

Finder Highlights: Field Evaluation and Design of an Augmented File Browser

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 to files through a hierarchy is often a slow, laborious, and repetitive task. Recent lab studies showed that file browser interface augmentations, such as Icon Highlights and Search Directed Navigation [15], have the potential to reduce file retrieval times. However, for this potential to be realised in actual systems, further study is necessary to address two important issues. First, there are important design and implementation challenges in advancing the research prototypes previously evaluated into complete interactive systems that can be used for real work. Second, it is unknown how real users would employ these systems while engaged in actual work; would the potential performance improvements suggested by the earlier lab studies be realised? We therefore describe the design, implementation, and longitudinal field study evaluation of Finder Highlights, a file browser plugin for the OS X ‘Finder’ that adds support for Icon Highlights and Search Directed Navigation. Study results confirm that the augmentations are effective in reducing real-world file retrieval times, with retrieval times 13% faster when using Finder Highlights compared to the standard tool (10.6 s versus 12.2 s), while also emphasising important differences between lab and field studies. In summary, the paper strongly suggests that large-scale deployment of interface augmentations to file browsers, particularly Icon Highlights, will have a marked effect in improving users’ real-world file retrieval.

Author Keywords

File navigation; prediction; file retrieval; revisitation

ACM Classification Keywords

H.5.2 User Interfaces: Theory and methods

INTRODUCTION

Every day, millions of computer users retrieve files from their file systems. Although there are many interface techniques for retrieving files, such as search tools and recent file lists, studies have shown that the predominant method of access is to navigate through a hierarchy using a file browser [3].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI 2014, April 26–May 1 2014, Toronto, ON, Canada
Copyright 2014 ACM 978-1-4503-2473-1/14/04...\$15.00.

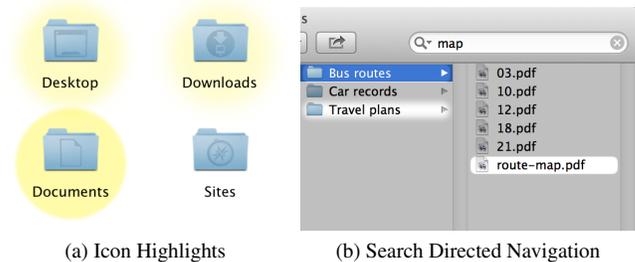


Figure 1: Icon Highlights and Search Directed Navigation, as implemented in Finder Highlights. Highlighted items are those most likely to be accessed next (IH, as yellow circles) or those that match the search query (SDN, as white cutouts).

Importantly, this frequent activity of navigating to a file is known to be slow, typically taking 12-17 seconds per retrieval [6]. There is therefore a pressing need and opportunity for interfaces that improve navigation-based file retrieval.

Previous work introduced two such interface techniques, called Icon Highlights (IH) and Search Directed Navigation (SDN) [15]. IH visually emphasises files and folders in the browser that are predicted to be more likely to be chosen, based on their access history. SDN also highlights items, but does so based on their match to a filename query explicitly typed by the user. Both techniques are designed to augment the normal behaviour of a file browser interface, with the highlighting intended to reduce visual search time for salient targets at each level of the hierarchy. For example, with SDN, if the user types “Spain”, the folder “Countries” would be highlighted, as would the target “Spain” once “Countries” was opened. As summarised later in the paper, an important part of the design objective for IH and SDN is to maintain and augment the key benefits of navigation-based retrieval, which allows users to take a series of reliable steps towards their target by moving through the hierarchy.

A lab study of Icon Highlights and Search Directed Navigation demonstrated their strong potential [15]. Participants performed navigation-based file retrievals much more quickly when using IH and SDN than when using a standard, non-augmented file browser. Subjective preferences also strongly favoured IH and SDN.

These findings are important in suggesting the potential for improving a core interaction that is used many times a day by most computer users. However, the results only demonstrate a *potential* – they improved performance when used in a lab setting that applied several experimental constraints.

Factors that limit the external validity of the findings, or that raise risks to generalisability, include both technical aspects of system behaviour and methodological constraints.

On the technical side, the evaluated browser augmentations were intentionally rudimentary, and they limited interaction to only the key interface features under study. They therefore excluded many of the interface components that would normally be present in a file browser, such as the following:

1. *Tools* – the evaluated systems eliminated navigation tools normally available in real file browsers, such as shortcuts, search features, buttons to go ‘back’ or ‘up’, etc.
2. *Views* – the evaluated systems permitted only a single-window icon view, whereas real file browsers allow extensive view configuration, including file trees, list details, thumbnails, and so on.
3. *Integration of IH and SDN* – a real-world file browser might be expected to support both IH and SDN, but the evaluated systems separated their support.
4. *Naïve back-end algorithms* – the evaluated systems made unrealistic simplifying assumptions to enable prediction of upcoming file selections.

On the methodological side, as with any lab study, aspects of the empirical methodology constrained external validity:

1. *Artificial and Static File Hierarchy* – the study used an artificial file hierarchy of static content, which differed from the user’s real file system in terms of size, structure, content, and importantly, familiarity.
2. *Revisitation Patterns* – participants were required to repeatedly revisit files within a short experiment, creating unrealistically dense revisitations.
3. *Task Stimuli* – rather than retrieving files when needed for actual work, participants retrieved them in response to a filename cue, when other cues may be more realistic [7].

Our intention is not to criticise the previous study. We believe it used an appropriate method for determining the potential of IH and SDN. However, these technological challenges and methodological issues need to be addressed before the potential of these systems can be realised in actual interactive systems. We therefore describe the design, implementation and longitudinal evaluation of a real-world file browser supporting Icon Highlights and Search Directed Navigation. Called ‘Finder Highlights’, the system is a plugin to Finder, the standard OS X file browser. Finder Highlights was evaluated in a four-week longitudinal study of real use. Results confirm that the techniques can improve file retrieval times and are subjectively preferred over standard techniques. Analysis of file retrieval performance shows that retrieval times when using Finder Highlights (mean 10.6 s) were 13% faster than retrievals accomplished without it (mean 12.2s). However, many participants did not use SDN, associating it with rarely-used search interfaces. This distinction between favourable lab study results for SDN (where all participants employed its features and benefited from them) and the field study results (where many participants ignored it) validates our motivation for the study. The finding also corroborates several prior stud-

ies showing that lab studies often produce more positive results than field studies [9, 11, 18, 26].

