

CS 247 – Scientific Visualization

Lecture 21: Volume Rendering, Pt. 8 [preview]

Markus Hadwiger, KAUST



Reading Assignment #11 (until Apr 11)

Read (required):

- Real-Time Volume Graphics, Chapter 10
(Transfer Functions Reloaded)
- Paper:

Joe Kniss, Gordon Kindlmann, Charles Hansen,

Multidimensional Transfer Functions for Interactive Volume Rendering,
IEEE Transactions on Visualization and Comp. Graph. (TVCG) 2002,

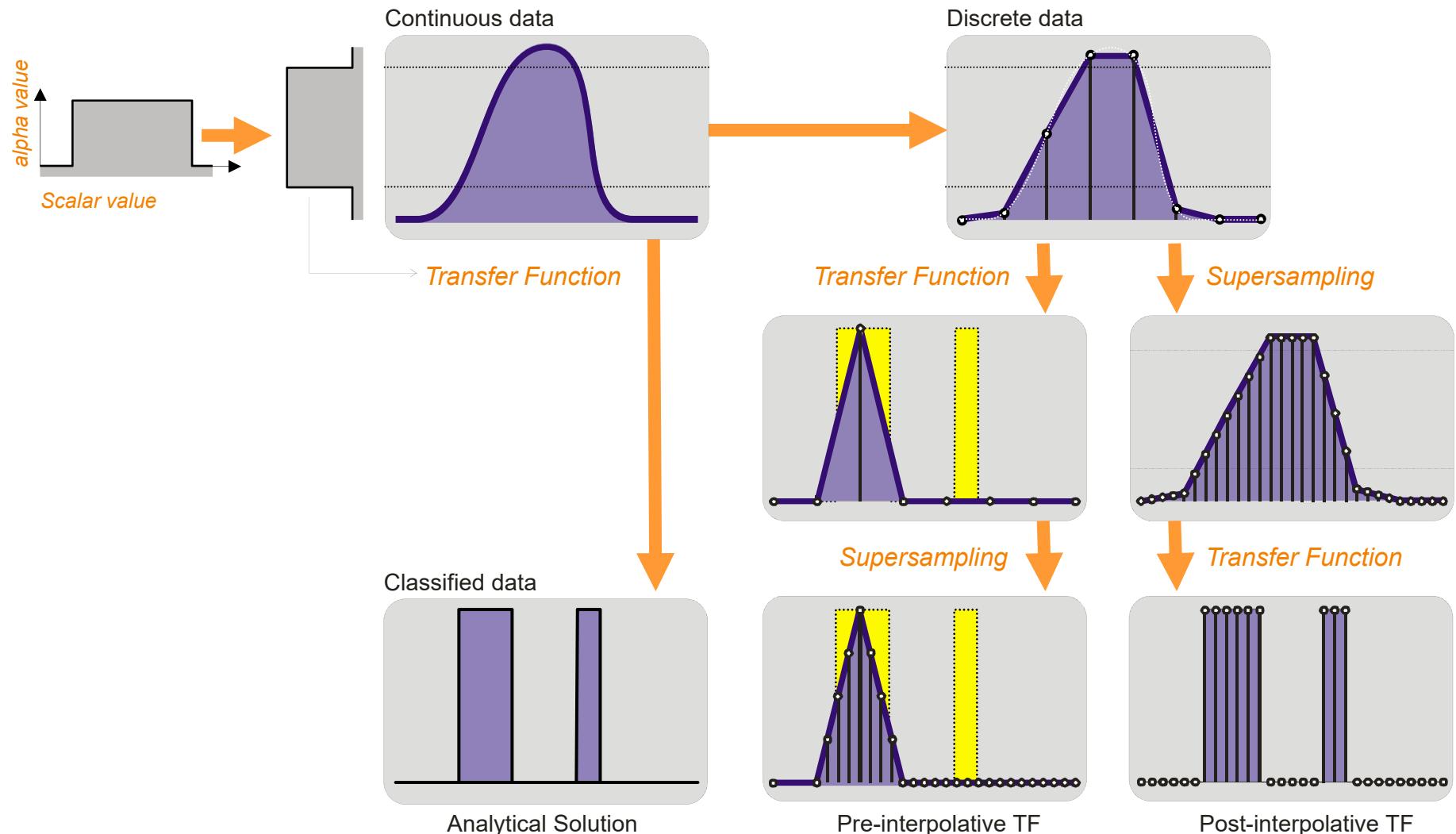
<https://ieeexplore.ieee.org/document/1021579>

Read (optional):

- Real-Time Volume Graphics, Chapter 14
(Non-Photorealistic and Illustrative Techniques)

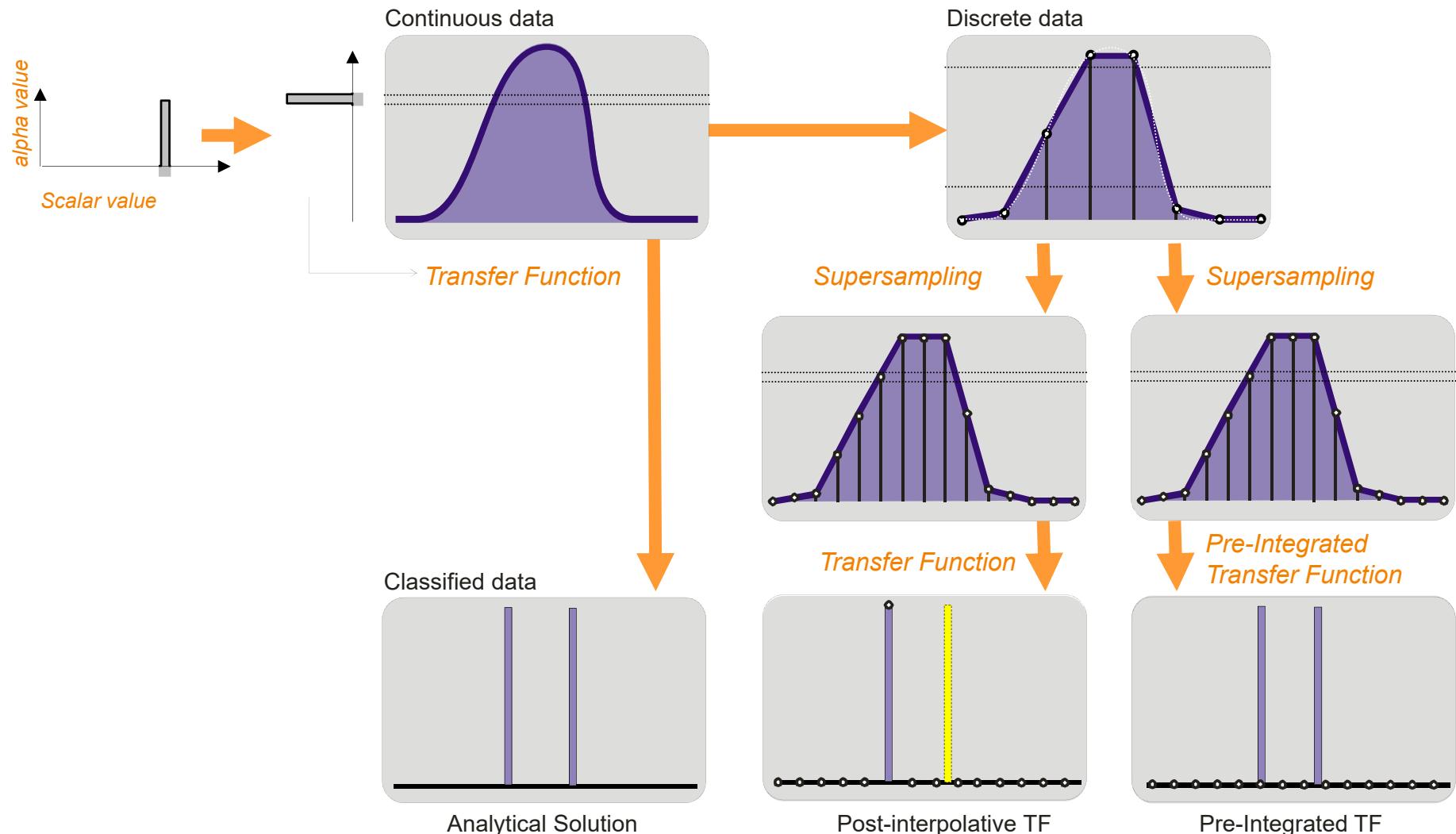


Pre- vs Post-Classification





Post- vs. Pre-Integrated Classification



2D Transfer Functions

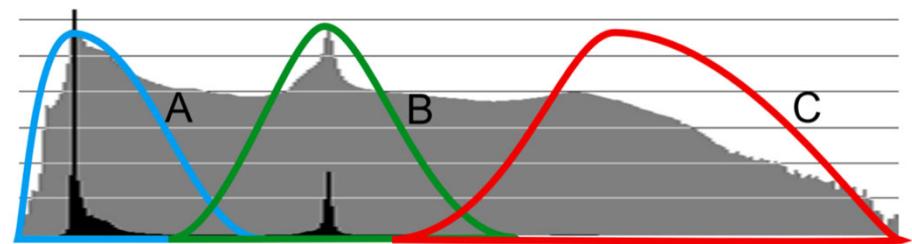


1D transfer function

Horizontal axis: scalar value

Vertical axis: number of voxels

1D histogram



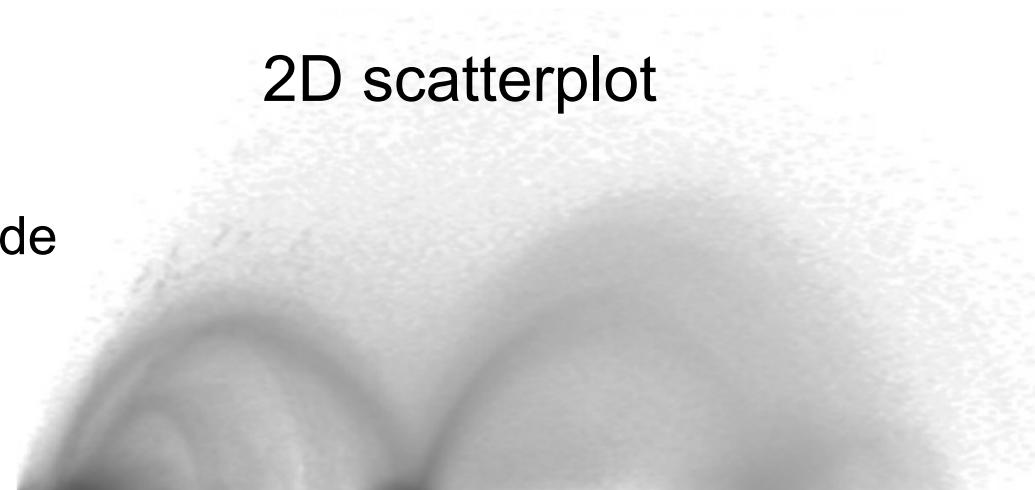
2D transfer function

Horizontal axis: scalar value

Vertical axis: gradient magnitude

Brightness: number of voxels
(here: darker means more)

