

CSCI 4550/6550 Interactive Visualization

<https://www.cs.rpi.edu/~cutler/classes/visualization/S24/>

Lecture 18: Visualization for Debugging

Today

- **Final Project Proposals & Accessing Academic Papers**
- Visualization for Debugging
 - *Not:* Traditional Debugging
 - *Not:* Software Development Dashboards
 - Lots of helpful examples from class & research!
- Today's Readings:
 - "Active Reading of Visualizations", Walny, Huron, Perin, Wun, Pusch, and Carpendale, IEEE InfoVis 2017
 - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Friday

Final Project Proposals

- Feedback (& grades) posted yesterday
- Overall the grades were not great...
 - Majority of proposals were missing a descriptive project title
 - Many missing timeline with sufficient milestones / “who will do what”
- Some proposals were confusing / vague / not self-contained
 - *This is a communication-intensive course!*
- COVID visualization seems popular again ...
 - Yes, we have LOTS of data – but it is not all consistent in format or collection methodologies (either by locale or by year!)
- If you are tackling a huge & well-studied topic
 - Has this already been done? *You aren't expected to do novel research!*
 - Suggest: Narrow focus to find a unique angle or question

How to get a paper when you hit the publisher paywall?

-

How to get a paper when you hit the publisher paywall?

- Give up. Read the other paper.
- Just watch the video (hey, you weren't going to read it anyway)
- Use Google – some other university may have it on their reading list.
- Go to the author's webpage. Many/most conferences/journals allow the author to post a "pre-print" on their own site.
- Go to the library! RPI "RensSearch" has online access to many/most technical journals. And other materials via Interlibrary Loan. You may need VPN off-campus and/or your RCS credentials.
- Ask if your employer has a subscription or procedure to obtain materials.
- Unpaywall, Open Access, ResearchGate, Library Genesis, Citationsy, Sci-Hub, Other questionably-legal methods...

https://en.wikipedia.org/wiki/Aaron_Swartz

Today

- Final Project Proposals & Accessing Academic Papers
- Visualization for Debugging
 - *Not: Traditional Debugging*
 - *Not: Software Development Dashboards*
 - Lots of helpful examples from class & research!
- Today's Readings:
 - "Active Reading of Visualizations", Walny, Huron, Perin, Wun, Pusch, and Carpendale, IEEE InfoVis 2017
 - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Friday


What is Debugging?

1. Identify the problem
2. Isolate the source of the problem
3. Correct the problem - OR -
Determine a way to work around the problem
4. Test the correction and make sure it works

<https://searchsoftwarequality.techtarget.com/definition/debugging>

The First Computer Bug

9/9

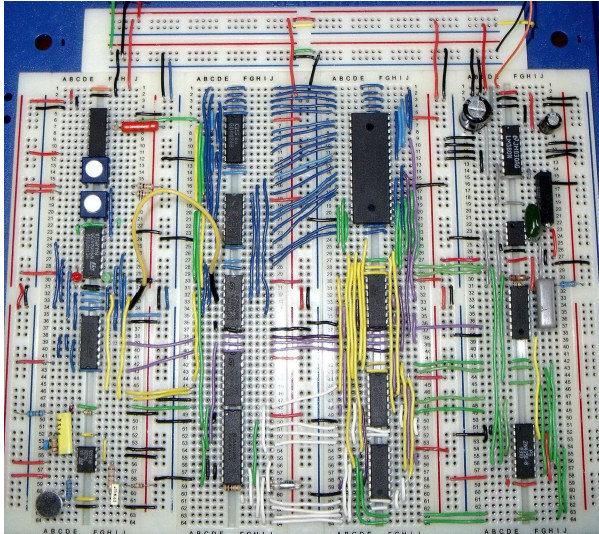
0800	Machine started		
1000	" stopped - machine	✓	{ 1.2700 9.037 847 025
	13°C (032) MP - MC	✓	9.037 846 995 correct
	(033) PRO 2	✓	2.130476415 (03) 4.615925059 (-2)
	conv	✓	2.130476415
	Relays 6-2 in 033 failed special speed test		
	in relay		" " " " " "
	Relays changed		
1100	Started Cosine Tape (Sine check)		
1525	Started Multi-Adder Test.		
1545			Relay #70 Panel F (moth) in relay.
1700	Machine started.		
1700	closed down.		

First actual case of bug being found.

Relay #70
3145
Relay 3378

<https://www.nationalgeographic.org/thisday/sep9/worlds-first-computer-bug/>

“Thermal Debugging”



Mis-wiring a computer chip can cause it to overheat, burn out and fail.

If your circuit isn't working touch each chip and if one is hot to the touch, that's likely the problem!

<https://electronics.stackexchange.com/questions/430152/the-art-of-using-breadboard>

Learning to Debug Software

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

*Being a CS1/DS Mentor:
Helping students
do steps 1-3...*

*Applies to software
development, and other
sciences too!*

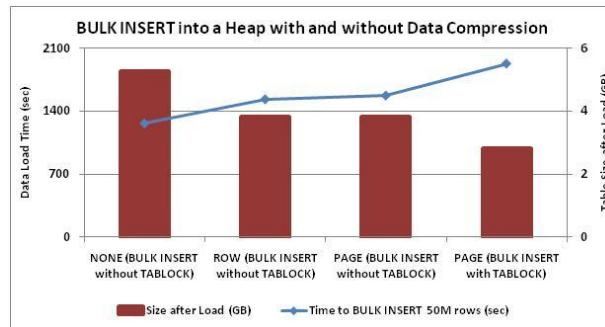
- Do the #'s match your intuition/hypotheses?
- Do you expect a positive correlation or a negative correlation?

SPLITS		
MI	PACE	ELEVATION
1	10:14	5 ft
2	10:07	-5 ft
3	9:59	35 ft
4	9:38	-43 ft
5	9:54	13 ft
6	9:58	-9 ft
7	9:45	-3 ft
8	9:55	11 ft
9	9:46	15 ft
10	9:46	-4 ft
11	8:25	-10 ft
12	8:24	-13 ft
13	8:08	3 ft

Did this person get faster or slower at the end of their run?

Which is "better" performance? Higher or lower values?

- Can you improve even simple plots to allow the viewer to read(understand) information faster?



Print Debugging is Not Enough

- *Recompiling* the whole program between step is slow
- *Non-deterministic* bugs can be difficult to reproduce
- *We don't know for sure what we need to print*
→ we often print too much or too little
- Requires a *long time to re-run* if error is late in program execution
- **Instead:** We can fully capture a specific run
 - Enable the program/system to “core dump” (save full memory content, all registers, etc.)
 - Then use traditional debugging tools to explore
 - May even facilitate *reverse step-by-step debugging!*

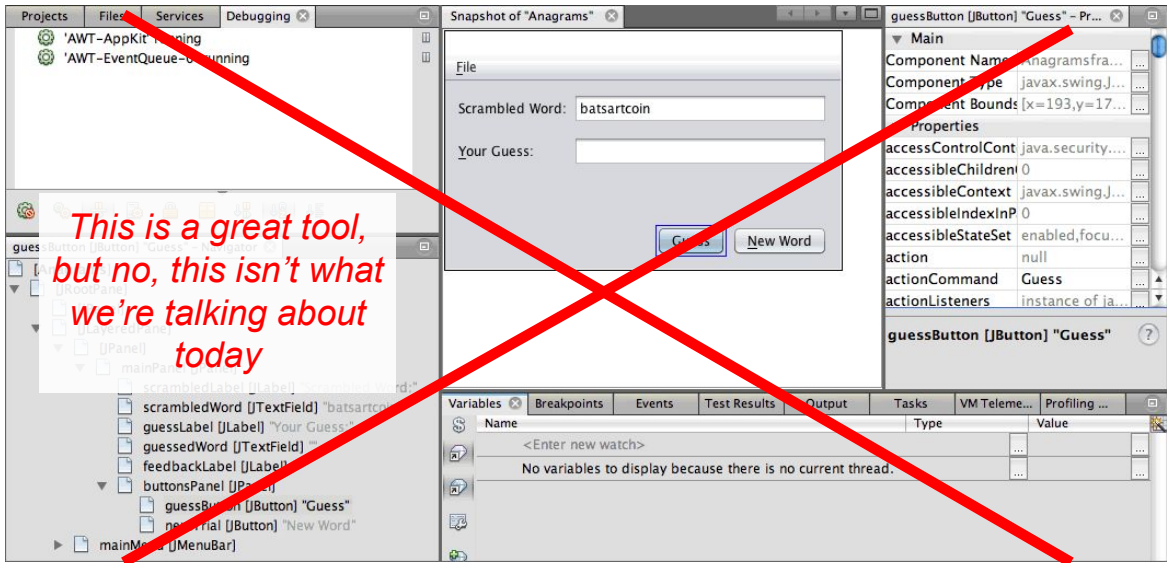
Traditional Debugging Tools

- *Inspect variable state*: don't try to print everything!
- *Inspect call stack*: who called this function with bad data??
- *Force program into a specific state*:
 - Externally modify specific variable values
 - Externally coordinate thread/process interleaving
- Operating System specific:
 - View OS specific system state,
e.g. other threads, loaded libraries, kernel level info, etc.

