Goldleaf Hierarchical Document Browser

Jolon Faichney and Ruben Gonzalez

School of Information Technology, Griffith University
PMB 50 Gold Coast Mail Centre, QLD 9726, Australia
{J.Faichney,R.Gonzalez}@gu.edu.au

Abstract

A two-dimensional, zoomable, space filling user interface is presented for browsing conventional, hierarchical file systems. Through user studies the Goldleaf browser was compared with the widely used Microsoft Windows Explorer user interface. The times and number of mouse clicks to locate directories and files were recorded. The user studies found that the Goldleaf browser required less than half the mouse clicks to locate a directory compared with Windows Explorer. Through the use of document thumbnails subjects were able to locate documents in less than two-thirds the time that it took using Windows Explorer. A majority of subjects felt that the ability of the Goldleaf browser to display multiple levels of the file system simultaneously was its most beneficial feature in completing the tasks. Subjects found that the Goldleaf browser required less mental and physical effort and was more enjoyable to use than Explorer.

1 Introduction

Since the introduction of the graphical user interface the number of documents that users are required to manage has increased by several orders of magnitude while user interface technologies and display sizes have only improved marginally. A majority of users currently manage their files using Microsoft Windows Explorer or similar user interfaces. The Explorer user interface allows users to locate directories and files by progressively expanding directories to reveal their contents. Context is preserved through an outline structure showing the user where they currently are in the file system. However, the user interface provides little context for where the user can go, makes poor use of the screen real estate, and provides very little indication of document contents.

A user interface has been developed to address these problems. The Goldleaf browser maximises the use of the available screen real estate to represent the tree structure of the file system (Figure 1). The radial layout of subfold-

ers allows users to see multiple levels in the file hierarchy without using the mouse and also takes advantage of spatial memory. Conventional user interfaces have used icons to represent file content but generally only represent the type of content rather than the content itself. Goldleaf uses large, document thumbnails which are stacked for maximum visibility.

In this paper we describe other research involving file browsing systems. A detailed description of the proposed user interface is provided along with user studies and a discussion of the results. We conclude with possible directions for future research.

2 Document browsing user interfaces

One of the greatest challenges of user interface design is to simultaneously provide the user with context as well as detail. Context and detail are often shown simultaneously through distortion techniques [5] such as fish eye lens, fractal views [3], hyperbolic geometries [4], and 3D perspective [11, 9]. Other techniques include the event horizon user interface [14] for small devices and overlaying layers of detail using translucency [6]. These techniques dedicate more screen real estate to the objects in focus whilst providing as much context as possible by scaling down the surrounding objects.

Distortion techniques for providing context and detail distort a two-dimensional plane. Two-dimensional user interfaces allow users to use spatial memory to remember where objects are rather than recognising names or icons. Spatial memory is especially useful for navigating structures such as trees and graphs [2]. However, spatial memory is sensitive to orientation [13] and distortion. Therefore user interfaces which preserve orientation and distortion will perform better when navigating structures.

The *Cone Tree* [12] user interface presents a fixed, rotatable, three-dimensional tree to the user. Cone Tree provides the user with the context of where they are and where they can go. Unfortunately, rotating the tree may disorient the user and the authors suggest that non-uniform trees

Figure 1. The Goldleaf user interface.

(a)

(b)

Figure 2. (a) Minimised folder and (b) Maximised folder.

should be used. The Cone Tree also provides little information about the content of documents other than the file name. Application examples are provided but no objective results are presented comparing Cone Tree's performance with other file browsers.

The zoomable *Pad++* [1] user interface has been used to represent file hierarchies with directories being represented as squares and subdirectories also displayed as squares within the parent. A user may select a directory and zoom into it. The *Pad++* user interface also displays file thumbnails which can be zoomed into to view the file. *Pad++* has the advantage that changing the focal point does not change the relative structure of the tree. However, no distinction is made between subdirectories, which represent the structure, and files which represent the content. No user studies were reported for the *Pad++* user interface.

Other user interface studies have found file thumbnails to be effective for locating and organising documents [7, 8, 10].

