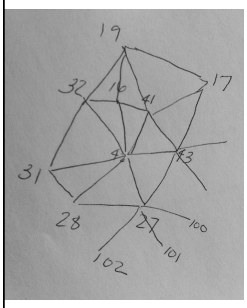
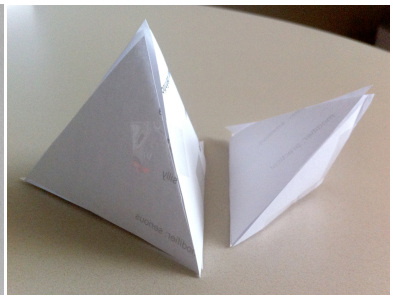
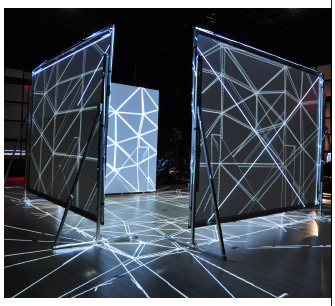
Visualization for Debugging



Today's Class

- Today's Readings:
  - "Nanocubes for Real-Time Exploration of Spatiotemporal Datasets" Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013
  - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Tuesday
- Examples of Visualization for Debugging
- Today's Crayon Exercise / Progress Post #2

- Motivation: *Exploratory* visualization challenging with ever growing datasets
  - Data summarization can be inaccurate misleading
- Memory goal: Work with the amount of memory available on a standard laptop
- Speed goal: query times < millisecond on single thread on laptop/server class compute node.
  Want query time that is proportional to the size of the visualization output (# of pixels)
- Datasets: work with large, spatiotemporal, and multidimensional
- Contributions
  - Novel data structures enables real-time use of DataCubes
  - Brushable, visual encodings
  - Case studies

"Nanocubes for Real-Time Exploration of Spatiotemporal Datasets",
Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013

"Data Cubes"

- Data cubes are structures that perform aggregations across every possible set of dimensions of a table in a database, to support quick exploration
- Requires a lot of memory as # of dimensions increase

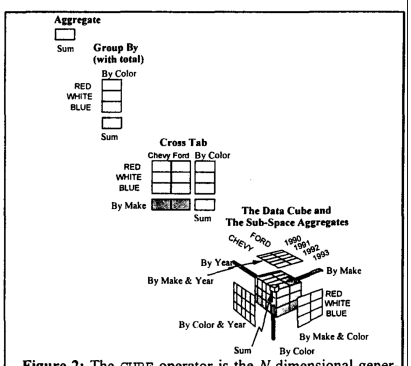


Figure 2: The CUBE operator is the *N*-dimensional generalization of simple aggregate functions. The 0D data cube is a point. The 1D data cube is a line with a point. The 2D data cube is a cross tab, a plane, two lines, and a point. The 3D data cube is a cube with three intersecting 2D cross tabs.

"Data Cube: A Relational Aggregation Operator Generalizing Group-By,
Cross-Tab, and Sub-Totals", Gray, Bosworth, Layman, & Pirahesh, IEEE, 1997

"Data Cubes"

- Proper data aggregation (count, sum, min, max) is better than random sampling summarization of the data (which can miss information)
- Database operations: GROUP_BY, CUBE, ROLL_UP

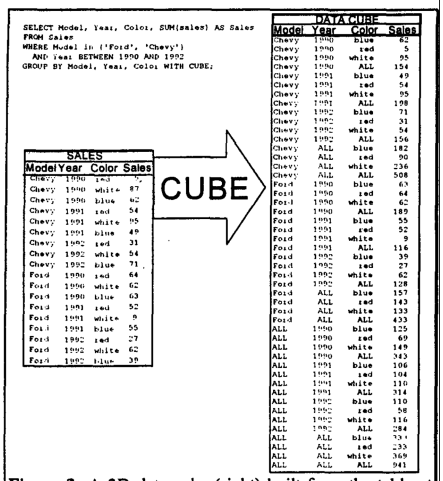


Figure 3: A 3D data cube (right) built from the table at the left by the CUBE statement at the top of the figure.

"Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals", Gray, Bosworth, Layman, & Pirahesh, IEEE, 1997

- Visual Information-Seeking Mantra "Overview first, zoom and filter, then details-on-demand" [Shneiderman 1996]
- Nanocubes Paper: added ROLL_UP_CUBE

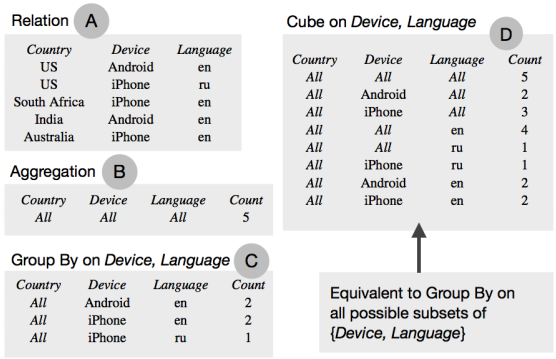


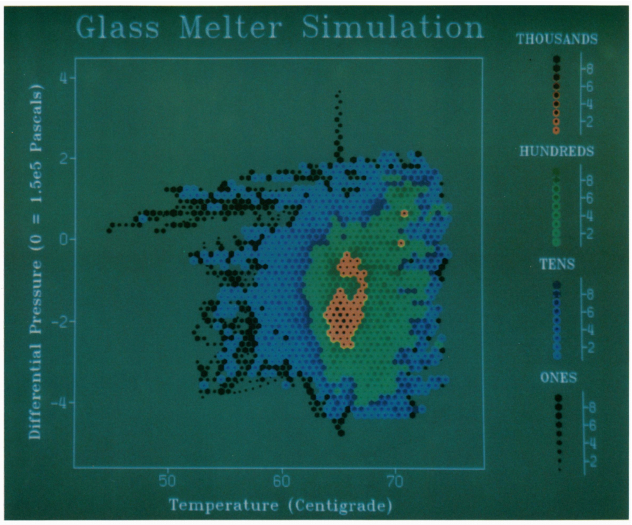
Fig. 4. A sample relation and its associated aggregation operators.

"Nanocubes for Real-Time Exploration of Spatiotemporal Datasets", Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013

- Previous work on parallelization (shipping data & computation to large set of compute nodes) does not achieve suitable low latency.
  - Query *prediction* is needed to provide interactive experience.
- Summed Area Table
- Applications:
  - Twitter feed, brightkite social network
  - Commercial Airline flight history
  - Cell phone Call Data Records (from one company privately shared with authors), anonymized
- Variety of test data:
  1.5 million – 1 billion objects, 4 MB -> 46 GB

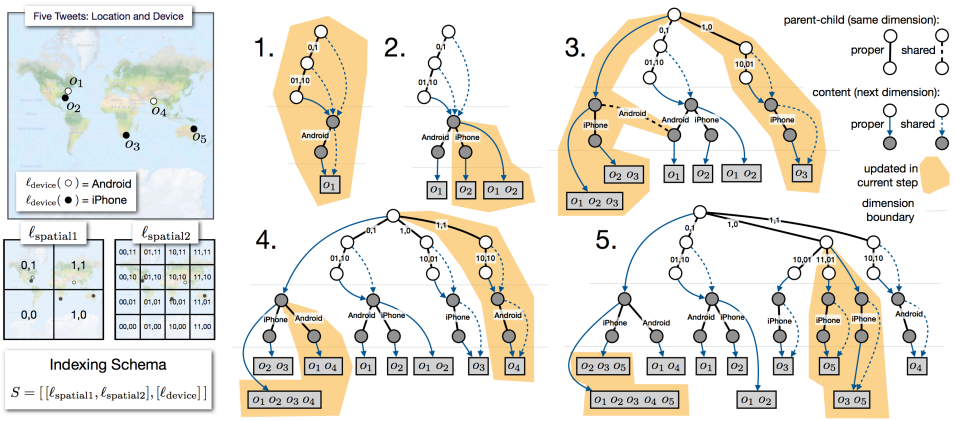
"Nanocubes for Real-Time Exploration of Spatiotemporal Datasets",
Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013

