Crawling for Images on the WWW

Junghoo Cho[1] and Sougata Mukherjea[2]

[1] Department of Computer Science, Stanford University,
Palo Alto, Ca 94305, USA
E-mail: chocs.stanford.edu
[2] C&C Research Lab, NEC USA,
110 Rio Robles, San Jose, CA 95134, USA
E-mail: sougataccrl.sj.nec.com

Abstract. Search engines are useful because they allow the user to find information of interest from the World-Wide Web. These engines use a crawler to gather information from Web sites. However, with the explosive growth of the World-Wide Web it is not possible for any crawler to gather all the information available. Therefore, an efficient crawler tries to only gather important and popular information. In this paper we discuss a crawler that uses various heuristics to find sections of the WWW that are rich sources of images. This crawler is designed for AMORE, a Web search engine that allows the user to retrieve images from the Web by specifying relevant keywords or a similar image.

Keywords: World-Wide Web, Crawling, Site-based Sampling, Non-icon detection.

1 Introduction

Search engines are some of the most popular sites on the World-Wide Web. However, most of the search engines today are textual; given one or more keywords they can retrieve Web documents that have those keywords. Since many Web pages have images, effective image search engines for the Web are required. There are two major ways to search for an image. The user can specify an image and the search engine can retrieve images similar to it. The user can also specify keywords and all images relevant to the user specified keywords can be retrieved. Over the last two years we have developed an image search engine called the **Advanced Multimedia Oriented Retrieval Engine (AMORE)** [5] *(http://www.ccrl.com/amore)* that allows the retrieval of WWW images using both the techniques. The user can specify keywords to retrieve relevant images or can specify an image to retrieve similar images.

Like any search engine we need to crawl the WWW and gather images. With the explosive growth of the Web it is obviously not possible to gather all the WWW images. The crawlers run on machines that have limited storage capacity, and may be unable to index all the gathered data. Currently, the Web contains more than 1.5 TB, and is growing rapidly, so it is reasonable to expect that most machines cannot cope with all the data. In fact a recent study has shown that the major text search engines cover only a small section of the Web [3]. The problem is magnified in an image search engine since image indexing takes more time and storage.

Therefore the crawler should be "intelligent" and only crawl sections of the WWW that are rich sources of images. In this paper we present the AMORE crawler and explain several heuristics that can be used to determine WWW sections containing images of interest. The next section cites related work. Section 3 gives an overview of the AMORE system. Section 4 explains the crawler architecture. Section 5 discusses the heuristics used by the crawler. Finally section 6 concludes the paper with suggestions of future work.

2 Related Work

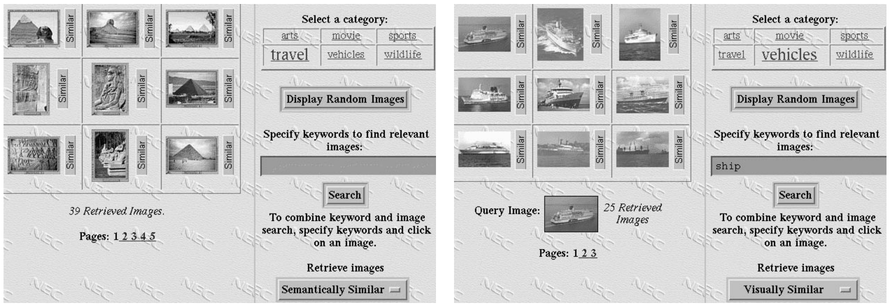
Crawlers are widely used today. Crawlers for the major search engines, for example, Alta Vista (*http://www.altavista.com*) and Excite (*http://www.excite.com*) attempt to visit most text pages, in order to build content indexes. At the other end of the spectrum, we have personal crawlers that scan for pages of interest to a particular user, in order to build a fast access cache (e.g. NetAttache *http://www.tympani.com/products/NAPro/NAPro.html*).

Roughly, a crawler starts off with the URL for an initial page. It retrieves the page, extracts any URLs in it, and adds them to a queue of URLs to be scanned. Then the crawler gets URLs from the queue (in some order), and repeats the process [6]. [1] looks at the problem of how the crawler should select URLs to scan from its queue of known URLs. To ensure that the crawler selects important pages first, the paper suggests metrics like backlink count and page rank to determine the importance of a WWW page. Instead of finding the overall importance of a page, in this paper we are interested in the importance of a page with respect to images.

Another research area relevant to this paper is the development of customizable crawlers. An example is SPHINX [4], a Java toolkit and interactive development environment for Web crawlers which allows site-specific crawling rules to be encapsulated.

