Information Visualization and Knowledge Management: Context & Detail

Dr. Margaret-Anne Storey
Dept of Computer Science, UVic

Different Techniques

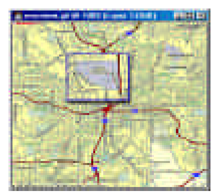
- Scrolling
- Context + detail displays
- Zooming techniques
- Distortion to provide integrated focus + context displays

Customization vs. Adaption

- Customization requires knowing ahead of time what the user wants with respect to navigation
- Adaption requires a log of the user's actions to determine his/her interests in certain parts of the information
- Consequently a distance function is required which makes it possible to evaluate the user's interests in all pieces of information depending on the user interaction

Context + Detail

- Overview windows (see mapexplorer)
  - Direct interaction for panning and zooming
  - Coordinated views
  - Combined overviews based on spatial layout and abstract structures (SHriMP)
  - "Field of view" or "panner", can be dragged around, note: views have to closely coupled
- Zoom and replace windows (e.g. Office applications, acrobat)
- Magnifying glass
- Inset (maps, charts)



Multiple versus Single views

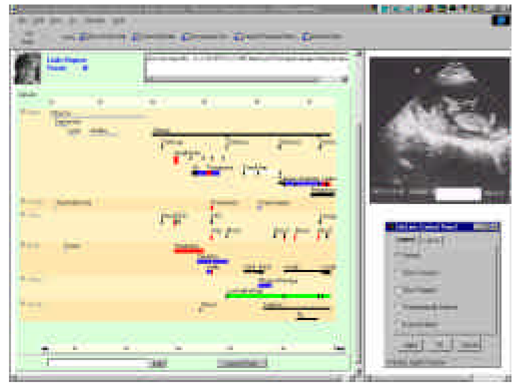
- Both techniques can be used to provide access to context + detail
- Requires significant effort for the "cognitive reintegration" of multiple views
  - Studies from psychology do indicate that humans integrate information in a single view much more easily than from multiple views
- Static overviews provide details on items that are *a priori* important to everyone, but they can't be customized to a single user

More sophisticated context + detail techniques

- Lifelines
- PDQ Tree viewer
- Elastic windows
- ZUIs
- Fisheye views

Lifelines

- Personal histories: court records, medical histories, …
- One scale overview – shows multiple facets of the records
- *Facets* shown as regions of the screen, distinguished by background
- *Aspects* are displayed as individual time lines (e.g. of an aspect – medical condition)
- *Icons* indicate discrete events
- Thickness/colour of lines to encode relationships (for example to show the severity of the crime, or use colour to highlight recent information)
- (Video -- http://www.open-video.org/segment.php?seg_id=700)



Lifelines (2)

- Approach:
  - Based on graphical time scales – natural ordering encodes time
    - Gantt, PERT charts
- Features:
  - Can click on lines to get more information
  - Overview – uses screen real estate effectively
    - Filter labels, cluster results, show node labels on a mouse-over, can select events
  - Colour can be used across facets to link that relationship with all the facets
  - Facets can be expanded/contracted, can rearrange columns and regions
  - Multiple focal areas can we used with split sliders
  - Rescaling – zooming in using sliders (or can show a sub-timeline in a detail window)

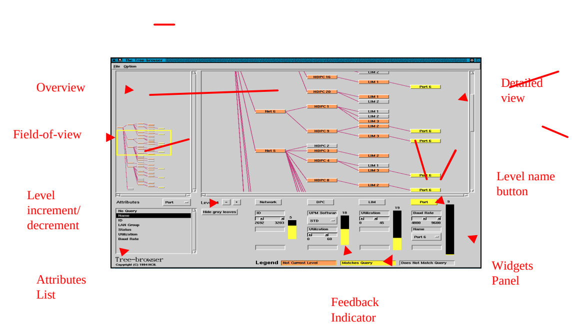
Lifelines (3)

- Strengths:
  - Help detect anomalies and patterns (look for cause and effect)
    - Easier than dealing with large text documents which could lead to missing links/effects
  - Helps avoid missing information
  - Provides easier access to details
  - Customizable
- Disadvantages:
  - Layout issues, unused space
  - Limited to case-variables information, or relational data

PDQ Tree Browser

- Pruning with dynamic queries
- Trees in 2 tightly coupled views
  - One is an overview
  - The other is detailed view
- Helping to make decisions by choosing what we *don't want*
- Iterative refinement/progressive querying
- Hierarchical data sets makes this easier
- Can do dynamic queries to filter and prune

The PDQ Tree-Browser Interface



PDQ- Dynamic Query Environment

- Homefinder and Filmfinder – widgets are hard coded into the user interface for doing querying
- PDQ allows the user to customize the query panels based on their current interests
- User selects which attributes they are interested in, widgets are created at run-time (therefore it is not tied to any particular application)
- Updates happen very smoothly (due to buffering of results), nodes matching the queries are highlighted
- When you select a node, the view zooms to that level
- Limitations:
  - Fine tuned to a depth of 5 (can it be extended to arbitrary number of levels)
  - Only 'AND' queries
  - Not a compact layout tree used (this could be fixed)

