

# ResultMaps: Visualization for Search Interfaces

Edward C. Clarkson, *Student Member, IEEE*, Krishna Desai and James D. Foley, *Fellow, IEEE*

**Abstract**— Hierarchical representations are common in digital repositories, yet are not always fully leveraged in their online search interfaces. This work describes ResultMaps, which use hierarchical treemap representations with query string-driven digital library search engines. We describe two lab experiments, which find that ResultsMap users yield significantly better results over a control condition on some subjective measures, and we find evidence that ResultMaps have ancillary benefits via increased understanding of some aspects of repository content. The ResultMap system and experiments contribute an understanding of the benefits—direct and indirect—of the ResultMap approach to repository search visualization.

**Index Terms**—Treemap, evaluation, user studies, digital library, digital repository, search engine, search visualization, infovis.

## 1 INTRODUCTION

Hierarchy is a fundamental organizational paradigm. People use hierarchy to abstract key concepts from groups of similar items and to help structure their reasoning. One common use of hierarchy is as an ontology, such as the ACM Computing Classification system. As libraries have moved to the World Wide Web (WWW), so has their use of hierarchical classifications. Naturally, such hierarchies are often a central component for browsing online repositories. But their comparable use in repository search is generally limited. In a survey of educational digital libraries, we have found hierarchies are underutilized in the context of repository search beyond their use with simple filtering [9].

We developed *ResultMaps*, a treemap-based [19] search visualization system, as a way of using existing hierarchical metadata to enhance digital library (DL) search engine result pages (SERPs). ResultMaps use hierarchical tree structures (e.g., a subject classification) to map each repository document into a treemap and highlight items that correspond to query results on the current SERP. Interaction links those nodes to traditional text listings.

The text listing leverages users' familiarity with that environment and provides a connection to the more sophisticated ResultMap representation of the same data. ResultMaps encode the full contents of a hierarchical dataset, preserving visual consistency between queries. Moreover, it provides a view into lower levels that is lost by flattening the hierarchy. This work details our ResultMap approach along with two lab evaluations of its use, which find that ResultMaps provide some ancillary and subjective benefits. We conclude with some comments on evaluating infovis and information-seeking tools.

## 2 RELATED WORK

Other types of information search applications that relate to hierarchy—faceted navigation systems, for instance—might also benefit from ResultMaps. Indeed, our current work concerns faceted applications from both a theoretical [10] and experimental perspective. But converting a data repository to use faceted metadata is not a trivial task, and directed keyword search systems are still highly prevalent, especially in DLs. Yet as a matter of course such environments lack useful contextualization: consequently, this work focuses specifically on a SERP application.

- Edward Clarkson graduated in August 2009 with a Ph.D. in Computer Science from Georgia Tech, E-Mail: edward.clarkson@gatech.edu.
- Krishna Desai is an undergraduate student in the College of Computing at Georgia Tech, E-Mail: krishna.desai@gatech.edu
- James Foley is a Professor in the College of Computing and GVU Center at Georgia Tech, E-Mail: foley@cc.gatech.edu.

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

Researchers have approached visualization of search results and relevancy in at least two ways: relevance with respect to search terms, and relevance within the context of the overall information space. The VIBE system, which shows how different queries influence relevance [32], was an early attempt to create meaningful context beyond simple relevance-based lists. Veerasamy and Belkin addressed the same problem using a different visualization approach [39]. They compared their system to a text-based interface and found a few significant differences in favor of their visualization.

But as a WWW search paradigm has emerged and become more ingrained for even casual users [29], it is a challenge to assist users without detriment to their expected usage environment. In-browser, VRML applications [34] are nominally WWW infovis systems, but do not mesh with everyday WWW environments. In contrast, Hearst's TileBars [16] do well at minimally invading users' expected environment by augmenting search results with small, informative glyphs. Our work focuses on the second of the two infovis approaches, so we turn now to previous work related to visualizing search results as they relate to a larger overall information space.

### 2.1 Contextualized Search Visualization

Search visualization for the global search context (vs. within individual documents) requires a structured information space to relate search results to one another and to the overall space. Kules' dissertation [23] focuses on features to use for automatic categorization of WWW search results, and also reports a study of a treemap-based overview of search results, finding them comparable to outline-form overviews (and more effective than no overview).

There are a host of 3D search visualization systems, among them LVis [4], Cat-a-Cone [17], NIRVE [35] and SPIRE/Themespaces [40]. A common theme among such 3D works is having limited or poor evaluation results, however. Themespaces is the most similar the ResultMaps, depicting an abstract document landscape with hills and valleys representing the relative strength and interrelations of various themes in the corpus. Themespaces can position individual search results as points on the 3D landscape as well.

The xFind system includes a search client with both a scatterplot and a 'thematic clustering' display called VisIslands [1]. VisIslands is a 2D tool that borrows conceptually from Themespaces, but visualizes documents matching specific search queries instead of the entire information space. Kohonen (or self-organizing) maps [22] are visually similar to treemaps, but use neural networks to cluster documents into semantically similar regions. Lin applied Kohonen maps to contextualize digital library search results in a manner that is similar to our SERP application [28]; Chen *et al.* evaluated a similar system, finding it less effective than the Yahoo! portal [7].

### 2.2 Visualization for Digital Repositories

In environments that have pre-existing metadata such as digital libraries, systems can forgo clustering algorithms. The Envision



Fig. 1. A ResultMap-enhanced search engine results page (SERP). Interactive behaviors (brushing, *et al.*) links the result listing (at left) to the ResultMap (at right) and *vice versa*.

digital library includes a visualization system that places documents in a 2D grid according to user-selectable attributes [31]. The CitiViz [20] and EtanaViz [36] systems use hyperbolic trees [25] and scatter plots to visualize library contents. The hyperbolic tree uses either pre-formed or automatically clustered hierarchies; tree node sizes correspond to the number of relevant documents within a category.

PARC researchers have used treemaps for access to personal digital stores [14], but focus more on issues with the transition from browsing to reading individual documents. Refinement via keyword queries crops irrelevant documents from the visualization. Evaluation is left to future work. Klerx, Duval and Meire implemented a treemap-based visualization interface into the Ariadne repository [21]. However, there is no search facility within their tool, no application of their visualization to search results, or evaluation.

