A Case for Query by Image and Text Content: Searching Computer Help using Screenshots and Keywords

Tom Yeh[§] Brandyn White[§] Jose San Pedro[†] Boris Katz[‡] Larry S. Davis[§]

[§]University of Maryland
College Park, Maryland, USA
{tomyeh,brandyn,lsd}@umd.edu

[†]TelefonicaR&D
Via Augusta, 171
Barcelona 08021, Spain
jsanpedro@mac.com

[‡]MIT CSAIL
32 Vassar St.
Cambridge, MA, USA
boris@mit.edu

ABSTRACT

The multimedia information retrieval community has dedicated extensive research effort to the problem of content-based image retrieval (CBIR). However, these systems find their main limitation in the difficulty of creating pictorial queries. As a result, few systems offer the option of querying by visual examples, and rely on automatic concept detection and tagging techniques to provide support for searching visual content using textual queries.

This paper proposes and studies a practical multimodal web search scenario, where CBIR fits intuitively to improve the retrieval of rich information queries. Many online articles contain useful know-how knowledge about computer applications. These articles tend to be richly illustrated by screenshots. We present a system to search for such software know-how articles that leverages the visual correspondences between screenshots. Users can naturally create pictorial queries simply by taking a screenshot of the application to retrieve a list of articles containing a matching screenshot.

We build a prototype comprising 150k articles that are classified into walkthrough, book, gallery, and general categories, and provide a comprehensive evaluation of this system, focusing on technical (accuracy of CBIR techniques) and usability (perceived system usefulness) aspects. We also consider the study of added value features of such a visual-supported search, including the ability to perform cross-lingual queries. We find that the system is able to retrieve matching screenshots for a wide variety of programs, across language boundaries, and provide subjectively more useful results than keyword-based web and image search engines.

Categories and Subject Descriptors

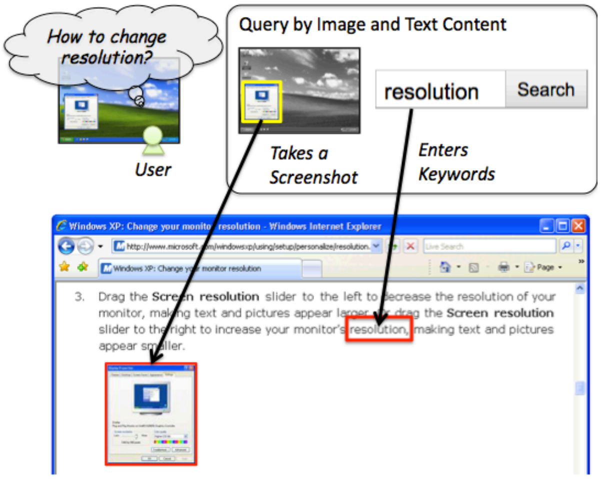
H.2.8 [**Database Applications**]: Image Databases; I.4 [**Computing Methodologies**]: Image Processing and Computer Vision

General Terms

Human Factors, Measurement

Keywords

Content-based Image Retrieval, Multimodal Search, Online Help


How to change resolution?
User
Query by Image and Text Content
resolution
Search
Takes a Screenshot
Enters Keywords
Windows XP: Change your monitor resolution - Windows Internet Explorer
http://www.microsoft.com/windowsxp/using/setup/personalize/resolution.
Live Search
Windows XP: Change your monitor resolution
Page

3. Drag the **Screen resolution** slider to the left to decrease the resolution of your monitor, making text and pictures appear larger, or drag the **Screen resolution** slider to the right to increase your monitor's resolution, making text and pictures appear smaller.

Figure 1. A practical scenario of query by image and text content. Users who want to know how to perform a task (e.g., how to change the resolution?) using a GUI application can use the application's screenshot and keywords to retrieve articles visually and textually relevant to the task.

1. INTRODUCTION

Many online articles contain useful know-how knowledge about popular GUI-based computer applications. Users can learn how to perform a wide variety of tasks, such as how to set up a home network, back up files, or change the speed of the mouse cursor. These resources are often created and made available online by software developers who want to provide users with a current up-to-date version of the documentation (e.g., support.microsoft.com). These resources are also created by unofficial, third-party experts, for example, by sites offering tutorials and tips on various software applications (e.g., osxfaq.com), by general-purpose *how-to* sites featuring software tutorials as one of their topics (e.g., eHow.com), and by computer book publishers who wish to make their books accessible online via subscription (e.g., safaribooksonline.com). Even more resources can be found in the form of user-generated content, for example, in blogs where users share their experiences and tips using a software application, in discussion boards where people discuss and learn from each other about software, and in QA communities such as Yahoo Answers where members can raise questions and get answers back from other members.

To search these rich resources of online computer help, most Internet users currently rely on conventional text-based search engines. They pose keyword queries and hope to retrieve content relevant to the GUI application they are using and the tasks they are performing. For example, suppose a user wishes to change the IP address and opens up the network properties dialog window. Then, suppose the user encounters some difficulties and needs to get help. Ideally, the help should be related to both the context (i.e., the dialog) and the task (i.e., changing the IP address). Using

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

a conventional search engine, this user may first type “change IP address” as keywords and retrieve a plethora of results. Many of these results may be about the wrong operating systems or dialog windows. The user may realize the search terms are too general and try to refine the original query by entering additional search terms to describe the operating system, the title of the dialog window, and any other identifying information about the dialog window in order to specify the application context correctly.

The keyword-based search process described above, however, can be especially challenging for inexperienced searchers. Studies have shown that keyword-based queries significantly limit the expressiveness of users and, therefore, degrade the effectiveness of search [25]. In the case for searching computer help, using only keywords to adequately express both the application context and the task can be problematic. Furthermore, this problem is compounded by the inadequate internationalization of many GUI applications. This inadequacy often forces users to formulate queries using a language they are not native in [4]. As a consequence, it may take users a considerable amount of time and effort to discover the right set of keywords through a trial-and-error process.

Given the highly visual nature of GUI applications, one way to overcome the limitation of keyword queries is to allow users to use screenshots of the target interface as visual queries. In our previous work [35], we demonstrated that GUI screenshots are effective queries for retrieving visually relevant documents such as the pages in an electronic book that contain image figures of the target GUI application. Technically, querying by screenshots is a type of content-based image retrieval (CBIR) problem. In unconstrained scenarios, CBIR faces a number of challenges including the sensory gap (difference between how users perceive and describe a target and how the target is internally represented in the search system) and the semantic gap (difficulties in extracting accurate semantic information from images) [30]. Therefore, most real-world image retrieval systems rely on keyword queries (e.g., Google, Bing Image Search). Some systems do analyze image content to automatically generate text annotations; but the purpose is to support text-based search (e.g., automatic tagging [21], concept detection [31]). A few pure CBIR systems do exist; but they often focus on specialized domains where these gaps are easier to overcome. Examples of these domains include images of product packaging, buildings, and paintings [8]. In our previous work [35], we identified GUI screenshots as another promising opportunity for building practical search systems because high retrieval performance is attainable. However, we discovered while a query-by-screenshot system can retrieve documents related to the right interface, the textual content of the retrieved documents may not necessarily relate to the tasks users need help on.

In this paper, we present a practical case where multimodal search (image + text) is advantageous—searching computer help. Online help documents about GUIs tend to be richly illustrated by screenshots. We propose a system that leverages this unique property. Image queries in the form of screenshots ensure the visual relevance to the target interface, whereas textual queries in the form of keywords ensure the textual relevance to the pertinent computing tasks. Figure 1 provides an example how a user can practically retrieve a Web page visually and textually relevant to the interface and the task.

