

FaThumb: A Facet-based Interface for Mobile Search

Amy K. Karlson¹ George Robertson² Daniel C. Robbins² Mary Czerwinski² Greg Smith²
¹Human-Computer Interaction Lab
 Computer Science Department
 University of Maryland, College Park, MD, 20742
 akk@cs.umd.edu

²Microsoft Research
 One Microsoft Way
 Redmond, WA 98052, USA
 {ggr, dcr, marycz, gregsmi}@microsoft.com

ABSTRACT

In this paper we describe a novel approach for searching large data sets from a mobile phone. Existing interfaces for mobile search require keyword text entry and are not suited for browsing. Our alternative uses a hybrid model to de-emphasize tedious keyword entry in favor of iterative data filtering. We propose navigation and selection of hierarchical metadata (facet navigation), with incremental text entry to further narrow the results. We conducted a formative evaluation to understand the relative advantages of keyword entry versus facet navigation for both browse and search tasks on the phone. We found keyword entry to be more powerful when the name of the search target is known, while facet navigation is otherwise more effective and strongly preferred.

Author Keywords

Mobile devices, faceted metadata, search interfaces, visual interaction, zoomable user interfaces, Piccolo.NET.

ACM Classification Keywords

H5.m Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Nearly 780 million mobile phones will be sold this year, with yearly worldwide demand projected to top 1 billion by 2009 [12]. As the mobile user base expands, so do device storage capacities and wireless services. Not only are these phones accumulating more device-resident data such as email, appointments and photos, but they are also increasingly used as front-end interfaces to ever-larger external data sets, including web sites, traffic information, and Yellow Pages data. Unprecedented volumes of data in the pockets of users have made mobile search a particularly timely topic. Nearly a dozen simple message service (SMS) query-answer systems and web browser interfaces

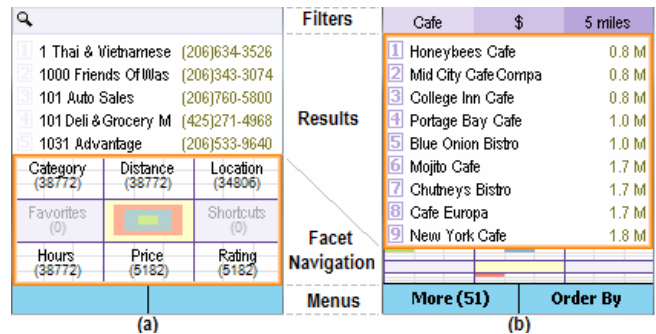


Figure 1. The FaThumb search interface; (a) displays the default configuration, (b) displays results for inexpensive cafés within five miles of the user.

[18,24,25,26] that target mobile platforms have debuted within the last year, including offerings from every major search engine [19,21,23,28].

While existing solutions do cater to small screens and low bandwidth, they are modeled after desktop web search, posing three primary usability issues for the mobile setting. First, they rely on text entry as the method of input, even though the persistent trend toward smaller phones is directly at odds with the goal of achieving the efficiency of full-size keyboard text entry. Second, the focus has been on search *off* the device, under-utilizing the device’s expanding processing power and storage capabilities and thus unnecessarily impoverishing the search UI. Finally, both the SMS and web search models support directed search tasks, but are less appropriate for browsing and exploratory search scenarios (“sense-making”) that are quite complementary to the mobile setting (e.g., “What is fun to do and inexpensive around here?”).

We propose instead FaThumb: a keypad-driven, compact query interface for browsing and searching large data sets from a phone. FaThumb uses a hybrid model to de-emphasize tedious keyword entry in favor of fluid, iterative data filtering using hierarchical faceted metadata navigation and selection, with incremental text entry to further narrow results. Our design is general enough for a wide variety of data sets both on and off device.

After introducing FaThumb (Figure 1), we report on a first iteration user study designed to understand the role of text-based versus facet-based search on a mobile phone.

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. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2006, April 22–28, 2006, Montréal, Québec, Canada.
 Copyright 2006 ACM 1-59593-178-3/06/0004...\$5.00.

RELATED WORK

Many information access interfaces present data attributes (metadata) that users can include in queries to large data sets, rather than expecting users to remember them. Dynamic query interfaces [15] encourage iterative composition and refinement of complex queries by providing continuous visual update of the query results as users restrict data attributes included in the query. When network latency prohibits dynamic queries, query previews [4] provide an overview of the data to lend insight into distribution characteristics and help avoid queries that return either massive or empty result sets. Many desktop systems (e.g., [5,17]) have successfully integrated these concepts to improve exploration and understanding of large data sets. FaThumb retains the spirit of these approaches, and maps hierarchically-organized attributes to the phone keypad in an attempt to offer the greatest access efficiency in the smallest display footprint.

Hierarchical classifications have been used previously in search interfaces. Search results that include hierarchical labels can help users identify relevant items or further refine (or broaden) searches. Search engines such as Yahoo and OpenDirectory order results by relevance, but display each with a human-assigned hierarchical category; following the category hyperlink allows users to browse via the hierarchical category directory. Other systems help users quickly identify appropriate subsets from voluminous results by organizing results into hierarchical categories [3,6,27]. FaThumb is most directly influenced by Flamenco [7] which couples faceted hierarchical attributes with query previews for use across *all* stages of search: query formulation, refinement and results browsing.

A large number of research solutions have been proposed to facilitate web search from a handheld device, primarily by improving results understanding and navigation for small screens [2,8,9,11], with only [16] designed specifically for the keypads. FaThumb instead emphasizes efficient query formulation for keypad-based interaction. Standard bookmarks and saved queries [3] help speed page revisitation, but most systems rely on device-specific text entry methods for ad hoc keyword search. Word prediction and completion algorithms such as Tegic's T9 have the potential to reduce the number of entered characters, but also have the drawback that most fail for non-dictionary words, and may still require users to select from several alternatives. While FaThumb supports keyword search, it favors dynamic result refinement through navigation and selection of hierarchically organized attribute values.