### 2.3 Faceted Navigation

Faceted classification—classifying items into multiple independent categorization schemes (or *facets*)—is a topic from library science that has become popular in computing. Systems based on this type of structure are known as faceted navigation systems. Flamenco is one of the first frameworks of this type [41]; commercial vendors (e.g., Amazon, eBay) have also adopted facet-based navigation. Facets can be hierarchical, so our SERP ResultMaps could be considered a degenerate (i.e., one facet) case of faceted navigation.

Microsoft Research’s FacetMap [37] and follow-on FacetLens [26] systems apply treemap-inspired visualizations to general faceted data. A treemap represents each facet, which update in response to user selections. Selections refine the information space by pruning documents. The FacetZoom widget acts as a stack of 1-dimensional treemaps [12]; like the Microsoft systems, selections reveal further refinement options but filter out disregarded choices.

### 2.4 Related Work Discussion

Our survey of prior work motivates our efforts. Treemap or map-like approaches to search visualization are similar in some cases [14] [21] [22] [23] [28] [31] to ResultMaps, but those works that actually present evaluations do not address side effects on dataset knowledge or query reformulation. We also employ an engagement measure, which is not present in those studies. Search visualization works tend to rely on a separate application environment [1] [6] [20] [35] [36] [37] [40] rather than integration in the browser.

The FacetMap, FacetLens and FacetZoom works are similar conceptually, but ResultMaps differ in several ways. First, these

systems do not show the entirety of a facet hierarchy. Second, selecting a facet value removes non-matching documents from the view (losing the context of the global document space). These differences suggest ResultMap-augmented search is an interesting approach, and evaluations of their more unique features can contribute a better understanding of how visualization can support information-seeking behavior.

## 3 DESIGN AND IMPLEMENTATION

We define a ResultMap as a treemap that:

- encodes a digital repository’s full contents according to a hierarchical metadata attribute,
- accentuates certain nodes indicated by a query engine, and
- interactively links to a text listing of query responses.

We refer to the metadata attribute used in ResultMap layout as the *mapped attribute*. Encoding the entire repository (rather than, for example, encoding only items relevant to a query) provides a stable context in which to accentuate query responses. The pairing of the data visualization and a traditional text display combines a familiar paradigm for exploring query results with a visual interface to the same data; the interaction between the two reinforces the pairing and allows the user to move between textual and visual processing of the same data.

The latter transition is significant: the ordering of search results biases user selections toward the top of the list regardless of the actual quality of the results [15]: users essentially substitute the judgment of search engines for their own. As a result, users likely never see highly relevant pages or documents that are ranked low on a SERP. This is especially problematic in the restricted context of a digital library. If editorial control reduces the number of low-quality results and link structure is less helpful in determining relevance, it is more likely that useful results will occur lower in SERP rankings. ResultMaps provide an alternative representation of results in which a more salient feature—like topical classification—can call attention to relevant results that are not highly-ranked in a SERP. Many such preexisting categorizations exist as potential applications: library collections, Usenet archives, consumer products, etc.

### 3.1 Research Platform

We have developed a digital library for Human-computer Interaction (HCI) and Human-centered Computing (HCC) educational materials [9]: the HCC Education Digital Library (HCC EDL). The HCC EDL provides free, high-quality resources for teachers, students and practitioners. Moreover, it classifies its content into a hierarchical taxonomy of HCI/HCC topics, and thus is also a convenient research testbed. To that end, we have augmented the HCC EDL search engine with ResultMaps for use with our studies.

In our implementation, the metadata from our taxonomy classification is the mapped attribute. We refer to individual components of the taxonomy as *categories*. We call categories at the root of the taxonomy *top-level*. When we say a document belongs to a category, we mean that it is classified in that category or one of its descendant categories (*subcategories*). In our library, we distinguish an item’s *document type* from its *file type*: the former refers to an item’s semantic nature (e.g., lecture, image) while the latter to its encoding (e.g., PowerPoint or JPEG).

### 3.2 Design Decisions

Two of the more significant design decisions implicit in our ResultMap definition are:

- Encoding the entire repository hierarchy.
- The choice of a treemap as our visual representation.

Encoding the entire repository (rather than, for example, encoding only items that are relevant responses to a query) provides a stable context in which to accentuate query responses (cf. similar choice made by the FundExplorer system [11]). ResultMap consistency could also have beneficial ancillary effects: exposing a

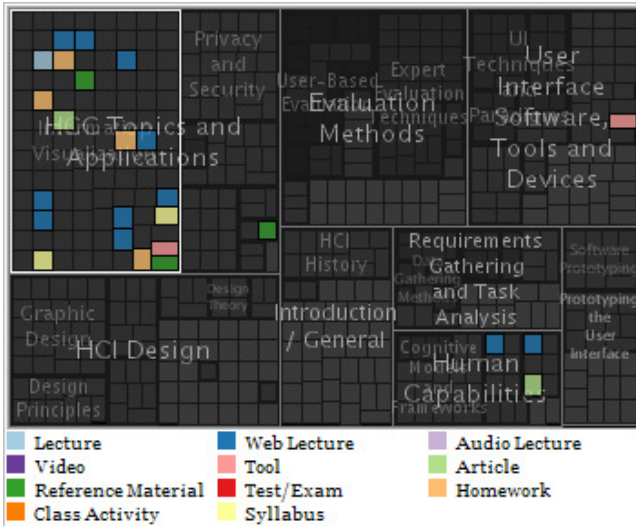


Fig. 2. Detail of a ResultMap. Color encodes document type as indicated by the legend; treemap hierarchy is by topic classification.

stable representation of the entire information space with every page view provides a means for users to gain additional knowledge about repository content as a side-effect of searching for perhaps specific documents (e.g., breadth/distribution of topical coverage). That kind of knowledge could be useful for future information-seeking tasks.

Treemaps are space-constrained and -efficient methods for representing complex hierarchies and as such suit ResultMaps, which augment text listings in a web usage environment. Constrained space usage means ResultMap pixel requirements are invariant to repository size. Other representations are not space-constrained (e.g., node-link diagrams) or space-efficient (e.g., hyperbolic trees [25]). Simpler approaches like data graphics or numeric indicators can give users a high-level indication of how a corpus is distributed over metadata values, but obscure skew at lower tree levels. Dataset features like outliers or clusters are also more apparent in a treemap rendering vs. simpler approaches. For instance, it is clear in Fig. 2 that there is an outlier at top right and a cluster near the top left.