Scatterplot vs. Equivalent Density Plot



"Scatterplot Matrix Techniques for Large N",
Carr, Littlefield, Nicholson, & Littlefield, Journal of American Statistical Association 1987

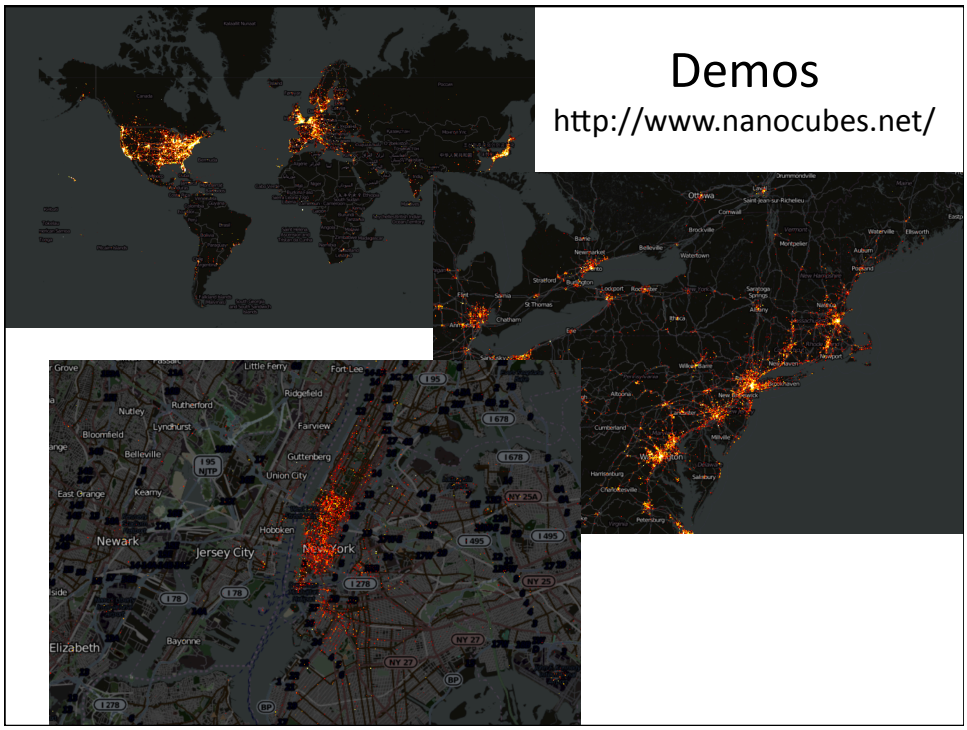
Building the Nanocube



"Nanocubes for Real-Time Exploration of Spatiotemporal Datasets",
Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013

Demos

http://www.nanocubes.net/



Today's Class

- Today's Readings:
  - "Nanocubes for Real-Time Exploration of Spatiotemporal Datasets" Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013
  - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Tuesday
- Examples of Visualization for Debugging
- Today's Crayon Exercise / Progress Post #2

DimpVis: DimP Information Visualization
(DimP = Prior paper "Video Browsing by *Direct Manipulation*")

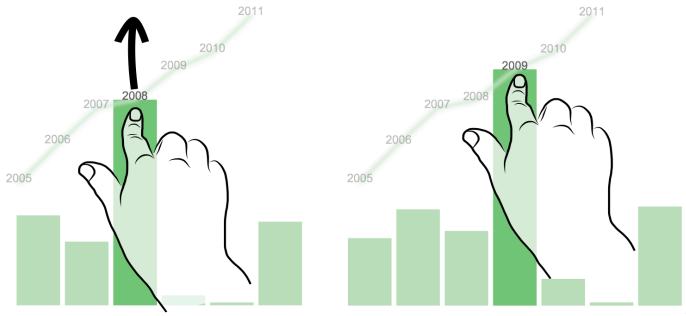


Fig. 4. Navigation in time for bar charts is achieved by dragging a selected bar vertically along its hint path. The hint path slides horizontally to stay connected with the bar and finger.

Is this intuitive? Is this practical?
Dynamic bar charts… cool!
No natural way to overlay the trajectory Time axis overlaps (left->right) with whatever (non time-based) data the horizontal axis represents

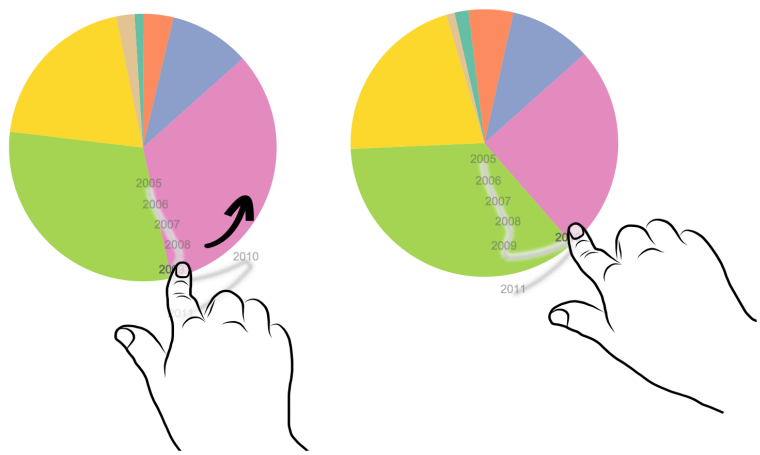


Fig. 6. Pie charts are adjusted in time by dragging an edge of a segment in angular directions along its hint path. The hint path slides in and out along the radius to remain connected with the finger position and edge of the segment.

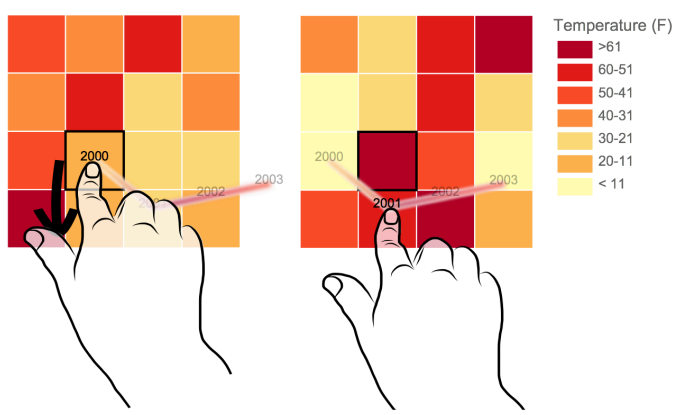


Fig. 7. Heat maps are adjusted in time by dragging vertically in the space of the colour scale. The hint path slides horizontally to stay connected with the cell and finger.

