Chapter 5

CONTENT-BASED MULTIMEDIA RETRIEVAL ON THE INTERNET

Oge Marques, Susan Fallon, and Borko Furht

Abstract

In this chapter we first discuss the motivation behind content-based multimedia retrieval systems. A complete overview of a Content-Based Visual Information Retrieval (CBVIR) system follows, from the user's perspective to the complex calculations that take place behind the scenes. The specifics of Web-based CBVIR systems are discussed next. Finally, we compile a list of currently available commercial systems and prototypes that illustrate the concepts and applications discussed in this chapter.

1. INTRODUCTION

In this section we describe the motivation and main challenges behind multimedia retrieval on the Internet.

1.1 THE GROWTH OF MULTIMEDIA AND THE NEED FOR BETTER SEARCH AND RETRIEVAL TECHNIQUES

The amount of audiovisual information available in digital format has grown exponentially in recent years. Gigabytes of new images, audio and video clips are generated and stored everyday, helping to build up a huge, distributed, mostly unstructured repository of multimedia information, much of which can be accessed through the Internet.

Digitization, compression, and archival of multimedia information has become popular, inexpensive and straightforward, and there is a broad range of available hardware and software to support these tasks. Subsequent retrieval of the stored information, however, might require considerable additional work in order to be effective and efficient.

There are basically three ways of retrieving previously stored multimedia data:

1. **Free browsing**: the user browses through a collection of images, audio, and video files, and stops when she finds the desired information.

2. **Text-based retrieval**: the user adds textual information (metadata) to the audiovisual files during the cataloguing stage. In the retrieval phase, this additional information is used to guide conventional, text-based query and search engines to find the desired data.

3. **Content-based retrieval**: the user searches the multimedia repository providing information about the actual contents of the image, audio, or video clip. A content-based search engine translates this information in some way as to query the database and retrieve the candidates that are more likely to satisfy the user's request.

The first two methods have serious limitations and scalability problems. Free browsing is only acceptable for the occasional user and cannot be extended to users who frequently need to retrieve specific multimedia information for professional applications. It is a tedious, inefficient, and time-consuming process and it becomes completely impractical for large databases.

Text-based retrieval has two big problems associated with the cataloguing phase:

(a) the considerable amount of time and effort needed to manually annotate each individual image or clip; and
(b) the imprecision associated with the subjective human perception of the contents being annotated.

These two problems are aggravated when the multimedia collection gets bigger and may be the cause of unrecoverable errors in later retrieval.

In order to overcome the inefficiencies and limitations of text-based retrieval of previously annotated multimedia data, many researchers, mostly from the Image Processing and Computer Vision community, started to investigate possible ways of retrieving multimedia information – particularly images and video clips – based solely on its contents. In other words, instead of being manually annotated using keywords, images and video clips would be indexed by their own visual content, such as color, texture, objects' shape and movement, among others. Research in the field of Content-Based Visual Information Retrieval (CBVIR) started in the early 1990's and is likely to continue during the next decade. Many research groups in leading universities and companies are actively working in the area and a fairly large number of prototypes and commercial products are already available.

In this chapter we examine the state of the art, ongoing research, examples, and open issues in designing Content-Based Visual Information Retrieval (CBVIR) systems to access multimedia information distributed over the Internet.

1.2 RESEARCH CHALLENGES BEHIND THE DESIGN OF CBVIR SYSTEMS

The design of CBVIR systems brings up many interesting problems and challenges, some of which are summarized in [1]. An important subset of these problems is related to the way human beings perceive visual information and how much the knowledge of the human visual perception can help translating it into a set of features, rules, and criteria for comparing and selecting images. An example of a problem that falls into this category is the understanding of how human beings express their (subjective) understanding of the contents of an image using words, drawings, sketches, or similar images. A closely related problem is to understand how human beings perceive visual similarities. These and other similar problems have a great impact on the design of contemporary CBVIR systems.

A good CBVIR system will be one whose design is guided by the following open questions:

(1) How to minimize the "semantic gap" between the low-level features that are automatically extracted from the visual contents of an image and the human interpretation of such contents?
(2) How to make it possible to users to express their queries in a way that allows them to clearly specify which image or type of image they are interested in?
(3) How to minimize the impact of human subjectivity in the performance of the system?
(4) How to measure the similarity between two images?

Attempting to provide meaningful and satisfactory answers to the questions above will possibly impact the design of almost every functional block of a CBVIR system. Some of the major decisions that might be influenced by these questions include:

(a) The choice of image features and their mapping into semantic contents.
(b) The decision on which type of query and retrieval will be supported: visual-based, text-based, or both.
(c) The inclusion of user feedback to improve the system's performance.

2. ANATOMY OF CBVIR SYSTEMS

In this section we describe the main aspects of CBVIR systems, from the user interface to the feature extraction and similarity measurement stages.

2.1 A TYPICAL CBVIR SYSTEM ARCHITECTURE

Figure 1 shows a block diagram of a generic CBVIR system, whose main blocks are:

- **User interface**: friendly GUI that allows the user to interactively query the database, browse the results, and view the selected images / video clips.
- **Query / search engine**: responsible for searching the database according to the parameters provided by the user.
- **Digital image and video archive**: repository of digitized, compressed images and video clips.
- **Visual summaries**: representation of image and video contents in a concise way, such as thumbnails for images or keyframes for video sequences.
- **Indexes**: pointers to images or video segments.
- **Digitization and compression**: hardware and software necessary to convert images and videos into digital compressed format.
- **Cataloguing**: process of extracting features from the raw images and videos and building the corresponding indexes.

Digitization and compression have become fairly simple tasks thanks to the wide range of hardware and software available. In many cases, images and videos are generated and stored directly in digital compressed format. The cataloguing stage is responsible for extracting features from the visual contents of the images and video clips. In the particular case of video, the original video segment is broken down into smaller pieces, called scenes, which are further subdivided into shots. Each meaningful video unit is indexed and a corresponding visual summary, typically a key frame, is stored. In the case of images the equivalent process could be object segmentation, which just a few systems implement. In either case, the cataloguing stage is also where metadata gets added to the visual contents. Manually adding metadata to image and video files is mandatory for text-based visual information retrieval systems. CBVIR systems, however, typically rely on minimum amount of metadata or none at all.

Digitization, compression, and cataloguing typically happen off-line. Once these three steps have been performed, the database contains the images and videos themselves, possible simplified representations of each file or segment, and a collection of indexes that act as pointers to the corresponding images or video segments.

The online interaction between a user and a CBVIR system is represented on the upper half of the diagram in Figure 1. The user expresses her query using a GUI. That query is translated and a search engine looks for the index that corresponds to the desired image or video. The results are sent back to the user in a way that allows easy browsing, viewing, and possible refinement of the query based on the partial results.

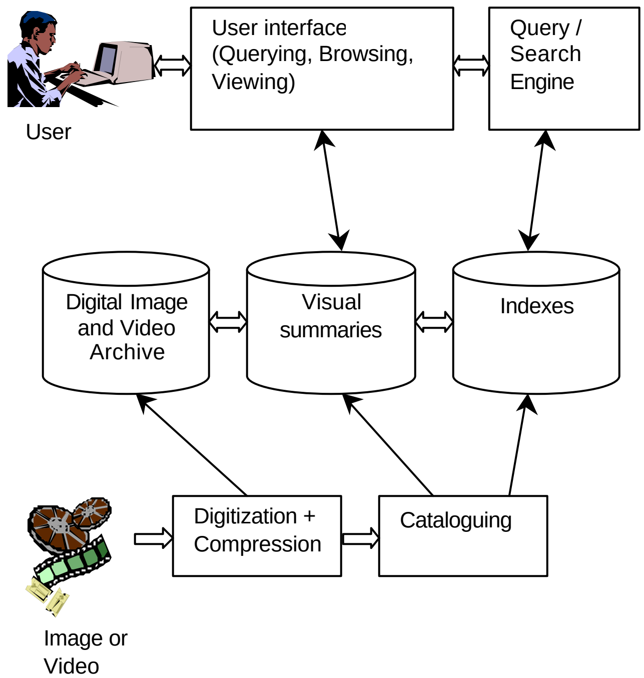


Figure 1. Block diagram of a CBVIR system.

2.2 THE USER'S PERSPECTIVE

The user interface is a crucial component of a CBVIR system. Ideally such interface should be simple, easy, friendly, functional, and customizable. It should provide integrated browsing, viewing, searching, and querying capabilities in a clear and intuitive way. This integration is extremely important, since it is very likely that the user will not always stick to the best match found by the query/search engine. More often than not the user will want to check the first few best matches, browse through them, preview their contents, refine her query, and eventually retrieve the desired video segment.

Searching the visual database contents should be made possible in several different ways, either alone or combined. For instance a user should be able to perform a pictorial query, e.g. querying by similarity (using another video clip or a static image as a reference) and a query by keyword simultaneously.

Query options should be made as simple, intuitive and close to human perception of similarity as possible. Users are more likely to prefer a system that offers the “Show me more video clips that look similar to this (image or video)" option, rather than a sophisticated interactive tool to edit that video shot key-frame's histogram and perform a new search. While the latter approach might be useful for experienced technical users with image processing knowledge, it does not apply to the average user and therefore has limited usefulness. An ideal CBVIR system query stage would hide the technical complexity of the query process from the end user. A search through visual media should be as imprecise as "I know it when I see it." [3]

2.3 FEATURE EXTRACTION

CBVIR systems should be able to automatically extract visual features that are used to describe the contents of an image or video clip. Examples of such features include color, texture, size, shape, and motion information. In specific contexts the process of feature extraction can be enhanced and/or adapted to detect other, specialized attributes, such as human faces or objects. Because of perception subjectivity, there is no best representation for a given feature [4]. The color information, for instance, can be represented using different color models (e.g., RGB, HSV, YCbCr) and mathematical constructs, such as color histograms [5], color moments [6], color sets [7,8], color coherence vectors [9], or color correlograms [10]. In a similar way, texture can be represented using co-occurrence matrix [11], Tamura texture features [12] or Wavelets [13-15], to name just a few.

2.4 HIGH-DIMENSIONAL INDEXING AND SIMILARITY MEASUREMENTS

The extracted features are grouped into some suitable data structure or mathematical construct (e.g., a normalized feature vector), and suitable metrics (e.g., Euclidean distance) are used to measure the similarity between an image and any other image. At this stage, the main challenges are the high dimensionality of the feature vectors (typically of the order of 10^2) and the limitations of Euclidean similarity measure, which although mathematically elegant might not effectively simulate human visual perception [4].

Solutions to the high dimensional indexing problem include reducing the dimension of the feature vectors and the use of efficient multi-dimensional indexing techniques. Dimension reduction is typically obtained using either the Karhunen-Loeve Transform or clustering techniques. Examples of multi-dimensional indexing techniques include specialized data structures (e.g., k-d tree, R-tree and its variants). To overcome the limitations of Euclidean similarity measures, researchers have proposed the use of clustering and neural networks.

2.5 THE SEMANTIC GAP

The human perception of visual contents is strongly associated to high-level, semantic information about the scene. Current Computer Vision techniques work at a lower level (as low as individual pixels). CBVIR systems that rely on low-level features only can answer queries such as:

- Find all images that have 30% of red, 10% of orange and 60% of white pixels, where orange is defined as having a mean value of red = 255, green = 130, and blue = 0.
- Find all images that have a blue sky above a green grass.
- Find all images that are rotated versions of this particular image.

In general case, the user is looking for higher-level semantic features of the desired image, such as "a beautiful rose garden", "a batter hitting a baseball", or "an expensive sports car". There is no easy or direct mapping between the low-level features and the high-level concepts. The distance between these two worlds is normally known as "semantic gap."

There are basically two ways of minimizing the semantic gap. The first consists of adding as much metadata as possible to the images, which was already discussed and shown to be impractical. The second suggests the use of rich user interaction with relevance feedback combined with learning algorithms to make the system understand and learn the semantic context of a query operation.

2.6 USER'S RELEVANCE FEEDBACK

Early attempts in the field of CBVIR aimed at fully automated, open-loop systems. It was hoped that current Computer Vision and Image Processing techniques would be enough for image search and retrieval. The modest success rates experienced by such systems encouraged researchers to try a different approach, emphasizing interactivity and explicitly including the human user in the loop. A clear example of this shift can be seen in the work of MIT Media Lab researchers in this field, when they moved from the "automated" Photobook [16] to the "interactive" FourEyes [17].

The expression "relevance feedback" has been used to describe the process by which a system gathers information from its users about the relevance of features, images, image regions, or partial retrieval results obtained so far. Such feedback might be provided in many different ways and each system might use it in a particular manner to improve its performance. The effect of relevance feedback is to "move" the query in the direction of relevant images and away from the non-relevant ones [18]. Relevance feedback has been used in contemporary CBVIR systems, such as MIT's FourEyes [17], UIUC's MARS [19-24], and NEC's PicHunter [25-29], among many others.

In CBVIR systems that support relevance feedback a search typically consists of a query followed by repeated user feedback, where the user comments on the items that were retrieved. The use of relevance feedback makes the user interactions with the system simpler and more natural. By selecting images, image regions, and/or image features, the user is in one way or another telling the system what she wants without the burden of having to describe it using sketches or keywords, for instance.

There are many ways of using the information provided by the user interactions and refining the subsequent retrieval results of a CBVIR system. One approach concentrates on the query phase and attempts to use the information provided by relevance feedback to refine the queries. Another option is to use relevance feedback information to modify feature weights, such as in the MARS project [19-24]. A third idea is to use relevance feedback to construct new features on the fly, as exemplified by [30]. A fourth possibility is to use the relevance feedback information to update the probability of each image in a database being the target image, in other words, to predict the goal image given the user's interactions with the system. The latter is the approach taken by Cox et al. [25-29] in the PicHunter project.

3. WEB-BASED CBVIR SYSTEMS

In this section we describe research and design issues specific of Web-based CBVIR systems and present a case study on a particular image metasearch engine, MetaSEEk.

3.1 ISSUES IN DESIGNING A WEB-BASED CBVIR SYSTEM

Making a CBVIR system accessible through the Internet, particularly on the Web, extends its usefulness to users anywhere in the world at the expense of new design constraints, which are summarized below [2]:

- Visual information on the Web is highly distributed, minimally indexed, and schema-less. Internet-based CBVIR systems lack a well-defined schema or consistent metadata.
- The query and retrieval stages have no control over the cataloguing process and must rely on possible metadata stored in HTML tags associated with the images and video clips.
- In order to keep the query response time below a tolerable limit (typically two seconds), the number of visual features used for comparison and matching has to be kept low.
- The user interface should work with reduced-size images and videos until the final stage, when the user issues an explicit request.
- The use of content-based query methods may be deferred until a stage where the scope of the search has been reduced to a specific semantic category, selected by the user.