Detecting outliers is not a spurious task. For example, when a user is having difficulty choosing keywords to retrieve appropriate documents, outliers may represent items of more interest than clusters. Outliers can also indicate data idiosyncrasies—which might be data errors (of interest to library administrators or analysts)—or simply unusual items—which might be interesting to a user precisely because they are topically distinct from most results (but potentially still related to the same query string).

### 3.2.1 Visual Design

Fig. 2 shows a ResultMap from our implementation. We use a squarified treemap layout algorithm [5]. The relative instability of the squarified algorithm is not critical, since the rendering changes only when items are added or removed from a library. When categories have inherent order (e.g., chronological groups), we use a strip layout [2]. A 1-pixel frame around category nodes distinguishes category groupings. We render labels for all top-level categories and lower-level categories as space allows without detecting label collisions. This is a naïve strategy, but graph and map labeling is a complex topic on its own and beyond the scope of this work.

Node size is unweighted in our implementation. Though this reduces the amount of information our ResultMaps convey, we have few quantitative attributes in our dataset that would be useful to encode. Search relevancy scores are one possibility, but would result in ResultMaps with unpredictable layouts and run counter to our goal of providing stable context for query result highlighting. Popularity (e.g., number of downloads) is one option; however, we explicitly do not want to emphasize more popular documents in our DL to avoid drowning out lesser-used content.

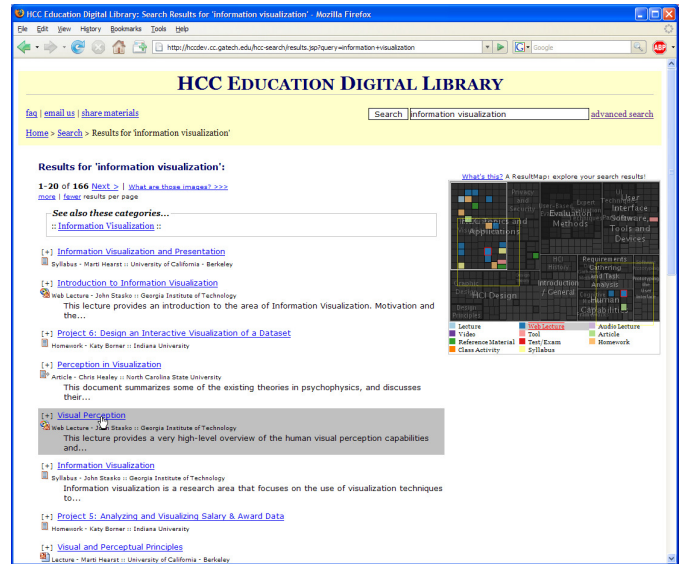


Fig. 3. The effects of brushing a result list entry: Yellow rectangles radiate from the highlighted nodes and disappear; the corresponding legend entry and ResultMap nodes are highlighted in red.

Highlighted nodes are colored and non-highlighted entries are grayscale (see Fig. 2). Brightness decreases with increasing tree depth, which makes hierarchical structure easier to interpret. Color hue indicates document type. A key<sup>1</sup> below the ResultMap shows the correspondence between our 11 document types (just within advisable limits for distinguishable nominal color schemes) and the corresponding colors. In general, a designer could use any metadata attribute to determine a color encoding; that attribute the *key attribute* (in this implementation, document type).

### 3.2.2 Interaction Design

We link the ResultMap, its legend and the search result listing via brushing: hovering over a node in the ResultMap highlights the appropriate legend entry, any other ResultMap nodes that correspond to the same document, and the appropriate entry in the search result listing. Brushing an entry in the search result list highlights the same items as well. In the implementations we tested for this work, no brushing occurs from the legend entries (but has since been added).

Highlighting entails all of the following (see Fig. 3 and Fig. 4):

- ResultMap document node: outline in red all instances of the document in the ResultMap; animate yellow boxes radiating from all document instances.
- SERP document list entry: change background to silver.
- ResultMap legend entry: change background to silver; change font color to red; add under- and over-line.

The search engine also matches whole categories, which can also be brushed within the ResultMap and list. That involves:

- Brushing from ResultMap category to SERP category list:
  - Outline in red the category in the ResultMap
  - Change SERP category list entry background to silver.
- Brushing from SERP category list to ResultMap:
  - Outline in red the ResultMap category and change its interior to white

Brushing a ResultMap node when the corresponding result listing is outside of the browser viewport shows a message indicating that the user should scroll up or down as necessary. The ResultMap is anchored to the same position within the window. Clicking a ResultMap node smoothly scrolls the viewport to the corresponding list result and expands the entry to show additional details about it.

<sup>1</sup> Generated by ColorBrewer, available at <http://www.ColorBrewer.org>



Fig. 4. The effects of brushing a ResultMap node: the corresponding result list item, key entry, and other ResultMap nodes are highlighted. This also shows the result of clicking on a ResultMap node: the corresponding result list entry expands to show additional details and scrolls the viewport to it.

### 3.2.3 Technical Details

The connection between the ResultMap and the rest of the SERP is central to its design. For that reason, our use of web standards (DHTML) to implement ResultMaps is important. Building a usable web-based visualization system means conforming to user expectations about the web [29]. The authors of the Relation Browser (RB), for example, speculated that users had minimal appreciation for its features partly because of “the layering effects of the RB over the [web interface] where the primary content is found” [6]. Browser plug-ins or standalone applications make more advanced interface features possible, but also make interaction with non-plugin portions of a web page cumbersome or impossible. Since it is precisely that interaction that is important, simplicity trumps additional capability, even without considering accessibility or search engine indexing.

We use Java Server Pages (JSPs) and the open-source Lucene search engine<sup>2</sup> with a prefuse-based [18] application-level persistent Java object to store the rendered treemap and provide formatted, relatively-positioned HTML to the JSP. We cache a background treemap image so that the JSP only requests highlighted node data, and use Cascading Style Sheets (CSS) and JavaScript to handle node presentation and interactivity. The background image sets the ResultMap size to 350x233 pixels. This static size, while enabling cached data use and its associated performance benefits, prevents us from dynamically determining and using all available vertical space.

### 3.3 Scalability Analysis

We would be negligent not to explore and quantify the representational limits of ResultMaps. Our SERP ResultMaps are 350x233 pixels with a 1 pixel frame; our library corpus as of March 2009 has 90 categories and 585 documents (including multiple categorizations). Frames consume 26.9% of the 81,550 total pixels, with an average node size of 10x10 pixels. If we allow average node size to decrease to 3x3, HCC EDL ResultMaps scale to 6,600 documents (one of the datasets in our experiments is 5,518 nodes). Clearly it is desirable to increase this threshold. The most direct way is enlargement: using 350x600 ResultMaps increases the limit to 17,500 nodes—still smaller than many real-world datasets.