Today

- Final Project Proposals & Accessing Academic Papers
- **Visualization for Debugging**
 - *Not*: Traditional Debugging
 - ***Not*: Software Development Dashboards**
 - Lots of helpful examples from class & research!
- Today's Readings:
 - "Active Reading of Visualizations", Walny, Huron, Perin, Wun, Pusch, and Carpendale, IEEE InfoVis 2017
 - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Friday

Is this Visual/Visualization Debugging?



Designs for smart & efficient software development UIs should use good color and other human perception / visualization knowledge!

Review required Add your review
At least 1 approving review is required by reviewers with write access. [Learn more about pull request reviews.](#)

Some checks were not successful Hide all checks
17 successful and 1 failing checks

- ✓ Submitty PR Title Check / title-check (pull_request) Successful in 3s (Required) Details
- ✓ Submitty CI / js-lint (pull_request) Successful in 43s (Required) Details
- ✓ Submitty CI / js-unit (pull_request) Successful in 42s (Required) Details
- ✓ Submitty CI / php-lint (pull_request) Successful in 2m (Required) Details
- ✗ Submitty CI / php-unit (pull_request) Failing after 23s (Required) Details
- ✓ Submitty CI / python-lint (pull_request) Successful in 14s (Required) Details

Merging is blocked
Merging can be performed automatically with 1 approving review.

Merge without waiting for requirements to be met (bypass branch protections)

Squash and merge You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

← Back to pull request #10276

✗ [DevDependency] Bump phpunit/phpunit from 9.6.9 to 10.5.15 in /site #26776

Summary

Jobs

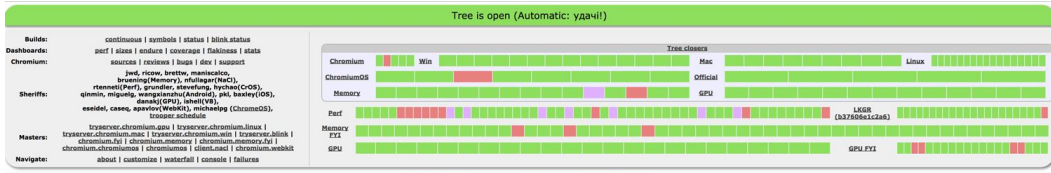
- ✓ css-lint
- ✓ js-lint
- ✓ js-unit
- ✓ php-lint
- ✗ php-unit
- ✓ python-lint
- ✓ python-unit
- ✓ yamllint
- ✓ ShellCheck
- ✓ Cypress-System
- ✓ Cypress (Gradeable)
- ✓ Cypress (Admin)
- ✓ Cypress (UI)
- ✓ Cypress (Feature)
- ✓ Integration

php-unit

Failed 3 days ago in 23s

- Set up job
- Run actions/checkout@v4
- Run shivammathur/setup-php@2.30.0
- Cache Composer
- Install Composer
- Install dependencies
- Run php unit tests
- Upload Coverage
- Post Install Composer
- Post Run actions/checkout@v4
- Complete job

<https://github.com/Submitty/Submitty/actions/runs/8398563239/job/23003574662?pr=10276>



240687234689 skia-depa-raller@chromium.org

Roll w/Chromium party/ku 232960_8f1778

Summary of changes available at: https://chromium.googlesource.com/skia/clog/232960_8f1778

CQ EXTRA: TRYBOT5=tryserver.blink_linux_blink_ref_linux_blink_dbg

TBR: need@googl.com

Review URL: <https://codereview.chromium.org/539973063>

Cr-Commit-Position: refs/heads/master@{#299066}

5c1208f58ba3 kulhari.a@amsung.com

[crChromium/browser] Declaring the weak_ptr_factory in proper order in ui and pre-render

Clipping to weak_ptr_factory destruction order: WeakPtrFactory should remain the last member so it's destroyed and invalidates its weak pointers before any other members are destroyed.

BUG=30318

Review URL: <https://codereview.chromium.org/512783003>

Cr-Commit-Position: refs/heads/master@{#299099}

8b124c12938 mtomasz@chromium.org

[tag] Add support for observing entries and notifying about changes.

This patch adds:

1. Page supports_notify_tag to chrome.Rdp.mount().
2. Method chrome.Rdp.notify() to notify about changes of observed entries.
3. Events chrome.Rdp.DrObserveDirectory and OnUnobserveEntry.

Missing things to be added in a separate patch:

1. Changing chrome.Rdp.notify() to the C++ user property, not with fake data.
2. Saving and restoring observed entries together with tags from preferences.

TEST=run_test "WebkitSystemProviderTest" "WebkitSystemProviderTest"

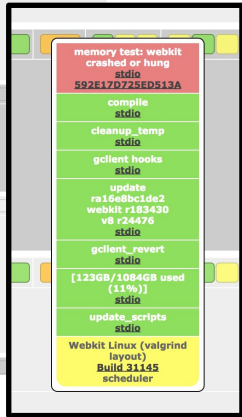
FidSystemProviderObserver *FidSystemProviderObserver*

BUG=24827

Review URL: <https://codereview.chromium.org/525663002>

Cr-Commit-Position: refs/heads/master@{#299058}

59283554051 ywano@chromium.org



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

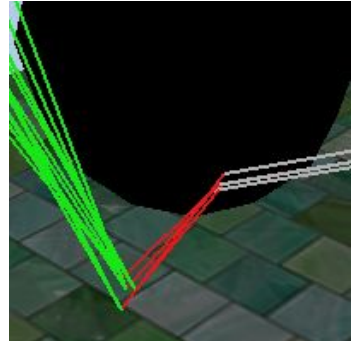
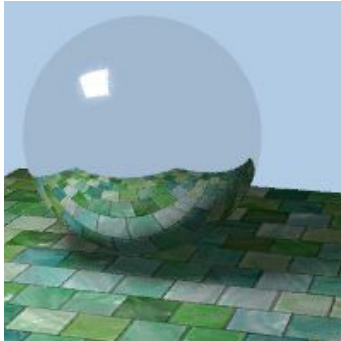
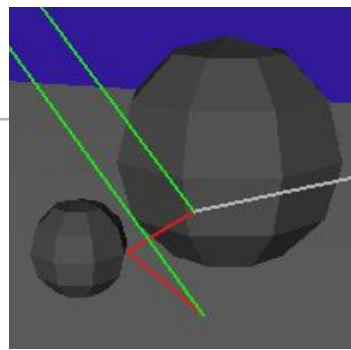
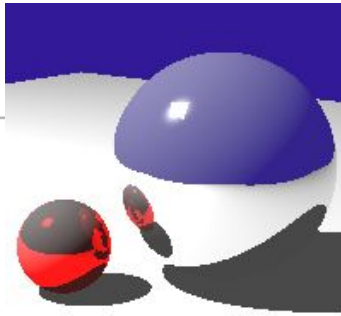
Today

- Final Project Proposals & Accessing Academic Papers
- Visualization for Debugging
 - Not: Traditional Debugging
 - Not: Software Development Dashboards
 - Lots of helpful examples from class & research!
- Today's Readings:
 - "Active Reading of Visualizations", Walny, Huron, Perin, Wun, Pusch, and Carpendale, IEEE InfoVis 2017
 - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Friday

Ray Tracing

- Need to debug angle & direction of reflection, shadow, & refraction rays

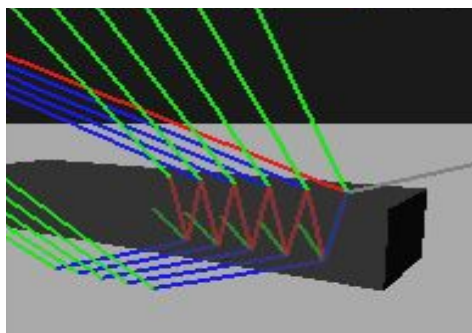
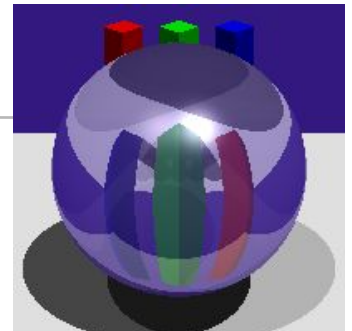
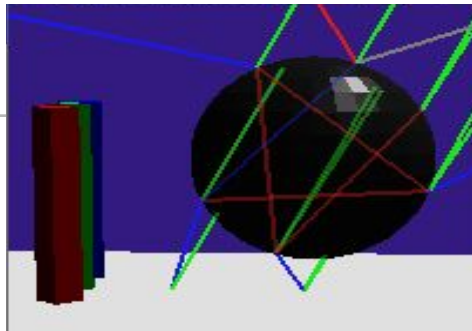
- *Solution:*
Draw the rays traced for a single pixel, use color for different ray types



