Improving Navigation-Based File Retrieval

Stephen Fitchett
Dept. of Computer Science
University of Canterbury
Christchurch, New Zealand
saf75@cosc.canterbury.ac.nz

Andy Cockburn
Dept. of Computer Science
University of Canterbury
Christchurch, New Zealand
andy@cosc.canterbury.ac.nz

Carl Gutwin
Dept. of Computer Science
University of Saskatchewan
Saskatoon, Canada
gutwin@cs.usask.ca

ABSTRACT

Navigating through a file hierarchy is one of the most common methods for accessing files, yet it can be slow and repetitive. New algorithms that predict upcoming file accesses have the potential to improve navigation-based file retrieval, but it is unknown how best to present their predictions to users. We present three design goals aiming to improve navigation-based file retrieval interfaces: minimise the time spent at each hierarchical level en route to the target file; reduce the number of levels traversed by providing shortcuts; and promote rehearsal of the retrieval mechanics to facilitate expertise. We introduce three interfaces that augment standard file browsers based on each of these goals: *Icon Highlights* give greater prominence to predicted items in the current folder; *Hover Menus* provide shortcuts to predicted folder content; and *Search Directed Navigation* uses predictive highlighting to guide users through the hierarchy in response to query terms. Results from a user evaluation show that all three interfaces improve file retrieval times, with Icon Highlights and Hover Menus best suited for frequently accessed items and Search Directed Navigation best suited for infrequent ones. We also show that the benefits are larger when folder content is spatially unstable. Finally, we discuss how the interfaces could be combined and deployed in existing file browsers.

Author Keywords

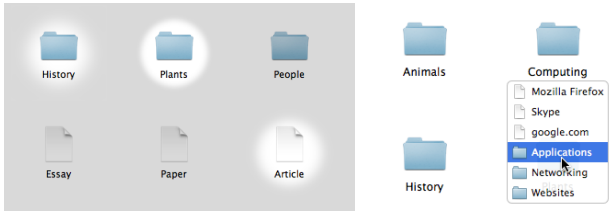
Revisitation; prediction; file retrieval; file navigation

ACM Classification Keywords

H.5.2 User Interfaces: Theory and methods

INTRODUCTION

Retrieving files is an extremely common task for all computer users. Of the many interfaces that are possible for retrieving files, navigating through a hierarchy using a file browser is dominant (used more than 60% of the time, according to previous studies [5]). However, file browser navigation is slow – more than 12 seconds per retrieval for Mac users and more than 17 seconds for Windows users [5]. This is a long time to retrieve a file, given that selecting a ready-to-hand file icon would take no more than a second or so. There are two main reasons for these long completion times: people may not know where a file is, resulting in extra time to explore the file system; and there are numerous navigation actions required (remembering folder names, finding icons in the current display, and clicking folders to open them). Regardless of the cause, it is clear that reducing retrieval time for hierarchical file browsers could result in large aggregate time savings.



(a) Icon Highlights (b) Hover Menus

Figure 1: Icon Highlights (left), showing 'Plants' as the most likely AccessRank prediction. Hover Menu (right) showing predictions under 'Computing', with $n = 3$.

The general problem of improving file retrieval has received substantial research attention. As reviewed below, researchers have studied several aspects of the problem: the ways that users choose to organise information [24, 4], the performance implications of different hierarchical structures [22], potential improvements to file access using search [9], visualisations that provide shortcut access to files [28], and predictive algorithms for anticipating retrievals [12]. Commercial systems have also iteratively refined their facilities, with tools such as 'Recently Used' menus, full text searching, and folder aliases now standard in most operating systems.

Although some alternative retrieval techniques explored in this previous work have been shown to be faster than standard navigation [28], the performance improvement often comes at the cost of switching to a completely different retrieval interface. This is a problem, because several studies have shown people's continued preference for hierarchy-based file navigation over alternative methods (e.g., [2, 3, 7]). Bergman et al. [3] give four explanations for this preference: first, the locations and mechanisms of navigation-based retrieval remain consistent and reliable, whereas the organisation and content of search results can vary from one retrieval to the next; second, navigation reduces cognitive load because users can rely on recognition of the steps toward the target rather than needing to recall file attributes; third, the mental and physical mechanisms used for retrieval may become partially automated due to their consistency, allowing users to remain focussed on their work; and fourth, the location-based mechanisms of hierarchical containment are familiar from the real world, which may serve an important sense-making function.

CHI 2013, April 27–May 2, 2013, Paris, France.

In contrast, research has shown that search, although an essential tool for some retrievals, is used mainly as a method of 'last resort' [26, 3], called upon when users cannot remember the location of files in the hierarchy.

Therefore, to solve the performance problems of real-world file retrieval, it is necessary to find improvements that are able to work with the presentation styles and interaction models of the standard hierarchical file browser. In this paper we present three new techniques that do this. Our new techniques are based on design goals for overcoming three performance constraints in navigation-based retrieval:

1. Overcome the *visual search* constraint: minimise the time spent at each hierarchical level, by reducing exploration and visual search;
2. Overcome the *levels* constraint: reduce the number of levels traversed, by facilitating shortcuts;
3. Overcome the *practice* constraint: improve navigation expertise, by promoting rehearsal of the retrieval mechanics.

Our three new techniques address these constraints using some of the same advances seen in previous work for prediction-based and search-based approaches (e.g., [12, 9]), but they differ dramatically in the way that the algorithms' results are presented to the user and the way that the user interacts with the new capabilities – in our techniques, results are tightly integrated into the interface and interaction paradigm of the file browser.

To reduce step times during file navigation (approach 1), *Icon Highlights* (Figure 1a) predict which items in the current folder are most likely to be accessed, and give them greater visual prominence. *Hover Menus* (Figure 1b) provide quick access to commonly accessed items inside folders, in order to reduce the number of steps required in many cases (approach 2). *Search Directed Navigation* (approach 3) guides users through a file hierarchy based on a filename query. This is designed to bring some of the advantages of search to file navigation, while facilitating the development of expertise when compared to search (approach 3).

A two-part experiment validates our new techniques. The first part of the study examines user performance and preference with the techniques in comparison to standard file browsers, using tasks that involve retrieving both previously-visited and unvisited files; it also examines how well users learn file locations with the techniques. The second part of the study examines the effects of spatial stability of the folder contents on the relative performance of the techniques, which is important when folder content varies and when view modes change. The study's results show that the new techniques provide substantial performance improvements over a standard file browser, and that they are strongly preferred by users.

