Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering
Outline

1. Background
2. Related Work
3. Opacity Determination
   (1) Ideal Boundary Characterization
   (2) The Histogram Volume
   (3) Opacity Function Generation
4. Conclusion
Background

What’s the specification of the transfer function?

It maps the original volume data to renderable optical properties, such as opacity, color or emittance.

Why is it difficult to find a transfer function?

There is no way to know how individually sampled values are representative of the whole feature, which is of importance to define an opacity function (subset of transfer function).
Background

A significant assumption in this paper is that features of interest in the scalar volume are the boundary between areas of relatively homogeneous material.

Isosurface Rendering & Direct Volume Rendering

To the extent that an object’s surface is associated with a range of values.
Related Work

Two methods have been proposed for exploring the transfer function.

• The user pick desirable transfer function from an automatically generated population until the iterative process of image selection and transfer function inter-combine converges.

• The range of the transfer function is reduced into a “design gallery” by using an image difference metric.

But both processes are entirely driven by analysis of rendered images, and not of the dataset itself.
Ideal Boundary Characterization

**Boundary Model**

The method in this paper is developed based on the model with an ideal boundary having a sharp, discontinues change in the physical property. But the measured boundary is usually blurred because of band-limited caused by Gaussian frequency response.

![Figure 2: Boundaries are step functions blurred by a Gaussian.](image)
Ideal Boundary Characterization

**Directional Derivatives along the Gradient**

The gradient vector is employed to find the direction which passes perpendicularly through the boundary.

![Diagram](image)

*Figure 3: $\nabla f$ is always normal to $f$’s isosurfaces.*
Ideal Boundary Characterization

Directional Derivatives along the Gradient (cont…)

The scalar field $f$ and its directional derivatives along the gradient $v$ is studied as one cuts through the object. An exact location for the boundary is defined with either the maximum in $f''$, or the zero-crossing in $f'''$.

Figure 4: Measuring $f$, $f'$, and $f''$ across boundary.
Ideal Boundary Characterization

**Relationship Between** $f$, $f'$ and $f''$

Three dimension graph is applied to illustrate the relationship between $x$, $f$, $f'$ and $f''$. With a tool to detect these curves and their position, one could generate an opacity function.
The Histogram Volume

Histogram Volume Structure

There is one axis for each of the tree quantities $f, f'$ and $f''$, and each axis is divided into some number of (one-dimensional) bins, causing the interior volume to be divided into a three dimensional array of bins.

1. each bin in the histogram volume the combination of a small range of values in each of three variables $f, f'$ and $f''$.

2. The value stored in each bin signifies the number of voxels in the origin volume within that some combination of ranges of these three variables.
The approach in this paper is to measure $f$ and its directional derivatives exactly once per voxel, at the origin sample points of the dataset.
The Histogram Volume

**Implementation**

1. Bin’s size. $80^3 \sim 256^3$

2. Range of value along each axis. Set with an educated guess.

3. The method of measuring the first and second directional derivatives.

$$\mathbf{D}_{\nabla^2 f} = \nabla f \cdot \nabla f = \nabla f \cdot \frac{\nabla f}{||\nabla f||} = ||\nabla f||.$$ \hspace{1cm} \text{(2)}

$$\mathbf{D}_{\nabla^2 f} = \mathbf{D}_{\nabla^2 f}(||\nabla f||) = \nabla(||\nabla f||) \cdot \nabla f$$

$$= \frac{1}{||\nabla f||} \nabla(||\nabla f||) \cdot \nabla f$$ \hspace{1cm} \text{(3)}

$$\mathbf{D}_f f = \frac{1}{||\nabla f||^2} (\nabla f)^T \mathbf{H}_f \nabla f$$ \hspace{1cm} \text{(4)}

where $\mathbf{H}_f$ is the Hessian of $f$, a $3 \times 3$ matrix of second partial derivatives of $f$. 

$$\mathbf{D}_f f \approx \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$ \hspace{1cm} \text{(5)}
The general algorithm for creating the histogram is listed as follows:

1. Initialize the histogram volume to all 0’s.
2. Make one pass through the volume looking for the highest values of \( f' \) and \( f'' \), and the lowest value of \( f''' \); assume 0 for the lowest value of \( f' \). Set ranges on the histogram volume axes accordingly.
3. On a second pass through the volume,
   3a. Measure \( f, f', \) and \( f'' \) at each voxel,
   3b. Determine which bin in the histogram volume corresponds to the measured combination of \( f, f', \) and \( f'' \), and
   3c. Increment the bin’s value.
The Histogram Volume

Histogram Volume Inspection

Figure 9: Dataset Slice, $f'$ versus $f$, and $f''$ versus $f$. 
The Histogram Volume

Histogram Volume Inspection (cont.)

