3D Scientific Visualization

Oregon State University

Prof. Mike Bailey
Computer Science

mjb@cs.oregonstate.edu
http://eecs.oregonstate.edu/~mjb

Slides for this presentation are at:
http://eecs.oregonstate.edu/~mjb/geovis.pdf

Why Visualize?

Visual!
Part I:

Using the GPU for Scientific Visualization
A Way-simplified View of Graphics Hardware

CPU → Vertices
Bus

Vertex Processor → Transformed Vertices

Assemble Geometry

Rasterizer → Interpolated Values

Fragment Processor → Pixels

“In the GPU”

Don’t Send Colors to the GPU, Send the Raw Data

Use the GPU turn the data into graphics on-the-fly

Visualization by Chris Janik
Terrain Explorer Program

Use the GPU to handle relations between datasets

Dome Projection for Immersive Visualization

Use the GPU to perform nonlinear vertex transformations
Bump-mapping to Create Apparent Surface Detail

Rock A Dropped  
Rock B Dropped  
Both Rocks Dropped

Use the GPU to create the appearance of height without geometrically creating height

Terrain Height Bump-Mapping

Visualization by Nick Gebbie
Use the GPU to work with aerial and satellite images that were originally larger than what will fit in graphics card memory.
Image Manipulation for Visualization: Un-Masking

I_{\text{out}} = (1-t)I_{\text{don't want}} + tI_{\text{in}}

Blend of what don't want and what have

Blend of what have and what want more of

More of what I do want

What I started with

What I don't want

0.0 1.0 2.0

t

Use the GPU to interactively manipulate images

Sharpening

Blur Convolution:

B = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}

I_{\text{don't want}} = I_{\text{blur}}
**Sharpening**

**T = 0.**

**T = 1.**

**T = 2.**

---

**Edge Detection**

**Horizontal and Vertical Sobel Convolutions:**

\[
H = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1
\end{bmatrix} \quad V = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix}
\]

\[
S = \sqrt{H^2 + V^2} \quad \Theta = \text{atan2}(V, H)
\]
Edge Detection

T = 0.  T = 0.5  T = 1.

Non-photorealistic Rendering

Use the GPU to enhance scientific and engineering illustration
Non-photorealistic Rendering

Use the GPU to abuse your friends

Image Manipulation Example – Identify the Deserts

Use the GPU to examine image component characteristics to identify image regions
Visualization: Point Clouds

Can change:
• Color
• Alpha
• Points size

Use the GPU to interactively change the appearance of 3D data

Visualization: Volume Rendering

Can change:
• Color
• Alpha
Volume Rendering in the Medical World

Visualizing Hillslope Drying with Temporal Volume Rendering
Visualizing Hillslope Drying with Temporal Volume Rendering

Visualizing Hillslope Drying with Temporal Volume Rendering

Visualization: Extruding Shapes Along Flow Lines

Use the GPU to show flow information

Add moving "humps" to create a peristaltic effect
Visualization: 2D Line Integral Convolution

At each fragment:
1. Find the flow field velocity vector there
2. Follow that vector in both directions
3. Blend in the colors at the other fragments along that vector

Visualization: 3D Line Integral Convolution
Part II:

Some Good Rules of Thumb When Using Color for Scientific Visualization
What Makes a Good Contrast?

- Many people think simply adding color onto another color makes a good contrast
- In fact, a better measure is the Δ luminance
- Knowing this also helps if someone makes a grayscale photocopy of your color output

Color Alone Doesn’t Cut It!

I sure hope that my life does not depend on being able to read this quickly and accurately!
Luminance Contrast is Crucial!

I would prefer that my life depend on being able to read this quickly and accurately!

The Luminance Equation

\[ Y = 0.30\text{Red} + 0.59\text{Green} + 0.11\text{Blue} \]
### Luminance Table

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>G</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td>White</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.00</td>
</tr>
<tr>
<td>Red</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.30</td>
</tr>
<tr>
<td>Green</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.59</td>
</tr>
<tr>
<td>Blue</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.11</td>
</tr>
<tr>
<td>Cyan</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.70</td>
</tr>
<tr>
<td>Magenta</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.41</td>
</tr>
<tr>
<td>Orange</td>
<td>1.0</td>
<td>0.5</td>
<td>0.0</td>
<td>0.60</td>
</tr>
<tr>
<td>Yellow</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.89</td>
</tr>
</tbody>
</table>