The specific contributions of the paper are as follows: first, three design goals for improving navigation-based file retrieval; second, three new interface designs for file navigation that improve different aspects of navigation-based retrieval using predictive and search-based algorithms; third, empirical results demonstrating that these interfaces improve navigation-based retrieval; and fourth, discussion of real-world deployment of the techniques, including description of how to combine them into a single file browsing interface.

RELATED WORK

There is extensive prior literature on file retrieval. The following sections briefly review key contributions on how users organise and retrieve files, the impact of structure, interfaces for file retrieval, and algorithms that can assist with retrieval.

How users organise and retrieve files

To successfully retrieve a file using navigation, users must remember its location (either by recall or recognition of folder names). However, organisational schemes are rarely optimal, raising challenges such as multiple possible locations due to semantic ambiguity, and the lack of *a-priori* knowledge that a file will be needed in the future (e.g., wanting to later retrieve a file initially stored in a transient 'downloads' folder).

Nardi et al. [26] describe three types of information that is organised in different ways: ephemeral information such as memos that is temporary; working information that is frequently used and relevant to current work; and archived information that is accessed infrequently. However, files often move between these categories, creating a tension between organisation for current and later reuse [20].

To examine the retrieval difficulties stemming from semantic ambiguity, Malone [24] asked study participants to find documents with descriptions given by coworkers. In two thirds of these probes, the documents were not filed under the dimensions (such as title, author or person it was about) used to describe them. Furthermore, he observed that the cognitive difficulty of classifying information acted as a barrier to filing. Several other studies have analysed snapshots of users' hierarchies to get a better idea of how they organise their files [16, 18], also finding that most files are rarely accessed and that a majority of retrievals are made up of a small minority of items [16, 30].

The performance impact of structure

The influence that hierarchical structure has on navigation time has been extensively researched, and is broadly encapsulated by the question 'broad and shallow or narrow and deep?' [25, 22]. Cockburn and Gutwin [8] provide a review of thirty years of empirical research on the topic, which shows diverse results. Several studies show that plots of navigation time against depth follow a 'U-shape', while others show that time increases with depth. Cockburn and Gutwin also present a simple mathematical performance model, called 'Search/Decision and Pointing' (SDP), that explains the result's diversity: broadly, a U-shape occurs if users must visually search for items at each level of the hierarchy, while shallow structures perform best when users can anticipate target location at each level.

The SDP model inspired the design goals presented later, and so it is briefly summarised here. SDP predicts the time taken to select an item at one level of a hierarchy by combining three factors: the time to visually *search* for the target; the time to *decide* about target location; and the time to *point* to it. Pointing time is modelled using Fitts' Law [13]. Visual

search is modelled as a calibrated linear function of the number of candidate items, and it is employed when users have no basis for anticipating a target's location (e.g., the user is a novice or the interface is unpredictable). Decision time is modelled as a calibrated logarithmic function of the number of candidate items, and is employed when users can anticipate target location (e.g., the user is experienced with a stable display, or the interface presents a predictable dataset, such as an alphabetic list of items). The model uses a power-law to predict the user's transition from search-based strategies to decision-based ones when interfaces offer consistent access methods across retrievals. Finally, the model sums the predicted time for each level across hierarchical levels.

The SDP model suggests three promising opportunities for improving performance in hierarchical navigation, described later: 1. reduce visual search time, which can be a performance bottleneck due to its linear function of candidate items; 2. reduce the number of hierarchical steps required through the hierarchy; and 3. help users transition from relatively slow search-based strategies to faster decision-based ones.

Contemporary commercial interfaces provide a variety of techniques for reducing the number of hierarchical steps to reach files. These include 'Smart folders' in OS X, 'Libraries' and 'Favourites' on Windows 7, and 'aliases', 'symbolic links' and 'shortcuts' in various operating systems. Methods to reduce visual search time or to facilitate transitions to decision-based retrieval are rarer, but research exemplars are described below.

File search and enhanced interfaces

Search is an essential tool for file access, particularly when item locations are unknown [3]. Search has many attractive features for file retrieval: any file attribute can be searched, rather than requiring memory of the item's location [23]; it does not depend on a hierarchy, relieving users from the need to develop semantic groupings before storage and assign files to them; and it enables retrieval in a single step [3], potentially allowing for significantly faster retrievals.

However, researchers have noted that search imparts a lower sense of control and that it relies on recall rather than recognition [2, 3], which elevates cognitive demands [29]. Consequently, it is often used as a method of last resort, applied after failing to retrieve the file by navigating [2, 3, 6, 7]. Additionally, search results are typically presented in list form, which facilitates rapid access to the target, but raises two potential problems. First, it does little to assist users in learning the navigation-based retrieval mechanisms that are likely to be tried first in any future access [3]. Second, search results lists can be ambiguous when multiple candidates match or partially match the search criteria.

Predictive algorithms and AccessRank

Recommender systems can assist file retrieval by predicting upcoming accesses. Two categories of recommender systems are those that base predictions on an individual's past actions (e.g., recency or frequency of access; briefly reviewed in [12]), and those that facilitate discovery based on retrievals across a community of users (e.g., [15]). Community-based recommenders are of limited use in personal information management where users have unique collections of files.

The common 'Open recent' menu in commercial software typically uses a simple 'Most recently used' (MRU) algorithm to allow quick access to the most recently used files within an application. However, the contents of the list fluctuates over time and, like search, it does not help users learn the retrieval methods that they are likely to need for future accesses. The AccessRank algorithm [12] was developed to facilitate both accurate predictions *and* prediction consistency over time. AccessRank uses two parameters α and δ to configure a balance between prediction accuracy and consistency. α determines the mix between two predictive models, one which is more accurate but produces a prediction list that fluctuates more over time, and another which produces slightly less accurate results but with higher prediction consistency. δ determines a score delta that must be overcome for an item to overtake another in a prediction list, increasing prediction consistency. Log-based analyses of real user data for web browsing, command use, and window switching, showed that AccessRank outperforms prior predictive algorithms for both accuracy and consistency. The systems described later in this paper use AccessRank as the back-end algorithm for new interfaces that present its predictions.

Although log-based analysis demonstrated that AccessRank improved prediction accuracy over previous techniques, achieving benefits in real-world file access is difficult. One primary problem is that predictive and search-based systems typically present a completely different interface for retrieving files than a standard file-navigation browser, however most users prefer the navigation-based interface [2, 3, 7].

Bergman et al. [3] provided several reasons for this preference, including familiar locations, a lower cognitive load, a degree of automaticity and familiarity with the real world. In contrast, search-based or prediction-based retrieval is often accompanied by new interfaces that need to be learned, widely varying search results, and considerable cognitive effort to compose and debug queries.