Our proposed system can potentially provide the following benefits: It offers users a faster and more intuitive method to describe an interface by capturing a single screenshot rather than thinking of many keywords. It allows for unambiguous separation between context and topic by delegating each query modality to a well-defined role (i.e., screenshot→context, keywords→topic). It simplifies the judging of the relevancy of the search results because users can quickly glance the matched figures and determine whether the figures are relevant to the interface, instead of based on matched keywords as in the case of keyword-only search. Finally, it offers a language independent way to by speaking the universal language of images and does not require users to translate technical terms such as interface titles.

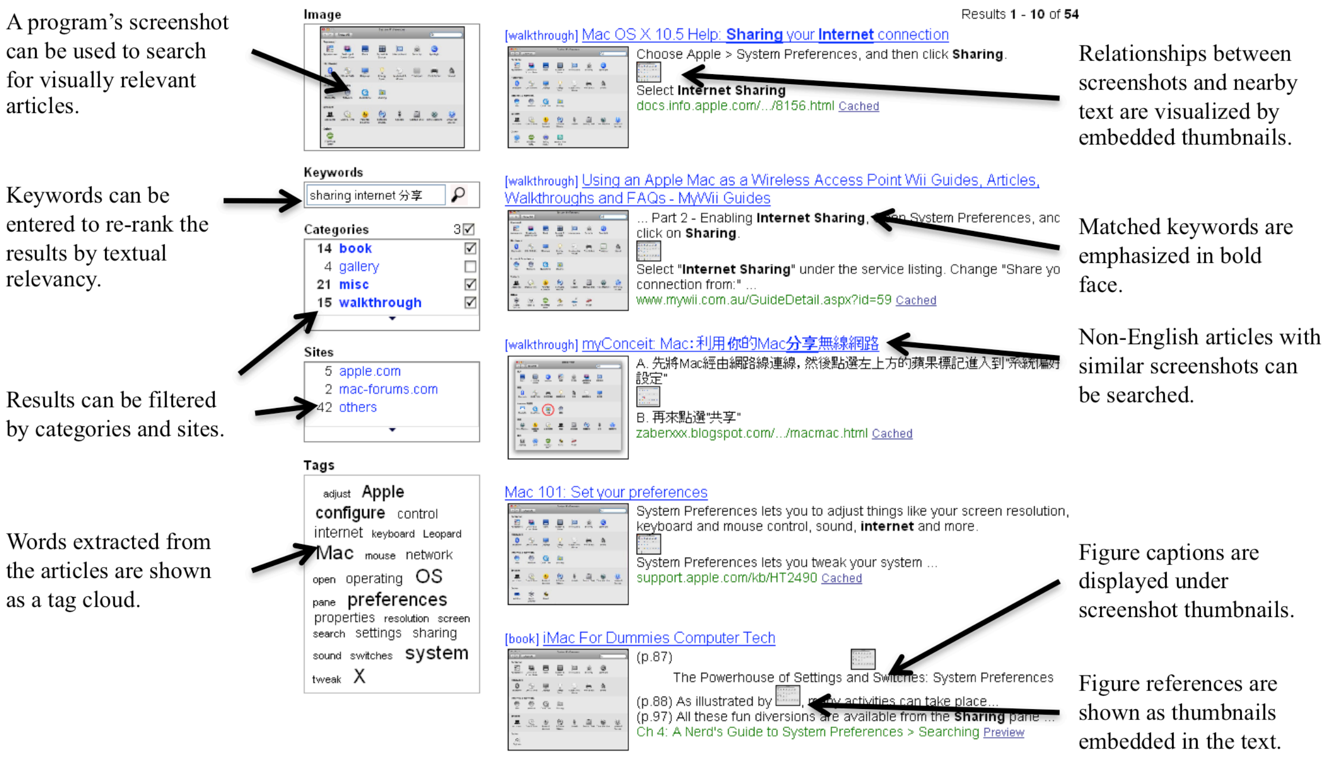
The major contribution of this paper is three-fold:

1. It presents a practical case for multimodal (visual + textual) query in the context of searching for online documentation about computer usage training and support.
2. A prototype with over 150K articles is built. This prototype implements the presented approach, and features state-of-the-art methods for indexing, retrieving and ranking search results. This prototype is used for evaluation and compared with related unimodal search engines.
3. The system is evaluated, and a preliminary analysis of the capabilities and limitations of query by image and text is provided. These results can be generalized to similar application scenarios to assess the impact of introducing multimodal search support in other systems.

2. RELATED WORK

End-user Applications. In terms of target users and tasks, the work closest to ours is that of Medhi et al. [23] who examined the optimal way to present audio-visual know-how knowledge to novice computer users. There have also been applications based on the ability to search images by content. For example, Google Goggles [8] is a mobile application that allows mobile users to take pictures of product labels (e.g., DVD, books) and submit these pictures as queries to lookup additional information about the products such as the price and the address of the online stores selling the products. TinEye [33] is a reverse-search engine that takes an image as input and finds other copies of the image on the Web for the purpose of detecting unauthorized uses of the image.

Content-based image retrieval by textual queries. Most content-based image retrieval systems use visual features as a way to support textual search. Textual queries are a common way to search for images (e.g., flickr.com, images.google.com). Different text sources are used to support search in these systems, including user generated annotations (e.g., tags) or text surrounding images in html pages. Many works have been proposed aiming at using visual content analysis to generate additional features for image retrieval, including automatic tagging and concept detection. Common applications include enriching indexes to improve retrieval effectiveness or enabling text queries under cold start conditions (e.g., when resources have not been assigned any tags). In general, automatic tagging methods use content and text features to derive models for predicting tag assignments. Generative mixture models have been widely used to learn joint probability distributions of visual features and vocabulary subsets [19]. A single model is derived for all vocabulary terms, limiting the effectiveness of these approaches. In contrast, discriminative models learn independent models for each target vocabulary term [21]. Discriminative models are commonly used to create concept detectors, with obvious applications to automatic tagging [31]. In both cases, the optimization criterion is related to the quality of generated annotations. In specific application settings, other criteria might serve to improve results. For instance, Grangier and



A program's screenshot can be used to search for visually relevant articles.

Keywords can be entered to re-rank the results by textual relevancy.

Results can be filtered by categories and sites.

Words extracted from the articles are shown as a tag cloud.

Image

Keywords

sharing internet 分享

Categories 3☑

14	**book**	☑
4	gallery	□
21	**misc**	☑
15	**walkthrough**	☑

Sites

5	apple.com
2	mac-forums.com
42	others

Tags

adjust Apple configure control internet keyboard Leopard Mac mouse network open operating OS pane preferences properties resolution screen search settings sharing sound switches system tweak X

Results **1** - **10** of **54**

[walkthrough] Mac OS X 10.5 Help: **Sharing** your **Internet** connection

Choose Apple > System Preferences, and then click **Sharing**.

Select **Internet Sharing**

docs.info.apple.com/.../8156.html Cached

[walkthrough] Using an Apple Mac as a Wireless Access Point Wii Guides, Articles, Walkthroughs and FAQs - MyWii Guides

... Part 2 - Enabling **Internet Sharing**, [illegible] System Preferences, anc click on **Sharing**.