The primary contribution of this paper is positive field study results for Finder Highlights, particularly those that demonstrate the effectiveness of IH in supporting real work. Secondary contributions include important methodological observations on the difference between prior lab results and results in the field, as well as design and algorithmic extensions to IH and SDN for use in a real-world file browser.

BACKGROUND

There is extensive prior literature on file retrieval. This includes studies of how users organise and retrieve both physical (e.g., [21]) and computer files (e.g., [25, 2]), the performance impact of hierarchical structure (e.g., [10, 5]), and analysis of the use of enhanced interfaces, such as search, for file retrieval (e.g., [2, 3, 7, 8]). Although there has been extensive research on personal information management and file retrieval, including substantial improvements in the area of search systems, recent studies show that file navigation – the process of traversing through a file hierarchy in order to obtain a file – remains the dominant method for retrieving files [2, 3, 8]. Bergman [3] offers four explanations for this preference: 1) it is consistent and reliable; 2) it is cognitively lightweight, in that users can rely on recognition to identify folders and files, rather than having to recall a file name; 3) the retrieval processes of pointing and clicking may become partially automated with experience; and 4) the use of a hierarchical structure may reflect real-world relationships, assisting sense-making. Other retrieval methods, such as search, while valuable when needed, are often used as methods of ‘last resort’ when navigation-based retrieval fails [3, 25], in part due to their higher cognitive load [4].

The following subsections briefly review three key areas of related work: the use of predictive highlighting in user interfaces; the interface augmentations for file navigation that motivated this work; and an algorithm for predicting upcoming user actions in user interfaces.

Predictive Highlighting

Considerable previous research has explored predictive highlighting, for example to highlight commands [16, 23], menu items [13], document locations [1] and files [24, 15]. Its use has also been suggested to support email revisitation [12].

The success of highlighting interfaces has been mixed. While they have often been beneficial (e.g., [1, 15, 24]), other studies have shown them to provide little benefit (e.g., [13, 16]). Possible reasons for these differing levels of success include conflicts between highlight appearance and other use of colour [13], how effectively users can predict highlights [29], how much additional information highlights provide [23] (for example, benefits may be lower for spatially consistent interfaces), and the level of prediction accuracy [13].

Icon Highlights, HoverMenus, Search Directed Navigation
Fitchett et al. [15] described three interface augmentations for file browsers, called Icon Highlights, Hover Menus, and Search Directed Navigation. These augmentations addressed

three particular performance constraints of the file retrieval task: *visual search constraint*, the need to visually search for an item among those displayed in each folder view; *levels constraint*, the need to take an explicit user action at every level of the hierarchy; and *practice constraint*, the limitation on performance improvement across successive retrievals.

Icon Highlights (Figure 1a) is designed to ease the visual search constraint by highlighting items that are predicted to be more likely for upcoming access. Highlight prominence is based on the score from a predictive algorithm (called AccessRank, described below), normalised based on the maximum score of items in the same folder and by the size of the folder's access history. Items that are predicted to be more likely are consequently highlighted more prominently than others. The number of highlighted items in a folder is limited to promote visual saliency.

Hover Menus (not shown) are designed to ease the levels constraint by allowing users to bypass intervening hierarchical levels and jump directly to predicted items. A pop-up menu on every folder icon provides this shortcut access.

Search Directed Navigation (Figure 1b) is designed to ease the practice constraint by bringing some of the benefits of file navigation to the results of search queries. When the user types a query with SDN, the system highlights items that match the query or contain descendants that match the query. The design intention is to help users learn the navigation-based methods for future retrievals of the same file. SDN used two highlight levels: a prominent highlight for previously visited matched items and their ancestors, and a dim highlight for previously unvisited matches.

Each of these augmentations was separately evaluated in a single lab study, with a non-augmented file browser used as a baseline comparator. Experimental tasks involved navigating through an artificially-generated file hierarchy to repeatedly select one of ten different target files. To reduce extraneous experimental factors, the file hierarchy was constructed so that all folders at a given level contained the same number of items, and all folder content was displayed without scrolling.

Experimental results showed that all three interfaces improved file retrieval times in comparison to the standard file browser. Subjective preferences and workload measures also favoured the augmented interfaces. Icon Highlights was particularly efficient when retrieving previously visited files in an unstable view (as occurs, for example, when icon views are reflowed to match a changing window configuration). SDN was the most preferred method overall, and was ranked as being the least frustrating and requiring the least effort. Hover Menus, while preferred to the standard browser, was rated as the most mentally demanding of the three augmentations.

AccessRank

The AccessRank algorithm [14] predicts upcoming user actions based on past actions. Specifically designed for deployment within user interfaces, AccessRank emphasises both accuracy and stability of results over time. The need for stability stems from the expertise benefits that users gain when interface elements remain in predictable spatial locations [10].

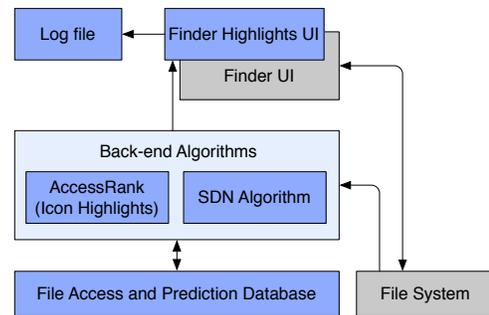


Figure 2: High-level architecture of Finder Highlights. Grey cells represent system components, blue cells represent parts of Finder Highlights, and arrows indicate information flow.

AccessRank calculates a score for each item by multiplying together several weighted values, with the weighting determined by a parameter. These values include a *Markov weight*, based on Markov chains [22] (a statistical model of events over time), as well as a *Combined Recency and Frequency (CRF) weight* [19], which efficiently considers every past access of an item with a bias towards more recent accesses.