Several search engines and wireless service providers offer solutions to mobile search, either modeled after desktop web search [18,19,21,23,24,28] or using text-based query-response [21,25,26]. A solution from the i411 Discovery Engine [22] which promises to provide incremental, category-based mobile search, has the most potential for

overlap with FaThumb, but neither the interface nor interaction designs have been disclosed.

Unique navigation and presentation techniques are required to accommodate the constrained input and output capabilities of small devices. Robbins *et al.* [13] classified these strategies into three categories in their ZoneZoom design: (1) reformat contents for vertical scrolling, (2) organize content into screen-sized chunks with associated navigation techniques, and (3) use zoomable user interface techniques to navigate data at multiple scales. FaThumb's approach most closely resembles category (2) above, but uses a zooming technique inspired by ZoneZoom for hierarchical navigation: ZoneZoom provides image zooming capabilities by spatially mapping areas of the display to hardware keys; pressing a key causes the display to zoom into the corresponding area of the user interface.

METADATA

Search vs. Browse

Information seeking strategies take many forms depending on the task at hand, user knowledge, and target data set. For example, a user may have in mind a specific document she wishes to find on her desktop. Though the task may be considered one of directed search, her approach will be influenced by whether she remembers the name of the file (in which case she would look for the file by name) or only the period of time in which it was last modified (causing her to look at modified documents within a constrained time span). If the task involves finding a research paper on the web, she might instead search by some combination of topic, author, or related papers. This approach may appear superficially similar to one she would use if instead she were generating a list of references, although in that case it might be considered a *browsing* activity given its open-ended nature. We will use the following definitions for the purposes of this paper: a *directed search* is one in which the user knows the precise target in advance, while a *browsing* task is one characterized by specifying criteria that describe a data need, and which may evolve during search. These concepts map closely to what others term *navigational* and *informational* queries for web search [1,14].

Faceted Metadata

Leveraging data attributes for search has received much attention in recent years. Hearst *et al.* [7] demonstrated the use of data attributes (metadata) organized into orthogonal dimensions (facets) as a means not only to structure search results, but as a tool to guide users in formulating powerful Boolean queries. This approach not only reduces cognitive load through recognition, but allows users to reliably restrict results by attribute values rather than by keyword alone. Hearst's evaluations over both text and image-based data sets have established the efficacy of faceted metadata within integrated search and browse environments.

Although the metadata values within a facet can be organized hierarchically, attribute values need not have a unique classification. This means that the faceted hierarchy is in fact a directed acyclic graph, and that an attribute value may be reached via multiple paths in the hierarchy. The same holds true for each data record, which may be found via any of its attributes. For example, the “Terrier” record may appear in a pet database under the “Small” value in the “Size” facet, the “Brown” value of the “Color” facet, and the “Dog” value of the “Type” facet.

With FaThumb, we propose an analogous method for search both on and off device. We have developed a prototype to explore whether the approach scales to mobile devices. In our study, we use a Yellow Pages data set for the Seattle metropolitan area (about 39,000 listings), but our design is intended to generalize to a variety of data sets, including personal (contacts, email, etc.) and public (web pages, movie listings, etc.) repositories.

Data Preparation

We obtained a Yellow Pages data table for Washington state with attributes of business name, address, phone number, latitude, longitude and associated Standard Industrial Classification (SIC) code number (http://www.osha.gov/pls/imis/sic_manual.html). For the purposes of investigating a realistically rich data set, we augmented the metadata to include neighborhood, relative distance from four pre-set static locations (a fictional user’s “current” location, school, home and place of work), price (\$ to \$\$\$\$), rating (one to four stars), hours, and days of operation. Location-based values were computed from latitude and longitude for each item, while values for the remainder were assigned randomly, with only restaurants receiving price and rating classifications.

Although several classification hierarchies for Yellow Pages data exist [20,28,29], they are characterized by large numbers of items at each level, and are thus best suited for

desktop presentation and navigation [10]. Drawing on these existing on-line Yellow Pages resources and intuition, we therefore developed our own attribute classification hierarchy targeting small screens with the following properties: eight top-level facets, a maximum of eight metadata attributes at each lower level, maximal depth of five, and just over 200 terminal attributes. Some attributes were naturally hierarchical (e.g., location, which can be organized by increasing granularity: address, neighborhood/town and region), while for others, hierarchy was imposed as an organization strategy (e.g., relative distance was broken down by type: from me, from work, from school, from home). To ensure Yellow Pages entries could be navigated by business type, we developed a mapping between SIC codes and more user-friendly business types, organized hierarchically into a top-level *category* facet.

INTERFACE DESIGN

Interaction Framework

The FaThumb interface has four distinct structural regions (Figure 1). Among the top three regions (Filter, Results and Facet Navigation) an orange activation border serves to identify the *active* region, which is the one receiving input from the keypad. In Figure 2a, the Facet Navigation region is the one that is active. Users change the active region by pressing the up or down directional keypad arrows, causing the border to animate to the adjacent region. In keeping with Smartphone design guidelines, the bottom region (Menu) is reserved for two menus accessed via proximal hardware buttons. The menus’ availability and functionality are dependent upon the state of the active region.

Facet Navigation Region

The Facet Navigation region represents a navigable tree of hierarchically organized facets, which contain metadata attributes of the Yellow Pages data set. Each non-leaf node in the tree depicts a 3x3 grid of nine zones (Figure 2). Each zone displays a facet or attribute label and a number indicating how many data records can be characterized by the label. Figure 2a displays the root of the facet tree, from which we see that Yellow Pages listings can be described according to business *category*, relative *distance* from both predefined and real-time locations, absolute *location*, *hours* of operation, *price* classification, and consumer *rating*.