Select "**Internet Sharing**" under the service listing. Change "Share yo connection from:" ...

www.mywii.com.au/GuideDetail.aspx?id=59 Cached

[walkthrough] myConceit: Mac:利用你的Mac**分享**無線網路

A. 先將Mac經由網路線連線，然後點選左上方的蘋果標記進入到"系統偏好設定"

B. 再來點選"共享"

zaberxxx.blogspot.com/.../macmac.html Cached

Mac 101: Set your preferences

System Preferences lets you to adjust things like your screen resolution, keyboard and mouse control, sound, **internet** and more.

System Preferences lets you tweak your system ...

support.apple.com/kb/HT2490 Cached

[book] iMac For Dummies Computer Tech

(p.87)

The Powerhouse of Settings and Switches: System Preferences

(p.88) As illustrated by , many activities can take place...

(p.97) All these fun diversions are available from the **Sharing** pane ...

Ch 4: A Nerd's Guide to System Preferences > Searching Preview

Relationships between screenshots and nearby text are visualized by embedded thumbnails.

Matched keywords are emphasized in bold face.

Non-English articles with similar screenshots can be searched.

Figure captions are displayed under screenshot thumbnails.

Figure references are shown as thumbnails embedded in the text.

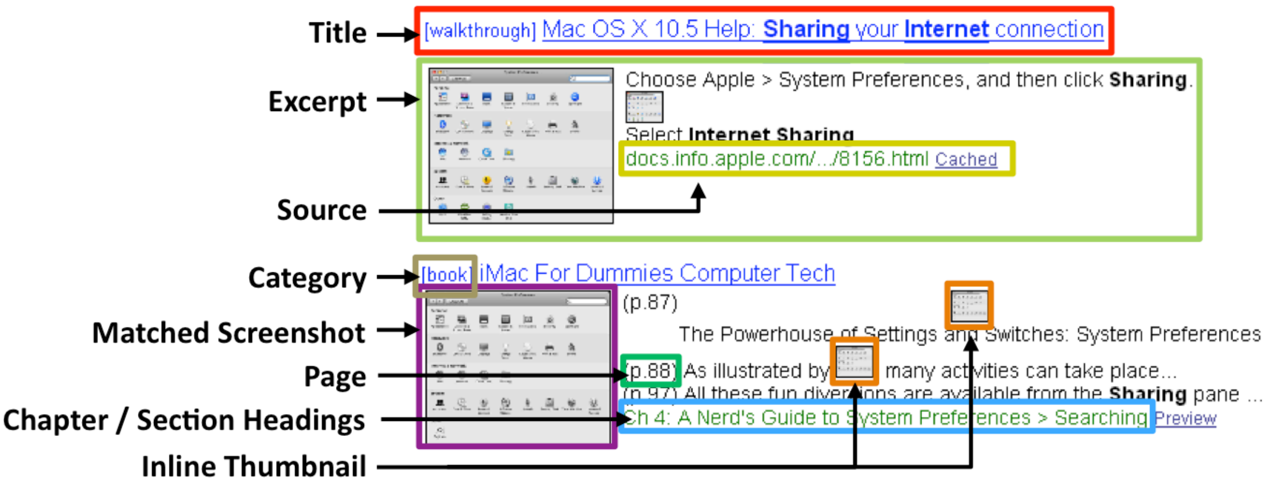
Figure 2. Example of the search results returned by our prototype system that supports query by image and text content for online help articles about GUI applications.

Bengio [9] propose a discriminative model targeted to image retrieval from text queries using an optimization criterion related to the final retrieval task, demoting the intermediate label generation stage. The problem can also be posed as a machine translation task, where models are created to translate visual features into image captions [7].

Indexing, searching, organizing Web images. Many classic problems originating from text search have been revisited in the new context of images. For example, Jing and Baluja [12] tackled the ranking problem in the context of searching product images using PageRank. Kennedy and Naaman [15] and Van Leuken et al. [34] both examined the problem of result diversification in the context of image search; the former considered a specialized case of landmark images, whereas the latter considered a more general case involving a wider range of topics such as animals and cars. Lempel and Soffer [18] dealt with the problem of authority identification for web images by analyzing the link structures of the source pages. Mehta et al. [24] presented a solution to the problem of spam detection for web images; they analyzed visual features and looked for duplicate images in suspiciously large quantities as potential spam. Li et al. [20] considered the problem of relevance judgment and demonstrated the ability of image excerpts (dominant image + snippets) to help users judge results faster. Crandall et al. [2] took on the challenge of organizing a large dataset in the setting of tens of millions of geo-tagged images, taking a comprehensive approach involving visual, textual, and temporal features. Similar to these existing works, the current work relies on state-of-the-art computer vision algorithms in order to index and search screenshots effectively. But unlike them, the current work finds similar images only as an intermediate step toward the ultimate goal of retrieving relevant text for users.

Multimodal queries. Dowman et al. [6] dealt with the problem of indexing and searching radio and television news using both speech and text. In the current work, we explore the potential of the particular modality pair of image and text for searching computer-related articles. Narayan et al. [26] developed a multimodal mobile interface combining speech and text for accessing web information through a personalized dialog. In the current work, we explore the potential of the particular modality pair of image and text for searching computer-related articles.

Internationalization. In response to accelerated globalization, there have been research efforts dedicated to making the Web more accessible to international users independent of the languages they speak. For example, Scholl et al. [29] proposed a method to search the Web by genres such as blog and forum in a language independent manner. Ni et al. [27] explored ways to analyze and organize Web information written in different languages by mining multilingual topics from Wikipedia. Tanaka-Ishii and Nakagawa [32] developed a tool for language learners to perform multilingual search to find usage of foreign languages. A plethora of literature about cross lingual information retrieval (CLIR) exists. These methods follow a very similar pattern: the query [28] or the target document set [3] is automatically translated and search is then performed using standard monolingual search. This process is heavily dependent on the results of the machine translation stage, and varies substantially across different languages and topics [3]. In relation to research efforts, the current work offers yet another interesting and promising possibility–searching based on the universal visual language of screenshots.



[walkthrough] Mac OS X 10.5 Help: **Sharing** your **Internet** connection

Choose Apple > System Preferences, and then click **Sharing**.

Select **Internet Sharing**

docs.info.apple.com/.../8156.html Cached

[book] iMac For Dummies Computer Tech

(p.87)

The Powerhouse of Settings and Switches: System Preferences

(p.88) As illustrated by many activities can take place...

(p.97) All these fun diversions are available from the **Sharing** pane ...

Ch 4: A Nerd's Guide to System Preferences > Searching Preview

Figure 3. Visual and text elements included in the summary of each search result to help users judge the visual and textual relevance of the result.

The problem of finding non-English information has been recently studied by Lazarinis et al. [17], and it has been shown that search engines typically ignore the intricacies of non-English natural languages resulting in lower retrieval accuracy. Typical problems include: language identification, handling of different encodings, and particularities of text pre-processing, segmentation, stopword determination and stemming in each different language. Our approach tries to decrease the impact of these multi-lingual issues by enriching queries with visual content.

3. WEB SEARCH FOR ONLINE HELP

We present the case of users searching the web for help on the use of GUI applications as an example of an application scenario suitable for multimodal (textual + visual) queries. In this section, we discuss the main reasons that support this hypothesis by focusing on two complementary aspects. On the one hand, multimodal queries in this context can be introduced in the users' query formulation process in a more natural way in contrast to other CBIR applications. On the other hand, the addition of the visual modality is likely to improve retrieval effectiveness and, therefore, enhance the user experience.