But is this intuitive way to interact?
No natural direction to spatially overlay the trajectory

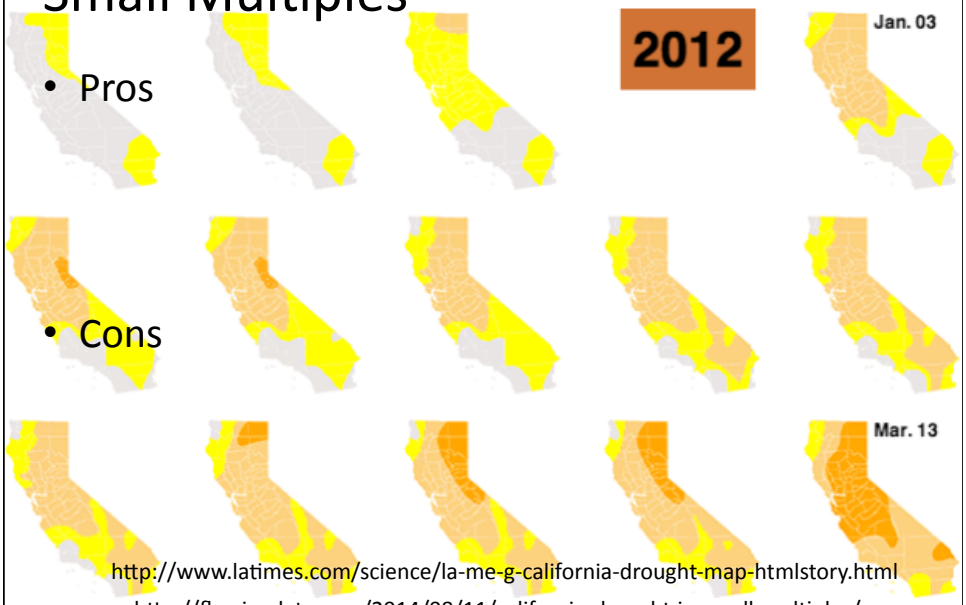
- "Interaction Ambiguity"
  - What if data is not clearly headed in a direction?
  - What if data has constant value for two or more timesteps?
  - What if data returns in the direction it came from?
  - What if there are multiple times that "the bar graph is at value 500"?
- Avoid/disambiguate:
  - Temporal continuity enforced at cusps
  - Fake loops can be traversed to change time, even if the object you are directly manipulating does not change.
  - Nonintuitive?
  - Does this work if the path has lots of overlaps/loops?

Fig. 3. Temporal ambiguity occurs in a scatter plot when a point does not move between time steps. Temporal navigation capabilities are provided in ambiguous using loops in which one transit around the loop corresponds to one time step.

- Related work
  - Visualize trends/trajectories of multiple individual data points
- Keeps your focus on the data you are interested in (not distracted by spatially separate slider).
- Touch is preferable to mouse (which also separates user from data).
  - somatic: affecting the body; corporeal or physical; the vertebrate nervous system that regulates voluntary movement
  - Very game-like, fun to use
  - Like playing connect the dots (but what if I want to go my own way?)
  - Futuristic & forward thinking
  - But mouse precision is so much better than "fat finger"
- Hard to understand interaction from static figures (despite lots of diagrams and overlays)
  - Video was more clear
    Did you read the paper too or just watch the video?

Small Multiples

- Pros
- Cons



http://www.latimes.com/science/la-me-g-california-drought-map-htmlstory.html
http://flowingdata.com/2014/08/11/california-drought-in-small-multiples/

Evaluation

- Quantitative evaluation compared to traditional time slider and small multiples – mixed results, but better in some cases
- Measure performance (time & error rate) to complete tasks in reading & observing trend
  - Spatial task "When is the bar at height 3?" vs.
  - Temporal task "In 1995, what is the height of the bar?"
  - Comparison "When is A greater than B?"
  - Distribution "When do both A & B decrease?"
  - Outliers "When does A move in opposite direction to other points?"
- Test cases for study
  - Real data, but adjusted to ensure just 1 "correct" answer
  - 3 techniques (DimpVis, slider bar, small multiples) x 2 visualizations (scatter plot, bar chart)

- Technique Comments/Concerns/Ideas
  - Dragging along a path isn't that you have true control over where the object should go, you are constrained to follow its path
  - DimP same as time slider, only more confusing since it the path bends in strange ambiguous ways – seems like a gimmick
  - Showing the paths of 1 or more data points forward & back in time seems helpful
  - Instead of the path (directly manipulating) of a single point, can we show (control time via) the average path of multiple points?
  - Instead of replacing a linear slider (so many other ways to do this) what about using this for a true 2D "slider"?

- Additional "features"
  - Traditional time slider
    - why did they have to add this? DimP manipulation is nonintuitive/hard to control?
  - Flexible rubber band (threshold for direct manipulation)
  - Snap to discrete timepoint
  - Fast forward (jump to time point with tap)
- Tasks removed from study because
  - Not enough time
  - Performing the task was "frustratingly difficult" during pilot study
- Set of clear hypotheses
  - Use of statistical tests (ANOVA), judges for data set size whether the measured differences are significant or could probabilitistically just be noise (not significant)
  - Their hypotheses partially supported or rejected by measurements
  - Subjectively DimpVis was preferred!
    But Is this reproducible with another set of users?

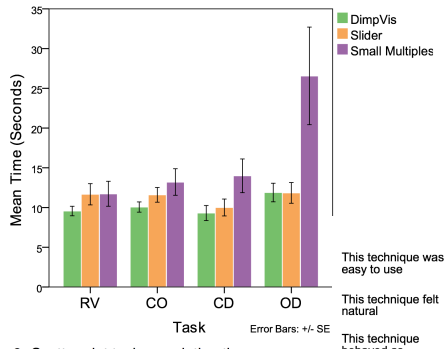


Fig. 8. Scatter plot task completion times.

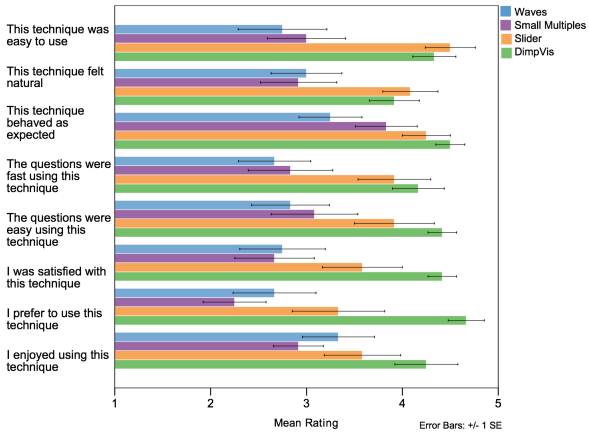


Fig. 9. Scatter plot subjective ratings.

- "Enhanced engagement with data"
- Some misuse of tool (intuition/prior experience w/ other interfaces)
  - Attempted 2 finger interaction
  - Attempted to drag in opposite direction vertically
- Open Questions
  - Scalability
- Great job removing outside factors from what was being tested
- Clearly listed hypotheses (# of hypotheses seemed low)
- Refreshing that they didn't just propose a visualization, they tested it thoroughly!
- "temporal navigation" felt like I was reading a paper on time travel ☺

Today's Class

- Today's Readings:
  - "Nanocubes for Real-Time Exploration of Spatiotemporal Datasets" Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013
  - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Tuesday
- Examples of Visualization for Debugging
- Today's Crayon Exercise / Progress Post #2

"Visualization, Selection, and Analysis of Traffic Flows", Scheepens, Hurter, van de Wetering, van Wijk, IEEE InfoVis 2015

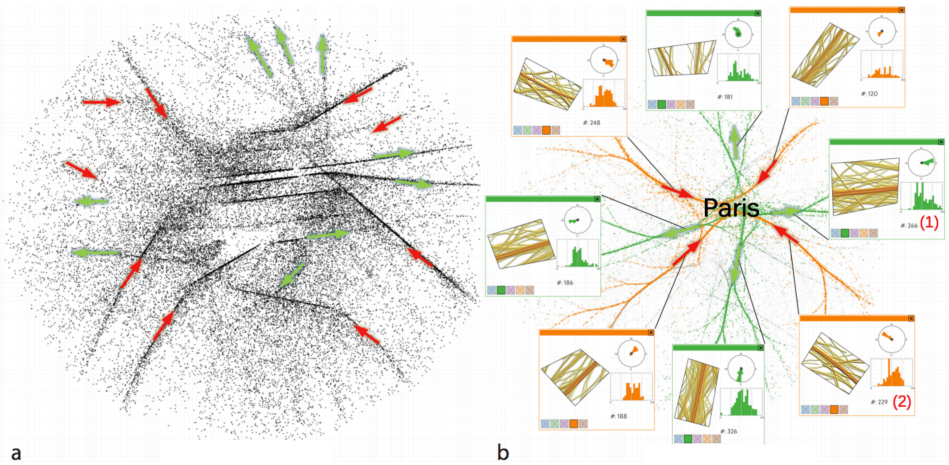


Fig. 10. (a) An overview of traffic flows over the Paris area. Outgoing traffic flows have been marked with the green arrows, while incoming traffic flows have been marked with a red arrow. (b) The traffic flows have been bundled, selected, and the dynamics of these traffic flows are displayed using the movable windows.