3.2 A CASE STUDY: COLUMBIA UNIVERSITY'S METASEEK

Researchers at Columbia University have a developed a prototype content-based metasearch engine for images, MetaSEEk (available at **http://www.ctr.columbia.edu/metaseek/**). The system contains three main components:

- the *query dispatcher*, which selects target search engines for each query;
- the *query translator,* which translates the user-specified query into a format understood by the target search engine; and
- the *display interface*, which merges the query results from all the target search engines and displays the best candidates on the screen.

MetaSEEk currently works with four target search engines, whose internal characteristics are hidden from the end user: IBM's QBIC, Columbia's VisualSEEk and WebSEEk, and Virage.

The user interface supports random browsing, content-based retrieval, or keyword-based retrieval. Content-based retrieval is possible either selecting a sample image from one of the supported databases or typing in the URL to an external image. MetaSEEk uses two visual features for content-based retrieval: color and/or texture. Several other parameters can be specified by the user, e.g., the number of search options to be searched simultaneously, the maximum waiting time (in seconds), the number of images to be displayed, and the category the user is interested in (the default category is "All").

Once a query has been received, MetaSEEk's dispatcher selects the target engines and search options to be used. For content-based visual queries the selection of target engines is based on past performance of each target search engine for each query image. Performance data is stored in form of scores in a performance database. If the user selects a query image that has never been used before, MetaSEEk downloads the image and matches it to the corresponding clustering structure to obtain a list of the most similar clusters. Images from the closest clusters are selected and presented to the user. Based on the average performance scores of the cluster selected by the user, MetaSEEk will then choose a suitable search engine. Moreover, the new query image is added to the performance database for future queries.

MetaSEEk uses the K-means clustering algorithm for grouping visually similar images into clusters, Tamura algorithm for detection of texture-related features, and color histograms for detection of color-related attributes. Two feature vectors are compared using Euclidean distance.

MetaSEEk allows users to search for images under a specific semantic category, e.g., flowers, people, or animals. Each category has its own performance database and clustering structure. MetaSEEk's database is hierarchical: images are first classified according to their semantic meaning (specified by the user), then grouped according to the visual features used (color, texture, or both), and finally clustered in classes for each particular visual feature.

Once the results are returned from each individual search engine, the MetaSEEk display component selects, organizes, and presents them back to the user. For random or keyword-based queries the results are presented in random order. For content-based queries, the results are sorted according to a combination of the relevance score informed by the target search engine and the performance metrics of the search engines themselves, stored in MetaSEEK's database.

The system also incorporates user relevance feedback. Once the best candidates have been displayed, the user can click on checkboxes indicating whether they like or dislike a particular result image. This information is used to update the performance metric of the corresponding search engine.

Ongoing work on MetaSEEk has been focusing on the following research issues [2]:

- More sophisticated approaches to user feedback and performance monitoring
- Support for customized search, e.g., color layout, and color percentages.
- Increase on the number of visual features and automatic or semiautomatic visual feature selection based on user preferences
- Faster clustering techniques
- Use of machine learning techniques to boost performance

4. EXAMPLES OF CBVIR SYSTEMS

Numerous CBVIR systems, both commercial and research, have been developed in recent years. From the point of view of the end user, most systems support one or more of the following options:

- Random browsing
- Query by example
- Query by sketch
- Text-based query
- Navigation with customized categories

Systematic studies involving actual users in practical applications still need to be done to explore the tradeoffs among the different options listed above [4]. Some of the currently available CBVIR systems are described below. Several of them are surveyed and explored in more detail in Section 5.

QBIC
QBIC (Query By Image Content) [32-34] was the first commercial CBVIR system. It was developed by IBM Almaden Research Center. Its framework and techniques have influenced many later systems. QBIC supports queries based on example images, user-constructed sketches, and selected colors and texture patterns. In its most recent version, it allows text-based keyword search to be combined with content-based similarity search. The online QBIC demo can be found at: **http://wwwqbic.almaden.ibm.com.**

Photobook
Photobook [16] is a set of interactive tools for browsing and searching images developed at MIT Media Lab. Photobook consists of three sub-books, from which shape, texture, and face features are extracted respectively. Users can query the system based on features from each of

the three sub-blocks. Additional information about Photobook can be found at: **http://www-white.media.mit.edu/vismod/demos/photobook/index.html.**

FourEyes

FourEyes [17] is an improved version of Photobook that includes user relevance feedback. Given a set of positive and negative examples, it decides upon which models or combinations of models to use and learns which combinations work best for solving particular types of problems. When presented with a new problem similar to one it has solved before, FourEyes can solve it more quickly than it could the first time. More details about the system can be found at: **http://www-white.media.mit.edu/vismod/demos/photobook/foureyes/.**

Netra

Netra is a prototype CBVIR system developed in the UCSB Alexandria Digital Library (ADL) project [35]. It uses color, shape, texture, and spatial location information in the segmented image regions to search and retrieve similar images from the database. An online demo is available at **http://vivaldi.ece.ucsb.edu/Netra/.**

MARS

MARS (Multimedia Analysis and Retrieval System) [19-24] was originally developed at University of Illinois at Urbana-Champaign. The main focus of MARS is not on finding a single "best" feature representation, but rather on how to organize the various visual features into a meaningful retrieval architecture, which can dynamically adapt to different applications and different users. MARS formally proposes a relevance feedback architecture in Image Retrieval and integrates such technique at various levels during retrieval, including query vector refinement, automatic matching tool selection, and automatic feature adaptation. More information about MARS can be obtained at: **http://www-db.ics.uci.edu/pages/research/mars.shtml.**

PicToSeek

PicToSeek [18] is an image search engine developed at University of Amsterdam. PicToSeek uses autonomous Web crawlers to collect images on the Web. Then, the collected images are automatically catalogued and classified into predefined classes and their relevant features are extracted. The users can query PicToSeek using image features, an example image, or simply browsing the precomputed image catalog. A demo version of PicToSeek is available at: **http://www.wins.uva.nl/research/isis/zomax/.**

VisualSEEk

VisualSEEk [36,37] is part of a family of CBVIR systems developed at Columbia University. It supports queries based on both visual features and their spatial relationships. An online demo is available at: **http://www.ctr.columbia.edu/VisualSEEk/.**

PicHunter

PicHunter [25-29] is a CBVIR system developed at NEC Research Institute, New Jersey. PicHunter uses relevance feedback and Bayes's rule to predict the goal image given the users' actions.

Virage

Virage [38] is a commercial content-based image search engine developed at Virage, Inc. Virage supports queries based on color, composition (color layout), texture, and structure (object boundary information) in any arbitrary combination. The users inform the system which weight should be associated with each atomic feature according to their own emphasis. More information about Virage products can be found at: **http://www.virage.com.**

Visual RetrievalWare

Visual RetrievalWare is a CBVIR engine developed by Excalibur Technologies Corp. [39]. Similarly to Virage, it allows combinations of several visual query features, whose weights are specified by the users. More information about Excalibur products can be found at: **http://www.excalib.com.**

Other systems

There are several other CBVIR systems available on the Web, such as Blobworld [40], Webseer [41], and ImageRover [42], among many others. Some of them are examined in more detail in the next Section.

5. SURVEY OF CBVIR SYSTEMS

In this section we explore some specifics of selected CBVIR systems.

ImageRover

ImageRover [42,48] is a CBVIR system developed by Boston University that is currently available as an online demo version. This is a Web-based tool, which gathers information about HTML pages via a fleet of automated robots. These robots gather, process, and store the image metadata in a vector format that is searched when a user queries the system. The user then receives relevance feedback with thumbnail images, and by selecting the relevant images to their search, can utilize the content-based searching capabilities of the system until they find their desired target image. The demo is located at **http://www.cs.bu.edu/groups/ivc/ImageRover/demo.html/.**

Figure 2 shows the basic setup of the HTML collection system utilized by ImageRover. There are two types of "modules" or robots, which gather and process the metadata: the gathering and digestion modules. The gathering modules recursively parse and traverse WWW documents, collecting the images as they go. The digestion modules compute image metadata over 6 subimages of the given image. Submodules compute the following data for each subimage:

- Color analysis via color histograms
- Texture orientation analysis using steerable pyramids and orientation histograms
- Text analysis via weighted word frequency histograms.

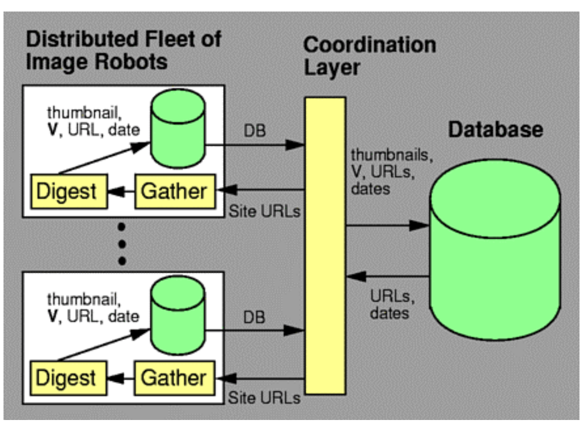


Figure 2. Diagram of the HTML document collection subsystem.

Figure 3 shows the query subsystem for the program, based on a client-server architecture. The system gives relevance feedback that selects L-m Minkowski distance metrics on the fly, allowing the user to perform queries by example on more than one image and successive iteration produces the desired result.

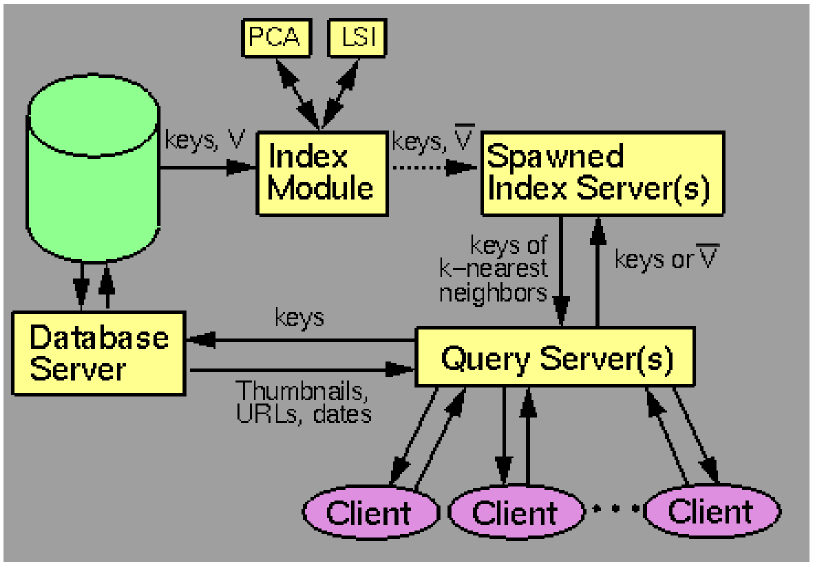


Figure 3. Diagram of the query server subsystem.

WebSeek

WebSeek [50] is similar to ImageRover in its HTML collection processes through Web robots, though it has the advantage of video search and collection as well. It was developed at Columbia University, and currently has a working demo available on the Web at **http://www.ctr.columbia.edu/webseek/**. The user receives relevance feedback in the form of thumbnail images and motion icons or spatially and temporally reduced video forms given as short *gif* files to the user.

Figure 4 illustrates the process of gathering of images and videos. Three types of Web robots utilized by the system are (a) the traversal spider, which assembles lists of candidate Web documents that may include images, videos, or hyperlinks to them, (b) the hyperlink parser, which extracts the Web addresses or URLs of the images and videos, and (c) the content spider, which retrieves, analyzes, and iconifies the images and videos.

Figure 5 shows how the images and videos are detected, and Figure 6 shows how each image or video is processed. WebSeek classifies the images/videos by their URLs and HTML tags with term extraction, directory name extraction, automated key-term to subject mapping using the key-term dictionary and semi-automated directory name to subject mapping. It then places these classifications in unique categories to be searched through textually, along with the image metadata for its content based searching which utilizes color histogram similarity. The user can even adjust the color histograms of the images to reiterate a search for a better match as shown in Figure 10.

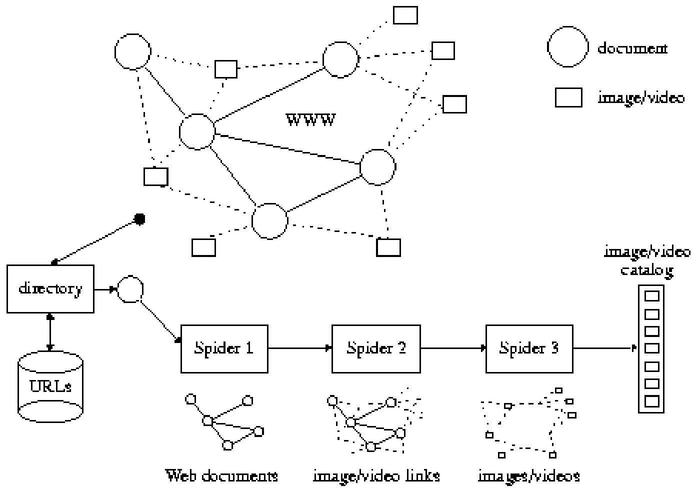


Figure 4. Image and video gathering process in WebSeek.

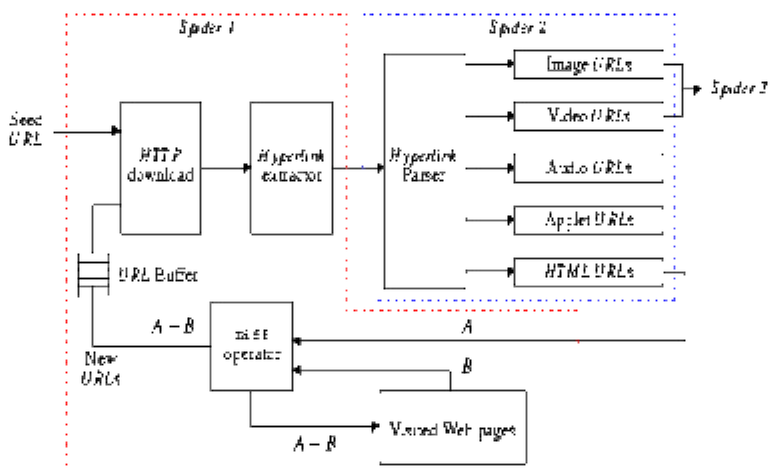


Figure 5. List assembly process of URLs of the images and videos.

