Late Breaking Hot Topics Proc., IEEE Visualization'97, Phoenix, Arizona, October 1997, pp. 49–52.

Information Pyramids™ : A New Approach to Visualising Large Hierarchies

Keith Andrews Josef Wolte Michael Pichler*

IICM, Graz University of Technology, Austria

ABSTRACT

This paper presents work in progress on a new technique for visualising and manipulating large hierarchies.

The Information Pyramids approach compactly visualises hierarchical structures in three dimensions using pyramid-like structures, which grow upwards as the hierarchy is descended. The technique is described in the context of our early experience with a prototype file system visualiser based on Information Pyramids.

Keywords: Information visualisation, hierarchies, trees, 3d.

1 INTRODUCTION

Over the last few years, the emerging field of information visualisation has resulted in a numerous techniques for helping visualise and make sense of large information spaces [11]. Among these are several techniques for visualising and interacting with large hierarchies which go beyond the traditional approach taken in 2d scrolling browsers with horizontal tree layout.

Cone Trees and Cam Trees [9] utilise three-dimensional space to fan out a hierarchical structure as a vertical or horizontal cone respectively.

Treemaps [6, 10] are space-filling visualisations of hierarchies based on successive horizontal and vertical subdivision of screen rectangles. The area of each rectangle is proportional to some attribute of the underlying hierarchy such as the (total) size of each subtree.

FSN [12] and the Harmony Information Landscape [1, 2] 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.

The Hyperbolic Browser [7] uses a focus and context technique based on hyperbolic geometry. A hierarchy is laid out uniformly on the hyperbolic plane and then mapped to the unit disk for display on screen. Nodes in the centre of the disk are largest and nodes are assigned progressively less space towards the perimeter of the disk.

Cheops [3] is a novel technique based on multiple re-use 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.

2 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 overall impression is that of pyramids (or mountains if you prefer) growing upwards as the hierarchy is descended.

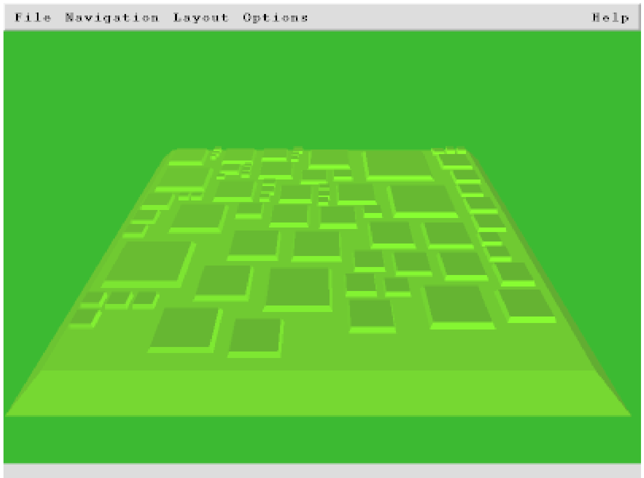

Figure 1: Information Pyramids visualisation showing a directory and its subdirectories.

The current working prototype of Information Pyramids scans and visualises the hierarchical tree structure of directories and files in a UNIX file system. Figure 1 shows the Information Pyramids display for the `tex` subdirectory of Keith Andrews' home directory. The `tex` directory shown here has 74 immediate subdirectories, a maximum depth of 4, and contains a total of 2241 files. Initially, only the `tex` directory and its immediate children are shown. Figure 2 shows the display after the user has chosen to have all files and directories displayed.

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. In Figure 2, the most visible directories are those containing the largest number of files. As will be discussed later, this mapping can be changed interactively by the user. The ordering of children atop a plateau can be based on alphabetical, chronological, or other criteria. Here, colour coding is used to indicate a file's type, but could also be mapped to age, or any of a number of other characteristics.

Users can freely navigate around the pyramid landscape. As pyramids are approached, more details are revealed. Figure 3 shows a close-up of a particular pyramid, the `ivis96` subdirectory and its children.

In order to reduce clutter, users can "crop" the pyramid to a particular directory, making its plateau the current root of the pyramid, and (temporarily) eliminating all other elements from the display, as shown in Figure 4. The reverse operation, "uncrop", makes the directory's parent and siblings visible once more.