Figure 10: Dataset Slice, $f'$ versus $f$, and $f''$ versus $f$. 

(a) Turbine Blade

(b) Head

(c) Engine Block
Opacity Function Generation

Mathematical Boundary Analysis

\[ v = f(x) = v_{\min} + (v_{\max} - v_{\min}) \frac{1 + \text{erf}(\frac{x}{\sigma \sqrt{2}})}{2} \]  

\[ f'(x) = \frac{v_{\max} - v_{\min}}{\sigma \sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \]  

\[ f''(x) = -\frac{x(v_{\max} - v_{\min})}{\sigma^3 \sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \]

\[ \frac{f'(0)}{f''(-\sigma)} = \sigma \sqrt{\varepsilon} \]  

\[ \frac{f''(x)}{f'(x)} = -\frac{x}{\sigma^2} \]  

Figure 4: Measuring \( f, f', \) and \( f'' \) across boundary.
Opacity Function Generation

Opacity functions of data value (or adding gradient magnitude)

Two important functions are defined as follows: $g(v)$ is the average first directional derivative of $f$ over all the positions $x$ at which $f(x) = v$, and $h(v)$ is likewise the average second directional derivative at value $v$.

\[
p(v) = \frac{-\sigma^2 h(v)}{g(v)} \approx \frac{-\sigma^2 f''(f^{-1}(v))}{f'(f^{-1}(v))} = x
\]

\[
p(v) = \frac{-\sigma^2 h(v)}{\max(g(v) - g_{\text{threshold}}, 0)} \quad (12)
\]

\[
p(v, g) = \frac{-\sigma^2 h(v, g)}{\max(g - g_{\text{threshold}}, 0)} \quad (14)
\]

\[
\alpha(v) = b(p(v)) \quad (13)
\]
Opacity Function Generation

Opacity functions of data value (cont..)

Figure 11: Relationship between \( b(x) \), \( \alpha(x) \), and the rendered result.
Results
Results
Conclusions

- The histogram volume structure presented in this paper captures information about the boundaries.

- It makes sense to apply computer vision object recognition techniques to histogram volume.

- Is it possible to determine the opacity function in the frequency domain?

- Is it possible to use histogram volume to detect boundary among different materials in volume segmentation?
Any questions?

Thank you!

Department of Computer Science, University of North Carolina at Charlotte
Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets
Outline

1. Background
2. Related Work
3. Multi-Dimensional Transfer Functions
4. Direct Manipulation Widgets
5. Hardware Considerations
6. Conclusion
Background

Why does it make sense to study multi-dimensional transfer function?
There are many features of interest in volume that are difficult to extract and visual with 1D transfer functions, such as a complex combination of boundaries between multiple materials in CT or MRI.

Why is it difficult to find a multi-dimensional transfer function?
It is obvious that it is difficult to determine an appropriate transfer function. Also it is hard to extend from one-dimension to high dimension.
Related Work

**Transfer Functions**

1. 1D transfer function: refer to the related work in the previous paper.

2. 2D transfer function: Levoy introduced two styles of transfer function. One was developed for the display of interfaces between materials, the other for the display of isovalue contours in more smoothly varying data. This taxonomy also contains previous paper.

3. Multi-transfer function. Extend the levoy’s work, combine various types of second derivatives and design colormap, etc…
Related Work (cont..)

**Direct Manipulation Widgets**

They are geometric objects rendered with a visualization and are designed to provide the user with a 3D interface.

**Hardware Volume Rendering**

Many volume rendering techniques based on graphics hardware utilize texture memory to store a 3D dataset.

1. **2D-texture based techniques**: they exploit hardware bilinear interpolation.

2. **3D-texture based techniques**: they typically sample view-aligned slices through the volume, leveraging hardware tri-linear interpolation.

Other hardware includes Cube-4 architecture and the subsequent VolumePro PCI graphics board.
Multi-Dimensional Transfer Functions

The reason of difficulty to accomplish a good transfer function:

• The transfer function has an enormous number of degrees of freedom.
• The usual interfaces for setting transfer functions are not constrained or guided by the dataset in question.
• Transfer functions are inherently non-spatial.
Multi-Dimensional Transfer Functions

This paper presented a three-dimension transfer functions for scalar data (based on data value, gradient magnitude, and a second directional derivative).