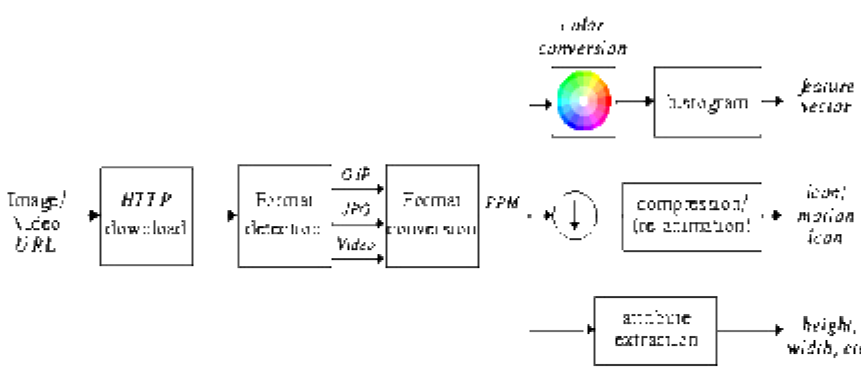


Figure 6. Processing of images and videos in WebSeek.

The user can navigate through the options shown in Figure 7 to perform a textual search. Search and retrieval process consists of record extraction, manipulation, search and ranking, and viewing, as shown in Figure 8. Figure 9 illustrates search results for a textual search for "*nature*" and a content-based search for images/videos of an image of a "*red race car.*" Finally, Figure 10 shows relevance feedback to the user, along with the color histogram manipulations available to use.

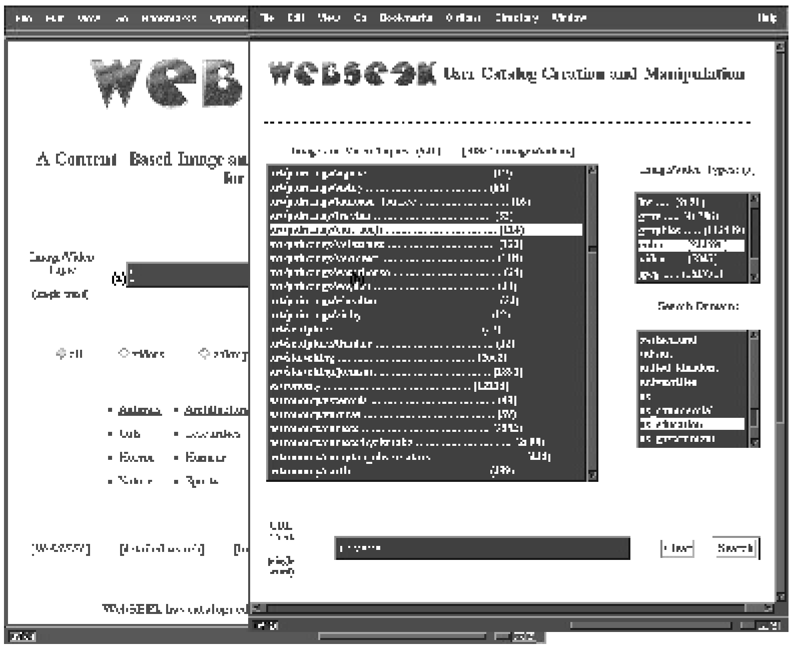


Figure 7. WebSeek - main user screen.

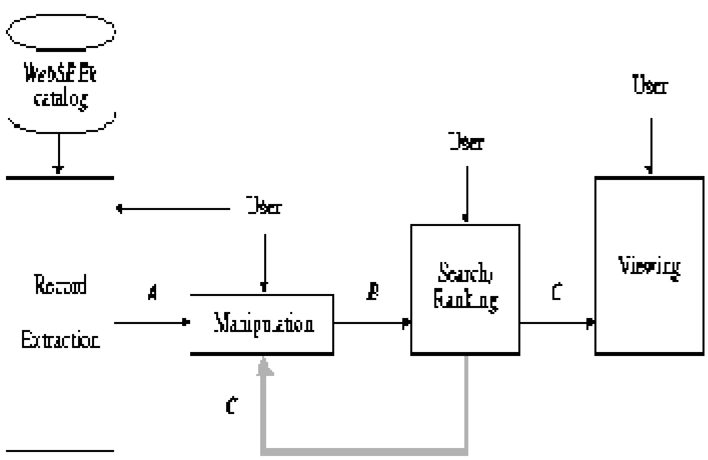


Figure 8. WebSeek - search and retrieval process.

Figure 9. WebSeek - search results for text based query "*nature*" and a content-based query for a "*red sports car.*"

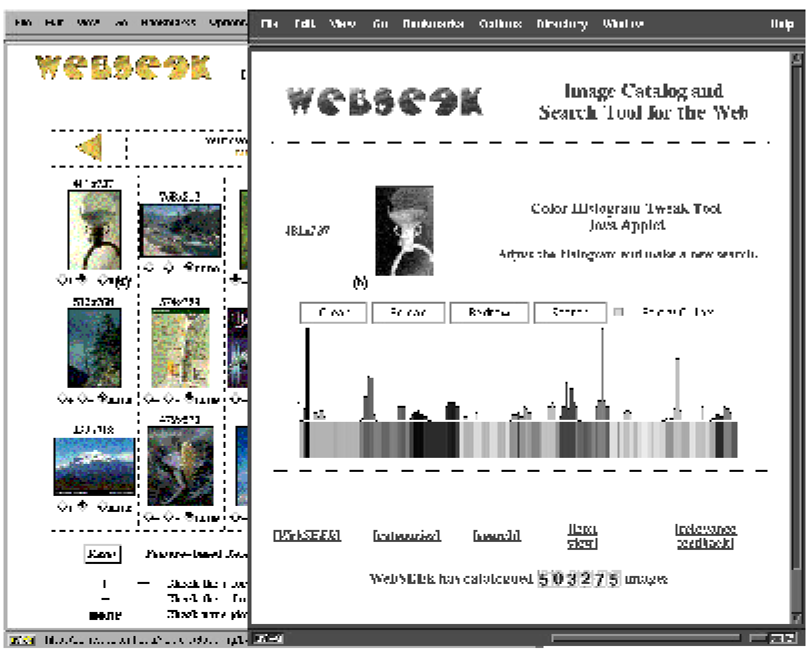


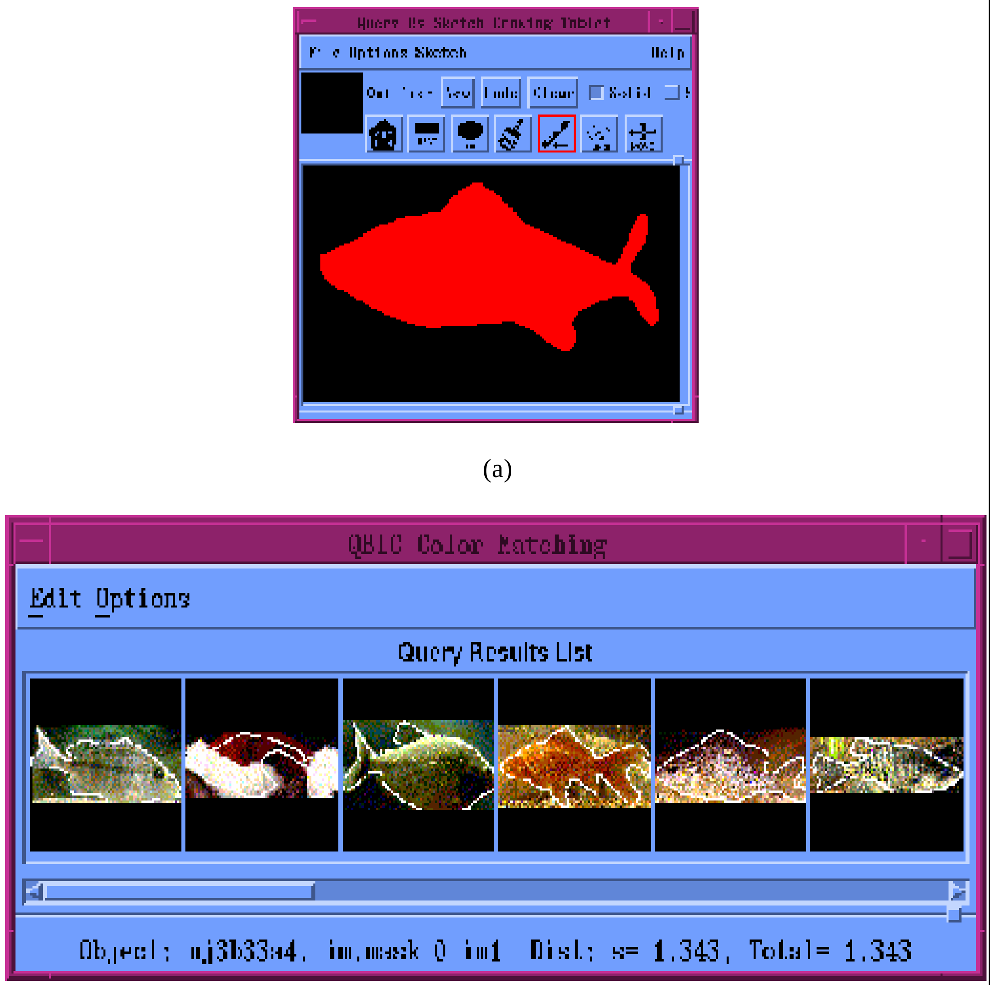
Figure 10. Relevance feedback search for user and color histogram manipulation.

QBIC

QBIC is a system created by IBM, which allows the user to search through an already existing database and find images by textual searches, as well as content-based searches. This system also allows the user to utilize a sketching program to find a similar image to the given sketch drawn. This system assumes an already existing database and does not allow for full Web searching through automated robots. A demo version is available at: **http://www.qbic.almaden.ibm.com/stage/.**

The user can search for images based on color percentages, texture, shape, color layout, and object position. Users can enter a textual description, browse through a given set of thumbnails, or enter a content-based description through an image or sketch. It now includes a standalone Java application providing a GUI front end for use with any database of images.

Figures 11 through 15 show some examples of sample searches one can perform on QBIC system. The sample images were extracted from: **http://www.almaden.ibm.com/cs/showtell/qic/Initpage.html/.**



(a)

(b)

Figure 11. (a) Sketch entry to QBIC, (b) results of search.

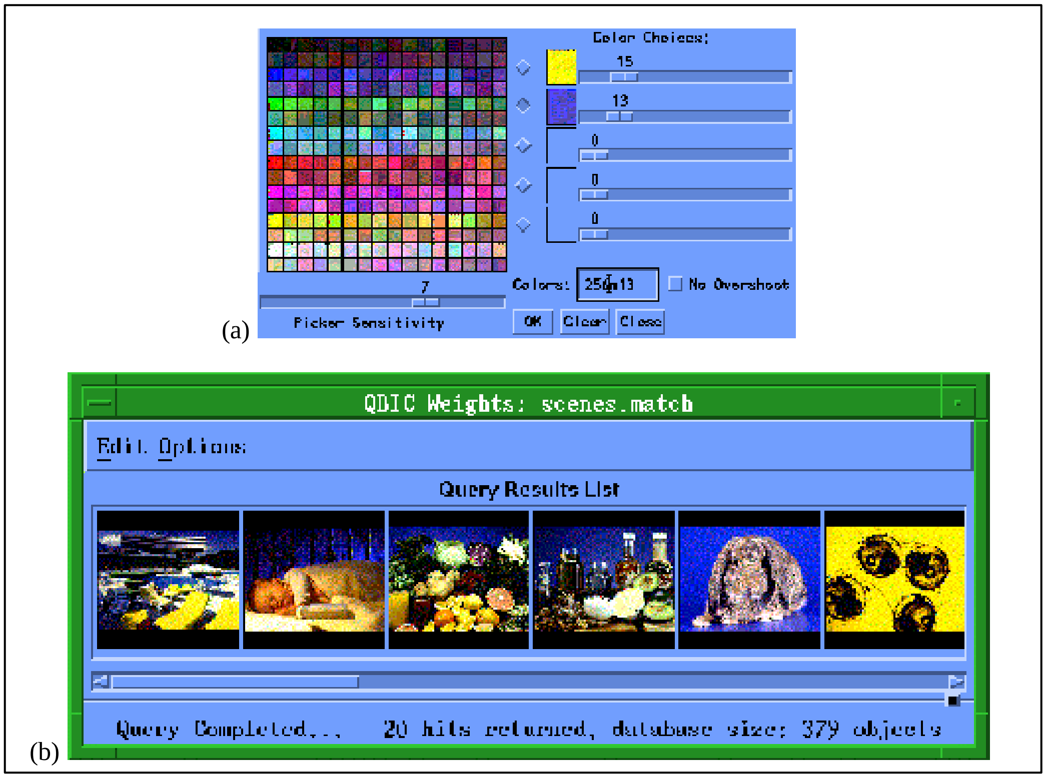


Figure 12. (a) QBIC: Search by color percentages: 15% yellow and 13% blue, (b) search results.

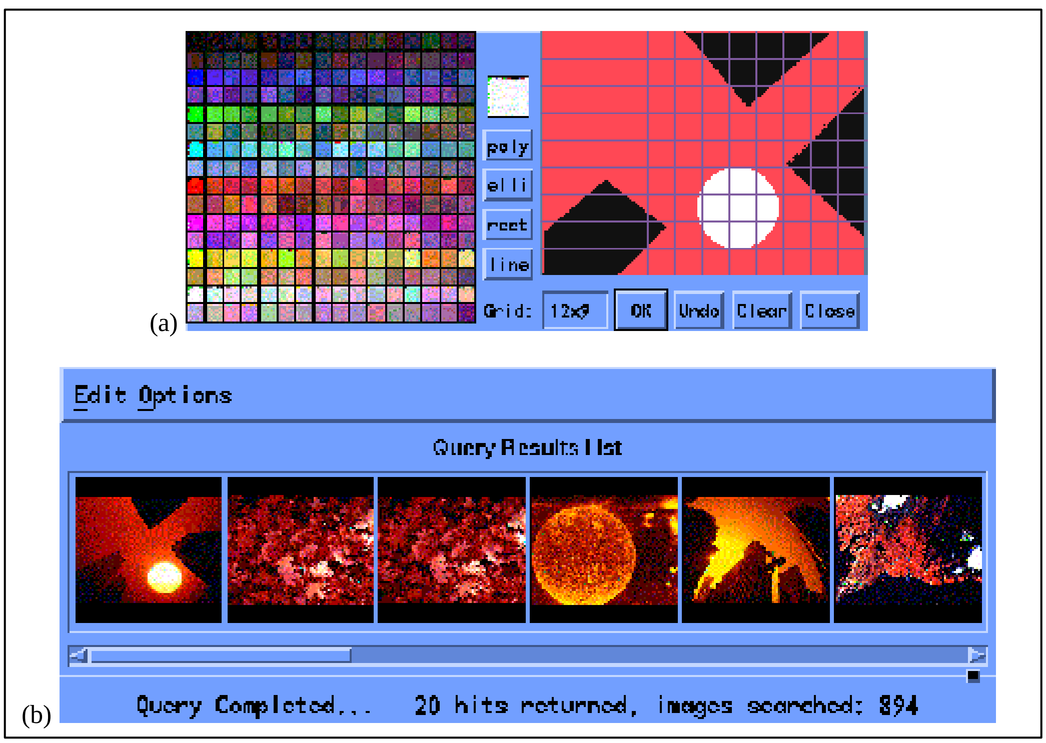


Figure 13. QBIC: Search by user drawn pattern, (b) search results.