A bird's eye view of the pyramid landscape provides a space-

* kandrews@iicm.edu, jwolte@iicm.edu, mpichler@iicm.edu
IICM, Schießstattgasse 4a, A-8010 Graz, Austria.

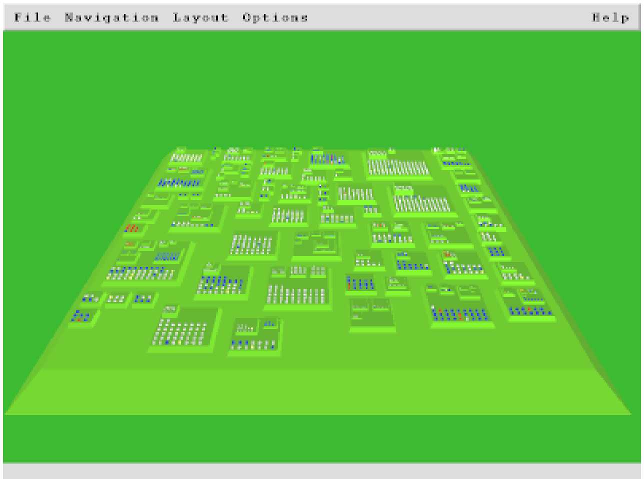

Figure 2: The detailed view showing 74 pyramids upto 4 tiers high and a total of 2241 files.

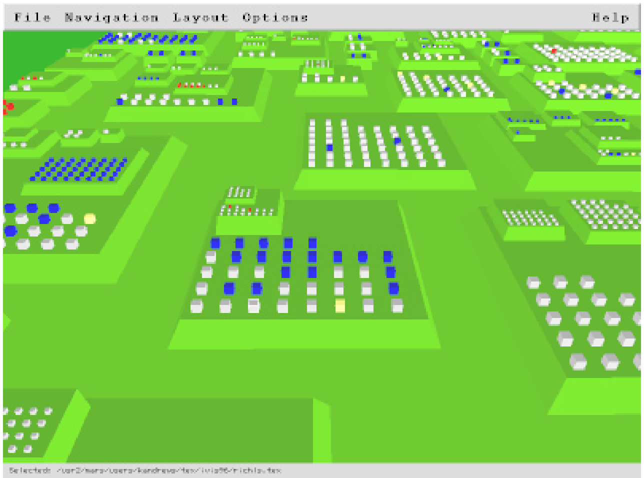

Figure 3: Navigating the pyramid landscape towards the `ivis96` subdirectory.

filling, freely zoomable overview. In this view, illustrated in Figure 5, it is easy to recognise (visually) which directories contain the largest number of files by their relative areas.

Changing the mapping of a plateau's area from number of files to the total size of files within it results in the graphical representation of disk usage shown in Figure 6. In this view of the `tex` directory, three subdirectories containing large, uncompressed PostScript files and large, uncompressed TIFF images stand out from the rest.

Another example of an Information Pyramids visualisation is shown in Figure 7. Here Sun's Java JDK 1.0.2 distribution tree is shown, in disk usage style. The pyramid at the top right represents the `apibook` subdirectory containing documentation in HTML (coloured blue) and GIF (coloured red) files. The pyramid at front left is the `src` subdirectory. The six individual blocks at the front are README and associated files.

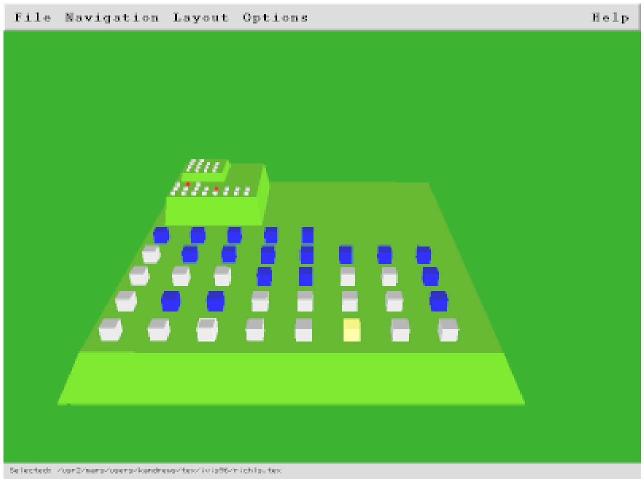

Figure 4: Cropping the pyramid to focus on the `ivis96` subdirectory, making it the new base of the pyramid.