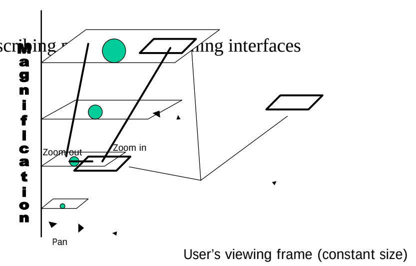
Zooming windows

- ZUIs (Zooming user interfaces)
  - Pad++/Jazz
  - Issues:
    - Panning – does the camera or the scene move?
    - Zooming in/out (speed, interaction issues)
  - Features:
    - Sticky objects
    - Lenses
    - Semantic zooming
    - Portals

http://www.cs.umd.edu/hcil/jazz/play/hinote1.1/hinoteapplet.html

Space-Scale Diagram

Describing ing interfaces



Furnas and Bederson '95

Slide adapted from John Stasko

Elastic windows

- Elastic windows
  - Grow and shrink to fill up available space (don't overlap)
  - Video: http://www.open-video.org/segment.php?seg_id=705

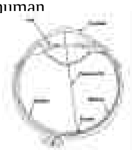
Macroscope

- Transparencies can be used to avoid occlusion and to show views of different scales at the same time, e.g. semi-transparent menus, macroscope (Lieberman et al.)
- Layers at different levels of details are overlaid
  - but too many layers can lead to visual clutter
  - hard to predict how the multiple layers will visually interact
- Video: http://web.media.mit.edu/~lieber/Lieberary/Macroscope/Macroscope.html

Fisheye Views

Why fisheye views?

- Visualizations fit their goal of providing intuition about data best when they conform to the nature of human perception
- How do our eyes perceive visual information?
  - Resolution at the center of our retina (fovea) is very high, enables reading etc.
  - Resolution falls off rapidly, our peripheral vision gives us rough orientation and relates local detail to the general context



- An effective visualization should mimic our visual perception abilities and present parts of the information we want to concentrate on in detail with other information as a simplified abstract form

History of Fisheye views

- Furnas first introduced the concept of Fisheye views in ‘81
  - Based on his observation on how humans comprehend the world and our thinking which is heavily influenced by the spatial distance to locations – how we leave out details systematically
- Our knowledge of things is not static – no matter where we are, certain things will always be important to us
- Besides spatial distance, other kinds of distance dominate our orientation – ability to remember events is clearly influenced by *temporal distance*
- Similarity of concepts is also important, concepts may be directly related to each other or distant in our cognitive model (*conceptual distance)*
- Abstraction (by filtering details) is the key to let us work at different scales

Generalized Fisheye Views

- Degree of Interest (DOI) – assigned to each element of the information space
- DOI comprised of two things:
  - A static component: API (a priori importance – global importance of the element)
  - A dynamic component: relates the importance of elements to the user’s current interest deduced from his or her latest interactions
- An element (node) – selected as the focus point (FP)
- An application-specific distance function “dist” captures the *distance* between any 2 elements in the information space
  - measures the degradation of the user’s interest on elements when the focus is on a particular node (element) -- dist may depend on several factors, e.g. # of inhabitants, importance of certain elements, spatial, temporal aspects etc.

Generalized Fisheyes views (2)

- DOI of some element *x* for a given focus *FP* is:
  $DOI(x) = API(x) – dist(x, FP)$
- Note that when the FP changes we will have to calculate the DOI for every node (but there are techniques to just consider the local influence – e.g. common ancestor in the tree for hierarchical fisheye views)
- But how do we deemphasize elements of minor importance…

Fisheye views of tree structures

- We use negative values for the API(*x*)
  - 0 if at the same level, -1, -2 for more detail
  - We display things only if its DOI is above a certain threshold (*k*), which is under the user’s control
  - dist(*x*,*FP*) -- #links on a path
  - API(*x*) = -dist(*root*, *x*)
  - DOI(*x*) = API(*x*) – dist(*FP*,*x*)
  - For example if the threshold (*k*) = -3, we see the nodes and all its ancestors – zero order fisheye view
  - *k* = -5, first order (all immediate progeny of all direct ancestors are included)
  - *k* = -7 second order fisheye view etc.
  - If we know the branching factor, we can use this to estimate # of nodes that will be visualized – can use number of nodes to help determine *k*
  - Fisheye view is a logarithmic compressor
  - Note the distance function could be a more sophisticated structure

Fisheye views of tree structures (2)

- This approach is not very good for bushy trees or trees with non-constant branching factor
- The DOI for an element increases with global importance and decreases with distance from focal point
- If there is a canonical ordering to the siblings, may be able to filter siblings further away
- Note: standard flat view is really the degenerate case of a fisheye view with API(*x*) = 0
- Other graph structures, i.e. when there are multiple paths connecting nodes…

Applications of Fisheye Views

- Abstract (graphs) and structured data (trees)
- Geographical views, maps
- Furnas applied his techniques to source code views
  - Hierarchical structure based on hierarchical structure in source code
  - Line numbers used to calculate distance
- Other applications:
  - Fisheye views of menus
  - Calendar
  - Shared text editor for CSCW (we will look at a video for this later in the course)