Ray Tracing

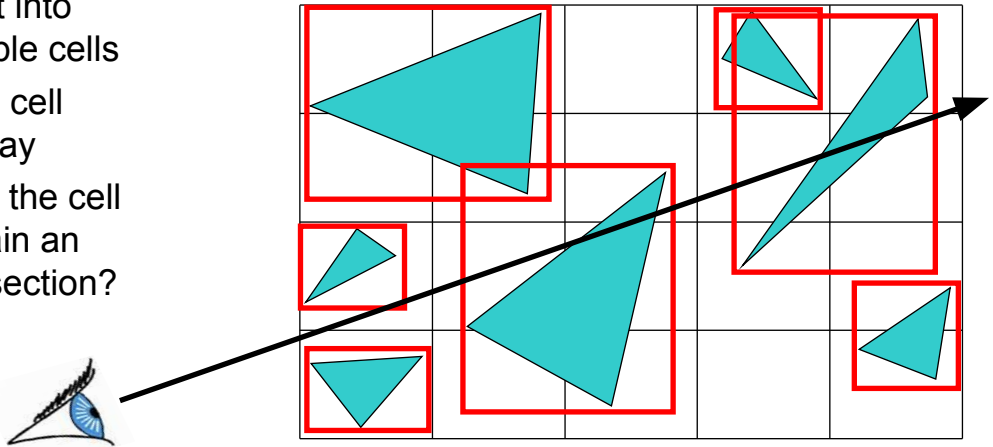
- Need to debug angle & direction of reflection, shadow, & refraction rays

- *Solution:*
Draw the rays traced for a single pixel, use color for different ray types



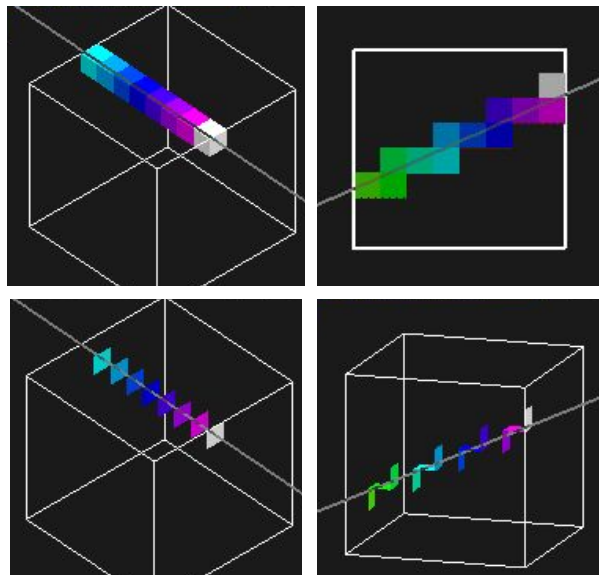
Spatial Data Structures for Efficiency

- Primitives that overlap multiple cells?
 - Insert into multiple cells
- For each cell along a ray
 - Does the cell contain an intersection?



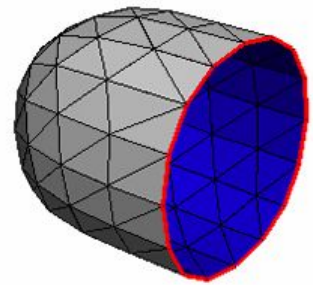
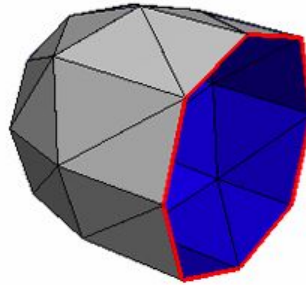
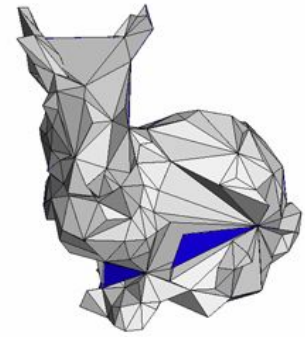
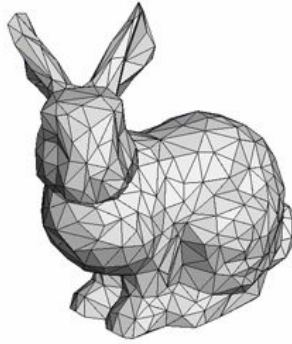
Traversing Spatial Data Structures

- *Solution 1:*
Draw solid box for each visited cell
- *Solution 2:*
Draw solid quad for each cross cell face
- Color code by order
(white = first cell, etc.)



Mesh Connectivity

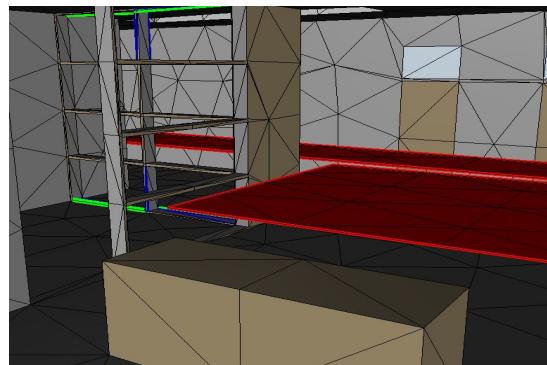
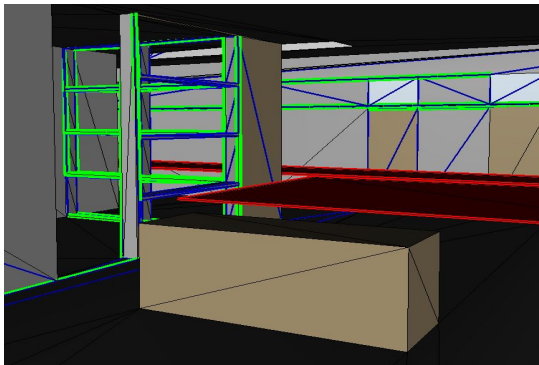
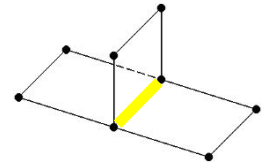
- Maintain consistent orientation of triangles
- Visualize surface self-intersections
- *Solution: Color back-facing triangles blue*
- Maintain connectivity through local simplification and subdivision operations
- *Solution: Color edges with only 1 triangle neighbor red*



“Watertight” Model Construction

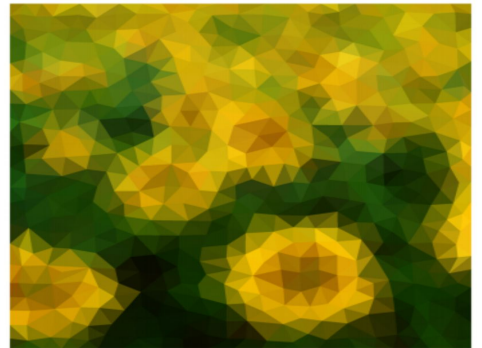
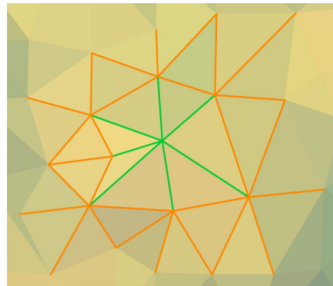
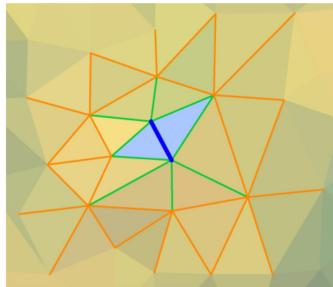
If rendering bugs / visual artifacts occur at or near these mesh conditions, we may have a lead on debugging!

