Visual Exploration of Large Hierarchies with Information Pyramids

Keith Andrews

IICM, Graz University of Technology, Austria

kandrews@iicm.edu

Abstract

Information pyramids compactly visualise large hierarchical structures in three dimensions using pyramid-like structures, which grow upwards as the hierarchy deepens. A plateau represents the root of the tree; other, smaller plateaus arranged on top of it represent its subtrees. Separate glyphs are used to represent leaf nodes such as files or documents and navigational facilities are provided for interactive exploration. Two generations of information pyramids prototype are described and discussed.

1. Introduction

Hierarchical structures are increasingly common in our information rich society as a means for organising large amounts of information. Examples of information hierarchies include the hierarchical organisation of employees in a company and the files and directories stored on a hard disk. Hierarchies are becoming ever larger; even the hard disk of an ordinary home PC may contain thousands of files. Provision of explorable, visual representations is essential to make large hierarchies accessible.

Traditional visualisation techniques, such as outlines or tree diagrams, are not ideal for visualising large hierarchies. Such methods are perfect for visualising small, compact hierarchies, which can be drawn on one sheet of paper or presented on a computer screen without having to scroll. However, if the hierarchy is too large, the visualisation fails to be simple and understandable.

Traditional tree browsers, such as the Macintosh File Finder or the Windows Explorer, use an outline method to visualise hierarchies. They are simple to implement and can also be used in text-based systems. Typically, not all subtrees of the hierarchy are expanded initially. Instead, the user navigates by specifically expanding and collapsing subtrees, and hence decides explicitly how broadly or deeply the hierarchy is displayed at any one time.

2. Related Work

In recent years, numerous techniques for visualising and exploring large, hierarchically organised information structures have emerged, which go beyond the traditional approach taken in 2d scrolling browsers with horizontal tree layout.

WebTOC extends the traditional tree view by overlaying supplementary statistical information [1]. Coloured bars to the right of the tree outline indicate the proportion of files of various media types by size. Their shadows indicate the number of files present. For conciseness, the bars are displayed using a logarithmic scale.

Tree maps [2] are an interactive visualisation method which map hierarchical information onto a rectangular two-dimensional display in a space-filling manner. The display space is partitioned successively into vertical and horizontal strips representing the tree structure, whereby the size of a child node within the rectangular boundary of its parent is proportional to its weight, typically the total number or size of items within it.

Market maps are a recent development of the tree map idea and are used to visualise stock market performance [3]. The original tree map algorithm was modified to generate more aesthetically pleasing layouts by avoiding excessively thin horizontal and vertical strips [4]. Harel and Yashchin [5] present an algorithm for the layout of an inclusion hierarchy using rectangular shapes. The layout attempts to strike a balance between various aesthetic layout criteria such as importance and proportions.

The hyperbolic browser [6] uses a focus plus context technique for visualising and manipulating large hierarchies: the entire hierarchy can be seen in context while focusing on a particular part. The hierarchy is initially laid out on a hyperbolic plane, a non-Euclidean geometry with infinite space available in each direction, and is then mapped to the unit disc for display. SInVis [7] achieves a similar visual effect using Euclidean geometry and projection onto the surface of a hemisphere.

Cheops [8] is a technique based on multiple re-use

Proceedings of the Sixth International Conference on Information Visualisation (IV'02)
1093-9547/02 $17.00 © 2002 **IEEE**

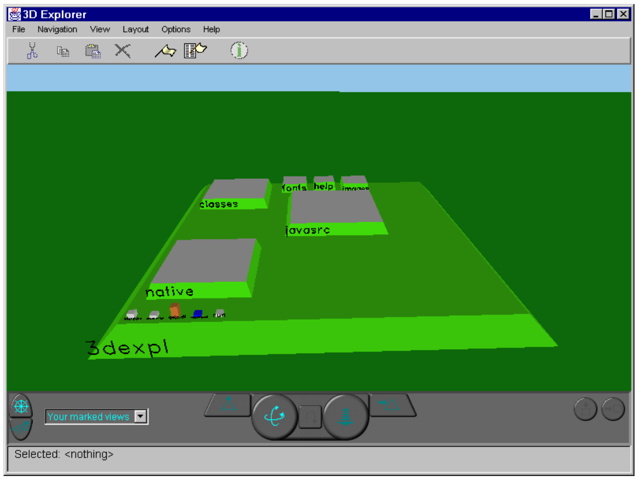

Figure 1: An information pyramids visualisation showing a directory "3dexpl" and its subdirectories and files. Grey plateaux represent directories not yet scanned in. The area allocated to each subdirectory is proportional to its weight, typically the total size or number of files it contains.

