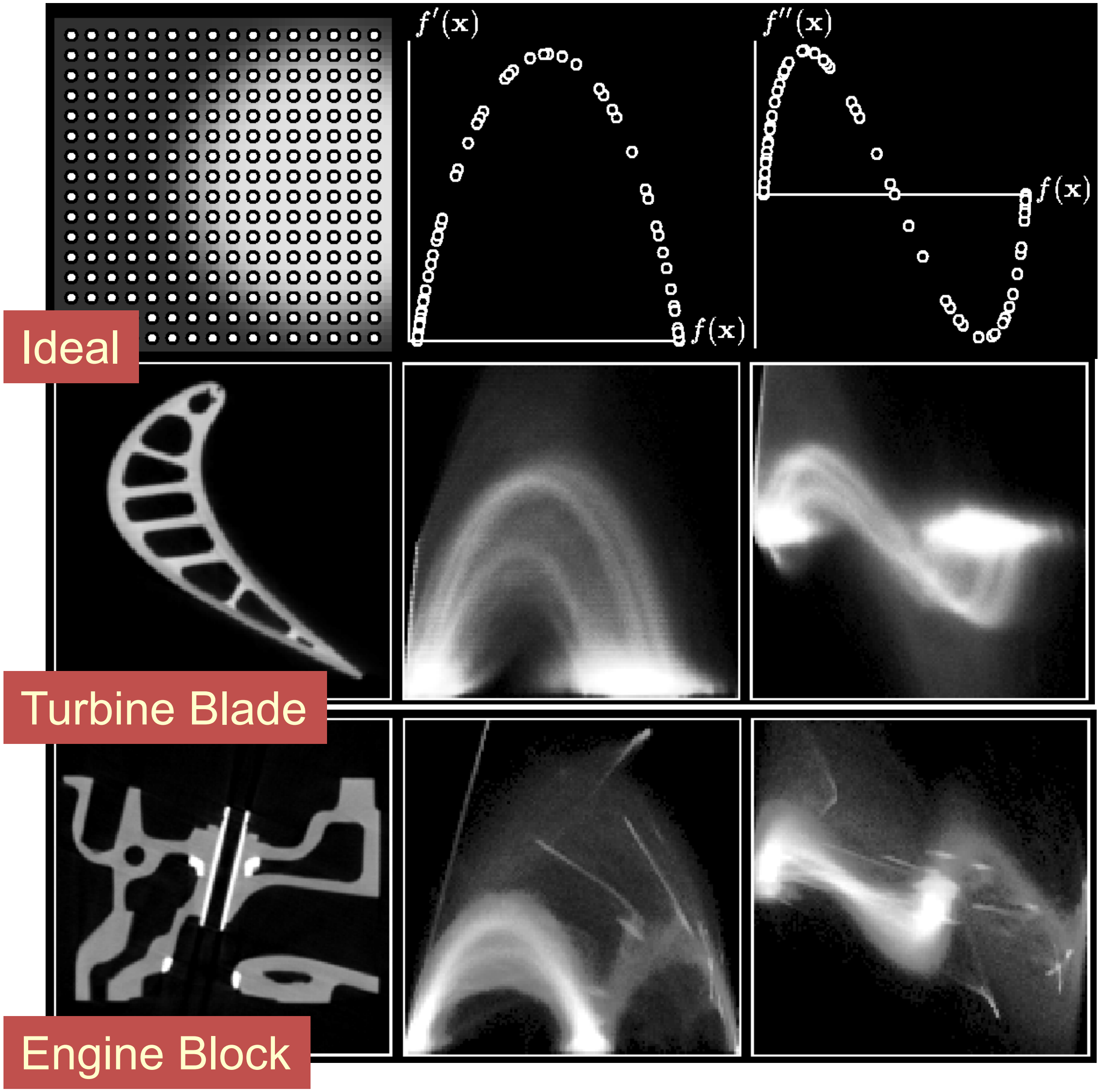



Edges at maximum of 1st derivative or zero-crossing of 2nd

# Scatterplots to find boundaries



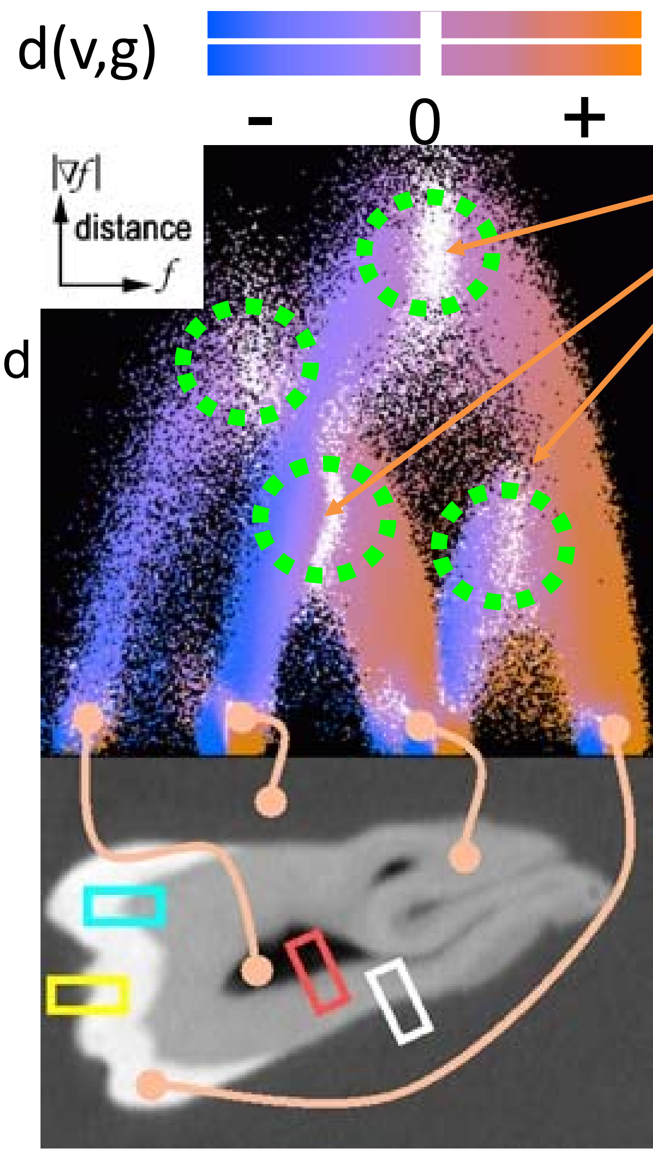

Project histogram volume to 2D scatterplots

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

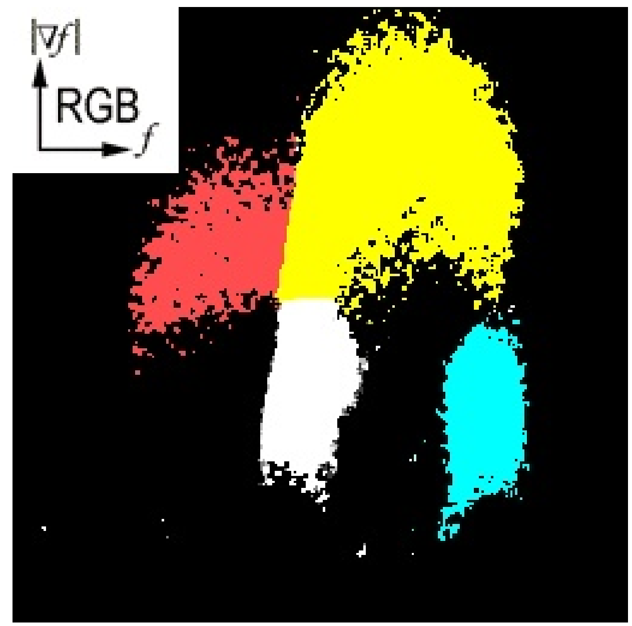
# Example: 2D transfer function for Tooth Data

Detected **4** distinct boundaries between **4** materials

- Pulp