- *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

- **Blue Edge:**
candidate edge
to delete
- **Light Blue Face:**
triangles to be deleted
- **Green Edge:**
edges that will merge
or change length
- **Orange Edge:**
“one ring” edges
that share a vertex
with a green edge

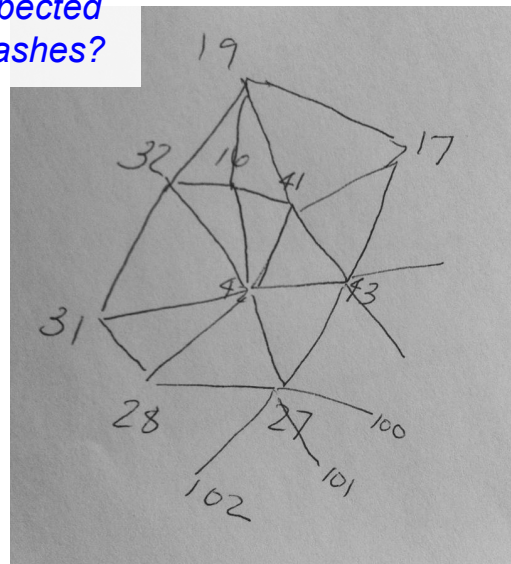


Mesh Topology

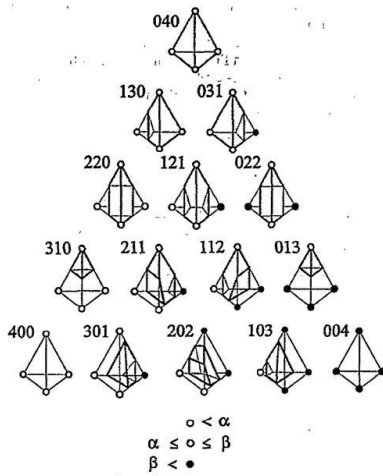
*What to do when
it mostly works, but
you have unexpected
& infrequent crashes?*

- Neighborhood & local editing
- Lots of print statements:

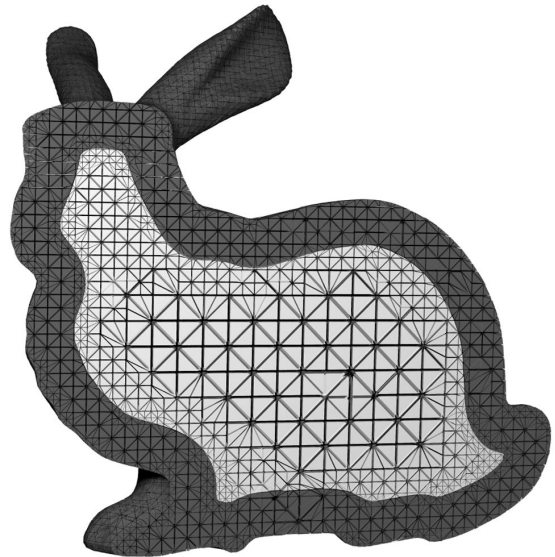
```
Triangle 206: 31 32 42  
Triangle 207: 31 42 28  
Triangle 208: 41 19 17  
Triangle 209: 42 41 43  
Triangle 210: 28 42 27  
<etc.>
```
- **Solution: Draw by hand**
- This graph drawing could be automated!



“Marching Tetrahedra”

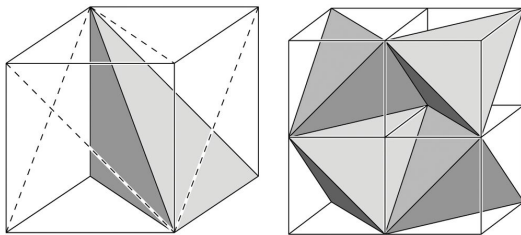


“Interval volume tetrahedrization”
Visualization '97, Nielson & Sung

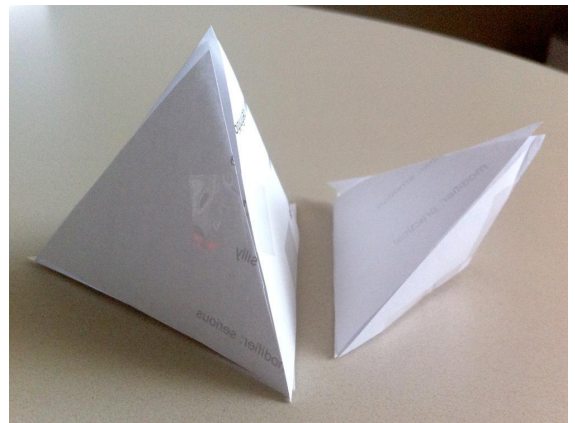


How Do Tetrahedra Fill Volumetric Space?

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



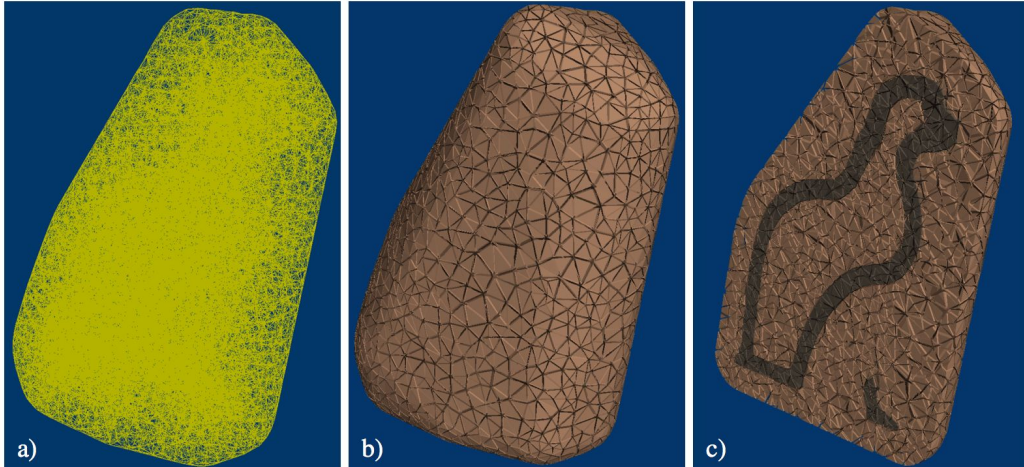
- *Solution: Build paper & tape models*



Volumetric Exploration

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

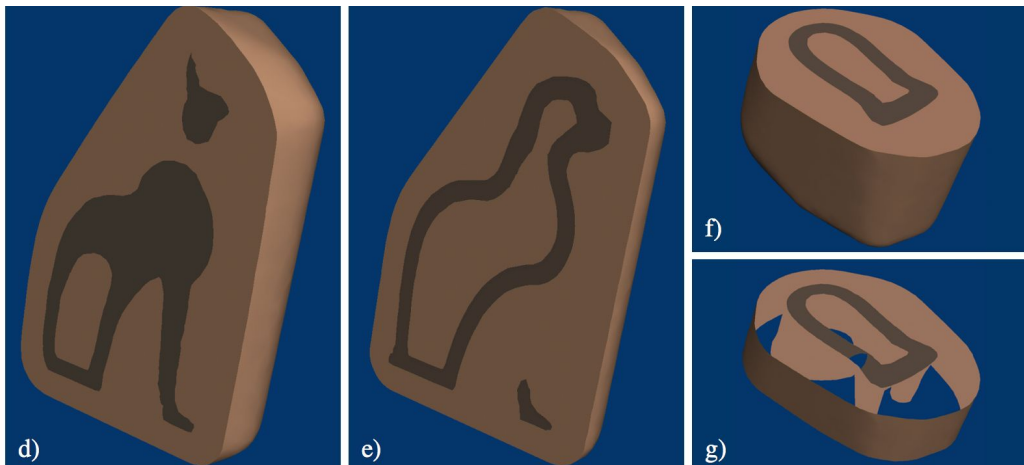
- *Solution: Render edges*
- *Solution: Render tetrahedral faces, shrinking each vertex inward by ~5%*



Volumetric Exploration

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

- *Solution: Use a cutting plane*
- *Solution: Only render faces separating different materials*



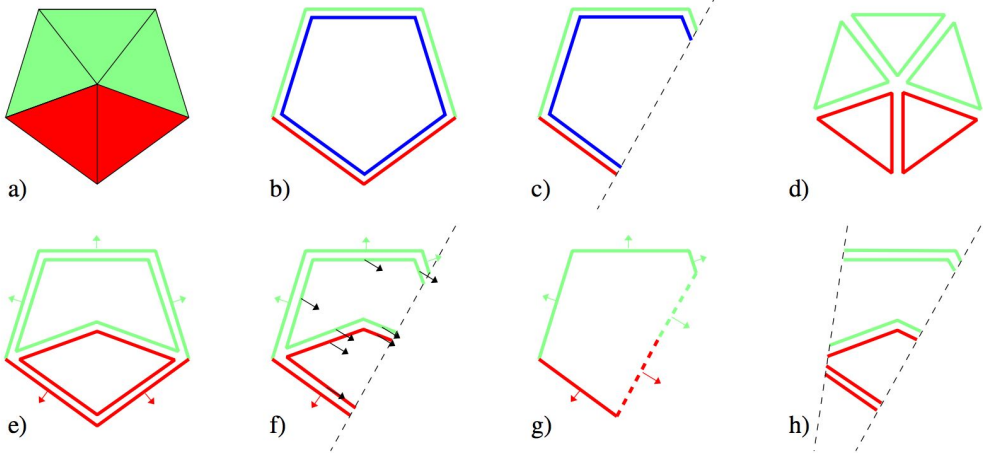
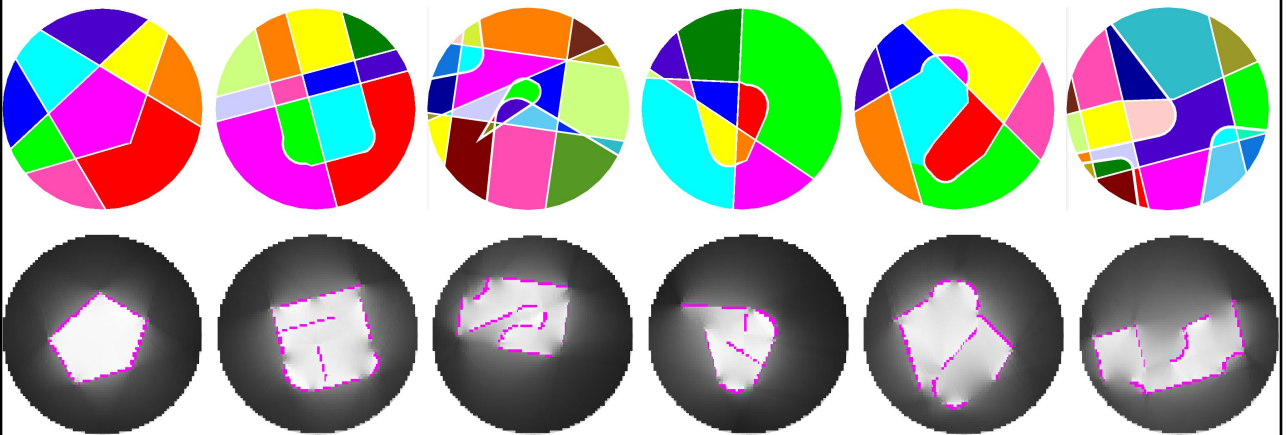