AccessRank results are normally ranked by score, however, to facilitate prediction stability, two items can only swap places in subsequent rankings if the score of the lower-ranked item exceeds that of the higher ranked item by a required amount.

BASELINE FINDER USAGE DATA

Before developing Finder Highlights, we conducted a four-week field study that logged details of Finder usage. This study was originally used to characterise file retrieval activities. In this paper we further use these logs to calibrate AccessRank parameters for Icon Highlights and to provide a baseline description of Finder usage, used for between subjects comparison of Finder usage with that of Finder Highlights. The original logs were gathered by a tool installed on the computers of 25 participants (11 female, 14 from computer science departments) running Mac OS X 10.6 or 10.7, aged between 20 and 64 (mean 31.9, s.d. 11.0). Logs recorded details of all file retrievals, including navigation and search.

FINDER HIGHLIGHTS

In order to investigate the real-world feasibility of IH and SDN (the most promising augmentations in [15]), we developed Finder Highlights, a fully functional plugin for the Finder, the standard OS X file browser. A high-level architecture of Finder Highlights is shown in Figure 2. It consists of a user interface layer (Finder Highlights UI) that extends the OS X Finder. Finder Highlights UI requires file access predictions, which are provided by two different algorithmic components. The first of these serves Icon Highlights predictions, and uses modifications and extensions to AccessRank that were required to enable AccessRank to support mutable, hierarchical data (e.g., to support score changes when a file is moved between folders). The second component is new, and serves predictive augmentations within SDN. Both the Finder and the predictive algorithms draw on data from the file system. The algorithms store data about retrievals and predictions in a database, and the Finder Highlights UI writes data describing all user interactions to a log file.

We describe the design and implementation of Finder Highlights in three sections. First, we describe the interface design challenges and solutions in integrating IH, SDN, and existing functionality in a real browser. Second, we detail the algorithmic extensions to AccessRank necessary to allow strong file retrieval predictions. Third, we describe the algorithm developed to support SDN. To reiterate, these interface and algorithm extensions are prerequisite research contributions that enable Finder Highlights to support real file retrievals. Although described in the context of the OS X Finder, the algorithms are designed to be platform independent.

Interface Design

Design challenges of integrating IH and SDN

IH and SDN were originally implemented and evaluated separately, but real-world use might require both to be integrated within a single browsing interface. Achieving this integration raises substantial design challenges, as both IH and SDN use highlighting to guide users to items within each folder view, although the basis for their highlighting is very different (AccessRank predictions for IH, and query matches for SDN).

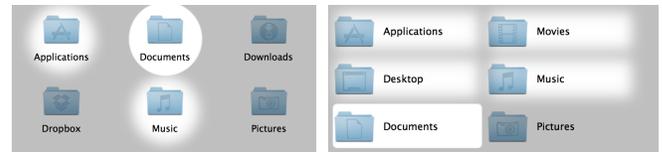
Concurrently displaying IH and SDN highlights would, in theory, provide users with access to the most complete set of information. However, we believed that this combined visualization would be hard to learn, visually cluttered, and potentially distracting. We therefore chose to show only one form of highlighting at a time.

IH highlights are shown without any explicit user request or action. They are displayed as yellow ‘spotlights’ over items, with the spotlight adapting to the size and shape of items in the folder view (e.g. Figure 1a, Figure 4). The blur level and brightness of each highlight varies based on the confidence of the associated prediction. Strongly predicted items, such as the *Documents* folder in Figure 1a, have bright highlights with a crisp border, while weakly predicted items are dimmer and more blurred.

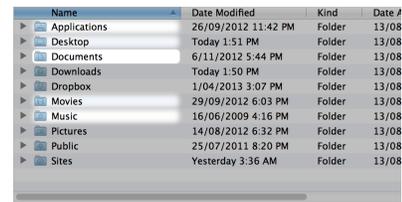
SDN highlights are shown in place of Icon Highlights whenever the user explicitly types a query into the SDN search field (as shown in Figure 1b). Like IH highlights, SDN highlights match the shape and size of highlighted items. However, to help users discriminate between IH and SDN highlights, they use clearly discriminable display methods. When SDN is active (by typing into the SDN search field), a semitransparent grey overlay is displayed over the view, with white ‘cutouts’ to indicate the highlighted items (Figure 1b). SDN uses two levels of visual prominence: previously visited search term matches have crisp borders; unvisited ones, blurred.

Design challenges in supporting multiple views

Finder Highlights supports all of the Finder’s view types, some of which are shown in Figure 3. We iteratively refined highlighting methods across view types, trying to balance visual saliency with aesthetics and clutter, resulting in some subtle variations between views. For example, Finder Highlights uses circular spotlights for icon view when the labels are beneath the items (Figure 3a), but a rectangular spotlight when the labels are to the right of the item (Figure 3b).



(a) Icon view (labels beneath icons) (b) Icon view (labels on right of icons)



(c) List view (Details view on Windows)

Figure 3: SDN highlights shown in different views.

Name	Date Modified
Applications	20/04/2013 3:22 AM
Desktop	Today 6:55 PM
Documents	1/06/2013 8:32 PM
Downloads	Yesterday 9:39 PM
Dropbox	Yesterday 9:01 PM
Movies	24/05/2013 10:35 PM
Music	16/06/2009 4:16 PM
Pictures	14/08/2012 6:32 PM
Public	25/07/2011 8:20 PM
Sites	26/05/2013 11:40 PM

Figure 4: Icon Highlights in list view.

For column view (Figure 1b), list view (Figure 3c), and Cover Flow views, a shorter, wider rounded rectangle highlight appears around the item’s icon and filename. The maximum blur level of a highlight is linearly scaled for highlights less than 48 pixels high, as pilot testing revealed that short highlight rectangles are not sufficiently salient when using the same blur levels as larger regions.

