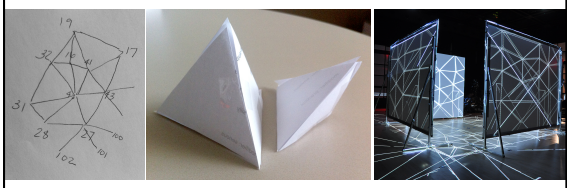
Visual Debugging



Today's Class

- Highlights from the HW7
- This Week's Readings
- Next Week's Readings
- Examples of Visualization for Debugging
- Info on Visual Debugging in VTK
- Your Debugging Challenges
- HW8: Final Project Progress Report

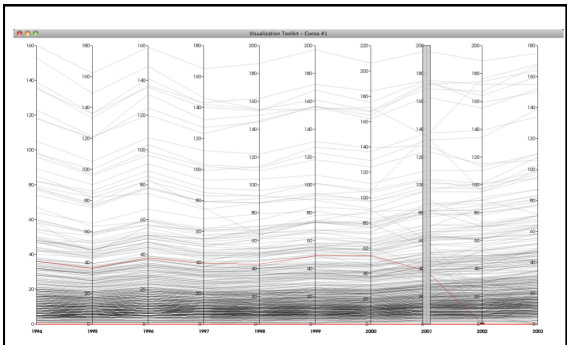


Figure 2: Chart showing Cortlandt St Station, closed after September 11, 2001

Artem

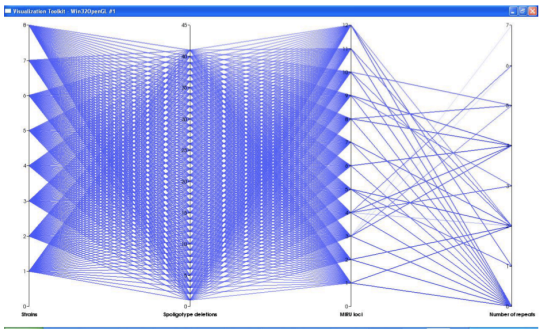
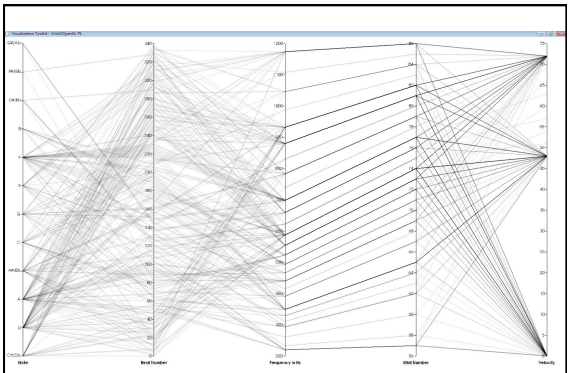
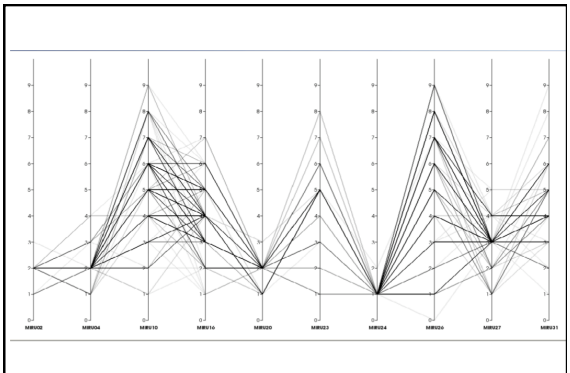