- Background
- Dentine
- Enamel



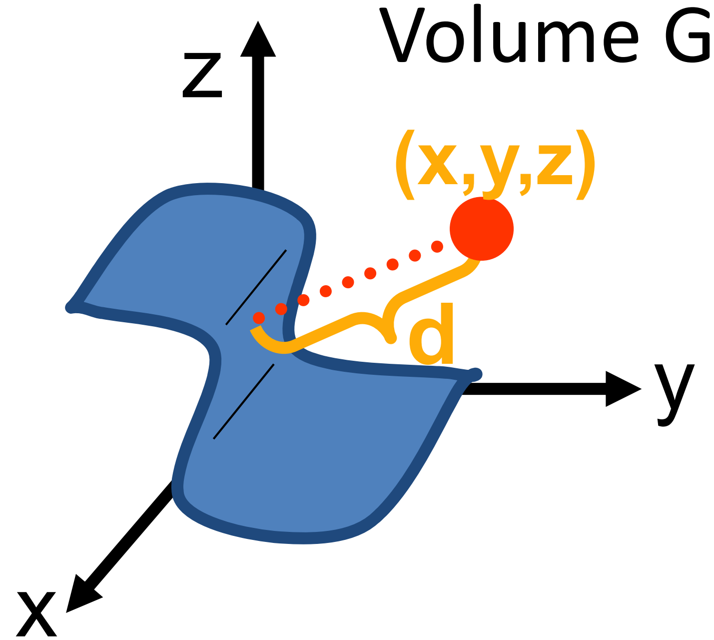

White regions in color mapped 2D distance function plot are boundary centers




Color transfer function

# Opacity Transfer Function: Analysis

## Volume Graphics Distance Map

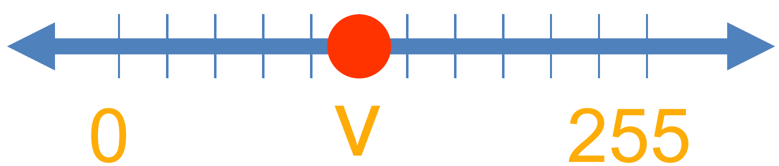



{ x y z }

3D position

d Signed distance to boundary

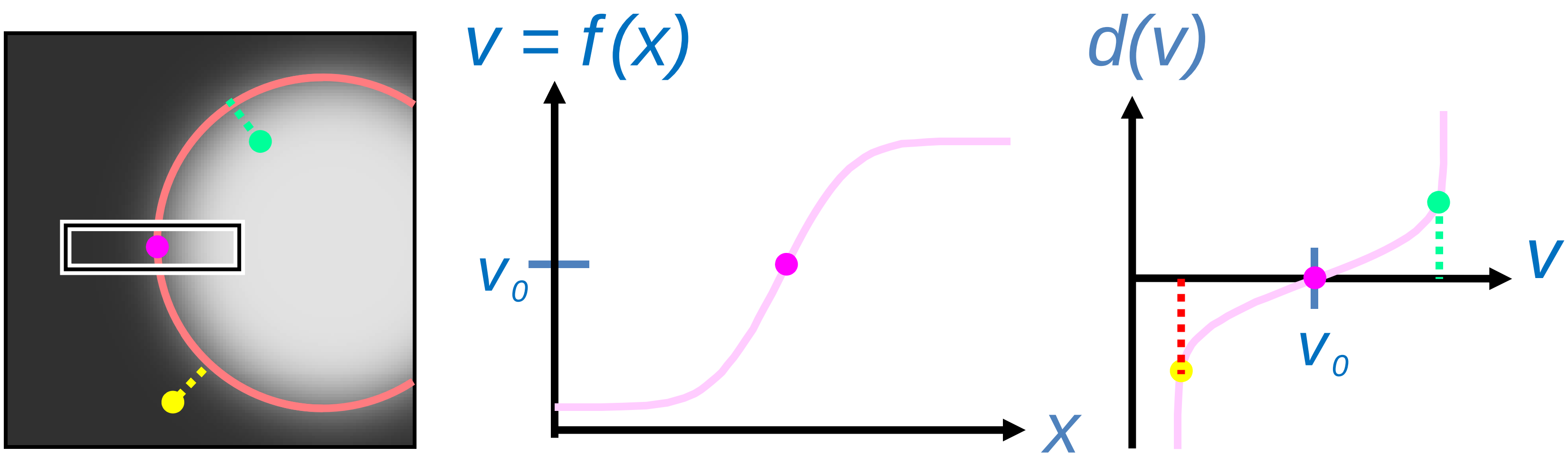
## New Distance Map in data range!