Design challenges of competing with other visual features

File browsers such as the Finder offer a range of features that potentially conflict with highlighting, such as colour labels (e.g., item ‘Dropbox’ in Figure 4) and the use of colour in icons. SDN’s highlighting method mitigates these concerns by uniformly applying a grey ‘transparency’ over all items except those highlighted. The use of yellow highlights in IH, in contrast, can conflict with other colour cues. IH applies two methods to mitigate the problem. First, in icon view, where there is a large amount of whitespace, IH spotlights are drawn *behind* icons, allowing their assigned colour to be unaffected by the IH spotlight. In other views, where there is inadequate whitespace to provide a sufficiently salient spotlight behind the item, the design solution is a compromise: highlights are drawn *on top* of the underlying view, combining colours as in a subtractive colour model (i.e., with colour values approaching black). This ensures that the highlight is visible without affecting text readability or the visibility of other content.

To reduce visual clutter and promote visual saliency of highlighted items, a maximum of seven IH highlights are shown in each folder. When the content of multiple folders are visible at once (e.g., multiple columns in column view), the highlight limit is applied to each folder individually.

Design challenges of integrating SDN and search

SDN and Finder's search feature are designed for differing use cases; for example, SDN supports folder- and file-name queries only, whereas search supports more advanced queries. However, both techniques are activated by typing a query into a search field. Providing multiple search fields in the file browser would clutter the browser interface and would be likely to confuse users, so Finder Highlights integrates SDN within Finder's existing search field (Figure 1b).

To achieve this, Finder Highlights adds a menu, activated by clicking on the search icon on the left of the field, allowing users to switch between search and SDN. The current mode is indicated by placeholder text in the field. New windows default to SDN mode unless created in response to a search related event (for example, via the system's Spotlight search feature). The Find command and its menu shortcut focus the active window's search field in its current search mode, while holding down the option key while using the command instead activates the other search mode. A progress indicator in the search field indicates an incomplete search. When opening a folder while SDN is active, Finder Highlights automatically scrolls to show the first highlighted item.

Extending AccessRank for Icon Highlights

IH uses a modified and extended version of AccessRank [14] to determine item highlights. The original AccessRank algorithm assumed a homogeneous dataset of independent items where every access carries the same weight. This section describes enhancements to AccessRank that better support predictions of hierarchical data, as well as describing the calibration of AccessRank parameters for use with file retrieval.

Configuring AccessRank parameters for Icon Highlights

AccessRank has two parameters which affect the accuracy and stability of its results: λ is the weighting between Markov and CRF algorithms (see earlier), and δ is the required threshold for items to swap ranks. These parameters are intended to be manipulated based on the characteristics of each application, requiring calibration.

Simulations on the baseline Finder usage data (described above), conducted using a similar method to the original AccessRank evaluation [14], confirmed the suitability of AccessRank for predicting file retrievals. We selected $(\lambda, \delta) = (1.65, 1)$ for subsequent use in IH, a configuration that provided the best-performing combination of accuracy and stability. Here, δ refers to a relative threshold for two items to swap places (i.e., for $\delta = 1$ an item's score must exceed that of a higher ranked item by at least 100% in order to swap places), which performed slightly better than an absolute threshold as used in the original version of AccessRank.

We also significantly lowered the weighting of retrievals made outside the Finder, so that items retrieved in this way were not given as prominent highlights as those retrieved within the Finder. Simulations showed that this modification substantially improved prediction accuracy, as files consistently retrieved without the Finder (e.g., using 'Open Recent' menus) are rarely also retrieved within the Finder.

Location-specific adaptations

The original version of AccessRank treated the entire dataset as a single homogeneous dataset, meaning that predictions and data describing actions did not need to account for any consideration of the 'location' or context in which they were made. Icon Highlights, in contrast, is designed to provide feedback to the user regarding predictions within a particular location (the current folder). Five modifications were made to the way that AccessRank scores are used by Icon Highlights to account for its use in file navigation, as follows.

First, scores for items within each folder are normalised relative to the maximum score of other items in the same folder. This ensures that at least one previously retrieved item will always be highlighted prominently by IH.

Second, AccessRank's stability delta is only applied when displaying physical folder locations. It is thus not applied to dynamic locations (e.g., search results), which vary with context and where highlight stability has little logical value.

Third, the highlight score for an alias or shortcut is that of the item it references, and opening the alias records a retrieval for the referenced item. The aim is to assist subsequent retrievals of that file regardless of the path taken to it.

Fourth, to remove the effect of navigation errors corrupting prediction accuracy, both file accesses and ancestor folder accesses are recorded only when the file is opened. For the purposes of AccessRank's Markov model, the previous item is always the previously accessed file, rather than any ancestor folders, with transitions recorded from this previous file to all path components of the subsequent file retrieval.

Fifth, OS X's *Quick Look* feature allows an item to be quickly previewed without opening a separate application. Once a preview window is shown, users can switch between items using the keyboard. To attempt to distinguish between use of Quick Look as a substitute for opening a file, and for skimming through a list of files, Finder Highlights records a retrieval only when a preview is visible for at least five seconds.

Updating scores after changes to the hierarchy structure

Finder Highlights stores its data based on file paths. When an item is moved or renamed, it carries its score and history data to its new path, and references contained in the Markov model are updated. When an item is deleted, its data is removed.

We consider three types of items when moving or deleting items. *Files* are leaf nodes in the hierarchy, *containers* are the direct parents of files, and *intermediaries* are the parents of containers or other intermediaries. If a file is moved or deleted, its container may still be an area of active work. If an intermediary is moved or deleted, this may be more indicative of a more fundamental change in the hierarchy structure. A given folder may act as both a container and an intermediary in different contexts.

Accordingly, containers and intermediaries are treated differently when updating CRF weights. If the child of an intermediary is removed, its CRF weight is subtracted from the intermediary's weight recursively up the hierarchy. However, if the child of a container is removed, the container's CRF

score is *not* affected. A moved item's CRF weight is added to its new ancestors, regardless of whether it is a file or folder.

Recording simultaneous retrievals

The Finder allows users to select and open multiple files simultaneously as part of a single retrieval event, for example to view a series of photos. To prevent this resulting in excessive score increases for ancestor folders, Finder Highlights records only a single access for each file or ancestor folder observed in a simultaneous retrieval.