Three zones at the root level have specialized functionality. The Favorites and Shortcuts facets are reserved for future use. Favorites will store entire queries (combinations of selected attributes and/or free-text search terms) the user has chosen to save, and Shortcuts will populate dynamically with labels from arbitrarily-deep locations in the overall value tree based on usage patterns, such as most recently or most frequently used, values. The middle zone, on the other hand, is not a facet at all, but an Overview zone which serves as a spatial overview during navigation.

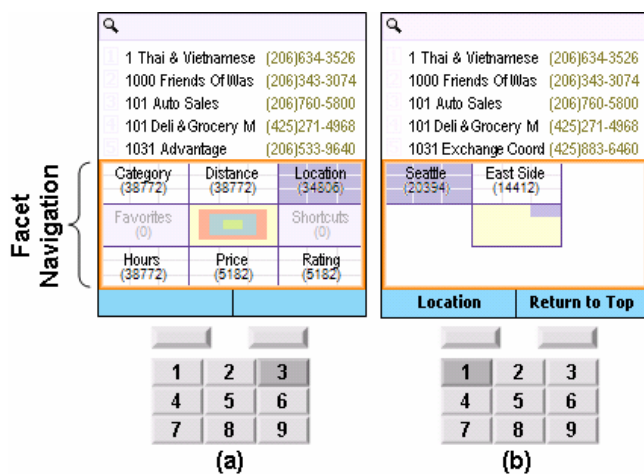


Figure 2. FaThumb facet navigation. (a) Pressing 3 navigates to Location. (b) Pressing 1 navigates to Seattle.

Structural Philosophy

Most mobile phones lack touch screens so FaThumb is optimized for keypad interaction. The Facet Navigation region is intentionally designed to map spatially to numbers 1 through 9 on the numeric keypad (e.g., Figure 3). While this design restricts the branching factor of the hierarchy (with a maximum 8 at each level), its depth and balance are dictated only by the target data set. For any domain, we believe consistency of facet location is crucial to promoting user mastery of the hierarchy. Thus we opted for a terminating tree, meaning users must return to the top of the tree to explore paths that lead from other top-level facets. On the other hand, as data sets grow, it may be appropriate to dynamically generate nodes within the facet tree to provide more efficient distribution of data among the available zones (e.g., consider date facets labeled by day for high frequency data, but by year for low frequency data). Dynamic strategies may be most effective at lower levels of the tree, since users may be more willing to browse a less familiar but more useful set of choices once they have already narrowed the data set by a few familiar initial selections.

Navigation Region Interaction

Users perform facet-based queries by interacting in the Facet Navigation region. A user navigates the facet space by selecting the zone she wishes to explore within the 3x3 grid. Pressing a number on the keypad zooms the display into the associated zone: internal tree nodes zoom to the next hierarchical level, while leaf nodes expand to occupy the entire navigation region (Figure 3b). Each traversal propagates navigation state information to the Results region, which serves as a query preview by showing the data set filtered by the selected facet value. A zooming animation accompanies each navigation.

At each traversal, the left menu changes to display the selected attribute label, both to remind the user of the filter that has been temporarily applied to the data set, as well as to allow the user to add the attribute permanently to the current query; pressing the left menu button adds the attribute label to the top Filter region as a search term (Figure 3b). Pressing the left menu button a second time removes the attribute term from the Filter region. When displaying nodes other than the root, the right menu button serves as a shortcut to the top of the facet hierarchy. Alternatively, pressing the back button returns to the previous level, and updates menus and results accordingly. As an example of query formulation via facet selection, a user interested in inexpensive restaurants in downtown Seattle, might (a) navigate to and select Location→Seattle→Downtown, (b) return to the root, navigate to and select Category→Restaurants, (c) return to the root, navigate to and select Price→\$. Elements of (a) are shown in Figures 2 and 3. Since the Restaurant facet can be subdivided further, the selection in (b) illustrates that

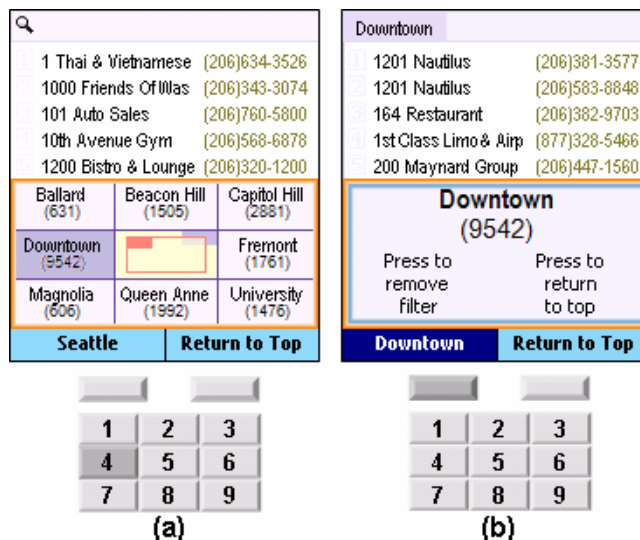


Figure 3. Facet navigation continued from Figure 2. (a) Pressing 4 navigates to Downtown. (b) Pressing the left menu button adds Downtown to the query.

attributes can be saved to the query at any level of the hierarchy, not just the leaves.

Overview

Although navigational context might take many forms (global vs. local, descriptive vs. iconic, etc.) our goal for the initial prototype was to explore designs for reinforcing spatial and motor memory to support users in learning key sequences to facet tree locations.