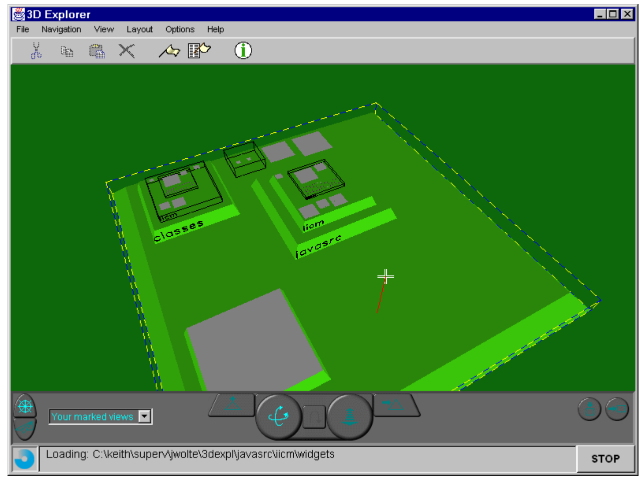

Figure 2: During navigation interactive frame rates are maintained by drawing at less detail or as wire frames, until the user again becomes stationary.

of overlaid triangles in the display. Working top-down, the selection of a node (triangle) at a particular level designates that node's children are to be represented by the next lower level of triangles.

Cone trees [9] lay out hierarchies in three dimensions. Each node in the hierarchy is the apex of a cone. The root of the hierarchy is placed near the top of the three-dimensional display space and its children are evenly spaced along its base. The next layer of nodes is drawn below the first layer, recursively until the whole hierarchy is drawn. Cone trees suffer from problems of occlusion as hierarchies become broad and branches become hidden behind their siblings, and interactivity is employed to rotate hidden branches.

The File System Navigator (FSN) [10] and the Harmony Information Landscape [11, 12] lay out a hierarchical structure on a plane, whereby directories (collections) are represented as pedestals, files (documents) within them are shown as icons atop the pedestal, and subdirectories (subcollections) are arranged behind the corresponding parent. The overall impression is that of hierarchies receding towards the horizon.

3. Information Pyramids

The information pyramids approach utilises three dimensions to compactly visualise large hierarchies. A plateau represents the top of the hierarchy (or root of the tree). Other, smaller plateaus arranged on top of it represent its subtrees. Separate icons are used to represent non-subtree members of a node such as files or documents. The general impression is that of pyramids growing upwards as the hierarchy grows deeper.

The first implementation of information pyramids was the 3D Explorer file system browser [13], implemented in Java and using OpenGL for its 3d graphics output. An updated, revised version of the 3D Explorer [14] is shown in Figure 1, displaying the source and class files of its own implementation in Java. By default, the space allocated to a directory's plateau is proportional to the total number of files belonging to it and all of its subdirectories. The grey plateaux indicate subdirectories whose contents have not yet been fully scanned.

In order to maintain interactive frame rates, the quality of rendering gracefully degrades during movement. Drawing performance is monitored, and less detailed representations are used when time becomes scarce, as shown in Figure 2. Once the user comes to a halt, the scene is rendered in its entirety once more.

Users can freely navigate around the pyramid landscape. As pyramids are approached, more details are revealed. In Figure 3 the user is approaching a close-up of the particular pyramid representing the "javarsc" subdirectory and its children.

The ordering of children atop a plateau can be based on alphabetical, chronological, or other criteria. In Figure 4, colour coding is used to indicate a file's type, but could also be mapped to age, or any of a number

Proceedings of the Sixth International Conference on Information Visualisation (IV'02)
1093-9547/02 $17.00 © 2002 **IEEE**

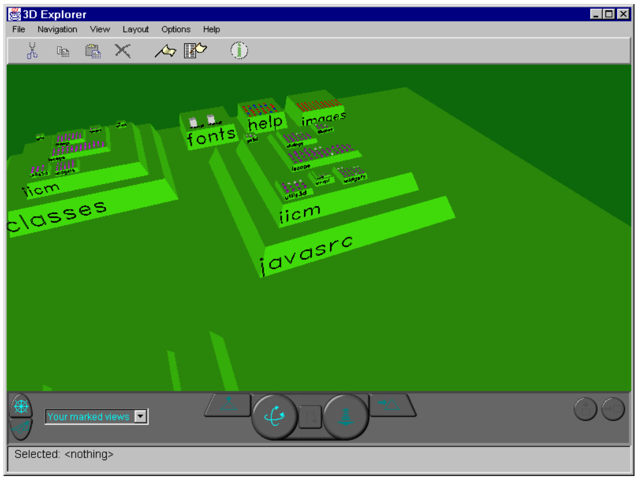

Figure 3: Zooming in on the subdirectory "iicm" of "javasrc".