2D scatterplot





2D Transfer Functions

1D transfer function

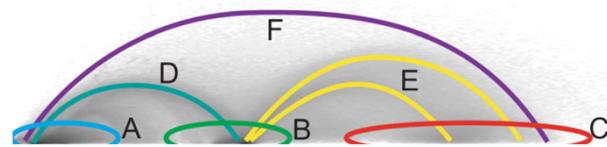
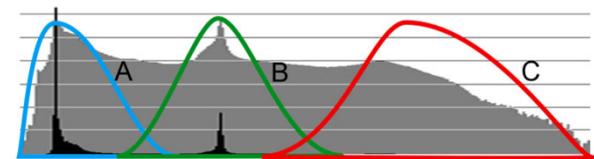
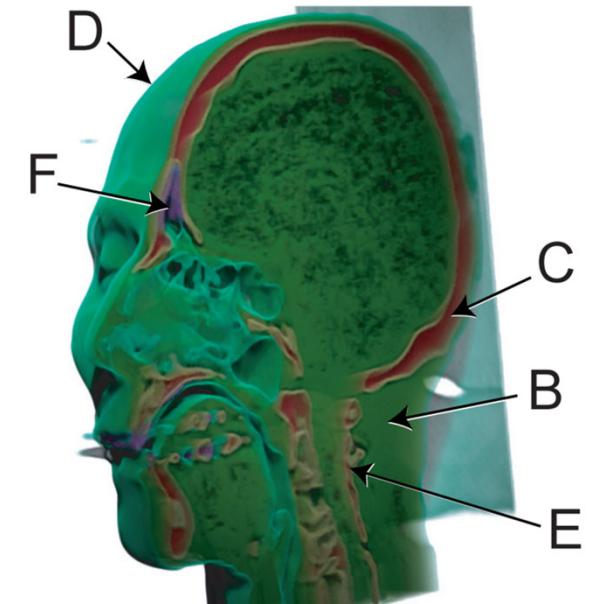
Horizontal axis: scalar value

Vertical axis: number of voxels

2D transfer function

Horizontal axis: scalar value

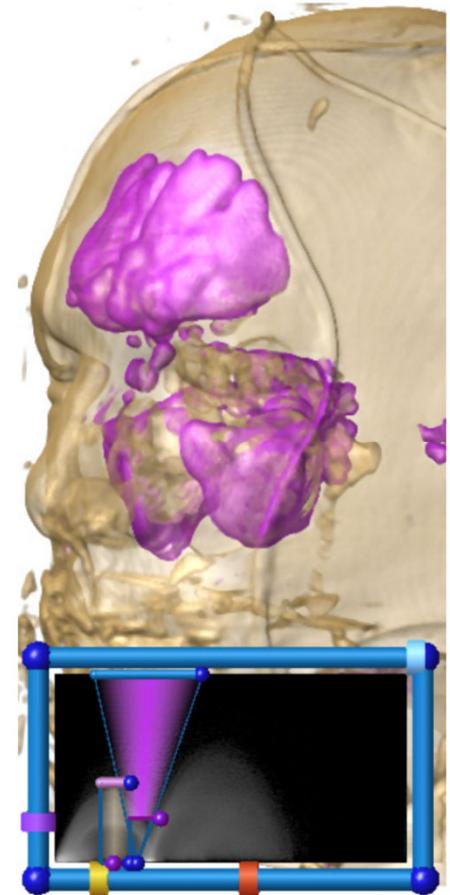
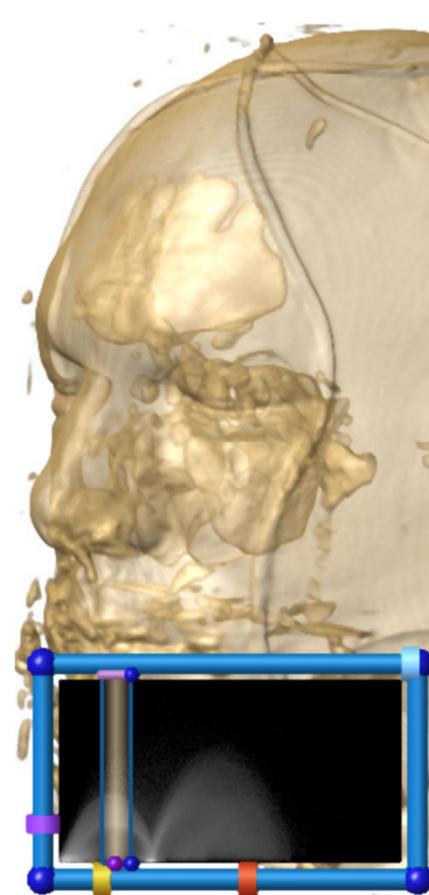
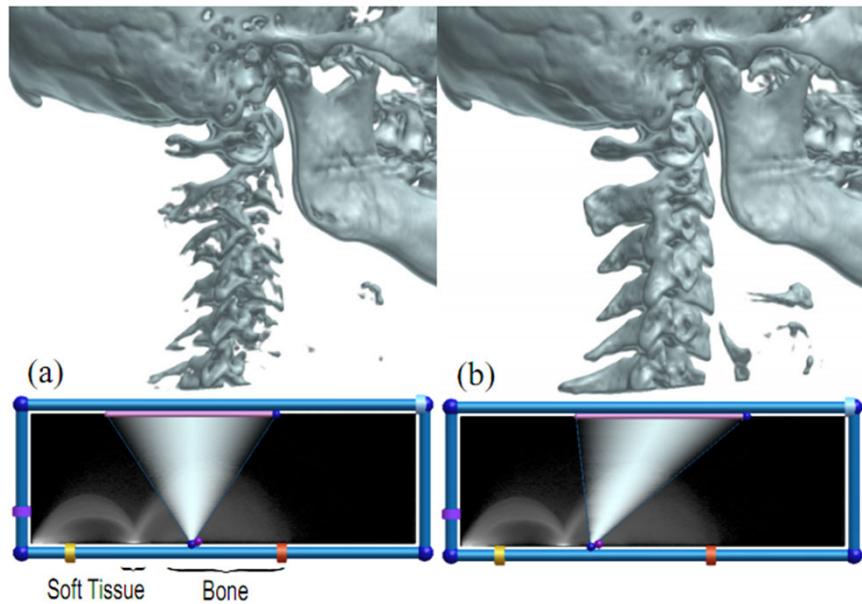
Vertical axis: gradient magnitude





2D Transfer Functions

Comparisons

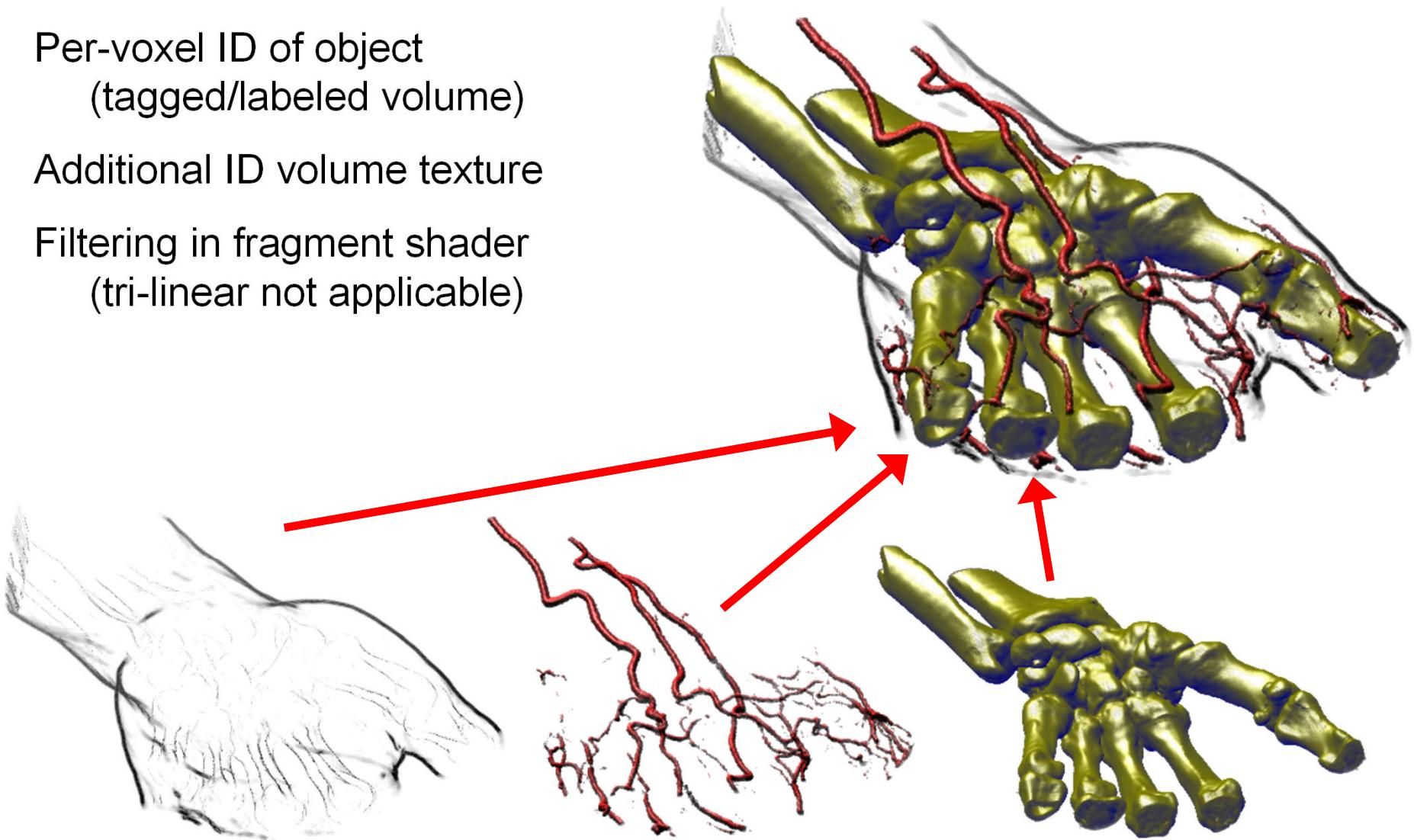


[Kniss et al. 2002]