Figure 1: A 1D transfer function (a) is emulated by assigning opacity regardless of gradient magnitude (vertical axis in lower frame). A 2D transfer function (b) giving opacity to only low gradient magnitudes reveals internal structure.

Figure 2: In (a), the 2D transfer function is intended to render all material interfaces except the enamel-background boundary at the top of the tooth. However, by using a 3D transfer function (b), with lower opacity for non-zero second derivatives, the previously hidden dentin-enamel boundary is revealed.
Direct Manipulation Widgets

A principle of the direct manipulation widgets presented in this paper is to link interaction in one domain with feedback in another. The method for the user to interact with this system is following:

The user uses a clipping plane to detect a region of interest, then track the changing around this area, and finally capture the transfer function.

The other interaction scenario begins with a pre-determined transfer function. It is then updated by investigating and exploring the dataset as described above.
Direct Manipulation Widgets (cont..)

- **Dual-Domain Interaction**

The transfer function is set by direct interaction in the spatial domain, with observation of the transfer function domain.

![Diagram](image)

Figure 3: Dual-Domain Interaction
Direct Manipulation Widgets (cont..)

- **Data Probe Widget**
  
  It is responsible for reporting its tip’s position in volume space and its slider sub-widget’s value with a pencil-like shape.

- **Clipping Plane Widget**
  
  It is a basic frame type widget with the following attributes:
  
  1. It reports its orientation and position to the volume widget.
  2. It reports the spatial position of a mouse click on its clipping plane.
Direct Manipulation Widgets (cont.)

**Transfer Function Widget**

Its main role is to present a graphical representation of the transfer function domain. The backbone of the transfer function widget is a basic frame widget.

**Classification Widget**

The opacity and color contributions from each classification widget sum together to form the transfer function. There are two types of classification widget: triangular and rectangular.
Direct Manipulation Widgets (cont..)

- **Shading Widget**

  It is a collection of spheres which can be rendered in the scene to indicate and control the light direction and color.

- **Color Picker Widget**
Hardware Considerations

Figure 6: Octane2 Volume Rendering pipeline. Updating the shade volume (right) happens after the volume has been rotated. Once updated, the volume would then be re-rendered.

Figure 7: GeForce3 Volume Rendering pipeline. Four-way multi-texture is used. The textures are: VGH, VG Dependant Texture, H Dependant texture, and the Normal texture (for shading). The central box indicates the register combiner stage. The Blend VG&H Color stage is not usually executed since we rarely vary color along the second derivative axis. The Multiply VG&H Alpha stage, however, is required since we must compose our transfer function separably as a 2D×1D transfer function.
Hardware Considerations (cont..)

Pixel Texture

It uses color fragments to generate texture coordinates, and replace those color fragments with the corresponding entries from a texture.

Classification

The resolution of the second derivative axis can be reduced by choosing a limited number of control points for this axis and represent them as sheets in the pixel texture.
Hardware Considerations (cont..)

- Shading

- Hardware Implementation

The left side of Figure 6 illustrates the rendering process. The slices from the VGH data volume are first rendered (1) and then pixel textured (2). The “Shade” slice is rendered and modulated with the classified slice (3), then blended into the frame buffer (4). When the volume is rotated, lighting must be updated (shown on the right side of Figure 6). For interactive efficiency, we only update the shade volume once a rotation has been completed. A new quantized normal pixel texture (for shading) is generated and each slice of the quantized normal volume is rendered orthographically in the scratch buffer (1) and then pixel textured (2). This slice is then copied from the scratch buffer to the corresponding slice in the shade volume (3). The volume is then re-rendered with the updated shade volume. Updating the shade volume in hardware requires that the quantized normal slices are always smaller than scratch buffer’s dimensions.
Results
Results
Conclusion

- Sample rates (?)
- Use a
- Dual-

(a) Clipping plane with probe  (b) Showing frontal sinuses

Figure 9: A clipping plane cuts through the region above the eyes. Probing in the area produces a re-projected voxel with the characteristic arc shape indicating the presence of a surface (a). A similarly placed triangular classification widget reveals the shape of the sinus (b).
Future work

- Consume huge memory
- Render multi-variate volume data
- Is it possible to choose multi-dimensional transfer function automatically? For example, based on feedback, then re-generate the function using machine learning, etc…
Any questions?

Thank you!