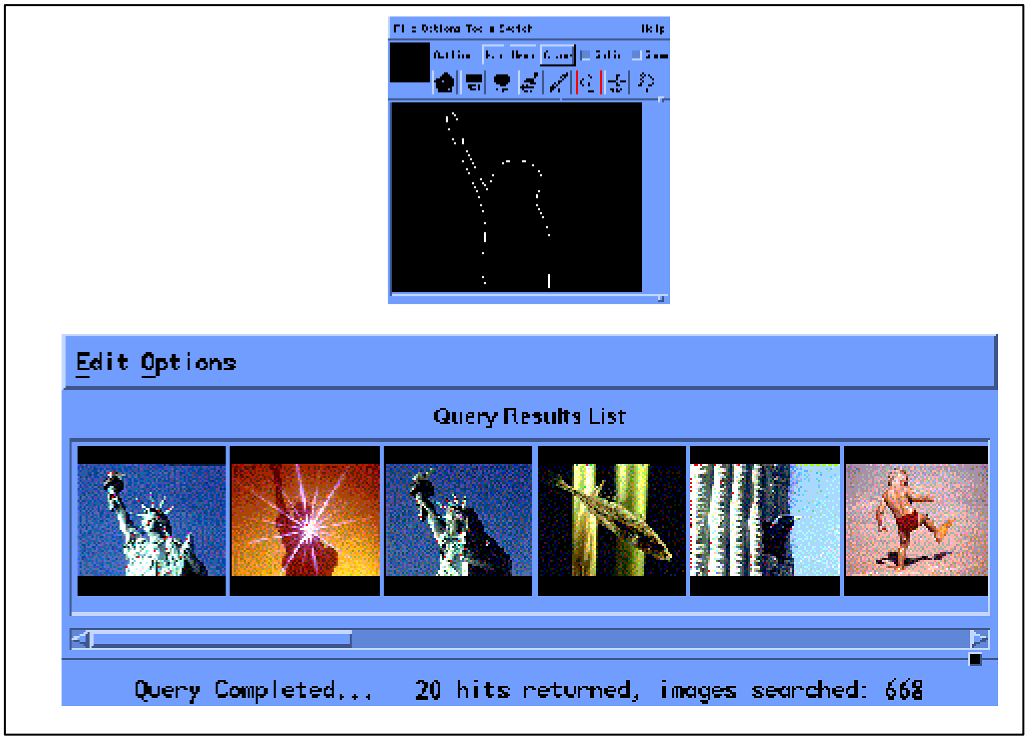


Figure 14. QBIC: Search for sketch created by the user, (b) results for sketch search.

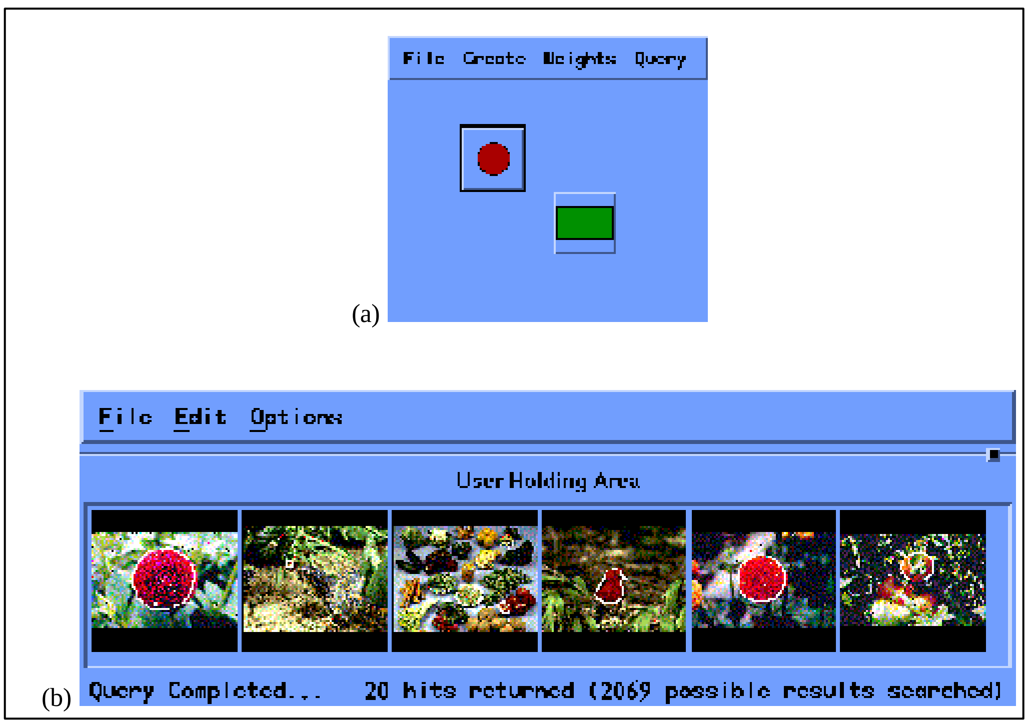


Figure 15. QBIC: (a) Combination of search properties: color (red), shape (round) and background color (green), (b) results from combination search.

QBIC is currently available as a demo form on the Web, as well as in use by the U.C. Davis Art History Department at **http://www.qbic.almaden.ibm.com/stage/**.

AMORE

AMORE is a search engine that also has a Web demo available to try at: **http://www.ccrl.com/amore**. This search engine only has image retrieval capabilities and does not have the ability to search the entire Web via automated robots, but it does have an automated robot (or harvest gatherer as they call it) which can scour and classify images from user specified URLs. The system uses the Harvest Information Discovery and Access System for text indexing and searching, and the content-oriented image retrieval (COIR) to index the images and retrieve them. COIR uses a region-based approach, using attributes like color, texture, size and position for indexing. Everything but the entry of the URL addresses is automated for the user. An example of the indexing mechanism in AMORE is shown in Figure 16.

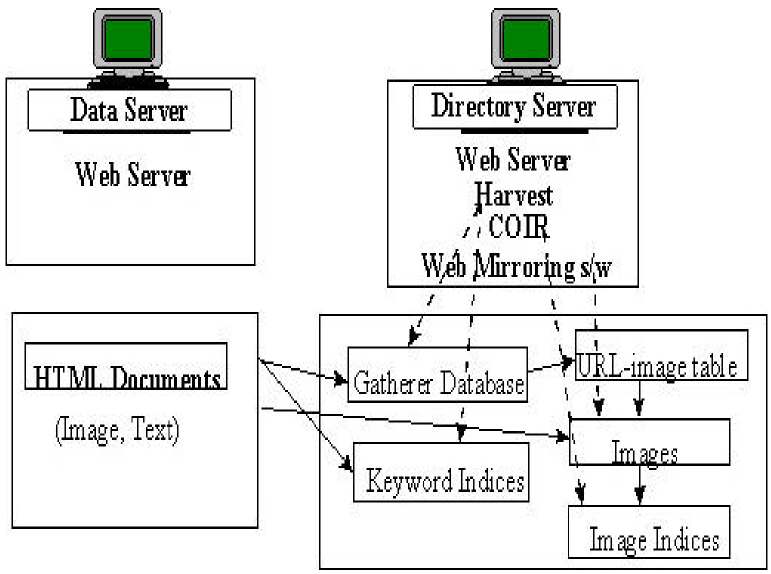


Figure 16. The indexing process in AMORE.

The user can query the system textually or content-based, by entering either a textual keyword image search (i.e. "sailboat"), a content-based search (entering a picture of a sailboat or a sketch of such), or a combination of both. The search engine used is known as "glimpse" and the retrieval engine, COIR, determines similarity of images by shape and color, indexing the images with a match rate and determining compatibility via a threshold value. Both the threshold value and color-shape ratio can be altered by the user for increased match capability. Figure 17 illustrates the querying mechanism.

The user can search for a particular document, search for a class of documents, perform media-based navigation by specifying an image to find a similar image, or enter a rough sketch input, much like in QBIC. Examples of various search queries and results are shown in Figures 18 through 21.

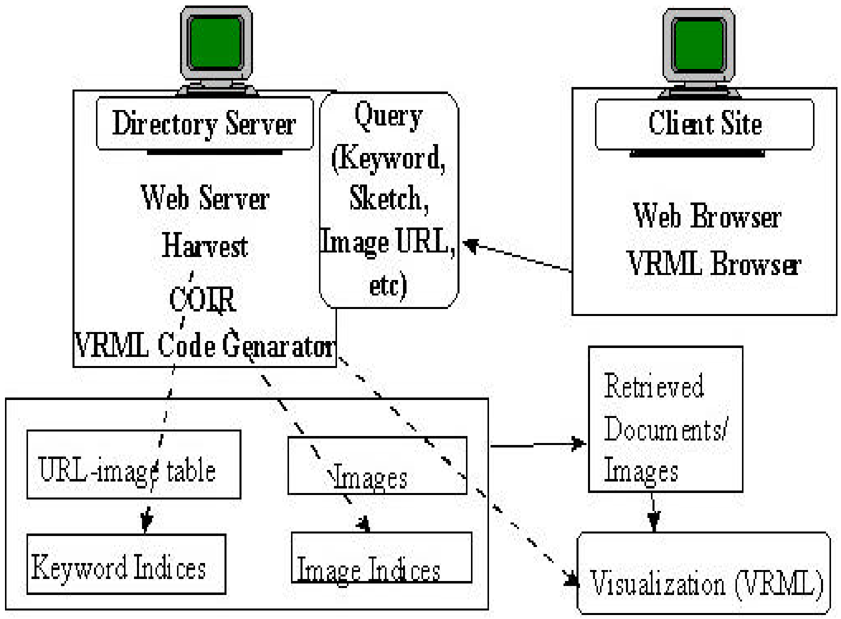


Figure 17. AMORE querying mechanism.

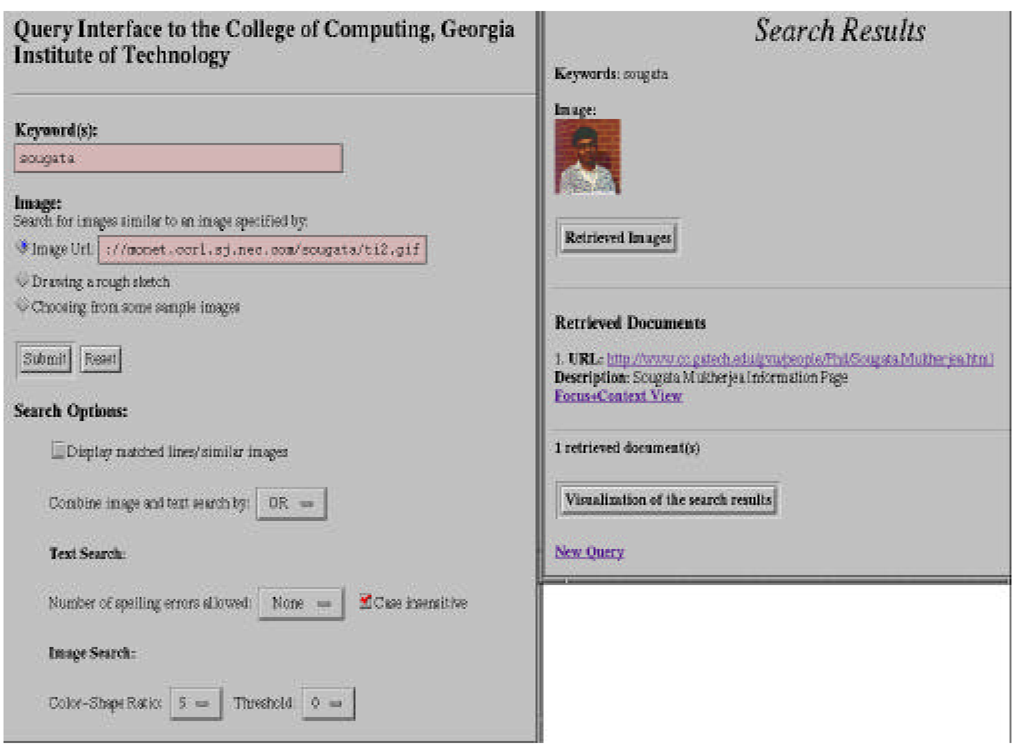


Figure 18. AMORE: Search by text and image to find a Web page.

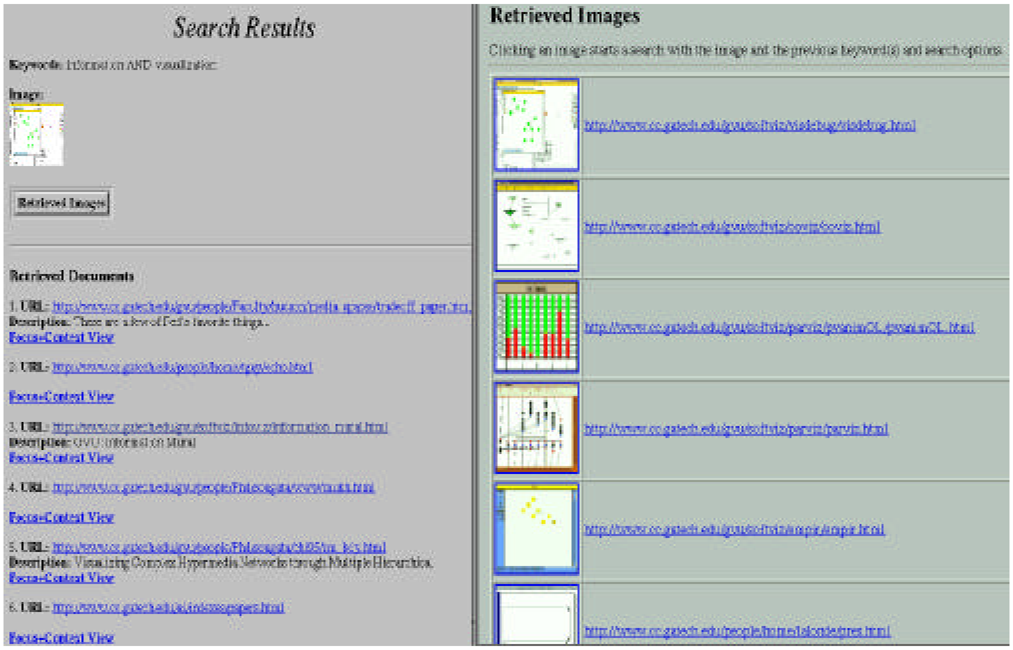


Figure 19. AMORE: Keyword search combined by “or” with an image search of an image of a window dump from an information visualization project. Search results are shown on right.