Rendering Segmented Volumes (1)



Per-voxel ID of object
(tagged/labeled volume)
Additional ID volume texture
Filtering in fragment shader
(tri-linear not applicable)



Rendering Segmented Volumes (2)

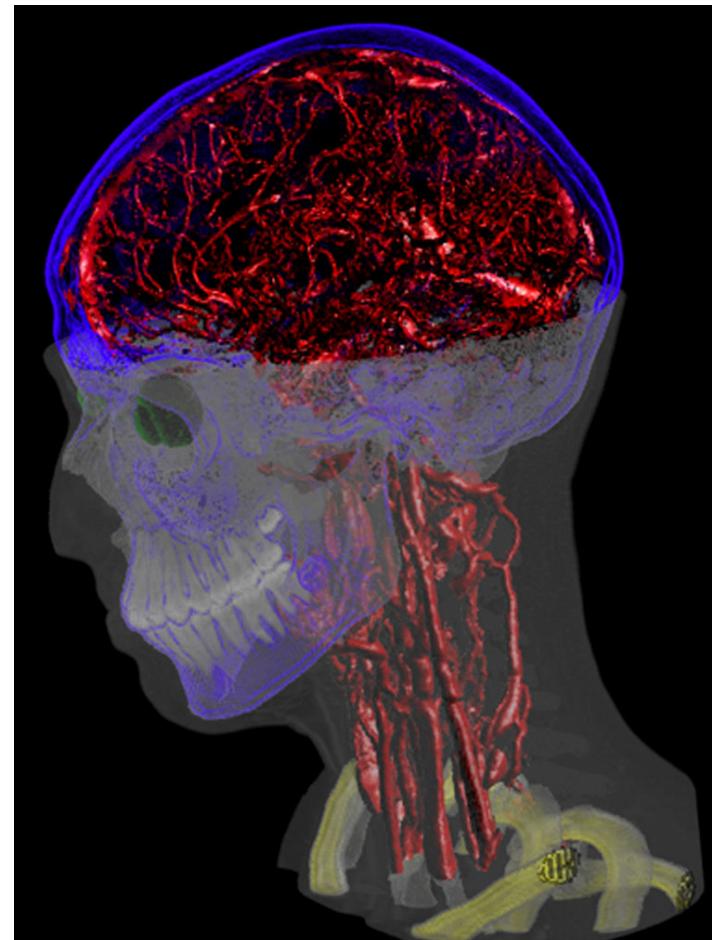


Focus and context

Per-object transfer function

Per-object rendering mode

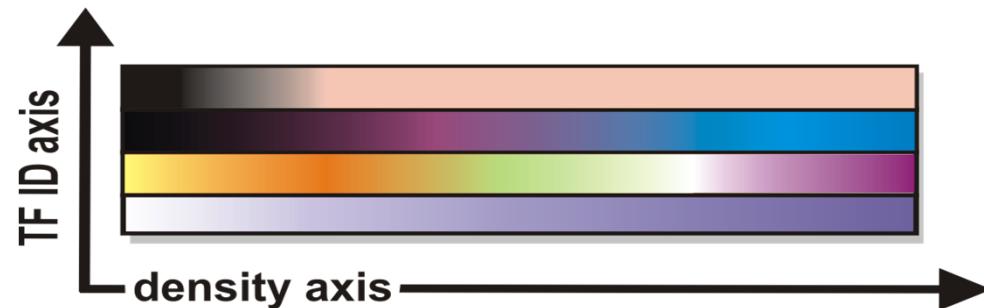
Per-object compositing





Per-Object Transfer Functions

Put all transfer functions in one global TF texture



index with object ID
as additional axis

```
tf_coords.x = tex3D( density_tex, sample_pos );  
tf_coords.y = tex3D( objectid_tex, sample_pos );  
classified_sample.rgba = tex2D( tf_tex, tf_coords );
```

1D transfer functions → 2D texture

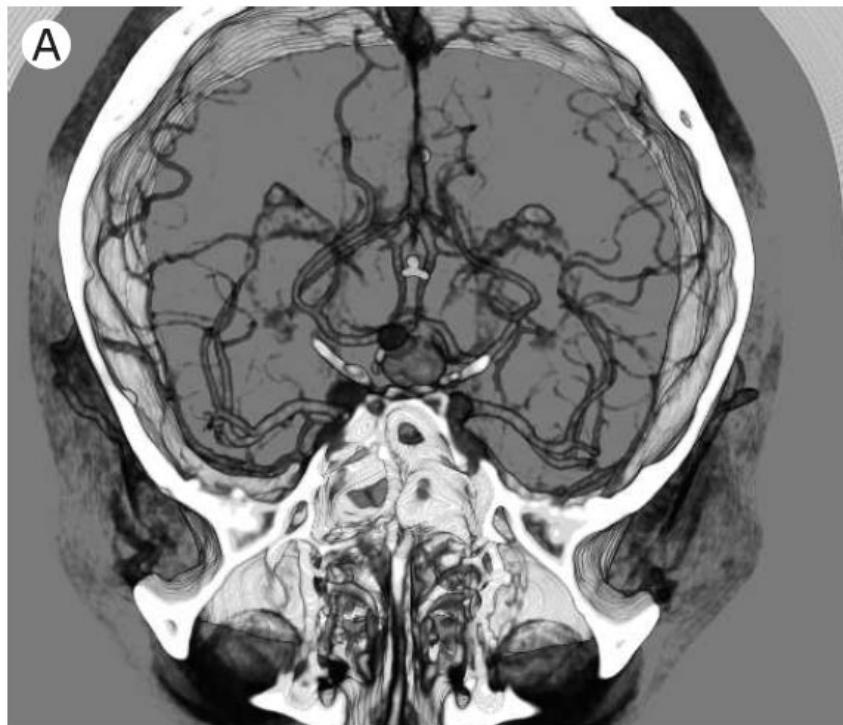
2D transfer functions → 3D texture



Maximum Intensity Projection

Alternative compositing mode (no alpha blending)

Keeps structure of maximum intensity visible





Volumetric Boundary Contours (1)

Based on view direction and gradient magnitude

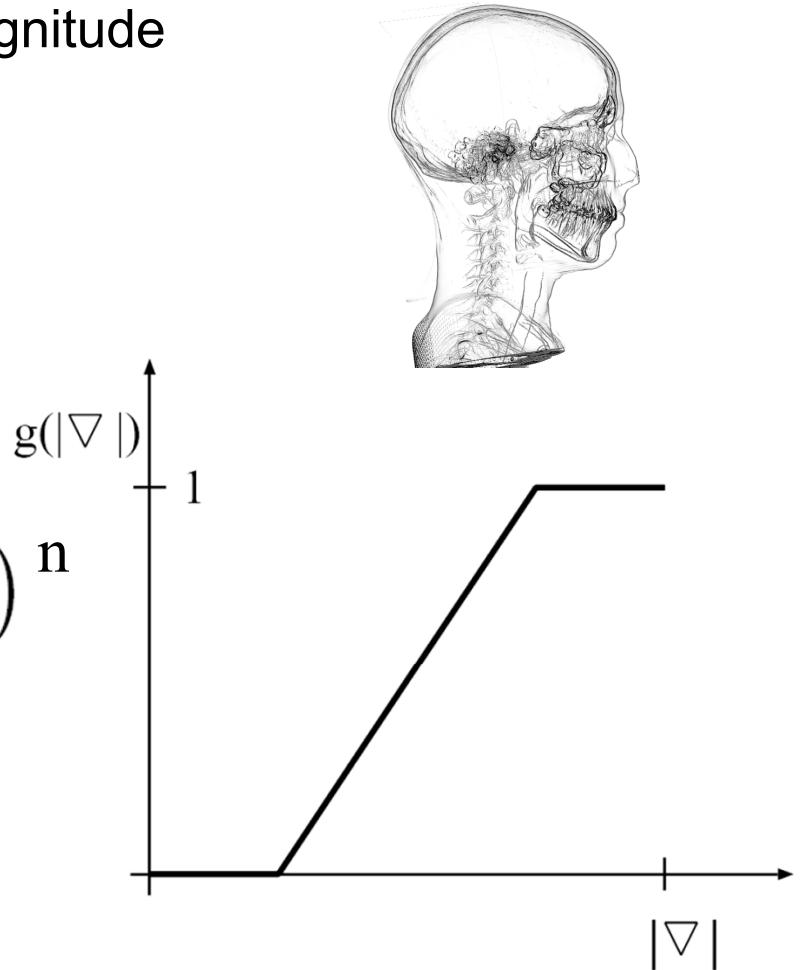
Global boundary detection
instead of isosurface

Gradient magnitude window $g(\cdot)$

$$\mathbf{I} = g(|\nabla f|) \cdot (1 - |\mathbf{v} \cdot \mathbf{n}|)^n$$

Exponent determines silhouette range

Does not work for distance fields!