The Overview can be thought of as a stepped-pyramid viewed from above; each hierarchical level corresponds to a layer of the pyramid, with the root as the pyramid's foundation. In the root view (Figure 2a) the Overview serves as a key for FaThumb's color cues, which map each level of the tree to a color. When a user presses a number on the keypad to navigate into a facet, the Overview captures the selection by displaying a block whose color corresponds to the current depth, and whose position within the Overview corresponds to the spatial position of the selection on the keypad. For example, pressing 3 at the root to navigate to Location adds a purple block to the Overview in the upper right corner since 3 is in the upper right position of the numeric keypad (Figure 2). As the user navigates deeper in the hierarchy, smaller levels are added to the pyramid to preserve previous selections (Figure 3a), and thus convey the entire spatial selection sequence from the root. This design is able to give both a sense of where the user is in the hierarchy and what command sequence resulted in that location.

Visual Encodings

Visual cues help convey fruitful navigation paths to users: leaf nodes are shown in flat purple (e.g., all nodes in Figure 3a), while zones corresponding to internal tree nodes are decorated with a faint grid to indicate the presence of

deeper facet levels (e.g., Location in Figure 4e). The distribution of data across the zones (depicted as numbers) serves as a preview to encourage or discourage exploration of a path, depending on the task. Zones corresponding to facet tree nodes with no associated data records are de-emphasized using gray text (e.g., Favorites in Figure 4c), and cannot be selected. Each facet-based search term added to the Filter region is visually tagged with a colored border in the Facet Navigation region. The visual tags propagate to higher levels when the selected facet is buried within a zone's sub-tree as a way of providing information scent for the query terms (see Location in Figure 4e).

Results Region

Results Region Interaction

Once a user is satisfied with the search terms that have been added to the query (any set of facet filters and/or free-text terms), she can explore the remaining records by using the Results region. When the Results region becomes the *active* region, it expands from the original five records to accommodate the first nine records (Figure 4a). The visible items are then numbered from 1 to 9 to support single-key access to listing details. (The current prototype stops short of showing the listing details but users in our study were able to easily extrapolate the behavior.)

The left menu indicates the number of results which follow those displayed, if any (Figure 4a). Pressing the left soft-key displays the next page (up to 9) of results. Once the end of the result list is reached, the left menu allows users to continue from the beginning of the list. (In anticipation of limited bandwidth and user patience, the current prototype only displays the top 200 records.) Users can page backward through results using the back button on the keypad. We chose to not use the up and down directional keys for scrolling of the results because these buttons are dedicated to moving between regions of the interface.

A result record consists of two pieces of data: an identifier (which for the Yellow Pages records is the business name) and an attribute (which by default is the phone number). Since data records have not one, but many attributes, users

can cycle through the alternative attributes by pressing the left and right navigational arrows.

By default, results are ordered alphabetically by identifier, but they can instead be ordered by other data attributes. Pressing the right soft-key triggers a menu of available order options, which for the Yellow Pages data include: by alphabetical sort (forward and backward), by relative distance (from user, home, school, and work), by neighborhood, by price and by rating (Figure 4b). A checkbox is displayed in the menu next to the current ordering. Each option is numbered 1 to 9; selecting the associated number reorders the result list and displays the selected sort attribute in each record's attribute column. For example, in Figure 4c, the results have been ordered by consumer rating.

Filter Region

The Filter region holds both the facet-based attributes and freeform text terms which define the current search query, generically referred to as filters. Filters are displayed in the Filter region as colored, labeled blocks that each occupy one third of the screen width. The Filter region can hold an arbitrary number of filters. While a maximum of only three filters are visible at any time, white arrows are displayed along the left and right borders of the Filter region to indicate the presence of off-screen filters.

Facet versus Text Filters

Facet-based filters partition the data into two sets: items with metadata that match the given attribute, and items without. Freeform text filters also partition the data, but instead by performing a string match over all meaningful record text. Although an attribute may have a hierarchical "address" describing its position within the facet hierarchy, only the text of the leaf node in the hierarchical path is displayed in the filter bar. However, the entire address from node to root is used as the filter criterion. In contrast, no assumptions are made about the context of a freeform text term, which is matched against a predefined subset of the data and metadata. This approach introduces a fundamental asymmetry between facet selection and text entry in query

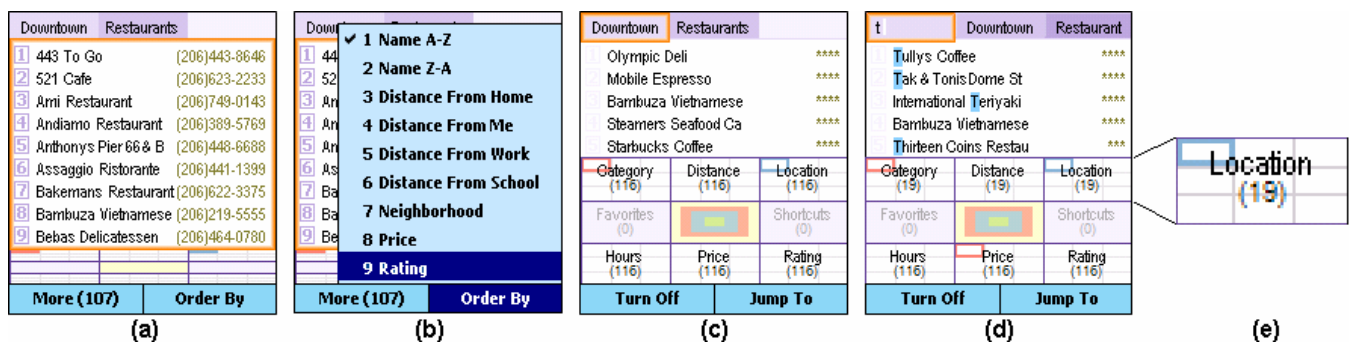


Figure 4. Results interaction. (a) When active, the results region expands to display up to 9 items. (b) The right menu button opens the Order By menu; pressing 9 sorts results by Rating. (c) The up arrow moves the activation region to the filter bar. (d) Typing in a search term updates the results. (e) A facet node in detail.

formulation: selecting “Italian” as a facet filter differs from entering “Italian” as a text filter. In the former case, only Italian restaurants will appear in the result set. The latter query term will include all of the listings of the former, as well as those that have “Italian” in any indexed attribute, such as a business named “Italian Auto Specialists”.