Algorithm for Search Directed Navigation

Icon Highlights guides users to predicted items within the current folder. Its backend algorithm does not need to inspect descendent files and folders until their parent folder is opened. Search Directed Navigation, in contrast, provides visual feedback that guides users to descendent files and folders that match a search term query at any lower level of the hierarchy. Search terms are case-insensitive, and the search is initiated and refined for every successively typed character.

The SDN algorithm is designed to provide rapid and accurate feedback to the user. It does so by using an *index of visited items* and *search prioritisation*, described below.

Index of visited items – immediate highlights for previous items

The index of visited items records all files, folders, and their ancestors that have been opened, regardless of the method by which they were opened. The index is stored on disk as a sorted list of file and folder paths. The SDN algorithm also stores in memory a sorted list of all potential lookup keys, which are the successive substrings of each file or folder name (for example, the filename “finderHigh-CHI.14” would generate substrings “finder”, “High”, “finderHigh”, “CHI”, etc). A match occurs when a query matches the start of any lookup key; in this way, all results for a query can be found using binary search in $O(\log(n) + m)$ time, where n is the number of key-path pairs, and m is the number of matches for the query. The index is updated in response to file management events, such as moving, renaming, or deleting items or their ancestors. When a search term is entered, the index is searched and results are returned in a tree form that mirrors the file hierarchy. Consequently, for previously visited items SDN can give essentially immediate feedback.

Search prioritisation

Following initial feedback generated through the index, the SDN algorithm performs an exhaustive breadth-first search of the subtrees displayed in the current folder. This is necessary to generate results for previously unvisited parts of the file hierarchy. Matches are dynamically added to the results.

Although the search is exhaustive, it is prioritised to maximise the speed with which highlights are identified. Once a folder is highlighted its priority is reduced (because finding more matches will not generate further changes to the current folder), so folders that have no current highlights are searched first. Further, user folders (such as *Documents* and *Downloads*) are prioritised above system folders.

Location and query changes are common while SDN is active, as users traverse their hierarchy or refine their query.

The SDN algorithm ensures that any additional searching is minimised when these changes occur, retaining or refining existing results where possible.

The SDN algorithm can optionally use OS X's Spotlight index to improve search performance. It issues a concurrent search term to Spotlight, restricted to descendent items of the current folder which have names starting with the SDN query.

FIELD EVALUATION OF FINDER HIGHLIGHTS

The primary objective of this research is to learn how techniques similar to IH and SDN are used in the real world. Finder Highlights addresses the interface and algorithm design issues described above, allowing for such an evaluation.

Finder Highlights was deployed in a four-week-long field study, logging details about Finder usage, IH predictions and SDN searches. To protect privacy while maintaining an ability to identify revisitations, logs recorded hash codes of file paths, rather than the original paths. Logs were stored locally on participants' computers until the end of the study.

Procedure

Twenty OS X 10.8 users (4 female) participated in the study, aged between 23 and 66 (mean 32.0, s.d. 9.8), with 19 submitting data at the end of the study. Self-reported daily computer use ranged from 1 to 14 hours per day (mean 7.2 hours). Participants were recruited from a range of sources: four were tertiary students (three in computer science), eight were university staff or faculty (seven in computer science), and eight worked outside academia.

Communication with participants occurred online. They were instructed to download the Finder Highlights software and to watch a three minute video that briefly explained the techniques. The video directed participants to use their computers as normal during the study, and to use the techniques as little, or as much, as they liked. After four weeks, participants were sent instructions to uninstall the software, submit their logs, and complete a post-study survey.

Log File Data Analysis

Ultimately, the techniques included in Finder Highlights are intended to reduce file retrieval times. However there are complexities in determining these times from interaction logs. In particular, certain assumptions must be made to establish an objective time at which the retrieval began (in contrast, lab experiments can begin timing at the point an artificial stimuli is shown). In our analyses, retrieval time measurements were determined to begin at the later of (1) the time a browser window was opened; (2) the time of the first navigation event after either another retrieval, use of another browser window, or a file management task (e.g. moving an item); and (3) a navigation event that was more than 30 seconds after the previous one. Time measurements ended when a file was opened. Retrievals of items on the desktop were excluded, since Finder Highlights had no effect on the desktop interface.

To further characterise retrievals, we also analysed *step times* – the time between opening successive folders while navigating to a file. Steps were excluded from the analysis if they included: an upwards step through the hierarchy; use of history

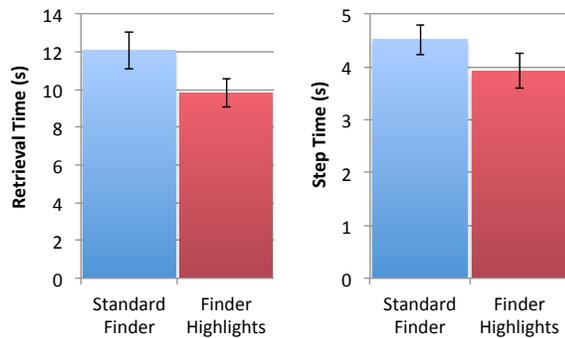


Figure 5: Non-search retrieval and step times with and without Finder Highlights, showing the effect of Icon Highlights. Error bars ± 1 st. err.

buttons, sidebar links, search, or another window; a change in the view configuration; file management activities (e.g., moving a file); or if the step took longer than 30 seconds.

Baseline between-subjects comparisons were performed between logs from the standard Finder (described previously) and from Finder Highlights. The two sets of logs were analysed with exactly the same assumptions using exactly the same analysis tools; in particular, the assumptions for measuring retrieval and step times were identical. All times were log-transformed for statistical analyses (both ANOVA and unpaired t-tests) to reduce the skew of the data.

Retrieval Times and Step Times

We conducted between-subjects analyses of retrieval and step times using Finder and Finder Highlights logs. The Finder logs contained data about 1689 retrievals, while the Finder Highlights logs contained 2521.