Figure 2: Multiple-biomarker tensor for AFRI strains using parallel coordinates

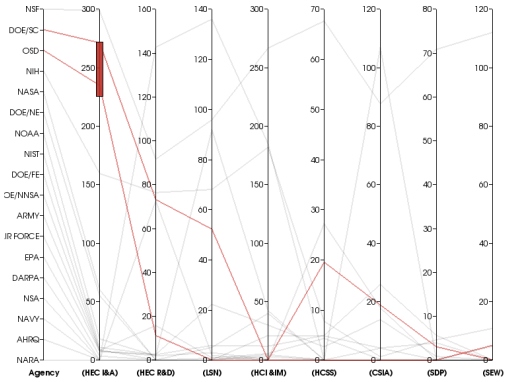
Cagri



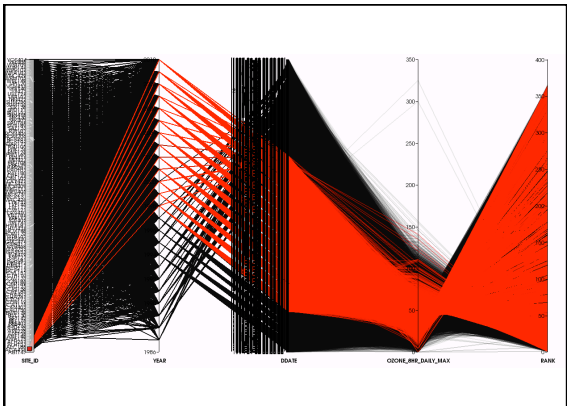
"Mozart", Tyler



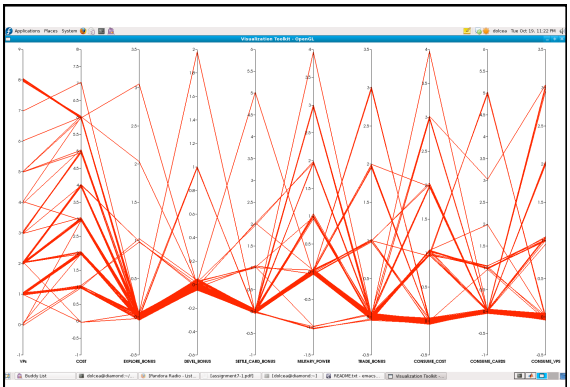
"Trends in MIRU", Amina



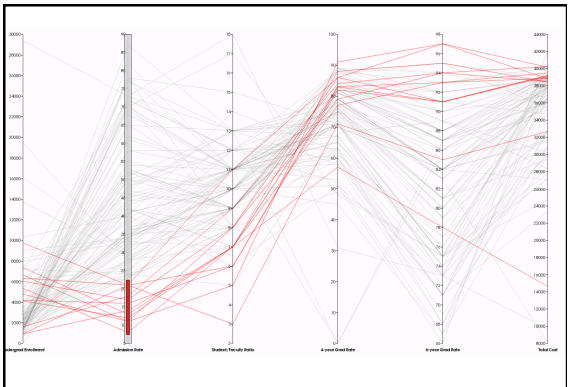
"Government Funding", Elsa



"Ozone", Daniel



"Race for the Galaxy", Andrew



"Top Colleges in US", Greg

Today's Class

- Highlights from the HW7
- This Week's Readings
- Next Week's Readings
- Examples of Visualization for Debugging
- Info on Visual Debugging in VTK
- Your Debugging Challenges
- HW8: Final Project Progress Report

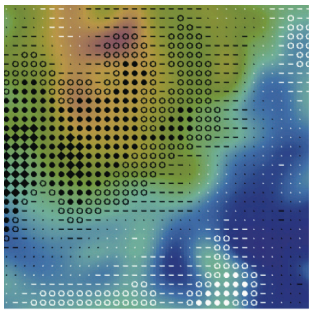
Readings for This Week:

- Ben Schneiderman, "The eyes have it: A task by data type taxonomy for information visualization", *Visual Languages, 1996*

- *Visual Information-Seeking Mantra:*
  - *overview first*
  - *zoom and filter*
  - *then details on demand*

Readings for This Week:

- Colin Ware, "Quantitative Texton Sequences for Legible Bivariate Maps," *IEEE Transactions on Visualization and Computer Graphics*, 2009.

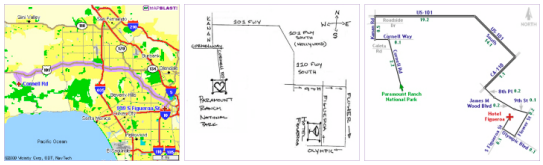


Today's Class

- Highlights from the HW7
- This Week's Readings
- Next Week's Readings
- Examples of Visualization for Debugging
- Info on Visual Debugging in VTK
- Your Debugging Challenges
- HW8: Final Project Progress Report

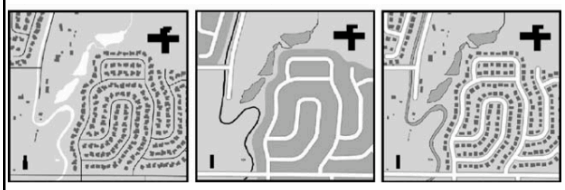
Readings for Next Week:

- Maneesh Agrawala & Chris Stolte, "Rendering Effective Route Maps: Improving Usability Through Generalization", SIGGRAPH 2001



Readings for Next Week:

- Cynthia A. Brewer and Barbara P. Buttenfield, "Mastering map scale: balancing workloads using display and geometry change in multi-scale mapping", Geoinformatica 2009



Today's Class

- Highlights from the HW7
- This Week's Readings
- Next Week's Readings
- Examples of Visualization for Debugging
- Info on Visual Debugging in VTK
- Your Debugging Challenges
- HW8: Final Project Progress Report

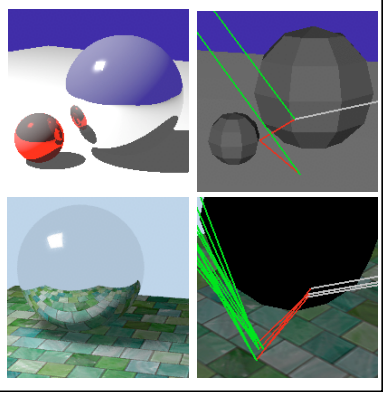
"Advanced" Debugging

Applies to software development, and other sciences too!