These realities about the way users retrieve files suggest that real-world performance could best be improved through a hybrid approach that takes the power of search and prediction algorithms and integrates it with enhanced presentation schemes within the interaction of the existing file browser.

IMPROVED FILE NAVIGATION: GOALS AND INTERFACES

The review of related work suggests several opportunities for improving human performance in navigation-based file access. In the following subsections, we distill these opportunities into three design goals, and then present three interfaces aiming to satisfy the goals. It is important to note that any file retrieval mechanism will have three parts: an underlying algorithm for deciding which items to present, a presentation approach for making items visible and salient, and a set of interaction techniques with which the user can select items and specify targets. Our primary focus is on the presentation aspects of the techniques, but some elements of the algorithms and interaction techniques will also be discussed as needed.

Design Goals

The design goals stem from insights within the 'Search Decision and Pointing' model [8] which predicts hierarchical navigation time based on the time taken at each hierarchical level ('step duration'), the number of levels traversed ('step count'), and the potential for the user to make a transition from novice to expert behaviour.

Goal 1: Minimise time spent at each hierarchical level

The SDP model suggests three possibilities for reducing step duration, by improving human performance in a) searching for targets, b) deciding about them, and c) pointing to them. Extensive prior literature has investigated improved pointing techniques ((c), see [1] for a review), and Goal 3 addresses the transition to decision-based methods (b). In Goal 1, therefore, we focus on mechanisms that improve visual search (a). This is particularly important because visual search can be a performance bottleneck [8].

Reducing visual search time involves two activities. First, the primary job of the retrieval technique's underlying algorithm is to determine a small set of likely candidates for the current retrieval task. Second, the technique's presentation approach must make those candidates visible and salient, to reduce the number of objects that need to be visually inspected.

Prior work on methods for improving the presentation of candidates includes a variety of highlighting mechanisms that enhance the visual salience of likely targets without changing their layout, such as ephemeral adaptation [11]. Other techniques segment the visual presentation across space or time, by moving salient items into prominent positions (e.g., split menus [27]), by segmenting images (e.g., [14]), or by rapidly presenting images across time (e.g., [10, 14]).

Icon Highlights, described below, uses AccessRank to predict probable items at each level of the hierarchy, and uses in-place highlighting to increase their visual salience.

Goal 2: Provide shortcuts to reduce levels traversed

The SDP model also suggests that efficiency gains can be achieved by reducing the number of hierarchical levels traversed en route to the target. Users might achieve this by creating flatter hierarchies or by creating shortcut links to important files and folders, but both require a-priori knowledge of the future need to retrieve the file, as well as an understanding of the file structure's implications on retrieval times.

Hover Menus, described below, also uses AccessRank to determine a set of likely candidates; but this time it creates these sets for each of the folders at the current level, thus indicating what is most likely within that folder's sub-hierarchy. The candidates are presented in a context menu associated with each folder, allowing users to skip one or more levels and get shortcut access directly to the most probable child targets in any branch of the sub-hierarchy.

Goal 3: Promote rehearsal to facilitate expertise

Kurtenbach [21] argues that expert performance is best facilitated when the actions that a novice uses for an interaction are a physical rehearsal of the mechanisms used when expert. Goal 3 adapts Kurtenbach's recommendation for file retrieval. While navigation-based retrieval naturally supports the goal by using consistent interaction mechanics across experience, search-based retrieval does not. Instead, search-based interfaces typically present their results to the user in a list, allowing direct access to each file. While this may facilitate rapid retrieval for the current access, the list presentation does not help users learn how to find the file in the navigation-based hierarchy (which, as described above, is often the preferred mechanism of retrieval).

Note that there is a tension between the recommendations of Goal 2 and Goal 3 because the presence of shortcuts allows alternative mechanisms for retrieval, and consequently it may reduce opportunity for rehearsal, particularly if the shortcuts vary due to fluctuations in prediction consistency.

Search Directed Navigation, described below, allows users to type characters to determine the set of likely candidates based on filename matches. The presentation approach uses item highlighting (similar to Icon Highlights) to guide users through the hierarchy, with the intention of facilitating rehearsal-based transitions to more expert performance.

File Navigation Interfaces

We developed a basic file browser that closely mimics the icon view of the OS X Finder. We then developed three interfaces that augment the basic browser with new features that are intended to satisfy each of the design goals.

As described above, each technique can have an underlying algorithm, a presentation approach, and various interaction techniques. Two of the interfaces, *Icon Highlights* and *Hover Menus*, use AccessRank [12] as their underlying algorithm to predict likely folders and files (both use AccessRank parameters $\alpha = 0.8$ and $\delta = 0.5$, see review above). *Search Directed Navigation* uses character-based filtering to determine candidates, and also keeps track of prior revisitations. Search-Directed Navigation uses an explicit interaction technique – that is, users type characters to specify the search target. In contrast, the AccessRank-based techniques use implicit information about selection and navigation history, rather than user input about the target. The sections below provide details of the presentation approach for the three retrieval techniques.

Icon Highlights: minimising search time

Icon Highlights increase the visual salience of items that AccessRank predicts, as shown in Figure 1a. The highlighting mechanism is similar to that used in Apple's System Preferences application, with graduated blurring of 'spotlight' highlighting dependent on the item's AccessRank score (crisp presentation for probable items, blurred for less probable ones). The highlighting scheme can be applied to any folder view, such as large or small icons/thumbnails, or filename lists. All items remain selectable, regardless of their highlight state. To indicate the presence or absence of highlighted items outside the current scroll view, the design marks items in the horizontal or vertical scrollbar [19].

Details of the highlighting and blurring algorithm are briefly summarised to aid replication of the technique. First, a minimum blur level b_{min} is calculated, based on the largest item score in the folder as a proportion of the total scores (p_{max})

as well as the total number of accesses (n) of items within the folder – see Equations 1 to 3. Next, blur levels are calculated for the top k AccessRank prediction items – thus limiting the amount of visual clutter. Blur levels range from b_{min} (most likely) to 1 (least likely) and are linearly translated from AccessRank scores. Finally, these levels are rounded to one of l discrete blur levels, so that users are not distracted trying to distinguish subtle differences between highlight prominences. In our implementation, we used $k = 5$ and $l = 4$.

$$b_{min_1} = \frac{2}{3}(1 - p_{max}) \quad (1)$$

$$b_{min_2} = \max(0.125(5 - n), 0) \quad (2)$$

$$b_{min} = b_{min_1} + b_{min_2} - b_{min_1} b_{min_2} \quad (3)$$

Hover Menus: provide shortcuts across levels

Hover Menus are designed to give users shortcuts to predicted targets located deeper in the hierarchy. They offer a recognition-based shortcut for reducing the number of levels that must be traversed through standard navigation actions.