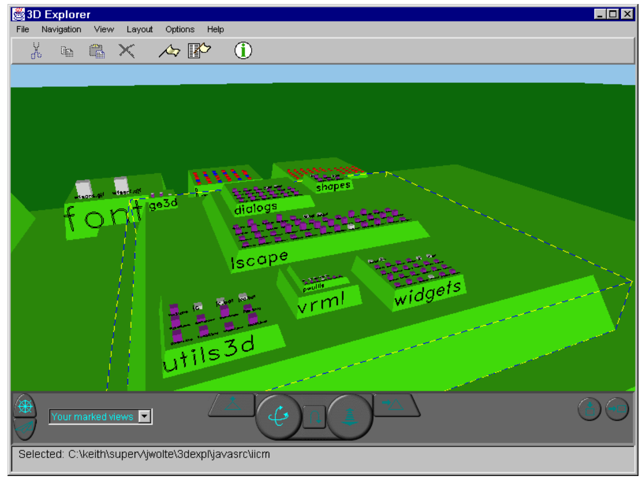

Figure 4: By default, files are colour-coded according to their extension. Here, for example, purple indicates ".java", red ".gif", and blue ".html". The ordering of children atop a plateau can be based on alphabetical, chronological, or other criteria.

of other characteristics.

As the user approaches the "utils3d" subdirectory in Figure 5, its contents become identifiable and the file names become readable. Notice how the file name labels are inclined towards the user's point of view. By default, the height of the glyphs representing files is logarithmically proportional to their size.

A bird's eye view of the pyramid landscape provides a space-filling, freely zoomable overview. In this view, illustrated in Figure 6 for the Java JDK 1.2.2 distribution, it is easy to recognise (visually) which directories contain the largest number of files by their relative areas.

The visual representation is only one component of the interface. Equally important is the provision of suitable navigational facilities, in order support intuitive exploration of the hierarchy. In the 3D Explorer, users can choose between a wide variety of navigation facilities. Clicking on a file glyph or directory plateau flies the user smoothly to view that object.

Special facilities in the navigation panel at the bottom of the screen allow users to freely navigate. Examine mode allows the user to rotate and scale the pyramids visualisation, Explore mode allows users to freely move their own viewpoint within the visualisation, and Fly mode allows the user to steer and control speed whilst flying over the visualisation. Two buttons provide direct access to a top view and a front view. Finally, the user can define customised viewpoints (place stakes in the ground), which can be returned to at will.

Directories are scanned and loaded in a separate thread, so that users can navigate whilst the visualisation is being updated. In order to reduce clutter, users can "prune" the pyramid to a particular directory, making its plateau the current root of the pyramid, and (temporarily) eliminating all other elements from the display. The reverse operation, "unprune", makes the directory's parent and siblings visible once more.

Formative usability evaluation of the 3D Explorer suggested that the information pyramids technique scales well to both wide and deep hierarchies. Nodes deep within the tree are not (initially) individually visible but contribute to the overall proportions of their parents. Broad hierarchies result in many pyramids at each tier, but remain manageable. Users were positive in their comments and liked the overview which information pyramids provided.

However, users had some difficulty in locating specific individual files of which they knew the location within the hierarchy. They also displayed a tendency to become immersed in the 3d navigational facilities and rather forget their initial intentions. Several users reported finding the textual labels to be too cluttered.

