Information Retrieval Based on Image Detection on Web Pages

Hazem M. El-Bakry
Faculty of Computer Science & Information Systems, Mansoura University, EGYPT
E-mail: helbakry20@yahoo.com

Nikos Mastorakis
Department of Computer Science, Military Institutions of University Education (MIUE) - Hellenic Naval Academy, Greece

Abstract: The main objective of Internet users is to find the required information with high efficiency and effectiveness. Finding information on an object’s visual features is useful when specific keywords for the object are not known. Since mobile agent technology is expected to be a promising technology for information retrieval, there is a number of mobile agent based-information retrieval approaches have been proposed in recent years. In this paper a new approach for image-based information retrieval using mobile agents is presented. Multiple information agents continuously traverse the Internet and collect images that are subsequently indexed based on image information such as the URL location, size, type and the date of indexation. In the search phase, the mobile agent receives the image of object as a query and searches the set of web pages that contain information about the object by matching the query to images on web pages.

Keywords: information retrieval, content-based image retrieval, image clustering, and mobile agent

I. Introduction

With the development of Internet technology, the fast growing World Wide Web has become one of the most important sources of information and knowledge. When searching the Web, a user can be overwhelmed by thousands of results retrieved by a search engine, of which few are valuable. The problem for search engines is not only to find topic relevant results, but results consistent with the user’s information need. How to retrieve desired information from the Internet with high efficiency and good effectiveness is become the main concern of internet user-based [1]. Users generally find information using search engines. The input to these search engines often consists of keywords or other written text like a question. But the Web contains not only text, but also information in other modalities such as images. They are however not yet being used as input for general Web searches. When the user wants to query about an object that has unique visual features, the image of the object that represents the visual properties can be used as a query. By enabling searches on object appearance provide users with a new, more convenient and direct means for finding information about objects encountered in everyday life. Many existing world wide search engines, for example Google and Yahoo, rely on meta-data (annotations) or the context in which the image is found and the query is performed using text-based information retrieval method, the matching is performed based on image captions or file names, thus the performance of image retrieval is based on the similarity between the user’s text query and image text annotation not on the image content itself [2]. Content Based Image Retrieval (CBIR) searches for images by using asset of visual features such as color, shape and texture, are extracted from the images that characterize the image content [3]. These techniques have been used in many areas, such as geographic information system, biomedical image processing and digital libraries [4]. One of the main advantages of the CBIR approach is the possibility of an automatic retrieval process, instead of the traditional keyword-based approach, which usually requires very difficult and time-consuming previous annotation of database images. Comparatively speaking, CBIR more objectively reflects the content of images [6]. Most of CBIR systems are only operate on a local demo database of a few thousand images stored at the host web site [7]. Unlike image retrieval from a fixed database, where each image is treated as an independent object, for Searching for image on the Web basically comes down to locating an appropriate Web page and to retrieve relevant information about the image from that site [8,9].

Mobile agent is a program that traverses the Internet, moves to different sites with different characteristics, searches for the desired information, and returns the search report to the user query [10,11]. When large quantities of data are stored at distributed sites, moving the computations to the data is a more realistic and feasible approach, compared with migrating data to the computations. In other words; instead of gathering information from distributed sites to be processed at a central site, users can dispatch mobile agents to a destination site to perform information retrieval and to return to a user the results of the analysis. Thus, the data transmitted over the network are minimized. This is especially important when using a low-bandwidth access network. Features of mobile agent, such as cross-platform migrating, dynamic and distributed computing make it possible to solve the problems of low bandwidth and unstable connection in the internet and to provide the support of searching in the internet via inconsistent connected devices (e.g. laptop, PDA,
mobile phone, etc). Mobile agents already have been used to build many distributed applications, including distributed information retrieval [12], and network management [13]. Furthermore, they provide an attractive paradigm for design and implementation of scalable, flexible, and extensible system for image retrieval from multiple, distributed heterogeneous image sources [14].

In this paper, an image-based information retrieval using mobile agent is presented. Image crawler continuously traverses the Internet and collect images that are subsequently indexed based on integrated feature vectors. These features along with additional information such as the URL location and the date of index procedure are stored in a database. The user can access and search these indexed contents through the Web with an advanced and user friendly interface. The output of the system is a set of links to the content available in the WWW, ranked according to their similarity to the submitted image by the user. The paper is organized as follows: Section II describes related work. Section III presents the architecture of the proposed system. Section IV illustrates Experiments and Evaluation of the proposed system. Conclusion and future work are presented in section V.