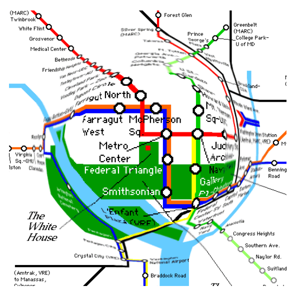
Fisheye views – presentation techniques

- 3 groups of emphasis techniques:
  - Filtering
    - Leave out elements with low DOI
    - Easier if hierarchical structures, can leave out entire substructures
  - Distorting
    - Like a photographic fisheye view lens, we can use size, position, shape and shading
  - Adorning
    - Techniques that emphasize or deemphasize, like colour, line style, transparency

Distorting Fisheye Views

- Distortion of the underlying layout
- Issues: can these views be interpreted correctly?
- Following properties can be crucial for user's comprehension (preserving the user's mental map):
  - *Topology:* the inside of a closed continuous curve should be mapped to the inside of a closed continuous curve
  - *Sequence of Orthogonal relationships*: relative orientation of nodes to each other wrt compass direction needs to be maintained
  - *Membership in clusters*: if nodes are close to each other in the original layout they should remain close in the distorted view
  - *Application specific properties:* for example shape, colour
  - *Temporal aspects of the distortion:* animation frame rate

Example



Slide adapted from John Stasko

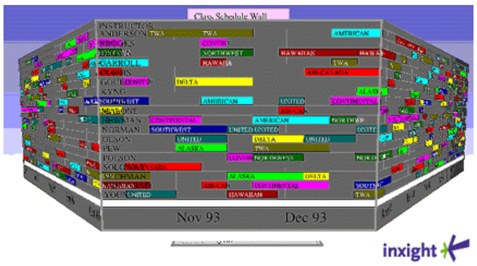
Implicit vs. Explicit Fisheye Views

- Distorted views can also be achieved by using perspective projection of 3D data
- Problems with distortion is that the effects can be hard to predict and to control
- Keahy & Robertson – have an approach where the distortion is explicit rather than an indirect side effect – amount of distortion is explicitly visualized and manipulated
- Advantage with this is that the user can specify which parts of the display not to distort

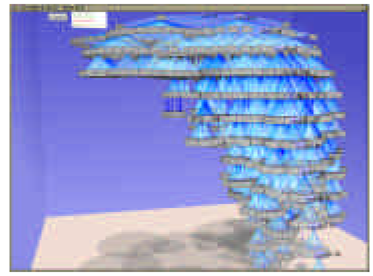
Fisheye Techniques

- Perspective wall
- Cone Trees
- Document Lens
- 3DPS
  - Extending from 2D to 3D
- SHriMP Constrained Fisheye layouts
- Intelligent Zoom
- Fisheye Menus
- Table Lens
- Focus Lens distortions
- Zooming in Application Space

Perspective Wall

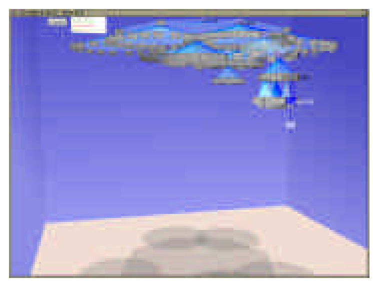


Large Cone Tree



Fisheye Cone Tree: Focus 1

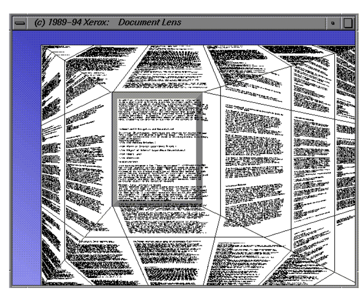
Fisheye ConeTree: Focus 2



Document Lens

- Suitable for laying out the entire contents of a multi-page document and for magnifying places the user is interested in
  - Discovered via direct interactions or via a search
  - Similar to perspective wall, mapped to a pyramid function
  - User can move lens in x,y and z directions

Document Lens

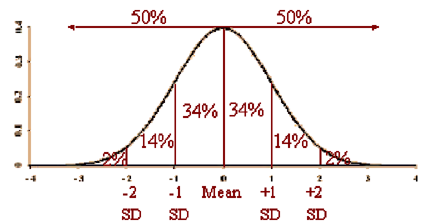


Robertson, Mackinlay UIST'93

Pliable Surfaces

- Also known as 3DPS
- 3D dimensional surface, mapped to the screen by perspective projection
- Gaussian curve – determined by the height of the curve (how magnified the focal point is) and the standard deviation (the amount of compression at the sides)
- http://pages.cpsc.ucalgary.ca/~sheelagh/personal/test/
- www.idelix.com (demo)