The mean retrieval time with Finder Highlights was 10.6 s (s.d. 3.4), 13% faster than the Finder mean of 12.2 s (s.d. 4.7): $T_{42} = 2.7, p < .01$. The mean for the Finder retrieval times is supported by a recent result from Bergman et al. [6], which reported a similar mean Mac OS X file retrieval time of 12.6 s.

More exact comparison of retrievals is possible by separating the Finder Highlights logs between those retrievals conducted using Icon Highlights, and those using Search Directed Navigation, as described below.

Icon Highlights

To examine the independent contribution of Icon Highlights, we excluded retrievals that involved search-based interfaces. Results were similar to the overall results above, which is best explained by the low proportion of retrievals using search or SDN (see below). Mean retrieval times with IH were 9.8 s (s.d. 3.3), compared to 12.1 s (s.d. 4.8) without it – $T_{42} = 3.1, p < .005$. To eliminate the possibility that this difference in retrieval times is explained by differences between the participants' file structures (rather than by the retrieval interfaces themselves), we compared the mean depth of retrieved files between the studies, with no significant difference. Further, a comparison of step times confirms that participants selected targets within the folder view more quickly when IH was available (mean 3.9 s, s.d. 1.4) than when using the standard Finder (mean 4.5 s, s.d. 1.5) – $T_{43} = 2.0, p < .05$.

Importantly, as IH is intended to reduce visual search time by highlighting likely targets, this reduction provides strong evidence that this design intention is fulfilled. Both step and retrieval times for IH are summarised in Figure 5.

Search Directed Navigation

In the earlier lab study, SDN was highly efficient and extremely well received by participants [15]. Results from the field study, however, are very different; not because SDN was inefficient or disliked, but rather because many participants never used it. It was actively used by only 4 of the 19 participants, who used it in between 1% and 44% of navigational steps (mean 14%); these participants averaged 9.8 retrievals and 89 steps using SDN. This contrast between extremely positive lab study results and substantially neutral findings due to disuse in the field amplifies the motivation for our research, in migrating analysis from the lab to the real world.

The lack of data on active use of SDN complicates its rigorous analysis. Further, as SDN introduces new functionality – guiding users to files through the hierarchy in response to a search term – there are difficulties in determining an appropriate baseline for comparison. Limited insights can be gained by comparing steps times following SDN activation for the four participants who used it (mean 2.9 s, s.d. 1.3) to other steps made by the same participants (3.8 s, s.d. 1.7), however statistical tests are inappropriate due to the above reasons.

Step Times with Correct and Incorrect Highlighting

Highlighting functionality is likely to assist users when their intended target is highlighted, but may distract when other items are highlighted and the target is not. To investigate this issue, we categorised all IH steps in one of three ways to produce an analysis factor *step type*: *positive highlights*, where the selected item was highlighted; *negative highlights*, where other items were highlighted and the selected item was not; and *no highlights*, where no items were highlighted. We also categorised steps in the baseline Finder logs in the same way, by first determining which items *would* have been highlighted, had IH been available.

An ANOVA with between-subjects factor *study* and within-subjects factor *step type* showed a significant effect of *step type* – $F_{2,82} = 9.4, p < 0.001$. With Finder Highlights, step times were fastest with *positive highlights* (3.4 s), followed by *no highlights* (3.9 s) and *negative highlights* (5.1 s). In the baseline data, step times were 4.2 s for *positive highlights* and *no highlights*, and 5.3 s for *negative highlights*. This result provides further evidence that IH, when providing correct predictions, was successful in helping users quickly identify and select targets. Additionally, negative highlights did not result in any noticeable change in performance.

The slow times for *negative highlights* can be partially explained by the variability in folder sizes: in the Finder Highlights study, folders with *negative highlights* contained many more items (mean 73.9, s.d. 111.3) than either *positive highlights* (mean 21.8, s.d. 16.5) or *no highlights* (mean 24.2, s.d. 22.5). This is likely because large folders have a smaller percentage of highlighted items, and are therefore more likely to produce *negative highlights*.

Statement (paraphrased)	Strongly Disagree (1)	Disagree (2)	Neutral (3)	Agree (4)	Strongly Agree (5)	Mean
Icon Highlights						
IH was useful	0	2	3	9	5	3.9
IH reduced cognitive load	0	3	7	5	4	3.5
Would like IH as permanent feature	0	4	3	8	4	3.6
Predictions seemed accurate	0	2	1	14	2	3.8
Could predict if target would be highlighted	0	1	4	8	5	3.9
Highlighting technique clear and intuitive	0	0	2	11	6	4.2
Search Directed Navigation						
SDN was useful	3	2	9	2	3	3.0
SDN affected file retrieval behaviour	2	2	10	2	3	3.1
Would like SDN as permanent feature	2	2	9	2	4	3.2
Results seemed accurate	0	0	4	2	4	4.0
SDN made it harder to use search	4	6	1	3	0	2.2
Confused between SDN and search	5	5	1	3	0	2.1
Highlighting technique clear and intuitive	0	0	2	4	6	4.3

Table 1: Participant agreement with statements about IH and SDN

Further analysis of steps showed that 46.5% had *positive highlights*, 29.3% had *negative highlights*, and 24.2% had *no highlights*. In contrast to the 46.5% of selected items that were highlighted, an average of 22.5% of items were highlighted in each folder, suggesting that the AccessRank prediction algorithm provided good predictions of upcoming file selections. The average rank of a revisited item in its folder was 3.4 (18.2% of all steps opened a top-ranked item), while an average of 3.0 items were highlighted at each step.

Questionnaire Responses

After four weeks of use, participants completed a post-study questionnaire that gathered both quantitative subjective responses (using five-point Likert scales from ‘strongly disagree’ (1) to ‘strongly agree’ (5)) and participant comments. The quantitative responses are summarised in Table 1, which shows abbreviated forms of the Likert-scale questions asked.