Web articles providing helpful information about GUIs tend to be richly illustrated by screenshots of the GUIs. In this scenario, users might prefer articles not only relevant to the particular problem or task they have encountered, but also relevant to the particular GUI window or dialog box they are using. This task could benefit from the use of multimodal queries. Using only keywords, users need to enter additional terms to describe the application context as well as the help topic. However, previous research works have studied textual query formulation and found it to be counter-intuitive and problematic for the vast majority of users [25]. Under these conditions, it is unlikely that users would be able to retrieve relevant content efficiently. A way to deal this problem is to allow users to provide a screenshot of the problematic window. Most of the contextual description will then get embedded into the visual modality. Therefore, we believe that querying by visual and textual features naturally helps ensuring that articles retrieved would be both visually relevant to the application context and textually relevant to the desired topic.

To query by visual features, the system needs to perform content-based image retrieval (CBIR). While CBIR has been the target of active research, Many technical challenges remain that limit CBIR's practicality in unrestricted contexts, such as the challenges of sensory and semantic gaps [30]. In terms of usability, one big challenge is related to the process of formulating queries. Different strategies have been proposed such as choosing the desired color distribution and brightness level of the retrieved images, and yet none has been proven suitable for deployment in large-scale systems. A promising approach is to use a pictorial example either in the form of a picture or a hand-drawn sketch, which is then used by the system to extract visual features to generate a query. A hindrance to this approach, however, is the need to have access to a representative image in the first place, a problem known as the *page zero* problem. Our proposed system is immune to this problem. When users need to acquire a representative query image of the target interface, they can simply capture the screenshot of the interface that is visible right in front of their eyes.

We developed an integrated search interface as a stand-alone Java application to support this multimodal search. This interface allows users to capture a screenshot of any interface, enter some query keywords, and submit the resulting multimodal query to the search engine, and display the search result in a Web browser. For users who do not wish to install a Java application, they can alternatively use a Web form to upload a screenshot and enter some keywords to perform search.

To evaluate multimodal search in this context, we built a prototype including all this functionality. Next section focuses on the discussion of the implementation of this prototype. In Section 5 we will evaluate the performance of this prototype in our problem setting.

4. PROTOTYPE

As proof of concept, we built a prototype search engine for online GUI help articles. Figure 2 shows the typical results returned by our prototype search engine. Building this prototype involves collecting and indexing a large number of articles containing GUI screenshots from the Web (Section 4.1), extracting relevant text from the articles (Section 4.2), and presenting search results to users in a meaningful way (Section 4.3).

4.1 Supporting Query By Image Content

The database of our prototype system currently has more than 150k articles containing screenshots of GUI-based applications. These articles were collected using three methods:

- **Image Search Engine.** We submitted keywords related to popular computer applications to collect screenshots of them from Bing Images (http://www.bing.com/images/). Some examples of these keywords are: *properties*, *preferences*, *option*, *settings*, *wizard*, *custom*, *installation*, *network*, *sound*, *keyboard* ... etc. We turned on the filter feature provided by Bing to accept only illustrations and graphics and reject natural images such as images of faces and sceneries. For each image accepted, we downloaded the source article and stored in the database. About 100k articles were obtained using this method.
- **Reverse Image Search Engine.** We used TinEye [11], a reverse image search engine that takes an image as input and returns a list of URLs to nearly identical copies of the image found on the Web. The main use of TinEye is for copyright infringement detection. To use TinEye for our purposes, we first manually captured screenshots of more than 300 interactive windows of popular programs across three popular OS platforms (XP, Vista, and Mac OS). These images were submitted to TinEye to obtain about 5,000 addition screenshots. The source articles of these screenshots were downloaded and saved in the database.
- **Electronic Books.** We collected a library of 102 electronic books of popular computer programs, such as the book *XP for Dummies* and *Windows Vista Inside Out*. We extracted all the image figures embedded in the PDF files of these books. Many of these figures are screenshots of computer programs. The source page of each screenshot is saved in the database. We obtained 50k pages using this method.

Each method has its own pros and cons. While Bing Image Search provides the best variety of images, many of them are not visually relevant to any program at all. TinEye is able to provide visually relevant images, but these images are ranked only by visual relevancy; the page containing the highest ranked image may not necessarily contain any useful information. Computer books are professionally edited and thus contain the highest quality text; however, they cover a relatively limited range of applications and their content is not as up-to-date when compared to the Web. By using all of these methods, we hope to create a rich repository of technical information that is both visually and textually relevant to, and accessible by, general computer users. Table 1 below lists the top ten sites from which we obtained the most screenshots and associated articles.

Table 1. Top 10 sites from which the articles in our prototype database were obtained.

microsoft.com	320	qweas.com	110
ehow.com	183	softpedia.com	110
brothersoft.com	163	bestshareware.net	108
techrepublic.com	121	gateway.com	106
versiontracker.com	113	askdavetaylor.com	106

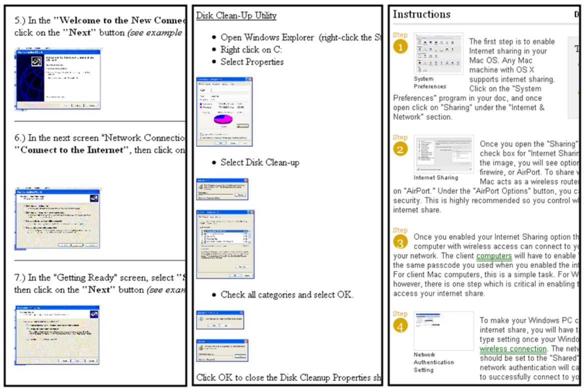
After collecting a large number of screenshots, we constructed a visual index of these screenshots in order to support search by visual content. GUI screenshots feature several unique properties that need to be taken into account when choosing a visual indexing algorithm. Screenshots included in web pages or books are often post-processed by authors. For instance, a screenshot may be cropped to draw readers' attention only to the relevant part of the application, scaled down to conserve space, or scaled up to improve readability. In contrast, rotation is less common. Also, screenshots of GUI applications contain text components such as captions, labels, and titles, which may suggest the possibility of applying OCR. However, current OCR engines have been optimized for printed text, assuming the text is formatted in columns, whereas GUI text components tend to be scattered.

We adopted the visual indexing algorithm developed by Jegou et al. [33]. This algorithm supports partial match of cropped images and is robust to variation in scale. While this algorithm can be configured to handle rotational variation, we turned this feature off to improve efficiency, because robustness to rotational variation is not necessary in our application. A brief description of the algorithm by Jegou et al. is given as follows. At the indexing time, all screenshots in the database are processed to extract SURF descriptors [1]. A large sample of SURF descriptors are clustered using k-means (we set k to 1000). For each cluster, the median vector is computed to represent the center of the cluster. Every descriptor of each database screenshot is then assigned to the closest cluster based on the descriptor's distance to the cluster center. Finally, each database screenshot (along with the link to the source article) is indexed by the clusters its SURF descriptors belong to.

At the query time, a query screenshot is processed to extract a set of SURF descriptors V. For each descriptor v in V, visually similar descriptors in the database are retrieved. This retrieval is done efficiently by first identifying the closest cluster and then comparing v only to the small subset of descriptors in the cluster. The comparison is based on Hamming Embedding, which compresses a descriptor's 64 floating numbers into a single 64-bit word while preserving the ability to estimate the distance between descriptors. Because of the compactness, the embedding can be efficiently stored and compared. A descriptor determined to be similar can then vote on the article from which it was originally extracted. After all the query descriptors in V are processed, the articles associated with the screenshots containing the most number of similar descriptors would have received the most votes and can be returned as the result.