“Guidelines for Effective Usage of Text Highlighting Techniques”, Strobelt, Oelke, Kwon, Schreck, Pfister, IEEE InfoVis 2015

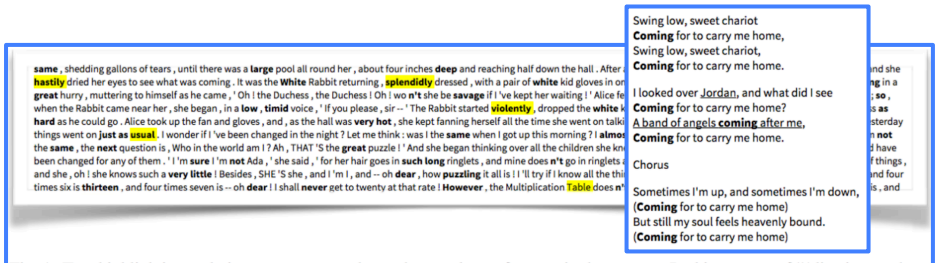


Fig. 1: Text highlighting techniques are commonly used to mark text features in documents. In this excerpt of “Alice in wonderland” all occurrences of adjectives and adverbs derived from part-of-speech tagging are highlighted in bold font, while words with typical adjective/adverb endings are highlighted with yellow background.

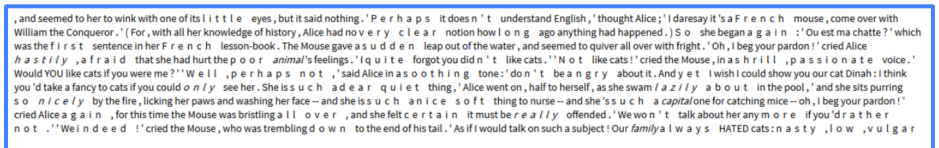


Fig. 11: Example of combining techniques letter spacing and italics – according to our analysis this is not an effective combination for highlighting two equally important text features.

“Poemage: Visualizing the Sonic Topology of a Poem”, McCurdy, Lein, Coles, Meyer, IEEE InfoVis 2015

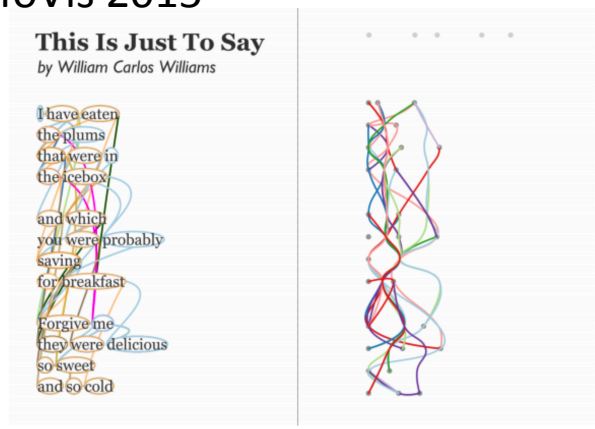


Fig. 4. The *beautiful mess* highlights the sonic isolation of the word *“you”* in this poem. A visualization from a technology probe obscures this isolation *(left)*, whereas rerouting in Poemage reveals the anomaly *(right)*.