There are many systems for Content–based image retrieval (CBIR) using mobile agent are presented e.g. [5, 15, 16]. None of these systems provides a web search engine; they are based on local image database. Comparing to image databases, the Web is an unlimited, immense repository of images, covering much broader resources, and is increasing continuously at high speed rate. In [17, 18] mobile agent is used for web information retrieval based on text query not image query. Unfortunately, keywords may not accurately or completely describe image content.

Two approaches for information retrieval from distributed database based on mobile agents are presented in [12]. The first approach utilizes the mobility of agent for moving the query to the desired site where the data resided while the second one is based on reduction of the number of migrating agents. The work in this paper presents a new approach for finding information related to an image from the World Wide Web by matching image itself and retrieving the web page which contains information related to the submitted image. This is called image-based information retrieval. The information retrieved can be in any format or media, but it’s found using an image as a query.

II. System Architecture

The overall system is split into two parts: (i) the off-line part and (ii) the on-line or user part.

In the off-line part, several Information Crawlers, continuously traverse the WWW, collect images and store them to a database for further processing. Then the image indexing module processes the image in order to extract descriptive feature using color and shape features. These features along with information of the images such as URL, date of process, size and a thumbnail are stored in the database.

In the on-line part, a user connects to the system through a common Web Browser using the HTTP protocol. The user can then submit queries either by example images or by sketch. The query then is processed by the server and the retrieval phase begins; the indexing module is repeated again for the submitted image and then the extracted features are matched against those stored in the database. The results containing the URL, as well as, the thumbnail of the similar images are transmitted to the user and the results are ranked according to their similarity to the submitted image.

A) Information Crawler

The image collection process is conducted by a different autonomous Web agent or Crawler run on a network of computers. The agent traverses the Web by parsing the hyperlinks, detects images, retrieves and transfers them for processing to the system server. The system server extracts features from the images and then stores it in the database.

The overall collection process is illustrated in Fig. 1; it is carried out using several distinct modules:

* The Traversal Crawler - assembles lists of candidate Web pages that may include images or hyperlinks to them.

* The Hyperlink Parser - extracts the URLs of the images.

* The Retrieval Crawler - retrieves and transfers the image to the system server for further processing.

This section gives a description of the information crawler as shown in Fig. 1; Traversal crawler browses the WWW to collect images. The web-sites listed in the categories Shopping and Recreation of the DMOZ directory [21] are chosen as a starting point. In the first phase, the Traversal Crawler follows a breadth first search across the Web. It retrieves pages via Hypertext Transfer Protocol (HTTP) and passes the Hypertext Markup Language (HTML) code to the Hyperlink Parser. In turn, the Hyperlink Parser detects new URLs, encoded as HTML hyperlinks, and adds them back to the queue of Web pages to be retrieved by the Traversal Crawler. The Hyperlink Parser detects the hyperlinks in the Web documents and converts the relative URLs to absolute addresses. By examining the types of the hyperlinks and the file extensions of the URLs, the Hyperlink Parser extracts the URLs of the images. In the second phase, the list of image URLs from the Hyperlink Parser is passed to the Retrieval Crawler. The Retrieval Crawler retrieves the images and provides them as input to the indexing module. After the indexing procedure, the extracted features are added to the database. Another important function of the Retrieval Crawler is to extract attributes
associated with the image such as URL, date of processing, size, width, height, file size, type of visual data, and so forth, and also generate a thumbnail icon, that sufficiently compacts and represents the visual information.

**B) Indexing Module**

After the image detection module collects and transfers images to the server, image indexing module processes them in order to extract low level feature of image. This module consists of two steps; feature extraction and image clustering.

1. Feature Extraction

The prototype implementation uses Color Coherence Vectors [19] as a feature extraction and comparison algorithm. Feature vectors consist of 128 float values; each vector is computed as follows: the image is blurred using a simple $3 \times 3$ convolution filter which averages the color values of all horizontal and vertical neighbors of the filtered pixel. The blurred image is then quantized to a color space of 64 colors. In the last step, the pixels of the image are classified into coherent and incoherent pixels. Coherent pixels are pixels which are part of a horizontally and vertically connected pixel area of the same color whose size exceeds a certain threshold. Incoherent pixels are pixels which are not coherent pixels. For each of the 64 colors, the coherent and incoherent pixel counts are summed up separately and normalized with regard to the total image area. This results in a 128 dimensional vector. The Color Coherence Vector algorithm has the advantage that it is easy to implement, reasonably fast, and achieves a high compression rate. Images are reduced to a vector whose encoding is less than 600 bytes.