Likert scale responses reflect the log results above. Participants responded more favourably to the question ‘IH is useful’ (14 responses above neutral, 2 below) than for the equivalent question with SDN (5 above neutral, 5 below). Participants also felt the IH predictions were accurate (16 above neutral, 1 below) and predictable (13 above neutral, 1 below), and that its highlighting technique was clear and intuitive (17 above neutral, none below). The positive response for the predictability of highlighting is important, as prior research has shown that users can more quickly identify targets using pre-attentive search if they know in advance that they will be highlighted in a particular way [29].

Icon Highlights

The survey also asked participants to rate the effect of IH on retrieval times and error rates. Participants generally stated that it improved retrieval times (7 felt it made them faster, 1 slower), especially when retrieving previously visited files (15 faster, none slower). IH also had a positive impact on perceived error rates (7 reported fewer errors, none more).

The participants’ general comments on IH referred to it being most useful when navigating through many successive direc-

tories, when lots of items in a folder had similar names, and when revisiting specific items – particularly in large folders. However, a number of comments suggested possible refinements: (1) that too many items were highlighted towards the end of the study (one participant), which could be rectified by reducing the per-folder highlight limit; (2) that there was not enough difference between prominent and dim highlights (one participant), which could be rectified by calibrating the highlighting formula; (3) that the highlight colour was not visually appealing or should be customisable (six participants); (4) a lack of keyboard navigation, which would be useful to move between highlighted items (one participant); and (5) confusing interaction between highlights and existing colour labels (one participant), which suggests the need for refining the colour blending methods used in Finder Highlights.

The participants’ comments also suggested a perception that the techniques might be most effective for users with disorganised file structures. P8: “*There is more and more mess on my hard drive, so it is useful to highlight files that I actually use.*”. Indeed, IH may be of more benefit to ‘pilers’ than ‘filers’ (terms used to describe file organisation behaviour [21, 28]); those with disorganised hierarchies may be more reliant on visual search, perhaps due to larger folder sizes or less inherent knowledge of file locations.

While IH was designed to help retrieve files, one participant said that its benefits were primarily in reducing *organisation* time. This participant had previously made extensive use of OS X’s manual colour labelling feature, meaning that IH provided no additional benefit at retrieval-time. However, he commented that it was a very useful automatic replacement for manual labelling. Another unexpected use, referred to in comments from four participants, was that sometimes the *absence* of highlights was particularly useful. For example, in ordered lists of movie files, the highlighted items would indicate which had already been watched, and the first unhighlighted item following them would correspond to the video they wanted to watch next. In another example, a participant sometimes knew a target item would not be highlighted, and could ignore highlights to speed up visual search.

Search Directed Navigation

As expected, considering its relatively infrequent and light use, the dominant response for questions concerning use of SDN was neutral. However, as supported by the logs, a minority of the participants found it useful, with three expressing strong agreement regarding its usefulness.

Those who used it appreciated the context it provided, using it both as a search substitute and in place of some navigation tasks. P5: “*Just loved it! Unlike the normal search, my current path stays the same, and I don’t have a list of items coming from nowhere... I was comfortably guided through what I wanted. Sometimes [with normal search] several files match the query and the results are all [shown together]. Here I can easily focus on the one file I was looking for. No ambiguities.*”

P4 commented that SDN resulted in lower error rates than search, and facilitated learning of item locations, in line with its design goals: “*Sometimes I will use [Spotlight search], but*

if a file doesn't appear that I am looking for, I am no closer to finding it. With Search Directed Navigation... I was able to use a starting top level folder as a starting point. This allowed me to usually find my file much more quickly, and with more success, but I also got the benefit of being reminded of where I stored the file and other files that I passed on the way."

P6 also mentioned this property, commenting "I really liked the way it seemed to teach the directory structure to me; traditional find would hide the structure from me", and referring to IH as a way to retrieve files faster, but SDN as a way to remember his structure.

However, most participants rarely or never used SDN. Common reasons for this were that participants already knew where their files were (12 participants), do not use search much (8), forgot or did not think to use it (4), prefer traditional search interfaces (2), do not navigate to files much (2) or that it was too slow to find results (2). One participant commented that he would have used SDN frequently had it supported navigation to highlighted items using the keyboard. Two participants did not realise that SDN only searched file-names, and were confused by this, with another suggesting that a content searching option would be useful. A common theme in the explanation for the lack of use of SDN was a reluctance to resort to search, which confirms results of previous studies suggesting that search is often used as a method of last resort [2, 3, 7, 8].

Despite the low uptake, the strong positive reactions from those who used it suggest that SDN can be a valuable tool once people are familiar with it.

Discussion

To briefly summarise, conducting a real world evaluation of IH and SDN required addressing several user interface design challenges in extending and integrating the previously reported research prototypes into a viable file browser. It also required several extensions to existing algorithms. Results of a four-week field study in which real users employed Finder Highlights for their everyday file retrieval activities showed that IH reduced file retrieval times and was enthusiastically received by users. SDN was ignored by most users, but appreciated by those who used it.

Methodology – Lab versus Field

The results of the lab study and field study are very different. In the lab, SDN caused marked performance improvement and strongly positive subjective assessments, but in the field, it was rarely used. This finding underlines and reiterates methodological concerns previously identified in several HCI studies [9, 11, 26]. The need for cross validation across multiple methods is well known (e.g., [18]), but it remains relatively uncommon in HCI. There are several disincentives for advancing beyond lab studies – the *potential* of innovative interface concepts can be established quickly and with strong rigour in controlled experimental studies; advancing beyond lab studies requires arduous and refined system engineering; and longitudinal evaluation is difficult to conduct, not least because of the difficulty of finding participants who will make a long term commitment to experimental software.

Although these disincentives are substantial, the results in this paper exemplify both the limits of lab studies and the value of field studies. We hope to stimulate more evaluations of novel interactive systems that go beyond the lab.

Overcoming Barriers to Adoption

It seems likely that SDN can improve user performance in file retrieval tasks, as suggested by the lab study and the enthusiasm of those who used it. However, until users have their first experience of its benefits, they have little reason to try to exploit it, as doing so requires an overhead in learning and adapting to its new functionality. The barriers to adopting new and ultimately more efficient interface methods were examined by Scarr et al. [27]. There are abundant opportunities for further research on methods that help users identify and capitalise on unused interface methods that will ultimately assist them in becoming more efficient with computing tools, with 'Skillometers' providing one recent example [20].