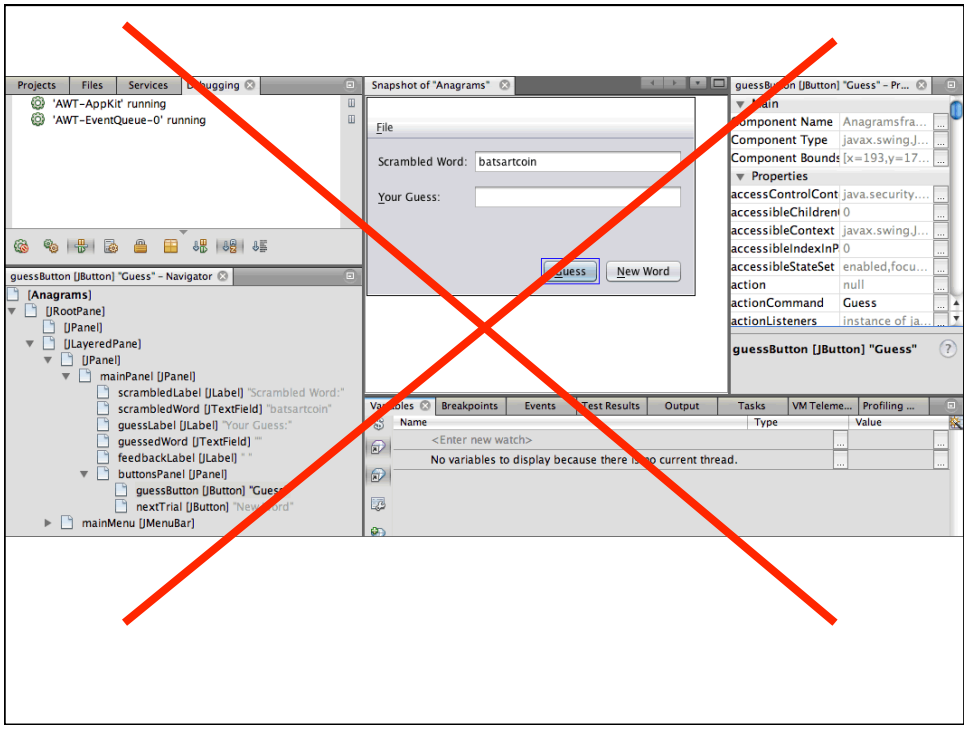
Today's Class

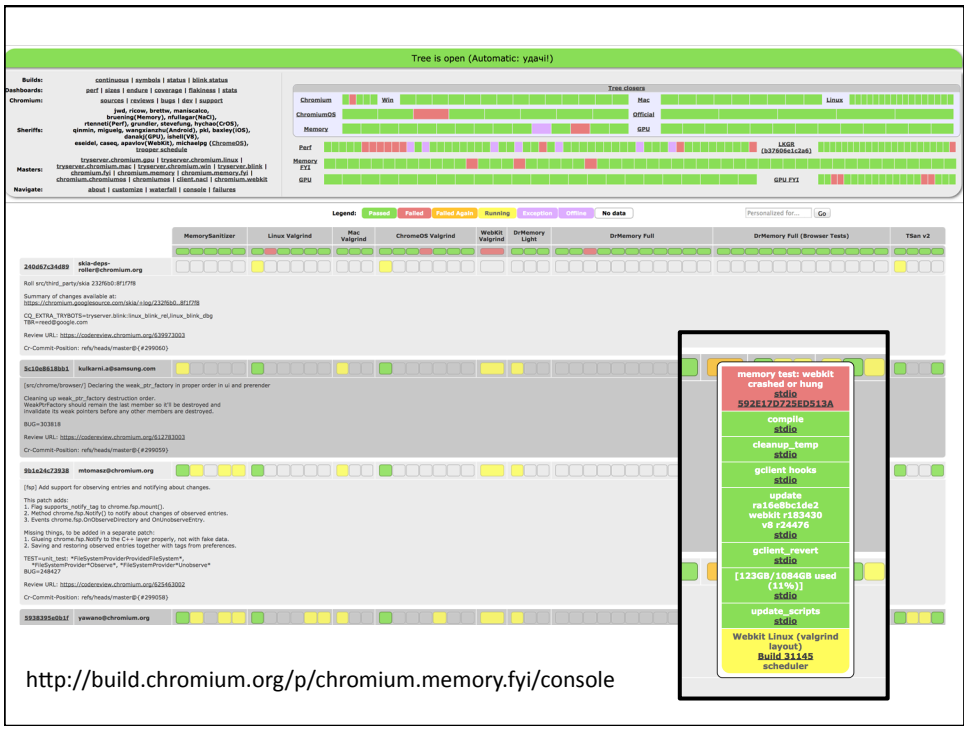
- Today's Readings:
  - "Nanocubes for Real-Time Exploration of Spatiotemporal Datasets" Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013
  - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Tuesday
- Examples of Visualization for Debugging
- Today's Crayon Exercise / Progress Post #2

"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)

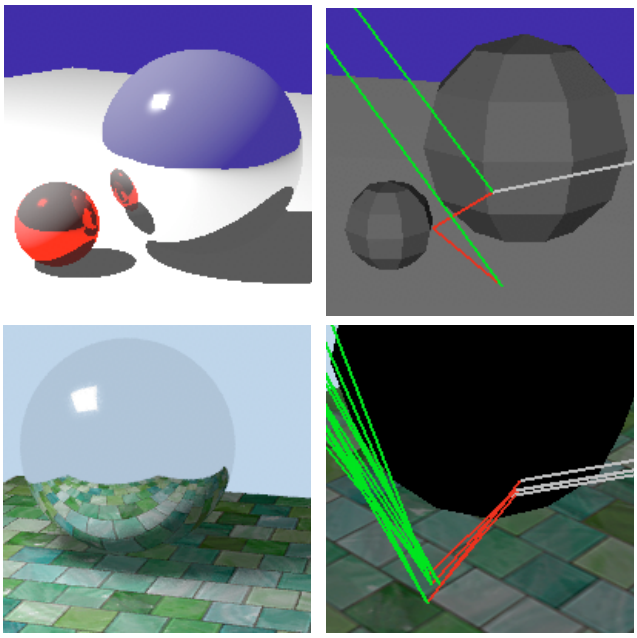




http://build.chromium.org/p/chromium.memory.fyi/console

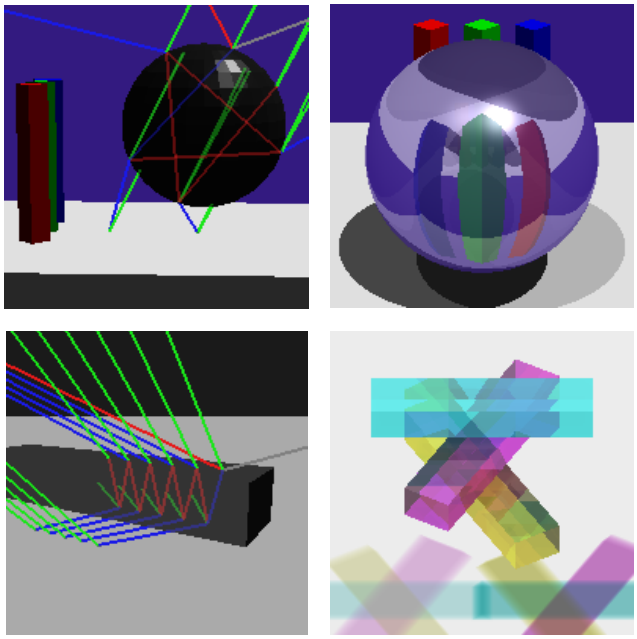
Ray Tracing

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



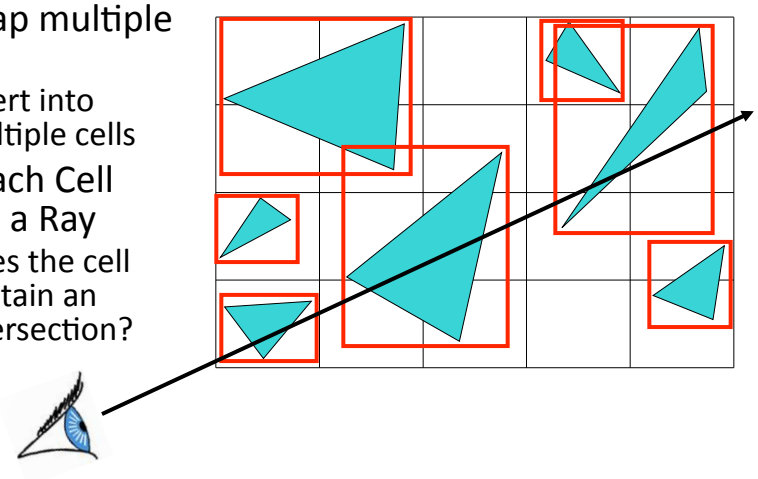
Ray Tracing

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



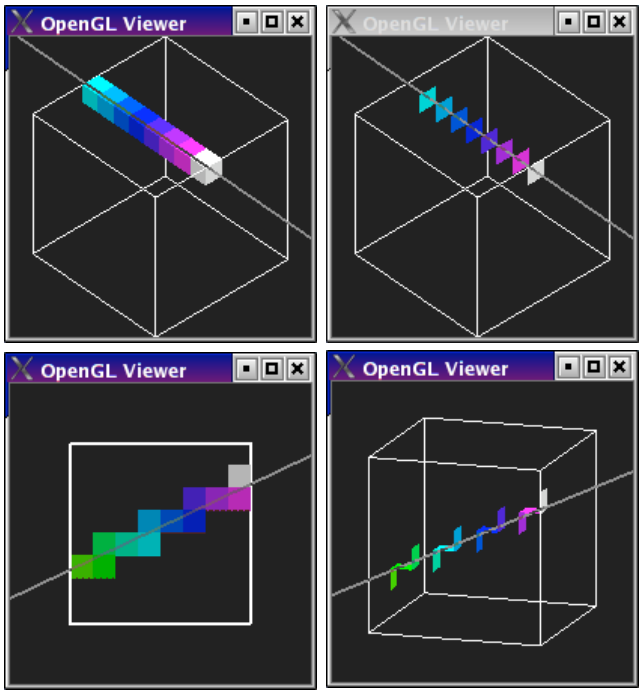
Spatial Data Structures for Efficiency

- Primitives that overlap multiple cells?
  - Insert into multiple cells
- For Each Cell Along a Ray
  - Does the cell contain an intersection?



