A FRAMEWORK FOR VISUAL INFORMATION RETRIEVAL ON THE WEB

Adil ALPKOÇAK **Esen ÖZKARAHAN**

Dokuz Eylül University
Department of Computer Engineering
İzmir / TURKEY
{alpkocak,esen}@cs.deu.edu.tr

Abstract: *Today on the Internet there is a wide variety of text based search engines, however the same is not true for searching visual information placed in Internet Web pages. There is increased activity and research for querying such databases especially for content based visual querying. The heterogeneous, distributed and transient nature of visual information, lack of interoperable retrieval systems and the limited bandwidth of Web environment presents bottlenecks for such efforts. In this study the difficulties of visual information retrieval on the Web are highlighted and a visual information retrieval system in such an environment is presented.*

Keywords: Content-based image retrieval, Web searching, image indexing and searching, image retrieval, digital libraries, distributed information retrieval.

1. Introduction

Information resources available on the Internet constitute a universal digital library. Such a library includes books, journals, reference volumes, newspapers, telephone directories, sound and voice recordings, images, video clips, and scientific data all accessible electronically.

Search engines on the Web are important tools for making use of digital libraries. Presently there are an abundance of text based search engines. As more and more information becomes available on the Internet the nature of information is becoming multimedia oriented and visual information constitutes bulk of such information. This brings the need for visual search engines and especially those that can query visual data by content.

2. Visual Information Retrieval Systems

The human perception mechanism is equipped with an amazing system for recognizing an infinite number of shapes, colors, patterns, textures, objects and backgrounds. The mechanics of such capabilities are of course not fully understood.

A visual document has similar ingredients of the human environment i.e., it has features such as color, line, region, corners and textures. A full functional visual information retrieval system provides means to store, organize, add and delete images and search those images by content.

Recently a few commercial visual search engines have become available such as QBIC system of IBM and Visual Information Retrieval (VIR) cartridge of Oracle. Putting such systems to the use of visual search engines for Internet requires further effort.

3. Visual Information Retrieval Search on Web

A large number of catalogs and search engines index documents on the WorldWide Web (WWW). For example, recent systems such as lycos, Altavista, InfoSeek and Yahoo, index the documents by their textual content. These systems periodically scan the Web, record the text on each page and through processes of automated analysis and/or (semi-)automated classification, condense the Web into compact searchable indexes. However, no tools are currently available for searching images. This absence is particularly notable given the highly visual and graphical nature of the Web.

Visual information on the Web is either embedded in documents or is present as standalone objects. The visual information is in the form of images, graphics, bitmaps, animations and videos.

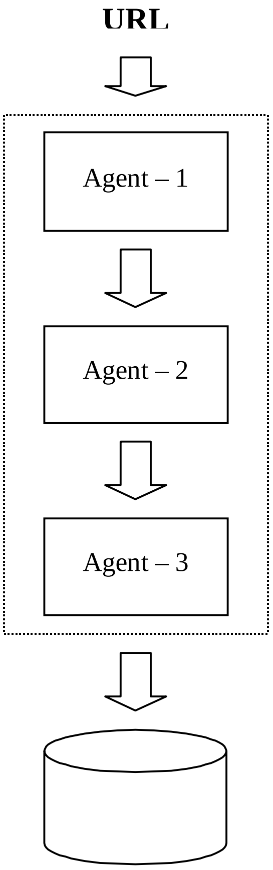
3.1 Collection Processes

One major task of a visual search engine is to scan the web and acquire the visual data which is then catalogued. The data to be catalogued is collected by a series of automated agents that traverse the Web and detect visual data. The result is several indexes for the visual data based upon

- Visual features
- Textual descriptions
- Text for the whole document

The term “documents” is used in the context of an HTML document, which may contain or link text, graphics, animation, still images and sound.

The visual data collection process takes place in three phases. The first phase starts with an input for the root URL of the web site whose content will be acquired and indexed. The overall collection process is illustrated in Figure 1. The agent in the first phase retrieves HTTP document identified by the given URL. After a successful retrieval, the document is passed to the second phase. The aim of the second phase is to analyze and parse the given document. After processing the document, if a link tag is recognized within the document, this new link is propagated backward to the first agent and added to the URL queue if it passes a series of controls. To stop the unpredictable growth of the hyperlink tree, a number indicating the maximum height of the tree can be set, or any of the links going outside of the root URL are eliminated. In a sense, this is very similar to many of the conventional spiders or robots that follow hyperlinks across the Web [SC96].