Figure 9.7: Two-dimensional illustrations of the different tetrahedral rendering options applied to the simple model shown in a). Two sided lighting is used to b) color the back faces of surface triangles blue, which can be viewed by c) cutting the object with a clipping plane. The individual elements 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.

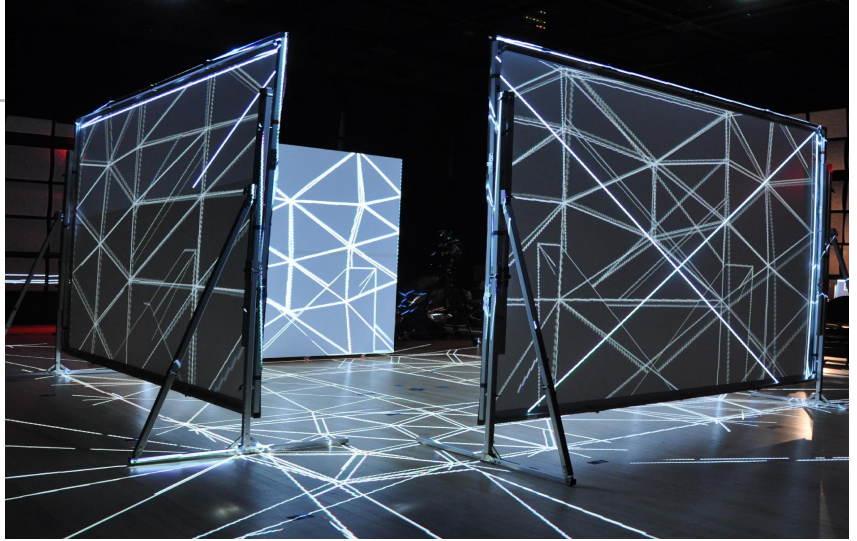
To Understand Relationships and Thresholds



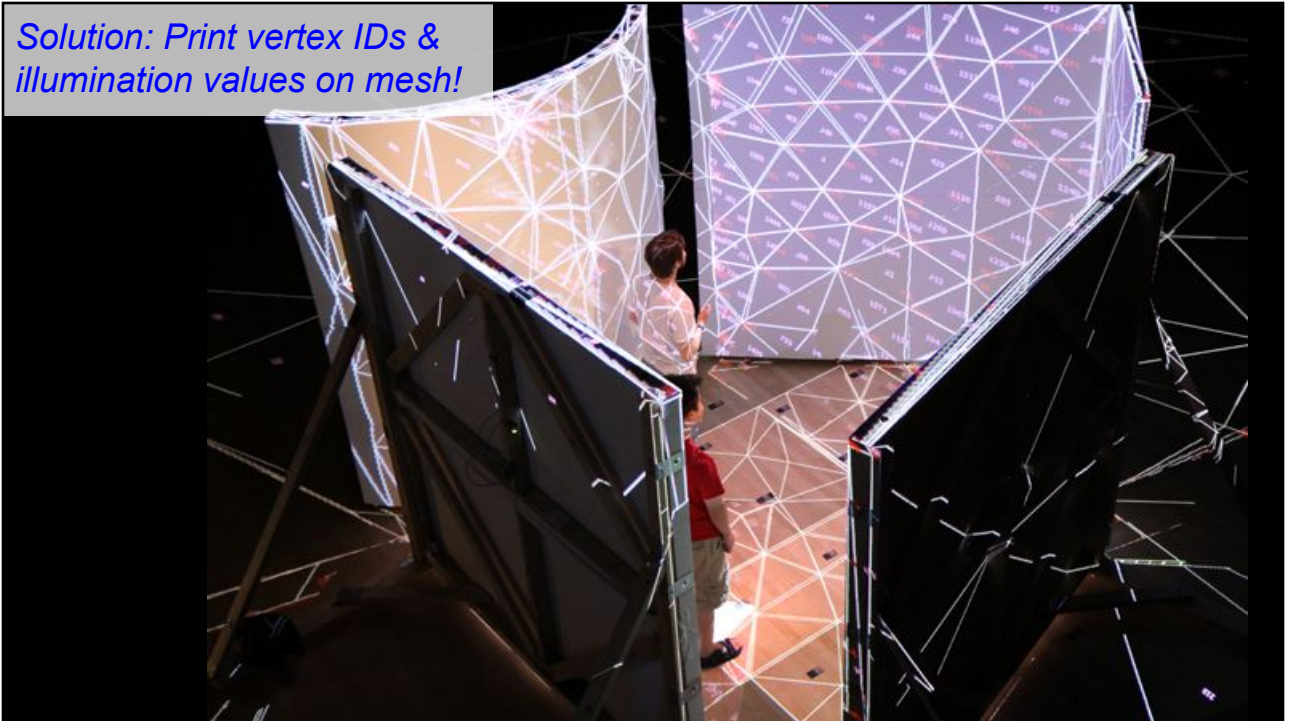
- *Solution: Visualize each region with a unique color*
- *Solution: Discretize and visualize enclosure of each pixel*

Calibration

- 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

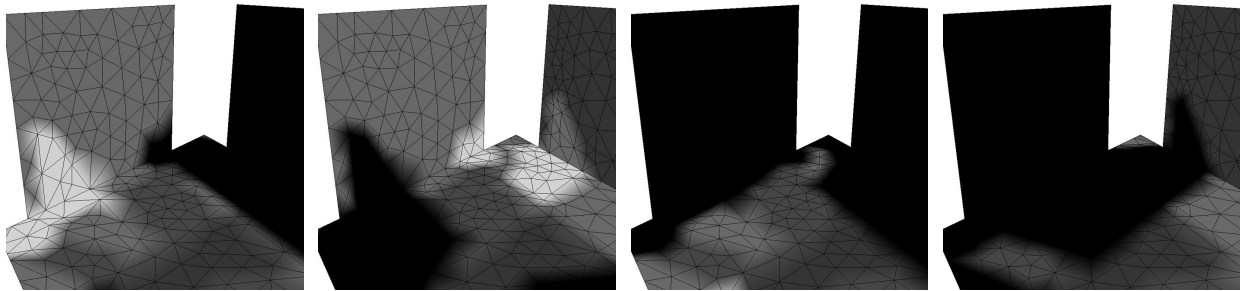
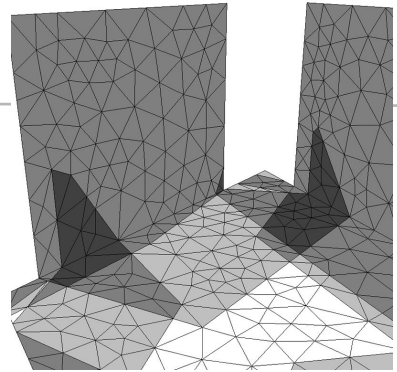


Solution: Print vertex IDs & illumination values on mesh!