When a user hovers over a folder for 500ms a menu appears below it (Figure 1b) showing lower level content (folders and files at any sublevel) that have high AccessRank scores. Selecting a menu item navigates directly to the associated folder or file. To facilitate rapid browsing of several folders, menus appear immediately if the cursor enters a folder within 500ms of leaving a folder with a menu already displayed. Menus also fade out if the cursor leaves a folder without a selection being made. The menus are populated with up to n files followed by n folders, each sorted by AccessRank score. By default, $n = 5$, although the value is user-configurable.

Search Directed Navigation

Search Directed Navigation (SDN) is intended to satisfy Goal 3 by providing search-based guidance through the navigation hierarchy towards the target, allowing the user's interaction with search results to be a rehearsal of navigation-based retrieval (Figure 2). SDN achieves this guidance by highlighting items in the hierarchy that match typed query terms, using a visually similar highlighting strategy to Icon Highlights. If the search term matches an item at any level, the highlighting propagates up the hierarchy to mark the path to the target. An item matches the search query if any word or words in the item's filename has the search query as a prefix.

The level of highlighting depends on the predicted probability of candidate targets, which may be calculated through AccessRank or through other metrics, such as visit count. Full content-based search strategies could also be used to determine the set of search matches, although our current implementation does not do so (to increase responsiveness and results specificity). The current version of the technique uses two highlight levels: one for previously-visited items (crisp border), and one for unvisited items (blurred border).

INTERFACE EVALUATION

Icon Highlights (*IH*), Hover Menus (*HM*), and Search Directed Highlights (*SDN*) are each intended to improve different aspects of navigation-based file retrieval, as emphasised by their associated design goals: improve visual identification of likely targets, provide shortcut target acquisition, and

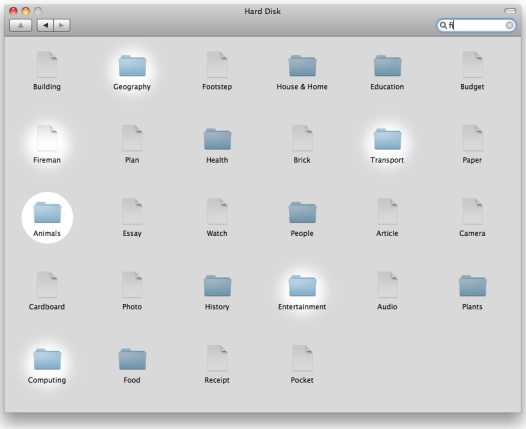


Figure 2: Experiment setup showing Search Directed Navigation. Predicted items (files and folders leading to files) are highlighted in response to search terms.

improve learning of the path-to-target. If deployed in real file browser applications, these techniques are probably best combined (discussed later). However, it is important to first understand how each of the methods is used in isolation – how they affect file navigation performance, and how participants subjectively respond to them.

Devising experimental tasks for evaluating file retrieval interfaces is complicated by the number of relevant factors that might be considered. These include the structure of the file hierarchy, the degree of ambiguity in the semantic relationship between hierarchical components, the stability or predictability of the data organisation in the interface, the recency and frequency of any previous visits to entities in the hierarchy, as well as individual user preferences. Our experimental design uses artificial tasks that control, or partially control, some of these factors in order to yield preliminary insights into the comparative usability and performance of the interfaces.

Experimental tasks involved navigating within a three-level semantically organised hierarchy to select a target file that was cued by displaying its name in a small window at the side of the file browser interface. Example targets include 'Paris' (within 'Geography' then 'Capital cities') and 'Horse' (within 'Animals' then 'Mammals'). The path to most targets was intentionally partially ambiguous – for example, the target 'Paris' might reasonably be contained within top-level folders 'Geography' or 'History', or in second-level folders 'Capital cities' or 'Large cities'. The ambiguity was intended to emulate imprecise recall of file locations and imperfect organisation as is typical of personal file hierarchies.

During the experiment participants visited files up to 5 times each, allowing analysis of how performance with the interfaces changed as users become more familiar with the file locations and retrieval mechanisms. The experiment also examined how well users remembered the location of target files.

Participants and apparatus

16 volunteers (9 female, mean age 25.2) participated in the study, which lasted 60-90 minutes. All had normal or corrected to normal vision and were fluent English speakers.

The experiment ran on an iMac with a display resolution of 1920×1080. The file browser window was 880×631 pixels, populated with 48×48 pixel icons representing files and folders. Software logged all user actions, including task time.

The experimental file browser application imitated the icon view in OS X at a fixed size of 6×5 icons (Figure 2). Items could be opened by double clicking, and toolbar navigation links allowed for navigation back, forward, or up to the parent directory. The basic interface, called *Standard* (ST) was then augmented with each of our three interfaces. IH used $k = 5$ and $l = 4$, HM used $n = 3$, and SDN used two highlight levels for visited (crisp) and unvisited (blurred) targets.

The same file hierarchy was used throughout the experiment, with targets located in different branches for each interface. The hierarchy structure was modelled on Bergman's findings of mean folder sizes [4]: 12, 10 and 8 folders, and 16, 12 and 9 files at the root, second and third level respectively. Folders containing no target nodes were not populated. For consistency, all target files used for analysis were located at the third level, which matches the observed mean file retrieval depth [4, 5]. In total, the hierarchy contained 388 folders and 448 files, of which 288 could be selected as target items.

Procedure

The experiment consisted of two parts: the first asked participants to carry out retrieval tasks (using each of *ST*, *IH*, *HM* and *SDN*) with icons that remained spatially stable in the file browser; the second repeated the procedure of part one, but used icons that were randomly rearranged after each selection in order to examine how performance with the interfaces was affected by unstable locations. Understanding susceptibility to unstable locations is important since many activities cause view instability, including changing a browser window's sorting criteria, view settings or size, or changing folder contents.

Part one of the experiment consisted of three phases using each of the four interfaces: *practice* (familiarisation with the interface using a training file hierarchy; data discarded), *retrieval*, and *standard-retrieval*. All three phases were completed using one interface before moving on to the next.