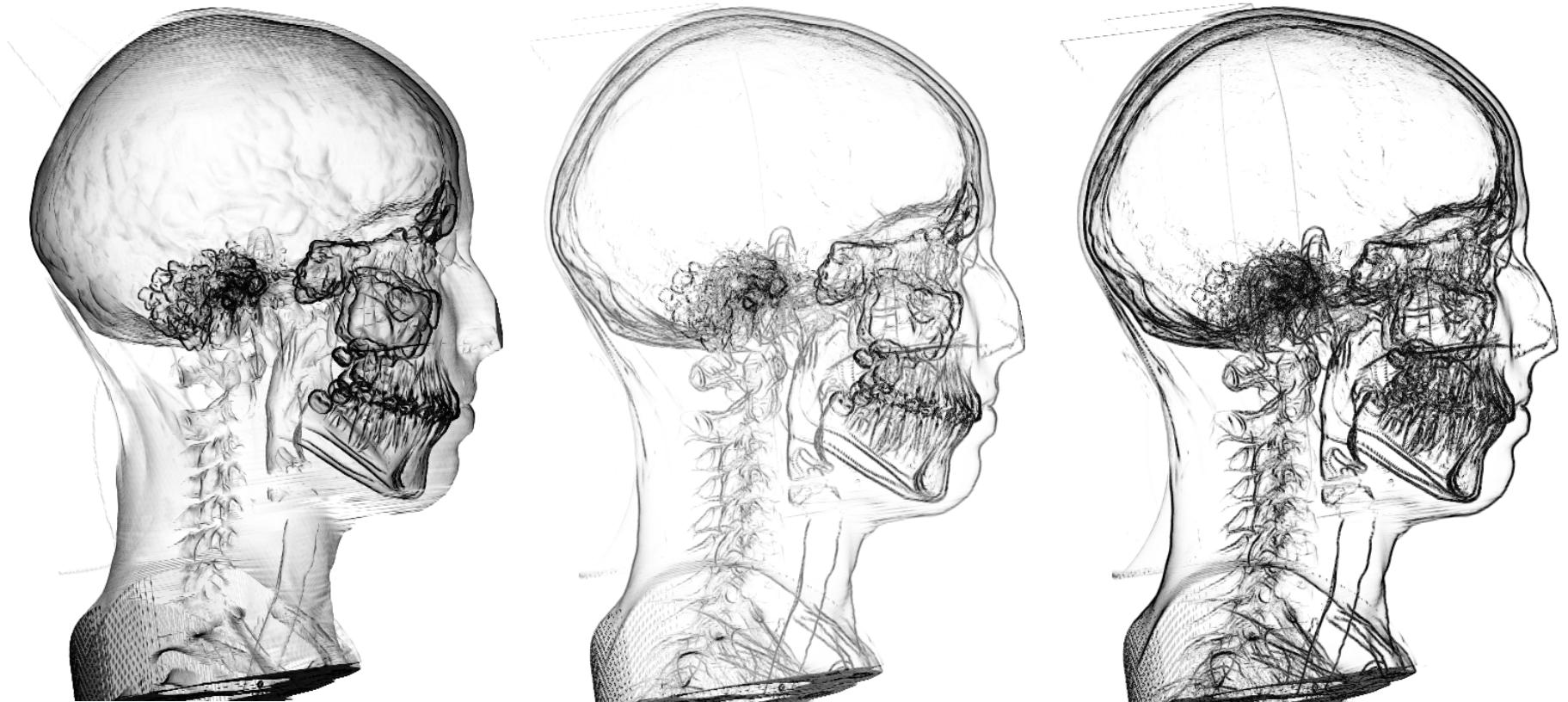




Volumetric Boundary Contours (2)

Gradient magnitude window is main parameter

Exponent between 4 and 16 is good choice



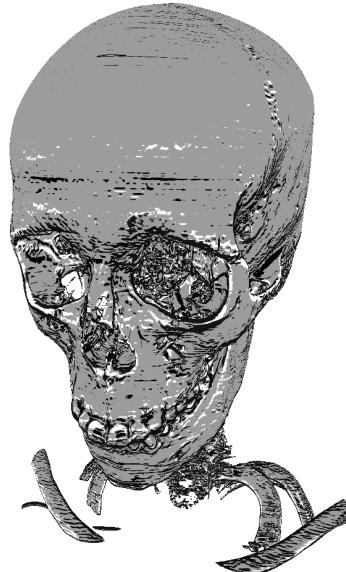
Curvature-Based Transfer Functions

Curvature-Based Isosurface Illustration

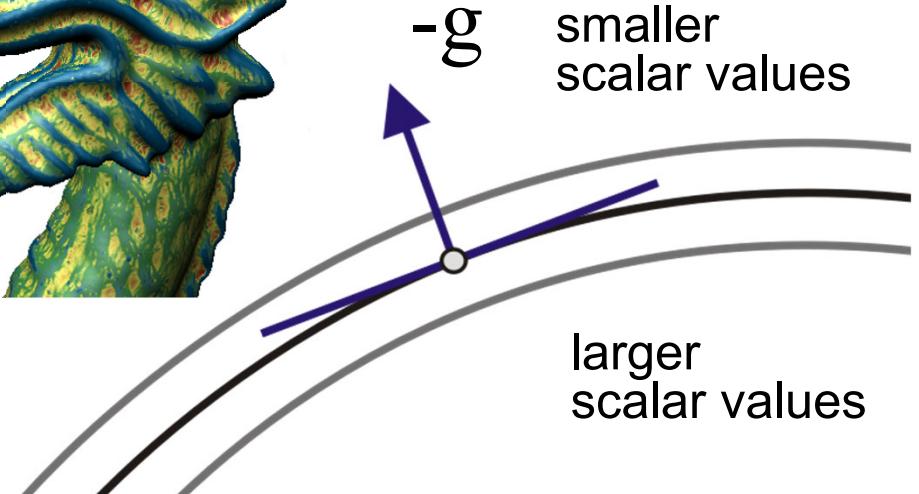


Curvature measure color mapping

Curvature directions; ridges and valleys



- Implicit surface curvature
- Isosurface through a point



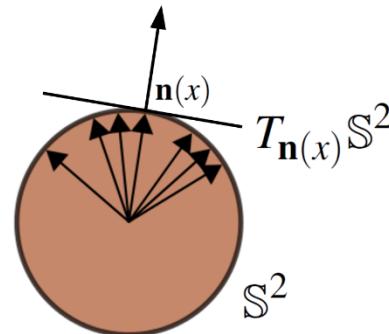
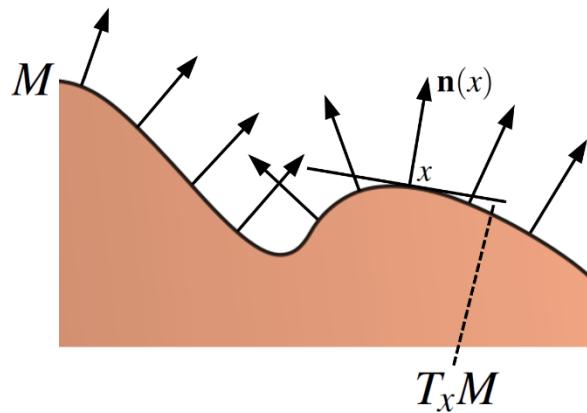


Curvature and Shape Operator

Gauss map

$$\mathbf{n}: M \rightarrow \mathbb{S}^2$$

$$x \mapsto \mathbf{n}(x)$$



Differential of Gauss map

$$d\mathbf{n}: TM \rightarrow T\mathbb{S}^2$$

$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

$$(d\mathbf{n})_x: T_x M \rightarrow T_{\mathbf{n}(x)} \mathbb{S}^2$$

$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

Shape operator (Weingarten map)

$$\mathbf{S}: TM \rightarrow TM$$

$$\mathbf{S}_x: T_x M \rightarrow T_x M$$

$$\mathbf{v} \mapsto \mathbf{S}_x(\mathbf{v}) = d\mathbf{n}(\mathbf{v})$$

Principal curvature magnitudes and directions are eigenvalues and eigenvectors of shape operator \mathbf{S}

$$T_{\mathbf{n}(x)} \mathbb{S}^2 \cong T_x M$$

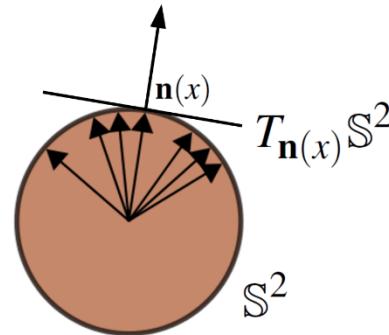
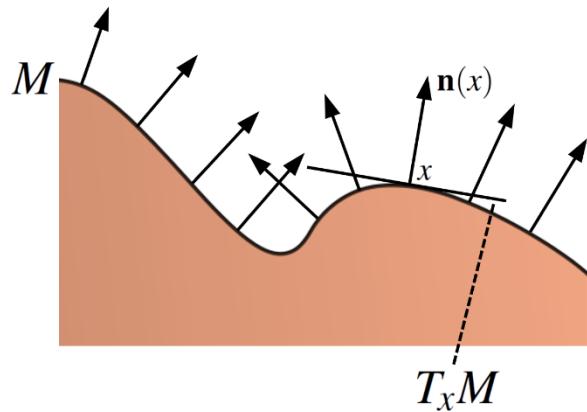


Curvature and Shape Operator

Gauss map

$$\mathbf{n}: M \rightarrow \mathbb{S}^2$$

$$x \mapsto \mathbf{n}(x)$$



Differential of Gauss map

$$d\mathbf{n}: TM \rightarrow T\mathbb{S}^2$$

$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

$$(d\mathbf{n})_x: T_x M \rightarrow T_{\mathbf{n}(x)} \mathbb{S}^2$$

$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

Shape operator (Weingarten map)

$$\mathbf{S}: TM \rightarrow TM$$

$$\mathbf{S}_x: T_x M \rightarrow T_x M$$

$$\mathbf{v} \mapsto \mathbf{S}_x(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbf{n}$$