### Contrast Table

<table>
<thead>
<tr>
<th></th>
<th>Black</th>
<th>White</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
<th>Cyan</th>
<th>Magenta</th>
<th>Orange</th>
<th>Yellow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>0.00</td>
<td>1.00</td>
<td>0.30</td>
<td>0.59</td>
<td>0.11</td>
<td>0.70</td>
<td>0.41</td>
<td>0.60</td>
<td>0.89</td>
</tr>
<tr>
<td>White</td>
<td>1.00</td>
<td>0.00</td>
<td>0.70</td>
<td>0.41</td>
<td>0.89</td>
<td>0.30</td>
<td>0.59</td>
<td>0.41</td>
<td>0.11</td>
</tr>
<tr>
<td>Red</td>
<td>0.30</td>
<td>0.70</td>
<td>0.00</td>
<td>0.29</td>
<td>0.19</td>
<td>0.40</td>
<td>0.11</td>
<td>0.30</td>
<td>0.59</td>
</tr>
<tr>
<td>Green</td>
<td>0.59</td>
<td>0.41</td>
<td>0.29</td>
<td>0.00</td>
<td>0.48</td>
<td>0.11</td>
<td>0.18</td>
<td>0.01</td>
<td>0.30</td>
</tr>
<tr>
<td>Blue</td>
<td>0.11</td>
<td>0.89</td>
<td>0.19</td>
<td>0.48</td>
<td>0.00</td>
<td>0.59</td>
<td>0.30</td>
<td>0.40</td>
<td>0.78</td>
</tr>
<tr>
<td>Cyan</td>
<td>0.70</td>
<td>0.30</td>
<td>0.40</td>
<td>0.11</td>
<td>0.59</td>
<td>0.00</td>
<td>0.29</td>
<td>0.11</td>
<td>0.19</td>
</tr>
<tr>
<td>Magenta</td>
<td>0.41</td>
<td>0.59</td>
<td>0.11</td>
<td>0.18</td>
<td>0.30</td>
<td>0.29</td>
<td>0.00</td>
<td>0.19</td>
<td>0.48</td>
</tr>
<tr>
<td>Orange</td>
<td>0.60</td>
<td>0.41</td>
<td>0.30</td>
<td>0.01</td>
<td>0.49</td>
<td>0.11</td>
<td>0.19</td>
<td>0.00</td>
<td>0.30</td>
</tr>
<tr>
<td>Yellow</td>
<td>0.89</td>
<td>0.11</td>
<td>0.59</td>
<td>0.30</td>
<td>0.78</td>
<td>0.19</td>
<td>0.48</td>
<td>0.30</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Look for a $\Delta$ luminance $\geq 0.40$
# Do Not Attempt to Fight Pre-Established Color Meanings

<table>
<thead>
<tr>
<th>Red:</th>
<th>Green:</th>
<th>Blue:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>On</td>
<td>Cool</td>
</tr>
<tr>
<td>On</td>
<td>Plants</td>
<td>Safe</td>
</tr>
<tr>
<td>Off</td>
<td>Carbon</td>
<td>Deep</td>
</tr>
<tr>
<td>Dangerous</td>
<td>Moving</td>
<td>Nitrogen</td>
</tr>
<tr>
<td>Hot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High stress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oxygen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shallow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Money loss</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Here’s What’s Important in a Color Scale:

Given any 2 colors, make it intuitively obvious which represents “higher” and which represents “lower”

Obvious:

Not obvious:

What was The Oregonian Thinking When They Chose This Color Scale?

Source:
The Oregonian,
January 11, 2006
Fortunately, They Got Better At It …

Source: The Oregonian, October 31, 2006

The Ability to Discriminate Colors Changes with the Surrounding Color
Afterimages

Afterimages
Beware of Mach Banding

Actual Intensity

Perceived Intensity

mjb – January 30, 2007
Beware of Mach Banding

Beware of Lots of Other Stuff
Beware of Lots of Other Stuff
Use a Black or White Line as the Boundary Between Colored Regions

Watch the Use of Saturated Blues for Fast-Moving Items or Fine Detail
Watch the Use of Saturated Reds and Blues Together

Reds and Blues are on opposite ends of the color spectrum. It is hard for your eyes to focus on both.

Do Not Display High Spatial Frequencies in Color
Understand the Limitations of going from Workstations to NTSC

- Use less saturated colors due to color gamut considerations
- Expect an effective resolution of (at best) ~640x480
- Some colors have better video resolution than others

NTSC Cycles-of-Encoding per Scanline

<table>
<thead>
<tr>
<th>What</th>
<th>Cycles/Scanline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intensity</td>
<td>267</td>
</tr>
<tr>
<td>Orange-Blue</td>
<td>96</td>
</tr>
<tr>
<td>Purple-Green</td>
<td>35</td>
</tr>
</tbody>
</table>
Conclusions

• Be careful with the use of color. If done poorly, it detracts from the effective display of information. Above all, do no harm.

• GPU programming is one of the most exciting things that has ever happened to CG

• It enables application developers to have very tight control over graphics effects without sacrificing display performance

• It was really made for game development, but it has significant applications in visualization, imaging, and scientific computing

• OSU is one of the few universities that has an organized course in GPU Programming – CS 519 – next offered in Spring Quarter 2007. The pre-requisite is having taken any of the other CS graphics classes.

For more details on the class, see:
http://eecs.oregonstate.edu/~mjb/cs519

3D Scientific Visualization

Thank You!

Mike Bailey

mjb@cs.oregonstate.edu
http://eecs.oregonstate.edu/~mjb

Slides for this presentation are at:
http://eecs.oregonstate.edu/~mjb/geovis.pdf