During the *retrieval* phase, participants completed 22 file retrieval tasks in response to cued target filename stimuli. Successfully completing a retrieval automatically initiated the next by displaying another target. The 22 retrievals comprised ten different target files (frequencies 5, 3, 2, 2, 2, 2, 1, 1, 1, and 1 times each, a near-Zipfian distribution) and two distractor targets. The two distractor targets were at the top level of the hierarchy, and were included to reduce participants' anticipation of targets always being located at the third level. They were inserted roughly one-quarter and two-thirds of the way through the overall sequence of tasks.

The ten true targets for each interface were randomly selected from the hierarchies contained within three top-level folders. To reduce learning effects stemming from familiarity with the hierarchy, different interfaces used targets from different top-level folders. Note that the two interfaces that used AccessRank (IH and HM) had no pre-populated history of file accesses, so for the initial selections of each target they provided no augmentations – no automatic highlighting (*IH*) and no menus (*HM*). As the retrieval phase progressed, however, these systems adapted to the user's navigation actions as described above. Because of the short duration of the experiment, a modified version of AccessRank was used that did not incorporate the time of day.

Once the *retrieval* phase was complete with each interface, participants completed NASA Task Load Index (TLX) worksheets [17] and provided comments on the interface.

The *standard-retrieval* phase consisted of a single selection of each of the ten targets from the *retrieval* phase using the standard file browser (i.e., the IH, HM, or SDN augmentations were unavailable). Its purpose was to analyse any differences in how the interfaces supported users in learning the traditional mechanisms for navigating to files.

Part two of the experiment repeated the method used for part one, but without the *standard-retrieval* phase and using maximally unstable icons – the spatial location of every icon in each folder was randomised after each target acquisition. The targets used with each interface were randomly selected within the top-level folders used for that interface in part one, so participants could be expected to have some familiarity with them from the beginning of part two. However, as with part one, the interfaces began with no usage history, and so provided no highlights or menus for the initial selections.

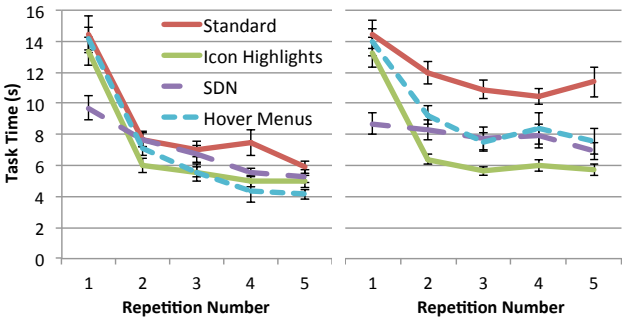
The following pseudo-code summarises the procedure:

foreach Part $\in$ {*stable, unstable*}
  foreach Interface $\in$ {*ST, IH, HM, SDN (counterbalanced)*}
    phase1: training
    phase2: retrievals
      if Part = stable
        phase3: standard-retrievals

Experimental Design

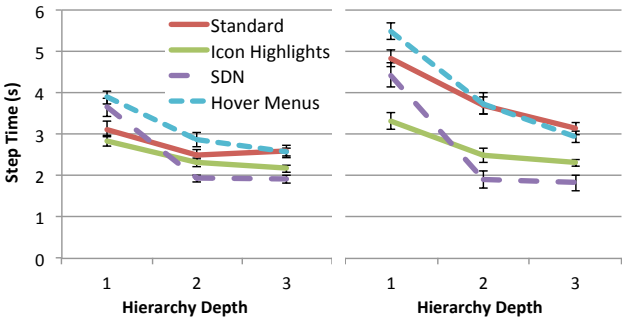
Data from the *retrieval* phase is analysed using a 4×5 repeated-measures analysis of variance (ANOVA) for within-subjects factors *interface* (levels *ST, IH, HM, and SDN*) and *repetition* (levels $1, 2, 3, 4, 5$, representing the count of repeated access to the same item). The primary dependent variable is total time to select the target file (log transformed to reduce the impact of positive skew). Navigational error rates are also analysed. Timing data from the *standard-retrieval* phase are analysed using one-way ANOVA across levels of *interface*, since each target is only visited once. To help characterise performance with the interfaces we also analysed step time as a secondary dependent measure (the time spent navigating down a level of the hierarchy) using factors *interface* and *depth* (hierarchical level $1, 2, 3$).

To reduce the impact of outliers when navigational errors were made, task times were capped at 30 seconds and step times at 10 seconds. This affected 3.4% of tasks and 1.7% of steps, distributed evenly between interfaces. Data from the first task for each interface in each phase was excluded. Post hoc tests use the Bonferroni correction. Where the ANOVA assumption of sphericity was violated (Mauchly test), Greenhouse-Geisser adjustments were used (as indicated by non-integral degrees of freedom).



(a) Spatially stable icons (b) Maximally unstable icons

Figure 3: Task times by repetition number. Error bars ± 1 st. err.



(a) Spatially stable icons (b) Maximally unstable icons

Figure 4: Step times by depth in the hierarchy. Error bars ± 1 st. err.

RESULTS

Results from part one (stable icons) are presented first, then part two (unstable icons), followed by subjective responses and further characterisation of how the interfaces were used.

Part 1: Spatially stable icons

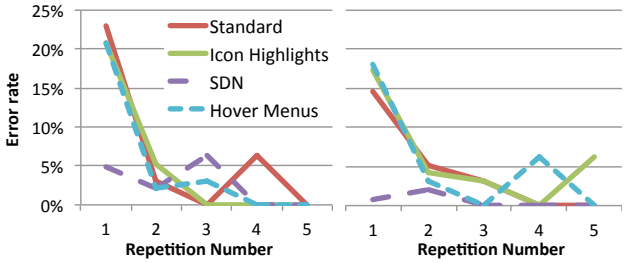
All three interfaces improved performance over the standard file browser, with SDN providing the largest benefits for unvisited files, and IH and HM working best for revisitations.

Retrieval time

Figure 3a summarises retrieval times. ANOVA showed a significant main effect of *interface* ($F_{1.9,28.8} = 8.2, p < .01$), with SDN, IH, and HM all similarly fast (means 6.98 s, 6.98 and 7.06 respectively) compared with ST slower at 8.51 s. Posthoc analysis confirmed pairwise differences between ST and both SDN and IH. There was an expected significant main effect of *repetition* ($F_{2.6,39.0} = 116.9, p < .001$), with mean times reducing from 12.9 s to 5.07 s for repetition 1 to 5.

Importantly, there was a significant *interface*×*repetition* interaction ($F_{7.1,106.8} = 7.1, p < .001$). Figure 3a shows that this is best attributed to SDN being substantially faster than the other interfaces for the first retrieval of an item, but slower than IH and HM for revisitations. We note that this is primarily due to differences in the techniques' underlying algorithms and interaction mechanisms, rather than their presentation approaches: SDN allows user input, and so can work immediately, whereas IH and HM must build up information about user selections before they can make good predictions.