4.2 Supporting Query by Text Content

After collecting and indexing screenshots, we extracted and indexed relevant text from the source articles in order to support search by text content. The objective of text extraction is to identify snippets in an article that can be displayed alongside the matched screenshots to suggest users what kinds of knowledge they may find if they read the articles containing the screenshots. For example, for an article containing a screenshot of a dialog window with a visible title *network configuration*, a good text snippet would be related to computer tasks such as *set IP address* or *enable wireless network* since it clearly indicates that the article is likely to explain how to set the IP address or enable the wireless network using this dialog window. Also, a sentence referring to a screenshot such as *click on the button shown above* is useful to relate the action to the right screenshot. On the other hand, a snippet merely describing the visual content of the screenshot, such as the words *network configuration* that are already visible, would be redundant; users can already see these words in the screenshot and do not learn anything new. Note that this is in sharp contrast with the needs of typical keyword-based image search engines that actually favor text describing image contents such as the alternative text for images given in the `alt` attribute. In summary, desirable text snippets in our application are those



5.) In the "**Welcome to the New Connec**
click on the "**Next**" button *(see example*

6.) In the next screen "Network Connectio
"**Connect to the Internet**", then click on

7.) In the "Getting Ready" screen, select "
then click on the "**Next**" button *(see exan*

Disk Clean-Up Utility

- Open Windows Explorer (right-click the St
- Right click on C:
- Select Properties
- Select Disk Clean-up
- Check all categories and select OK.

Click OK to close the Disk Cleanup Properties sh

Instructions

Step 1 The first step is to enable Internet sharing in your Mac OS. Any Mac machine with OS X supports internet sharing. Click on the "System Preferences" program in your doc, and once open click on "Sharing" under the "Internet & Network" section.

System Preferences

Step 2 Once you open the "Sharing" check box for "Internet Sharin the image, you will see optior firewire, or AirPort. To share w Mac acts as a wireless router on "AirPort." Under the "AirPort Options" button, you c security. This is highly recommended so you control wl internet share.

Internet Sharing

Step 3 Once you enabled your Internet Sharing option th computer with wireless access can connect to yo your network. The client computers will have to enable the same passcode you used when you enabled the int For client Mac computers, this is a simple task. For W however, there is one step which is critical in enabling t access your internet share.

Step 4 To make your Windows PC c internet share, you will have t type setting once your Windo wireless connection. The netw should be set to the "Shared" network authentication will ca to successfully connect to yo

Network Authentication Setting

Figure 4. Examples of step-by-step walkthrough articles.

that (1) describe typical computer tasks and (2) make references to screenshots. We consider the following four types of text snippets:

- **Title/Heading.** An article's title often highlights the most important point covered by the article, whereas headings indicate the topics of the sections. Headings are especially useful when there are multiple screenshots in the same article, for example, a step-by-step instruction. For web articles, titles, and headings can be extracted from HTML tags (e.g., `<title>`, `<h1>`). For book articles, we extracted the chapter and section headings from the outline of the PDF files. We linked an article's title to every screenshot in the article and each heading to the screenshots in the section below the heading.
- **Action Phrases.** Since the primary use of our system is to help users find articles to learn how to perform certain computer tasks, articles containing common computer action words are particularly desirable. We compiled a list of common action words such as *configure*, *click*, *type*, and *open* and identified sentences containing these words around the screenshots.
- **Caption.** Screenshots are often accompanied by captions to succinctly describe the idea or step illustrated by the screenshots. For book articles, we looked for sentences adjacent to screenshots that begin with the word *Figure*, which turned out to be a reliable way to identify captions. For web articles, we were unable to rely on this method. Instead, we used two heuristics: (1) whether the text is situated immediately above or below the screenshot, and (2) whether the text has a different style compared to surrounding paragraphs.
- **References**. When a sentence makes an explicit reference to the screenshot of an application, its content is likely to be useful to the users regarding the application. Some references are based on explicit labels, such as the phrase *as shown in Figure X.* Some references are based on layout relationships, such as the phrase *see below*. These references can sometimes be distant from the screenshots they refer to. To resolve a label-based reference, we looked for the screenshot whose caption contains a matching label. In web articles, screenshots and references are expected to be on the same page, whereas in book articles, they may be a few pages apart due to page size limitations. To resolve a layout-based reference, we scanned in the direction indicated in the reference (e.g., scanning forward for *see below*) until a screenshot is found. After resolving the reference, a link between the referred screenshot and the referring sentence was created and stored in the database.

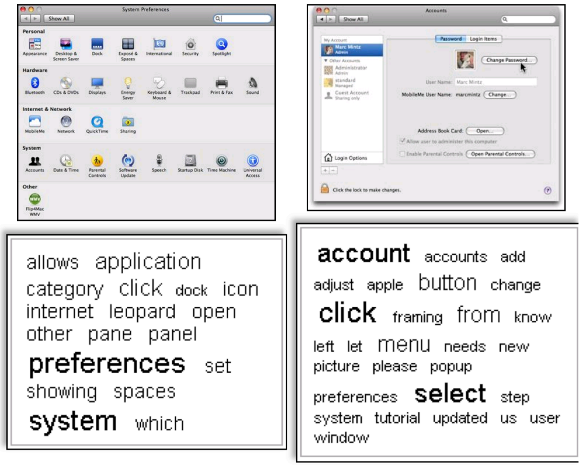


Figure 5. Two examples of the tag clouds (bottom) generated from the set of articles containing screenshots matching the query screenshots (top).

In addition to extracting useful text from the articles, we can also classify them into distinct categories based on their origins, structures, and contents. The prototype included four categories: book, walkthrough, gallery, and general. The book category consists of the articles extracted from online electronic books. The walkthrough category consists of the articles providing step-by-step instructions (see Figure 4 for examples). We detected these walkthrough articles based on three heuristics: (1) whether there is an ordered or unordered list of short sentences interspersed with screenshots, (2) whether these short sentences begin with a common computer action word, and (3) whether there are indicative terms such as walkthrough, step, and guide. The gallery category consists of the articles that include many images but relatively small amount of text. The general category consists of all the other articles. Table 2 shows the distribution of the articles over the four categories.

Table 2. Number of the articles in each category in the database of our prototype system

General	Walkthrough	Gallery	Book	Total
51,743 (33.1%)	45,249 (28.9%)	4,464 (2.9%)	55,244 (35.1%)	156,700

4.3 Presenting Search Results

We designed and implemented a front-end interface to present the search results to users. The primary design goal of this interface is to enable users to quickly judge the relevancy of each search result by image and text content. The secondary design goal is two-fold: users can filter results and can discover new information through exploratory search.