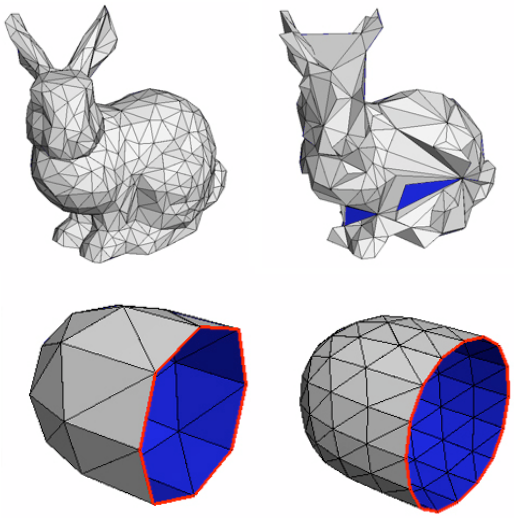
Traversing Spatial Data Structures

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



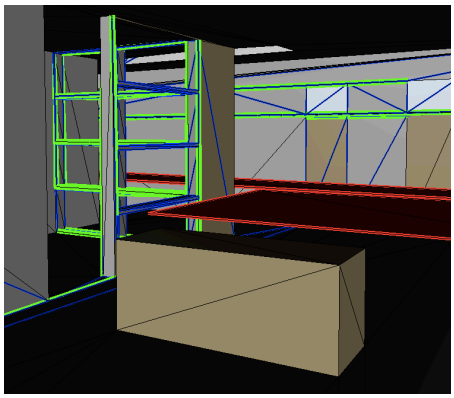
Mesh Connectivity

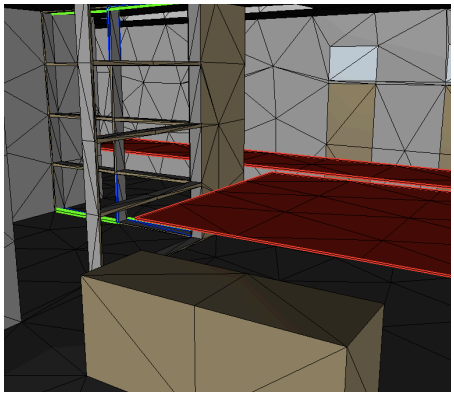
- Maintain consistent orientation of triangles
- Visualize surface self-intersections
- Solution: Color the "back" side blue
- Maintain connectivity through local simplification and subdivision operations
- Solution: Color edges with only 1 triangle neighbor red



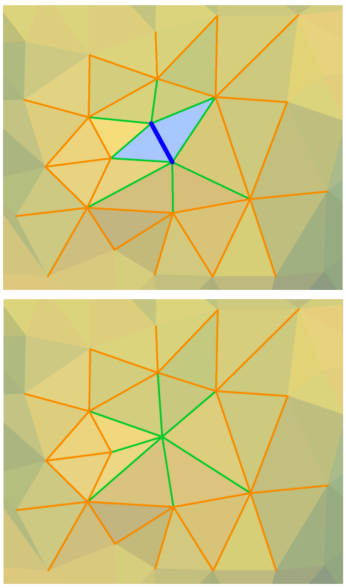
"Watertight" Model Construction

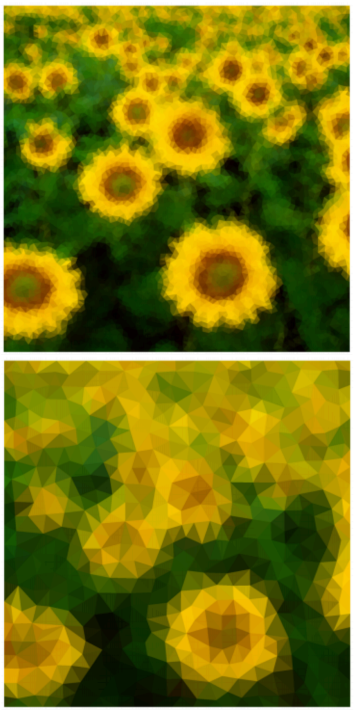
- Red = edge with only 1 triangle neighbor
- Yellow = edge with > 2 triangle neighbors (non manifold)
- Green = triangle with zero area
- Blue = triangle that is neighbor to a zero area triangle





Mesh Simplification



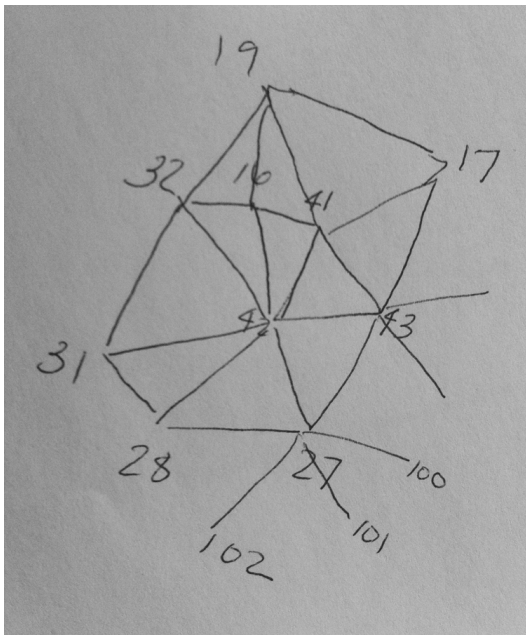


Mesh Topology

- Neighborhood & local editing
- Lots of print statements:

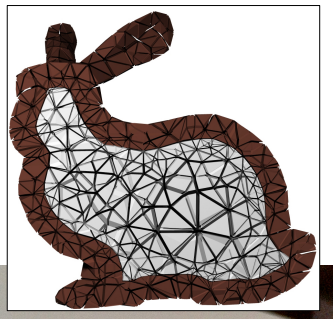
```
Triangle 206: 31 32 42
Triangle 207: 31 42 28
Triangle 208: 41 19 17
Triangle 209: 42 41 43
Triangle 210: 28 42 27
<etc.>
```

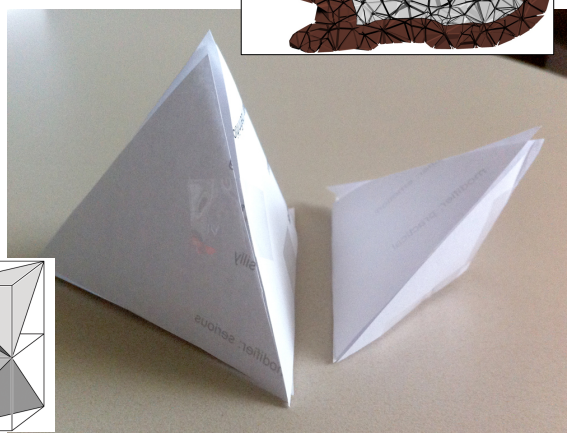
- Solution: Draw by hand
- This graph drawing could be automated!

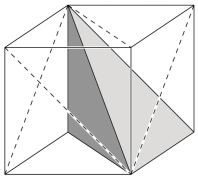


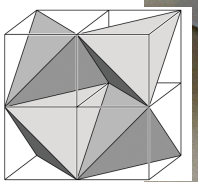
How Tetrahedra Fill Volumetric Space

- Drawing (in 2D) didn't work
- Creating an OpenGL visualization didn't work (even with transparency)
- Solution: build lots of paper & tape models









Volumetric Exploration

- For developer or designer/ modeler
- Fast interaction to understand sublayer details

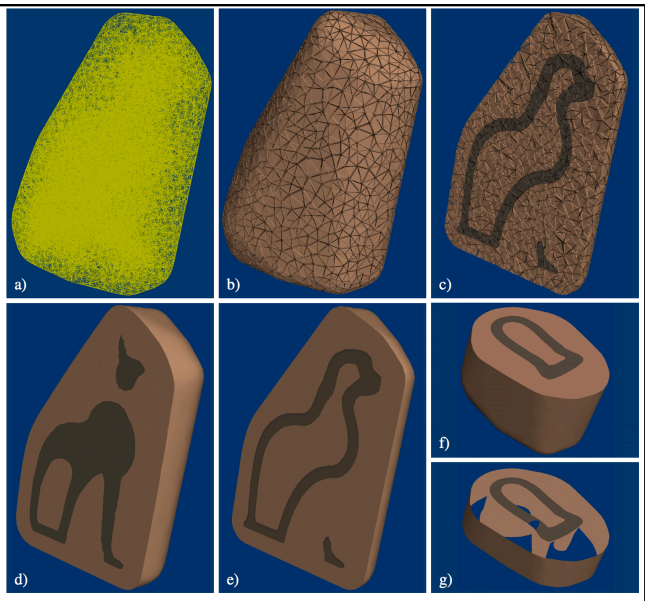


Figure 9.8: A wireframe rendering of the 49,995 tetrahedra cat mesh, shown in a), is not very informative. To better visualize the shape and material of the elements, we can b) render the tetrahedra with cracks between them and c) move a clip plane through the model interactively. To view just the structure, we can render only the air/material and material/material interfaces to create d), e), & f), the illusion of a slicing through a solid material. The illusion is broken if g) a second clip plane is added.