v

data value

# Opacity Transfer Function: New Distance Map

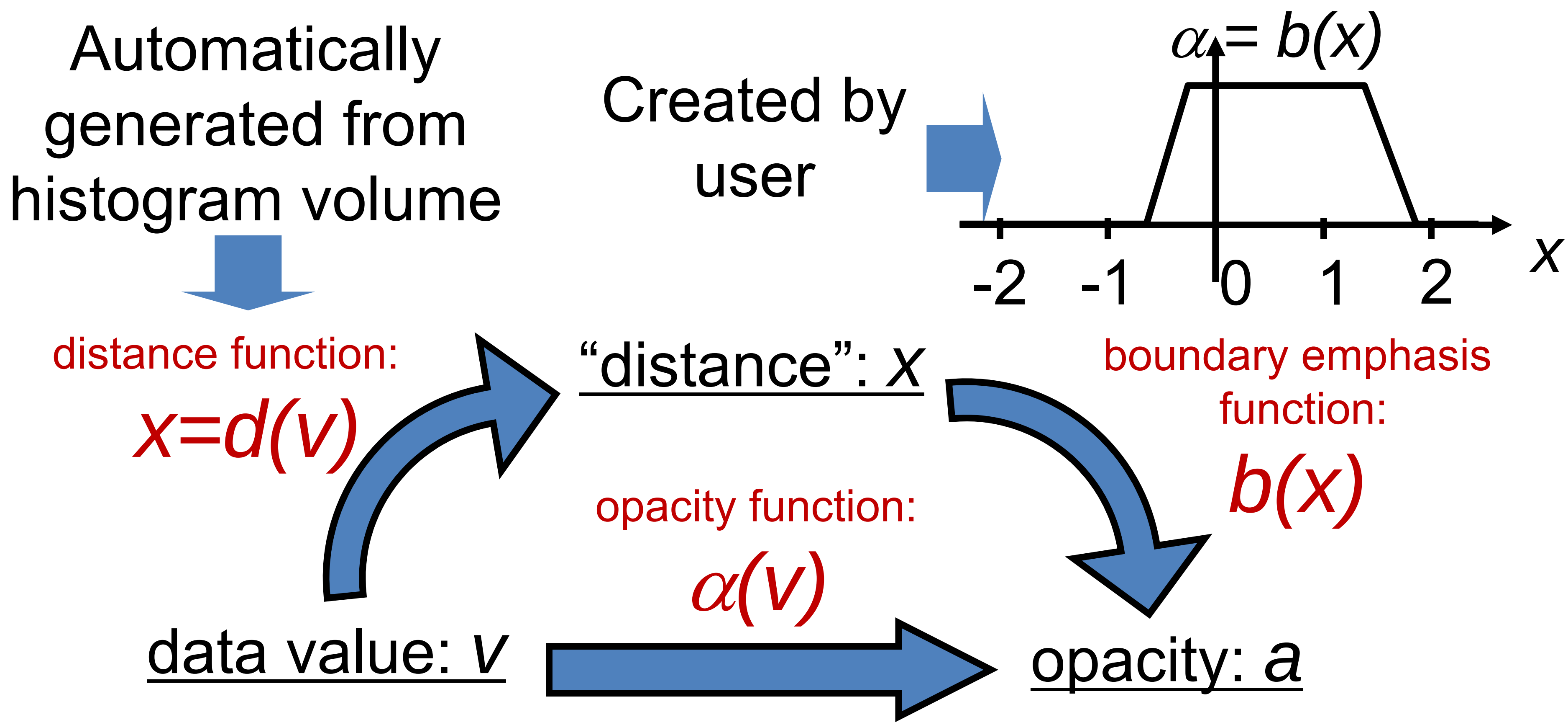



A nonlinear function

- Supports 2D **distance map**:

  d(v,g); g = gradient magnitude
- Produced automatically from histogram volume via boundary model