To serve the primary design goal, the search results include both the thumbnails of the matched screenshots and useful text snippets extracted from the source articles. Users can look at the thumbnails to judge the visual relevance and read the text snippets to judge the textual relevance of the results. Figure 3 shows a typical presentation of a single result. The presentation consists of three major elements: *title*, *excerpt*, and *source*. The *title* element displays the title of the article. A marker is shown in the beginning of the title line to indicate the category of the article (i.e., walkthrough, book, or gallery). Users can click on the title to

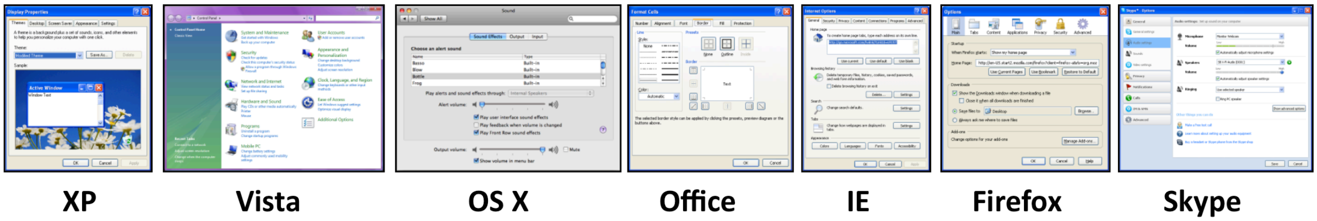


Figure 6. Examples of the screenshot queries used in the evaluation of visual matching performance (Section 5.1).

open the original article in the browser and read the entire content. The excerpt element displays the thumbnail of the matched screenshot and relevant text snippets. The dimension of the thumbnail is at least 150 by 150 pixels to ensure enough detail is available for identification [14]. Keywords in the snippets are highlighted. A tiny inline thumbnail (15 by 15 pixels) of the same matched screenshot is inserted into the snippet to indicate the relationship between the snippet and the screenshot. If a snippet is the caption of a screenshot, a tiny thumbnail of the screenshot is displayed above the caption and centered. If a snippet contains a textual reference to a screenshot, the reference is visually represented by a tiny thumbnail of the screenshot (e.g., *as shown in Figure 2→as shown in* **).** The *source* element displays the abbreviated URL of the web article or the chapter or section heading of the book article

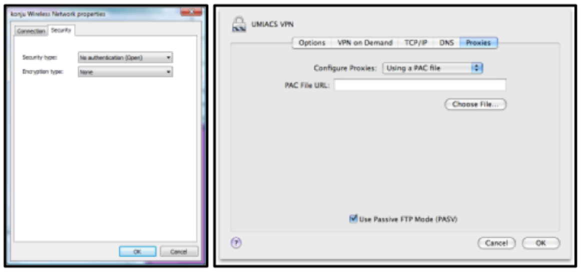
To serve the secondary goal, we used the Exhibit framework developed by Huynh et al. [10] to provide a set of faceted search capabilities. In Figure 2, two faceted filtering controls can be seen in the left-side panel to enable users to filter the results based on categories and sites. Furthermore, to enable users to discover unexpected but useful information, the interface also displays a tag cloud of the words found in all the articles in the search results. Figure 5 shows two examples tag clouds. The larger the word, the more frequently the word co-occurs with the query screenshot. Looking at the tag cloud allows users to get a quick overview of the range of the topics covered by the retrieved articles. If the name of the right OS or application appears in the cloud as prominent word, users can be more confident in the relevance of the results. Moreover, by clicking on a tag, users can filter the results to show only those containing the tag.

5. Evaluation

This section present the evaluation of the prototype described, focusing on the effectiveness and usability of the system.

5.1 Image Matching Performance

We first considered the study of our system's ability to find matching screenshots by visual content, one of its most critical functions. We created a test set by capturing the screenshots of 352 unique application windows. This test set covers three major operating systems (Windows XP, Windows Vista, and Mac OS X) and several popular programs such as Microsoft Office, Firefox, and Skype. Figure 6 shows some examples of the screenshots in the test set. We retrieved the top 10 matches for each test screenshot and evaluated their correctness. A match is considered correct if it is visually similar to the query screenshot. Since we did not threshold the similarity score, it is possible that none of the 10 matches is correct.


konju Wireless Network properties

Connection | Security

Security type: No authentication (Open)

Encryption type: None

UMIACS VPN

Options | VPN on Demand | TCP/IP | DNS | Proxies

Configure Proxies: Using a PAC file

PAC File URL:

☑ Use Passive FTP Mode (PASV)

Figure 7. Examples of query screenshots that are difficult to match due to the lack of visual features.

We recruited users from the Amazon Mechanical Turk (AMT) to evaluate the correctness of these matches. AMT is a vibrant micro-task market where people can be recruited to perform simple tasks for small monetary awards, such as labeling images and fixing broken links. AMT has been applied in various research contexts. For instance, Kittur et al. [16] reported success in recruiting workers from AMT to rate Wikipedia articles and achieved quality comparable to that by expert raters. However, as identified by Kittur et al., while AMT makes it easy for researchers to obtain opinions from a diverse population rather than from 1 or 2 hired experts, there are risks of getting low-quality answers from dishonest subjects. These risks can often be prevented by building into the tasks a mechanism to detect and discourage cheating. Thus, we inserted into the list of matches one known good match (the test image itself), two known bad matches (other test images), and two duplicate matches, and randomly shuffled the list. To pass the quality test, a subject must correctly identify all the known good and bad matches as well as give consistent answers to duplicate matches.

Out of 352 screenshots of unique GUI application windows, 317 (90%) retrieved at least one correct match. This finding suggests that a typical state-of-the-art content-based image retrieval algorithm originally invented for general images, such as the one proposed by Jeguo [11] and implemented in our prototype, is applicable to the specific type of images (i.e., screenshots) in our application. We also looked at the query screenshots that did not retrieve any good match and noticed that these hard-to-match screenshots tend to contain very few visual features (see Figure 7) compared to the screenshots that yielded good results. As a future work, there are two possible ways to deal with these problematic screenshots. First, we can consider structural features such as dividing lines, widget borders, and the arrangements of widgets. Second, we can consider embedded text such as captions and labels. To extract text reliably from screenshots, it would be necessary to modify existing OCR engines (which have been optimized for printed text) to accommodate the characteristics of GUI screen text, such as low resolution (e.g., some text can be as small as in 5 point font) and arbitrary layout.

5.2 Perceived Usefulness

The previous section concentrates on the performance of screenshot matching, whereas in this section we turn attention to the utility of multimodal queries, as proposed in our approach. Our goal is to evaluate the perceived usefulness of search results between three different query approaches:

1. Query-by-image to retrieve text articles with images (Our approach)
2. Query-by-keywords to retrieve text articles (Bing Web Search)
3. Query-by-keywords to retrieve images (Bing Image Search)

To form the query set, we selected 20 popular applications, 10 for XP and 10 for Mac. For our query-by-screenshot prototype, a query is simply a screenshot of the application. For Bing Web Search and Bing Image Search, a query is a set of words in the title of the application window plus the name of the operating system (e.g., system preferences mac). Queries were submitted to respective systems to retrieve the top 10 results.

We recruited users from the Amazon Mechanical Turk to rate the results in terms of usefulness. The rating was done on a 5-point *Likert* scale (1: completely useless, 5: very useful). To control quality, we inserted into the results at random positions four junk results. For a rater to pass the quality test, these junk results must be marked as less useful than the other real results. Figure 8 shows an example of a search result returned by each system. After rejecting disqualified ratings, we obtained 1677 ratings from 33 unique workers.

Table 3 shows the results found in this experiment. On average, users tend to perceive results obtained using our query-by-screenshot approach as useful in the proposed application scenario, with an average value of 4.08. Values obtained in this experiment clearly illustrate that our approach produces comparable results to those of well-established search systems, and users tend to find them slightly more useful. However, it is important to note that the results obtained by our approach in this experiment have been generated using only visual information. The fact that we can enrich this query by a) using robust OCR to automatically generate text keywords and b) allowing the user to provide custom keywords further support the viability of this approach. In general, text and visual modalities are orthogonal and merging features can dramatically boost retrieval precision as shown in [36].

