Query by Sketch and Relevance Feedback for Content-Based Image Retrieval over the Web

Eugenio Di Sciascio and Marina Mongiello

Dip. Elettrotecnica ed Elettronica, Politecnico di Bari,

Via E. Orabona 4, I-70125,Bari, Italy

Corresponding author:
Eugenio Di Sciascio
Dip. Elettrotecnica ed Elettronica
Politecnico di Bari
Via E. Orabona 4, I-70125,Bari, Italy
phone +390805460641
fax +390805460410
e-mail disciascio@poliba.it

ABSTRACT

Content based Image retrieval systems are being actively investigated thanks to their ability to retrieve images based on the actual visual content rather than by manually associated textual descriptions.

This paper considers the issues related to the porting of such systems to the World Wide Web and proposes some ways to solve them.

To substantiate our ideas we propose a web-based image retrieval system that allows the user to express a query as a simple sketch portraying "what" s/he is looking for. The system relies on a three-layer relevance feedback architecture to progressively refine retrieval results according to the user's preferences. We also emphasize the use of the vector space model for features representation and the cosine distance for similarity ranking.

Performances are presented using both well-assessed information retrieval measures and subjective evaluation criteria.

1. INTRODUCTION

The advent of World Wide Web has led to a massive increase in the amount and complexity of digitized data being stored, transmitted and accessed. Largest part of this information is multimedia in nature, including images, audio and video. Images are of primary importance among these media types; they convey much information that can be evaluated at a glance and are also the basis of video representation and indexing [1].

Traditional approaches to database content modeling use alphanumeric data to represent structured documents. Multimedia documents, containing image, audio and video components, and hence unstructured, can be described by attaching textual descriptors to non-textual content, and base indexing and retrieval on such descriptors. This labeling process requires a transformation from signals to symbols. Humans can do this effortlessly, but the process requires time and it is far from objective.

Content-based image retrieval (CBIR) [2-4] information systems use information extracted from the content of images for retrieval and help the user retrieve images relevant to the contents of the query. CBIR has become in the 90's an actively researched area. A number of methodologies, techniques and tools, related to image content processing, have been studied for identification and comparison of image features in order to develop classification and retrieval systems based on (almost) automatic interpretation of image content. Complete image classification, indexing and retrieval based on the content interpretation require semantic interpretation and cannot be afforded with current technology. A surrogate of semantic interpretation is the computation of visual features that can be used as quantitative parameters for the identification of similar images. Thus, the problem of retrieving images with homogeneous content is substituted with the problem of retrieving images visually close to a target one. A way to achieve this goal is the automatic computation of features such as color, texture, shape, and spatial relationships of objects within images. Images are represented as a collection of features and retrieval is performed computing similarity in the feature space.

Several systems have been proposed in recent years in the framework of content-based retrieval, both for still images and video sequences. Although some characteristics are common to them, there are a number of approaches, mainly differing in terms of number and type of extracted features, degree of automation and domain independence, feature extraction algorithms and complexity in database population and query.

QBIC has been the first CBIR system actually built [5,6]. It allows queries to be performed on shape, texture, color, by example and by sketch using as target media both images and shots within videos. The system is currently embedded as a tool in a commercial product, Ultimedia Manager. Later versions have introduced an automated foreground / background segmentation scheme in order to improve retrieval performances. Though notable, it appears to have a substantial degree of user interaction during the database population stage.

In the Candid system [7] each image stored in the database has associated a global signature including color, texture and shape descriptors. Queries are asked by example.

The Chabot system [8] is also based on interactive feature interpretation. It performs content retrieval based only on color, and relies on a textual description for the content selection, thus matching a descriptive document, which is actually searched, with a visual browsing of the retrieved images.

The Virage Image Search Engine [9] provides an open framework for building content based image retrieval systems. The Virage Engine expresses visual features as image primitives. Primitives can be very general (such as color, shape, or texture), or quite domain specific (face recognition, cancer cell detection, etc.). The basic philosophy underlying this architecture is a transformation from the data-rich representation of explicit image pixels to a compact, semantic-rich representation of visually salient characteristics.

Domain knowledge is also used as a basis for image interpretation. In [10] an object-oriented database is provided with domain knowledge appearing in form of classes, that manages image features and operators semantics during query interpretation.

Other approaches are based on fuzzy searching, taking into account the subjective interpretation of image features [11] and domain specific image distinctive landmarks [12]. In general, databases and retrieval systems designed for specific application fields can use domain knowledge in several forms, in order to improve the classification and retrieval processes.

