

Tuning SDAZ to Address Overshooting and Course-correction Problems

Thomas Dang, dqluan@gmail.com

Domain: map navigation and other cartography-related applications

Tasks: find a target on a map, comparing two points (multiscale comparison)

Dataset: map and satellite photo

Other applications: stock chart (semantic zooming!), scatterplot

Why Zooming instead of...

- Overview-detailed Views:
 - small screens
 - scale factor between global and detailed views not too great
 - number/complexity of object to find/compare not too great (Ware, and Plumlee 2006)
- Focus-context in the same view:
 - Spatial distortion aggravated by small screen

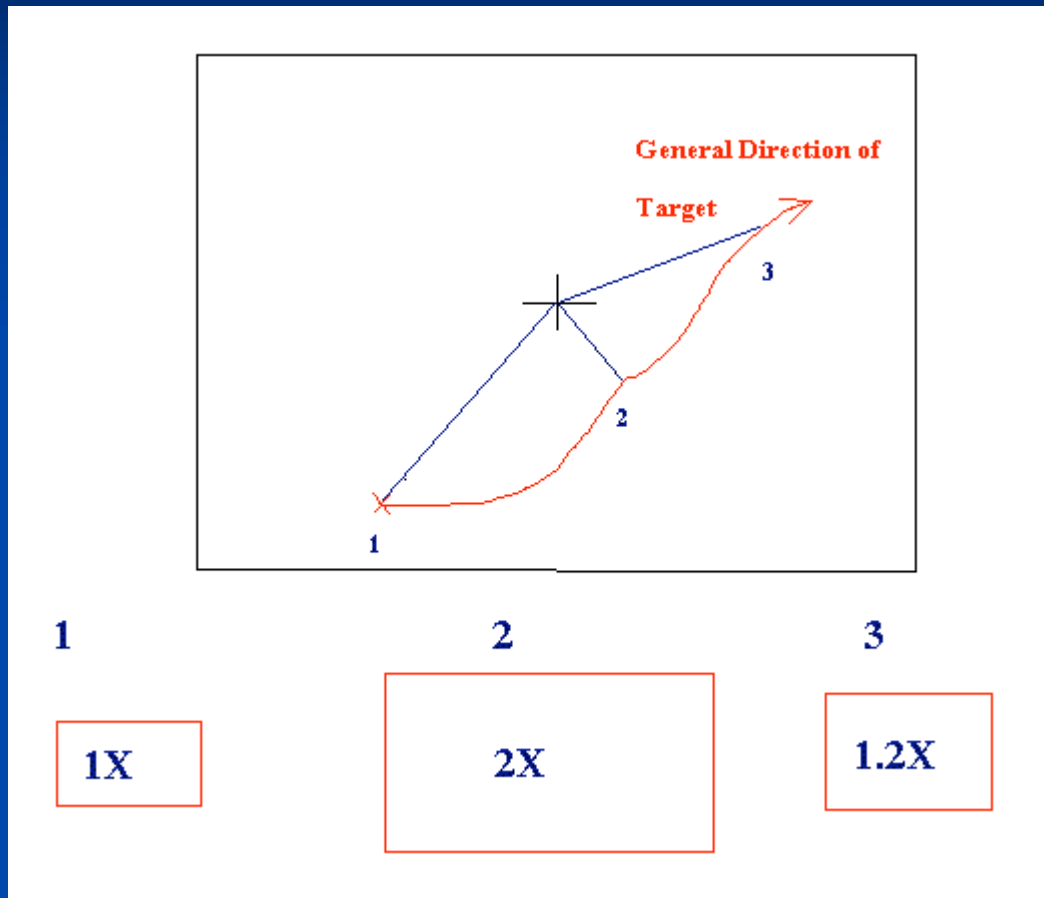
Previous works on Automatic Zooming

- SDAZ - Igarashi and Hinckley (2000)
- 3D Point of Interest - Mackinlay, Card, and Robertson (1990)
 - Slower velocities in the latter stage of object approach -> influenced my zoom-in solution
- Depth Modulated Flying – Ware, and Fleet (1997)
- Bi-manual Zoom/Pan - Bourgeois, and Guiard (2002)
- OrthoZoom Scroller - Appert, and Fekete (2006) -> 1D only, not for 2D navigation!