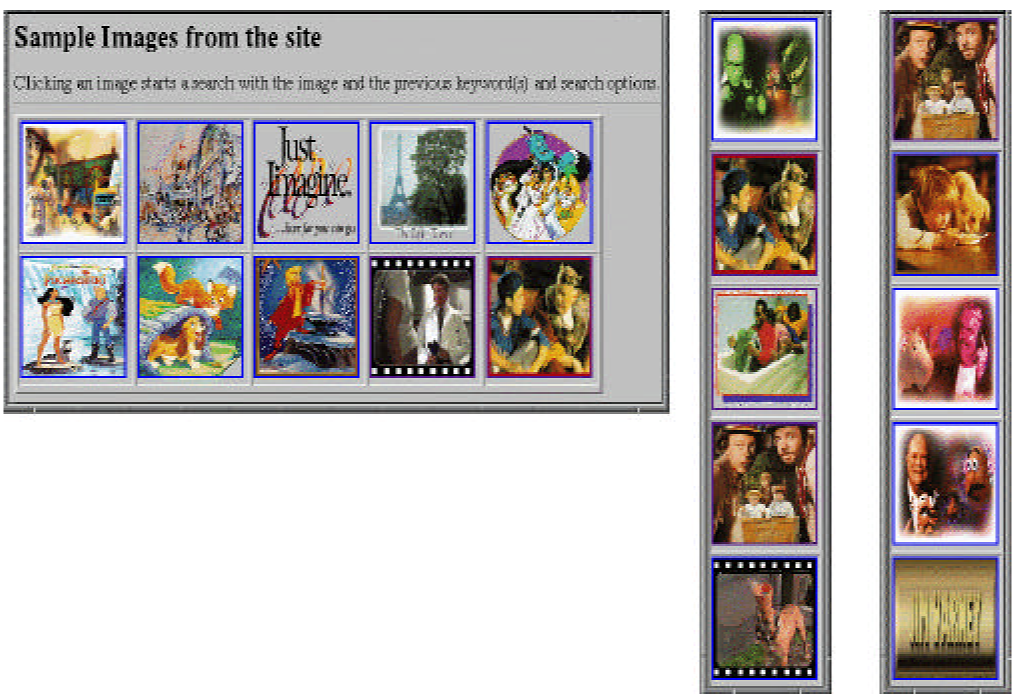


Figure 20. AMORE: Media-based navigation by combining text and image similarity search for movie clips from the Walt Disney Web site.

Figure 21. AMORE: Rough sketch input and results.

The user can visualize results in a number of ways. As shown above, the user can view thumbnail images, Web pages, etc., as a normal view. The user can also view a scatterplot visualization, perspective wall visualization, or a "focus + context" view of Web nodes. These types of visualizations give the user a better understanding of the search results, why they happened, and for the "focus + context" view, the user can also be given the actual location of a particular image in the scheme of a Web page. Examples of the scatterplot view, the perspective wall visualization, and the "focus + context" view are shown in Figures 22, 23, and 24, respectively.

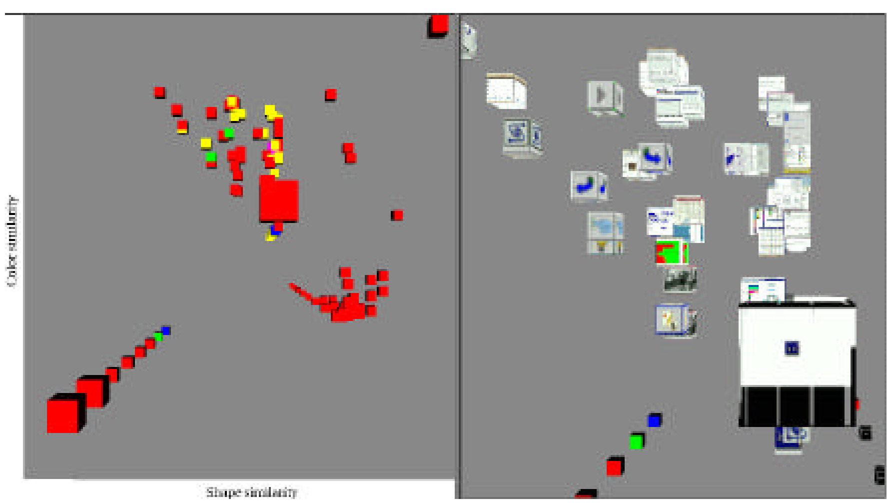


Figure 22. Scatter-plot view of search results plotted by color similarity and shape similarity. The left part shows an overview, and the right part shows the user zoomed into the document with the maximum keywords.

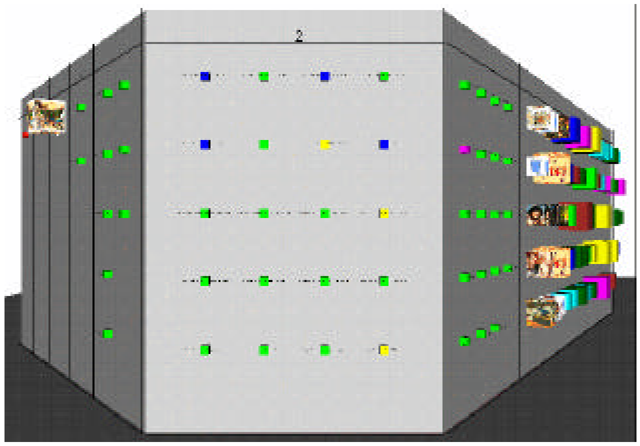


Figure 23. Perspective wall visualization, with each "wall" containing documents with similar number of keywords and the sizes of the cubes representing the similarity of the images in the documents to the user specified image.

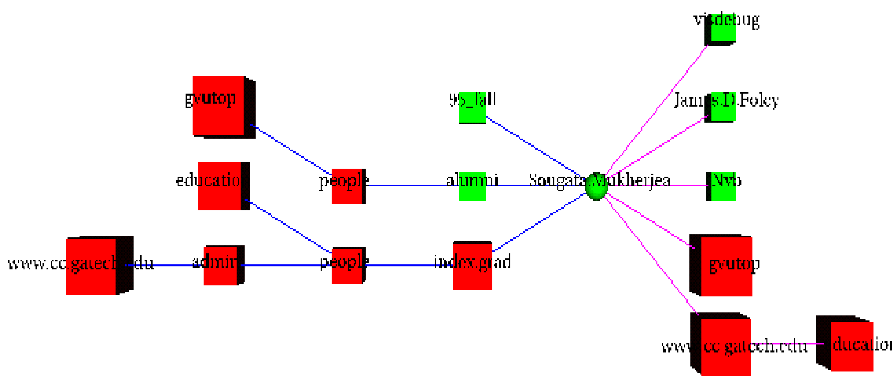


Figure 24. "Focus + context" view of a homepage of one of the authors of [53]. It maps out documents' positioning.

Blobworld

Blobworld [40] is a system, developed at U.C. Berkeley, which allows for content-based image retrieval. The program automatically segments an image into regions, which roughly correspond to object or parts of objects allowing users to query for photographs or images based on the objects they contain. Their approach is useful in finding specific objects and not, as they put it, "stuff" as most systems which concentrate only on "low level" features with little regard for the spatial organization of those features. It allows for both textual and content-based searching.

This system is also useful in its feedback to the user, in that it shows the internal representation of the submitted image and the query results. Thus, unlike some of the other systems, which allow for color histogram similarity metrics, which can be adjusted, this can help the user understand why they are getting certain results.

Images are segmented by modeling the joint distribution of the color, texture, and position features of each pixel in the image. The Expectation-Maximization (EM) algorithm is used to fit a mixture of Gaussian model to the data. The resulting pixel-cluster memberships provide the segmentation of the image. After the segmentation is performed, the description of each region's color, texture, and spatial characteristics is produced.

Figure 25 shows a picture of a wolf and its "blobworld" representation. User screens for content-based and/or text-based image search are shown in Figure 26 and 27. Sample query results from **http://www.cs.berkeley.edu/~carson/blobworld** are shown in Figure 28.

(a)

(b)

Figure 25. (a) Picture of a wolf, (b) blobworld representation of the image.

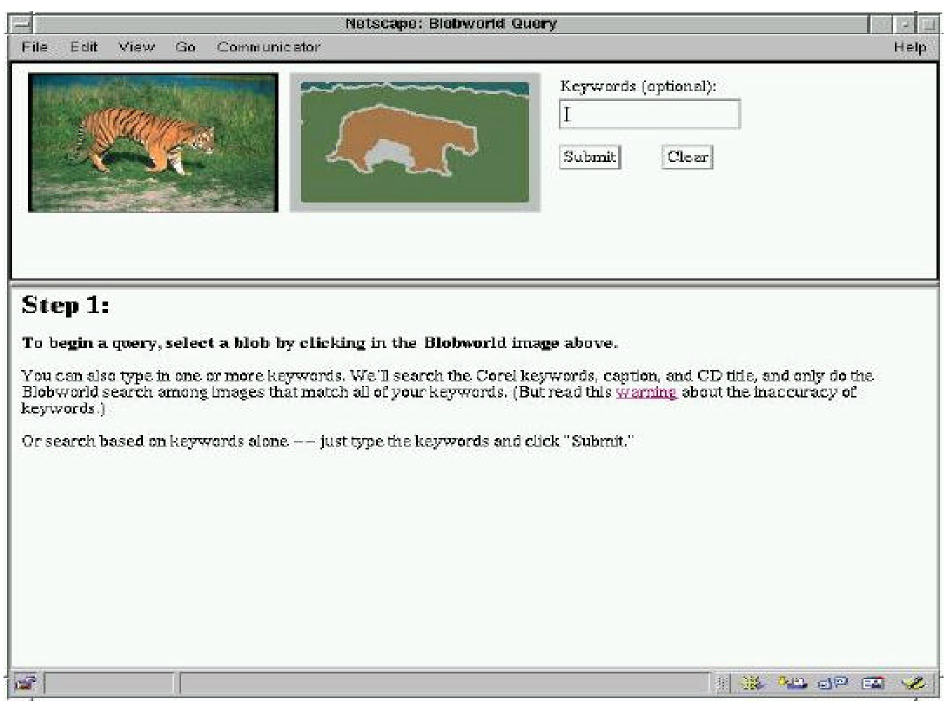


Figure 26. Blobworld: User screen for content-based and/or text based image search.

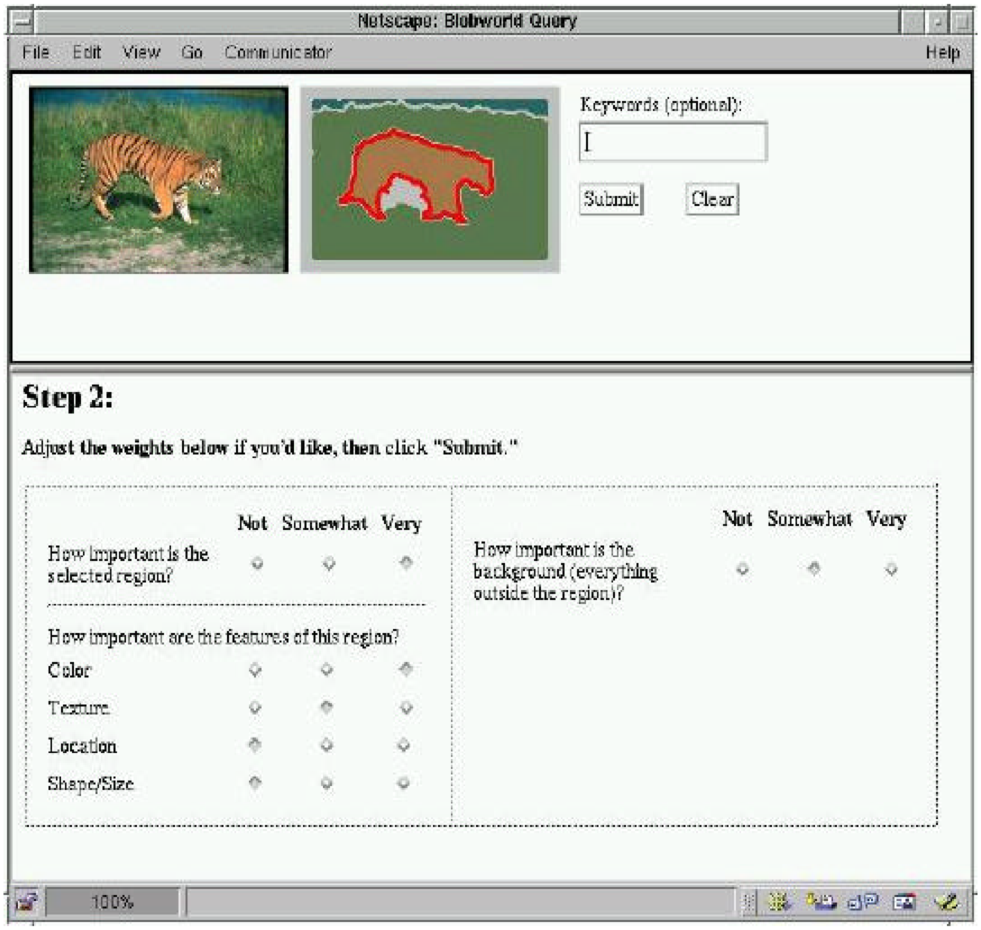


Figure 27. After highlighting the Blobworld region to be looked for, the user can rank the importance of features such as color, texture, location, size/shape and background.

Blobworld is currently available in online demo form at: **http://galaxy.cs.berkeley.edu/photos/blobworld/.**

The UC Berkeley Digital Library Project

The UC Berkeley Digital Library Project is similar in some ways to the Blobworld program. It assumes an existing database of images and bases its queries on "high-level" content of the images. It begins with low level color and texture processing to find coherent regions, and then it uses these properties and their relationship with one another to group them at progressively higher levels.

First, the system locates isolated regions of color in the images for classification by looking for 13 specified colors in each image. The image's hue, saturation, and value (HSV) channels are mapped into the 13 color channels, it is filtered to isolate "dots" and ignore uniformly colored regions, and the output of these filters is then compared to a threshold value. This system is known as "color dots."