2. Clustering Similar Image

The increase of number of images in the database, the increase in the time required to retrieve results. To avoid search over the whole images database, the images with similar features vector should be grouped together in the same class. K-means clustering algorithm [20] is used for this operation because it is quite fast, simple and has been applied and shown to be useful in many applications. The basic step of k-means clustering is simple. At the beginning, numbers of clusters $K$ are determined and random feature vectors are selected as the initial centroids of these clusters. The grouping is done by minimizing the sum of squares of distances between feature vector and the corresponding cluster centroid.

**C) Image Query Module.**

The image query module includes methods for extracting features from query images, classifying query image, and determining similarity between features extracted from the query and those stored in the database.

1. Query Feature Extraction

When a user initiates a query for information about the image, the **Feature Extractor** module performs feature extraction on the example image and sends the feature vector, contained in a query message, to the broker server.

2. Query Classification

Once the broker server receives the query, it will calculate the distances from the Query-image feature vector to the centroids of the set of each cluster to determine which clusters are closest to the query. The minimum square error is used to obtain the distance between a Query Image feature vector and the centroid vector of the cluster, which can be described as:

$$
\text{Sim} \left( \tilde{C}, \tilde{Q} \right) = \text{Min} \left( \sum_{i=1}^{k} \left\| \tilde{Q} - \tilde{C} \right\| \right) 
$$

3. Distance Measure

In content-based image retrieval, a distance measure is usually used to check similarity or dissimilarity between two images. The $L1$ distance is taken as a measure of similarity between two color coherence vectors.

Let $\left( h_1, \tilde{h}_1 \right), \ldots, \left( h_n, \tilde{h}_n \right)$ be a color coherence vector where $h_i$ is the percentage of coherent pixel of color $i$ and $\tilde{h}_i$ is the percentage of incoherent pixels of color $i$, then the $L1$ distance is defined as:

$$
\left\| h - \tilde{h} \right\| = \sum_{i=1}^{n} \left( | h_i - \tilde{h}_i | \right)
$$

**III. Experimental Results**

According to the experiments results, it has been observed that a single-threaded information crawler can traverse the web gathering images at an average rate of one image every 82 seconds. Gathering the images has the potential to take long time with a single-threaded information crawler on a single computer. This problem is address by employing a parallel information crawler distributed across many machines (image servers). Experiments indicate that the framework can allow using 16 information crawlers to collect over one million images monthly.

The system was developed using c#.net on Pentium 4 3.2GHz CPU (2GB RAM) running on Windows vista platform. For image related operations, the Matlab image processing toolbox was used to assist in extracting visual features. During experiments, different port numbers are used to simulate different image servers. Fig.2, shows an overview and example result from the system: the image of the object is selected. The object image is used to query the database to find the most relevant object, which in turn retrieves the relevant web pages for the user.

**A) Implementation Models**

1. Centralized-Broker Approach:
The architecture is capable of gathering, indexing, caching, replicating, and accessing Internet images in a distributed way, which assures scalability. This model consists of a central image broker, several image servers, index agents, search agents, and fetch agents (all agents are mobile and relocate during their life cycle). The image broker dispatches index agents which transport feature extraction algorithms to one or more image servers. On these servers, the index agents extract relevant feature vectors from local images and send or take image entries back to the image broker. At the image broker, all image entries are merged into the central index. Each image entry consists of a feature vector, the URL pointing to the host where the image was retrieved, an image ID that uniquely identifies the image at the server, an optional thumbnail, and optional further information on the image such as its size. The globally unique ID of an image consists of the globally unique URL plus the locally unique image ID. Based on the index data, image brokers can either serve requests in a client/server fashion, or they support mobile agent queries as follows; A client sends a search agent to the broker which queries information related to a specific image, a sketch, or a feature vector which is extracted from either of these query images. The query result consists of extended image entries which contain the normalized distance between the query and the entry’s feature vector, in addition to, the entry itself. The search agent transports the result set back to the client who selects images for retrieval based on the included thumbnails. Once an image is selected for retrieval, the client sends a fetch agent that migrates between image servers to collect information about the query image. An advantage is that this information can then be sent to Brokers in a compressed form, which reduces the network load. Another advantage is that one indexer agent can send the collected data to many Brokers, saving indexing costs. Finally, a third advantage is that the server load is reduced since indexer agents run at the image server, which contributes to the scalability of the system. A Broker can retrieve indexing information from many indexers to build an index of widely distributed data. The Broker can also retrieve indexing information from other Brokers, which gives the Broker an indexed view of the other Brokers.