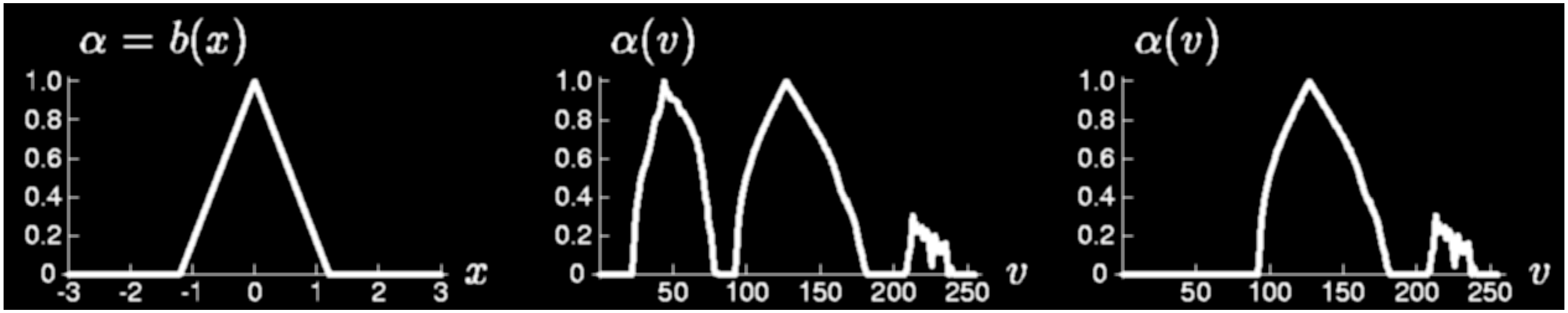
# Opacity Transfer Function: Whole Process