3 The Goldleaf user interface

A user interface is required which retains a fixed structure to take advantage of spatial memory, whilst providing context and detail, and a distinction between the structure of the tree and the documents within the tree. The user interface must make maximum use of the available screen real estate and also provide an indication of document contents.

A distinction between folders and files is important because the maximum number of folders in a directory is usually less than the maximum number of files. User interfaces such as Cone Tree [12] or the Hyperbolic browser [4] display both files and folders in the tree structure increasing the branching factor of nodes by a significant amount making it harder for the user to see the structure. The Goldleaf browser only displays folders in the tree structure while files are displayed within folder windows similar to conventional GUI windows. Subfolders are arranged around the parent folder in a radial layout (Figure 1).

A radial layout was chosen because it provides better use of screen real estate than a more conventional tree layout which displays all nodes in the same level of the tree across one dimension. For example, in Figure 1, 35 level-2 folders are displayed simultaneously using a 1024×768 pixel display. Each of these folders is approximately 60 pixels wide and 43 pixels high. If these folders were laid out in one dimension they would occupy 2100 pixels horizontally or 1505 pixels vertically with no spacing between nodes. In contrast, in Figure 1 there is ample spacing between nodes. With the 7 level-1 folders shown in Figure 1 up to 52 level-2 folders of size 60×43 pixels could be displayed simultaneously, far exceeding the display capabilities of conventional tree layouts. However, one major disadvantage of the ra-

dial layout is that the alphabetical ordering of subfolders becomes less obvious because there is no visual indication as to where the list starts.

The relative sizes and positions of folders and subfolders remain rigid as the user navigates through the tree structure. When a user reaches a folder and wishes to explore the files in the folder they may also maximise the folder which also reduces the screen space occupied by subfolders so that the files in the current folder can be more easily explored.

Context for where the user can go is provided by displaying multiple folder levels simultaneously, filling the screen in two dimensions. The maximum depth that the user can see from the current folder is dependent on the branching factor of the tree. A high branching factor provides less screen space for subfolders and hence will diminish to one-pixel or invisible folders more quickly. Context for where the user has been is provided by the parent stack which is displayed to the left of the tree. Users may select a folder in the parent stack to zoom directly out to that parent folder. As the user explores the tree folders are added and removed from the parent stack.

3.1 Layout

The size and location of folders depend on the number of sibling and child folders. The more subfolders a folder has, the smaller they will appear in the display so that they will all fit. The user interface is most effective when the subfolders can be displayed as large as possible providing maximum context for where the user can go, which is achievable with 10 or fewer subfolders. The user interface is still very usable with more than 10 subfolders because subfolders are large enough to click on and folder titles are still readable. However, with more subfolders it will be more difficult for the user to see and navigate to folders that are a few levels deeper because the amount of screen space they occupy is smaller.

Folders which have no children are sized as large as possible so that the file thumbnails will be clearly visible. Folders with children will generally be displayed quite small, so the user may choose to maximise the folder which also shrinks the size of the surrounding subfolders (Figure 2). Folders a number of levels deep will appear very small on the display and not enough room will be available to display their names. Moving the cursor over these smaller folders will temporarily display their names in a tooltip.

Files can be laid out within a folder using a stacked or spread out layout (Figure 3). The stacked layout allows the user to see the names of files as well as some of the file's thumbnail. Moving the cursor over a file in the stacked view will move covering files down to reveal the file's thumbnail. Selecting a file zooms into the file and reveals its content. Currently HTML, plain text, JPEG, and GIF files are sup-

(a)

(b)

Figure 3. (a) Stacked folder and (b) spread out folder.

ported.

3.2 Animation

All user interface changes are animated using an accelerate-decelerate animation model. Animation durations were chosen to be long enough so that the user can understand what is happening but also fast enough so that the user doesn't feel as though they are waiting. Animations are performed in separate threads, and multiple animations can occur simultaneously.