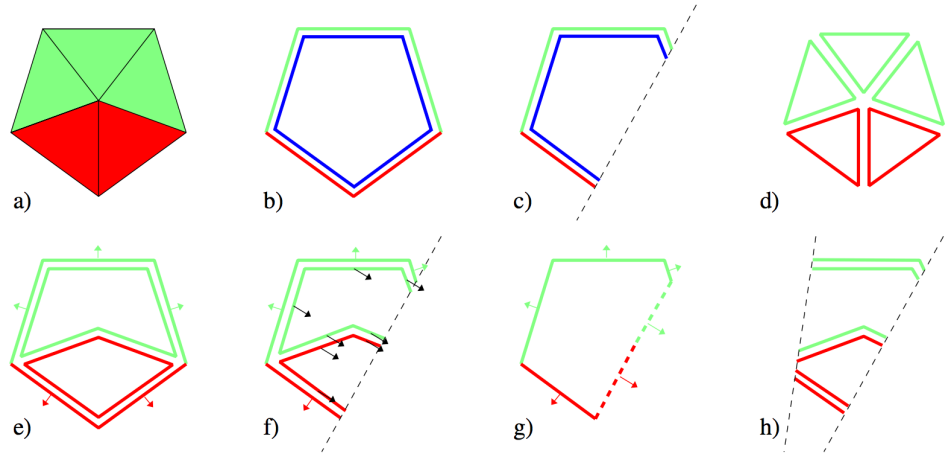
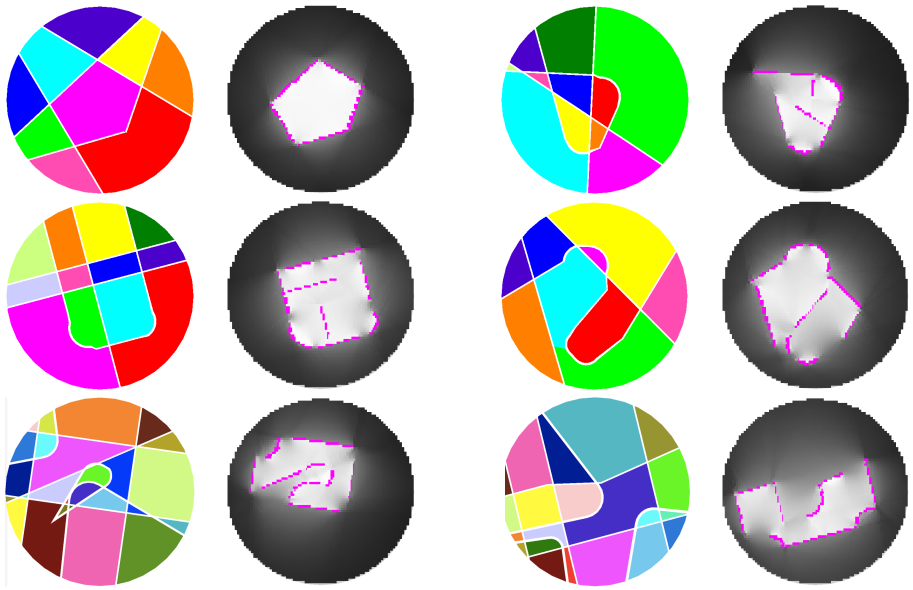
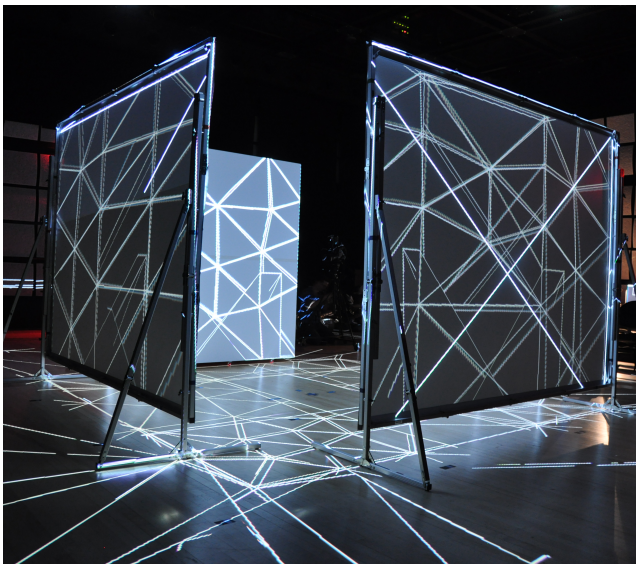


Figure 9.7: Two-dimensional illustrations of the different tetrahedral rendering options applied to the simple model shown in a). Two sided lighting is used to b) color the back faces of surface triangles blue, which can be viewed by c) cutting the object with a clipping plane. The individual elements can be visualized with d) crack-style rendering. The shape of the internal structures can be viewed by rendering e) two-sided internal material interfaces. The internal surfaces are f) assigned the normal of the clipping plane (black arrows), to give the illusion of a g) ghost surface where the object has been sliced. The illusion is broken if h) a second clipping plane is added.

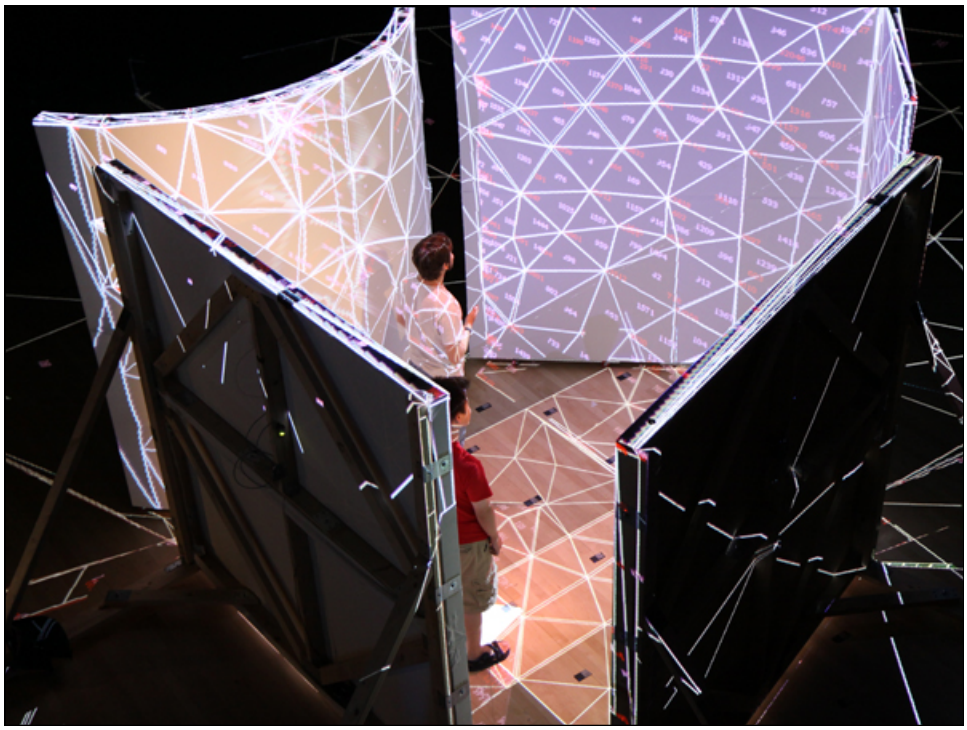
To understand relationships and threshholds