To summarise, the standard interface (ST) was consistently slowest. Search Directed Navigation (SDN) was fastest for the first retrieval, when users had not yet learned item locations (and when the other interfaces had no history to work on), but third fastest with repeated items. Hover Menus (HM) were relatively slow for initial selections, but fastest once their menus had been populated with shortcuts. Icon Highlights (IH) was first or second fastest at each repetition level.



(a) Spatially stable icons (b) Maximally unstable icons

Figure 5: Error rates by repetition number.

Step times

We logged the time taken at each hierarchical level (i.e., a 'step') from a user's final arrival at a particular level (through a previous selection or feature such as the 'back' button) until their final departure. Note that the sum of step times for a task may be less than the total task time, since superfluous navigations are not included in any step time.

Figure 4a shows step times for each interface across file hierarchy depth. ANOVA showed significant main effects of *interface* and *depth* as well as a significant *interface*×*depth* interaction (all $p < .001$). SDN and HM both show particularly slow performance at the first level, which can be attributed to their interaction mechanisms – the time costs of typing a query (SDN) or browsing menus (HM). However, they provide faster times at deeper levels because users can quickly acquire deeper level targets, either through specific item highlighting (SDN) or direct mechanisms for access (HM).

Error rates

We logged any navigation into an incorrect folder as an error. However, as stated previously, the hierarchy was intentionally partially ambiguous, leading us to anticipate high levels of error, particularly for the first selection of any target.

Figure 5a summarises the results, showing no significant effect of *interface* ($F_{3,45} = 1.8, p = .16$), but an anticipated effect of *repetition* ($F_{2.6,38.8} = 33.7, p < .001$). A significant *interface*×*repetition* interaction ($F_{4.4,66.0} = 3.15, p < .05$) is best attributed to the marked difference between SDN's low error rate in the first repetition (4.9%) compared to other techniques (20-23%). This is in part an artefact of the experimental method, which used exact filename stimuli (favouring SDN) but inexact, ambiguous hierarchical structures to guide navigation-based retrieval.

Standard retrieval phase

The standard retrieval phase involved navigating to targets without any of the augmentations of IH, HM or SDN, to test for differences in participants' learning of the actual file locations. We found no main effect for *interface* (p = .338), with mean times ranging from 6.8 s with ST and IH to 7.7 s for SDN. Error analysis also showed no significant effect.

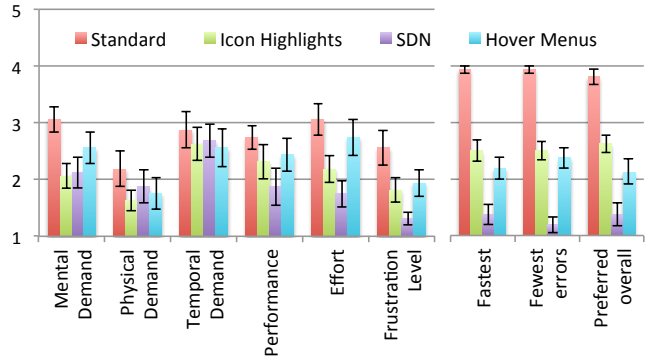


Figure 6: NASA TLX scores (left) and subjective rankings (right). Lower is better. Error bars ± 1 st. err.

Part 2: Maximally unstable icons

Results from part 2 showed that the interfaces provide additional value when icon locations are not spatially stable.

Retrieval times, step times, and error rates

Figure 3b summarises the results, showing that the new interfaces yield greater benefits with unstable icons than with stable icons. Mean retrieval times were fastest with IH (6.1 s), followed by SDN (8.0 s) and HM (8.6 s), with ST much slower at 11.5 s ($F_{1.9,28.7} = 25.4, p < .001$). Posthoc analyses show pairwise differences between ST and all three augmented interfaces, as well as between IH and HM. *Repetition* showed the anticipated significant main effect ($p < .001$).

The significant *interface*×*repetition* interaction ($F_{5.5,82.2} = 6.4, p < .001$) is again attributed to SDN showing much less performance improvement across repetition compared with other interfaces. The figure also shows that IH performed particularly well with unstable icons when revisiting items.

Analysis of step times showed similar (though larger) effects to those described above for stable icons (see Figure 4b), with both main effects and the interaction significant ($p < .001$).

In analysing errors, there were significant main effects for interface ($p < .05$) and repetition ($p < .001$), as shown in Figure 5b. SDN had substantially fewer errors (0.6%) than the other interfaces (4.6, 6.2, and 5.5% for ST, IH and HM respectively), with consistently low error rates, unlike the other interfaces, which had much higher error rates for the first repetition.

Subjective results

Subjective responses in all categories of the NASA TLX worksheets favoured the new interfaces in comparison to the standard file browser. Friedman tests show significant differences between interfaces for mental demand, performance, effort and frustration. Participants also ranked the interfaces (1 to 4) for speed, errors, and overall preference. SDN was consistently ranked first, and the standard interface consistently last. Responses are summarised in Figure 6.

When asked whether they would prefer SDN or traditional file search tools, 12 of the 16 chose SDN. However, several participants commented that directly listing results was an advantage of search, with one noting that SDN would be a good complement to search. One noted that they would like a combination of HM and SDN. Another noted that SDN was most useful when first accessing items, but after a few times they remembered their locations, as intended in design goal 3.

Two participants commented that the hover menus appeared too slowly, suggesting the need for a shorter dwell timeout or the ability to immediately post the menu (e.g., right click).

One participant also commented that Icon Highlights made it harder to select unhighlighted items, because the highlighting 'dragged' their eyes to items. Another complained that it highlighted folders that were opened in error. This problem could be addressed by only updating the visitation data for a folder when a leaf node it contains is ultimately selected.

Characterisation of use

Search Directed Navigation and Hover Menus require active use – users must explicitly choose to type a query or use a menu. We therefore examined the logs of interaction to determine how participants used these interfaces.

Figure 7 shows the proportion of SDN and HM tasks in which their features were actively used, across repetition of access. The interesting result is in the contrast between SDN usage patterns when navigating stable and unstable icons. With stable icons, participants made extensive use (77%) of SDN in the first repetition, decreasing to 19% in the fifth repetition, suggesting that they gradually learned icon locations and decided not to use explicitly typed queries as they could quickly select items at their known locations. With unstable locations, however, participants continued to use queries across repetition (65-76% of tasks for all repetitions). Referring back to the Search, Decision and Pointing model that motivated design goal 3, this suggests that users will naturally transition to decision-based mechanisms for identifying and selecting items when stable spatial locations allow them to do so, but that they prefer to use explicit search criteria instead of completing the time-consuming activity of visually searching for targets when items appear in random (unlearnable) locations.