The Problem With SDAZ

- Initialized by clicking and holding down mouse
- Zoom-out or in based on distant of mouse cursor to screen center (an analog for speed)
 - This speed changes when approaching the target due to limit dexterity
 - The constantly changing zoom-level is high cognitive load
 - The objects changing (especially in semantic zooming) can overload visual working memory
 - Mouse Speed/Zoom Level change is aggravated when user over shoot the target

Overshooting and Course Correction





Coping Strategies

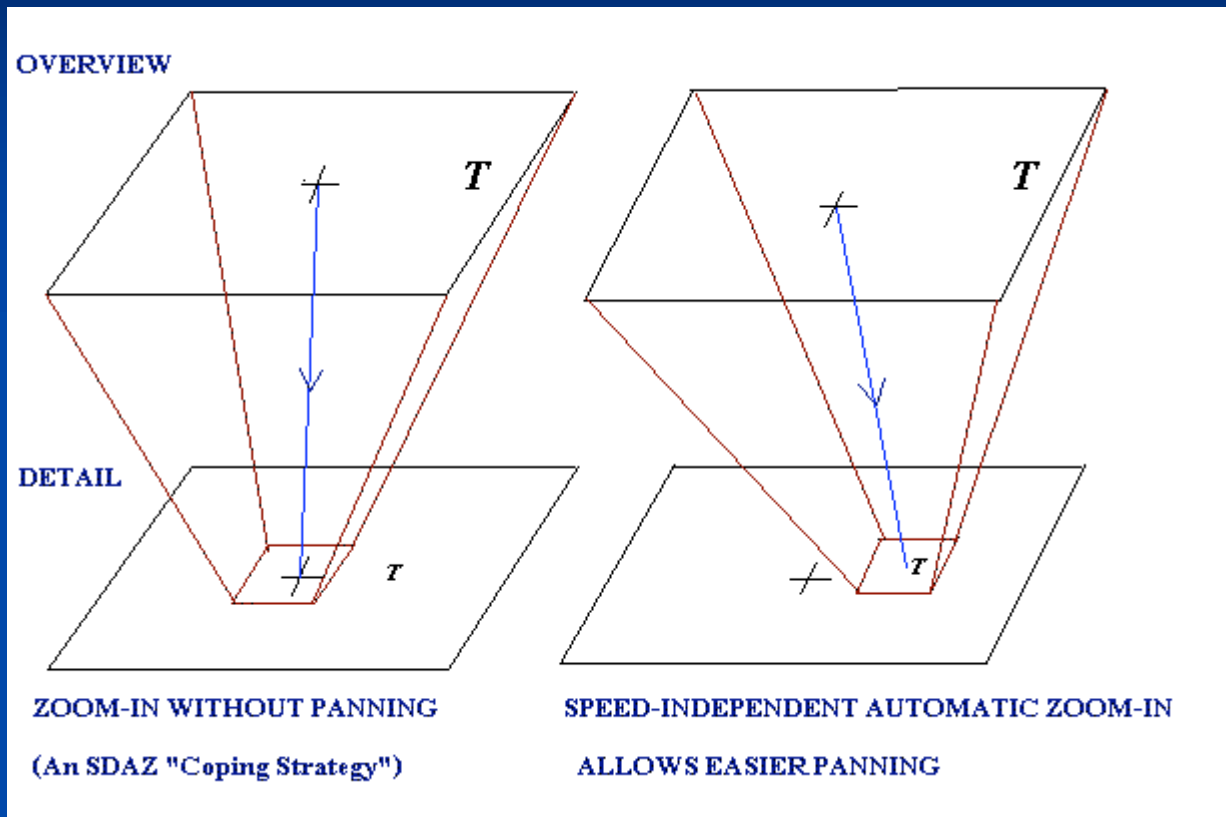
1. Scroll really slow, zoom very little (to keep the motion more accurate)
2. Zoom out all the way really quickly, move to the target, then release mouse.

⇒ When zooming out, can overshoot if zooming too fast. When zoom in, cannot pan because the mouse already released

Hybrid SDAZ – The zoom-out problem

-  Intention of Zooming: Sample only 0.5-1sec of initial mouse motion
-  At the target Zoom-level, switch to rate-based scrolling
 - Rate-based scrolling in detailed view causes exceedingly high visual flow
 - But hypothetically much less of a problem after the initial zooming
 - No more changing zoom-levels due to accidental overshoot or hamhandedness!











Hybrid SDAZ – the zoom-in problem



1. Zoom-in triggered by clicking and holding down mouse again
2. Zoom-in is speed-independent – Zoom-in speed scaled by a constant fraction of distance to the detailed view.

3. Panning while zooming is easier since the zoom level does not change when you pan! => less cognitive load and dexterity required

Project Milestones and Progress

-  Background research (Nov 5)
-  Java/J-OPENGL Project Setup (Nov 14)
 -  Basic keyboard-controlled zooming prototype
 -  Logging of zoom level, x-y-z world coordinates and mouse states: screen xy, buttons, distance to center
-  Acquire the data set from Google (Nov 15)
-  Implement SDAZ demo with mouse input (Nov 22)
-  Implement Hybrid SDAZ (Dec 2)
-  Implement user testing facilities and cases (Dec 5)
-  Debugging and Uploading Demo (Dec 7-8)
-  Final Report and Presentation (Dec 14-16)

Q&A

Thomas Dang, dqluan@gmail.com