4. Java Pyramids Explorer

To incorporate some of the lessons learned, a second prototype information pyramids hierarchy browser was implemented, called the Java Pyramids Explorer [15]. The main changes were:

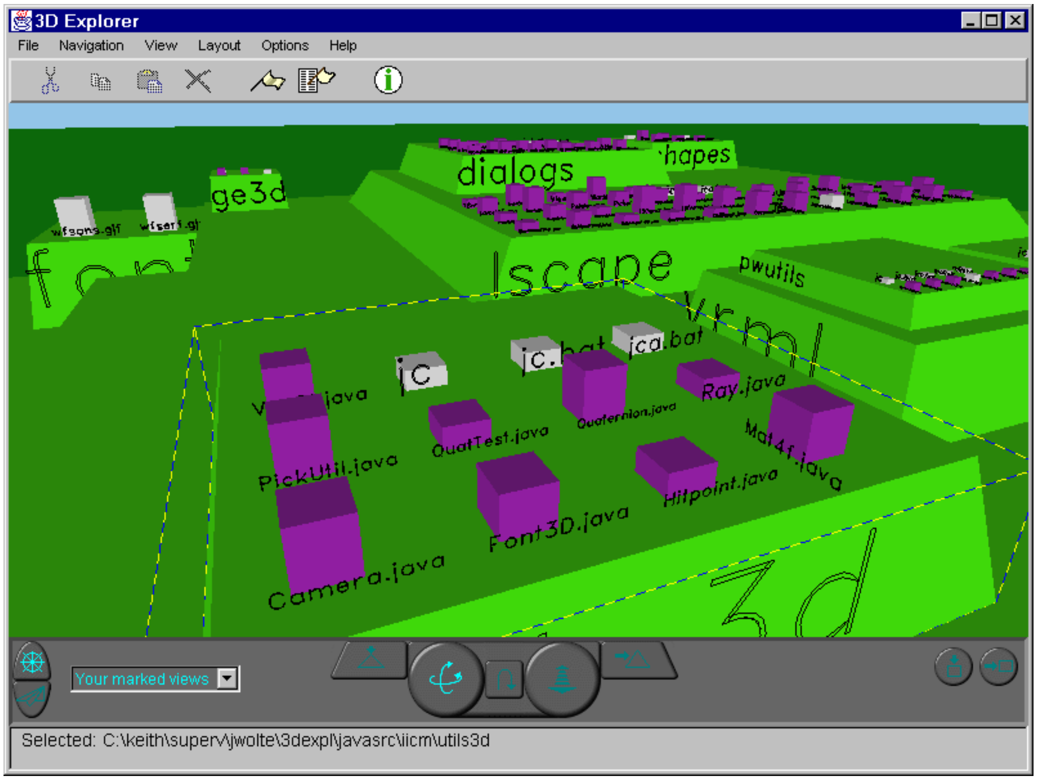

Figure 5: The file names of individual files are inclined towards the user's viewpoint and are now readable.

- To facilitate directed browsing in a known hierarchy, a traditional outline browser, the Java JTree component, was paired and synchronised with the information pyramids browser.

- The extensive 3d navigational facilities were replaced with three simple sliders controlling the 3d view.

- For increased portability, the OpenGL 3d graphics was replaced with Java2d code implementing simple 3d projection.

- In order to reduce clutter, textual labels are now displayed in the margins of the pyramids view, and only for selected nodes and their parent, siblings, and children.

Figure 7 shows the Java Pyramids Explorer displaying the Java source tree of its own implementation. Figure 8 shows the Java JDK 1.2.2 distribution in the Java Pyramids Explorer. Here, individual files are not being displayed, only the directory structure.

5. Discussion and Future Work

This paper presented the information pyramids technique for visualising and exploring large hierarchies. Although the example prototype applications of information pyramids presented in this paper refer to the tree structure of files and directories in a file system, the information pyramids metaphor is of course applicable to any kind of hierarchical structure: hierarchical classification schemes, decision trees, or the hierarchical structure of a web site derived by traversing hyperlinks from a selected initial page.

Work is currently ongoing to define and implement a framework for hierarchical visualisation. The framework will provide multiple, synchronised visualisations, of which information pyramids are one component, and search facilities.

In information pyramids, future work includes the implementation of user-configurable mappings between an object's metadata (such as age, owner, or size) and its visual representation. It is also intended to explore the placement of items (files) on a plateau into clusters according to similarity of their content using force-directed placement techniques.