3.3 Progressive loading

Not all of the file system structure and file thumbnails can be loaded simultaneously because of memory requirements. Instead, only four levels of the tree are loaded at one time. File and thumbnail loading occurs in separate threads and the display is updated as each level is loaded. To conserve memory, tree pruning is also performed when the user returns to a parent folder. Tree pruning involves a depth-first search through all subfolders of the current folder. Folders that are four levels deep relative to the current folder have all of their children removed and after pruning the Java garbage collector is invoked.

4 User study

The purpose of the user study was to determine whether the Goldleaf browser is more effective for locating documents organised in a hierarchical tree compared with the conventional Explorer user interface.

Five undergraduate and postgraduate computer science students tested the user interfaces for over an hour. A sixth

subject was not able to complete the experiment in the same day which affected his results (one task took 90 seconds when the maximum time for the other subjects was less than 21 seconds). So the results for the sixth subject were not included. Each subject completed the experiment individually at different times rather than being in a group. Each subject was required to locate one of 16 preselected documents which had been printed and shown to the subject. The subjects were not shown document file names only the print outs of the contents of the document. The documents were chosen from levels 3, 4, 5, and 6 in the file hierarchy (with root at level 0). Before the testing began the subjects were encouraged to familiarise themselves with the directory structure and were also notified of the two main locations where the documents would be located (a directory containing clip art and a directory containing web pages). The users were also walked through the features of the Goldleaf user interface.

The documents were divided into four tasks, each task containing a document from each level (Table 1). No breaks were given between tasks. Two tasks were to be completed as a tutorial to familiarise the user with the types of tests, the user interfaces, and the directory structure. One of the remaining tasks was tested with Explorer and the other tested with Goldleaf. After the first two tasks, subjects should be familiar with the directory structure and the last two tasks should be relatively unbiased. Every alternate subject had the tasks swapped between Goldleaf and Explorer. Each subject was required to locate each document 3 times to test the effects of spatial memory. After each document was located the user returned to the root directory in both browsers and in Explorer the root directory was also expanded. The user was able to choose their own way back to root. In Goldleaf this is most easily achieved by clicking the folder

(a)

(b)

Figure 4. (a) Average number of mouse clicks and (b) times to reach the correct folder.

at the bottom of the parent stack. In Explorer the users had to also collapse all expanded folders before reaching the root or alternatively open a new Explorer window.

Times as well as mouse clicks for the subject to locate the correct folder and correct file were manually recorded by one of the authors using a stop watch. Once the user was in the correct directory the stop watch was split. The stop watch beep was also a cue notifying the subject that they were in the correct directory.

The Goldleaf user interface uses single clicks exclusively, whereas Explorer may also use double clicks. To simplify comparisons double clicks were treated as a single mouse click. Users were stopped if they couldn't find the correct directory in 90 seconds. The time and number of mouse clicks were recorded at this point and the user was shown the correct directory containing the desired file. The time and number of mouse clicks were then recorded for finding the correct file within the directory.

The documents tested included HTML, JPEG, and GIF documents. One directory included a selection of clip art images whilst another contained files from a web site. The clip art images had meaningless names but were organised in meaningful folders whilst the web pages had meaningful names. Test documents were carefully selected so that some would occur at the beginning, middle and end of alphabetical lists. For the Explorer test, Internet Explorer was used to view the files. Goldleaf is able to display the files within its own user interface.

After the subjects completed the tasks they were asked 8 subjective questions:

1. What aspect of the Goldleaf browser was most beneficial in completing the tasks?
 - (a) Displaying multiple levels simultaneously
 - (b) Image thumbnails

- (c) Zooming animation
- (d) Stackable documents
- (e) Parent stack
- (f) Other

2. What aspect of the Goldleaf browser was least beneficial to completing the tasks?
3. What aspect of Windows Explorer was most beneficial to completing the tasks?
4. What aspect of Windows Explorer was least beneficial to completing the tasks?
5. Would you choose the Goldleaf browser or Windows Explorer to perform similar tasks in the future?
6. Which browser was more enjoyable to use?
7. Which browser required the least mental effort?
8. Which browser required the least physical effort?