A more promising approach is to relax our assumption of directly rendering documents individually. Instead, we can collectively

represent documents by bucketing leaf nodes so that one node represents multiple logical documents. That is, group leaf nodes within their parent categories by their key attribute values. This is equivalent to rendering ResultMaps that sort leaf nodes by key values and halt the layout one level above the leaf nodes. This approach introduces collisions when we need to highlight multiple nodes in the same bucket, but we can address this by highlighting all relevant nodes and list items from their shared ResultMap representation. While this precludes acting on a ResultMap surrogate identically with a document’s list representation, this tradeoff is warranted to support real-world library sizes, and we have done so in subsequent work.

The scalability of this collective approach is thus bounded by the number of categories rather than the number of documents in a collection. Corpora of any size are well supported when there are a few thousand categories (depending on ResultMap size and desired node size), suitable for use with Dewey Decimal or Library of Congress classifications. This method also has the advantage of permitting a graceful transition from individual to collective representation: the position or size of the category nodes does not change.

## 4 EVALUATION

Evaluation of infovis applications is notoriously tricky [33]. Many systems are oriented towards browsing or exploring data, so strictly performance-based evaluations (e.g., task time) are frequently less relevant than other measures. Additionally, testing is made more difficult by the fact that data exploration is inherently an open-ended activity, making it difficult to achieve experimental control without artificially constraining tasks in an investigational setting.

Despite these complications, our research approach nonetheless has a strong empirical testing bent. Our studies have the objective of quantifying and qualifying the effects of ResultMaps on users’ information seeking behavior. In addition to objective metrics like task time and accuracy, we examine more subjective factors such as user preference and engagement. Engagement is one aspect of interface flow [13], which is usually described in terms of absorption and pleasure while executing a task. Tests of and hypotheses about such affective measures for visualization and information-seeking systems are well-represented in the literature (e.g., [6] [23] [41]).

Like most projects, ResultMap development has been iterative in nature along with their evaluation. Here, we discuss the two studies: the first was summative in the sense that we wanted to assess ResultMaps as they existed at the time and formative in the sense that the results affected the later iteration we tested in Study 2.

### 4.1 Study 1 (Summative/Formative)

We have supposed that ResultMaps assist with detecting outlying results and that rendering the entire repository is beneficial because of consistency. One way in which that benefit might be manifest is making users more aware of the full scope of the repository, and in turn giving users a more accurate model of the entire library. This study examines these questions: are there any performance benefits for particular types of search results (i.e., outliers, clustered, etc.) and do ResultMap users get a better understanding of a digital library’s characteristics vs. a text-only search interface?

ResultMap nodes can be outliers by their color (i.e., document type) or position (i.e., topic classification). We define a *color outlier* as a node with a unique document type (and thus color) within a SERP. We classify a node  $h$  as a *position outlier* when there are few other nodes in  $h$ ’s top-level category and many in others. Conversely,  $h$  is in a *position cluster* when most of the highlighted nodes are within  $h$ ’s own top-level category. A deterministic process makes this determination and is detailed elsewhere for brevity [8].

#### 4.1.1 Design, Equipment and Procedure

We use a single-factor between-subjects design with two levels: ResultMap (RM) vs. non-ResultMap (control). The control interface simply removes the RM widget. The dependent measures include

<sup>2</sup> <http://lucene.apache.org/java/docs/>

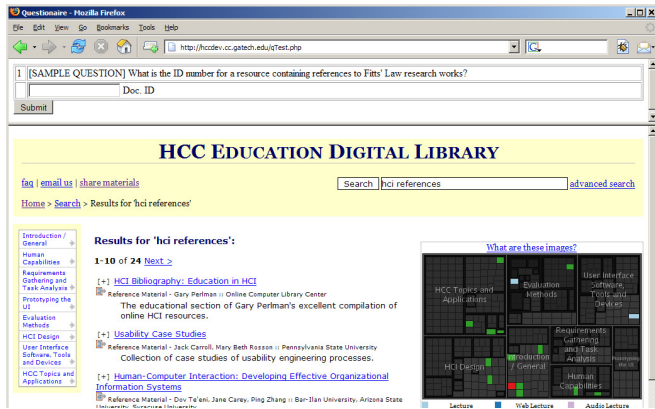


Fig. 5. The test environment for Studies 1 and 2. The question/answer area is in a frame above the search interface below.

task completion time, accuracy, and post-tests of users' knowledge of and subjective attitudes about the DL environment. The RM condition is the search engine interface in use with the HCC EDL, but was an older version than that described in Section 3, which did not brush elements in the ResultMap from the result listings and lacked any animation effects. It also only labeled the top-level categories. The HCC EDL repository is relatively small, but represents a useful proof-of-concept dataset. We make several hypotheses about our treatment effects:

- The RM group will have faster and more accurate task performance on outlier detection tasks.
- The RM group will have higher self-reported satisfaction scores.
- The RM group will have higher scores on repository knowledge assessments.

Subjects used a Windows XP desktop workstation in the GVU Center usability lab. The workstation was connected to a 1280x1024 pixel LCD monitor and ran a maximized instance of the Mozilla Firefox web browser. Experimental text appeared in a 150-pixel tall frame above the search interface (see Fig. 5). Experimental tasks had users identify a target document from search results based on criteria specified in the experimental text. To control for search expertise, we used the same query terms for each task, and showed those queries (along with a motivating scenario) to users prior to the task itself. We used search strings from the production HCC EDL logs as a basis for the query terms/scenarios. For example: *The following question relates to the search results for the query 'hci references'. You might search for this if you were looking for a bibliographic listing of HCI works in the repository.*

The system then revealed a SERP (10 items per page) from the HCC EDL and asked subjects to identify a target document satisfying a set of conditions, such as “[w]hat is the ID number for a resource containing references to Fitts’ Law research works?” All target documents were present on the first page of the search results, although we did not inform subjects of that fact. We presented 13 search result listings—one practice item followed by 12 randomly-ordered tasks. Of the 12 targets, 5 were color outliers, 3 were position outliers and 4 were in position clusters. After subjects finished the tasks, they completed a set of questions testing their knowledge about characteristics of the repository as a whole and about their subjective impressions of the system. There was no advance notice of the repository knowledge questionnaire, so participants were not primed to search for that information during their primary tasks (i.e., the experimental tasks served as distractors for the purposes of this survey). The repository characteristics included size, number of categories in the classification, frequency of document types, recognition of top-level categories and relative size of top-level categories. This kind of knowledge is beneficial because users who are more aware of the breadth and depth of a repository can apply that knowledge in later information-seeking tasks.