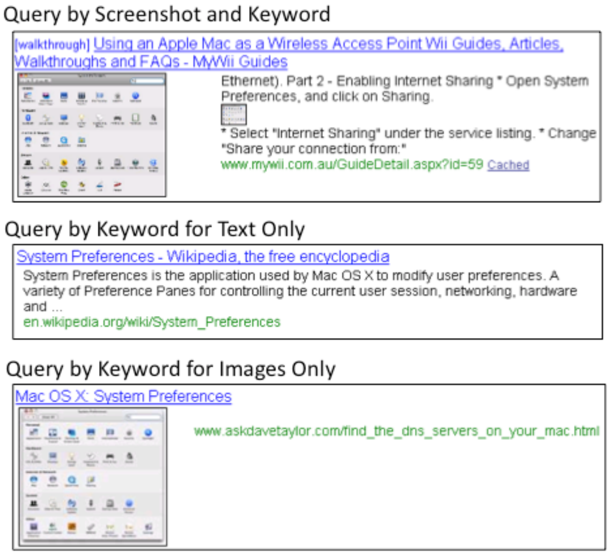
These findings are consistent with our hypothesis that being able to see a visually matching screenshot as well as an informative text excerpt in a result can help users feel more confident in the usefulness of the result.

Table 3. Perceived usefulness of the search results.

Method	Rating
Ours	4.08±1.28
Query-by-keywords for Images	3.27±1.48
Query-by-keywords for text	3.31±1.54

5.3 Features for Ranking Articles

The two previous sections have dealt with the issue of establishing the viability of our approach, both in terms of image identification performance and usefulness of search results obtained. Textual features were purposely dismissed to provide evidence of the utility of the visual modality for query formulation. In this section we consider the analysis of the combined set of features (textual + visual). We aim at studying which features are perceived by users as being more important to rank search results, which in a way reflects their relative usefulness from the user perspective perceived by users. We considered the following types of features:



Query by Screenshot and Keyword

[walkthrough] Using an Apple Mac as a Wireless Access Point Wii Guides, Articles, Walkthroughs and FAQs - MyWii Guides

Ethernet). Part 2 - Enabling Internet Sharing * Open System Preferences, and click on Sharing.

* Select "Internet Sharing" under the service listing. * Change "Share your connection from:"

www.mywii.com.au/GuideDetail.aspx?id=59 Cached

Query by Keyword for Text Only

System Preferences - Wikipedia, the free encyclopedia

System Preferences is the application used by Mac OS X to modify user preferences. A variety of Preference Panes for controlling the current user session, networking, hardware and ...

en.wikipedia.org/wiki/System_Preferences

Query by Keyword for Images Only

Mac OS X: System Preferences

www.askdavetaylor.com/find_the_dns_servers_on_your_mac.html

Figure 8. Examples of the results retrieved using three different query methods.

- **Visual.** Is the screenshot in the article visually similar to the query screenshot?
- **Relative Size.** Has the screenshot in the article been enlarged or downsized relative to the actual size of the GUI application?
- **Position.** Where in the article does the screenshot appear (beginning, middle, or end)?
- **Visual Density.** Does the article have a lot or a few images?
- **Category.** Does the article belong to the walkthrough, gallery, book, or general category?
- **Text Description.** Is the screenshot referred to or captioned by some text in the article?
- **Keywords.** Does the article contain search keywords specified by the users?
- **Authority.** Is the article hosted by a site with authority, for instance, by software vendors such as Microsoft and Apple or by dedicated knowledge sharing sites such as eHow?

We applied RankSVM [13] to learn how to weight these features. RankSVM is a machine learning technique developed by Joachims to learn feature weights from training data that consists of a set of ordering constraints. Typically these constraints are inferred from user click-through data based on the assumption that the search results users clicked on should have been ranked higher than the rest of the results users did not click on. Since we did not have any user click-through data to work with, we instead recruited users from the Amazon Mechanical Turk to give us subjective ratings on a set of search results given by our prototype system in the earlier empirical analyses. We randomly shuffled the order of the results to minimize the potential effect of the ordering bias on rating given by the users. Similar to before, users were asked to give a rating of the usefulness of each search result on a 5-point *Likert* scale. To control quality, two duplicate results and two junk results were added at random positions. Users must give consistent ratings to duplicate results and lower ratings to junk results in order for their input to be accepted. Each pair of user ratings would be an ordering constraint for applying the RankSVM technique.

Table 4 below gives a complete list of features included in RankSVM. They are shown in two groups, *important* and *less important*. Features that were given high weights by RankSVM are considered important for ranking. Not surprisingly, both **visual similarity** and **keywords** were found to be important for ranking, which supports our hypothesis that users care about both the visual and textual relevance of the search results. For **category** features, users preferred articles in the *walkthrough* categories to those in other categories. For **text description** features, we speculated that *caption* and *nearby text* were less important because they sometimes merely describe the title of the particular window already shown in the screenshot. On the other hand, *title* and *reference* tend to describe the topic or task and are more likely to provide useful information.

Table 4. Ranking features grouped by importance as determined by RankSVM.

Ranking	Feature Name
Important	**Visual Similarity** **Keywords** **Text Description** (*Title*, *Reference*) **Category** (*Walkthrough*) **Authority**
Less Important	**Visual Density** **Position** **Category** (*Book*, *Gallery*, *General*) **Relative size** **Text description** (*Caption*, *Nearby text*)

5.4 Cross-lingual Search

In this analysis, we examined the extent to which our system can support cross-lingual search when the screenshots of GUI applications in different languages are submitted as queries. We took the screenshots of 15 representative programs on Mac in five languages (Spanish, French, German, Chinese, and Korean) and obtained a total of 75 test images. We used these query images to retrieve the top 10 matches and counted the number of correct matches. Table 5 summarizes our findings. Some screenshots retrieved matching images even though they are in a different language. These screenshots tend to contain visually rich patterns such as logos. On the other hand, some screenshots failed to retrieve any good match; they tend to be dominated by text and look very different across languages. This is especially true in block-based languages (i.e., Chinese and Korean) whose printed words often occupy smaller areas and appear denser than western alphabet-based words. Our finding suggests that query by screenshot can provided the added benefit of cross-lingual search in cases when screenshots are dominated by visual features.

6. Conclusion and Future Work

We introduced a practical case for multimodal (screenshot + keywords) query in the context of searching for online articles about computer usage training and support. We built a prototype search system with over 150K articles in its corpus and described how the corpus was created, how the articles were indexed, retrieved, and ranked. Using this prototype system, we analyzed the potential capabilities and limitations of query by image and text. Possible future research directions include scaling the prototype system up to millions of articles, deploying the prototype system to a wider user population, and evaluating the system's technical performance and usability in real-world use situations.

7. Acknowledgements

This work is supported by the ONR grant N000140910044 and by the EU FP7 Marie Curie IOF project MIESON (PIOF-GA-2009-254370).

8. REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Speeded-up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the World's Photos. In *WWW '09: Proc. of the 18th International Conference on World Wide Web*, pages 761–770, New York, NY, USA, 2009. ACM.

[3] P. Clough. Caption and Query Translation for Cross-Language Image Retrieval. *Multilingual Information Access for Text, Speech and Images,* Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005.