5 Results

Goldleaf performed better than or equivalent to Explorer in all of the objective experiments. The most impressive difference between Goldleaf and Explorer was the number of mouse clicks required to locate the correct directory. Goldleaf consistently required less than half the mouse clicks required by Explorer to locate the correct directory (Figure 4a). For level 6 directories, half the attempts using Goldleaf only required two clicks compared with over 6 clicks using Explorer to locate the correct directory. On the third repetition users were able to navigate between 2 and 3 levels with a single click using the Goldleaf user interface.

Table 1. Test documents and tasks.

	Task 1	Task 2	Task 3	Task 4
Level 3	SSGP1594.JPG	SSGP1595.JPG	SSGP0078.JPG	SSGP0210.JPG
Level 4	B8175.JPG	B8176.JPG	SSGP0401.JPG	SSGP0530.JPG
Level 5	about.html	events.html	progress.html	research.html
Level 6	01START.JPG	02.NEWPRJ.JPG	13BUILD.JPG	14OUTPUT.JPG

Figure 5. Average times to reach level 3 folders.

Explorer required one click for every level to the destination folder.

Locating level 3 folders in Goldleaf was noticeably faster than Explorer (Figure 5). Goldleaf has the advantage that level 3 folders can be easily be accessed with a single click and over half of the attempts were able to locate the correct level 3 folder with a single click. Higher level folders generally required more than a single click because they are smaller and may not have labels displayed. Locating folders for the first time was faster in Goldleaf (Figure 4b) but the second and third times were almost identical to Explorer. However, the Goldleaf times also include animation delays of 0.8 seconds for each folder change. With between 1 and 4 clicks to locate a correct folder, between 0.8 and 3.2 seconds will be spent waiting for the animation to complete. Hence, Goldleaf requires less time for mental and physical effort. Some subjects expressed that shorter animation durations would be useful.

Locating files for the first time was noticeably faster and required fewer mouse clicks using Goldleaf but the difference was negligible for subsequent file locations (Figure 6). Clip art file names were numeric and didn't aid users in locating the correct file. Some web page file names were also ambiguous making it difficult to determine the correct file based on name alone. Subjects generally used the strategy of opening every file when using Explorer, except for one subject who selected all files in the directory and opened

them at the same time. Subjects were able to locate the correct file with a single click using Goldleaf once the thumbnails had loaded. Some subjects had difficulty selecting the correct document in the stacked view. As the cursor rolls over the documents they move out of the way to reveal the current document. However, this made it difficult for subjects to click the correct file if it was moving away and sometimes another document was selected, requiring more clicks and time to locate the correct document. One subject suggested using a short delay before a document was revealed to stop spurious document movements.

The results from the subjective questionnaire favoured the Goldleaf browser overall. Four out of five subjects felt that the feature most beneficial to completing the tasks was the ability to display multiple levels simultaneously. Aspects of the Goldleaf browser that subjects liked the least included the zooming time delay and small folders that were difficult to select. Only one of the subjects used the feature to spread the files out whilst the others felt that the stacked view with file revealing was more than adequate. All subjects thought that the Goldleaf browser required less mental and physical effort (reflected by the number of mouse clicks) and was more enjoyable to use.

Subjects found that the most beneficial features of Explorer were their familiarity with it, no delay when expanding folders, and one subject found the widgets to expand directories most beneficial. However, two other subjects found the expander widgets the least beneficial feature of Explorer. Subjects also complained of the difficulty of selecting Explorer items because of their small size. In addition, a double click using Explorer may result in the name of a file or folder being changed rather than opening the file or folder. After using Goldleaf, users preferred using a consistent single click model rather than the sometimes double click or single click model of Explorer. One subject also found collapsing directories in Explorer a chore if they had taken the wrong path.