## 4.1.2 Results

We completed Study 1 using 20 participants (10 male) from the graduate student population at Georgia Tech, ages 22-34. All were in HCI-related degree programs with at least a year of HCI experience, which we required because of the nature of the HCC EDL materials. Most (75%) were familiar with treemaps, but comparisons between familiar and non-familiar groups (including within the RM group) were not significant. Participants completed tasks quickly and accurately: a mean of 34s per task ( $\sigma=13s$ ) with 90% accuracy. We performed a series of t-tests and a Mann-Whitney non-parametric test on accuracy comparing the RM and control groups:

- There were no significant differences in task time or accuracy between groups. The RM group had marginally higher accuracy on tasks when the target was an outlier: 93% vs. 80% ( $p=0.07$ ).
- RM users rated the impact of the interface on task difficulty more favorably than the control: means of 2.8 vs. 3.9 on 7-point Likert scale with 1 being easier ( $t=2.31$ ;  $df=18$ ;  $p<0.05$ ).
- RM users scored better on portions of the repository knowledge post-test<sup>3</sup>: means of 3.0 vs. 0.2 ( $t=-3.38$ ;  $df=18$ ;  $p<0.01$ ).

We draw mixed conclusions from the data. On one hand, we found only marginal support for our hypothesis that SERP ResultMaps improve performance on outlier tasks, and there were fewer survey differences between the groups than we would have hoped. On the other hand, the significant differences we did find all favored ResultMaps and outlier performance trended in favor of ResultMaps. Those differences give some credence to our other two hypotheses that users subjectively prefer and understand the gist of a digital library better after using ResultMaps. ResultMaps also perform no worse than a control in all cases, indicating that designers can use ResultMap-augmented engines without negative effects.

## 4.2 Study 2 (Summative)

The complexity of the Study 1 tasks were relatively simple (cf. 34s mean task time), and as such may have been too straightforward to warrant ResultMap use. Study 2 uses more complex and open-ended tasks to generate sequences of queries, following recent infovis trends such as in insight-based evaluation [30]. We posit that ResultMap usage can affect users' query sequences by making clusters and outliers more apparent; that detection in turn may feed into future queries. The precise nature of any changes is unclear: are users more or less likely to abandon a query sequence or to start a new one? Are there qualitative differences between queries?

Our hypotheses concerning subjective ratings and repository knowledge had better support than task performance, so we add an engagement measure previously used in exploratory search studies [6] and a validated usability survey [27]. We also test ResultMaps against a larger dataset, adding a second experimental factor.

### 4.2.1 Design, Equipment and Procedure

We use a split-plot design with interface type (RM vs. control) as the between-subjects factor and repository size (small vs. large) as the within-subjects factor. Dependent measures are task time/accuracy, number of query strings, average query length, query string change between successive queries, and our surveys of repository knowledge, subjective ratings and engagement. We hypothesize:

- The RM group will have task times no worse than the control.
- The RM group will have higher self-reported satisfaction scores.
- The RM group will have higher scores on an assessment of repository knowledge.
- The RM group will have higher self-reported engagement/enjoyment scores.
- There will be differences in the query string characteristics (length, number, etc.) between the two conditions.

<sup>3</sup> Users identified topic categories present in the library from a list of 6 possibilities, which had 3 valid and 3 invalid choices. We awarded 1 point for a correct and -1 point for an incorrect decision for a scoring range of [-6,6] and circling all, none or random choices would result in a score of 0.

Table 1. Categorized statistical analyses for Study 2.

Behavioral Objective	Mixed repeated measures/between-subjects univariate ANOVAs on task completion times and scores/accuracy. Mixed repeated measures/between-subjects factor ANOVAs on number of query strings per task, mean query string length, and mean change between successive query strings (as measured by Levenshtein minimum string distance [38]).
Surveyed Objective	Two-factor ANOVAs on the repository knowledge indicators.
Surveyed Subjective	Mixed repeated measures/between-subjects factor ANOVAs on CSUQ, enjoyment, and engagement aggregate measures.
	Mixed repeated measures/between-subjects factor ANOVAs on satisfaction indicators.

For the small repository condition, we used the HCC EDL; for the large condition we used a portion of the Intute digital library, a collection of web resources for education and research. We extracted the ‘Computing’ portion for use as our large dataset<sup>4</sup>. The HCC EDL consisted of 583 items (487 unique) classified into a 90-node hierarchy. The Intute data had 5,518 items (2,602 unique) in 224 categories, and had over 30 document types. To reduce those to fewer than a dozen (for a realistic color palette), we combined similar types into broader labels. The physical equipment was the same as in Study 1. There were a few differences in the software:

- The ResultMaps were a more advanced implementation (containing all features described in Section 3).
- ResultMaps in the large condition added a white border to highlighted nodes to improve perceptibility.
- The SERP had 20 items per page by default, and users could add or decrease that number using links at the top of the item listing.

We created 6 tasks (one practice, 5 test) for each repository ranging in specificity from locating a specific item (*You are looking for modeling and design software for use in engineering applications. What is the name of a company that supplies such software?*—Intute dataset) to open-ended directives (*Find an interesting homework assignment that you would assign if you were an instructor for an undergraduate HCI survey course. Identify its title and author.*—HCC EDL dataset). As with the first study, we used search strings as a basis for creating the HCC EDL tasks. We did not have similar data for the Intute library, so we brainstormed to create *ad hoc* analogous tasks. The tasks within each repository appeared in the same order; the order of the repository conditions (small and large) alternated. As in the first study, the system presented the task in a browser frame above the search interface. The system prompted users to finish after 5 minutes and forced them to move on after 7.

Participants completed two sets of tasks, one for each repository size (the order of which was counterbalanced). Before the first condition, participants read a short (300 word) description of the experimental terminology (*category, document, etc.*) and a summary of the type of documents within the HCC EDL and Intute repositories. All participants then watched a 2-minute, narrated screen-captured video about the repository search engines and their basic features (relevance ordering, advanced search features, etc). To control for prior familiarity with treemaps, users in the RM condition watched another 3-minute video summarizing the ResultMap features and its interaction with the rest of the SERP.