- Debugging Level 1:
  - Remove syntax errors in compilation
- Debugging Level 2:
  - Produces an answer
- Debugging Level 3:
  - Matches the output provided by the instructor
- Debugging Level 4:
  - Hypothesize system behavior
  - Develop & run experiments
  - Collect data & analyze results
  - Validate (or repeat process)

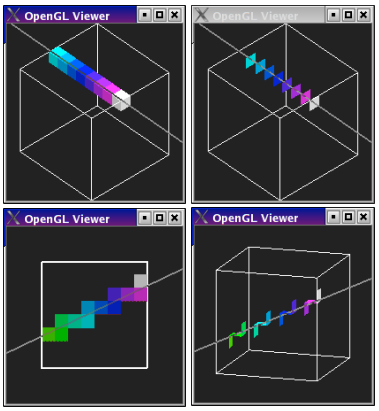
Ray Tracing

- Debug angle & direction of reflection, shadow, & refraction rays
- Solution: Draw the rays traced for a single pixel, use color for different ray types



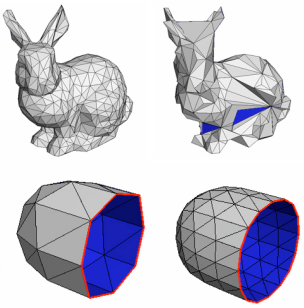
Traversing Spatial Data Structures

- Solution: Draw solid box for each visited cell
- Solution: Draw solid quad for each cross cell face



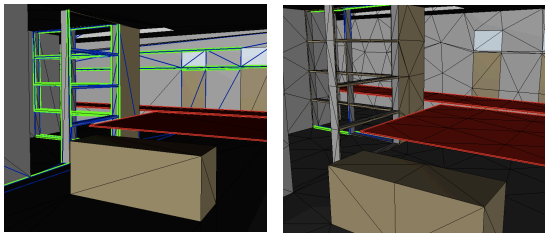
Mesh Connectivity

- Maintain consistent orientation of triangles
- Visualize surface self-intersections
- Solution: Color the "back" side blue
- Maintain connectivity through local simplification and subdivision operations
- Solution: Color edges with only 1 triangle neighbor red



"Watertight" Model Construction

- Red = edge with only 1 triangle neighbor
- Yellow = edge with > 2 triangle neighbors (non manifold)
- Green = triangle with zero area
- Blue = triangle that is neighbor to a zero area triangle

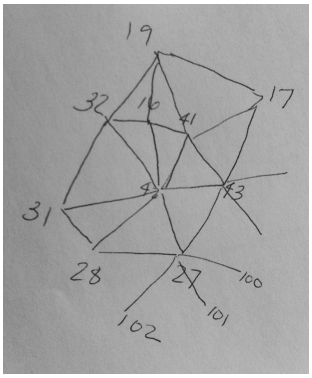


Mesh Topology

- Neighborhood & local editing
- Lots of print statements:

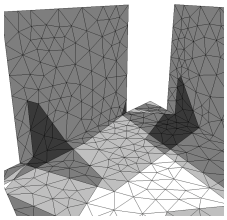
```
Triangle 206: 31 32 42
Triangle 207: 31 42 28
Triangle 208: 41 19 17
Triangle 209: 42 41 43
Triangle 210: 28 42 27
<etc.>
```

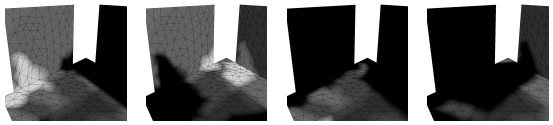
- Solution: Draw by hand
- Could use VTK graphs instead!