In [13, 14] segmenting techniques of video clips are based on content analysis for identifying the shots and the transitions between different scenes.

VisualSeek [15] proposes a feature back-projection scheme in order to extract relevant image regions allowing both content-based and spatial similarity evaluations. Other approaches have been investigated by researchers of the same group, mainly focused on the automatic indexing of images and videos from the web.

NETRA [16] computes color, shape, texture and spatial location features. It uses image segmentation to retrieve similar regions from images in the database, that is the focus is not on retrieval of similar images, but on similar regions within an image.

Photobook, developed at MIT, is a prototype system that shares various aspects with QBIC but mainly concentrates on textural features. An interesting use of texture for semi-automatic image regions labeling has also been developed there [17]. The system has been recently increasing the number of adopted features.

With reference to papers related to query by sketch, the interested reader is also referred, apart from the already cited QBIC, to QVE [18]. Also worth of note is a recent work on the use of elastic deformation of user sketches [19]. It must anyway be noticed that these works emphasize more the pattern matching problem than the retrieval by similarity one.

Also noteworthy is the work in [20], which uses wavelet-based indexing and query by sketch for color images retrieval. Here the emphasis is in avoiding any user specification but the submitted query sketch.

Approaches proposed in MARS [21-23] and [24-26] introduce relevance feedback as a distinguished aspect that can allow improving retrieval results using feedback provided by the user.

An interesting reading is [27] where a system for the retrieval of images is presented based on descriptive captions queried using natural language. This proposal goes in a direction some way opposite to the previously referred work, but it is worthy of note to be fair towards more conventional retrieval systems.

When CBIR systems are ported to the WWW several other issues arise. Largest part of currently available systems basically rely on query by example, letting the user browse into the collection until he/she finds an image visually similar to searched ones. Once such an image has been found the user can pose a query and retrieve similar images.

This process is cumbersome. As image collections grow in size this process may take a lot of time, and eventually reduce the query-retrieval process to trivial browsing. Furthermore the well-experienced problem of current Internet low speed makes the all process much tedious.

We believe that increasing user interactivity in posing queries over image repositories on the Internet, using query by sketch approach, and letting the user progressively refine the query by relevance feedback [28] is a crucial stage in order to allow an effective use of content based image retrieval techniques over the WWW. This approach has proved effective in textual retrieval systems and various authors have also reported interesting results in the CBIR framework [21-26]. Differently from text-based documents, browsing of the retrieved set of images is a fast and simple task since the relevance of an image can be stated at a glance, hence user's response to the initial query results can be almost immediate.

To address these and other issues, related to image information systems, we have started the DrawSearch [29] project with our group at Politecnico di Bari.

In this paper, we specifically discuss our approach to web-based image retrieval, which allows the user to express the query as a rough sketch and relies on relevance feedback to progressively refine retrieval results according to the user's preferences. We also emphasize another key aspect: the use of the vector space model for features representation and the cosine distance for similarity ranking. These are both taken from the textual retrieval framework, allowing a straightforward extension of our approach to the use of other features and of textual descriptors. A problem all content-based image retrieval systems share is the lack of acknowledged "ground truth" performance measures. This is mainly due to the inherent uncertainty of similarity based retrieval. Images considered similar by one user may differ in the judgement of another one. There is a growing need for acceptable test collections. In this paper, to evaluate the retrieval effectiveness and search efficiency of our approach, we reverted to measures typical of textual information retrieval systems, though we had to keep into account the inherent differences existing between images and text. We also present results obtained evaluating the retrieval effectiveness and search efficiency using well-assessed information retrieval measures.

The remaining of the paper is organized as follows. Section 2 provides an overview of the system developed to substantiate our ideas and briefly describes how image features are extracted. Section 3 is devoted to the description of the user interface and query processing. Section 4 presents the adopted relevance feedback architecture. Section 5 illustrates obtained results. The final section draws the conclusions and outlines future research directions.

2. SYSTEM OVERVIEW

In order to explore the effectiveness of the query and retrieval strategies outlined in the introduction; we implemented a prototype system. Figure 1 shows its main components.

Our system extracts as relevant features, as other systems do, color, shape and texture. In this work we concentrate only on shape and color distribution, which are currently embedded in our query by sketch interface. Retrieval by texture is currently done using a different interface.