Prediction Versus Interaction

IH and SDN use sophisticated algorithms to predict the user's upcoming actions. The algorithms' predictions are currently deployed to improve the user's *manual* methods for file retrieval – they help users identify targets and learn the navigational actions required to facilitate improvement over time. However, there is an interface design tension between the accuracy of the predictive algorithm and the interface methods used to present those predictions. For example, if Access-Rank or any other algorithm had near perfect prediction, then it would be near pointless to require the user to manually retrieve the file; instead, the system could autonomously open the file on the user's behalf. There are interesting research opportunities in analysing the theoretical and design challenges of different interface methods for working with algorithmic predictions, with some of the issues discussed in [17].

CONCLUSION

Finder Highlights is a robust and real software system that augments the OS X file browser, the Finder, with techniques designed to improve navigation-based file retrieval. It was designed and implemented in order to test Icon Highlights and Search Directed Navigation, two techniques that earlier lab studies showed to have strong potential for improving file retrieval. We presented the interface and algorithmic design challenges associated with advancing from the lab to the field. The algorithmic advances, in particular, represent research contributions that can be reused by future researchers working on interfaces that predict upcoming user actions.

A four week field study of Finder Highlights showed marked differences between the lab and field results. While the field study results show that the potential of Icon Highlights identified in the prior study is realised in the field, the lab study results for Search Directed Navigation did not fully translate to the field. The findings have clear implications for the design of future file browser interfaces, which should incorporate features similar to those of Finder Highlights, and for HCI experimental methodology. We hope to see more novel interactive techniques tested in the field as well as the lab.

ACKNOWLEDGEMENTS

This work was partially funded by Royal Society of New Zealand Marsden Grant 10-UOC-020. Thanks to the anonymous participants for their time and efforts.

REFERENCES

1. Alexander, J., Cockburn, A., Fitchett, S., Gutwin, C., and Greenberg, S. Revisiting read wear: analysis, design, and evaluation of a footprints scrollbar. In *Proc. CHI '09* (2009), 1665–1674.
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 TOIS* 26, 4 (2008), 1–24.
4. Bergman, O., Tene-Rubinstein, M., and Shalom, J. The use of attention resources in navigation versus search. *Personal and Ubiquitous Computing* 17, 3 (2013), 583–590.
5. 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.
6. 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.
7. 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.
8. 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.
9. Chiasson, S., Biddle, R., and van Oorschot, P. C. A second look at the usability of click-based graphical passwords. In *Proc. SOUPS '07* (2007), 1–12.
10. Cockburn, A., and Gutwin, C. A predictive model of human performance with scrolling and hierarchical lists. *HCIJ* (2009), 273–314.
11. Duh, H. B.-L., Tan, G. C. B., and Chen, V. H.-h. Usability evaluation for mobile device: a comparison of laboratory and field tests. In *Proc. MobileHCI '06* (2006), 181–186.
12. Elswailer, D., Harvey, M., and Hacker, M. Understanding re-finding behavior in naturalistic email interaction logs. In *Proc. SIGIR '11*, ACM (2011), 35–44.
13. Findlater, L., Moffatt, K., McGrenere, J., and Dawson, J. Ephemeral adaptation: the use of gradual onset to improve menu selection performance. In *CHI '09* (2009), 1655–1664.
14. Fitchett, S., and Cockburn, A. Accessrank: predicting what users will do next. In *Proc. CHI '12* (2012), 2239–2242.
15. Fitchett, S., Cockburn, A., and Gutwin, C. Improving navigation-based file retrieval. In *Proc. CHI '13* (2013), 2329–2338.
16. Gajos, K. Z., Czerwinski, M., Tan, D. S., and Weld, D. S. Exploring the design space for adaptive graphical user interfaces. In *Proc. AVI '06* (2006), 201–208.
17. Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., and Weld, D. S. Predictability and accuracy in adaptive user interfaces. In *Proc. CHI '08* (2008), 1271–1274.
18. Hertzum, M. User testing in industry: A case study of laboratory, workshop, and field tests. In *Proceedings of the 5th ERCIM Workshop* (1999), 59–72.
19. Lee, D., Choi, J., Kim, J.-H., Noh, S. H., Min, S. L., Cho, Y., and Kim, C. S. On the existence of a spectrum of policies that subsumes the least recently used (lru) and least frequently used (lfu) policies. *SIGMETRICS Perform. Eval. Rev.* 27, 1 (1999), 134–143.
20. Malacria, S., Scarr, J., Cockburn, A., Gutwin, C., and Grossman, T. Skillometers: reflective widgets that motivate and help users to improve performance. In *Proc. UIST '13* (2013), 321–330.
21. 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.
22. Markov, A. The theory of algorithms. *American Mathematical Society Translations* (1960).
23. Matejka, J., Li, W., Grossman, T., and Fitzmaurice, G. Communitycommands: command recommendations for software applications. In *Proc. UIST '09*, ACM (2009), 193–202.
24. Moon, J. M., and Fu, W.-T. Effects of spatial locations and luminance on finding and re-finding information in a desktop environment. In *Proc. CHI '09*, ACM (2009), 3365–3370.
25. Nardi, B., Anderson, K., and Erickson, T. Filing and finding computer files. *Proc. East-West HCI* (1995), 162–179.
26. Nielsen, C. M., Overgaard, M., Pedersen, M. B., Stage, J., and Stenild, S. It’s worth the hassle!: the added value of evaluating the usability of mobile systems in the field. In *NordiCHI '06* (2006), 272–280.
27. Scarr, J., Cockburn, A., Gutwin, C., and Quinn, P. Dips and ceilings: understanding and supporting transitions to expertise in user interfaces. In *Proc. CHI '11* (2011), 2741–2750.
28. Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proc. CHI '04*, ACM (2004), 415–422.
29. Treisman, A. Perceptual grouping and attention in visual search for features and for objects. *J Exp Psychol [Hum Percept]* 8, 2 (1982), 194.