Filter Region Interaction

When the Filter region becomes *active*, the leftmost filter is automatically selected and shown with an orange selection border (Figure 4c). The left and right buttons move the selection border to adjacent filters within the Filter Region, scrolling to off-screen filters as necessary. Moving the activation border beyond the rightmost filter adds a new (empty) text filter to the query.

As long as the Filter region has focus, users can enter freeform search text using the numeric keypad. If the active filter is a text filter, text will be appended to the existing filter text. If instead the activation border is over a facet-based filter, the existing filters will slide right to make room for a new text filter (Figure 4d). Multi-word text in the same filter is treated as a phrase, while multiple separate text filters are treated as separate search terms.

Text Entry

At present, FaThumb supports multi-tap text entry, whereby each keypad number 2-9 is assigned 3 or 4 alphabetic characters. Pressing a key once enters the first character, twice the second, etc. A 600ms timeout determines whether a consecutive tap on the same key overwrites the last character or appends a new character. After each timeout, the entire query is reissued and results updated, effectively providing incremental text search. Substrings in the results region that match any of the freeform text filters are displayed highlighted in light blue (Figure 4d).

Multi-tap is the original method for text entry using a numeric keypad. While T9 single-tap word prediction has gained popularity, without modification it is unlikely to be effective for search over domains that use proper names (e.g., Yellow Pages) or user defined names (e.g., local data). FaThumb can be adapted to incorporate the most appropriate technique for the data.

Menu Options

When the selection border is over a filter, menu options only affect the active filter. The left softkey allows users to preview the removal of a filter. When pressed, the selected filter is disabled and removed from the query: the filter text becomes gray and the Results and Facet Navigation regions repopulate to reflect the new query. Pressing the left softkey again re-enables the filter, again propagating to the Result and Facet Navigation regions. This toggling behavior lets users easily see how that filter narrows the results set. This behavior is especially important when the user is inspecting data with which they are unfamiliar. If the activation border

is moved to the Results region while some filters are inactive (grayed), those filters are removed permanently from the filter region and the current query. Remaining filters then slide left to remove gaps and visually reinforce the current query. Zero-length text filters are also removed.

For facet-based filters, the right menu serves as a shortcut to its position within the facet navigation hierarchy. Pressing the right softkey animates movement of the activation border to the facet navigation region, at the same time animating a traversal through the hierarchy to the target filter. The shortcut may be convenient in the case that a user wants to expand a query to explore options at the same level in the facet hierarchy as a previously selected facet. We believe such convenient exploration interaction eliminates the need for an explicit OR query operator.

FaThumb Implementation

FaThumb is implemented on top of Piccolo.NET (<http://www.cs.umd.edu/hcil/piccolo/>), chosen because it runs on both desktop PC’s running Windows and on the Microsoft Windows powered Smartphones. Piccolo.NET supports zooming and animated transitions, both key design elements of FaThumb. The current prototype runs in a simulated phone environment on a desktop PC that has the same display and input limitations as the Smartphone. For the user study, we use an external keypad programmed to have the same key layout as the Smartphone. The multi-clause faceted queries defined by the FaThumb users’ interactions are implemented by building and executing dynamically-generated Structured Query Language (SQL) statements for wide portability across a variety of database systems. The prototype currently connects to a desktop Microsoft SQL Server database, but it is expected that a full SQL (SQL Mobile) will be available soon on the Windows Mobile Smartphone platform. Meanwhile, our desktop prototype can lend insight into the value of faceted search.

USER STUDY

We conducted a user study designed to help us understand the value of facet-based navigation compared to text-based search on a mobile phone. It seemed clear to us that text-based search should be better for certain types of tasks, like looking for a particular business when the name is known. It also seemed clear that facet-based navigation should be better for less well-defined or more complex searches. For example, finding the closest, expensive, Chinese restaurant would be quite hard (or impossible) using only text-based search, since the name of the business is not known. But to level the playing field between the two methods of search, we allowed search of the facet space using text entry. That is, users could type “Chinese” and “\$\$\$” and get a list of expensive Chinese restaurants. The study was designed to test the above hypotheses as well as assess subjective preferences for the two techniques, in addition to collecting first iteration feedback about the FaThumb user interface.

Participants

Participants were recruited with the following characteristics: each must have owned a mobile phone with numeric keypad entry for 3 months and have sent at least 2 text messages a week. Seventeen participants (10 female) ranged in age from 20 to 53 with an average of 29.2 years of age and sent an average of 17 text messages per week. Participants received 2 software gratuities for their time.

Method

The study design was a 2 (input type: text entry v. facet navigation) x 3 (search type: directed v. simple browse v. complex browse) x 4 trials within subjects design. Input type was counterbalanced across participants, as were the task sets (2 isomorphic task sets were rotated through the conditions such that each task was performed by half the users via text entry, and half via facet navigation). However, search type was presented to subjects in order of increasing complexity, moving from directed to simple to complex browse.

Equipment

The study was run using a desktop simulation of the Smartphone user interface. A 2.8 GHz Pentium 4 Compaq PC with 2G of RAM was used to drive two side-by-side NEC MultiSync LCD 1880SX monitors set at 1280x1024 resolution. Users interacted with the FaThumb interface using an Ergodex DX1 Input System configurable keypad arranged to mimic the Smartphone keypad. A Compaq keyboard and Microsoft Intellisense mouse were used as input for question answering at the completion of each task.

Tasks

To investigate whether facet navigation could outperform keyword search, even with an unfamiliar hierarchy, task were carefully balanced for query complexity. Under these constraints, we did not expect to faithfully preserve user-based tasks. Tasks were therefore chosen to be representative of those we imagined real users would perform on their phone under various conditions: at work, at home, and while in mobile situations.