After the first condition, subjects completed the surveys from Study 1, the IBM Computer System Usability Questionnaire (CSUQ) [27], and the task enjoyment/engagement instrument mentioned above. All surveys used 7-point Likert scale responses. The repository knowledge questionnaires for each size condition were identical apart from using appropriate category names for the

respective datasets. We did not use the repository knowledge survey after the second condition (which was otherwise the same) since repeat deployment could skew users towards learning about the repository features relevant to the survey. We also used a survey and informal interview to debrief RM participants about their opinions.

#### 4.2.2 Results

We completed Study 2 using volunteer students from Georgia Tech enrolled in introductory undergraduate HCI and psychology courses. The instructors of the classes offered extra credit to students participating in research studies. Forty-one participants completed the experiment. Five subjects yielded spurious data because of incomplete surveys, tasks, or software errors, resulting in a final N=36 (15 female), 19 in the RM group and 17 in the control. Three members of the RM group had prior familiarity with the treemaps. We graded the task answers on a scale of 0-2 with 0 being completely wrong and 2 being correct. Two of us scored 4 participants independently, according to our ordinal grading guidelines; Spearman’s rho indicated strong agreement at 0.878. We report the answer grades below on a 0-100% scale, treating 1 as 50% correct.

We performed a series of univariate analyses of variance (ANOVAs) on our dependent measures, which we categorize in Table 1 by how the data was collected (behavior records vs. questionnaires/surveys) and the kind of the data collected (subjective vs. objective). Because participants only completed one repository knowledge assessment, we used repository size as a between-subjects factor for that ANOVA.

#### Behavioral Objective Measures

Table 2 shows that RM users had a faster mean task time and a higher accuracy rate than the control group: 12% faster on the small condition and 8% faster on the large, though the main effect of interface (RM vs. control) was not statistically significant for either measure ( $0.11 \leq p \leq 0.38$ ). Repository size had significant main effect on time ( $F=27.503$ ;  $df=1$ ;  $p<0.01$ ): participants took more time on the larger repository. There were no other statistically significant effects on time or accuracy.

Likewise, there were no differences in query characteristics (number, length or change) other than due to repository (the larger repository resulted in more, longer, and more varied search queries, all  $p<0.05$ ). In fact, we were surprised at exactly how little interface mattered: within each repository, variation of mean query string length and mean change between successive strings was 2% or less.

#### Surveyed Objective Measures

The results for our surveyed objective measures—i.e., our survey assessing knowledge about characteristics of each repository—were similar to our behavioral objective measures: RM users generally provided more accurate assessments of repository characteristics, but no results were statistically significant ( $0.06 \leq p \leq 0.23$ ).

#### Surveyed Subjective Measures

On our subjective surveys, RM users consistently gave higher scores than the control group, but only significantly so on the enjoyment. RM users enjoyed their usage significantly more than the control group ( $F=8.24$ ,  $df=1$ ,  $p<0.01$ ). On the satisfaction survey (also given in Study 1), RM users rated their interface design as making the tasks less difficult and take less time than a standard SERP, but these results were again not significant ( $0.09 \leq p \leq 0.12$ ).

Table 2. Task times (in s) and accuracy by interface and repository.

	ResultMap (n=19)	Control (n=17)
Small (587 nodes)	164s, $\sigma=59s$ 70.5%, $\sigma=14.4\%$	183s, $\sigma=71s$ 59.4%, $\sigma=23.8\%$
Large (5,518 nodes)	198s, $\sigma=49s$ 72.3%, $\sigma=17.8\%$	213s, $\sigma=66s$ 66.5%, $\sigma=24.5\%$

<sup>4</sup> <http://www.intute.ac.uk/sciences/computing/>

Table 3. Measures (7-point Likert) of usability (CSUQ), enjoyment (ENJ) and engagement (ENG) by interface and repository. Effect of interface on ENJ is significant.

	ResultMap (n=19)	Control (n=17)
Small (587 nodes)	CSUQ: 5.17, $\sigma=0.69$ ENJ: 5.08, $\sigma=1.10$ ENG: 5.33, $\sigma=0.88$	CSUQ: 4.84, $\sigma=0.83$ ENJ: 4.22, $\sigma=0.97$ ENG: 5.00, $\sigma=0.95$
Large (5,518 nodes)	CSUQ: 4.61, $\sigma=0.94$ ENJ: 4.91, $\sigma=1.04$ ENG: 5.28, $\sigma=1.09$	CSUQ: 4.44, $\sigma=0.79$ ENJ: 3.94, $\sigma=0.90$ ENG: 5.02, $\sigma=0.77$

The results of our debriefing survey and interviews reflected an overall appreciation for ResultMaps. The utility of the category grouping and color-coding ResultMap functions were both highly rated (6.11,  $\sigma=1.05$  and 5.63,  $\sigma=1.26$  respectively, 7-point Likert scale), and had a mean estimation of using ResultMaps on 65% of the tasks and being helpful 61% of the time.

Free-form comments reflected those statistics and centered on their general utility (“The search was awesome”; “The visualization tool was useful”) or the functions we identified (“Visual representation of materials in the library is useful”; “It is very easy to see how results are divided into their document types on the map.”).

Suggestions for improvement had several themes: size and legibility; animation; and more brushing. Size and legibility refer the general visualization and labels specifically. Our (purposeful) attention-grabbing brushing animations were divisive: some had an active dislike and some an active preference for the animations. Finally, the inability to highlight all items of a document type from the ResultMap key was a common concern.

### 4.3 Study Limitations

Beyond the results themselves, we should note some broader limitations of these studies. The relatively small sizes of the study datasets limit our ability to make definitive scalability findings. However, we have subsequently deployed ResultMaps on 16,000+ item datasets, lending some practical weight to our Section 3.3 analysis. More significant is the issue of user training. In Study 1, users received no significant training, but Study 2 included 3 training elements: a 300 word written overview, a 2 minute search engine overview video and a 3-minute ResultMap features video. The former two are not notable: the written materials pertained to experimental factors that are not relevant when users generate their own search goals, and the 2-minute video covered basic search engine features (relevance ordering, etc.). No users expressed any indication of unfamiliarity with the content of that video.