Projection

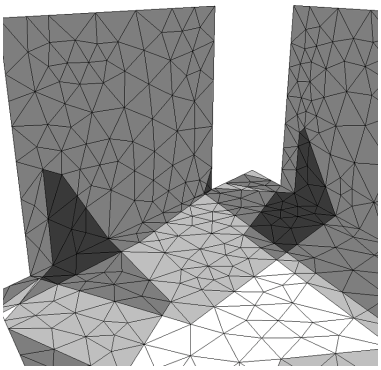


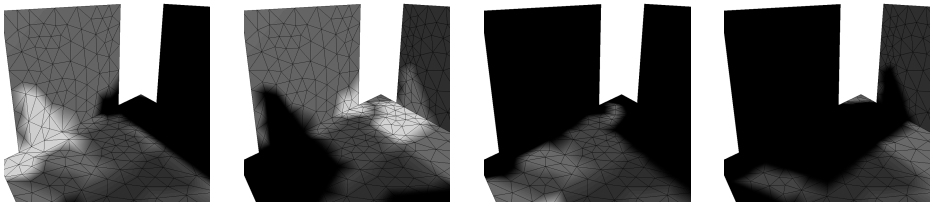
- Validate the projector world coordinate calibration
- Solution: Project the mesh from each projector, verify that the images closely align
- Surprisingly, this became one of our more popular "demo"s & this image made the RPI 2010 Research calendar



Visibility & Smooth Projection

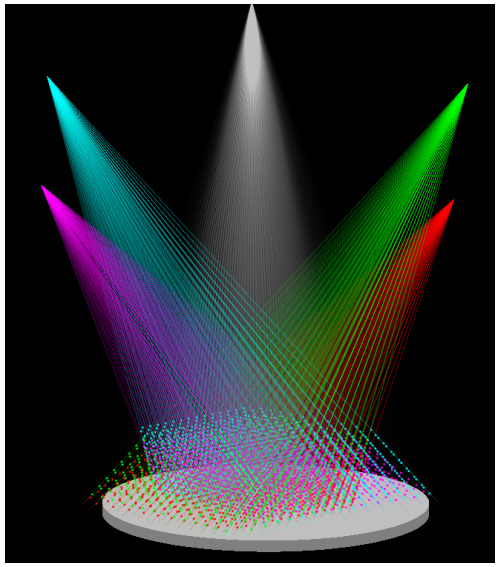
- Occlusions & Projector Visibility
- Fade in/fade out for transitions
- Make sure the sum of all projectors = 1
- Solution: Visualization the number of projectors each patch can see
- Solution: Visualize blending weights for each projector





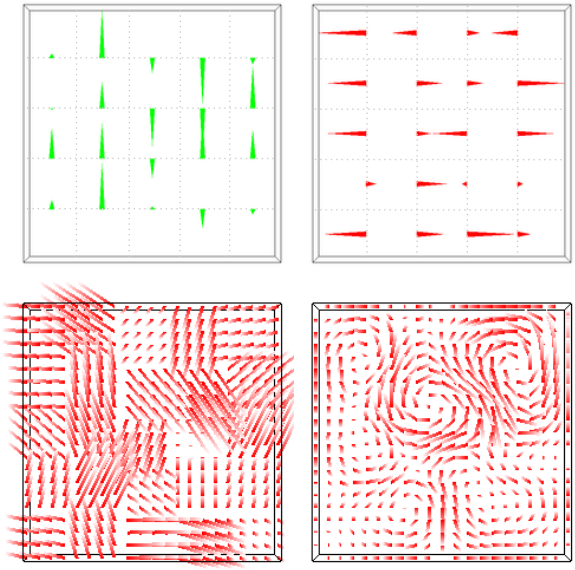
4x4 Calibration Projection Matrices

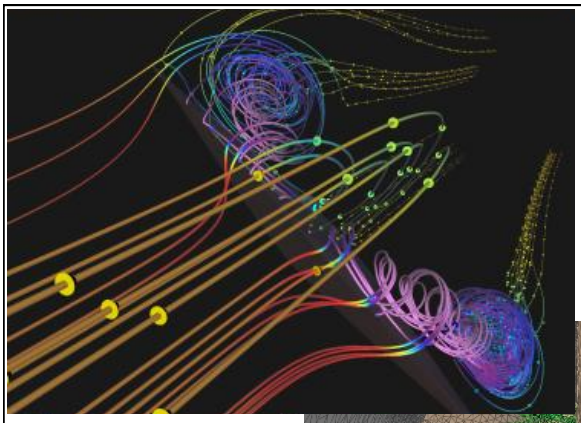
- Sanity check position & direction of camera & each projector
- Understand distribution of calibration error
- Solution: Render all point samples in a common coordinate system



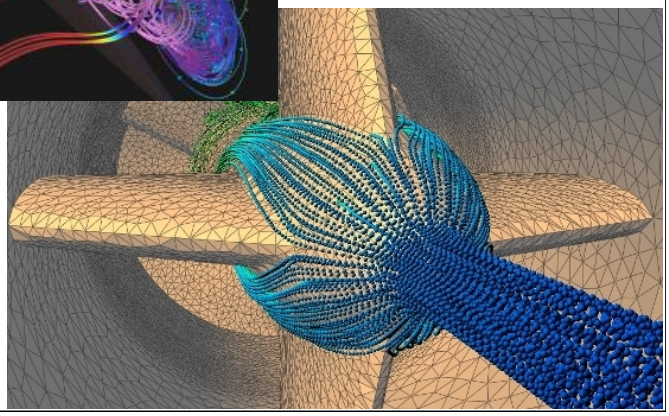
Fluid Dynamics

- Visualize grid cell face velocities
- Interpolate face velocities to create smoothly varying vector field
- How to visualize a 3D vector field?





http://www.eng.utah.edu/~ehan/math6790/project2/Report.html



http://tobiasrick.org/research_particles.html

Today's Class

- Today's Readings:
  - "Nanocubes for Real-Time Exploration of Spatiotemporal Datasets" Lins, Klosowski, and Scheidegger, TVCG, IEEE Visualization 2013
  - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Tuesday
- Examples of Visualization for Debugging
- Today's Crayon Exercise / Progress Post #2

Crayon Exercise: Visual Debugging!

- Teams of two. Someone you haven't worked with before…
  You may not team up with your final project partner!
- Each person should explain a software debugging/validation challenge (either current or anticipated) from their final project.
  - Note: You should probably focus on "unit test" debugging of a subtask of the project not an whole picture "end-to-end" debugging.
- Explain the details of a sample input, intermediate computation information that could be gathered, and expected output for that subtask.
- The other person should propose and sketch a visualization that might be helpful to diagnose and solve the problem.
  - Use color, spatial layout, labels, animation, connections, thoughtfully, to cram as much information as might be helpful in debugging.
  - It doesn't need to be beautiful, but the user should be able to find patterns or outliers.