CP SC 8810
Data Visualization

Joshua Levine
levinej@clemson.edu
Agenda

• Design Critiques - one more next week

• Lab05 - incoming (tonight/tomorrow)
Refresher from Lec20
• 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?
Pixel Compositing Schemes

![Graph showing max intensity, accumulate, average, and first on the intensity axis against depth.](Image)

**Exact Isosurface**
Pixel Compositing Schemes

Similar to X-rays

Synthetic Reprojection
maximum intensity projection (MIP)

Pixel Compositing Schemes

- max intensity
- accumulate
- average
- first

Used in PET and Magnetic Resonance Angiograms
Pixel Compositing Schemes

- max intensity
- accumulate
- average
- first

Color to distinguish structures, opacity to show inside.
Computational Strategies

• Image Order:
  • Ray casting (many options)

• Object Order:
  • Splatting, texture mapping

• Combinations:
  • Shear-warp
What must be integrated?

physically correct: emission and absorption of light

\[ I(s) = I(s_0) e^{-\tau(s_0, s)} \]
What must be integrated?

physically correct: emission and absorption of light

\[ I(s_0) \]

\[ I(s) = I(s_0) e^{-\tau(s_0,s)} \]
What must be integrated?

physically correct: emission and absorption of light

\[ I(s) = I(s_0) e^{-\tau(s_0, s)} + q(\tilde{s}) e^{-\tau(\tilde{s}, s)} \]
What must be integrated?

**physically correct: emission and absorption of light**

\[ I(s) = I(s_0) e^{-\tau(s_0,s)} + q(\tilde{s}) e^{-\tau(\tilde{s},s)} \]
What must be integrated?

physically correct: emission and absorption of light

\[ I(s) = I(s_0) e^{-\tau(s_0, s)} + \int_{s_0}^{s} q(\tilde{s}) e^{-\tau(\tilde{s}, s)} d\tilde{s} \]
Discrete Solution

Resample the scalar field at discrete locations along the viewing ray:

\[ I(s_i) \quad I(s_{i+1}) \]

Ray

\[ s_0 \quad s_i \quad s_{i+1} \]

\[ I(s_0) \quad q(s_i), A(s_i) \quad q(s_{i+1}), A(s_{i+1}) \]

Back-to-front Compositing with

\[ \alpha = A(s_i) \]

\[ I(s_{i+1}) = \alpha \cdot q(s_{i+1}) + (1 - \alpha)I(s_i) \]
Texture-Based Volume Rendering

- Texturing (trilinear interpolation)
- Compositing (blending)

2D textures
axis-aligned

3D textures
view-aligned
A Bit of a Review on Texturing
How does a texture work?

$(s_0, t_0)$

$(s_2, t_2)$

$(s_1, t_1)$
How does a texture work?
How does a texture work?
How does a texture work?

For each fragment: 
interpolate the texture coordinates (barycentric)
How does a texture work?

For each fragment:
- interpolate the texture coordinates (barycentric)

Texture-Lookup:
- interpolate the texture color (bilinear)
2D Textures

- Draw the volume as a stack of 2D textures
- Bilinear Interpolation in Hardware
- Decomposition into axis-aligned slices

- 3 copies of the data set in memory
- Choose stack that is most perpendicular to view direction
3D Textures

\((s_0, t_0, r_0)\) 

\((s_2, t_2, r_2)\) 

\((s_1, t_1, r_1)\)
3D Textures

\[(s_0, t_0, r_0)\]

\[(s_1, t_1, r_1)\]

\[(s_2, t_2, r_2)\]

\[(s, t, r)\]
Don’t be confused: 3D textures are not volumetric rendering primitives!
Only planar polygons are supported as rendering primitives.
Transfer Functions
Classification

How do I obtain the emission values $q(s)$ and Absorption values $A(s)$?
Classification

How do I obtain the emission values $q(s)$ and Absorption values $A(s)$?
Classification

How do I obtain the emission values $q(s)$ and absorption values $A(s)$?
Classification

How do I obtain the emission values $q(s)$ and Absorption values $A(s)$?

- Scalar value $s$
- $T(s)$
- Emission RGB
- Absorption $A$
Classification

How do I obtain the emission values $q(s)$ and Absorption values $A(s)$?
Classification

How do I obtain the emission values $q(s)$ and Absorption values $A(s)$?
Transfer functions make volume data visible by mapping data values to optical properties.
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
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
Transfer Functions (TFs)

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

RGB

Shading, Compositing…

Human Tooth CT
Transfer Functions (TFs)

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

Shading, Compositing…

Human Tooth CT
Basic Transfer Functions:

- Space
- Vol Domain
- Data Value
- TF Domain
- Vol Range/TF Domain
- Color And Opacity
- Range
What Can Be Controlled by the Transfer Function?