3 AMORE Overview

During indexing the AMORE crawler, discussed in the next section, gathers "interesting" Web pages. The images contained and referred to in these pages are downloaded and the **Content-Oriented Image Retrieval (COIR)** library [2] is used to index these images using image processing techniques. We also use various heuristics, after parsing the HTML pages, to assign relevant keywords



39 Retrieved Images.

Pages: 1 2 3 4 5

Specify keywords to find relevant images:

To combine keyword and image search, specify keywords and click on an image.

Retrieve images: Semantically Similar

Query Image:

25 Retrieved Images

Pages: 1 2 3

Specify keywords to find relevant images: ship

To combine keyword and image search, specify keywords and click on an image.

Retrieve images: Visually Similar

(a) Semantic Similarity Search with a picture of Egypt

(b) Integrated Search with the keyword *ship* and the picture of a ship

Fig. 1. Examples of different kinds of AMORE searches

to the images and create keyword indices.

During searching, AMORE allows the user to retrieve images using various techniques. Figure 1 shows some retrieval scenarios. The user can specify keywords to retrieve relevant images. The user can also click on a picture and retrieve similar images. The user has the option of specifying whether the similarity is semantic or visual. For semantic similarity, the keywords assigned to the images are used. If two images have many common keywords assigned, they are considered to be similar. Thus in Figure 1(a) images of Egypt are retrieved even though they are not visually similar. For visual similarity, the COIR library is used. It looks at features of the images like color, shape and texture to determine similarity using the image indices. AMORE also allows the integration of keyword search and similarity search. Thus Figure 1(b) shows images visually similar to the picture of a ship that are also relevant to the keyword *ship*.

4 AMORE Crawler

The design of AMORE image crawler embodies two goals that we pursue. First, the crawler should crawl the web as widely as possible. More precisely, we want the crawler to visit a significant number of the existing Web sites. If the crawling is performed only to a small set of sites, the scope and the number of images crawled may be limited and biased.

Second, the crawler should not waste much of its resource examining "uninteresting" parts of the Web. For now, the information on the Web is mostly textual, and only a small portion of the Web contains images worthy of being indexed. The crawler should not waste its resource trying to crawl mostly textual parts of the Web.

Note that these two goals are conflicting. On one hand, we want to gather images from as many sites as possible, which means that the crawler should visit a significant portion of the web. On the other hand, we want to limit the scope of the crawler only to the "interesting" sections. We tried to achieve these two conflicting goals by *site-based sampling* approach, which will be discussed next.

4.1 Architecture of the crawler

The crawler of AMORE consists of two different sub crawlers: Explorer and Analyzer. Informally, Explorer discovers "interesting sites" and Analyzer filters out "uninteresting" sections from the identified sites. Figure 2 represents the data flow between these two crawlers.

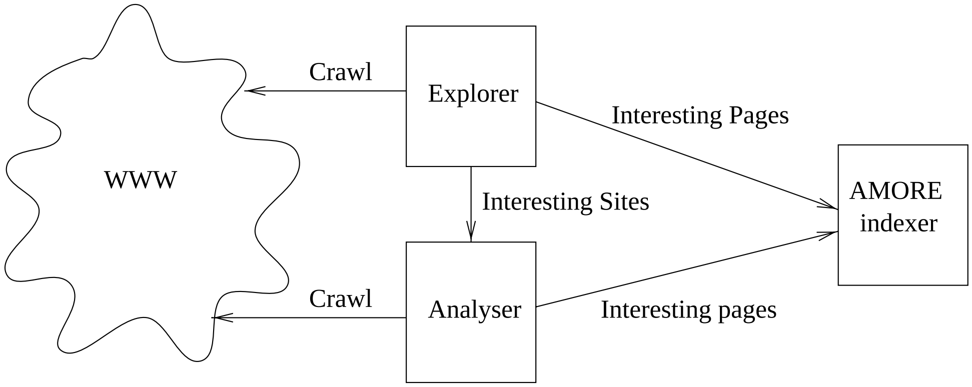


Fig. 2. The architecture of the AMORE crawler.

- **Explorer**
  Explorer is the big scale crawler whose main job is to discover "interesting" sites from the web. It is optimized to find as many interesting sites as possible, and therefore it tries to visit the web widely but shallowly. More precisely, it differs from most web crawlers in that it only crawls k sample pages for each and every site it found. After sampling k pages from a site, it checks the sample pages to see how many non-icon images the pages contains or refers. (The criteria for icon detection is described in section 5.1 in detail). If more than $r\%$ of pages have more than one non-icon image, then the site is considered "interesting". The Analyzer works on these interesting sites that the Explorer found. Note that even if a site is not found to be interesting, the interesting pages in the site are sent to the AMORE indexer.