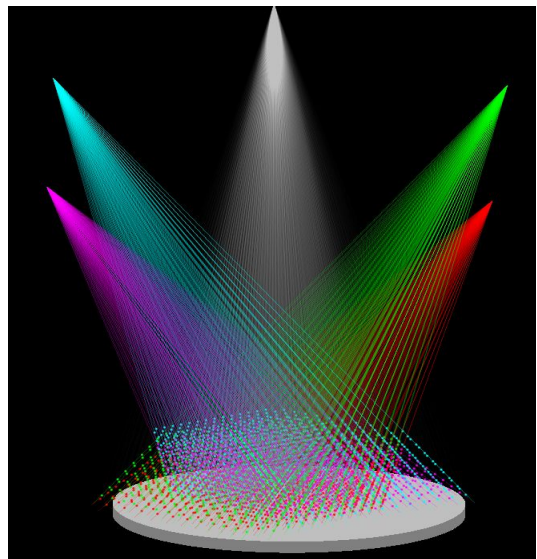
Visibility & Smooth Projection

- Occlusions & Projector Visibility
- Fade in/fade out for transitions
- Sum of projectors = 1
- *Solution: Visualize*
 - # of projectors seen by each patch
 - 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
and rays in a common
coordinate system*

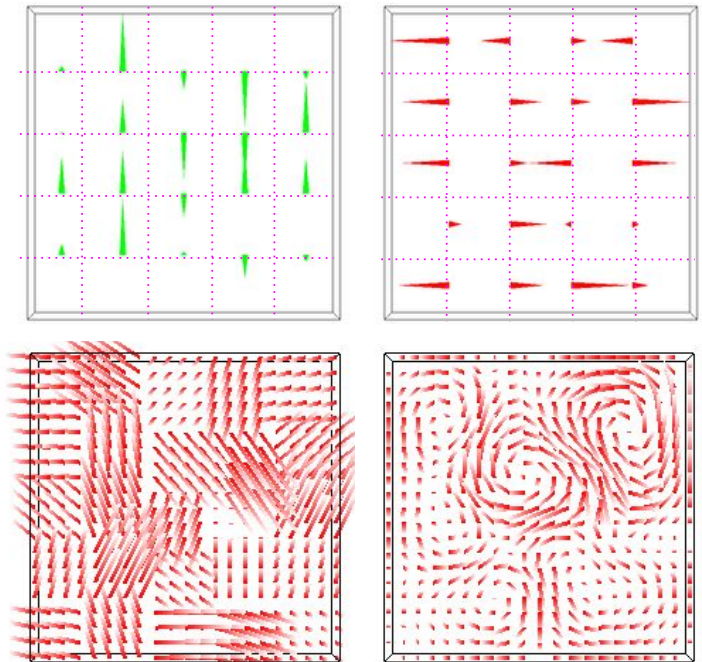




When debugging requires a reboot AND your output devices are all tied to projectors you may be running around to find where important error codes might be displayed...

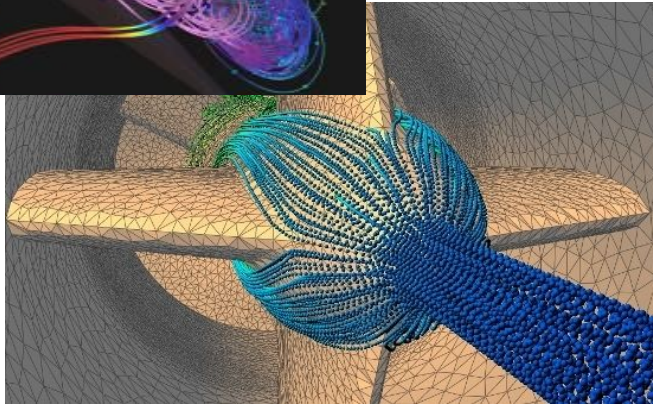
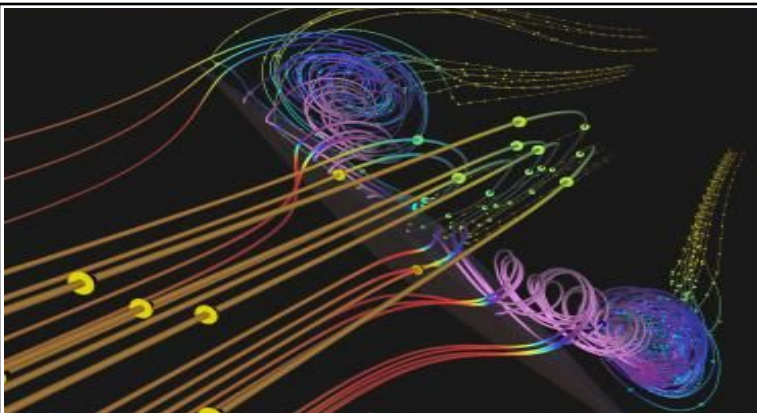
Fluid Dynamics

- Problem: Understanding data structure used by Navier Stokes algorithm
- *Solution: Visualize vertical & horizontal grid cell face velocities*
- Problem: Offset grids of Vertical & horizontal grids
Are interpolation headache
- *Solution: Densely visualize interpolated vector field (it should be smooth!)*



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

Solution: Trace streamlines through the vector field and search for any non-intuitive / unexpected behavior



http://tobiasrick.org/research_particles.html

Instructor view

Student view

USERNAME	PERFECT	LOWEST A-	LOWEST B-	LOWEST C-
FIRST	approximate	approximate	approximate	
LAST				
OVERALL	75.52	100.00	83.90	73.33
FINAL GRADE	B			
LAB %	14.11	15.00	13.50	12.00
HOMEWORK %	31.31	35.00	31.70	27.59
TEST %	16.10	30.00	22.30	19.21
EXAM %	14.00	20.00	16.40	14.53
Lab 1	3.0	3.0	2.7	2.4
Lab 2	3.0	3.0	2.7	2.4
Lab 3	3.0	3.0	2.7	2.4
Lab 4	2.0	3.0	2.7	2.4
Lab 5	3.0	3.0	2.7	2.4
Lab 6	2.0	3.0	2.7	2.4
Lab 7	3.0	3.0	2.7	2.4
Lab 8	3.0	3.0	2.7	2.4
Lab 9	3.0	3.0	2.7	2.4
Lab 10	2.5	3.0	2.7	2.4
Lab 11	3.0	3.0	2.7	2.4
Lab 12	3.0	3.0	2.7	2.4
Lab 13	3.0	3.0	2.7	2.4
Lab 14	3.0	3.0	2.7	2.4
Homework 1	47.0	50.0	47.0	45.5
Homework 2	42.0	50.0	46.5	42.0
Homework 3	47.0*	50.0	45.0	34.0
Homework 4	44.5	48.0	46.0	42.5
Homework 5	39.0	50.0	41.0	35.5
Homework 6	27.0*	50.0	40.0	27.0
Homework 7	53.0	50.0	49.0	47.0
Homework 8	48.5	50.0	46.0	40.0
Homework 9	30.5*	50.0	41.0	30.5
Homework 10	48.0	50.0	49.5	48.0
Homework 11 Extra Credit	19.0			
Test 1	57.0	100.0	76.0	66.0
Test 2	52.0	100.0	74.5	64.0
Test 3	34.0	100.0	63.0	38.0
Exam 1	105.0	150.0	125.0	109.0

USERNAME	PERFECT	LOWEST A-	LOWEST B-	LOWEST C-
FIRST	approximate	approximate	approximate	
LAST				
OVERALL	75.52	100.00	83.90	73.33
FINAL GRADE	B			
LAB %	14.11	15.00	13.50	12.00
HOMEWORK %	31.31	35.00	31.70	27.59
TEST %	16.10	30.00	22.30	19.21
EXAM %	14.00	20.00	16.40	14.53
Lab 1	3.0	3.0	2.7	2.4
Lab 2	3.0	3.0	2.7	2.4
Lab 3	3.0	3.0	2.7	2.4
Lab 4	2.0	3.0	2.7	2.4
Lab 5	3.0	3.0	2.7	2.4
Lab 6	2.0	3.0	2.7	2.4
Lab 7	3.0	3.0	2.7	2.4
Lab 8	3.0	3.0	2.7	2.4
Lab 9	3.0	3.0	2.7	2.4
Lab 10	2.5	3.0	2.7	2.4
Lab 11	3.0	3.0	2.7	2.4
Lab 12	3.0	3.0	2.7	2.4
Lab 13	3.0	3.0	2.7	2.4
Lab 14	3.0	3.0	2.7	2.4
Homework 1	47.0	50.0	47.0	45.5
Homework 2	42.0	50.0	46.5	42.0
Homework 3	47.0*	50.0	45.0	34.0
Homework 4	44.5	48.0	46.0	42.5
Homework 5	39.0	50.0	41.0	35.5
Homework 6	27.0*	50.0	40.0	27.0
Homework 7	53.0	50.0	49.0	47.0
Homework 8	48.5	50.0	46.0	40.0
Homework 9	30.5*	50.0	41.0	30.5
Homework 10	48.0	50.0	49.5	48.0
Homework 11 Extra Credit	19.0			
Test 1	57.0	100.0	76.0	66.0
Test 2	52.0	100.0	74.5	64.0
Test 3	34.0	100.0	63.0	38.0
Exam 1	105.0	150.0	125.0	109.0

Instructor view

Student view

- Identify outliers
- Set appropriate curve for each assignment
- Look for patterns over time
- Notice individual student's strengths & weaknesses
- Identify fair final grade boundaries