Most subjects, when presented with the Goldleaf browser for the first time, knew how to use most of the features of the user interface without any prior instructions. Subjects expressed that even though the layout of folders is unfamiliar initially compared with existing user interfaces, the operation of the user interface was very intuitive. In addition, subjects expressed that the enjoyability of the user interface was a major deciding factor for whether they

(a)

(b)

Figure 6. (a) Average number of mouse clicks and (b) times to reach the correct file.

would use Goldleaf or Explorer for similar tasks in the future.

6 Conclusions

Our user study has shown that the Goldleaf user interface requires less clicks and less mental effort for locating files than Explorer. We found that users were able to jump on average 2.3 levels with a single click using Goldleaf but only one level using Explorer. Users are constrained with Explorer because only one level of subfolders is viewable at a time. The reduced clicks required by Goldleaf can be attributed to the simultaneous display of multiple levels of the file system. Goldleaf also requires less clicks when selecting the correct file if its name is not known because the user can see file contents in the thumbnail.

Our experiments were limited by a number of factors. Firstly, our comparison for locating files was based on knowing the file contents rather than the file name. The result is that on the first attempt to locate a file Goldleaf had a clear advantage because it provides file thumbnails. The files that could be displayed were limited to HTML files and images which are quite distinguishable through thumbnails. Other types of documents such as source code, spreadsheets, and databases may not be as distinguishable through thumbnails. These types of documents may benefit from semantic thumbnails which display salient attributes of the documents.

Secondly, our study used an artificial directory structure which had a branching factor of no more than 8 subfolders per node which is ideal for Goldleaf. However, typical users' file systems contain much larger branching factors per directory. We would argue that the reason for such large branching factors is a result from current user interfaces, either command line or GUI, which make it difficult for users

to navigate to more than 6 levels. The result is that users are encouraged to create shallow directory structures designed for list-based user interfaces. Further work on the Goldleaf browser will include adding the capabilities to create, edit, move, and delete files and folders. A study will be conducted to determine whether users will create deeper and more meaningful hierarchies if they can be browsed using a more productive user interface such as Goldleaf.

The Goldleaf software may be downloaded from <http://www.int.gu.edu.au/~j.faichney/goldleaf/Browser.html>.

Acknowledgements: We would like to thank Sidhi Kulkarni, Justin Simpson, Andrew Stevenson, Tom Tilley, Michael Blumenstein, and Paul Calvino who dedicated their precious time evaluating the Goldleaf user interface and the reviewers who provided invaluable input.

References

- [1] B. B. Bederson and J. D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *ACM UIST '94*, pages 17–26, 1994.
- [2] C. Chen and M. Czerwinski. From latent semantics to spatial hypertext – an integrated approach. In *The Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, pages 77–86, 1998.
- [3] H. Koike. Fractal views: A fractal-based method for controlling information display. *ACM Transactions on Information Systems*, 13(3):305–323, July 1995.
- [4] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *ACM CHI'95*, pages 401–408, 1995.
- [5] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, June 1994.

- [6] H. Lieberman. Powers of ten thousand: Navigating in large information spaces. In *ACM UIST'94*, pages 15–16, November 1994.
- [7] P. Lucas and L. Schneider. Workscape: A scriptable document management environment. In *ACM CHI'94 Conference Companion*, pages 9–10, April 1994.
- [8] R. Mander, G. Salomon, and Y. Y. Wong. A 'pile' metaphor for supporting casual organization of information. In *ACM CHI '92*, pages 627–634, 1992.
- [9] R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of ACM CHI'94*, pages 318–482, 1994.
- [10] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich. Data mountain: Using spatial memory for document management. In *ACM UIST'98*, pages 153–162, 1998.
- [11] G. G. Robertson and J. D. Mackinlay. The document lens. In *ACM UIST'93*, pages 101–108, November 1993.
- [12] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *ACM SIGCHI'91*, pages 189–194, April 1991.
- [13] B. Roskos-Ewoldsen, T. P. McNamara, A. L. Shelton, and W. Carr. Mental representations of large and small spatial layouts are orientation dependent. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24(1):215–226, 1998.
- [14] A. Taivalsaari. The event horizon user interface model for small devices. Technical Report SMLI TR-99-74, Sun Microsystems, March 1999.