Principal curvature magnitudes and directions are eigenvalues and eigenvectors of shape operator \mathbf{S}

$$T_{\mathbf{n}(x)} \mathbb{S}^2 \cong T_x M$$

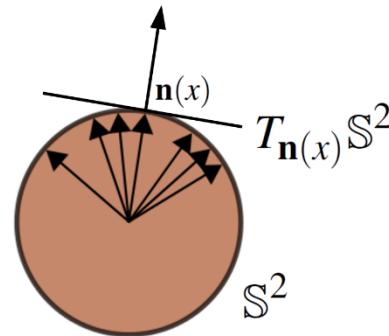
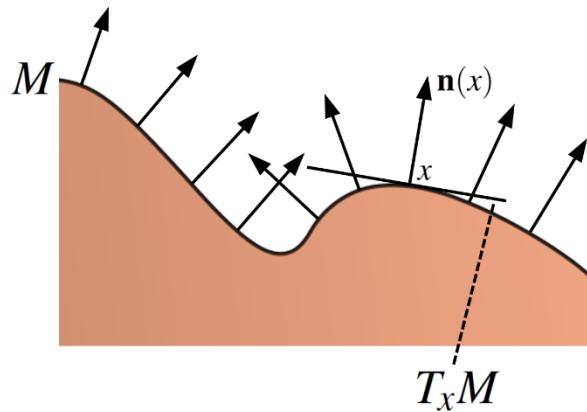


Curvature and Shape Operator

Gauss map

$$\mathbf{n}: M \rightarrow \mathbb{S}^2$$

$$x \mapsto \mathbf{n}(x)$$



Differential of Gauss map

$$d\mathbf{n}: TM \rightarrow T\mathbb{S}^2$$

$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

$$(d\mathbf{n})_x: T_x M \rightarrow T_{\mathbf{n}(x)} \mathbb{S}^2$$

$$\mathbf{v} \mapsto d\mathbf{n}(\mathbf{v})$$

Shape operator (Weingarten map)

$$\mathbf{S}: TM \rightarrow TM$$

$$\mathbf{S}_x: T_x M \rightarrow T_x M$$

$$\mathbf{v} \mapsto \mathbf{S}_x(\mathbf{v}) = -\nabla_{\mathbf{v}} \mathbf{n}$$

Principal curvature magnitudes and directions are eigenvalues and eigenvectors of shape operator \mathbf{S}

$$T_{\mathbf{n}(x)} \mathbb{S}^2 \cong T_x M$$



(Extrinsic) Curvature

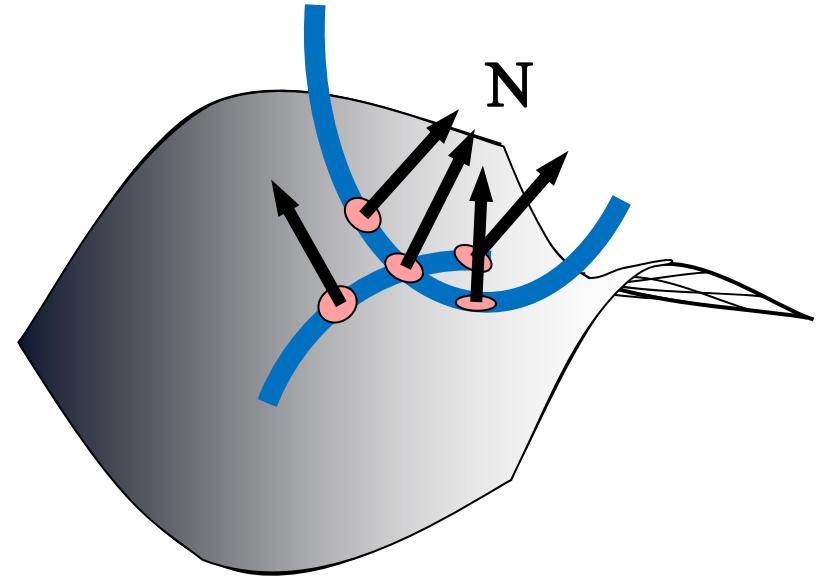
How fast do positional changes (in different directions) on the surface change the normal vector?

- Gauss map: assigns normal to each point

$$\begin{aligned}\mathbf{N}: M &\rightarrow \mathbb{S}^2, \\ x &\mapsto \mathbf{N}(x).\end{aligned}$$

- Differential of Gauss map:
Shape operator / Weingarten map

$$\begin{aligned}\mathrm{d}\mathbf{N}: T_x M &\rightarrow T_{\mathbf{N}(x)} \mathbb{S}^2, \\ \mathbf{v} &\mapsto \mathrm{d}\mathbf{N}(\mathbf{v}).\end{aligned}$$



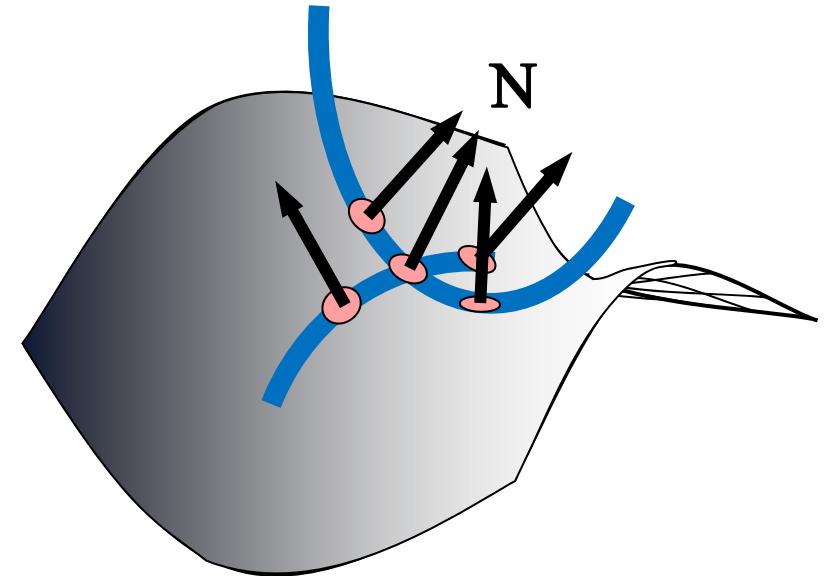
courtesy of Gordon Kindlmann



(Extrinsic) Curvature

Analyze shape operator S

- Eigenvalues: principal curvatures (magnitudes)
 - First and second principal curvature
 - Maximum: K_1
 - Minimum: K_2
- Eigenvectors: principal curv. directions
- Gaussian curvature (intrinsic!): $K_1 K_2$



courtesy of Gordon Kindlmann

(Extrinsic) Curvature Computation



Simple recipe for implicit isosurfaces in volume

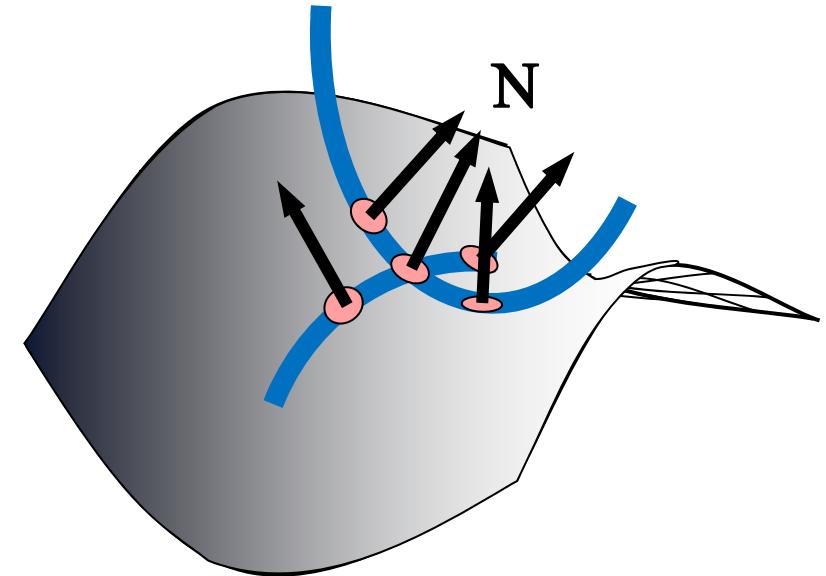
- Build on gradient and Hessian matrix
- Hessian contains curvature information

Transform Hessian into tangent space

- Curvature magnitudes:
Eigenvalues of 2×2 matrix
- Curvature directions:
Eigenvectors of 2×2 matrix

Alternative:

- Compute in 3D (see Real-Time Volume Graphics, 14.4.4)