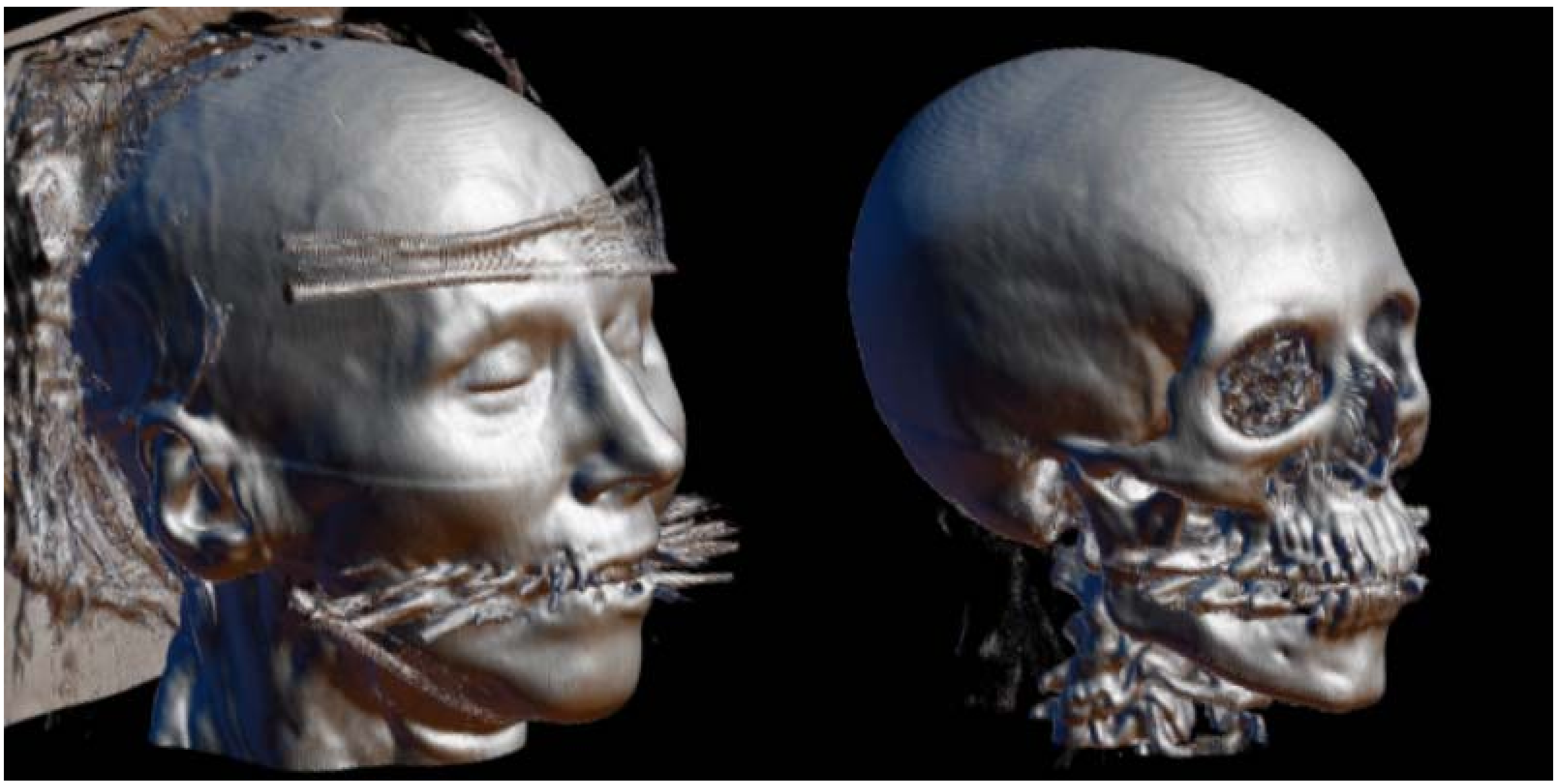



Opacity function: $\alpha(v) = b(d(v))$

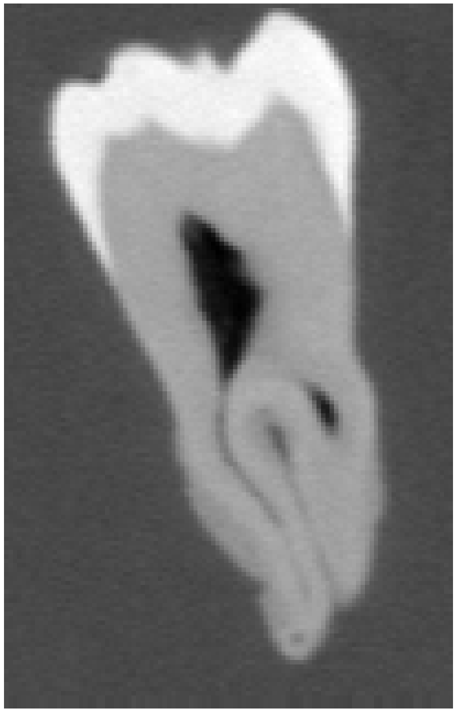
$\alpha(v,g) = b(d(v,g))$

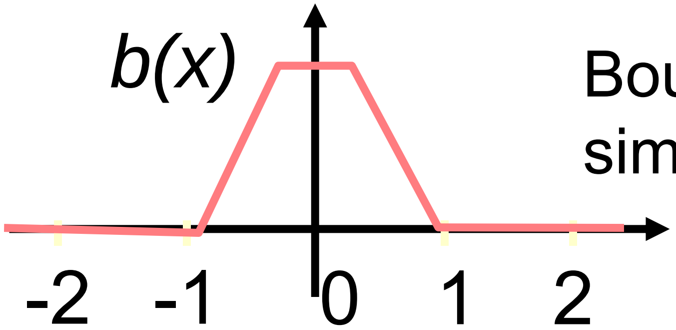