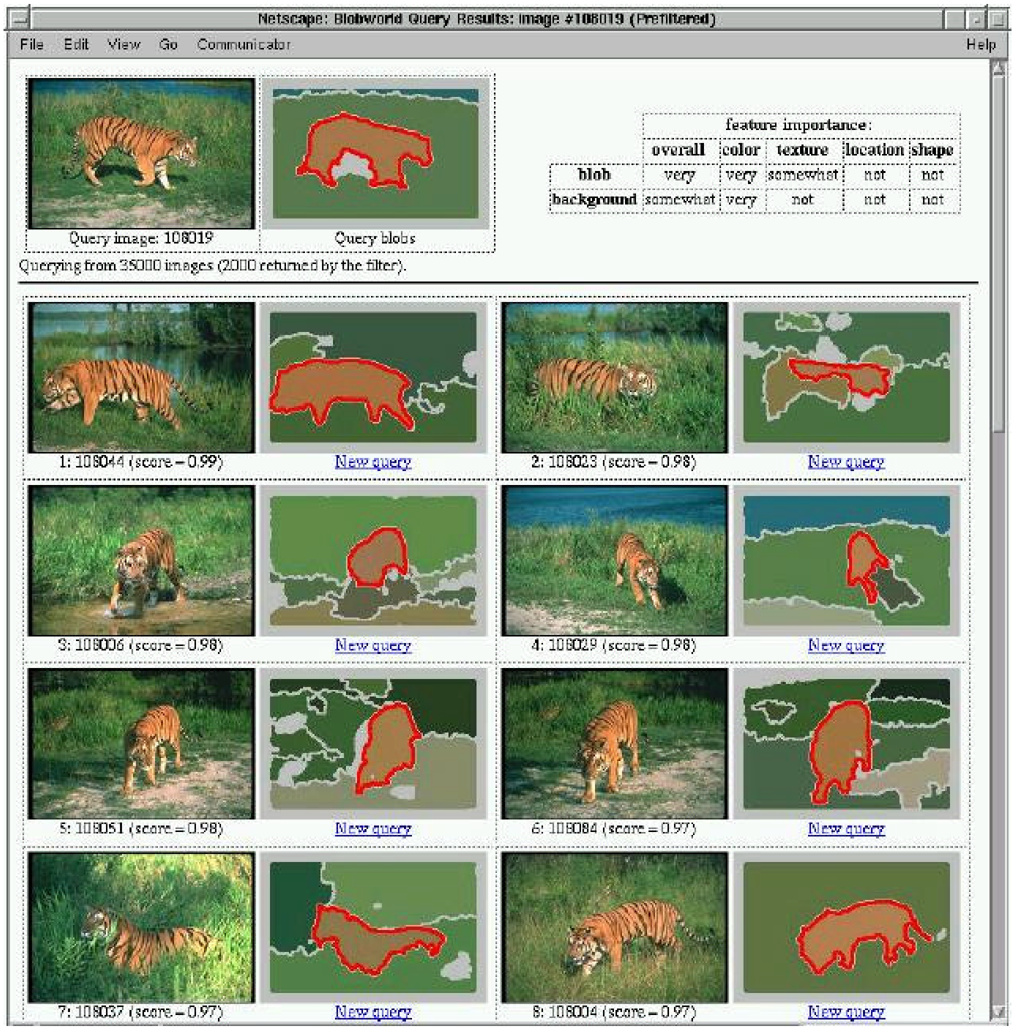


Figure 28. Results from Blobworld query looking for a tiger, which illustrates similar images, their "blobworld" representations, and the chance to perform another query.

Each image is stored only as image metadata in the database available for textual and content-based searching. Each image has 24 textual attributes, including a brief description, a category, subject and internal identification number, as well as some keywords. All of the image's information is stored as text for speed of searching.

This system was designed to handle images that are basically heterogeneous in nature, thus hard to distinguish by most searching means. Figure 29 shows a sample user interface from **http://elib.cs.berkeley.edu/papers/db/**, while results of various searches are presented in Figures 30 through 33.

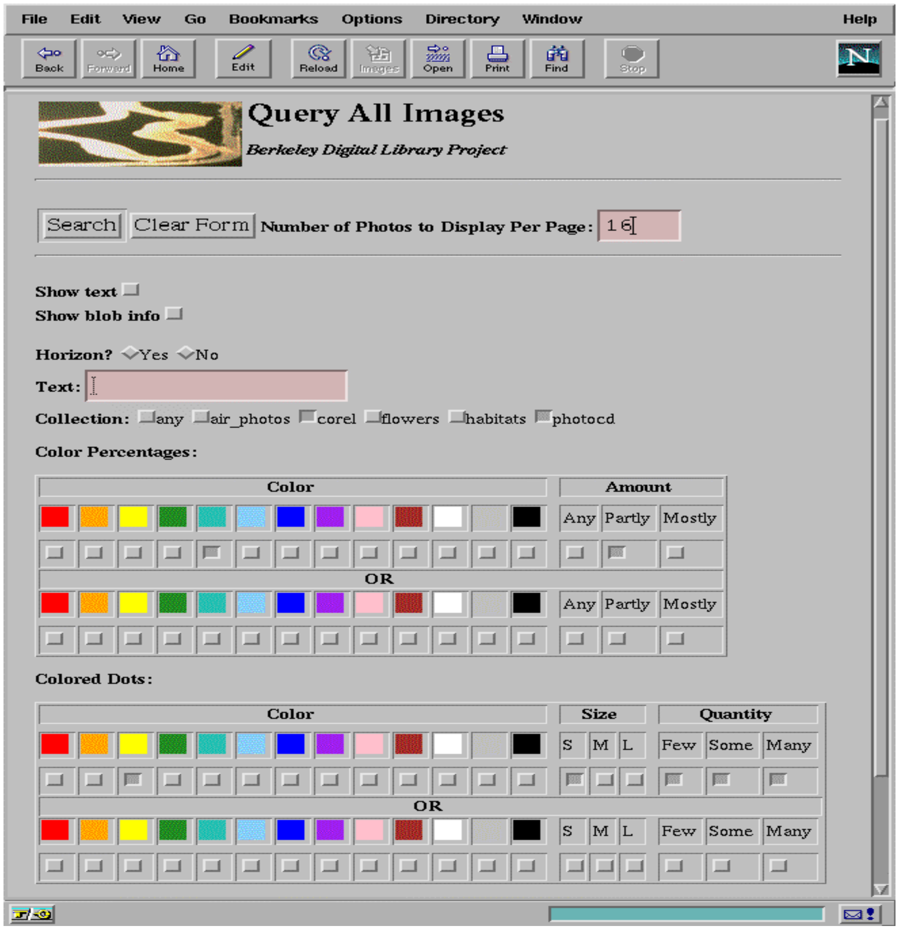


Figure 29. Sample user query interface for The Berkeley Digital Library Project. The user can adjust color, colored dots, database info, or enter text information as well.

Figure 30. The Berkeley Digital Library Project:
Sample results from "sailing and surfing" textual query.

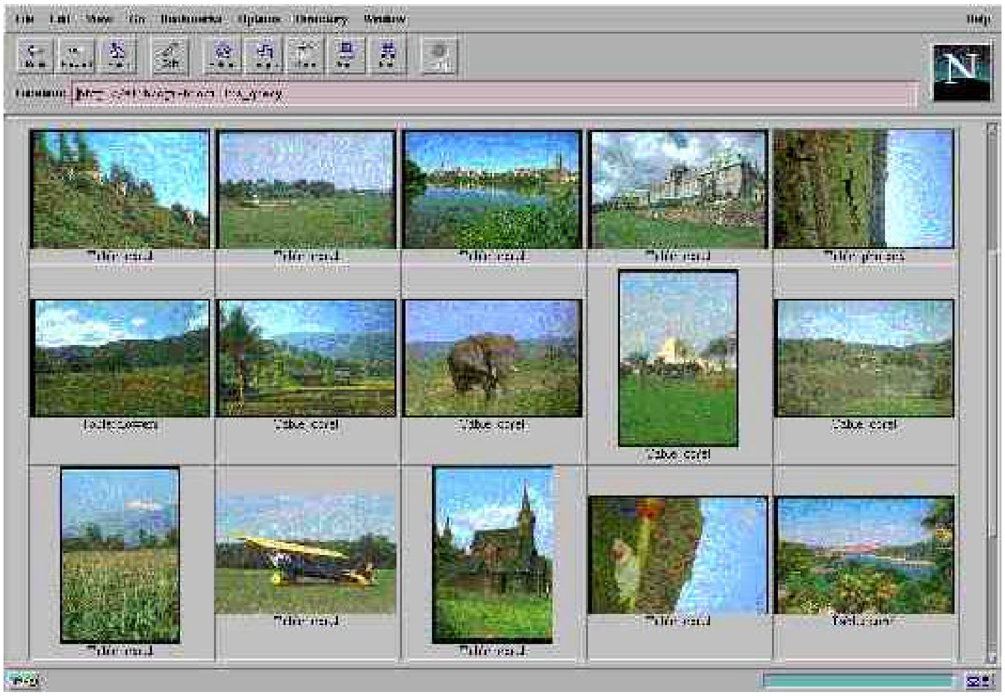


Figure 31. The Berkeley Digital Library Project: Sample query results for "pastoral" query.

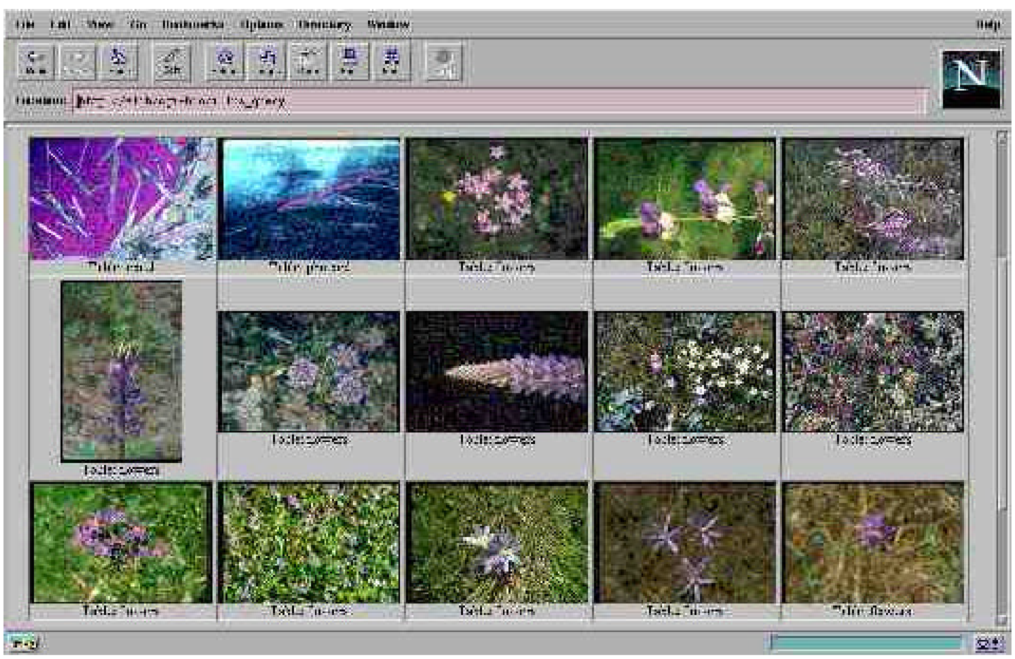


Figure 32. The Berkeley Digital Library Project: Sample results for "purple flowers" query.

Figure 33. The Berkeley Digital Library Project: Sample results for "people" query.

WebSeer

WebSeer [41] was developed by the University of Chicago and is completely automated in all aspects of its searching capabilities. This system only searches for images and allows for only text-based queries. It allows for a user to enter dimensions of an image, a file size range, relevant colors, type of image (photograph, sketch, graphic, etc.) and even the number of faces or size of a portrait. The user receives a resulting selection of thumbnails from their search, which also list the size of the original image. The user can then either click on the thumbnail to go to the URL of the original image or they can click on the page icon next to the image and be transferred to the URL of the page that contains the image.

The system is composed of five major executables and, with the exception of the URL Server, which is written in Java, all executables are written in C++ and run on a Windows NT 3.51 platform. WebSeer works in five steps:

1) The WebSeer Crawler searches the Web for images by scanning for them in the HTML code,
2) The URL Server receives requests to download URLs from the Crawler,
3) The WebSeer Indexer creates the searchable index of images for the user to search,
4) When a user posts a query from the WebSeer form, the CGI script opens a connection to the WebSeer Search Server and formats the results for the user, and
5) The Search Server accepts the request and the search is performed.

Images are indexed and classified by their surrounding HTML information, much the same as WebSeek and ImageRover. This means that the text on the page containing the image is examined, such as the image file name, the captions, the textual description of the image (if any), the document title, hyperlinks, or other text. Further, the image content is looked at (such as size, file type, color/grayscale, and date) in order to further categorize the image in the database. Tests such as the band difference test, the farthest neighbor test, the color test, the most common color test and the narrowness test each are used to decide if the image is a photograph or drawing. Face matching techniques, developed by MIT and Carnegie Mellon University, are used to detect faces in the images.

Virage

Virage [51] was developed by Virage Inc. and it allows for query by text and by example. This system also allows for both video and image retrieval, as well as query by sketch. This system is now available through AltaVista and snap.com, greatly improving these companies' search engines' capabilities. Virage, Inc. has an online demo of its search engine in which images/videos can be searched based on color, texture, color distribution in the image, and general structure. Searches can be based on various weighted characteristics of the images/videos. This system runs on an existing database (no auto searching).

This system works with four layers of information abstraction, the raw image or Image Representation Layer, the process image or Image Object Layer, the user's feature of interest or Domain Object Layer, and the user's events of interest for videos or the Domain Even t Layer. Virage features three primitives to calculate image metadata, namely, color, texture and shape, which the user has the power to adjust the weights of during a search. Information such as scene breaks, cuts, dissolves or fades are taken into account for videos. Image metadata is stored in vector form for videos and images, much like the other systems.

Virage includes image processing tools to normalize, scale, or crop images, as well as perform more advanced operations such as color histograms. Users can insert images for comparison or perform textual searches. Also, users can enter a sketch to compare and find similar images/videos too.

Figure 34 shows an example of Virage's technology on snap.com, which runs textual searches of videos (in this case for the presidential election 2000 and "education") and presents results to the user who can then click on the icons to play the video or view the text of the video.