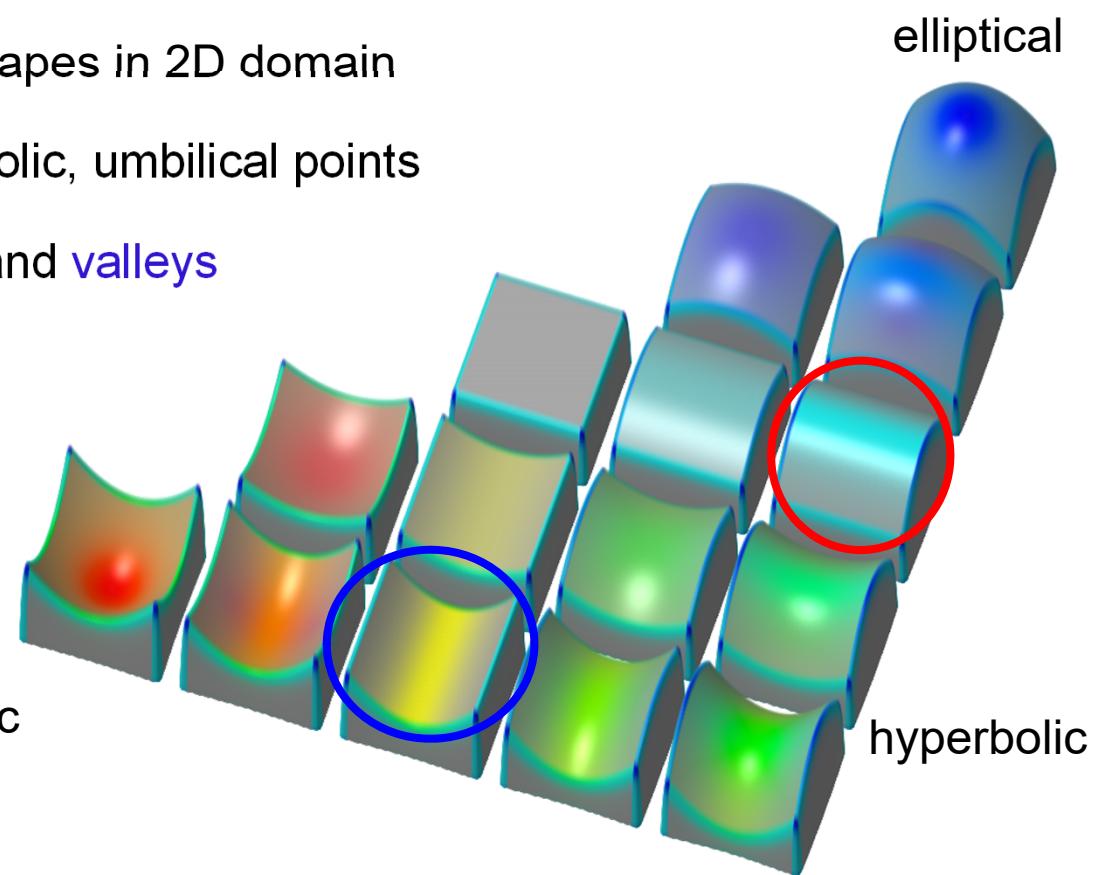
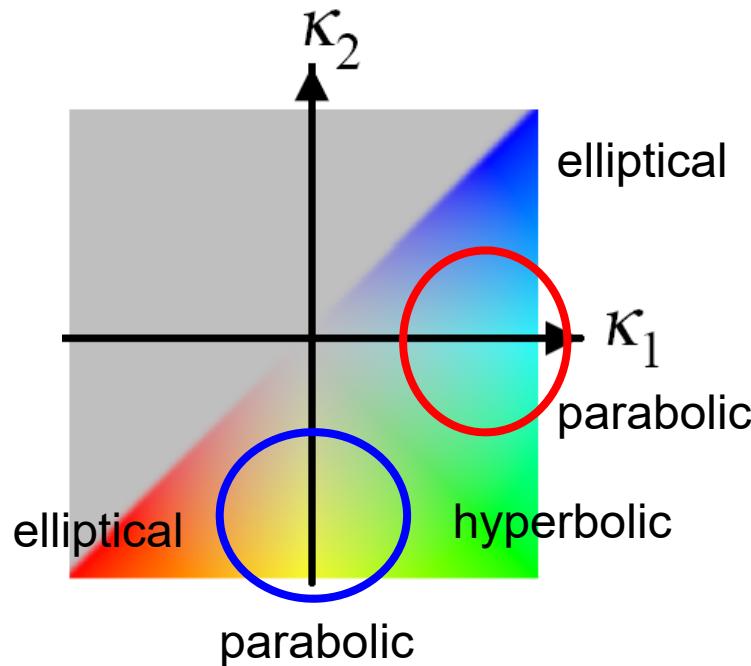
courtesy of Gordon Kindlmann

The Principal Curvature Domain



Maximum/minimum principal curvature magnitude

- Identification of different shapes in 2D domain
- Elliptical, parabolic, hyperbolic, umbilical points
- Feature lines: e.g., **ridges** and **valleys**

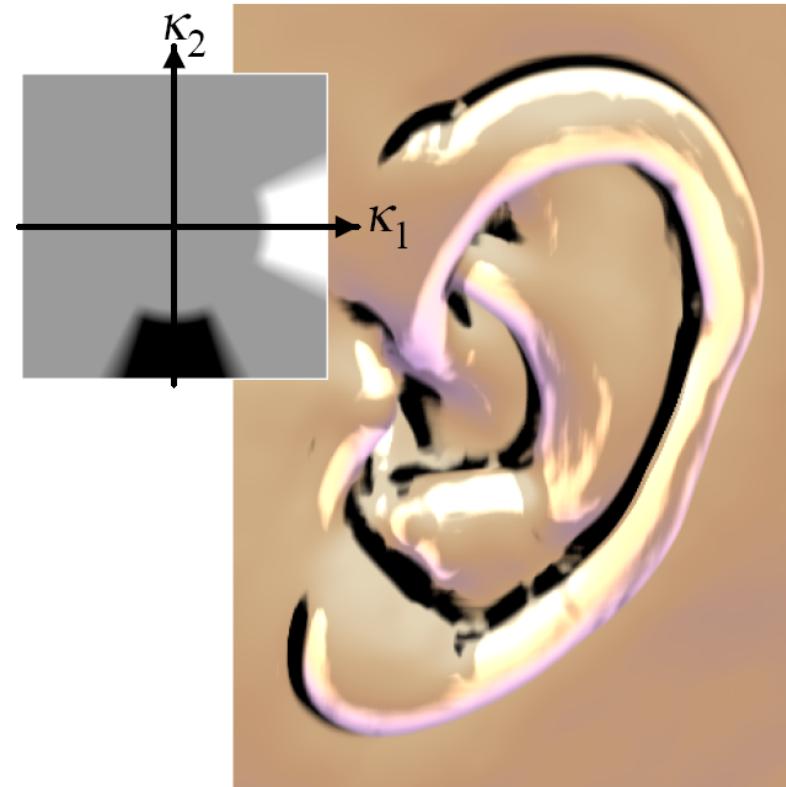
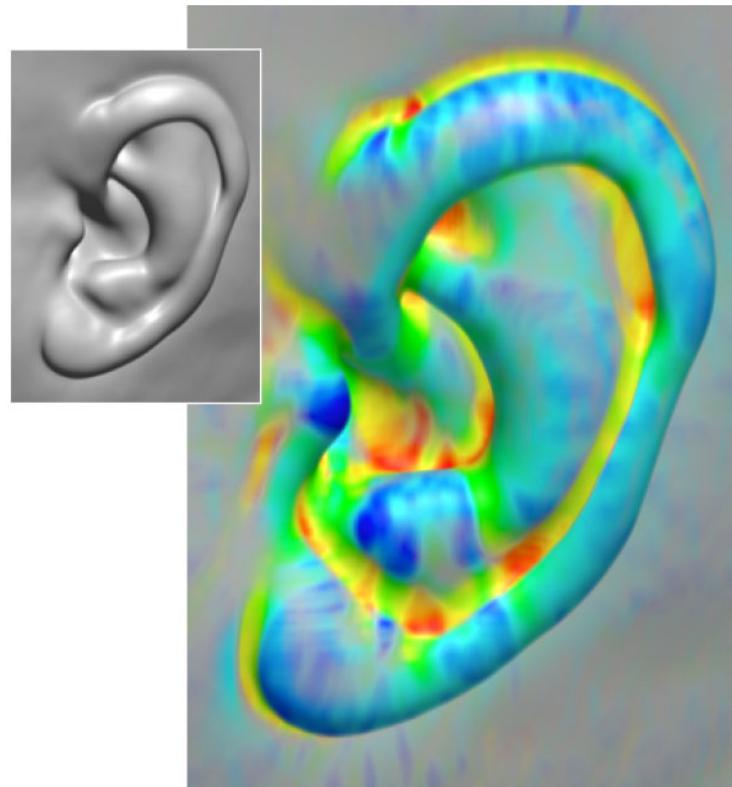


courtesy of Gordon Kindlmann

Curvature Transfer Functions



- Color coding of curvature domain
- Paint features: ridge and valley lines

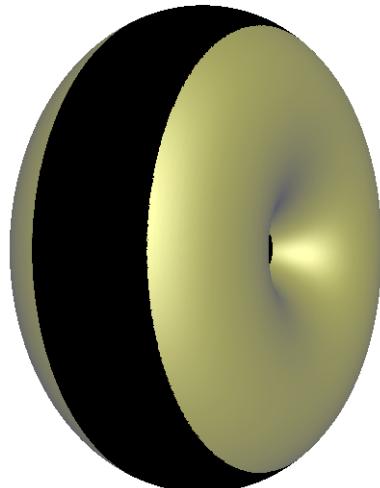
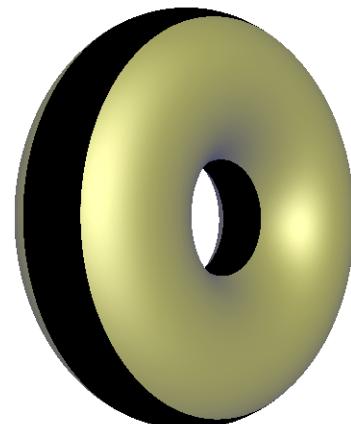
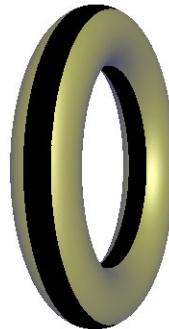
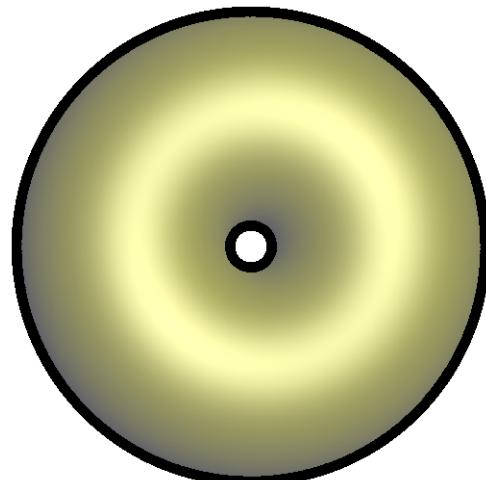
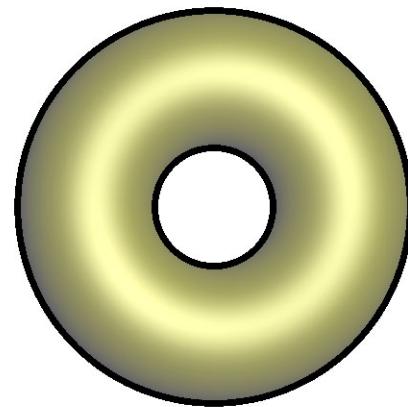
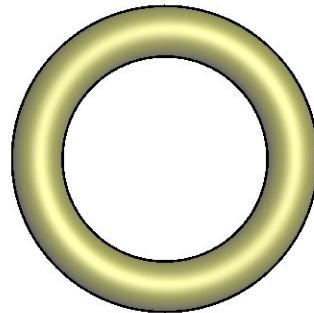