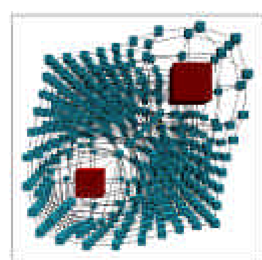
Gaussian curve



Extending from 2D to 3D

- Colin Ware: approx a 3 fold increase in usable space from 2D to 3D (not *n* times more space)
- Visual Access Distortion
- Sheelagh Carpendale's website at University of Calgary: http://pages.cpsc.ucalgary.ca/~sheelagh/personal/test/

3D Visual Access Distortions



Sheelagh Carpendale et al 1997

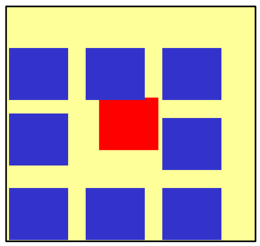
Preserving the mental map

- Various layout algorithms are used for presenting software structures
- When adjusting a layout, it is important to preserve the user's mental map
- The following properties affect a user's mental map [Misue94]:
  - Orthogonal relationships
  - Proximity relationships
  - Graph topology

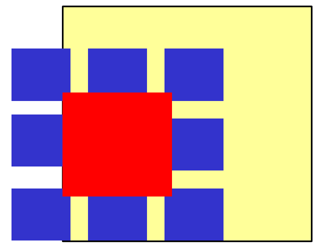
SHriMP Context+Detail Algorithm

- Simple algorithm [Storey95]
- Nested graphs
- Multiple focal points
- $O(k\ n)$ where k = # of focal points
- Suitable for different layouts

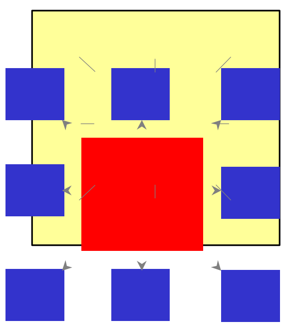
SHriMP fisheye view algorithm



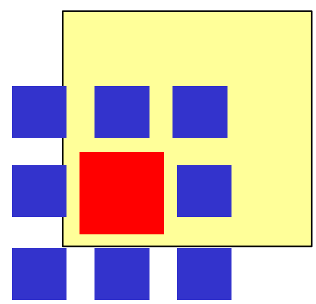
SHriMP fisheye view algorithm



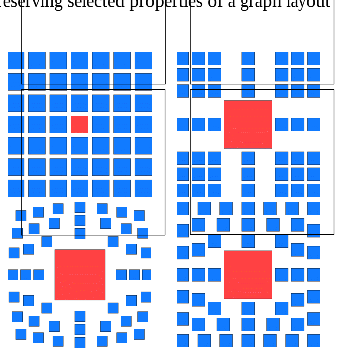
SHriMP fisheye view algorithm



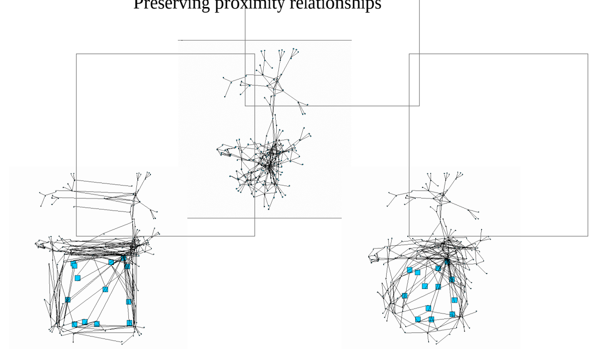
SHriMP fisheye view algorithm



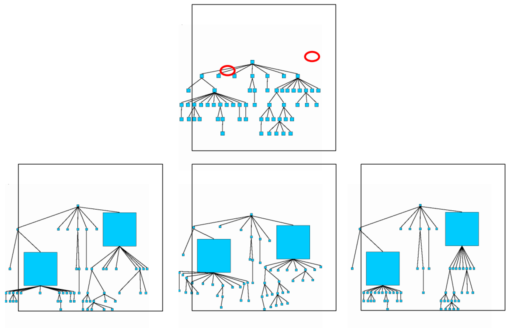
Preserving selected properties of a graph layout



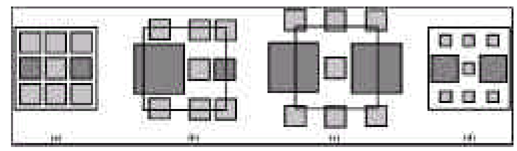
Preserving proximity relationships



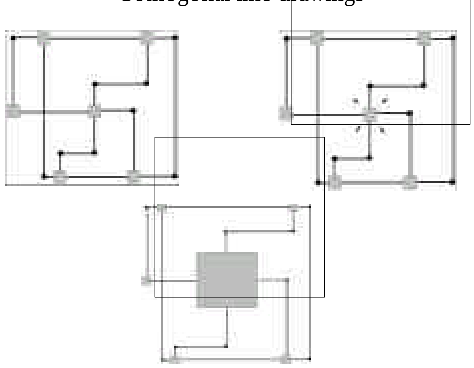
Hybrid strategies



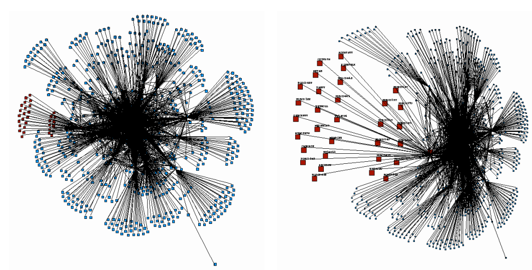
Multiple focal points



Orthogonal line drawings



Visualizing large software structures



Intelligent Zoom

- Intelligent zoom incorporates a context based selecting of an appropriate representation e.g. whether to show a video for part of a network or to present a bar-chart with numeric representation of traffic within the network
- Representations differ in their level of detail but also in their *aspect*
- Each node has a different representation
- DOI affected by static and dynamic influences as well as generic versus application specifics