This allows AMORE to index images from a large number of Web sites.

- **Analyzer**
  Analyzer is the small scale crawler whose main job is to identify "interesting" sections from a web site. The input to Analyzer are the "interesting" sites that Explorer found. For each input site, the Analyzer performs more crawling to gather m ($>> k$) sample pages. These sampled pages are then analyzed to evaluate the directories in the site. For each directory, we calculate its *importance* as discussed in in section 5.2. Then the Analyzer crawls the directories in the order of their importance. The Analyzer examines all directories whose importance is greater than a threshold.

Note that our two step crawling approach is conceptually similar to *iterative deepening* [7]. Informally, we expand all high level nodes (crawl root level pages of each web site), and we go into deeper (perform more crawling) for the interesting nodes expanded.

Also note that there are various parameters in the crawling process like the number of pages to be sampled by the Explorer and the threshold value for importance of the directories in the Analyzer. The AMORE administrator can set these values based on the resource constraints.

5 Heuristics

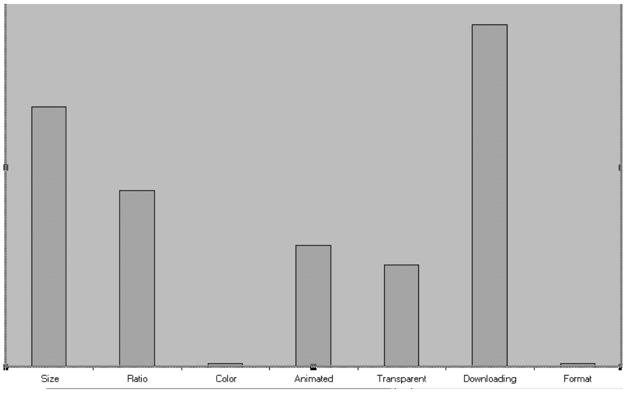


Fig. 3. Comparing the reasons why images referred to in HTML files were not indexed by AMORE.

5.1 Removing icon images

The Web is well-known for its heterogeneity of information. The heterogeneity is also true for images, and different types of images coexist on the Web. At one extreme, a small icon is used as the bullet of a bulleted list and at the other extreme, a page embeds a 1024x768 gif image of Gogh's painting.

We believe the images on the Web can be classified into two categories: icons and authentic images. Icons are the images whose main function is enhance the ”look” of a web page. They can be substituted by a symbol (e.g. bullets) or by text (e.g. advertizing banners), but they are used to make the page more presentable. In contrast to icons, authentic images are the images that cannot be replaced by non-images. We cannot substitute the image of Gogh's painting or the picture of Michael Jordan with text without losing information that we want to deliver. An usability study of AMORE has also shown that people were not interested in the icons when they are using a WWW image retrieval engine.

It is generally difficult to identify icons without analyzing the semantic meaning of an image. However, our experiments show that the following heuristics work reasonably well for icon detection:

- **Size:** We remove very small images such as dots which are generally used for HTML page beautification. We only extract images that are more than a certain size (generally > 2000) and have a certain width and height.
- **Ratio:** We don't extract images if their width is much greater or smaller ($>$ 3 or $< 1/3$) than their height. This filters out the headings and banners that appear at the top and the sides of many Web pages.
- **Color:** We also remove color images if they have very few colors ($<$5). This removes uninteresting computer generated logos, etc.
- **Animated:** Surprisingly, most users were also not interested in animated gifs! So they are also removed.
- **Transparent:** We also remove transparent images since they are generally used as headings and logos.

About 10% of the images gathered by AMORE were not indexed because they were uninteresting (by the above criteria) or they could not be downloaded to our site (for example, some images were missing even though they were referenced in HTML pages) or they were not the right format (like most browsers AMORE supports only gif and jpeg). Figure 3 is a chart showing the various reasons for which an image was not indexed.

5.2 Finding interesting directories

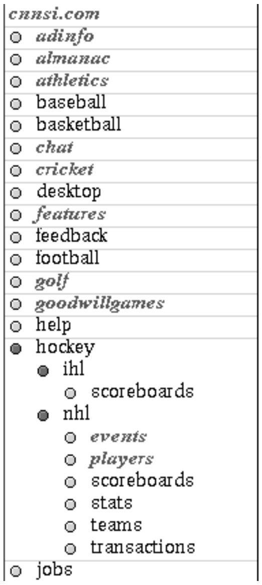
An analysis of the major WWW sites have shown that they are well organized; all pages dealing with the same subject are generally organized in the same directory. It is also true that if a sample of the pages in a directory have images, a majority of the pages are “interesting” from our point of view. These observations are utilized by the Analyzer when it tries to find sections of interest in large Web sites.