Root URL is given as an input

Traverses all documents within the current WEB tree of given URL

Parses the documents, detects visual data and retrieves them.

Analyzes all the data, extracts visual features and updates indexes and builds thumbnails for preview.

At the end of the processes, all the information required for content based retrieval is acquired.

Figure-1. Visual data collection process on Web

Agent-2 in the above flow can detect visual as well as non-visual related information by scanning the visual HTML tags for building indexes. Table 1 shows HTML tags used to indicate visual content.

Table-1: Some well-known HTML tags regarding visual content.

Description	HTML Tag
Backround Image	`<body background=URL …>`
Inlined Image	`<img src=URL …>`
Standalone object	`<a href=URL … >`

All the descriptions of documents containing visual data are, firstly, kept in a list to determine if visual data appears more than once, which is quite common in case of company logos and bullet images of list items. If they exist, duplication is removed from the list, to assure that each image appears in the list only once. Afterwards, the whole list is passed to phase 3. Figure 2 shows a more detailed view of the visual data collection process envisioned for our proposed Web-based visual information retrieval framework.

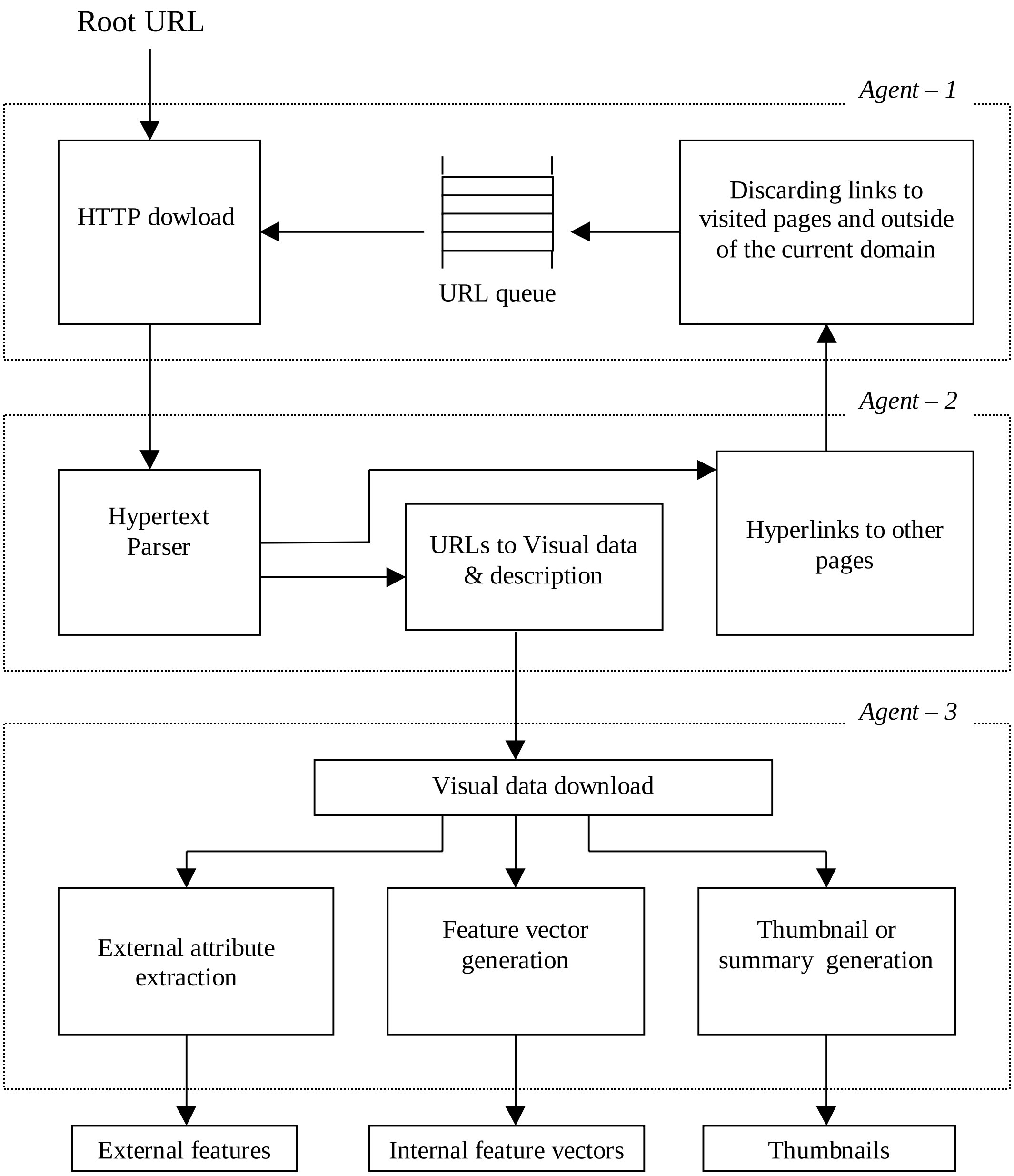

Root URL
Agent – 1
HTTP dowload
URL queue
Discarding links to visited pages and outside of the current domain
Agent – 2
Hypertext Parser
URLs to Visual data & description
Hyperlinks to other pages
Agent – 3
Visual data download
External attribute extraction
Feature vector generation
Thumbnail or summary generation
External features
Internal feature vectors
Thumbnails

Figure-2. Gathering visual data from web using software agents

3.2 Detection of Visual Data From the Hypertext Documents

Images are published on the Web in two forms: inlined and referenced. The HTML syntax differs in these two cases. To inline, or embed, an image in a HTML document, the following code is included in the document:

```
<img src=URL alt=[alternate text]>
```

where URL gives the relative or absolute address of an image or video. The optional `alt` tag specifies the text that may appear in place of the image or video when browser is loading the image or has trouble finding or displaying the visual data.

Alternatively images and videos may be referenced from parent Web pages using the following code:

```
<a href=URL>[hyperlink text]</a>
```

where the optional [hyperlink text] provides the highlighted text that describes the objects pointed to by the hyperlink. In this case, the image is regarded as an individual object identified by the URL and detection of visual data is performed by MIME settings.