Fisheye Menus

http://www.cs.umd.edu/hcil/fisheyemenu/

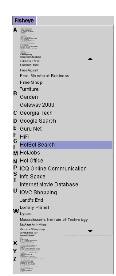


Table Lens

- Demo: http://www.tablelens.com/
- Fisheye technique applied to tabular information – cases-by-variables table (relational tables)
- Allows direct manipulation for exploratory data analysis
- Can merge graphical representations in the table with detailed content
- Distortions are simpler due to the properties of the table
- Distortions in either direction are independent of each other (no bending of lines)
- Easier to have multiple focal areas

Table Lens (2)

- The DOI maps from a cell address to an interest level
- Each dimension has an independent DOI function
- Similar to the Bifocal display
- 3 canonical operations:
  - Zoom changes ✍ amount of space to the focal point
  - Adjust changes ✍ contents in the focal area
  - Slide changes ✍ location of the focal area
- Order effect on multiple focal areas (earlier selected areas will shrink as new focal areas are selected)

Table Lens (3)

- Presentation types for the cells:
  - Text
  - Color
  - Shading
  - Length
  - Position
- What to show in a cell:
  - Value of an attribute
  - Value-type
  - Region-type
- Can use text and colour to help integrate detail in an overview picture
- Similar in some ways to Bertin's permutation matrix
- Focus is manipulated using control points
  - 3 different levels: hidden/focal/non-focal
- Columns can be rearranged/sorted

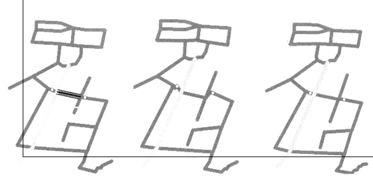
Fisheye Techniques

- Perspective wall
- Cone Trees
- Document Lens
- 3DPS
  - Extending from 2D to 3D
- SHriMP Constrained Fisheye layouts
- Intelligent Zoom
- Fisheye Menus
- Table Lens
- Focus Lens distortions
- Zooming in Application Space

Focus line distortions

- Application area is symbol displacement in maps
- Significant aspects are the shape and the amount of displacement of objects in its vicinity

Focus Line



(a) (b) (c)

Figure 8.7: The use of angle damping: The Focus Line used in the left image leads to a highly distorted orientation of street segments close to the focus line (see middle image), whereas angle damping, as shown on the right, treats segments in the vicinity more carefully.

Focus Line

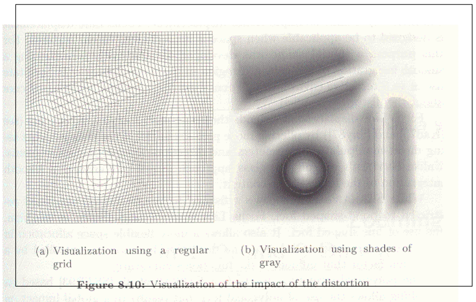
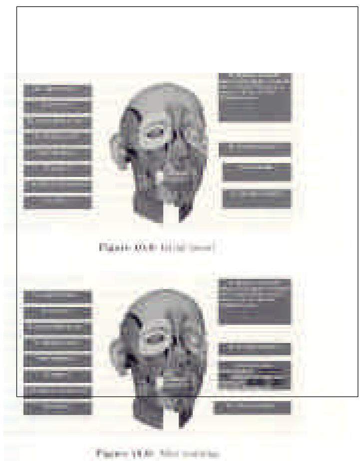


Figure 8.10: Visualization of the impact of the distortion

Zoom navigation in application space

- Talked about geographic and information space
- We can also support zooming and fisheye views in *application space*
- -- Similar problems to navigating in large information spaces (heterogeneous information spaces)
- One familiar approach supports in-place editing (e.g. Microsoft OLE2 and OpenDoc) to avoid explicitly chaing application windows
- Helpful.. but doesn't address the lack of space issue
- Semantic zooming and fisheye views combined can be very powerful!