USERNAME	PERFECT	LOWEST A-	LOWEST B-	LOWEST C-
FIRST		approximate	approximate	approximate
LAST		approximate	approximate	approximate
OVERALL	75.52	100.00	83.90	73.33
FINAL GRADE	B			
LAB %	14.11	15.00	13.50	12.00
HOMEWORK %	31.31	35.00	31.70	27.59
TEST %	16.10	30.00	22.30	19.21
EXAM %	14.00	20.00	16.40	14.53
Lab 1	3.0	3.0	2.7	2.4
Lab 2	3.0	3.0	2.7	2.4
Lab 3	3.0	3.0	2.7	2.4
Lab 4	2.0	3.0	2.7	2.4
Lab 5	3.0	3.0	2.7	2.4
Lab 6	2.0	3.0	2.7	2.4
Lab 7	3.0	3.0	2.7	2.4
Lab 8	3.0	3.0	2.7	2.4
Lab 9	3.0	3.0	2.7	2.4
Lab 10	2.5	3.0	2.7	2.4
Lab 11	3.0	3.0	2.7	2.4
Lab 12	3.0	3.0	2.7	2.4
Lab 13	3.0	3.0	2.7	2.4
Lab 14	3.0	3.0	2.7	2.4
Homework 1	47.0	50.0	47.0	45.5
Homework 2	42.0	50.0	46.5	42.0
Homework 3	47.0*	50.0	45.0	34.0
Homework 4	44.5	48.0	46.0	42.5
Homework 5	39.0	50.0	41.0	35.5
Homework 6	27.0*	50.0	40.0	27.0
Homework 7	53.0	50.0	49.0	47.0
Homework 8	48.5	50.0	46.0	40.5
Homework 9	30.5*	50.0	41.0	30.5
Homework 10	48.0	50.0	49.5	48.0
Homework 11 Extra Credit	19.0			
Test 1	57.0	100.0	76.0	66.0
Test 2	52.0	100.0	74.5	64.0
Test 3	34.0	100.0	63.0	51.5
Exam 1	105.0	150.0	123.0	109.0

Today's Takeaway Message:

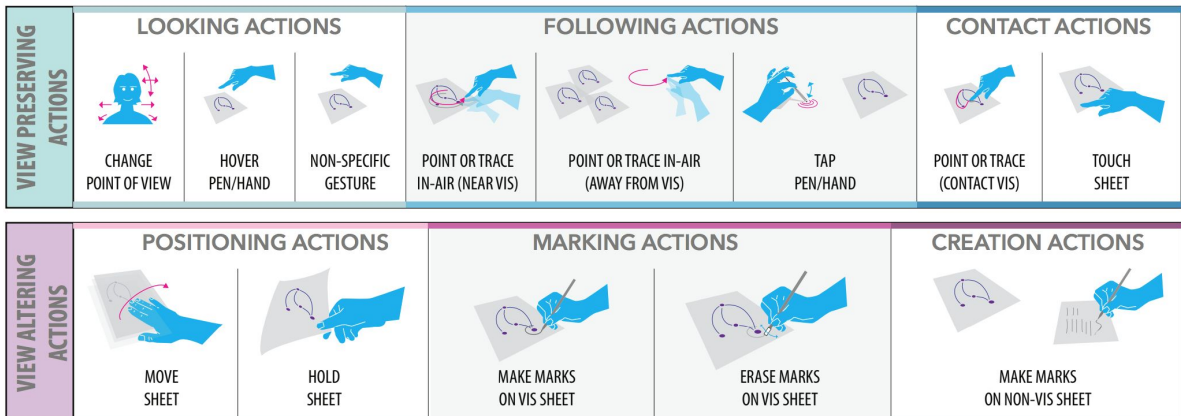
- Invest your time & energy in Visualization for Debugging
- Make Intermediate Visualizations
Even if you don't think they'll go in your final report
- Take screenshots of "Bloopers" as you work
They may be helpful for explaining the challenges and how you overcame them!

Today

- Final Project Proposals & Accessing Academic Papers
- Visualization for Debugging
 - *Not*: Traditional Debugging
 - *Not*: Software Development Dashboards
 - Lots of helpful examples from class & research!
- Today's Readings:
 - "Active Reading of Visualizations", Walny, Huron, Perin, Wun, Pusch, and Carpendale, IEEE InfoVis 2017
 - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Friday

Reading for Today

- "Active Reading of Visualizations", Walny, Huron, Perin, Wun, Pusch, and Carpendale, IEEE InfoVis 2017



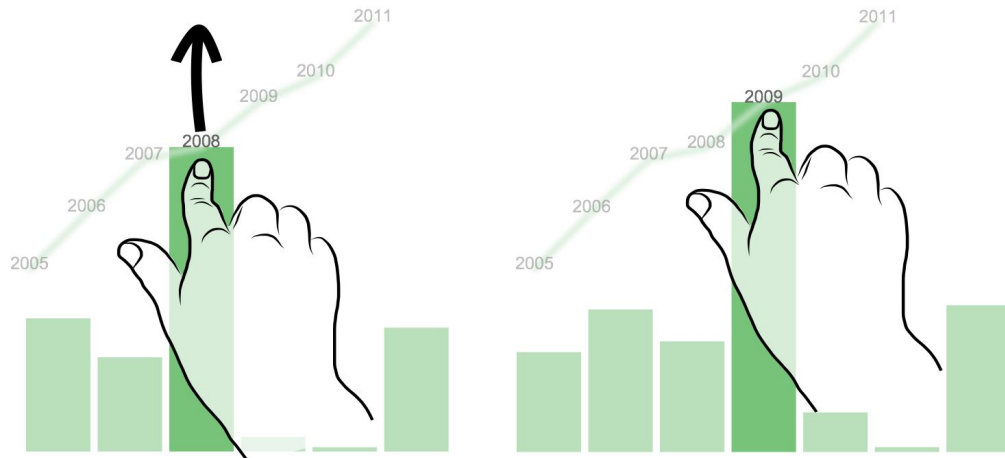
- Parallels to research in active reading of text
- Pen & paper annotations easier (more fluid) than digital (have to learn option menus)
- Pilot study to explore qualitative use + second study for precision
- Action types:
 - View preserving (looking/following/contact)
 - View altering (positioning/marketing/creation)
 - Goals: recognizing, tracking, reorganizing, decoding, and analyzing
- Vary graph size -- people use different techniques with larger graphs
- Recommendations for interactive visualization design (supporting active reading) & further studies

Today

- Final Project Proposals & Accessing Academic Papers
- Visualization for Debugging
 - *Not:* Traditional Debugging
 - *Not:* Software Development Dashboards
 - Lots of helpful examples from class & research!
- **Today's Readings:**
 - "Active Reading of Visualizations", Walny, Huron, Perin, Wun, Pusch, and Carpendale, IEEE InfoVis 2017
 - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Friday

Reading for Today

- “DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation”, Kondo and Collins, IEEE Visualization 2014



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

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

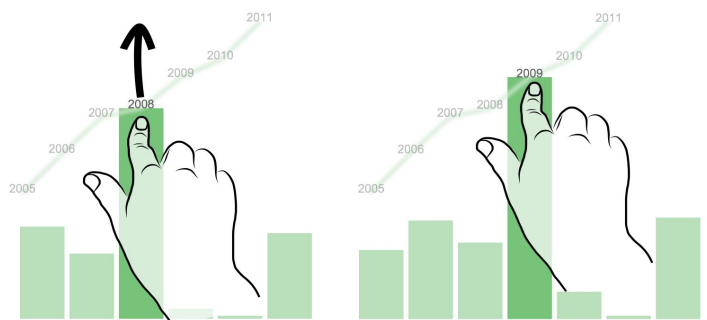


Fig. 4. Navigation in time for bar charts is achieved by dragging a selected bar vertically along its hint path. The hint path slides horizontally to stay connected with the bar and finger.

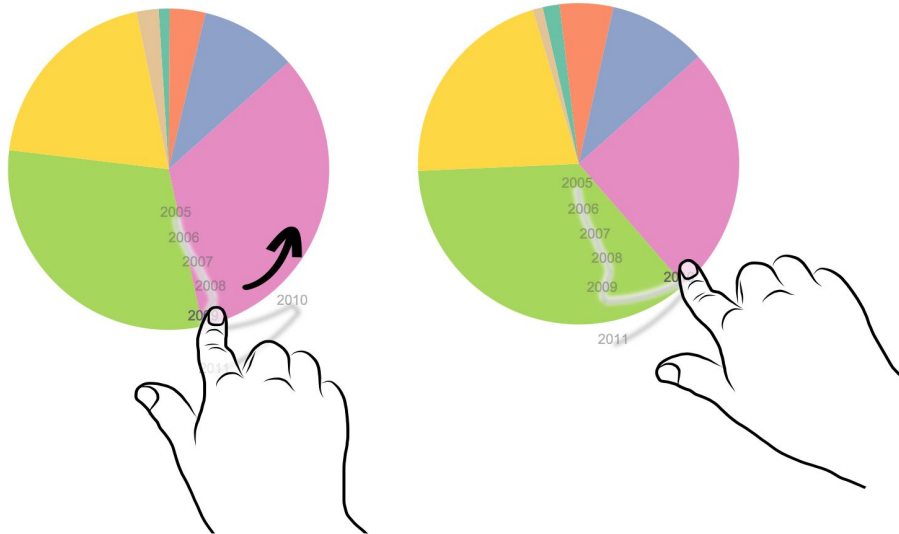


Fig. 6. Pie charts are adjusted in time by dragging an edge of a segment in angular directions along its hint path. The hint path slides in and out along the radius to remain connected with the finger position and edge of the segment.