The latter 3-minute video is more relevant to study generalization, since no such training occurs in general web usage. As a result, we cannot claim Study 2 applies directly to novice web users. However, the video content (the existence of the ResultMap, interactive behavior, etc.) is discoverable in the interface without great effort. This claim is bolstered by Study 1, in which users received no RM training at all, but nonetheless used and benefited from their addition.

Table 4. Marginally significant results from Studies 1 and 2.

Study	Measure	P-value
1	Outlier Task Accuracy	0.07
2	Task Accuracy	0.11
2	Repository Knowledge (Categories)	0.06
2	Design Impact on Difficulty	0.12
2	Design Impact on Time	0.09

## 5 CONCLUSIONS AND FUTURE WORK

Like many previous works, the most common theme from our statistical results is ‘no significant difference’. However, every

significant result we found favors ResultMaps: positive impact of interface on self-reported task difficulty, repository understanding and self-reported enjoyment. Moreover, there were enough near-significant results (see Table 4) that we suspect our statistical power was not sufficient to detect ResultMaps’ effect size.

Looking back at our combined list of hypotheses, we find no definitive evidence in favor of any of them, but some mild support for the hypotheses that ResultMaps are subjectively preferred (Study 2) and increase knowledge of repository characteristics (Study 1 and Study 2). The only hypothesis that seems to have strong evidence against it is that about ResultMaps affecting query formulation. The extent to which those measures were identical was striking, and to us a somewhat interesting result.

Especially given our results, the lack of statistical power inherent to our between-subjects designs was among the most dissatisfying aspects of our evaluations. However, we are faced with a conundrum: it is difficult to separate dataset tasks from the datasets they operate on, and such tasks cannot be repeated because of practice effects. Consequently, within-subjects comparisons of interface alternatives require tasks that are different in their particulars but somehow matched. But that matching process is vague (with some recent efforts to change that [24])—most works simply state that such tasks sets are matched along broad guidelines (e.g., lookup, complex or exploratory tasks; cf. [6] [41]) but no rigorous methodology beyond intuitive comparison. Furthermore, experimental tasks are difficult to design well in the first place, especially without bias towards a particular interface—and this process requires two such sets. We faced this difficulty in our second experiment, in which it was difficult to generate complex naturalistic tasks that probed ResultMaps’ emphasis on outlier and cluster detection without biasing the question toward readily apparent data in the ResultMap.

Insight-based protocols [30] are an alternative to structured tasks, but seem less-well suited to DL search applications, and suffer even more from practice effects (since insight accumulates over time). Nonetheless, insight-based methods show considerable promise, but as North notes, present considerable resource challenges. Moreover, the nature of insight itself—relatively rare, unpredictable and qualitative—makes it difficult to measure in lab studies. But evaluating insights also benefits from precision and detail about its circumstances, making it difficult to assess in longitudinal studies: diary, interviews, or other self-reported methods can be unreliable and not provide the sufficient details about insight.

On a practical level, we have come to distinguish two aspects of users’ interest in and motivation towards a dataset: *analytic* or *meta-interest* and *direct* interest in the data itself. DL users, for example, are most often interested in the content of the DL documents rather than their metadata. In contrast, users such as intelligence analysts may be just as (if not moreso) interested in metadata and its distribution among intelligence or police reports. Our evaluations were hampered by the fact that users—even when they did engage with the library data—had more direct than analytic interest, while many of the ResultMap benefits are more analytic. One implication is that future ResultMap evaluations might target DL users with more analytic interests: library curators, for example.

Going forward, we make several suggestions for work in this area. With respect to matching dataset tasks, empirical measures of what constitutes task isomorphism would be helpful, such as:

- Measures of similarity (via one of many metrics [3]) between the local tree structures around the target documents.
- Measures of similarity between target items (e.g., number of identical attributes and similarity of relative distribution of differing attribute values).

For complex tasks that yield specific (or sets of) items, a *post hoc* analysis over all results could indicate some indication of congruence. These kinds of measures ignore task semantics, but identifying high-level data similarities and crafting tasks around them might prove easier than trial-and-error task creation based only on intuitive notions of similarity.

With respect to insight capture, lightweight mechanisms of reporting or recording insight data are critical, especially in longitudinal studies. The exact nature of such features depend on circumstances, but simple data annotation tools to mark data views (which can trigger additional logging features) is one possibility.

We have presented the ResultMap concept, a treemap-based system for enhancing query-based search engines, and two controlled lab studies of their use. We found some evidence users subjectively prefer them and yield comparable performance to a text-only engine. We discussed our evaluation approach and suggested ways to improve similar studies based on our experience. Our current work focuses on applying and evaluating ResultMaps in faceted metadata contexts, with the hypothesis that ResultMaps allow users to better preview, diagnose and link relationships between facets, and is more suited to exploratory and insight-based evaluation.

## ACKNOWLEDGEMENTS

This work was supported by the Stephen Fleming Chair in Telecommunications and the National Visualization and Analytics Center (NVAC™), a U.S. Department of Homeland Security Program, under the auspices of the Southeast Regional Visualization and Analytics Center.