- Optical Properties: Anything that can be composited with a standard graphics operator ("over")
  - Opacity: "opacity functions"
  - Color: Can help distinguish features
  - Phong parameters (ka, kd, ks)
  - Index of refraction
Setting Transfer Function: Hard
Volumes as Consisting of Materials

Grey-Level Histogram

Data value

Num voxels

Material 1
Material 2
Material 3
Transfer Function

» RGB components
» Opacity
» Histogram helps in designing transfer function
Transfer Function
Transfer Function
Transfer Function

Different colors, same opacity
Finding edges: easy

“Where’s the edge?”

$v = f(x)$

“here’s the edge”
"Where’s the edge?"

\[ v = f(x) \]

"here’s the edge"

Result: edge pixels
Finding edges: easy

"Where’s the edge?"

\[ v = f(x) \]

"here’s the edge"

Result: edge pixels
Transfer function Unintuitive

\[ v = f(x) \]
TFs as feature detection

\[ v = f(x) \]

“here’s the edge!”

Domain of the transfer function does not include position

Data Value

Domain

TF
What Makes Designing TF’s Challenging?

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

1. Trial and Error (manual)

2. **Image-Centric Approach**

3. Data-Centric Approach
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)
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; 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
The Contour Spectrum
(colored lines correspond to different isosurface metrics)

The contour spectrum allows the development of an adaptive ability to separate interesting isovalues from the others.
The Contour Spectrum
(colored lines correspond to different isosurface metrics)

The contour spectrum allows the development of an adaptive ability to separate *interesting* isovalues from the others.
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.
Some Background: Gradients
Gradient

\[ f(x) \]
\[ \nabla f \]
\( \nabla f = (dx, dy, dz) \)
\[ \nabla f = (dx, dy, dz) \]
\[ = \frac{(f(1,0,0) - f(-1,0,0))}{2}, \]

**Gradient**

\[ f(x) \]
\[ \nabla f \]
$\nabla f = (dx, dy, dz)$

$= \left( \frac{f(1,0,0) - f(-1,0,0)}{2}, \frac{f(0,1,0) - f(0,-1,0)}{2}, \frac{f(0,1,0) - f(0,-1,0)}{2} \right)$
\[ \nabla f = (dx, dy, dz) \]
\[ = \left( \frac{f(1,0,0) - f(-1,0,0)}{2}, \frac{f(0,1,0) - f(0,-1,0)}{2}, \frac{f(0,0,1) - f(0,0,-1)}{2} \right) \]
\[ \nabla f = (dx, dy, dz) \]
\[ = \left( \frac{f(1,0,0) - f(-1,0,0)}{2}, \right. \]
\[ \left. \frac{f(0,1,0) - f(0,-1,0)}{2}, \right. \]
\[ \left. \frac{f(0,0,1) - f(0,0,-1)}{2} \right) \]

- Approximates "surface normal" (of isosurface)
Derivative relationships

Edges at maximum of 1\textsuperscript{st} derivative or zero-crossing of 2\textsuperscript{nd}
Project histogram volume to 2D scatterplots

- Visual summary
- Interpreted for TF guidance
- No reliance on boundary model at this stage
Higher-Dimensional Transfer Functions
Basic Transfer Functions:

Space → Vol Domain → Vol Range/TF Domain → TF → Range

Vol

Value + Grad

Mag

Color And Opacity
1D TFs: limitation

1D transfer functions cannot accurately capture all material boundaries.
1D TFs: limitation

1D transfer functions cannot accurately capture all material boundaries.
1D $\rightarrow$ 2D Transfer Function

$\text{RGB}(f)$

$\alpha(f)$

Generalize...
1D $\rightarrow$ 2D Transfer Function

$\text{RGB}(f)$

$\alpha(f)$

Generalize...
1D $\rightarrow$ 2D Transfer Function

\[
\begin{align*}
\text{RGB}(f) \\
\alpha(f)
\end{align*}
\]

Generalize...
1D $\rightarrow$ 2D Transfer Function

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

Generalize...
1D → 2D Transfer Function

\[ \text{RGB}(f) \]

\[ \alpha(f) \]
1D $\rightarrow$ 2D Transfer Function

$\text{RGB}(f)$

$\alpha(f)$

Generalize...
2D Transfer Function

\[ \text{RGB}(f) \]

\[ \alpha(f) \]

Generalize...

\[ \alpha, |\nabla f| \]

\[ \text{RGB}(f, |\nabla f|) \]

\[ \alpha(f, |\nabla f|) \]
2D Transfer Function

\[ \text{RGB}(f, \| \nabla f \|) \]

\[ \alpha(f, \| \nabla f \|) \]

\[ \{ \text{Modify} \ldots \]
2D Transfer Function

\[ \text{RGB}(f, |\nabla f|) \]

\[ \alpha(f, |\nabla f|) \]