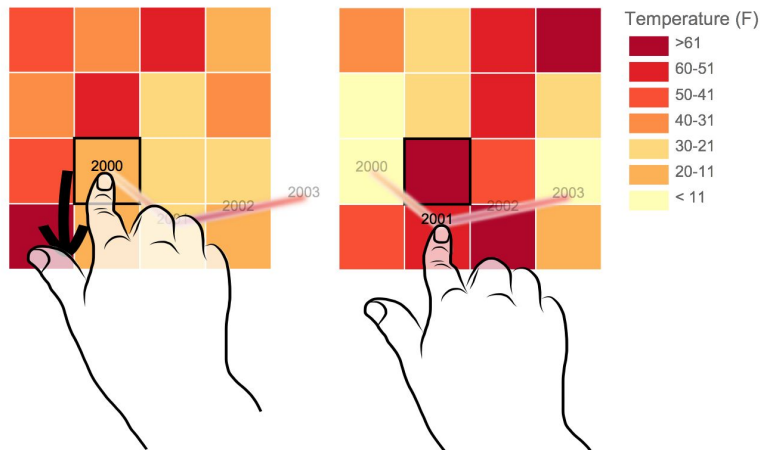


Fig. 7. Heat maps are adjusted in time by dragging vertically in the space of the colour scale. The hint path slides horizontally to stay connected with the cell and finger.

But is this an 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.

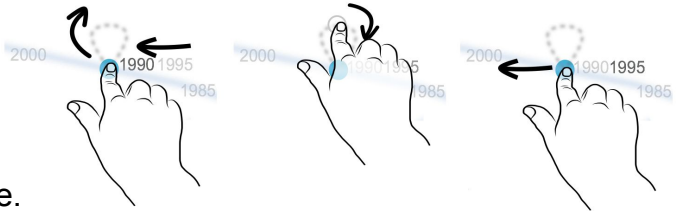


Fig. 3. Temporal ambiguity occurs in a scatter plot when a point does not move between time steps. Temporal navigation capabilities are provided in ambiguous using loops in which one transit around the loop corresponds to one time step.

- *Nonintuitive?*
- *Does this work if the path has lots of overlaps/loops?*

- 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



<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?
Because 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 probabilistically 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?

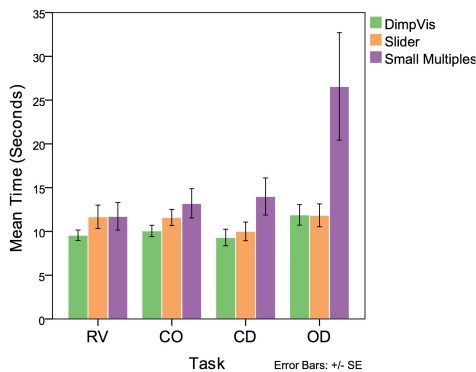


Fig. 8. Scatter plot task completion times.

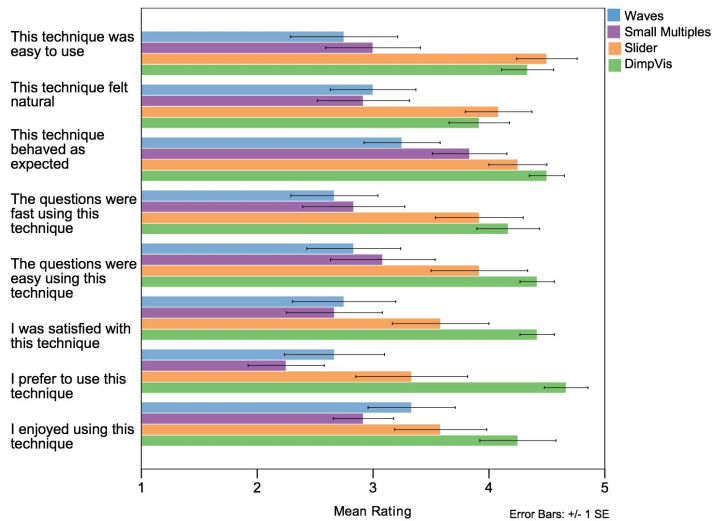


Fig. 9. Scatter plot subjective ratings.

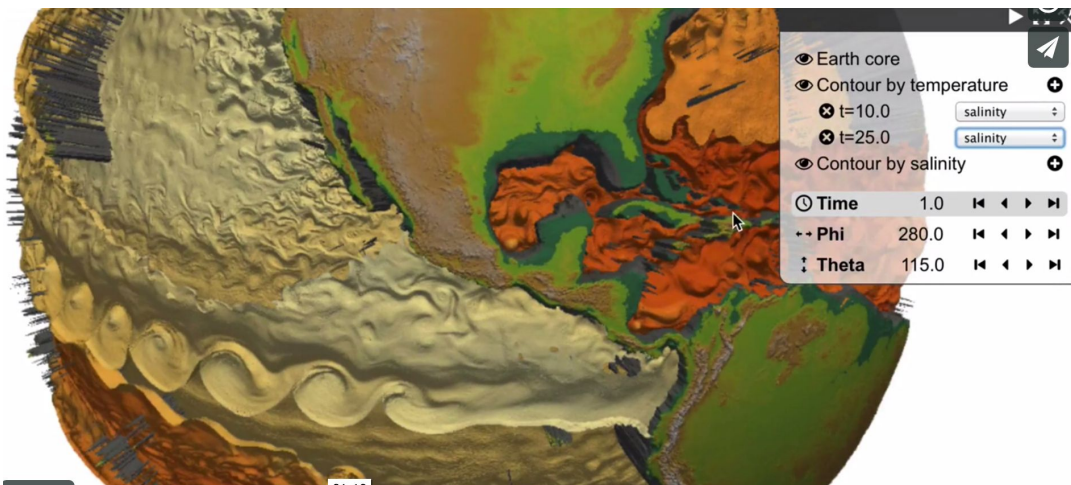
- “Enhanced engagement with data”
- Some misuse of tool (intuition/prior experience w/ other interfaces)
 - Attempted 2 finger interaction
 - Attempted to drag in opposite direction vertically
- Open Questions
 - Scalability
- Great job removing outside factors from what was being tested
- Clearly listed 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

- Final Project Proposals & Accessing Academic Papers
- Visualization for Debugging
 - *Not:* Traditional Debugging
 - *Not:* Software Development Dashboards
 - Lots of helpful examples from class & research!
- Today's Readings:
 - "Active Reading of Visualizations", Walny, Huron, Perin, Wun, Pusch, and Carpendale, IEEE InfoVis 2017
 - "DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation", Kondo and Collins, IEEE Visualization 2014
- Readings for Friday

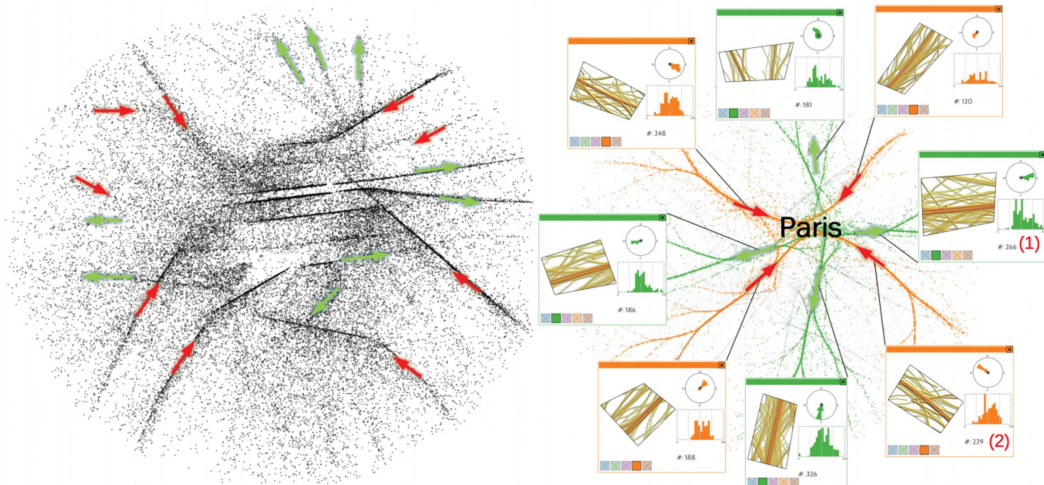
Reading for Friday *pick one*

- "An Image-based Approach to Extreme Scale In Situ Visualization and Analysis", Ahrens, Patchett, Jourdain, Rogers, O'Leary, & Petersen, Supercomputing 2014



Reading for Friday *pick one*

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



Reading for Friday *pick one*

- “Learning Patterns of Activity
Using Real-Time Tracking”,
Stauffer & Grimson, IEEE PAMI 2000