Hover Menus were used consistently regardless of the stability of icons. Only the first repetition, where the menus were unlikely to be populated with assistive data, showed low levels of use. When menus were shown for stable icons, target items were present 58.0% of the time, and their parent folders 80.0% of the time, including 62.3% of the time when the target file was not. When the target file was present, it was selected 99.0% of the time; when absent, but with its parent folder present, the parent was selected 93.0% of the time.

DISCUSSION AND FUTURE WORK

Results show that all three interfaces improve performance in navigation-based file retrieval over the standard icon view. Icon Highlights and Hover Menus are particularly efficient for revisiting items, while Search Directed Navigation performs best for newly (or rarely) visited items.

We also found that interfaces' relative performance increases as spatial stability decreases. Both IH and SDN partially overcome spatial instability by quickly focusing attention to relevant icons. Hover Menus still requires visual search to find

a target folder, but the shortcuts it allows eliminate the need for visual search to be repeated at every level of the hierarchy. The standard view was notably slow with unstable icons, important because item locations can change when folder content changes or when window settings or sizes are changed.

Combining the interfaces

As anticipated from the design goals, each of the interfaces performs best in different scenarios of use. For example, the Icon Highlights presentation of AccessRank predictions was the best interface for frequently-revisited files in spatially-unstable views, and SDN was best for infrequently-accessed files that are deeper in the hierarchy. Importantly, however, the interface designs are complementary and can be combined to gain increased benefit. In such a combination, Icon Highlights would be always present, while the features of Hover Menus and SDN would be available to the user on-demand.

Icon Highlights and SDN currently use the same visualisation to highlight items. This overlap is unlikely to cause confusion, since users who enter a search query to SDN have made an explicit choice to use that technique, and will likely expect that the highlighting is reflecting their search rather than the general AccessRank prediction.

Hover Menus could be modified to adapt to SDN queries. Ultimately this combination converges with traditional lists of search results, but could be implemented to either highlight menu items or repopulate the menu accordingly.

One area for further study in combining these interfaces is to determine whether users face additional cognitive load in deciding which technique to use. We suspect that in a combined deployment users would default to using the automatic feedback provided by Icon Highlights, and only resort to Hover Menus or SDN when retrieval difficulties arise. This seems similar to current retrieval (where users resort to search after fruitless navigation), but the combined interface would provide a richer set of tools for assisting navigation without a complete change in retrieval mechanism.

Table 1 summarises properties of all the retrieval methods discussed in this paper, including navigation, search, the three interfaces and a combined interface. Values are based on a mixture of evaluation results and theoretical underpinnings. While all approaches have strengths and weaknesses, in theory, combining IH, HM and SDN provides the most benefits.

	ST	Srch	IH	HM	SDN	Cmb
Reduces step times	✗	N/A	✓	✗	✓	✓
Good for deep targets	✗	✓	✗	✓	✓	✓
Location learning	✓	✗	✓	✗	✓	±
Copes with instability	✗	✓	✓	±	✓	✓
Reminding ability	✓	✗	✓	✓	✓	✓
Fast revisitations	✗	✗	✓	✓	✗	✓
Handles unknown locations	✗	✓	✗	✗	✓	✓

Table 1: Properties of retrieval methods discussed in this paper (Srch = Search, Cmb = Combined). ✓= yes, ✗= no, ± = mixed support.

Interface refinements and implementation

Implementing the current designs of Icon Highlights, Hover Menus and SDN requires relatively minor cosmetic changes to the input and output behaviour of current file browsers

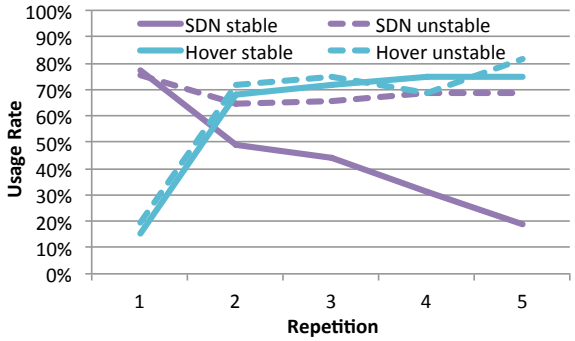


Figure 7: Percentage of tasks where SDN and Hover Menus were used, by repetition number.

– highlighting probable items, adding marks in the scroll-trough to demark their location, adding hover menus, and adding a search query field. Implementing the back-end predictive algorithms requires recording and tracking file access data, but is otherwise relatively straightforward.

The current implementations of IH, HM and SDN are relatively rudimentary, developed to explore their key concepts. There are therefore several avenues for interface refinement, including adapting them to other view types, and enhancing Hover Menus to provide more information about file and folder locations (e.g., by displaying full paths in tooltips).

Limitations

We summarise the limitations of our study below.

Highlights and AccessRank: Our highlighting method is based on the 'spotlights' seen in OS X, but other methods may further improve performance (e.g., ephemeral adaptation [11]). Our results do not indicate whether AccessRank, the highlighting method or some combination of them contributed to the success of IH and SDN, nor do they show whether the AccessRank parameters we used were optimal.

Non-natural setting: While we tried took steps in our experiment design to emulate a familiar dataset, a large field study involving participants' own personal file collections would yield more accurate findings. However, our preliminary evaluation allowed us to answer specific research questions that can be further explored in future work.

Task cuing strategy: The evaluation method initiated tasks by showing a stimuli consisting of the target file's name, which allowed users to simply type the name for a certain match with SDN. In real life, recall of filenames is imperfect (Blanc-Brude and Scapin's studies found that correct filename portions – necessary for use of SDN – were recalled for 83% of files, though only 25% of filenames were perfectly recalled [6]), and consequently our results for initial selections with SDN can be considered a best case analysis.

SDN vs search: Our evaluation compared performance between alternative interfaces supporting navigation-based file retrieval. We did not address performance comparison between search and navigation. A future evaluation could directly compare how search and SDN influence file location learning, whether they have different perceived workloads, and which techniques are best in different circumstances.

Sample size: While our results were highly significant, a larger sample size would produce more robust findings.