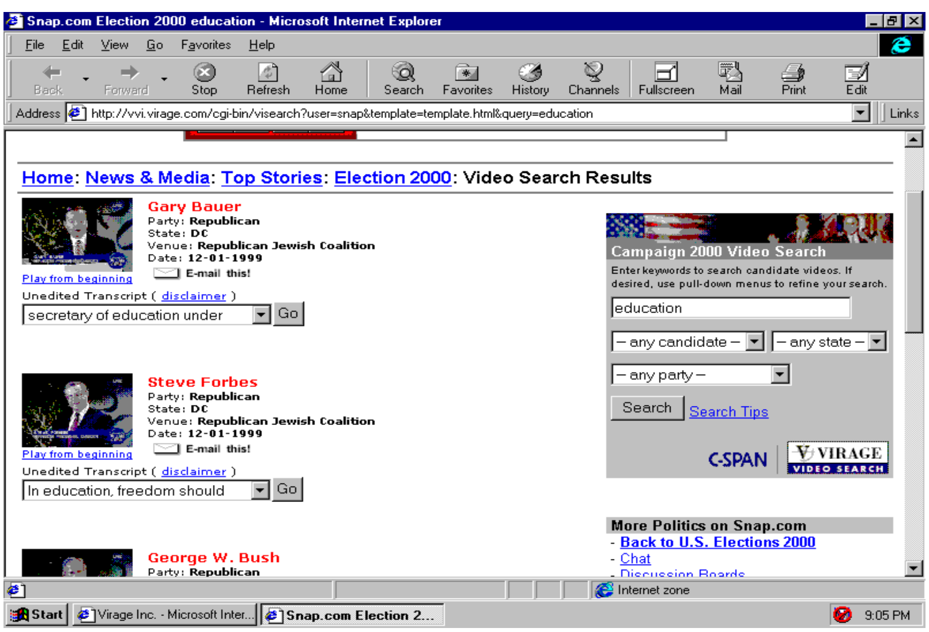


Figure 34. Example of Virage technology with snap.com. The user can query the system for video clips of presidential candidates by entering text.

VisualSEEk

VisualSEEk is a part of the Columbia University projects such as WebSeek and WebClip. The system enhances the search capability by integrating the spatial query (like those used in geographic information systems) and the visual feature query. Users ask the system to find images/video that include regions of matched features and spatial relationships. Figure 35 shows a query example in which two spatially arranged color patches were issued to find images with blue sky and open grass fields.

Figure 35. Sample query for color patches using VisualSEEk.

Yahoo's Isurf

Yahoo is a popular commercial based search engine, which utilizes Excalibur technologies in its image-based search engine. Images are searched in a text-based only method and Web addresses must be entered manually by the user for categorization. The system is similar to AMORE, only the images are added to a global database for all users and image classification is manually dependent on user descriptions/keywords. Figure 36 illustrates an example of text-based search for "cats" using Yahoo's Isurf.

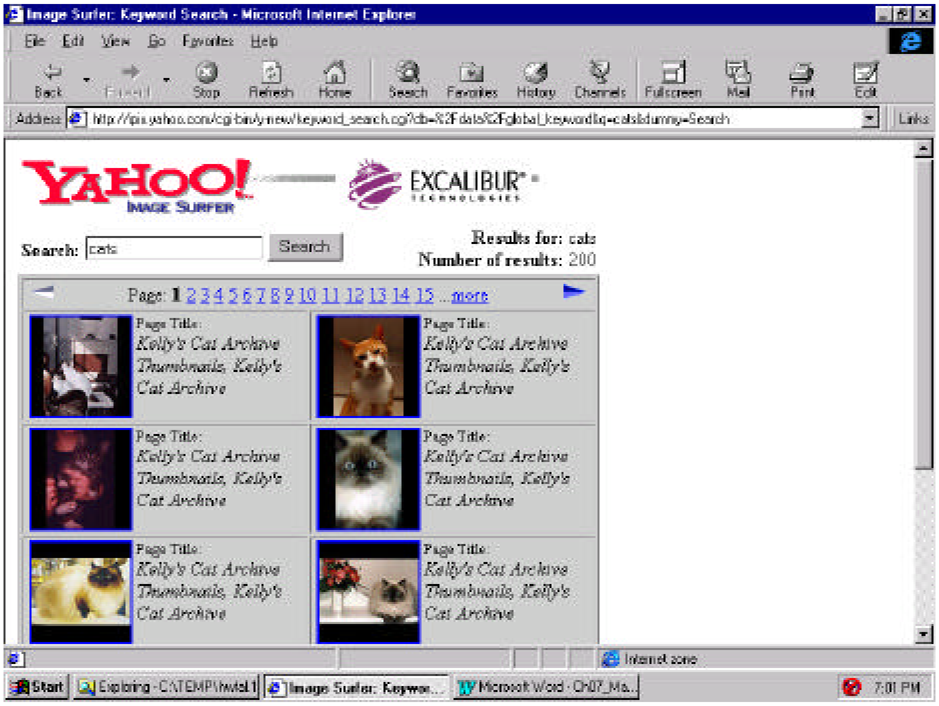


Figure 36. Yahoo's Isurf: Results for "cats" search.

The results in Figure 36 are in thumbnail form and the user can click on the images to go to the Web page of origination of the image. Namely, an image cannot be retrieved and is not stored on Yahoo's database.

AltaVista

As mentioned before, AltaVista has acquired image and video searching capabilities through Virage, Inc. This search engine, like Yahoo, is only text-based currently, allowing the user to enter textual data to locate images or videos categorized textually via keywords/categories in the system. The user cannot add pages to this database as in Yahoo. Once a set of images is returned to the user, the user can click on the text below the image or video to find out more information about it, and then can refine the search by "finding similar pages to that page."

Figure 37 illustrates an example of text-based search for "cats." By clicking on one of the images the user can get additional information about the particular image, as illustrated in Figure 38. The results for video searches are the same and cannot be performed in conjunction with image searches.

Netra

NETRA is a prototype image retrieval system that is currently being developed within the UCSB Alexandria Digital Library project. NETRA uses color, texture, shape and spatial location information in segmented image regions to search and retrieve similar regions from a database. This is a region-based program (much like Blobworld), so the user can choose a region and find similar image regions to the submitted one, thus being a content-based system only. Also, the user can choose multiple region attributes for a search.

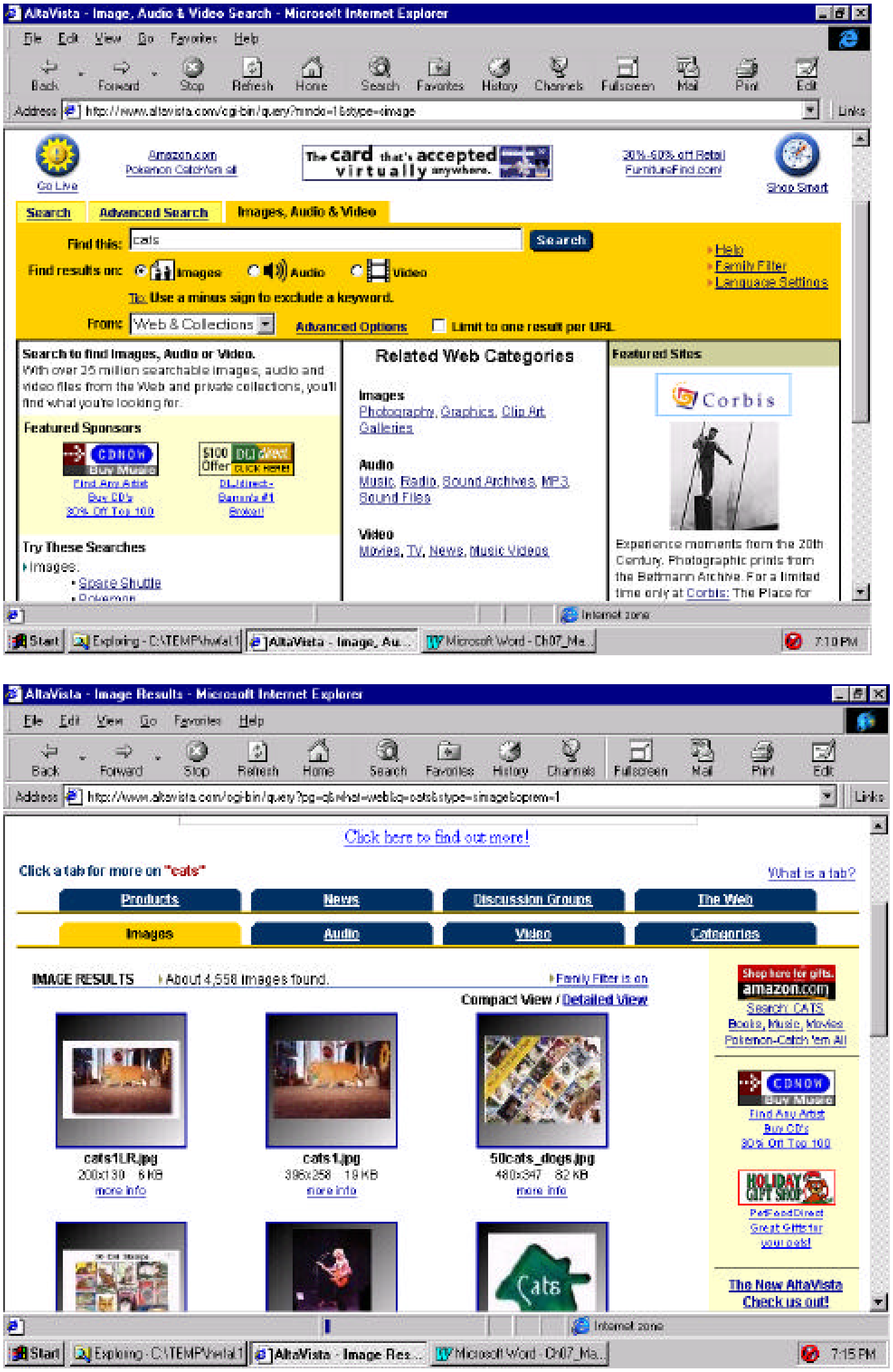


Figure 37. AltaVista: Textual search for "cats." Each image can be clicked on to go to that page, or find out more information about the image.

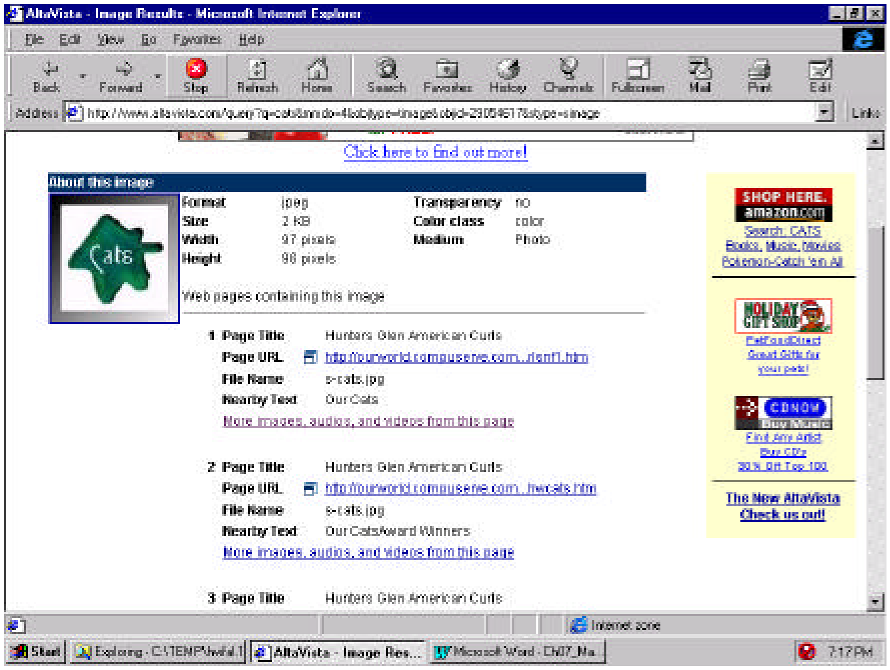


Figure 38. AltaVista: Additional information about a particular image.

PicToSeek

Much like ImageRover, WebSeek and WebSeer, PicToSeek automatically scans the Web for images and allows the user to search by content-based techniques of image matching. Figure 39 shows a sample search of the image on the left, with results on the right. When the user clicks on the small images on the left, that image is loaded and the Web address is given to the user.

Figure 39. PickToSeek: Results of picture-based query.

VideoSeeker

As part of snap.com, VideoSeeker search engine lets the user enter a textual search for a video clip. As the result of search, the user receives textual results, which he/she can click on to download and view a short video as shown in Figures 40 and 41.

Figure 40. VideoSeeker: Search engine interface to locate videos based on textual query, say “Friends”.

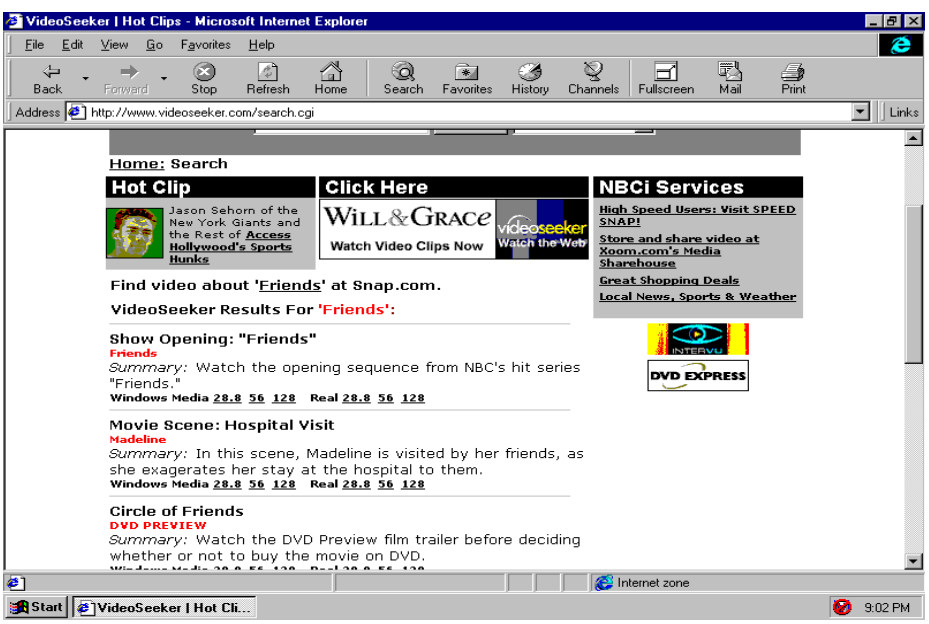


Figure 41. VideoSeeker: Results of search for “Friends,” which the user can click on to download and view video clips from the NBC show.

