Guaranteed Frame Rate Drawing for Interactive Navigation of Large Graphs

Tamara Munzner

Stanford University

outline

- motivation
- video
- details
- future directions

why bother drawing graphs?

- simple, powerful and elegant
- small graphs common:
  - figures in books and papers
- big graphs rare:
  - why?

- inadequate tools
  - computational complexity
    - dozens or hundreds, not thousands or millions
  - visual clutter

different tasks:

explanatory vs exploratory

small

illustrate high level or toy example

big

can't explain until after you've explored!

explanatory vs exploratory

small:
- show detailed structure

big:
- overview of structure
- find interesting places
- context of part in whole
- structure as index
- structure as scaffolding to show attributes
  - static, dynamic

fluid interactive experience

- guaranteed frame rate
- adaptive algorithm
- large graphs on high end machines
- medium graphs on low end machines

video

layout

drawing

layout

- appropriate spanning tree
  - link filtering
  - fast layout
- nontree links for given nodes drawn on demand

draw loop until allotted time gone

use node at top of queue

draw: graph structure
  draw nodes, links for incoming, outgoing

add candidate nodes: tree structure
  enqueue parent, children

sort candidate nodes: viewpoint
  projected screen area

base case
  frame to frame coherency
  use largest node from previous frame

modes

- active draw
  - mouse drags, animated transitions
- pick
  - passive mouse motion
  - locate highlight
    - draw node in front buffer
  - use sorted drew queue
- idle

idle

- fill in fringe
- bite-sized pieces
  - multiple calls to draw routine
  - use previous leftover candidate list
  - check for active draw or pick between calls
- limit on CPU usage
- well-mannered part of ecology of tools

ideal:
  O(visible) not O(total)

actual:
  O(visible) common case

  remapping global not local

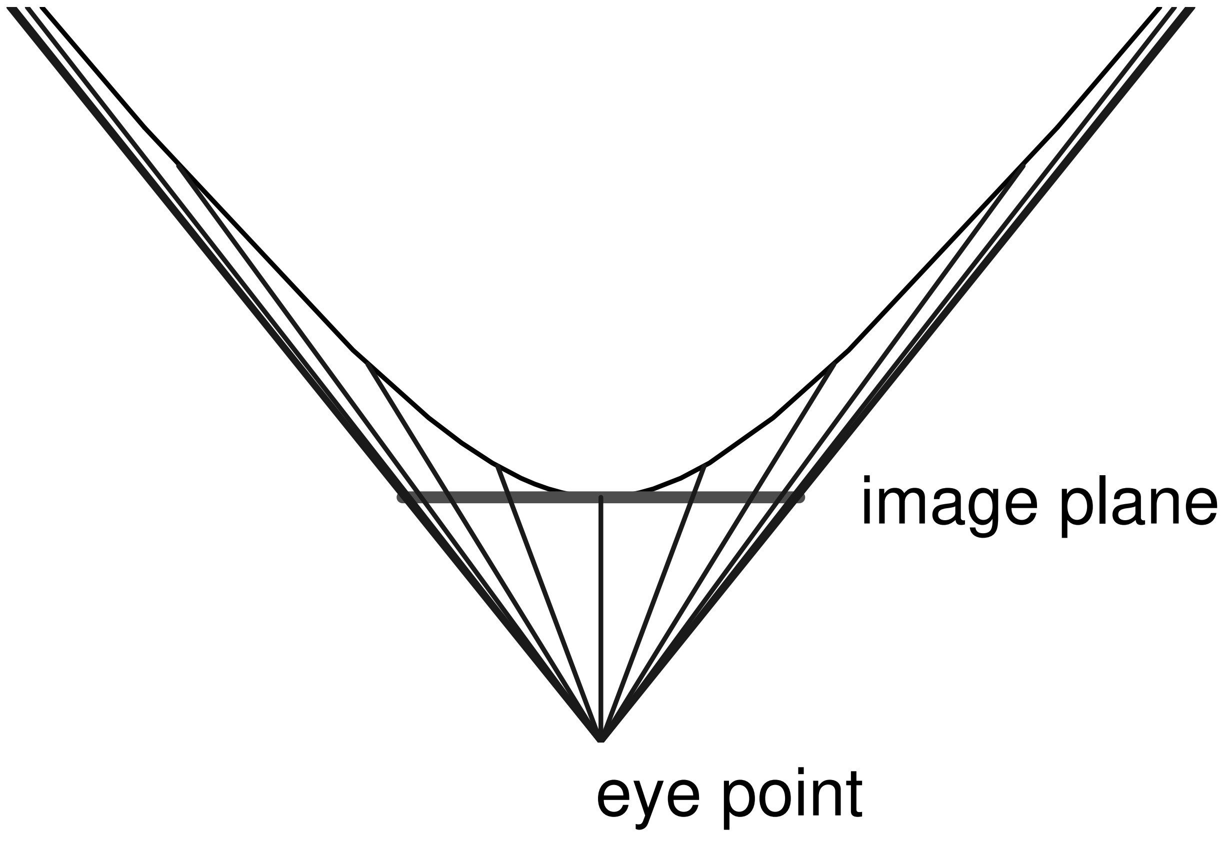
    seconds for >100K edges

- hyperbolic space
  - layout and navigation
  - exponential amount of room
    - place to put nodes
    - specifics in InfoVis '97
  - mapping from hyperbolic to euclidean space
    - viewpoint external to entire space
    - details for large neighborhood
      - focus + context view

hyperbola to plane

- move infinitely far on hyperbola
- planar projection to finite area

- limited precision



possible solutions

- incremental mapping:
  - drawing O(visible)
- incremental layout
  - layout O(visible)

current layout

- fast
- global
- limited by main memory
- 100,000 nodes with 1 GB
- (unoptimized space usage)

conclusion

- interactive system for layout and drawing
- fast
  - two orders of magnitude size improvement
- fluid user interaction
  - (almost) guaranteed frame rate

acknowledgements

Stanford
Pat Hanrahan, Francois Guimbretiere

SGI
Alan Braverman, Greg Ferguson, Ken Kershner

funding
NSF, ARPA, SGI

software

H3 layout
H3Viewer drawing

black box with straightforward API
free noncommercial release soon?

SGI Site Manager
free software for web management

http://www.sgi.com/Products/software/sitemgr.html

further information

http://graphics.stanford.edu/papers/h3

http://graphics.stanford.edu/~munzner