3.3 Representation of Visual Information

In phase 3 of the flow in Figure 2, all the visual data is retrieved, processed and catalogued. The three important functions of agent 3 are to

- Extract visual features that allow for content-based searching,
- Extract external subject attributes such as width, height, format of the file etc.
- Generate thumbnails for visual objects.

4. Detecting Textual Descriptions for Visual Information

Utilization of text is essential for the cataloguing process. In particular, every image on the Web has a unique Web address and possibly other HTML tags, which provide for valuable interpretation of the visual information. We use the Web addresses and alt tag of the images as a source of textual description of images. The search terms are extracted from the image and video URLs, alt tags and hyperlink text by chopping the text at non-alpha characters. The URL of an image or video has the following form

```
http://host.site.domain[:port]/[directory/][file[.extension]]
```

where brackets ('[' and ']') denote optional arguments. Using a set of string manipulation functions, textual descriptions are obtained from the `[directory/][file[.extension]]` part of the URL. For example, several typical URLs can be in the following form:

```
http://www.kodak.com/animals/wild-life/bear2.gif
```

Terms extracted from the directory and file string are "wild", "life" and "bear". This process assumes that meaningful names are assigned to images. However, it may easily fail if numbers are used instead of names. After extracting terms for visual objects as external subjective attributes of textual description, text-based information retrieval techniques can be used.

5. Database Design

Using all the visual information gathered visual features are extracted and indexed. Our database is Oracle 8.0 database with visual information retrieval (VIR) extension. The features extracted in the indexing processes are local and global color, texture and sketches. Each retrieved image is processed as described above, and the following tables are populated:

```
IMAGES (IMAGE_ID, URL, NAME, FORMAT, WIDTH, HEIGHT)
FEATURES (IMAGE_ID, ICON, DESCRIPTION, FEATURES)
```

Where special data-types are given as follows:

```
DESCRIPTION  ∈   {free form character}
ICON         ∈   {BLOB containing the thumbnail/icon of the
                  image}
FEATURES     ∈   {BLOB containing visual features extracted
                  from the original image}
```

Standard SQL can be used for querying the table IMAGES. However, content-based queries involving the table FEATURE require special processing involving spatial similarities of feature vectors.