FourEyes

FourEyes is a tool, which segments and annotates images and it is included in the Photobook-6 interface. It was developed by MIT; more information can be found at **http://www-white.media.mit.edu/vismode/demos/photobook/foureyes/**. With this program, the user can click on certain regions of an image and label them. Depending on the type of feature the user is looking for, FourEyes will choose a labeling model among its "society of models" or combine models in order to label the user's image. Figure 42 shows an example of FourEyes extrapolating the labels of "building," "car," and "street" from an image. In this way, this search engine can intelligently search for various regions on an image.

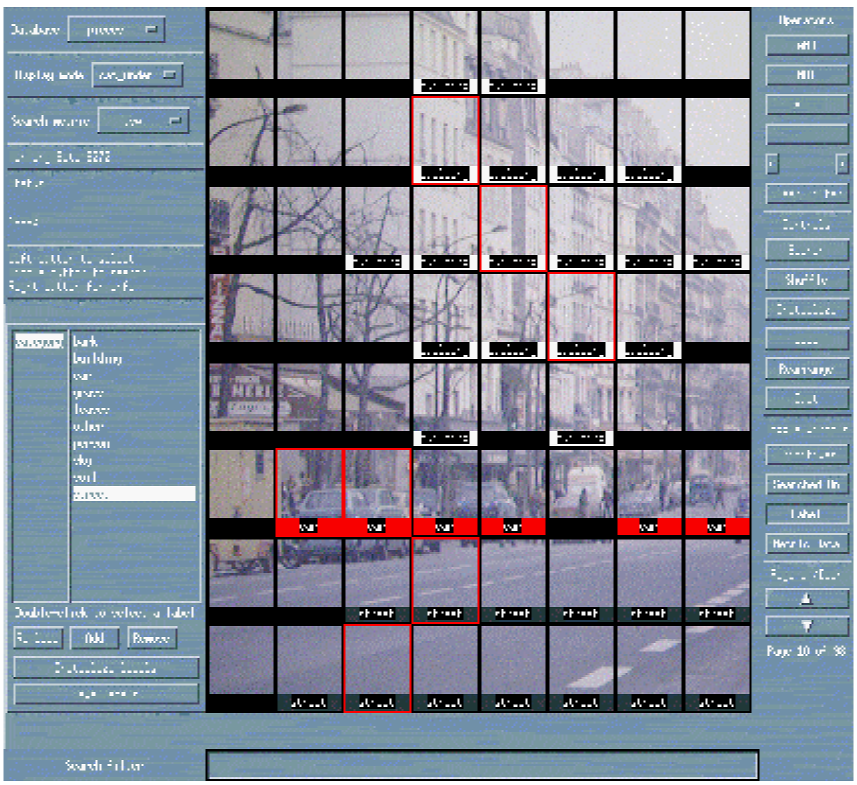


Figure 42. FourEyes: Automatic labeling and segmentation of an image for searching based on regions.

6. SUMMARY

In this chapter we discussed the main issues and challenges in designing multimedia search and retrieval systems with the focus on the Web-based CBVIR systems, which provide search and retrieval on the Internet. We presented the system architecture of a CBVIR system and addressed open issues in designing and developing these systems. We also presented a number of commercial and prototype image and video retrieval systems, which are both text-based and content-based. Table 1summarizes the features of these systems.

Table 1. Characteristics of Commercial Multimedia Retrieval Systems

SYSTEM	SEARCHES BY	SEARCHES FOR	DATABASE Where it searches?	MAIN USE
AltaVista	Text	Images/Videos	Separately created collection	To find images/videos on the Web easily
Lycos	Text	Images/Videos	Entire Web With Web Robots (FAST)	To find images/videos on the Web easily
Yahoo	Text	Images/Videos	Collection of URLs manually entered by all Web users	To find images/videos on the Web easily
VideoSeeker	Text	Videos	Separate collection of video clips	To find video clips to download online
ImageRover	Text/Content	Images	Entire Web With Web Robots	To find images on the Web and be able to refine searches through image comparison
AMORE	Text/Content	Images	URLs specified by the user	To find images on specific Web pages and have adjustable values to search/refine searches by or search by sketch
WebSeek	Text/Content	Images/Videos	Entire Web With Web Robots	To find images/videos on the Web and have adjustable values to search/refine searches by
WebSeer	Text/Content	Images	Entire Web With Web Robots	To find images and have adjustable values to find images or specific people/number of faces
The Berkeley Digital Library Project	Text/Content	Images	Separately created	To find images in a database of heterogeneous images
PicToSeek	Text/Content	Images	Entire Web With Web Robots	To find images with adjustable values/ refineable search means
Netra		Images	Separately created	To find by region matching via content-based retrieval
PhotoBook/Foureyes	Text/Content	Images	Separately created	To find specific regions of images or automatically label regions of images
VisualSeek	Text/Content	Images	Separately created	To search a specific URL or through a database of images to find by content or by region
Blobworld	Text/Content	Images	Separately created	To find specific objects through “blob” or object comparisons
Virage	Text/Content	Images/Video	Separately created	To find images/videos in an online collection with adjustable values to search by or to search by sketching
QBIC	Text/Content	Images	Separately created	To find images in an online collection with adjustable values to search by or to search by sketching

References

1. O. Marques and B. Furht, "Issues in Designing Contemporary Video Database Systems," *Proc. IASTED Conference on Internet and Multimedia Systems and Applications*, Nassau, Bahamas, October 1999.
2. S.-F. Chang, J. R. Smith, M. Beigi, and A. Benitez, “Visual Information Retrieval from Large Distributed Online Repositories,” *Communications of the ACM*, Vol. 40, No. 12, December 1997.
3. Gupta, S. Santini, and R. Jain, "In Search of Information in Visual Media," *Communications of the ACM*, Vol. 40, No. 12, 1997.
4. Y. Rui, T.S. Huang, and S.-F. Chang, "Image Retrieval: Past, Present, and Future,"
5. M. J. Swain and D. H. Ballard, "Indexing via Color Histograms", *Proc. Third International Conference on Computer Vision*, 1990.
6. M. Stricker and M. Orengo, "Similarity of Color Images," *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1995.
7. J. R. Smith and S.-F.Chang, "Tools and Techniques for Color Image Retrieval," *Proc. SPIE Conference on Storage and Retrieval for Image and Video Database IV, 2670*, San Jose, CA, February 1996.
8. J. R. Smith and S.-F.Chang, S.-F., "Single Color Extraction and Image Query," *Proc. IEEE International Conference on Image Processing*, 1995.
9. G. Pass, R. Zabih, and J. Miller, "Comparing Images Using Color Coherence Vectors," Proc. ACM Conference on Multimedia, 1996.
10. J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, R. and Zabih, "Image Indexing Using Color Correlogram," *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 1997.
11. R. Haralick, K. Shanmugam, and I. Dinstein, "Texture Features for Image Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, No. 6, 1973.
12. H. Tamura, S. Mori, and T. Yamawaki, "Texture Features Corresponding to Visual Perception", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 8, No. 6, 1978.
13. T. Chang and C.-C. Jay Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," *IEEE Transactions on Image Processing*, Vol. 2, No. 4, 1993.
14. A. Laine and J. Fan, "Texture Classification by Wavelet Packet Signatures," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Vol. 15, No. 11, 1993.
15. M. H. Gross, R. Koch, L. Lippert, and A. Dreger, "Multiscale Image Texture Analysis in Wavelet Spaces," *Proc. IEEE International Conference on Image Processing*, 1994.
16. A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases," Chapter 2 in *Multimedia Tools and Applications*, Furht, B., Ed., Kluwer Academic Publishers, Boston, 1996,
17. T. Minka, "An Image Database Browser that Learns from User Interaction," MEng Thesis, MIT, 1996.
18. T. Gevers and A. W. M. Smeulders, "The PicToSeek WWW Image Search System," *Proceedings of the International Conference on Multimedia Computing and Systems*, Florence, Italy, 1999.

19. Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 644-655, 1998.

20. Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "A Relevance Feedback Architecture for Content-Based Multimedia Information Retrieval Systems," 1997.

21. Y. Rui, T. S. Huang, and S. Mehrotra, "Content-Based Image Retrieval with Relevance Feedback in MARS," *Proc. IEEE Int. Conf. Image Processing*, 1997.

22. Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "Automatic Matching Tool Selection Using Relevance Feedback in MARS", Proc. 2nd Int. Conf. Visual Information Systems, 1997.

23. Y. Rui, T. S. Huang, and S. Mehrotra, "Relevance Feedback Techniques in Interactive Content-Based Image Retrieval," *Proc. S&T SPIE Storage and Retrieval of Images/ Video Databases VI, EI'98*, 1998.

24. M. Ortega, Y. Rui, K. Chakrabarti, A. Warshavsky, S. Mehrotra, and T.S. Huang, "Supporting Ranked Boolean Similarity Queries in MARS,"

25. I.J. Cox, M.L. Miller, T.P. Minka, T. Papathomas, and P.N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments," *IEEE Transactions on Image Processing*, 2000 (to appear).

26. I.J. Cox, M.L. Miller, S.M. Omohundro, and P.N. Yianilos, "PicHunter: Bayesian Relevance Feedback for Image Retrieval," *Proc. Int. Conference on Pattern Recognition*, Vienna, Austria, 1996.

27. I.J. Cox, M.L. Miller, S.M. Omohundro, and P.N. Yianilos, "Target Testing and the PicHunter Bayesian Multimedia Retrieval System", Advanced Digital Libraries ADL'96 Forum, Washington D.C., 1996.

28. I.J. Cox, M.L. Miller, T. Papathomas, J. Ghosn, and P.N. Yianilos, "Hidden Annotation in Content Based Image Retrieval," *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

29. I.J. Cox, M.L. Miller, T.P. Minka, T. Papathomas, and P.N. Yianilos, "An Optimized Interaction Strategy for Bayesian Relevance Feedback," CVPR, 1998.

30. T.P. Minka and R. Picard, “Interactive learning using a 'society of models,” MIT Media Laboratory Perceptual Computing Section Technical Report No. 349, 1995.

31. A. Benitez, M. Beigi, and S.-F. Chang, "Using Relevance Feedback in Content-Based Image Search", *IEEE Internet Computing*, Vol. 2, No. 4, pp. 59-69, 1998.

32. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," in *Intelligent Multimedia Information Retrieval*, Maybury, M. T., Ed., American Association for Artificial Intelligence (AAAI), Menlo Park, CA: 1997.

33. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC Project: Querying Images By Content Using Color, Texture and Shape," 1993.

34. J. Ashley, R. Barber, M. Flickner, J. Hafner, D. Lee, W. Niblack, and D. Petkovic, "Automatic and Semi-Automatic Methods for Image Annotation and Retrieval in QBIC," Research Report RJ 9951 (87910), IBM Research Division, Almaden Research Center, 1995.

35. Y. Deng and B.S. Manjunath, "NeTra-V: Toward an Object-Based Video Representation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5, 1998.

36. J. R. Smith and S.-F.Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System," *Proc. ACM Multimedia '96*, Boston, MA, November, 1996.

37. J. R. Smith and S.-F.Chang, "Querying by Color Regions Using the Visual SEEk Content-Based Visual Query System," in *Intelligent Multimedia Information Retrieval*, Maybury, M. T., Ed., American Association for Artificial Intelligence (AAAI), Menlo Park, CA, 1997.

38. J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. Shu, "The Virage Image Search Engine: An Open Framework for Image Management," *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*,

39. J. Dowe, "Content-Based Retrieval in Multimedia Imaging," *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*, 1993.

40. Blobworld home page. http://www.cs.berkeley.edu/~carson/blobworld/.

41. Webseer home page. http://infolab.cs.uchicago.edu/webseer/.

42. ImageRover home page. http://www.cs.bu.edu/groups/ivc/ImageRover/Home.html.

43. W.-Y. Ying Ma and H.J. Zhang, "Content-Based Image Indexing and Retrieval," Chapter 11 in the Handbook of Multimedia Computing, B. Furht, Ed., CRC Press, Boca Raton, Florida, 1999.

44. H.J. Zhang, "Content-Based Video Browsing and Retrieval," Chapter 12 in the Handbook of Multimedia Computing, B. Furht, Ed., CRC Press, Boca Raton, Florida, 1999.

45. G. Ahanger and T.D.C. Little, "A Survey of Technologies for Parsing and Indexing Digital Video," *Journal of Visual Communication and Image Representation*, Special Issue on Digital Libraries.

46. S.-F. Chang, Q. Huang, T. Huang, A. Puri, and B. Shahraray, "Multimedia Search and Retrieval," Chapter in Advances in Multimedia: Systems, Standards, and Network, A. Puri and T. Chen, Eds., Marcel Dekker, New York, 1999.

47. B.-L. Yeo and M.M. Yeung, "Retrieving and Visualizing Video," *Communications of the ACM*, Vol. 40, No. 12, December 1997, pp. 43-52.

48. S. Sclaroff, L. Taycher, and M. La Cascia, "ImageRover: A Content-based Image browser for the World Wide Web," *Proc. IEEE Workshop on Content-based Access of Image and Video Libraries*, June 1997.

49. J. R. Smith and S. Chang, "Searching for Images and Videos on the World-Wide Web,"*Center for Telecommunications Research Technical Report #459-96-25*, http://www.ctr.columbia.edu/webseek/paper/.

50. C. Frankel, M. J. Swain, and V. Athitsos, "WebSeer: An Image Search Engine for the World Wide Web," *University of Chicago Technical Report # 96-14*, August 1996.

51. A. Gupta, "Visual Information Retrieval: A Virage Perspective," *Virage, Inc.*, formerly available at http://www.virage.com/papers/.

52. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying", from http://www.cs.berkeley.edu/~carson/papers/pami.html/.

53. S. Mukherjea, K. Hirata, and Y. Hara, "Towards a Multimedia World Wide Web Information Retrieval Engine", http://www.decweb.ethz.ch/.

54. V. Ogle and C. Carson, "Storage and retrieval of Feature Data for a Very Large Online Image collection", http://elib.cs.berkeley.edu/pepers/db/.

55. K. Hirata, S. Mukherjea, W. Li, and Y. Hara, "Integrating Image Matching and Classification for Multimedia Retrieval on the Web," *Proceedings of the International Conference on Multimedia Computing and Systems*, Florence, Italy, 1999.

56. J. R. Smith and S. Chang, "Visually Searching the Web for Content," *IEEE Multimedia*, Vol. 4, No. 3, July - September, 1997.

57. G. Raik-Allen, "Virage Unleashes Video-Search on Clinton Testimony," *Red Herring Online*, September 29, 1998 Available at http://www.herring.com/insider/1998/0929/ virage.html/.