Distorting Fisheye Views -- Issues

- Distortion effects
- Performance
- Interaction mechanisms
- Reversibility
- Multiple focal points -- not all techniques support multiple foci, and some are non-commutative (that is the order that they are applied affects the layout)
- How to hide the underlying mathematical function and provide "intuitive" interaction facilities

Unified theory of distortion views

- In principle, distortions displace points surrounding the focus by adding a displacement vector to each point (for example, radially pointing outwards)
- Then the points are magnified using some magnification function
- Not always necessary to distort equally in all directions, or to translate in the same direction
- We can decide how and where to borrow space
- Ideally techniques should support direction-dependent distortions

Visual Transfer Functions

- Way of describing the visual distortions used for fisheye views
- A distorted view is created by applying a mathematical function (a transformation) to an undistorted image
- The magnification function is the derivative of the transformation function
- Some areas can have negative magnification factors – resulting in folding

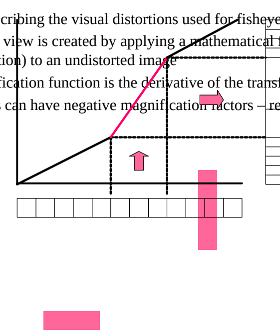


Figure from Jock Mackinlay's website

Transformation and Magnification Functions

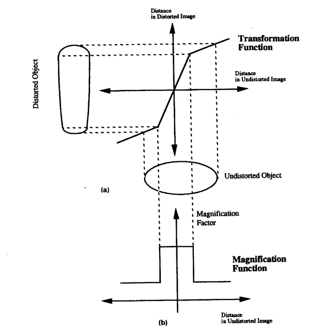


Fig. 3. (a) The transformation of an elliptic object by applying the transformation function of a Bifocal Display in one dimension; (b) the corresponding magnification function of the Bifocal Display.

One Unifying Metaphor….

- Think of the space like a big rubber sheet mounted on a rigid frame, any stretching in one area will result in shrinking in another area

A Taxonomy of Distortion Oriented Techniques

- Provides a unifying theory
- Distorted versus non-distorted

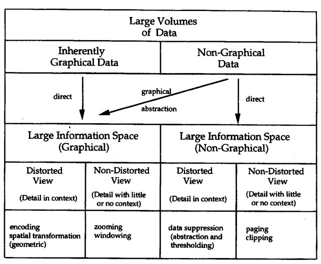


Fig. 1. A taxonomy of presentation techniques for large graphical data spaces.

Taxonomy (2)

- Observations on existing techniques:
  - Bifocal –discontinuous borders between magnified and non-magnified areas (relatively simple to implement)
  - Fisheye view (Furnas)
  - Perspective wall – similar to Bifocal, except that side panels are increasingly small towards the edges
    - Rate of increase depends on angle and width of the focal area and available screen space (the viewer's position (camera) can also be varied)
    - One problem is wasted screen space

Bifocal transformation function

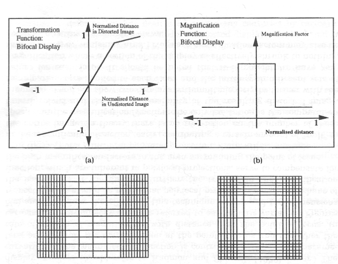


Fig. 6. The Bifocal Display: (a) a typical transformation function; (b) the corresponding magnification function; (c) the application of the display in one dimension; (d) the application of the display in two dimensions.

Fisheye view transformation function

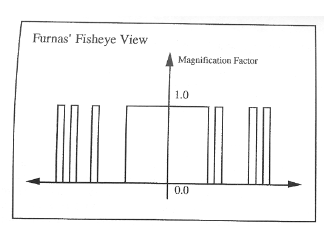
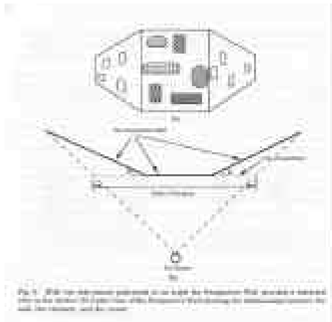
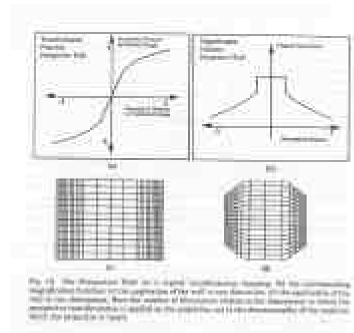


Fig. 8. A typical magnification function for Furnas' "Fisheye View."

Perspective Wall



Perspective Wall



Taxonomy (3)

- Some methods are piecewise non-continuous magnification functions (Bifocal/Perspective Wall)
  - Also different types of magnification – constant (bifocal) versus varying (perspective)
- Whereas others have continuous magnification functions (fisheye and polyfocal)
- Problems with continuous magnification function is that the boundaries of the image tend to be distorted – corner areas are pulled in towards the center (resulting in lost screen space)
  - Sarkar and Brown fixed this by mapping polar coordinates to cartesian coordinates.

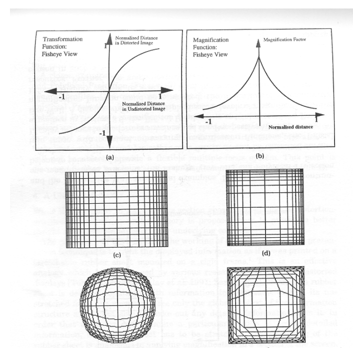


Fig. 11. The Fisheye View: (a) a typical transformation function; (b) the corresponding magnification function; (c) the application of the Fisheye View in one dimension; (d) a Cartesian Fisheye View in two dimensions; (e) a polar Fisheye View; (f) a normalized polar Fisheye View.