Proceedings of the Sixth International Conference on Information Visualisation (IV'02)
1093-9547/02 $17.00 © 2002 **IEEE**
IEEE Computer Society

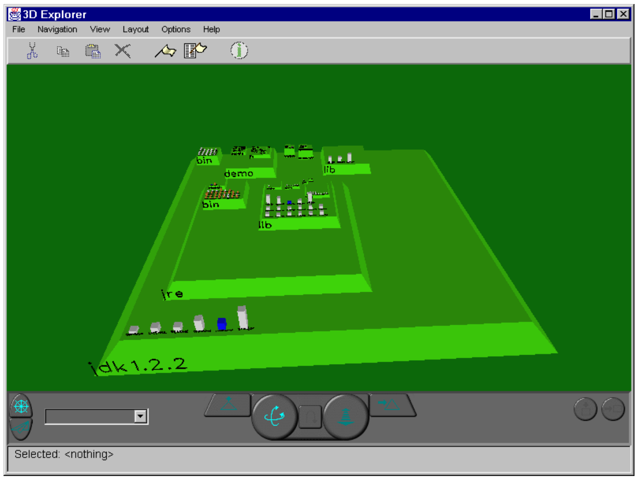

Figure 6: The hierarchical structure of the Java JDK 1.2.2 distribution, which contains 6 first level subdirectories, extending down to a depth of six. It can be seen at a glance that the subdirectory "jre" is by far the largest and that subdirectories "bin" and "jre/bin" contain many files.

6. Acknowledgements

I would like to acknowledge the contributions of my former and current Master's students: Josef Wolte and Michael Welz (both since graduated), and Helmut Heidegger and Martin Kaufmann.

References

[1] David A. Nation, Catherine Plaisant, Gary Marchionini, and Anita Komlodi. Visualizing websites using a hierarchical table of contents browser: Webtoc. In *Proc. 3rd Conference on Human Factors and the Web*, Denver, Colorado, June 1997. US WEST. ftp://ftp.cs.umd.edu/pub/hcil/Demos/WebTOC/Paper/WebTOC.html.

[2] Brian Johnson and Ben Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proc. IEEE Visualization '91*, pages 284–291, San Diego, California, October 1991. IEEE Computer Society. http://www.cs.umd.edu/hcil/treemaps/.

[3] Martin Wattenberg. Visualizing the stock market. In *CHI 99 Extended Abstracts*, pages 188–189, Pittsburgh, Philadelphia, May 1999. ACM. http://www.smartmoney.com/marketmap/.

[4] Ben Shneiderman and Martin Wattenberg. Ordered treemap layouts. In *Proc. IEEE InfoVis 2001*, pages 73–78, San Diego, California, October 2001. IEEE Computer Society. http://www.computer.org/proceedings/infovis/1342/13420073abs.htm.

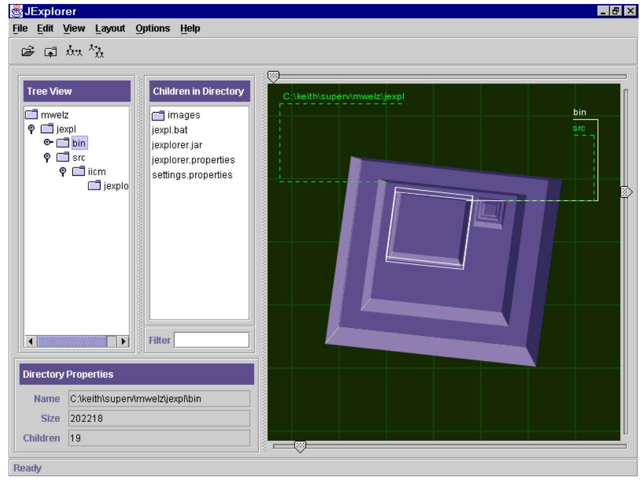

Figure 7: The Java Pyramids Explorer displaying the Java source tree of its own implementation. The tree view and pyramids view are synchronised.

[5] David Harel and Gregory Yashchin. An algorithm for blob hierarchy layout. In *Proc. AVI 2000*, pages 29–40, Palermo, Italy, May 2000. ACM. http://doi.acm.org/10.1145/345513.345240.

[6] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. CHI'95*, pages 401–408, Denver, Colorado, May 1995. ACM. http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/jl_bdy.htm.

[7] Matthias Kreuseler, Norma Lopez, and Heidrun Schuhmann. A scalable framework for information visualization. In *Proc. IEEE InfoVis 2000*, pages 27–36, Salt Lake City, Utah, October 2000. IEEE Computer Society. http://www.informatik.uni-rostock.de/~mkreusel/SInVis/infovis.html.

[8] Luc Beaudoin, Marc-Antoine Parent, and Loius C. Vroomen. Cheops: A compact explorer for complex hierarchies. In *Proc. Visualization'96*, pages 87–92, San Francisco, California, October 1996. IEEE Computer Society. http://www.crim.ca/hci/cheops/paper.html.