courtesy of Gordon Kindlmann

Problems of Implicit Surface Contours



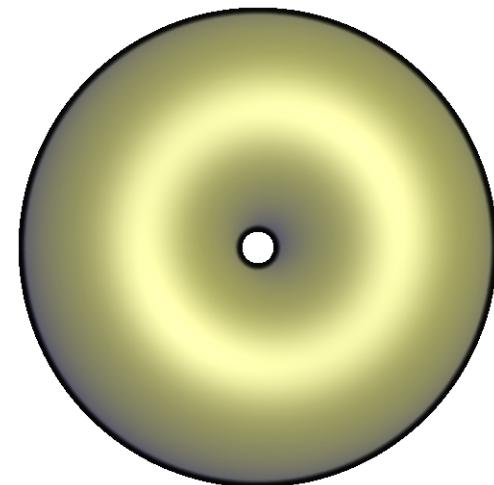
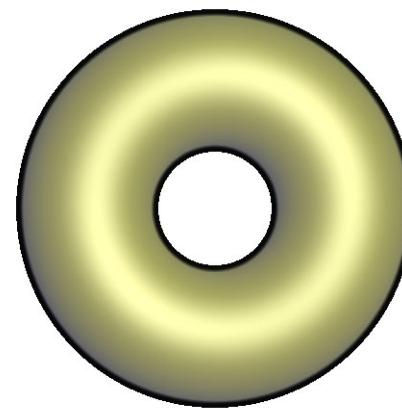
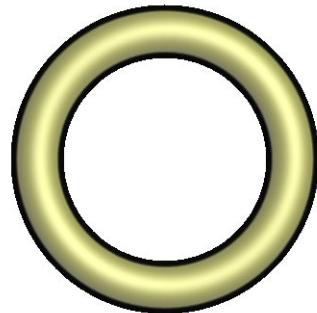
Constant threshold on $|\mathbf{v} \cdot \mathbf{n}|$



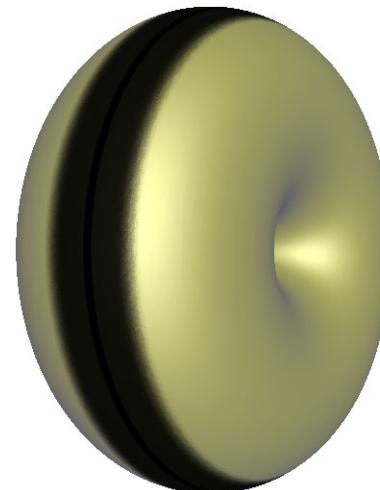
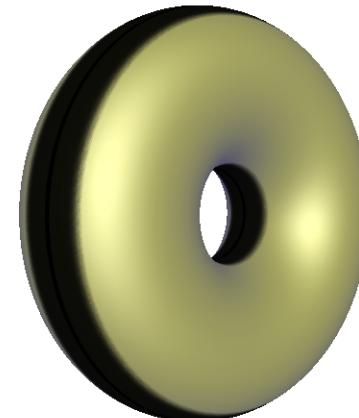
Curvature-Based Contour Threshold (1)



Threshold dependent on curvature in view direction

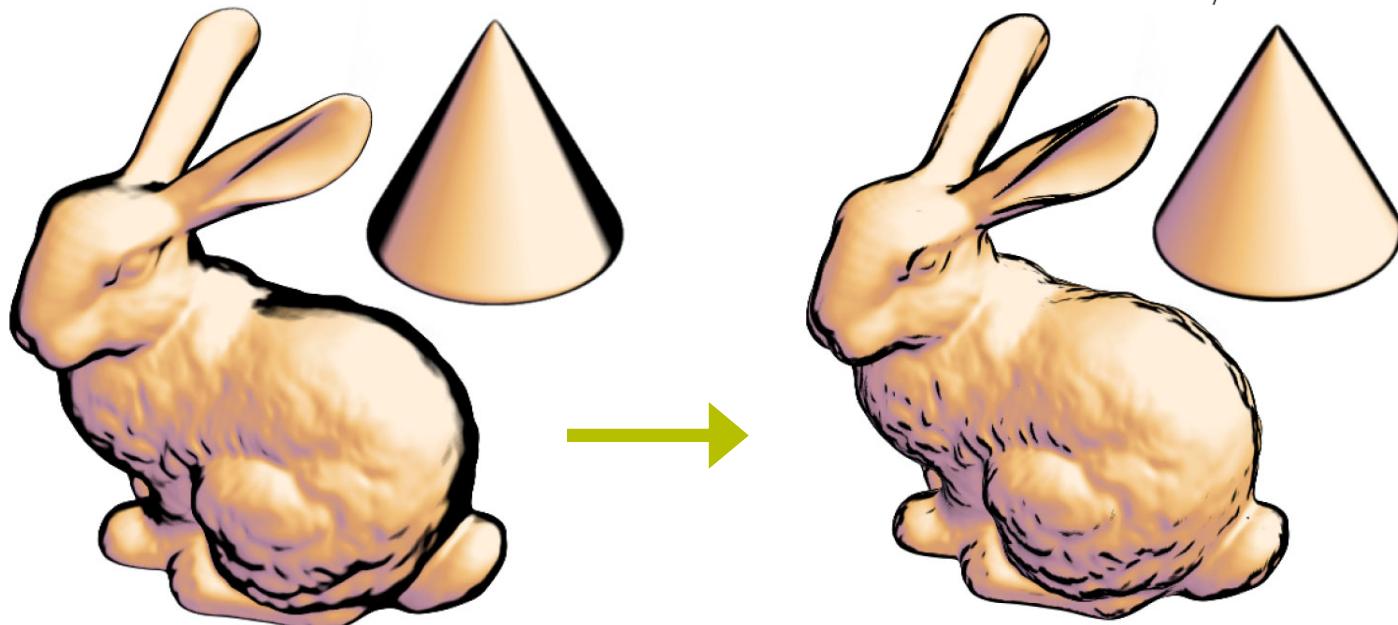
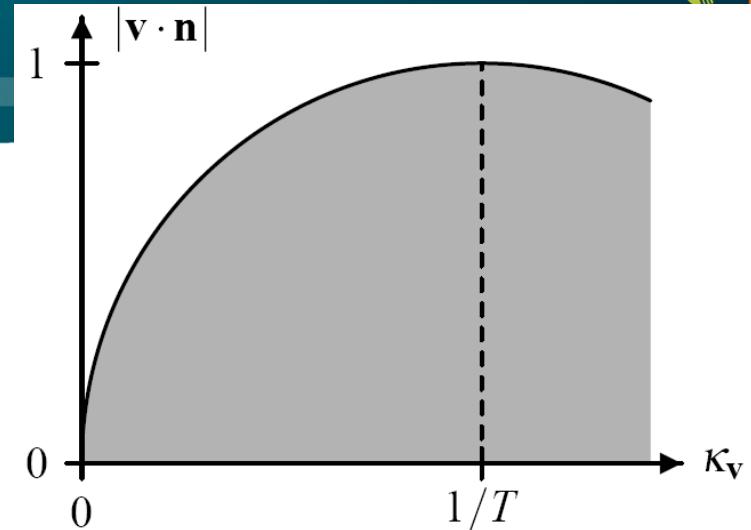


Thickness constant!



Curvature-Based Contour Threshold (2)

Higher curvature in view direction needs higher threshold



courtesy of Gordon Kindlmann

Example





Deferred Isosurface Shading

- Shading is expensive
- Compute surface intersection image from volume
- Compute derivatives and shading in image space



intersection image



curvature color coding



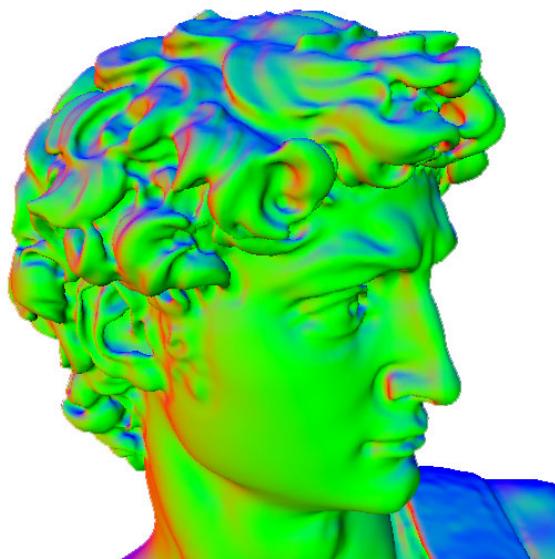
ridges and valleys

Implicit Curvature via Convolution



Computed from first and second derivatives

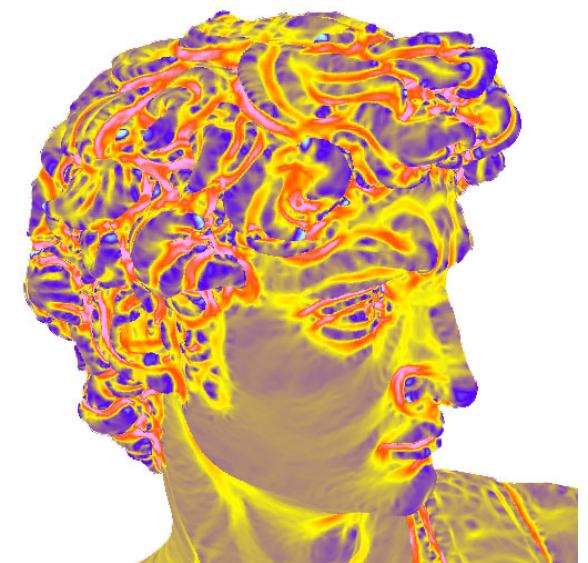
- Can use fast texture-based tri-cubic filters in shader
- Can use **deferred** computation and shading