[4] P. Clough and I. Eleta. Investigating Language Skills and Field of Knowledge on Multilingual Information Access in Digital Libraries. *International Journal of Digital Library Systems (IJDLS)*, volume 1(1), pp. 89–103.

[5] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Comput. Survey*, 40(2):1–60, 2008.

[6] M. Dowman, V. Tablan, H. Cunningham, and B. Popov. Web-assisted Annotation, Semantic Indexing and Search of Television and Radio News. In *WWW '05: Proceedings of the 14th International Conference on World Wide Web*, pages 225–234, New York, NY, USA, 2005. ACM.

[7] P. Duygulu, K. Barnard, J. de Freitasand D. Forsyth. Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In *Proceedings of ECCV 2002*, pages 349–354.

[8] Google Goggles, Use Pictures to Search the Web, http://www.google.com/mobile/goggles/

[9] D. Grangier and S. Bengio. A Discriminative Kernel-based Approach to Rank Images from Text Queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 8, pages 1371–1384 (August 2008).

[10] D. F. Huynh, D. R. Karger, and R. C.Miller. Exhibit: Lightweight Structured Data Publishing. In *WWW '07: Proc. of the 16th International Conference on World Wide Web*, pages 737–746, New York, NY, USA, 2007. ACM.

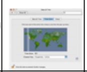
[11] H. Jegou, M.Douze, and C.Schmid. Hamming Embedding and Weak Geometric Consistency for Large-scale Image Search. In *ECCV '08: Proceedings of the 10thEuropean Conference on Computer Vision*, pages 304–317, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] Y. Jing and S. Baluja. PageRank for product image search. In *Proc. of the 17th International Conference on World Wide Web*, pages 307–316, 2008. ACM.

[13] T.Joachims. Optimizing Search Engines using Clickthrough Data. In *KDD '02: Proceedings of the8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002. ACM.

[14] S. Kaasten, S. Greenberg and C. Edwards. How People Recognize Previously Seen Web Pages from Titles, URLs and Thumbnails. In *Proc. of HCI,* pages 247–265, 2001.

Table 5. Number of correct top 10 matches for 15 application windows in five non-English languages: Spanish (S), French (F), German (G), Chinese (C), and Korean (K).

															
S	10	9	6	6	6	4	6	2	3	1	0	0	1	0	0
F	10	8	6	7	8	5	0	1	4	1	4	0	0	0	0
G	10	8	6	7	7	5	2	2	3	7	0	1	0	0	0
C	4	9	7	7	8	5	1	1	1	0	0	0	0	0	0
K	9	9	7	8	8	3	2	1	0	0	1	1	0	0	0

[15] L. S. Kennedy and M. Naaman. Generating Diverse and Representative Image Search Results for Landmarks. In *Proc. of the 17th International Conference on World Wide Web*, pages 297–306, 2008. ACM.

[16] A.Kittur, E. H. Chi, and B.Suh. Crowd-sourcing User Studies with Mechanical Turk. In *CHI '08: Proceeding of the 26th SIGCHI Conference on Human Factors in Computing Systems*, pages 453–456, 2008.ACM.

[17] F. Lazarinis, J. Vilares, J. Taitand E. N. Efthimiadis. Current Research Issues and Trends in non-English Web Searching. *Information Retrieval.* Vol. 12, Issue 3, pages 230–250. Kluwer Academic, 2009.

[18] R. Lempel and A.Soffer. Picashow: Pictorial Authority Search by Hyperlinks on the Web. In *Proc. of the 10th International Conference on World Wide Web*, pages 438–448, 2001. ACM.

[19] J. Liand J. Z. Wang. Real-time Computerized Annotation of Pictures. In *Proc. of the 14th ACM International Conference on Multimedia*, pages 911–920, 2006. ACM.

[20] Z. Li, S. Shi, and L. Zhang. Improving Relevance Judgment of Web Search Results with Image Excerpts. In *WWW '08: Proceeding of the 17thInternational Conference on World Wide Web*, pages 21–30, 2008. ACM.

[21] S. Lindstaedt, R. Mörzinger, R. Sorschag, V. Pammer, and G. Thallinge. Automatic Image Annotation using Visual Content and Folksonomies. *Multimedia Tools and Applications.* Vol. 42, No. 1, pages 97-113, 2009.

[22] Y. Liu, J. Bian, and E. Agichtein. Predicting Information Seeker Satisfaction in Community Question Answering. In *Proc. of the 31stSIGIR*, pages 483–490, 2008. ACM.

[23] I. Medhi, A. Prasad, and K. Toyama. Optimal Audio-visual Representations for Illiterate Users of Computers. In *Proc. of the 16th International Conference on World Wide Web*, pages 873–882, New York, NY, USA, 2007. ACM.

[24] B. Mehta, S. Nangia, M. Gupta, and W. Nejdl. Detecting Image Spam using Visual Features and Near Duplicate Detection. In *Proc. of the 17th International Conference on World Wide Web,* pages 497–506, 2008. ACM.

[25] V. Murdock, D. Kelly, W. B. Croft, N. J. Belkin, X. Yuan, Identifying and Improving Retrieval for Procedural Questions, *Information Processing & Management*, Volume 43, Issue 1, January 2007, Pages 181–203.

[26] M. Narayan, C. Williams, S. Perugini, and N. Ramakrishnan. Staging Transformations for Multimodal Web Interaction Management. In *WWW '04: Proceedings of the 13th International Conference on World Wide Web*, pages 212–223, 2004. ACM.

[27] X. Ni, J. T. Sun, J. Hu, and Z. Chen. Mining Multilingual Topics from Wikipedia. In *WWW '09: Proceedings of the 18th International Conference on World Wide Web*, pages 1155–1156, 2009. ACM.

[28] K.Reiter, S. Soderlandand O. Etzioni. Cross Lingual Image Search on the Web. In *Proc. of the 20th International Joint Conference on Artificial Intelligence*, 2007.

[29] P. Scholl, R. D. Garcia, D. Bohnstedt, C. Rensing, and R. Steinmetz. Towards Language-independent Web Genre Detection. In *Proc. of the 18th International Conference on World Wide Web*, pages 1157–1158, 2009. ACM.

[30] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. Pattern Anal. Mach. Intell.* Volume 22, N. 12.

[31] C. G. Snoek, and M. Worring. Concept-Based Video Retrieval. *Found. Trends Inf. Retr.*Vol 2, N 4, pp 215-322, 2008.

[32] K. Tanaka-Ishii and H. Nakagawa. A Multilingual Usage Consultation Tool based on Internet Searching: More Than a Search Engine, Less Than QA. In *Proc of the 14th International Conference on World Wide Web*, pages 363–371, NewYork, NY, USA, 2005. ACM.

[33] TinEye, Reverse Image Search Engine, http://www.tineye.com/

[34] R. H. van Leuken, L. Garcia, X. Olivares, and R. van Zwol. Visual Diversification of Image Search Results. In *WWW'09: Proc. of the 18th International Conference on World wide Web*, pages 341–350, New York, NY, USA, 2009. ACM.

[35] T. Yeh, T. H. Chang, and R. C. Miller. Sikuli: Using GUI Screenshots for Search and Automation. In *Proc. of the 22nd Symposium on User interface Software and Technology*, pages 1–10, 2009. ACM.

[36] T. Yeh and B. Katz. Searching Documentation using Text, OCR, and Image. In *Proc. of the 32nd SIGIR,* pages 776–777, 2009. ACM.