[9] George G. Robertson, Jock D. Mackinlay, and

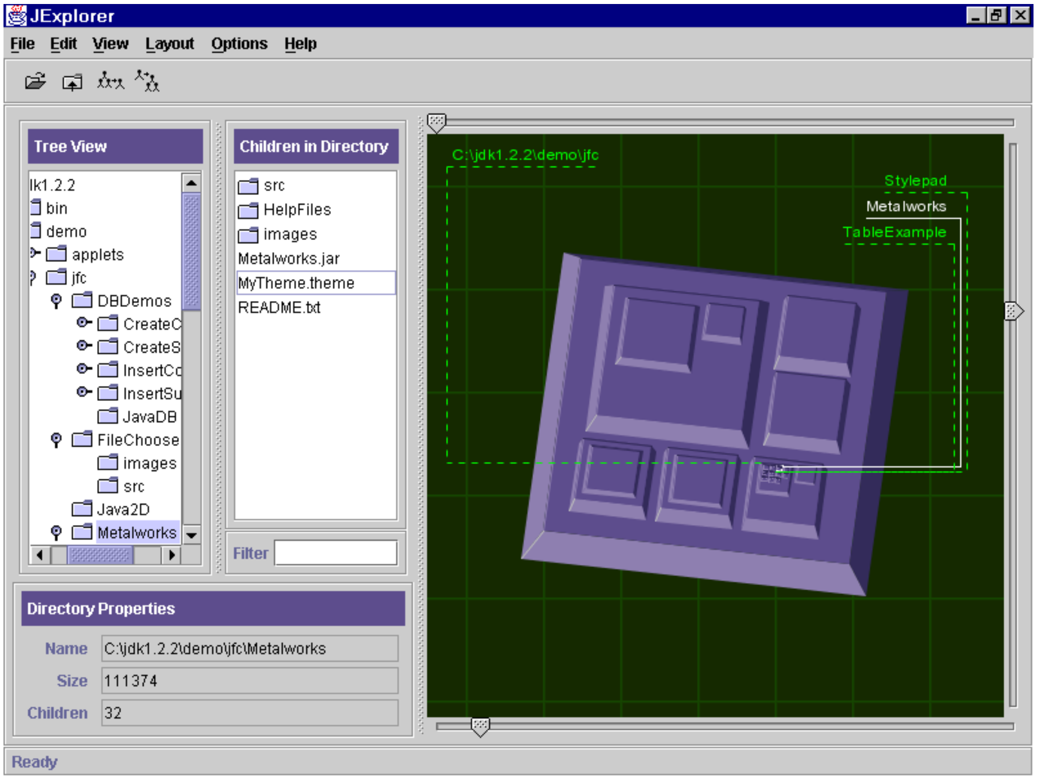

Figure 8: The Java JDK 1.2.2 distribution in the Java Pyramids Explorer.

Stuart K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proc. CHI'91*, pages 189–194, New Orleans, Louisiana, May 1991. ACM.

[10] Joel D. Tesler and Steven L. Strasnick. Fsn: The 3d file system navigator. Silicon Graphics, Inc., 1992. ftp://sgi.sgi.com/sgi/fsn.

[11] Keith Andrews. Visualising cyberspace: Information visualisation in the Harmony internet browser. In *Proc. First IEEE Symposium on Information Visualization (InfoVis'95)*, pages 97–104, Atlanta, Georgia, October 1995. ftp://ftp.iicm.edu/pub/papers/ivis95.pdf.

[12] Keith Andrews, Michael Pichler, and Peter Wolf. Towards rich information landscapes for visualising structured web spaces. In *Proc. 2nd IEEE Symposium on Information Visualization (InfoVis'96)*, pages 62–63, San Francisco, California, October 1996. ftp://ftp.iicm.edu/pub/papers/ivis96.pdf.

[13] Keith Andrews, Josef Wolte, and Michael Pichler. Information pyramids: A new approach to visualising large hierarchies. In *IEEE Visualization'97, Late Breaking Hot Topics Proc.*, pages 49–52, Phoenix, Arizona, October 1997. ftp://ftp.iicm.edu/pub/papers/vis97.pdf.

[14] Josef Wolte. Information pyramids: Compactly visualising large hierarchies. Master's thesis, Graz University of Technology, Austria, October 1998. ftp://ftp.iicm.edu/pub/theses/jwolte.pdf.

[15] Michael Welz. The java pyramids explorer: A portable, graphical hierarchy browser. Master's thesis, Graz University of Technology, Austria, September 1999. ftp://ftp.iicm.edu/pub/theses/mwelz.pdf.