Erroneous predictions: While we did not directly investigate the effect of erroneous predictions, their effect was incorporated into the main results, showing that the interfaces still provided significant overall benefits. Further studies are needed to investigate their effect in more detail.

Variation in users: Further work is required to investigate the difference in effectiveness of the interfaces for users who organise their hierarchies in different way (e.g., pilers vs filers).

CONCLUSION

We presented three interface designs to improve user performance in accessing files by navigating through hierarchical structures. The designs use underlying algorithms or search term queries to predict likely target files, and they use these predictions to assist users in traversing the hierarchy. The interfaces are designed to *assist* with hierarchical traversal, rather than simply deliver a list of likely targets, because they are intended to help users learn the location of files, and the associated mechanisms for retrieval, to assist with future navigation-based file accesses. The Icon Highlights interface is designed to assist users in visually identifying likely targets at each level of the hierarchy. Evaluation results show that it performs particularly well when users are revisiting files in folder views that are spatially unstable. The Hover Menus interface is designed to facilitate shortcuts to likely folders and files across levels of the hierarchy, and evaluation results showed that it is particularly effective for revisiting files in spatially stable views. Search Directed Navigation highlights items that match search query terms, emphasising those also predicted as more probable. It offers an alternative to search by guiding users through the hierarchy to matching items. Evaluation results showed that it performs particularly well when users do not have location knowledge. All of the interfaces allowed faster task completion than the standard file browser in all conditions, and subjective preference favoured them. Further work will combine the techniques and evaluate them in longitudinal use, but current results suggest that these relatively easy to implement features will improve the common activity of navigation-based file retrieval.

ACKNOWLEDGEMENTS

This research is supported by Royal Society of New Zealand Marsden Grant 10-UOC-020, NSERC and the GRAND NCE.

REFERENCES

1. Balakrishnan, R. "beating" fitts' law: virtual enhancements for pointing facilitation. *Int. Journal of Human-Computer Studies* (2004), 857–874.
2. Barreau, D., and Nardi, B. A. Finding and reminding: file organization from the desktop. *SIGCHI Bull. 27*, 3 (1995), 39–43.
3. Bergman, O., Beyth-Marom, R., Nachmias, R., Gradovitch, N., and Whittaker, S. Improved search engines and navigation preference in personal information management. *ACM Trans. Inf. Syst.* (2008), 1–24.
4. Bergman, O., Whittaker, S., Sanderson, M., Nachmias, R., and Ramamoorthy, A. The effect of folder structure on personal file navigation. *JASIST 61*, 12 (2011), 2300–2310.
5. Bergman, O., Whittaker, S., Sanderson, M., Nachmias, R., and Ramamoorthy, A. How do we find personal files?: the effect of os, presentation & depth on file navigation. In *Proc. CHI '12* (2012), 2977–2980.
6. Blanc-Brude, T., and Scapin, D. L. What do people recall about their documents?: implications for desktop search tools. In *Proc. IUI '07* (2007), 102–111.
7. Boardman, R., and Sasse, M. A. "stuff goes into the computer and doesn't come out": a cross-tool study of personal information management. In *Proc. CHI '04* (2004), 583–590.
8. Cockburn, A., and Gutwin, C. A predictive model of human performance with scrolling and hierarchical lists. *HCIJ* (2009), 273–314.
9. Cutrell, E., Dumais, S., and Teevan, J. Searching to eliminate personal information management. *CACM 49*, 1 (2006), 58–64.
10. de Bruijn, O., Spence, R., and Chong, M. Y. Rsvp browser: Web browsing on small screen devices. *Personal Ubiquitous Comput. 6*, 4 (2002), 245–252.
11. Findlater, L., Moffatt, K., McGrenere, J., and Dawson, J. Ephemeral adaptation: the use of gradual onset to improve menu selection performance. In *Proc. CHI '09*, ACM (2009), 1655–1664.
12. Fitchett, S., and Cockburn, A. Accessrank: predicting what users will do next. In *Proc. CHI '12*, ACM (2012), 2239–2242.
13. Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. 1954. *J Exp Psychol Gen 121*, 3 (1992), 262–269.
14. Forlines, C., and Balakrishnan, R. Improving visual search with image segmentation. In *Proc. CHI '09*, ACM (2009), 1093–1102.
15. Géry, M., and Haddad, H. Evaluation of web usage mining approaches for user's next request prediction. In *Proc. WIDM '03*, ACM (2003), 74–81.
16. Gonçalves, D., and Jorge, J. An empirical study of personal document spaces. *DSV-IS* (2003), 403–412.
17. Hart, S., and Staveland, L. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human Mental Workload*, P. Hancock, Ed. (1988), 139–183.
18. Henderson, S., and Srinivasan, A. An empirical analysis of personal digital document structures. *Human Interface and the Management of Information. Designing Information Environments* (2009), 394–403.
19. Hill, W., and Hollan, J. Edit Wear and Read Wear. In *CHI '92*, Addison-Wesley (1992), 3–9.
20. Jones, W., Phuwanartnurak, A. J., Gill, R., and Bruce, H. Don't take my folders away!: organizing personal information to get ghings done. In *Ext. abstracts CHI '05*, ACM (2005), 1505–1508.
21. Kurtenbach, G. P. *The design and evaluation of marking menus*. PhD thesis, University of Toronto, Ontario, Canada, 1993.
22. Landauer, T. K., and Nachbar, D. W. Selection from alphabetic and numeric menu trees using a touch screen: breadth, depth, and width. In *Proc. CHI '85*, ACM (1985), 73–78.
23. Lansdale, M. The psychology of personal information management. *Applied Ergonomics 19*, 1 (1988), 55–66.
24. Malone, T. W. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst. 1*, 1 (1983), 99–112.
25. Miller, D. P. The depth/breadth tradeoff in hierarchical computer menus. *Proc. HFES '81 25*, 1 (1981), 296–300.
26. Nardi, B., Anderson, K., and Erickson, T. Filing and finding computer files. *Proc. EWHCI* (1995).
27. Sears, A., and Shneiderman, B. Split menus: effectively using selection frequency to organize menus. *ACM TOCHI 1*, 1 (1994), 27–51.
28. Stasko, J. An evaluation of space-filling information visualizations for depicting hierarchical structures. *Int. J. Hum.-Comput. Stud. 53*, 5 (Nov. 2000), 663–694.
29. Tulving, E., and Thomson, D. Encoding specificity and retrieval processes in episodic memory. *Psychological review 80*, 5 (1973), 352–373.
30. Zipf, G. *Human behavior and the principle of least effort: an introduction to human ecology*. Addison-Wesley Press, Mass., 1949.