Modify…
2D Transfer Function

\[ \text{RGB}(f, \| \nabla f \|) \]

\[ \alpha(f, \| \nabla f \|) \}

\( f \)

\( \alpha \)

\( \| \nabla f \| \)

Modify…
2D Transfer Function

\[ \text{RGB}(f, \| \nabla f \| ) \]

\[ \alpha(f, \| \nabla f \| ) \}

Modify…
2D Transfer Function

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

\[ \alpha(f, |\nabla f|) \]

Modify…
2D Transfer Function

\[
\begin{align*}
\text{RGB} & \left( f, \left| \nabla f \right| \right) \\
\alpha & \left( f, \left| \nabla f \right| \right)
\end{align*}
\]

Modify…
2D Transfer Function

\[ \text{RGB}(f, |\nabla f|) \]

\[ \alpha(f, |\nabla f|) \}

Modify…
2D Transfer Function

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

Modify…
2D Transfer Function

\[ \text{RGB}(f, \nabla f) \]

\[ \alpha(f, \nabla f) \}

Modify…

2D transfer functions give greater flexibility in boundary visualization

Display of Surfaces from Volume Data, Levoy 1988
2D Transfer Function

\[
\text{RGB}(f, \lVert \nabla f \rVert) \begin{cases} \alpha(f, \lVert \nabla f \rVert) \end{cases}
\]

Trying to reintroduce dentin / background boundary …

Modify…
2D $\rightarrow$ 3D Transfer Function

$\alpha \rightarrow f$ [transfer function]

$\nabla f$ [gradient]

$\nabla^2 f$ [Hessian]

$D^2 \nabla f \cdot f$ [second directional derivative, measured with Hessian]

RGB $\alpha(f, |\nabla f|, D^2 \nabla f \cdot f)$
3D Transfer Function

\[ \alpha \left( f, |\nabla f|, D^2_{\nabla f} f \right) \]

RGB

Modify…
3D Transfer Function

\[ \alpha( f, |\nabla f|, D^2_{\nabla f} f ) \]

Modify…
3D Transfer Function

\[ \alpha(f, |\nabla f|, D^2_{\nabla f} f) \]

Modify...
3D Transfer Function

\[ \alpha(f, |\nabla f|, D^2 f) \]

Modify...
3D Transfer Function

\[
\alpha(f, |\nabla f|, D^2_{\nabla f} f) + RGB 0 \quad \text{Modify…}
\]
3D Transfer Function

\[ \alpha \left( f, |\nabla f|, \mathbf{D}^2_{\nabla f} f \right) \]

\[ \text{RGB} \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]

\[ \alpha \]

\[ |\nabla f| \]

\[ f \]
3D Transfer Function

enamel / background
dentin / background
dentin / enamel
dentin / pulp

1D: not possible
2D: specificity not as good
Original TF  Boundaries (gradient)
Multi-Dimensional TFs

• Strengths:
  • Better flexibility, specificity
  • Higher quality visualizations

• Weaknesses:
  • Even harder to specify
  • Unintuitive relationship with boundaries
  • Greater demands on user interface
Other TF Methods
Curvature

“Curvature-Based Transfer Functions for Direct Volume Rendering”, Hladuvka, König, Gröller: SCCG ’00

- Uses 2D space of $K_1$ and $K_2$: 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

Principal curvature magnitudes
Principal curvature directions

Scientific Computing and Imaging Institute, University of Utah
Curvature measures

\[
\begin{align*}
(K_1 + K_2)/2 & \quad \text{(mean)} \\
K_1 K_2 & \quad \text{(Gaussian)} \\
\sqrt{K_1^2 + K_2^2} & \quad \text{(total)}
\end{align*}
\]
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\textsuperscript{st}, 2\textsuperscript{nd} derivative)
• “Paint” into the transfer function domain
Lec 22
Required Reading
8.1 - 8.3 ONLY!!

8.1 The Big Picture

For datasets with spatial semantics, the usual choice for *arrange* is to use the given spatial information to guide the layout. In this case, the choices of *express*, *separate*, *order*, and *align* do not apply because the position channel is not available for directly encoding attributes. The two main spatial data types are geometry, where shape information is directly conveyed by spatial elements that do not necessarily have associated attributes, and spatial fields, where attributes are associated with each cell in the field. (See Figure 8.1.) For scalar fields with one attribute at each field cell, the two main visual encoding idiom families are isocontours and direct volume rendering. For both vector and tensor fields, with multiple attributes at each cell, there are four families of encoding idioms: flow glyphs that show local information, geometric approaches that compute derived geometry from a sparse set of seed points, texture approaches that use a dense set of seeds, and feature approaches where data is derived with global computations using information from the entire spatial field.