Interaction issues

- Scrolling, pointing, selecting and dragging
- Detail can be omitted from non-focus areas during interaction (during zooming)
- Responses that are too fast can also be disconcerting for the user
  - Little empirical work available on optimum settings…
- Multiple focal areas sometimes result in unintended focal areas
- Hybrid techniques can be very powerful.. Combining fisheye with zooming with multiple views, use different approaches for different kinds of information (e.g. use fisheye for the geographic information, detailed view window for annotation information)
- Note current software and hardware *can* handle most of these algorithms

Videos

http://www.cs.indiana.edu/hyplan/tkeahey/research/fad/fad.html#examples

Our Taxonomy

Alternate Geometry

Hyperbolic Spaces

- Hyperbolic spaces do not exist, they are merely a mathematical abstraction
- Advantage is that available space as you move away from the center grows exponentially – just as the width of a tree grows exponentially
- Properties of hyperbolic space – parallel lines diverge and the circumference of the circle grows exponentially with the radius
- Do a layout of the tree on a hyperbolic plane and then map it to a Euclidean disk
- Nodes are embedded in rectangles to ensure legibility of textual information

Hyperbolic Spaces (2)

- Originally the root is the focus, can drag the root or select other nodes to become the focus
- Can change how much space is dedicated to the focal areas, and can easily change the focal area
  - Pointing and dragging for manipulation
- Modest computations required to do new focal points (don't need to render nodes below a certain resolution – only need to translate nodes that will be visible in the new layout)
- Don't need to reissue the layouts, just need to change the translation (mapping) from the hyperbolic plane to the Euclidean disk
- Layouts simpler in hyperbolic space – more space, children are positioned on an arc far enough out so that they all fit
- Different mappings – Klein model (preserves straightlines) and Poincare model (preserves angles – preferred)

Zooming windows / Zoomable User Interfaces

- ZUIs: Pad++/Jazz
  - Issues:
    - Panning – does the camera or the scene move?
    - Zooming in/out (speed, interaction issues)
  - Features:
    - Sticky objects
    - Lenses
    - Semantic zooming
    - Portals

Pad++: A Zoomable Interface

- Pad++ is a general substrate for exploring visualization of graphical data with a zooming interface
- Goal: move beyond the traditional desktop (windows and icons) metaphor!
- Concepts:
  - Uses an infinite resolution sketchpad
  - With history-enriched digital objects
  - Use semantics and history to help choose how to render objects

Pad++ (2)

- Supported objects:
  - Coloured text
  - Text files
  - Hypertext
  - Graphics
  - Images
- Basic UI interactions:
  - 3 button mouse:
    - Left is mode dependent
    - Middle zooms in, right zooms out
    - Other approaches for 1 and 2 button mice
- "Semantic zooming"
  - Zoomed in, see details
  - Zoomed out, see a different representation which is more useful

Pad++ (3)

- Interaction goal:
  - Smooth animations are need to maintain context
  - Smooth animation even with large datasets
  - Achieved interactivity for 600,000 objects (our experiences weren't quite as good, depends on the underlying structures you use)
  - Many attempts to improve performance:
    - Don't render very small objects
    - Lower resolution for small objects
    - Clipping, delete objects if not accessed in a while
    - Delay refinement
    - Incremental loading of objects
    - etc

Pad++ (4)

- Physics versus metaphor (why?)
  - Metaphors pre-exist their use
  - Useful for temporary bridging concepts
  - If successful the metaphor dies
  - Existing metaphors may restrict is from thinking about alternative organizations of computation
  - As our information needs, our interfaces need to scale, but many of the existing metaphors cannot be scaled…
  - Consequently, if we use simple laws we can maybe deal with larger, more complex spaces by applying such laws
- Learning how to use/interact with a physics based system may be hard (the user needs to learn and understand how the laws work separately and together)

Zoomable interfaces

- Pad++ applications:
  - Directory browser
  - Timelines
  - Hypertext browsing
- More recently replaced by Jazz
  - URL: http://www.cs.umd.edu/hcil/jazz/
  - Now augmented by Piccolo: http://www.cs.umd.edu/hcil/jazz/download/index.shtml

Space–scale diagrams and multi-scale diagrams

- Space scale diagrams provides a framework for representing both a spatial world and its different magnifications explicitly
- These diagrams allow the direct visualization and analysis of important scale related issues for interfaces