first derivative



maximum curvature

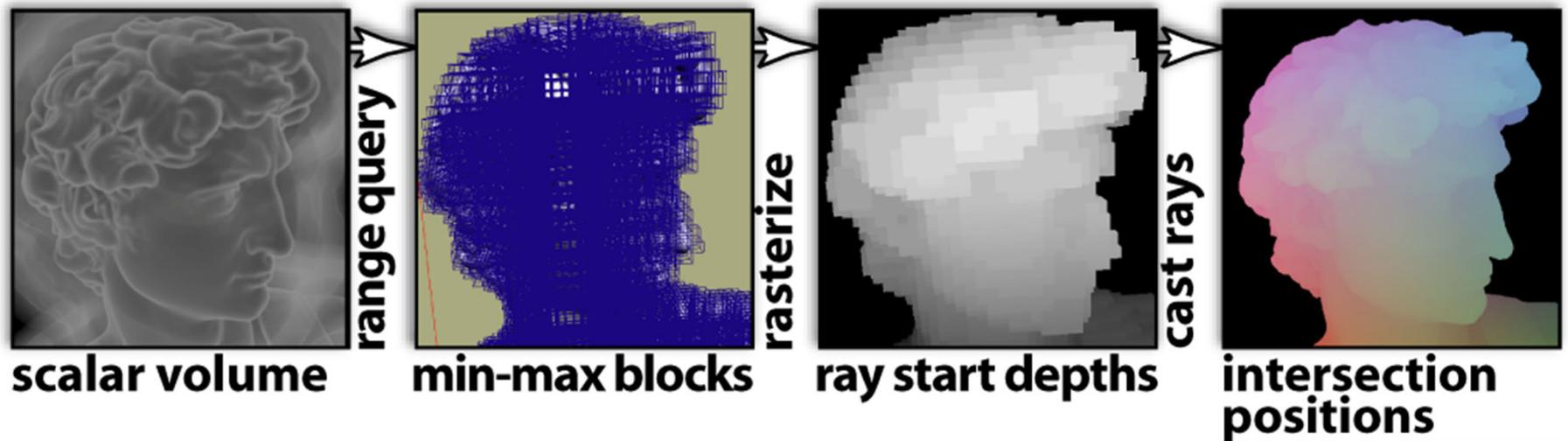


minimum curvature

Pipeline Stage #1: Ray-Casting



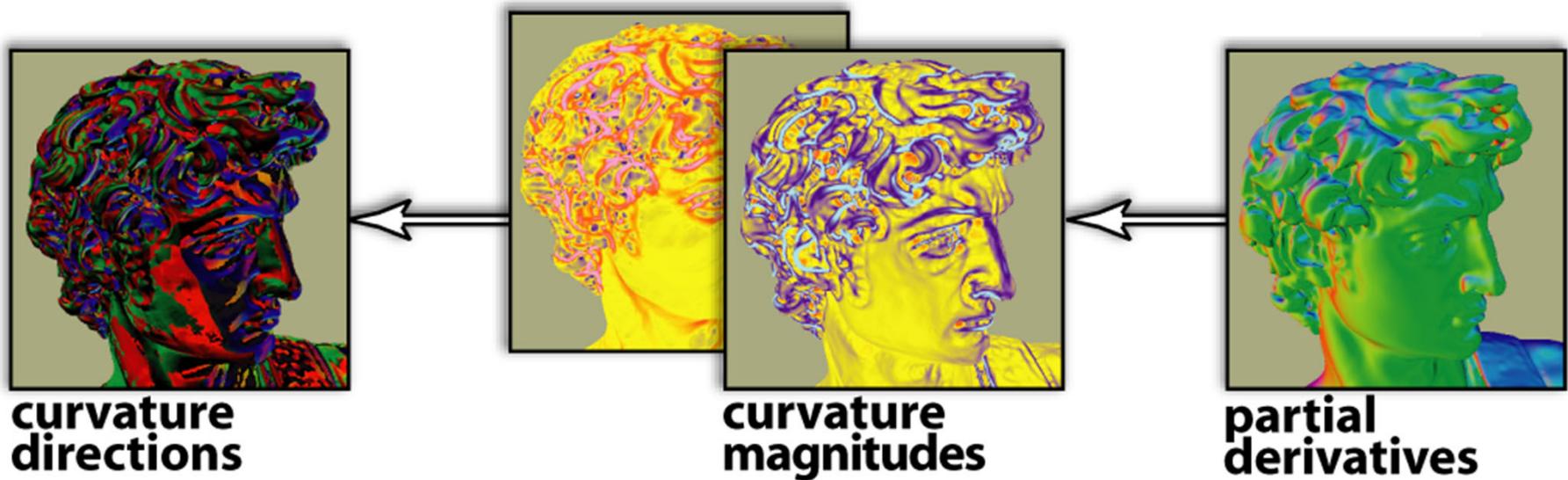
- Rasterize faces of active min-max blocks
- Cast into the volume; stop when isosurface hit
- Refine isosurface hit positions (root search)



Pipeline Stage #2: Differential Properties



- Basis for visualization of surface shape
- First and second derivatives (gradient, Hessian)
- From these: curvature information, ...





Pipeline Stage #3: Shading

Build on previous images

- Position in object space
- Gradient
- Principal curvature magnitudes and directions



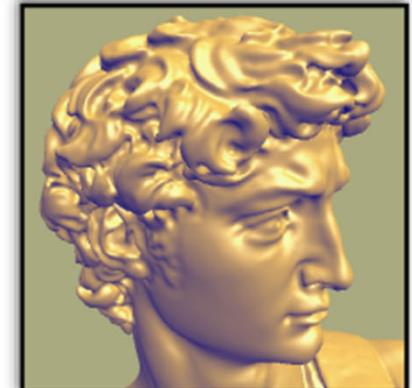
curvature flow



ridges+valleys



curvature mapping

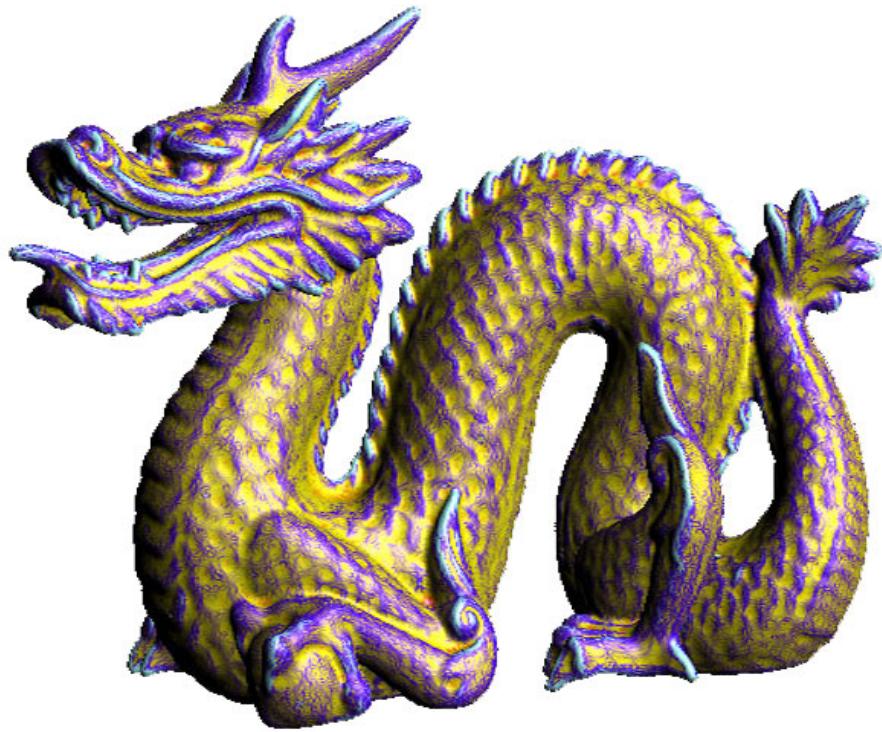


tone shading

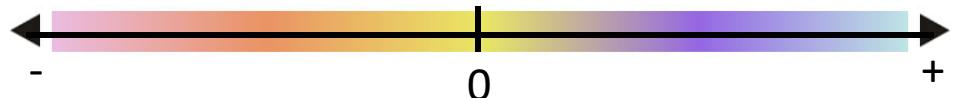
Color Coding Scalar Curvature Measures



- 1D color lookup table



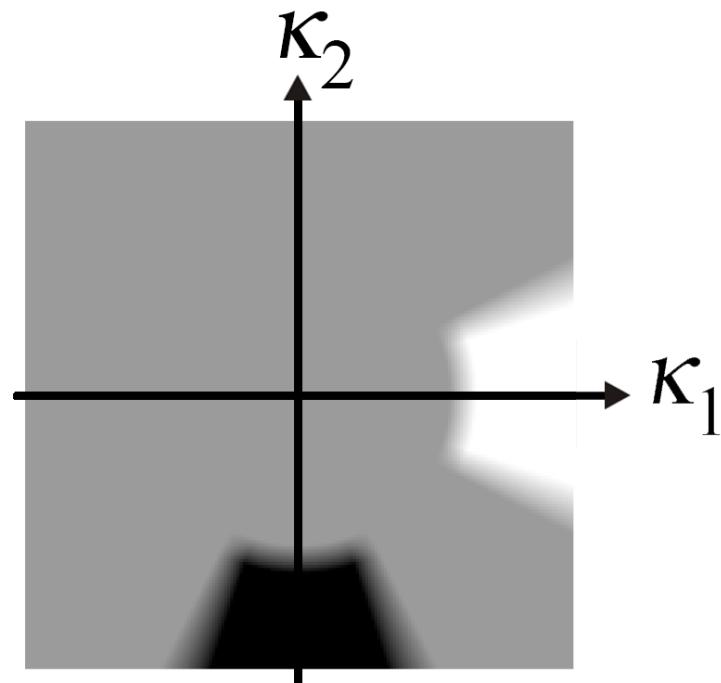
maximum principal curvature



2D Curvature Transfer Functions



- 2D lookup table in domain of principal curvatures



ridges and valleys, plus contours:





Visualizing Curvature Directions (1)

- Use 3D vector field visualization on curved surfaces
[van Wijk, 2003], [Laramee et al., 2003]
- Project 3D vectors to screen space
- Advect dense noise textures in screen space



Visualizing Curvature Directions (2)



Thank you.

Thanks for material

- Helwig Hauser
- Eduard Gröller
- Daniel Weiskopf
- Torsten Möller
- Ronny Peikert
- Philipp Muigg
- Christof Rezk-Salama