Twelve pairs of isomorphic tasks were developed. To be representative, four task pairs were assigned to each of three search types, which increased in complexity based on the number of facet selections or free-text terms required (directed search, simple browse, and complex browse). For example, directed search tasks involved searching for a specific business, while browse tasks presented varying numbers of target attributes. We also classified tasks as “shallow” or “deep” according to the minimum number of hierarchical levels visited to complete the task, although final analysis was strictly by search type. Tasks were ordered by complexity (first by search type, then depth) to provide users familiarity with the hierarchy before proceeding to more complex tasks, which might be unfairly penalized otherwise. Example tasks are presented in

Table 1. Although task descriptions contained the required search terms, the terms were not emphasized visually in any way and so had to be derived as part of the task.

Search Type	Depth (Levels)	Task Text (query terms underlined here only)
Directed Search	Shallow (3)	What is the address of the <u>BlueBottle Art Gallery</u> ?
	Deep (4)	What is the address of the <u>Safeway Grocery</u> closest to me?
Simple Browse	Shallow (2+2)	What 4-star (****) <u>restaurant</u> is closest to home?
	Deep (4+3)	What <u>seafood</u> places are there in <u>Issaquah</u> ?
Complex Browse	Shallow (2+3+3)	What inexpensive (\$) <u>repair store</u> in <u>Queen Anne</u> is closest to me?
	Deep (3+3+4)	What's the name of the <u>24-hour pharmacy</u> in <u>Fremont</u> ?

Table 1. Example study tasks.

Measures

Dependent variables collected during the study included task time, error rate, logs of interactions per task, user satisfaction ratings and overall preference. We also collected observed usability issues and comments from our participants. All measures other than satisfaction ratings and preferences were automatically collected via logging tool installed on the participant's machine.

Procedure

Participants were run in pairs with experimenter present. After greeting the participants, the experimenter presented the study procedure and walked the participants through an interactive tutorial, highlighting all key interaction features of the interface design. Upon completing the tutorial, eight practice tasks representing each condition of the study were carried out with the experimenter watching. Once the practice phase was completed (approximately 1 hour), the study proper was begun. Participants performed all tasks relating to either text entry or faceted navigation in a single block of trials before moving on to the other input type.

Study control software was run on the left monitor and the interface was presented on the right. A trial consisted of the following steps. After reading the task, users hit a start button to begin task timing. Upon finding the search target, users hit a stop button and entered the answer in a text field. While correctness was not required to move onto the next trial, we did analyze for percent correct after the session. Participants hit a ‘next task’ button to proceed. Upon completion of all 12 search trials (3 search type x 4 trials) for a given input type, users completed a satisfaction questionnaire and proceeded to the next block of trials. At the end of the entire session, users provided overall preference ratings and general comments. The experimenter debriefed the participants as to the purpose of the study,

provided the software gratuity and escorted them out of the building. Total session time was approximately 2 hours.

Results

Task Times

A 2 (input type: text entry v. facet navigation) x 3 (search type: directed search v. simple browse v. complex browse) x 4 trials repeated measures Analysis of Variance (RM-ANOVA) was carried out on the task time data. In order to run the analysis, averages were used to fill cells of missing data. Missing data was a problem because incorrect answers were sometimes provided and could not be included in the task time analysis (~22% of the data). In addition, 4 trials were lost due to system failure and 3 trials were thrown away due to misleading task phrasing. Main effects of search type, $F(2,32)=13.9$, $p<.001$, and trial, $F(3,48)=3.9$, $p=.015$, were observed. Interactions of input x search, $F(2,32)=12.7$, $p<.001$, search x trial, $F(6,96)=4.36$, $p<.001$, and a 3-way interaction of input x search x trial, $F(6,96)=3.7$, $p=.002$, were also observed.

On average, facet navigation was slightly faster than text entry, (62.9s v. 63.9s), but the difference was not significant. As for search type, post hoc analyses with Bonferroni corrections showed that the simple browse condition was significantly faster than either directed search or complex browse at the $p=.05$ level. The trials did show an increase in complexity as per design, since the fourth trial was significantly slower than the others (no other significant differences between trials were observed).

For an explanation of the search x input interaction, see Figure 5. Starting with text entry, it is clear that going from directed search to simple and then complex browse slows the user down. This is not surprising since knowing a name of a search target (directed search condition) is advantageous to the text entry task. However, this was not the case for faceted search task times. The faceted simple browse condition was significantly faster than the others. However, it is also clear that when the task provided a specific target name as in the directed search condition, facet navigation was much slower. In other words, if one knows the search target by name, searching via text entry will be significantly faster than finding the same target

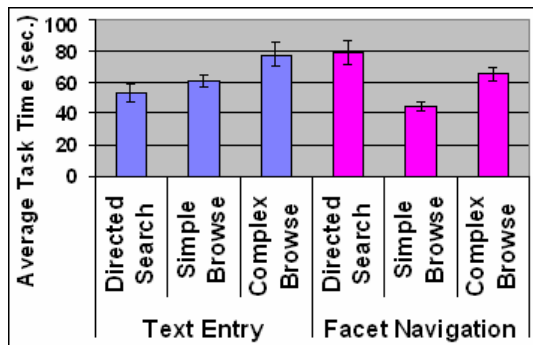


Figure 5. Task times for search conditions by input type.