2. The Broadcasting Approach

In this model, one index agent per image server is dispatched and takes residence at the image server to extract features from the images and sets up an index directly at the image server. The index agent may also monitor the local image repository for changes, and it can update its index accordingly and incrementally. The only communication between the broker and the index agent is a short notice that the computation of the index is completed and the index agent is ready to provide service to search agents. The image broker is still a central point of access but it resembles more a yellow page server. It refers search agents to the image servers where index agents reside.

Once the client selects an image for retrieval, the process continues as in the gatherer model. Based on its index the index agent serves queries of search agents which visit the image server. On each image server, the search agent merges previously collected results with the results of its local search, and prunes the overall number of results.

B) Clustering Approach

In this model, each index Agent creates metadata (centroid of clusters) describing its own image collection and sends it to broker server. The broker collects the metadata from each index agent. This broker can route every query to the appropriate image server which contains images belongs to the category of query image. At each image server, the search agent matches the query image with the image collection on the appropriate image server and sends related web pages which contain the related information back to the user.

C) Performance Measure

In order to decide the best model for designing the framework, a comparative analysis is performed using two important parameters that influence the performance of the model. These parameters are; size of intermediate data transfer and number of comparisons required answering the query. The response time, i.e., time elapsed between a user initiating a request and receiving the results, as the metric for performance measure. This includes the time taken for agent creation, time taken to visit mediator server to determine list of image servers to be visited and processing time at each image server to obtain the required images.

_response_time: time elapsed between a user initiating a request and receiving the results,

\[
 RT = Y + \frac{X}{DBR}
\]

Where: \( RT \): Response Time (sec.), \( Y \): Communication Initialization Time and Client Processing Time (sec.), \( DBR \): Data Bit Rate (Network Data Transmission speed: bytes/sec.), and \( X \): Amount of transmitted data (bytes).

\[
 X = (\text{Mobile Agents Transfer}) + (\text{images descriptor Transfer})
\]

Each image descriptor consists a thumbnail, the ID of that image (unique within the domain of the image server), a measure of similarity to the original query image, and the URL of the image server from which the full size of image can be retrieved.

The Response Time can be written as

\[
 RT = \text{time (migrating mobile agents)} + \text{time (Total Access Delay)} + \text{time (Transferred images descriptors)}
\]
The increase in the number of comparisons which are required to be performed to respond query will increase the response time. Moreover, distributing this task between different image servers overcome problems of centralized server and make all hosts participating in the system act as servers. Fig. 4, shows the number of comparison in broadcasting and clustering approaches.

3) The Response Time

Number of visited servers during query processing varies at Broadcasting and Clustering approaches. For the broadcasting approach, the search agent will visit all servers in the network in order to find the results for the query so number of servers that will be visited is equal to the total number of servers in the network. But in clustering approach, number of visited servers is only that having the same category as the query image. By calculating total access delay of the system which is used to implement these approaches, it is clear that by allowing search agent to visit all servers, total access delay increase proportionally. Fig. 5, shows the effect of increasing number of image servers on access delay in the two cases. Total access delay is the summation of access delay time of all image servers while access delay time in clustering is the access delay of servers only that maintain images belonging to the category of the query. This can be concluded as fewer numbers of migrations of mobile agents used to execute the tasks will cause lower network traffic and consume less bandwidth, and the total time taken to retrieve the query results is minimized.

V. Conclusion

According to the work in this paper, an approach for image-based information retrieval using mobile agent is introduced. The proposed approach is based on using different web crawlers to collect images from the World Wide Web located on different servers; consequently similar images at each image server are clustered together. The mobile agents travel across the image servers and analyze image databases at their resource to perform feature extraction, clustering similar images at each node together to speed up the retrieving process. The advantages of the image-based information retrieval using mobile agents are a natural parallelization of the computation and the reduction of bandwidth by avoiding the transfer of huge intermediate data through the network.

The image based information retrieval system was tested by three approaches; central indexing, broadcasting and clustering approach. These were subsequently simulated to quantify their effectiveness in the information retrieval function. It is concluded that the three approaches did not differ much in discovering the requisite information, but differ in the size of intermediate data transfer and the response time to the user’s query. Future work will conclude
the routing optimization of search agent in the system. Search agent does not have to visit image servers by their order list; instead, it can determine its route using an optimization technique. The future work also will concern on investigating the performance of the proposed system using other clustering algorithms and other distance measures.

References


Figure 1: Information Crawler Components

Figure 2: Image-based information retrieval.

Figure 3: Size of intermediate data vs. Increase in image categories
Figure 4: Number of comparison in clustering and broadcasting approaches

Figure 5: The total access delay vs. number of image servers