## REFERENCES

- [1] Andrews, K., Gutl, C., Moser, J., Sabol, V. and Lackner, W. Search Result Visualization with xFIND. In *Proc. of IEEE Workshop on User Interfaces to Data Intensive Systems '01*, pp. 50-58.
- [2] Bederson, B., Shneiderman, B., and Wattenberg, M. Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies. *ACM Trans. Graph.* 21(4), 2002, pp. 833-854.
- [3] Bille, P. A Survey on Tree Edit Distance and Related Problems. *Theor. Comput. Sci.* 337(1-3), pp. 217-239.
- [4] Börner, K., Dillon, A. and Dolinsky, M. LVis - Digital Library Visualizer. In *Proc. of IEEE International Conference on Information Visualization '00*, pp. 77-81.
- [5] Bruls, M., Huizing, K. and van Wijk, J. Squarified treemaps. In *Proc. of Eurographics/IEEE TCVG Symposium on Visualization '00*, pp. 33-42.
- [6] Capra, R., Marchionini, G., Oh, J. S., Stutzman, F., and Zhang, Y. Effects of Structure and Interaction Style on Distinct Search Tasks. In *Proc. of ACM/IEEE JCDL '07*, pp. 442-451.
- [7] Chen, H., Houston, A., Sewell, R., and Schatz, B. Internet Browsing and Searching: User Evaluations of Category Map and Concept Space Techniques. *Journal of the Am. Soc. for Inform. Sci.* 49(7), pp. 582-603.
- [8] Clarkson, E. *Visual Search Interfaces for Online Digital Repositories*. Ph.D. Dissertation, Georgia Institute of Technology, 2009.
- [9] Clarkson, E., Day, J. and Foley, J. The Development of an Educational Digital Library for Human-Centered Computing. GVU T.R. GIT-GVU-03-33. Available at [http://gvu.cc.gatech.edu/research/tr/tr05\\_33.html](http://gvu.cc.gatech.edu/research/tr/tr05_33.html)
- [10] Clarkson, E., Navathe, S., and Foley, J. Generalized Formal Models for Faceted User Interfaces. In *Proc. of JCDL '09*, pp. 125-134.
- [11] Csallner, C., Handte, M., Lehmann, O and Stasko, J. FundExplorer: Supporting the Diversification of Mutual Fund Portfolios Using Context Treemaps. In *Proc. of IEEE Information Visualization '03*, pp. 203-208.
- [12] Dachsel, R., Frisch, M., and Weiland, M. FacetZoom: A Continuous Multi-Scale Widget for Navigating Hierarchical Metadata. In *Proc. of ACM CHI '08*, pp. 1353-1356.
- [13] Ghani, J., Supnick, R. and Rooney, P. The Experience of Flow in Computer-mediated and in Face-to-face Groups. In *Proc. of the International Conference on Information Systems '91*, pp 229-237.
- [14] Good, L., Popat, A., Janssen, W. and Bier, E. A Fluid Treemap Interface for Personal Digital Libraries. In *Proc. of ECDL '05*, pp. 162-173.
- [15] Guan, Z. and Cutrell, E. An Eye-tracking Study of the Effect of Target Rank on Web Search. In *Proc. of ACM CHI '07*, pp. 417-420.
- [16] Hearst, M. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *Proc. of ACM CHI '95*, pp. 59-66.
- [17] Hearst, M. and Karadi, C. Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy. In *Proc. of ACM IR '97*, pp. 246-255.
- [18] Heer, J., Card, S. and Landay, J. prefuse: a Toolkit for Interactive Information Visualization. In *Proc. of ACM CHI '05*, pp. 421-430.
- [19] Johnson, B. and Shneiderman, B. Tree-maps: A space filling approach to the visualization of hierarchical information structures. In *Proc. of IEEE Visualization '91*, pp. 284-291.
- [20] Kampanya, N., Shen, R., Kim, S., North, C., Fox, E. Civiz: A Visual User Interface to the CITIDEL System. In *Proc. ECDL '04*, pp. 122-133.
- [21] Klerkx, J., Duval, E. and Meire, M. Using Information Visualization for Accessing Learning Object Repositories. In *Proc. of IEEE International Conference on Information Visualization '04*, pp. 465-470.
- [22] Kohonen, T. The Self-organizing Map. *Proc. of the IEEE* 78(9), pp. 1464-1480.
- [23] Kules, B. *Supporting Exploratory Web Search with Meaningful and Stable Categorized Overviews*. Ph.D. Dissertation, Univ. of Md., 2006.
- [24] Kules, B. and Capra, R. Constructing Exploratory Tasks for a Faceted Search Interface. In *Proceedings of the Human-Computer Interaction and Information Retrieval Workshop*, pp. 18-21.
- [25] Lamping, J. and Rao, R. A focus+context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proc. of ACM CHI '95*, pp. 401-408.
- [26] Lee, B., Smith, G., Robertson, G., Czerwinski, M., and Tan, D. FacetLens: Exposing Trends and Relationships to Support Sensemaking within Faceted Datasets. In *Proc. of ACM CHI '09*, pp. 1293-1302.
- [27] Lewis, J. IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *International Journal of Human-Computer Interaction*, 7(1), pp. 57-78.
- [28] Lin, X. Map Displays for Information Retrieval. *Journal of the Amer. Soc. for Inform. Sci.* 48(1), pp. 40-54.
- [29] Nielsen, J. "Mental Models For Search Are Getting Firmer." *Jakob Nielsen's Alertbox*, Internet column, May 9, 2005. Available at <http://www.useit.com/alertbox/20050509.html>
- [30] North, C. Toward Measuring Visualization Insight, *IEEE Comput. Graph. Appl.* 26(3), pp. 6-9.
- [31] Nowell, L., France, R., Hix, D., Heath, L. and Fox, E. Visualizing Search Results: Some Alternatives to Query-Document Similarity. In *Proc. of ACM Information Retrieval '96*, pp. 67-75.
- [32] Olsen, K., Korfhage, R., Sochats, K., Spring, M. and Williams, J. Visualization of a Document Collection: the VIBE System. *Inf. Process Manage.* 29(1), pp. 69-81.
- [33] Plaisant, C. The challenge of information visualization evaluation. In *Proc. of ACM Advanced Visual Interfaces '04*, pp. 109-116.
- [34] Rohrer, R. and Swing, E. Web-based information visualization. *IEEE Computer Graphics and Applications* 17(4), pp. 52-59.
- [35] Sebrecchts, M., Vasilakis, J., Miller, M., Cugini, J. and Laskowski, S. Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces. In *Proc. of ACM IR '99*, pp. 3-10.
- [36] Shen, R., Vemuri, N., Fan, W., da S. Torres, R. and Fox, E. A. Exploring digital libraries: integrating browsing, searching, and visualization. In *Proc. of IEEE/ACM JCDL '06*, pp. 1-10.
- [37] Smith, G., Czerwinski, M., Meyers, B., Robbins, D., Robertson, G. and Tan, D. FacetMap: a Scalable Search and Browse Visualization. In *Proc. of IEEE Information Visualization '06*, pp. 797-804.
- [38] Soukoreff, R., and MacKenzie, I. S. Measuring errors in text entry tasks: An application of the Levenshtein string distance statistic. In *ACM CHI '01 Extended Abstracts*, pp. 319-320.
- [39] Veerasamy, A. and Belkin, N. Evaluation of a Tool for Visualization of Information Retrieval Results. In *Proc. of ACM IR '96*, pp. 85-92.
- [40] Wise, J., Thomas, J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., Crow, V. Visualizing the Non-visual: Spatial Analysis and Interaction with Information from Text Documents. In *Proc. of IEEE Information Visualization '95*, pp. 51-58.
- [41] Yee, K., Swearingen, K., Li, K. and Hearst, M. Faceted Metadata for Image Search and Browsing. In *Proc. of ACM CHI '03*, pp. 401-408.