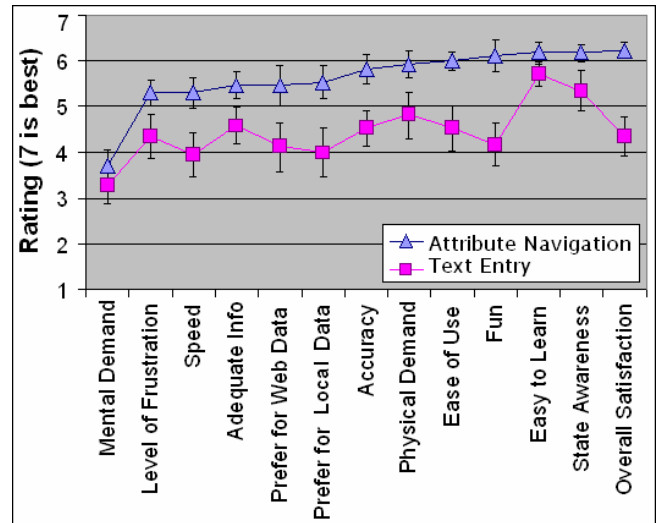


Figure 6. Satisfaction ratings for usability criteria.

using facet navigation. If instead a user is browsing for targets unknown in advance, facet navigation will outperform text entry, with time increasing with task complexity. All other interactions can be explained by the designed increasing complexity of the trials within a block.

Percent Correct

A 2 (input type: text entry v. facet navigation) x 3 (search type: directed search v. simple browse v. complex browse) x 4 trials repeated measures Analysis of Variance (RM-ANOVA) was carried out on the percent correct data from the search trials. No significant main effects were observed at the $p=.05$ level. However, input x trial, $F(3,48)=4.6$, $p=.007$, and search x trial, $F(6,96)=2.8$, $p=.01$, interactions were observed. The input x trial interaction can be explained by the fact that as trials got increasingly more complex, text entry became more difficult, while faceted navigation did not suffer as much. The search x trial interaction suggests that the simple browse search tasks suffered more from the increasing complexity (depth) of trials than did the other two search task types.

Satisfaction

A 2 (input type: text entry v. facet navigation) x 13 (satisfaction questions) RM-ANOVA was carried out on the satisfaction questionnaire ratings. Significant main effects of input type $F(1,16)=10.2$, $p=.005$, and questionnaire item $F(12,192)=7.02$, $p<.001$ were observed. In addition, a significant interaction between input type and questionnaire item was observed, $F(12,192)=2.4$, $p=.006$. Text entry was rated significantly lower in terms of satisfaction than was facet navigation overall (average text entry rating = 4.4, average faceted navigation rating = 5.6, with 7 being high). Scales for mental demand, physical demand, and frustration were reversed so that high ratings indicated positive opinion. Average ratings for each input type are indicated in Figure 6. User ratings were high overall for both input types given this was a first iteration study of FaThumb.

User comments

User comments were optional, and overall quite positive. We report only on the strongest trends, starting first with those for text entry, then those for facet navigation.

Although two users' comments were strongly supportive of text entry, the remainder (12 users) tended toward neutral or negative. Participants most consistently (5) expressed frustration with text entry when tasks included constraints that were not searchable, such as days or hours of operation. Some (4) offered (correctly!) that text entry would likely be most effective when a specific business name was known in advance, but would prefer facet navigation otherwise.

By contrast, participant comments suggested general enthusiasm for facet navigation, including that it was easy to learn and use (4 participants), fun (4), fast (3) and effective in increasing response confidence (2). The most common negative comment was frustration with identifying the appropriate business classification for task targets. Because the four facet navigation practice tasks provided users only minimal exposure to the hierarchy prior to the timed phase of the study, users were forced to rely on exploration and intuition to locate facets, which at times resulted in lengthy navigation.

Usability Observations

Participants primarily requested minor feature additions rather than offered fundamental usability suggestions. Thus we draw on our analysis of user error types and experimental observations to better understand usability issues in the current design. To investigate reasons for error, we reconstructed navigation paths for tasks answered incorrectly, and then classified error types into 5 categories:

1. *Incomplete search criteria (~36%)*: The majority of errors occurred when users failed to use all search criteria, or used incorrect criteria. Since these errors manifested only for tasks that required multiple search terms, the most likely explanation is that complex tasks taxed user memory. Had tasks originated from personal needs, such errors may have been greatly reduced.
2. *Failure to sort (20%)*: Users sometimes forgot to (or chose not to) sort the data set, which resulted in task error. This suggests that some users do not map tasks involving comparison terms (closest, cheapest, best, etc.) to sort operations.
3. *Incorrect sort criteria (~13%)*: While some portion of the sorting errors can be attributed to memory errors, the sort interaction clearly posed a usability problem. To order results, users were required to map a linear list of menu items to the numbers 1-9 on the keypad. Despite visual feedback provided during they key press, users were observed returning to the menu to verify selection. Serial item selection using the up and down arrows may have improved confidence and reduced errors, but would have been slower. A 3x3 grid may also be viable for

generalized menu design, with the caveat that 2D layout disrupts visual scan, and would thus be more suited to menus with familiar content.

4. *Results scan errors (~16%)*: When sorting could not be used to place the target item at the top of the results list, participants made identification and copy errors when scanning results. These problems were likely due to the fact that the prototype did not allow users to highlight or select items, which otherwise would have allowed users to visually anchor or verify choices. Item selection within the result list presents similar problems to those for menu item selection. In this case, however, a 3x3 layout is inappropriate since the dynamics of the list make fast linear scan a priority. Using up/down arrows to highlight items is also problematic, since it interferes with their dedicated use in activating interface regions. Fortunately, highlighting an item by pressing numbers 1-9 has a lower commitment cost than for menus, which close immediately upon item selection. Here, users can verify a selection before pressing "enter" for further details.
5. *Business classification errors (10%)*: Small zones limited the length of facet and attribute labels, while a narrow, relatively shallow navigation tree lead to business categories that were quite general. Both issues resulted in classification ambiguity. At the same time, some facets were arguably misclassified (Salons were located under Category→Shopping→Personal→Hair, rather than Category→Services→Businesses). Iterative refinement of the hierarchy and user familiarity would certainly reduce these errors, but they are likely to be typical of errors encountered for any new classification. This makes it all the more encouraging that users provided such strong preference for facet navigation.