Visibility & Smooth Projection

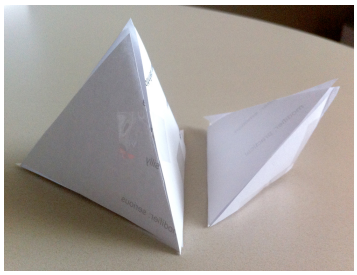
- Occlusions & Projector Visibility
- Fade in/fade out for transitions
- Make sure the sum of all projectors = 1
- Solution: Visualization the number of projectors each patch can see
- Solution: Visualize blending weights for each projector





How Tetrahedra Fill Volumetric Space

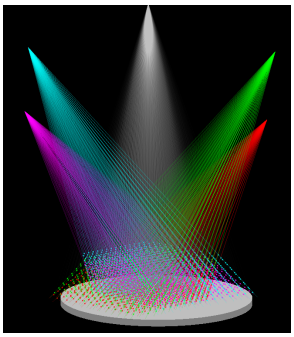
- Drawing on (in 2D) didn't work
- Creating an OpenGL visualization didn't work (even with transparency)



- Solution: build lots of paper & tape models

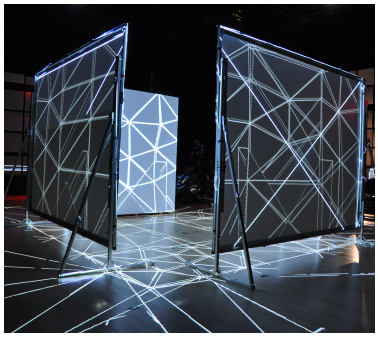
4x4 Calibration Projection Matrices

- Sanity check position & direction of camera & each projector
- Understand distribution of calibration error
- Solution: Render all point samples in a common coordinate system



Projection

- Validate the projector world coordinate calibration
- Solution: Project the mesh from each projector, verify that the images closely align



- Surprisingly, this became one of our more popular "demo"s & this image made the RPI 2010 Research calendar

Today's Class

- Highlights from the HW7
- This Week's Readings
- Next Week's Readings
- Examples of Visualization for Debugging
- Info on Visual Debugging in VTK
- Your Debugging Challenges
- HW8: Final Project Progress Report

Introduction to VTK: Visual Debugging

Visual Debugging

- Look at the state of an algorithm…
- Without inserting messy rendering code into the algorithm itself

Primer: Writing a VTK class

- vtkInformation and vtkInformationVector
- RequestData()

Header

```
#ifndef __vtkTestFilter_h
#define __vtkTestFilter_h
#include "vtkPolyDataAlgorithm.h"
#include "vtkSmartPointer.h"

class vtkTestFilter : public vtkPolyDataAlgorithm
{
public:
  vtkTypeMacro(vtkTestFilter,vtkPolyDataAlgorithm);
  static vtkTestFilter *New();
 vtkTestFilter();

protected:
int RequestData(vtkInformation *, vtkInformationVector **, vtkInformationVector
*);

private: vtkTestFilter(const vtkTestFilter&); // Not implemented.
void operator=(const vtkTestFilter&); // Not implemented.
};
#endif
```

Visual Debugging

- Need to store the state of the algorithm

```
// … Do something…
appendFilter->Update();

// Store the state:
this->Output->DeepCopy(appendFilter->GetOutput());
```

- Need to give access to the "state" of the algorithm

```
vtkPolyData* GetIntermediateOutput()
{
  return this->Output;
}
```

Use events/observers

- Inside the algorithm:
  - this->InvokeEvent(this->RefreshEvent, NULL);
- Outside the algorithm:
  - this->Filter->AddObserver(this->Filter->RefreshEvent, this, &CustomStyle::CallbackFunction);
- You can define multiple events by indexing from vtkCommand::UserEvent –
  - this->RefreshEvent = vtkCommand::UserEvent + 1;

Example

- http://www.vtk.org/Wiki/VTK/Examples/Cxx/Utilities/VisualDebugging

Today's Class

- Highlights from the HW7
- This Week's Readings
- Next Week's Readings
- Examples of Visualization for Debugging
- Info on Visual Debugging in VTK
- Your Debugging Challenges
- HW8: Final Project Progress Report

Your Debugging Challenges

- Describe a challenging coding or logic bug from your coursework/research/internship/ summer job
  - Ultimately, what was the bug (if known)?
- What tool(s) did you use in trying to fix the bug?
  - Did you use any form of visualization in debugging?
  - What tools from VTK (if any) could you use to create a quick visualization to aid in debugging?
  - What infrastructure could you create to solve general debugging problems of this type?

Today's Class

- Highlights from the HW7
- This Week's Readings
- Next Week's Readings
- Examples of Visualization for Debugging
- Info on Visual Debugging in VTK
- Your Debugging Challenges
- HW8: Final Project Progress Report

Final Project Guidelines

- Read & summarize 2-3 papers related to your project & incorporate/extend components of this work
- Save early iterations of the visualization (and any "bloopers"), to show the progression of your visualization design and data exploration
- Tues Nov 16th *or* Fri Nov 19th 10-11:50am? pre-review of final projects in *Visual Design* class

Final Project Progress Report #1

- Assignment #8 (due next week!)
- Read & summarize of 2-3 relevant papers (or equiv. quantity of books/reference webpages), not just papers we have read for class!
- Some progress to report… Ideally something visual that we can critique as a group and give you feedback