6. Prototypes for Visual Query Interfaces

The most important task in a visual information retrieval is query interfacing. Due to the nature of multimedia data, multimedia information is hard to describe. Standard SQL would be insufficient for this purpose. An image in a query would not be completely represented for querying. A fully functional visual information retrieval system must support user interfaces that can enable user to easily express his (her) query [AA97].

Visual queries can be grouped into tree broad categories: *Expressive queries*, *navigational queries* and *query by example* [AA97]. Expressive queries are those where user can specify what (s)he wants via proper interfaces. Conventional SQL a is typical expressive query language. However, specifying the requirement is not a trivial task for querying images. The indexed visual image features are used in query interfacing. The typical visual features are local and global color, texture and sketch. For global color distribution, a simple interface where user can specify the color and its percentage in the resulting images is sufficient. In global color distribution, location of the colors in the image layout is not important. The user may query an image containing 50% red and 50% blue colors. The retrieved result can contain an image with its top half is blue and bottom half is red or vice versa. In the case of local color, however, the location of the colors is important. For this feature, several kinds of user interfaces can be provided. A simple grid box can be presented to the user to choose and fill the colors from a given palette. Alternatively, user interfaces can be a set of tools where user can draw rectangles, move them, and fill with proper colors.

In forming an expressive query for texture, it is quite unrealistic to expect the user to draw the texture (s)he wants. Instead, all the textures extracted from the database can be classified into clusters and a representative texture for each cluster can be chosen. These representative textures can be presented to the user and asked to choose one which (s)he believes is closest to that of the requested image. In this way, textures can be used for expressive querying. Although several researchers have been working on texture classification, none of them is aiming to use texture classification for expressive querying. It is said that information is meaningful only when it can be retrieved through an expressive query [GS+97]. Searching images with keywords also falls into expressive category, because the keywords are specified by the user.

Navigational queries use a navigational mode of search in which the next query is based upon the result of the previous query. While the task of searching is to locate an image from its partial specification, a browser starts with little user specification. The browser assumes that the exploring user has an unexpected mental model of the images of interest. Suppose the user is looking for an image that "best suits the theme of a product to be advertised". Stated as such, the query is far too ill-specified and given a large image collection, the user is likely to lose interest in his search endeavor soon, unless the system can navigate him to his interest subspace within a small number of steps. Thus initially browser behaves as a random sampler of the database. *Query by navigation* is most effective if it can smoothly guide the user interaction to a potential search-set without the user having to make a conscious choice of query parameters until he or she is in a position to start querying. In other words, an important aspect of the browser design is to make an informed sampling based on an incrementally constructed model of the user's needs. Once the user reaches an identifiable superset of the images of interest, he or she can instantly switch to search and continue with more specific query specification.

A somewhat different notion of a browser is that of an incremental query refinement mechanism. In this case, the browser starts with the results of a search, and lets the user change query parameters by adding (or removing) restrictions on the search conditions, changing relative weights on a "search by example feedback" (a method by which a user specifies how much a specific item suits the information need). The system then partially reevaluates the query given the previous results and the new criteria.

Query by example is the most commonly used method for multimedia retrieval. This type of query uses a different paradigm, where the system constructs the query parameters from an example image i.e., referred to as *like-this* query. If the example is given from outside of database population, this paradigm faces additional issues to address as the user no longer mentions attribute values requiring human involvement for perception. In order to search for query, the system has to perform a task which is similar to one that is done at the time of database population. It has capabilities to extract only internal attributes which is a small part of the whole attribute set of the image. Within this context, query is highly incomplete since the object recognition and computer vision state of the art are still poor to comprehend the contents of multimedia data. In the future, when we can extract most of the semantics automatically this paradigm can produce acceptable retrieval.

Secondly, if the example image is chosen from the database, assuming that labor intensive perception tasks have already done that before, the query processor can use the pre-extracted attribute values of the given example. In a multimedia query, a weight should be assigned to each attribute value. The system has to have a set of features as comparison axes for color, texture etc. against which two images can be "matched". As the query is not precisely specified, there is no "perfect match" for the query image in this case. Therefore, the “matching” is on the basis of a overall similarity ranking made up of similarity measures for each comparison axis.