Space-scale diagrams

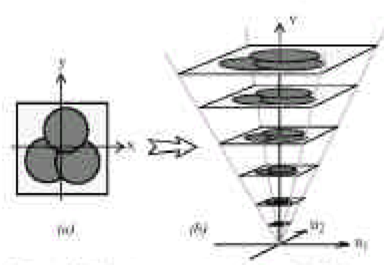


Figure 1. *The basic construction of a Space-Scale diagram from a 2D picture.*

Understanding pan + zoom interactions

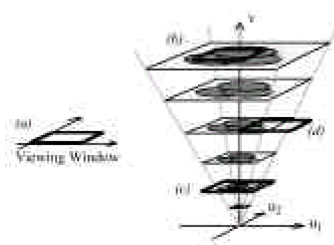


Figure 2. *The viewing window (a) is shifted rigidly around the 3D diagram to obtain all possible pan/zoom views of the original 2D surface, e.g., (b) a zoomed in view of the circle overlap, (c) a zoomed out view including the entire original picture, and (d) a shifted view of a part of the picture.*

Shear invariance

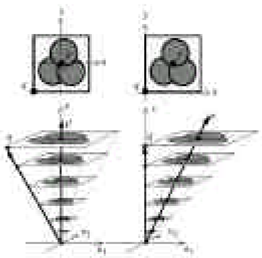
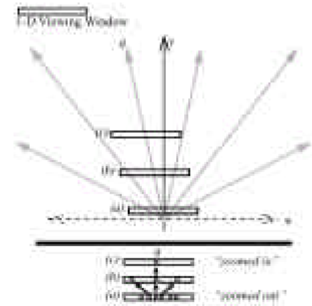


Figure 4. *Shear invariance. Shifting the origin in the 2D picture from it is a correspondence to shearing the layers of the diagram so the q-line becomes vertical. Each layer is unchanged, and great rays remain straight. Only those conclusions which remain true under all such shears are valid.*

1 + 1D



Visualizing basic pan+zoom trajectories

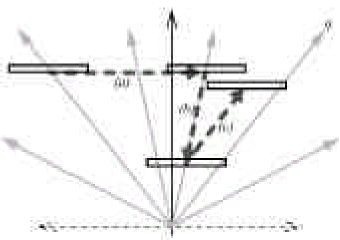


Figure 6. *Basic Pan-Zoom trajectories are shown in the heavy dashed lines: (a) is a pure Pan, (b) is a pure Zoom (out), (c) is a "Zoom-around" the point q.*

Panning and zooming at the same time

- Many times when the system must automatically pan and zoom from one place to another
- Pan is linear at any scale but a zoom is logarithmic (changing magnification by a constant factor in a constant time)
- If you execute both in parallel a step at a time, the point will run away from you… (because when you zoom in the world view expands exponentially fast, and the target point will run away faster than you can catch up to it, "guessing" where to zoom to is not trivial!)

Solving the pan-zoom problem

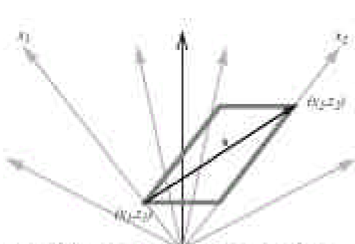


Figure 7. *Solution to the simple joint pan-zoom problem. The trajectory s monotonically approaches point 2 in both pan and zoom.*

Advantages of pan+zoom

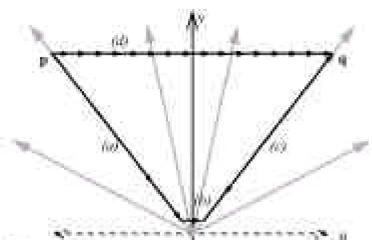
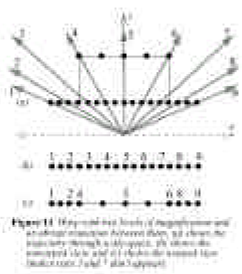
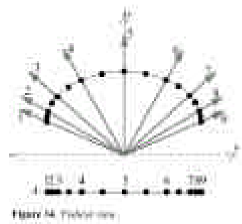


Figure 8. *The shortest path between two points is often not a straight line. Here each arrow represents one unit of cost. Because zoom is logarithmic, it is often "shorter" to zoom out (a), make a small pan (b), and zoom back in (c), than to make a large pan directly (d).*

Space-scale diagrams of fisheye views





Empirical observations

- User interface experiment showed multiple windows were slower to complete tasks than a non-windowed system
  - Window management distracts users from their tasks and uses up time
  - More effective window management doesn't receive enough attention… unfortunately
  - Structural relations between windows are not exploited to help provide between window placement strategies
- Lack of user studies (still) -- We need more empirical observations for different domains/tasks, and for the different techniques
- Things to measure:
  - Ease of learning
  - Ease of use
  - Task completion times

Summary -- Takeaways

- Importance of task in selecting which approach to use!
  - How much time is spent at the detailed level
  - How often does the user switch between a detailed view and contextual view
  - How familiar is the user with the larger information space
  - How important is it that relationships with the larger information space always be visible (for example, diagnostics)
  - What is the task (exploring, navigating, understanding, monitoring…)

References

- Computational Visualization by Thomas Strothotte
- Space scale diagrams, http://www.cs.umd.edu/hcil/pad++/papers/chi-95-spacescale/chi-95-spacescale.pdf
- Jazz website: http://www.cs.umd.edu/hcil/jazz/