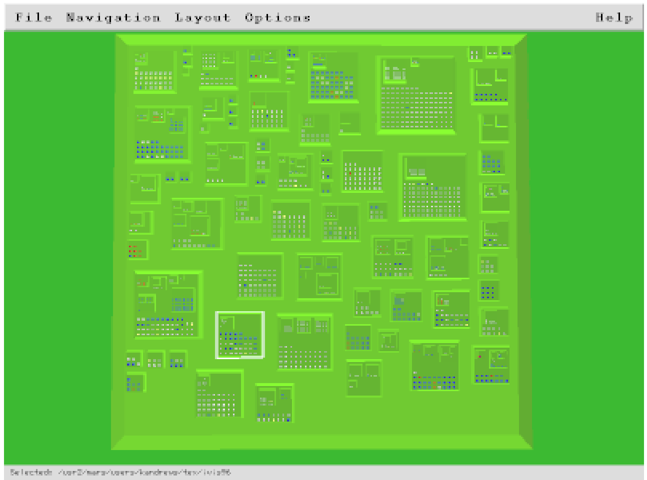

Figure 5: A bird's eye view of the pyramid landscape provides a visual overview.

3 EARLY EXPERIENCE

From our early experience, the Information Pyramids technique appears to scale well to both wide and deep hierarchies. Nodes deep within the tree are not (initially) individually visible but do contribute to the overall proportions of their parents. Broad hierarchies result in many pyramids at each tier, but remain manageable. Decreasing the relative height of successive plateaus decreases the prominence of deeper levels of the hierarchy in the overall landscape, i.e. deeper levels become invisible sooner.

The best visual representation is of little use without accompanying navigational facilities. By navigating through the pyramid landscape, users can, for example, focus on a particular part of the hierarchy and can see progressively more detail of that part, or can float above the pyramid landscape for an overview. Early testers expressed a desire for a simple way of switching between front and top views, leading to the introduction of buttons/commands to automatically switch between pre-defined viewpoints (through a smooth animation).

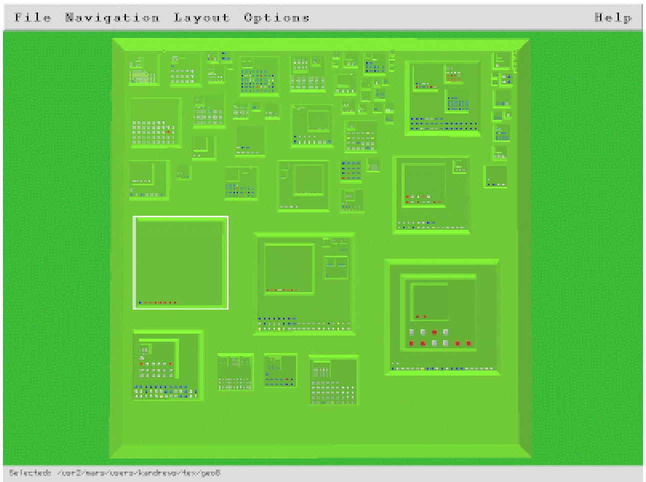

Figure 6: Mapping total file size to plateau area results in a graphical disk usage utility.

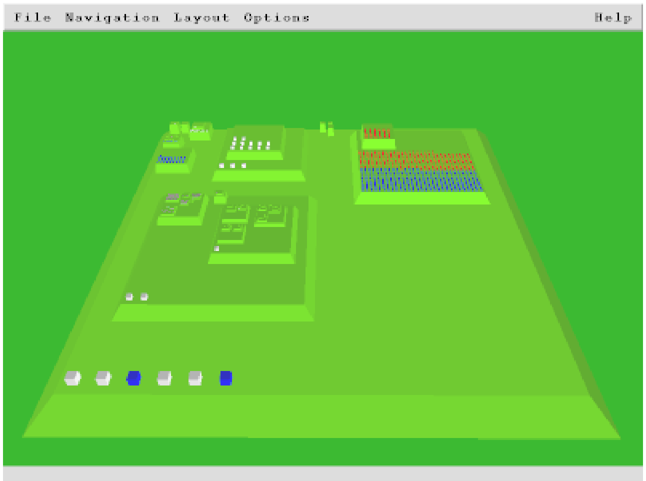

Figure 7: Visualisation of the Java JDK 1.0.2 tree, containing 13 subdirectories up to a depth of 10, and 4455 individual files.