7. Distributed Visual Information Retrieval

Although visual information retrieval systems are still only in their infancy, it is important to understand the challenge in developing scalable solutions. A scalable Internet retrieval solution needs to solve three main problems: heterogeneity, complexity and bandwidth [CS+97].

Heterogeneity: Heterogeneity has three major aspects: *format*, *meta-data* and *system level heterogeneity*. Today a large number of image formats are used and a standardized format is necessary. Meta-data, which is about labeling or descriptions for visual data must also be standardized. Such standardization will enhance interoperability among different retrieval systems and improve the effectiveness of individual systems. While standards are developed for meta-data of text documents, efforts such as CNI/OCLC Metadata have been made to develop metadata for images [WM96].

The third aspect of heterogeneity is about systems used to retrieve visual data. The ultimate goal of distributed visual information retrieval is to obtain integrated search or meta-search engines. Meta-search engines serve as common gateways linking users to multiple cooperative or competetive search engines. A number of successful examples of meta-search engines for text based information retrieval are available. However, there are several problems that need to be solved for visual information retrieval. The first problem is about the requirement for a distributed database query protocol, such as Z39.50 which is used in information retrieval. Different systems use almost the same feature set to index visual data, such as color percentage, color layout, texture and shape. But there is no uniformity among them about color spaces, distance metrics and indexing methods, and scoring. Some of them have special functionalities. All of this heterogeneity on target search engines makes distributed integrated search more difficult. Developing distributed visual database query protocol would be helpful in abstracting this kind of heterogeneity.

Complexity: The process of searching for visual information is complex. For example, a user may not know what (s)he are looking for or how to describe content. Some of the searching tools available today try to solve this problem by creating a static cataloguing hierarchy. However, this is not a true or acceptable solution for a dynamic environment where scores of images are inserted or deleted.

Online visual information retrieval systems solve this problem by providing mechanisms for browsing and ways to express visual properties of a query. Also, a feedback mechanism is supported for the user to be able to preview the initial results and provide relevance feedback as to the relevance of the returned item to refine the query. Especially in the Internet environment, previews of query results reduce the amount of transmission required for full-resolution images.

Bandwidth: Slow Internet connection is another major barrier faced by the visual information retrieval systems on Web. Although image compression reduces the amount of bandwidth required by these systems, it still limits their performance. The download time clearly influences user interaction with Internet visual information retrieval.

8. Conclusion

In this work, visual information retrieval on Internet is studied, current issues are highlighted and a framework for a web based visual information retrieval search engine is designed. The Internet contains a growing amount of information, which is intensive in visual content. We use the term information for both textual and visual data. Although many textual search engines are available in the Internet community, a few of them is capable to search visual data by content. However, the data ocean of Internet web pages contains vast amount of visual data and full functional Internet search engines must provide mechanisms to search visual data by content efficiently.

Visual information retrieval needs high processing power, wide transmission bandwidth and large and fast storage devices. The best way to fulfill all of these requirements is to combine all search engines in a meta search engine. However, visual search engines have other problems. The first problem is the slow Internet bandwidth in delivering visual data. The other problems are due to heterogeneity of data formats in distributed visual retrieval systems. This necessitates having a protocol such as Z39.50 for building meta search engines.

An implementation of the system presented here is under way. After completion of this work, integration of this system with other visual search engines is planned to obtain a distributed visual retrieval system.

References

[SC96] John R. Smith and Shih-Fu Chang, "Searching for Images and Videos on the World Wide Web, Columbia University, Dept. of Electrical Engineering, Technical Report #459-96-25, 1996.

[CS+97] Shih-Fu Chang, John R. Smith, Mandis Beigi and Ana Benitez, "Visual Information Retrieval From Large Distributed Online Repositories", *Communications of ACM*, Vol.40, No.12, 1997.

[GS+97] Amarnath Gupta, Simone Santini and Ramesh Jain, "In Search of Information in Visual Media", *Communications of ACM*, Vol.40, No.12, 1997.

[AA97] Adil Alpkoçak, "A Parallel Multimedia Information Representation and Retrieval System", Ph.D. Dissertation, Ege University CS Dept., 1997.

[WM96] Stuart L. Weibel, Eric J. Miller, "Image Description on the Internet", *Summary of CNI/OCLC Image Metadata Workshop*, Dublin, 1996. Available at http://www.oclc.org/oclc/publications/review96/image.htm