# Results: CT Head



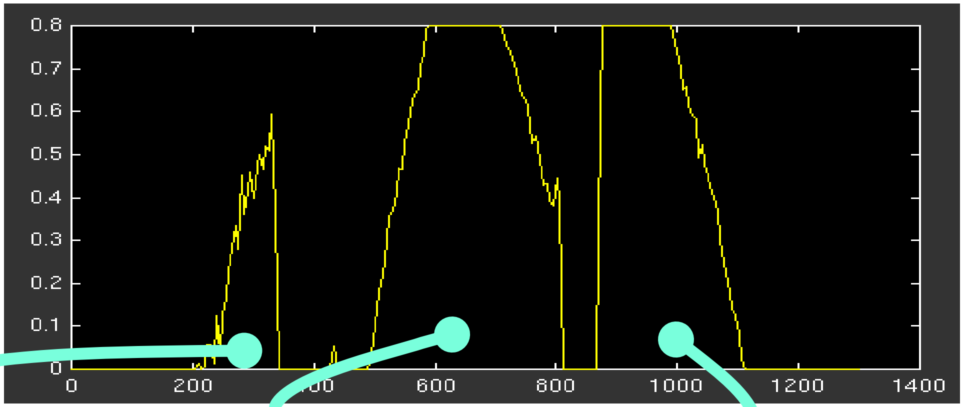
# Results: Tooth

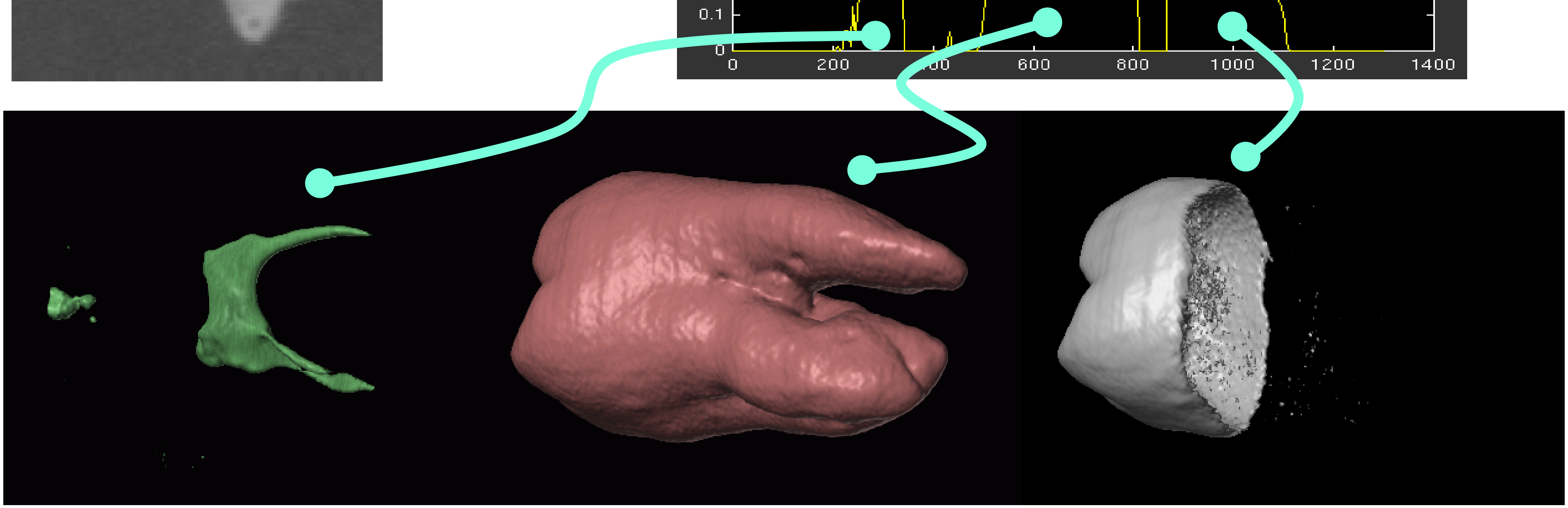





Boundary emphasis function simple to set