We use various heuristics to find interesting directories:

- In many Web sites images are kept in directories with relevant names. For example, directories like *http://www.nba.com/finals97/gallery/*, *http://www.si.edu/natzoo/photos/* and *http://www.indiabollywood.com/gallery/* contain good images. Individuals also organize their site so that the images are kept in directories with meaningful names. For example, we found good images in *http://fermat.stmarys-ca.edu/∼jpolos/photos* and *http://www.mindspring.com/∼zoonet/galleries*. Therefore, we have a list of keywords (like *gallery, photo, images*) and if a directory has any of these words, they are considered interesting and crawled in detail.
- However, not all interesting directories may have meaningful names. For example, *http://cbs.sportsline.com/b/allsport/* has many images. Therefore, for most directories more analysis is necessary.

  We calculate the importance of a page as $i + 1/w$ where i is the number of non-icon images in the page and w is the number of words (excluding HTML tags). This makes the importance of a page with a lot of text and one image less important than a page with one image and fewer text.

  The overall importance of a directory is the average of the importance of the sampled pages of the directory. Since the analyzer crawls the directories in order of their importance, image intensive pages will be gathered first and if there are resource constraints, the AMORE administrator can stop the crawling of a site after sometimes. At present, directories whose importance is greater than a pre-defined threshold are considered interesting.



cnnsi.com
- ○ *adinfo*
- ○ *almanac*
- ○ *athletics*
- ○ baseball
- ○ basketball
- ○ *chat*
- ○ *cricket*
- ○ desktop
- ○ *features*
- ○ feedback
- ○ football
- ○ *golf*
- ○ *goodwillgames*
- ○ help
- ● hockey
  - ● ihl
    - ○ scoreboards
  - ● nhl
    - ○ *events*
    - ○ *players*
    - ○ scoreboards
    - ○ stats
    - ○ teams
    - ○ transactions
- ○ jobs

Fig. 4. An evaluation of our directory importance heuristic for *http://cnnsi.com*. The interesting directories are shown in *italics*.

To determine if our heuristics are correct, we have built a visual evaluation interface. Figure 4 shows the interface. Here we are evaluating our heuristics for the *http://cnnsi.com* Web site. The site is represented as a tree and the interesting directories are shown in italics and red color. It is seen that directories like *almanac* (URL: *http://cnnsi.com/almanac*) and *features* are found interesting while directories like *jobs* and *help* are found uninteresting. For a large directory, even if the whole directory is uninteresting, several subdirectories may be found to be interesting. For example, on exploring the *hockey* directory using our evaluation interface, we find that the directories *events* and *players* are found to be interesting while directories like *scoreboards* and *stats* are not. On examining the Web site, we found that the directory importance heuristics performed upto our expectation.

6 Conclusion

In this paper we presented the crawler for the AMORE WWW Image Search Engine. The crawler uses several heuristics to crawl at least some pages of all Web sites as well as the "interesting" sections of "interesting" Web sites. This allows the AMORE crawler to achieve the conflicting goals of gathering as many "interesting" images as possible by visiting as few sites as possible.

In the future we are planning to do an extensive evaluation of the crawler and extend the technique to other media like video and audio. Our ultimate objective is to develop an effective Multimedia WWW Search engine.

References

1. J. Cho, H. Garcia-Molina, and L. Page. Efficient Crawling through URL ordering. *Computer Networks and ISDN Systems. Special Issue on the Seventh International World-Wide Web Conference, Brisbane, Australia*, 30(1-7):161–172, April 1998.
2. K. Hirata, Y. Hara, N. Shibata, and F. Hirabayashi. Media-based Navigation for Hypermedia Systems. In *Proceedings of ACM Hypertext '93 Conference*, pages 159–173, Seattle, WA, November 1993.
3. S. Lawrence and C. Giles. Searching the World-Wide Web. *Science*, 280(5360):98, 1998.
4. R. Miller and K. Bharat. SPHINX: a framework for creating personal, site-specific Web crawlers . *Computer Networks and ISDN Systems. Special Issue on the Seventh International World-Wide Web Conference, Brisbane, Australia*, 30(1-7):119–130, April 1998.
5. S. Mukherjea, K. Hirata, and Y. Hara. Towards a Multimedia World-Wide Web Information Retrieval Engine. In *Proceedings of the Sixth International World-Wide Web Conference*, pages 177–188, Santa Clara, CA, April 1997.
6. B. Pinkerton. Finding what People Want: Experiences with the WebCrawler. In *Proceedings of the First International World-Wide Web Conference*, Geneva, Switzerland, May 1994.
7. S. Russell and P. Norvig. *Artificial Intelligence: A Morden Approach.* Prentice Hall, 1995.