DISCUSSION AND FUTURE WORK

Based on user feedback and our own observations, we found users adapted quickly to the facet navigation and selection, suggesting the spatial arrangement and hierarchical structure of facets holds great potential for search. However, users were at times frustrated when items were not classified as expected, and some either did not understand the role of sort for search tasks or did not understand how sort was supported in FaThumb.

FaThumb's primary challenge may be its generalization to alternate data sets. Given the positive response to facet navigation, successes must have either been more common than failures, or otherwise were more gratifying than user mistakes were frustrating. Even so, first-time users struggled with the classification scheme for some tasks. We would like to extend the FaThumb paradigm to a wider variety datasets such as email, contacts, appointments, photos, cached web pages, traffic information, stocks, music collections, or movies listings, in order to understand the elements of classification structures that promote satisfying user experiences.

Another potential concern is that some users did not sort the result list when necessary. One user gave insight into the problem when he commented he would have liked “closest to me” as an attribute in the facet tree. This suggests some users may not view attribute selection and sorting as fundamentally distinct operation types. We would therefore like to investigate whether integrating a subset of sort operations into the facet space improves performance for appropriate classes of tasks.

We designed FaThumb as a search solution for devices with limited input and display capabilities - specifically keypad-based mobile devices. While we have demonstrated the viability of our interface, technical limitations prevented us from validating the design under mobile scenarios. We therefore plan to port our prototype to Smartphones and study FaThumb in the field with up-to-date databases either stored locally or accessed remotely via GPRS.

Finally, we anticipate user comfort with zone-based navigation will transfer across a broad array of devices and contexts. We intend to investigate its potential for phone-based menu structures, as well as explore how a zone-based faceted search interface might gracefully transition from a phone to a desktop to a wall-size display form-factor.

CONCLUSION

We described a novel search user interface for navigating large data sets on a mobile phone. We discussed our design motivations and explored their efficacy via an empirical user study using a wide variety of search complexity in the tasks. The user study confirmed our basic hypotheses – if you know something specific like the target name, text entry (even on a phone) is faster. If you only know data characteristics, facet navigation is faster. Ultimately, both techniques are needed because real world tasks require both. Comments made by study participants suggest that when users become familiar with the taxonomy, the FaThumb facet navigation technique is highly desirable.

ACKNOWLEDGMENTS

We would like to thank Sue Dumais for her insights during design discussions of FaThumb.

REFERENCES

1. Broder, A. A taxonomy of web search. *SIGIR Forum*, 36, 2 (2002), 3-10.
2. Büring, T. and Reiterer, H. ZuiScat - querying and visualizing information spaces on personal digital assistants. *Proc. Mobile HCI'05*, ACM Press (2005), 129-136.
3. De Luca, E.W. and Nurnberger, A. Supporting information retrieval on mobile devices. *Proc. Mobile HCI* (2005), 347-348.
4. Doan, K., Plaisant, C. and Shneiderman, B. Query previews in networked information systems. *Proc. ADL*, IEEE Press (1996), 120-129.
5. Dumais, S., Cutrell, E., Cadiz, J.J., Jancke, G., Sarin, R. and Robbins, D.C. Stuff I've seen: a system for personal information retrieval and re-use. *Proc. SIGIR'03*, ACM Press (2003), 72-79.
6. Dumais, S., Cutrell, E. and Hao, C. Optimizing search by showing results in context. *Proc. CHI'01*, ACM Press (2001), 227-284.
7. Hearst, M., Elliot, A., English, J., Sinha, R., Swearingen, K. and Yee, K.-P. Finding the flow in web site search. *Comm. of the ACM*, 45, 9 (2002), 42-49.
8. Jones, M., Buchanan, G. and Thimbleby, H. Sorting out searching on small screen devices. *Proc. Mobile HCI'02*, Springer-Verlag (2002), 81-94.
9. Jones, S., Jones, M. and Deo, S. Using keyphrases as search result surrogates on small screen devices. *Personal and Ubiquitous Comp.*, 8, 1 (2004), 55-68.
10. Larson, K. and Czerwinski, M. Web page design: implications of memory, structure and scent for info. retrieval. *Proc. CHI'98*, ACM Press (1998), 25-32.
11. Milic-Frayling, N., Sommerer, R., Rodden, K. and Blackwell, A. SearchMobil: web viewing and search for mobile devices. *Posters of WWW'03*, ACM Press (2003), 320.
12. Forecast: Mobile Terminals, Worldwide, 2000-2009, Gartner, Inc., 2005.
13. Robbins, D.C., Cutrell, E., Sarin, R. and Horvitz, E. ZoneZoom: map navigation for smartphones with recursive view segmentation. *Proc. AVI'04*, ACM Press (2004), 231-234.
14. Rose, D.E. and Levinson, D. Understanding user goals in web search. *Proc. WWW*, ACM Press (2004), 17-22.
15. Shneiderman, B. Dynamic queries for visual information seeking. *IEEE Software*, 11, 6 (1994), 70-77.
16. Xie, X., Miao, G., Song, R., Wen, J.-R. and Ma, W.-Y. Efficient browsing of web search results on mobile devices based on block importance model. *Proc. Pervasive Computing and Communications*, IEEE Press (2005), 17-26.
17. Zhang, J. and Marchionini, G. Evaluation and Evolution of a Browse and Search Interface: Relation Browser++. *Proc. Digital Gov't Research*, Digital Gov't Research Center (2005), 179-188.
18. <http://www.4info.net/>.
19. <http://www.aol.com/>.
20. <http://www.citysearch.com/>.
21. <http://www.google.com/>.
22. <http://www.i411.com/>.
23. <http://www.msn.com/>.
24. <http://www.nokia.com/>.
25. <http://www.synfonic.com/>.
26. <http://www.upsnap.com/>.
27. <http://www.vivisimo.com/>.
28. <http://www.yahoo.com/>.
29. <http://www.yip.com/>.