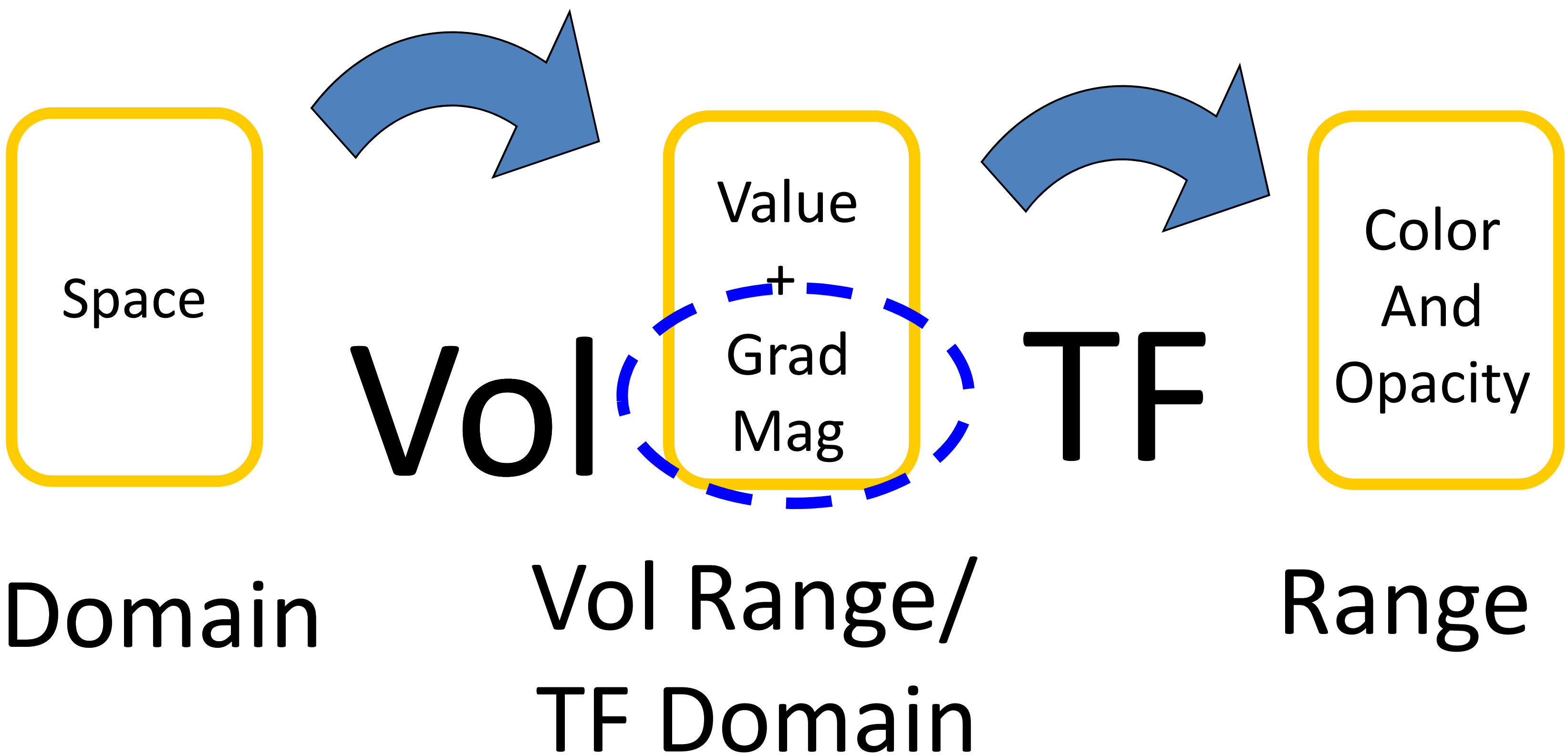
$a(v) = b(d(v))$



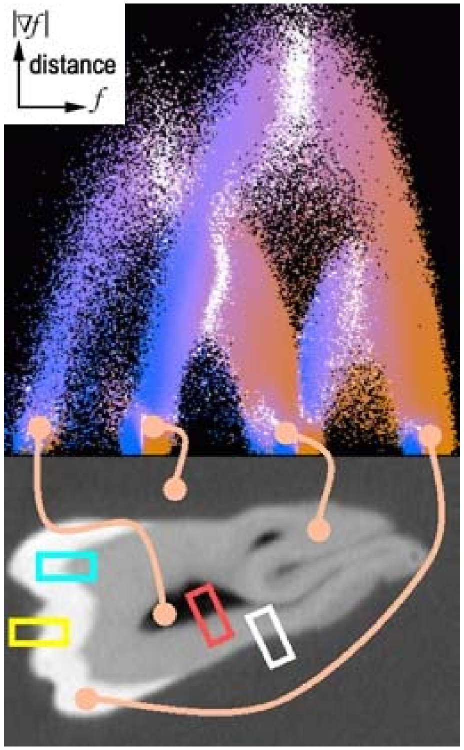



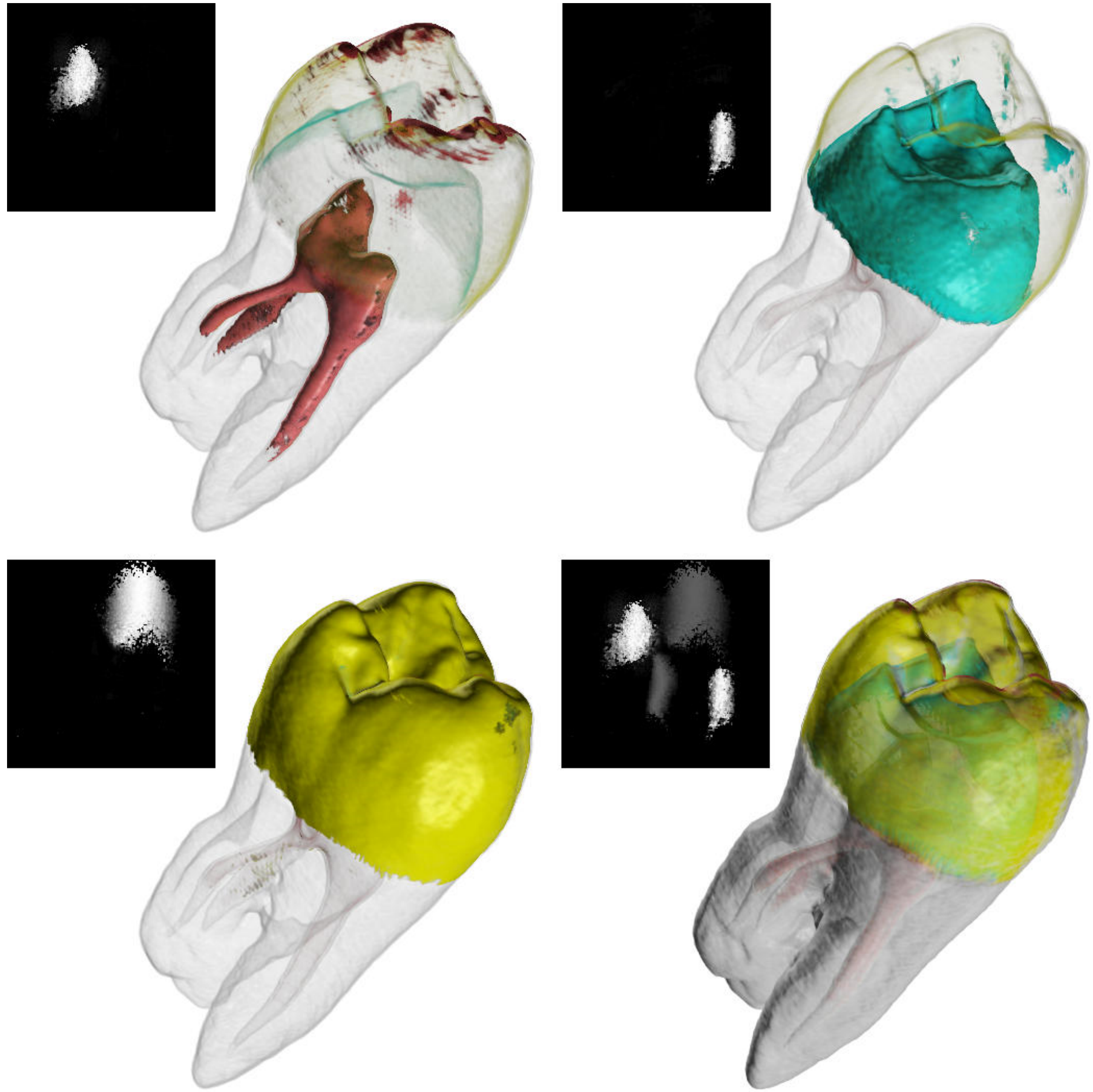
# Math Terminology

- Basic Transfer Functions:



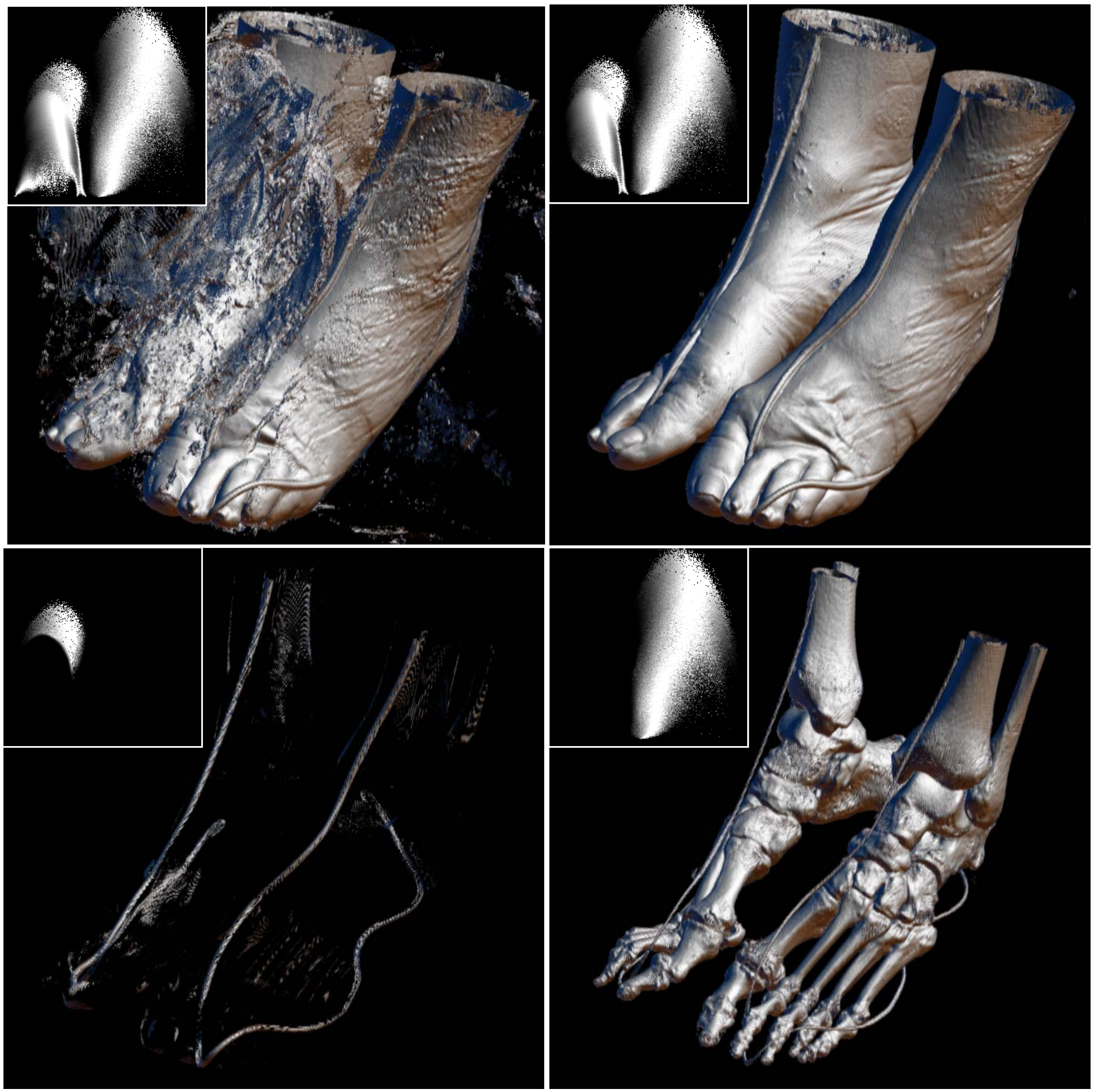
# 2D Opacity Functions



Mostly accurate isolation of all material boundaries

# 2D Opacity Functions



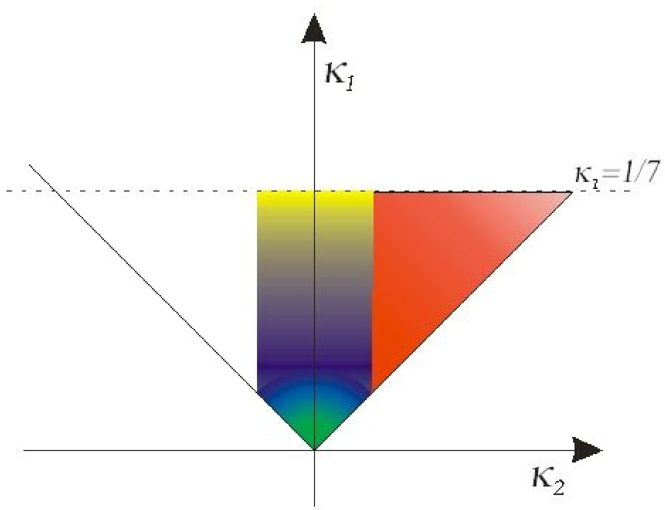
# TF Methods

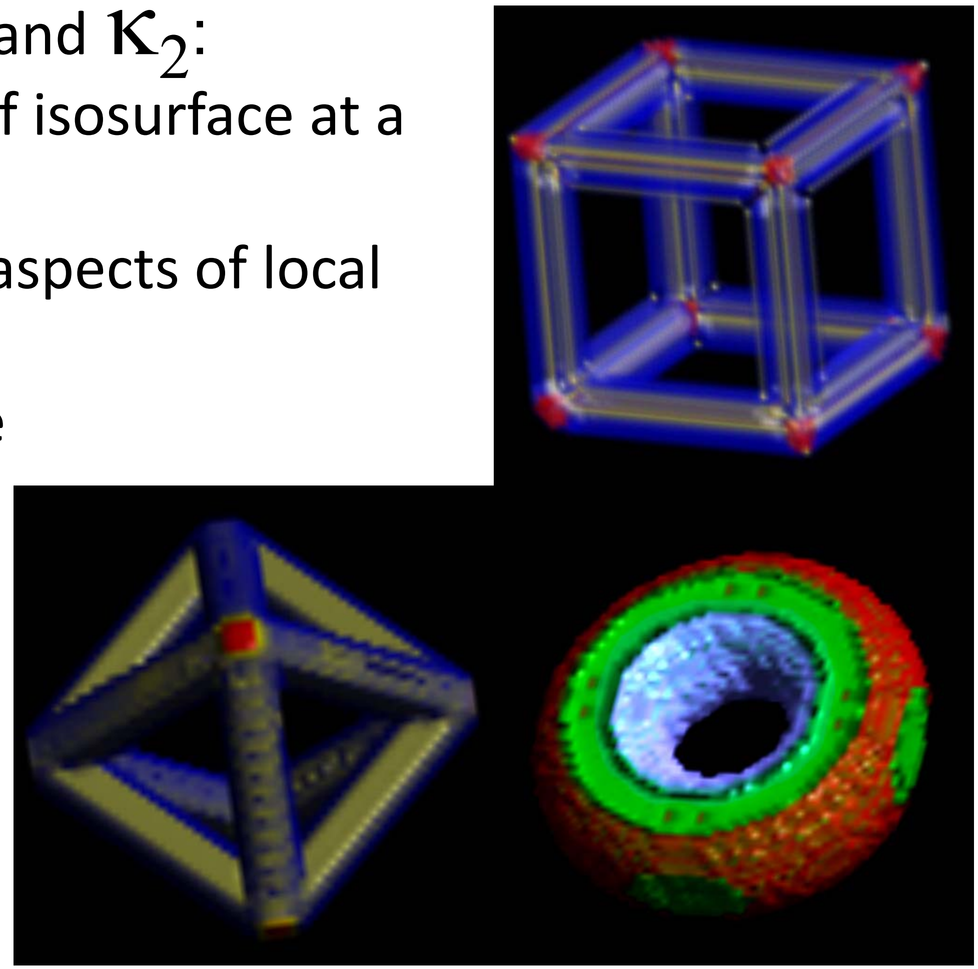
1. Trial and Error (manual)
2. Spatial Feature Detection
3. Image-Centric
4. Data-Centric
5. Others

# Curvature

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

- Uses 2D space of $\kappa_1$ and $\kappa_2$: principal curvatures of isosurface at a given point
- Graphically indicates aspects of local shape
- Specification is simple



# 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, $1^{st}$, $2^{nd}$ derivative)
- "Paint" into the transfer function domain