4 CURRENT AND FUTURE WORK

The current prototype file system visualiser using Information Pyramids is written in Java and uses our in-house GE3D graphics interface to either OpenGL or Mesa libraries for 3d graphics output.

In order to maintain interactive frame rates, especially for very large hierarchies, a current focus of work is to implement graceful degradation of the rendering during movement. Important parts of the scene are rendered fully, less important parts only in wireframe or not at all depending on how much time is available, until the user comes to a halt, at which point the scene is rendered in its entirety once more.

We are working on extending the mapping to additional attributes of the nodes being visualised: document age to colour, document type to a particular 3d icon, etc. and on making the mapping user-configurable. We are also working on providing further navigational facilities to traverse the pyramid landscape: for example user-defined viewpoints (stakes in the ground), which can be returned to at will.

Although the example application of Information Pyramids presented in this paper refers to the tree structure of files and directories in a file system, the Information Pyramids metaphor itself is applicable to any kind of hierarchical structure. For example, the Hyperwave Information Server [5] supports explicit hierarchical structuring of the documents it manages into collections and subcollections. Information Pyramids will be used to provide a visualisation and navigation facility for a Hyperwave web site's collection structure.

In the future, the Information Pyramids visualisation of a Hyperwave web site will be extended to display both referential hyperlink and hierarchical collection relationships in a single display. Similar to the 3d Local Map of Harmony [2], a document in the pyramid landscape will be able to be selected and its incoming and outgoing hyperlinks displayed as arcs or wires overlaid atop the pyramid.

We would also like to experiment with placing documents on a plateau into clusters according to similarity of their content.

Other applications of Information Pyramids include the visualisation of decision trees such as those in MineSet [4], or the visualisation of hierarchies constructed by traversing hyperlinks from a selected initial web page such as those in the Navigational View Builder [8].

In order to gain an appreciation of the merits and drawbacks of Information Pyramids compared to some of the other hierarchical visualisation techniques described in Section 1, we intend to run a number of informal usability tests.

5 CONCLUDING REMARKS

We have presented the Information Pyramids technique for visualising and interacting with large hierarchies. Although this work is still very much "in progress", we believe that it will become a valuable tool for hierarchical navigation.

6 ACKNOWLEDGEMENTS

We would like to thank our colleagues at the IICM for their support and suggestions during this work. The name Information Pyramids is a trademark of IICM.

REFERENCES

[1] 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.ps.gz`.

[2] 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, CA, October 1996. `ftp://ftp.iicm.edu/pub/papers/ivis96.ps.gz`.

[3] 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`.

[4] Barry G. Becker. Using mineset for knowledge discovery. *IEEE Computer Graphics and Applications*, 17(4):75–78, July/August 1997. `http://www.sgi.com/Products/software/MineSet/`.

[5] Hyperwave. `http://www.hyperwave.com/`.

[6] 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.

[7] 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`.

[8] Sougata Mukherjea, James D. Foley, and Scott Hudson. Visualizing complex hypermedia networks through multiple hierarchical views. In *Proc. CHI'95*, pages 331–337, Denver, Colorado, May 1995. ACM. `http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/sm_bdy.htm`.

[9] George G. Robertson, Jock D. Mackinlay, and 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] Ben Shneiderman. Tree visualization with tree-maps: A 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, January 1992.

[11] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. 1996 IEEE Symposium on Visual Languages*, pages 336–343, Boulder, Colorado, September 1996. IEEE Computer Society. `ftp://ftp.cs.umd.edu/pub/papers/papers/3665/3665.ps.Z`.

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