Use of color features to index and retrieve images dates back to earlier works on color histograms. Many variants to basic histogram indexing exist, and almost all similarity-based systems include, as a relevant feature, the color distribution.

Shape is another attribute that can be used to represent image information. However it is a feature much harder to extract and describe than color, as it includes extraction of region boundaries. Our current system only allows single shapes within images.

Feature extraction is a two-fold problem in our approach. Basically the features described here are extracted off-line for images during the database population stage. Hence, though always relevant, time performance is not critical when referred to this stage. On the other hand, when the features have to be extracted from the user's sketch, time, and hence algorithmic complexity, becomes a primary issue. Furthermore, while features computation on the server side can be done with high efficiency standard compiled languages, the client side has to rely on the much less efficient Java language.

Rather than putting much processing effort into answering a query in the sharpest way, in an image retrieval system it can be reasonable to give a fast reply with a rough retrieval method (yet offering non-trivial discrimination performance), and give the user the ability to interact easily with the system by browsing the retrieved images and tuning the response through relevance feedback analysis. Differently from text-based documents, browsing of a small retrieved set is a fast and simple task since the relevance of the content can be stated at a glance, rather than by reading.

Another issue has to be pointed out. We believe that the vector space model has an intrinsic strength due to its widespread use and its reliability. It also allows an extremely simple integration with text-based information systems. Hence we considered of primary importance feature evaluation schemes that allow a straightforward implementation in this model.

2.a Color feature extraction

As hinted above, the typical approach to color indexing is based on histograms. We preferred here, for simplicity in the extraction of color information from the user's sketch, to limit the computation to the average values within 16 predefined image blocks (in a 4 by 4 arrangement) the sketch is divided into. Computing the color feature in this way is obviously not very precise; but it allows, with limited computational effort, to represent the color distribution of the most immediately visible components like large objects or background, hence also providing a limited degree of spatial information.

The adopted color model is RGB (Red, Green, Blue) [30]. Resulting data are normalized to a sum of 1 and arranged in a vector of 48 components.

The RGB model has been widely adopted because of its implementation simplicity. Despite this, the RGB model has proved unable to separate the luminance and chromatic components; furthermore values are perceptually non-uniform, i.e. perceptual changes in color are not linear with numerical changes. We are currently changing our color model to the HSB one. Anyway results reported in this paper still refer to color distribution computed using RGB.

2.b Shape feature extraction

Shape extraction in real images requires image segmentation. Although a number of segmentation algorithms have been proposed in the literature, this issue is far from being solved [30]. Most problems come from the ill-posed nature of the edge detection problem. Integration of the edge information with color and texture characteristics is almost mandatory, yet a satisfactory segmentation algorithm is still unavailable. The problem becomes harder when we need a segmentation into semantically coherent regions, i.e. objects. The images in the database were taken from repositories available in the public domain. They are of variable quality and almost all in compressed lossy formats. A segmentation algorithm has been devised, described in [31], which starts with a morphological simplification of the image and a subsequent edge enhancement. Segmentation is performed merging region growing and detected edges. The images in our prototype normally have a single object, which is not, usually, on the boundary. The algorithm uses this application knowledge to reach a main-object / background segmentation. Currently our database images belong to three domains: airplanes, cars, and ships. We obtained a correct automatic segmentation with the following percentages for the three domains: 61 % for airplanes, 89 % for cars, and 28 % for ships. Shape characteristics extraction has been performed using a modified version of the Fourier descriptors approach, proposed in [32]. Our shapes are hence assumed simply connected with borders represented by a closed curve.

The algorithm has a *O(Nc)* complexity, being Nc the number of Fourier terms considered. We currently use the real part of the lower 100 Fourier coefficients for shape description, which are arranged in a single vector. As we discard the continuous component, we obtain scale, rotation and traslation invariant coefficients. After various experiments, we decided not to weight differently higher frequency coefficients. We only normalize the coefficients with respect to each corresponding term within the collection.

3. USER INTERFACE AND QUERY PROCESSING

The user can graphically formulate joint color / shape queries by drawing a sketch portraying, to some extent, his / her information need. The user can draw on a canvas, selecting one or more colors from a color bar, to sketch the characteristics retrieved images should have. He / she can then trace the outline of a shape.

It is worth noticing that, to avoid the further burden of extracting the shape information from a color image, the user is asked to draw the shape sketch on a separate layer. This has drawbacks, as it may require drawing twice an object within the shape: the first one on the color layer and the second on the shape layer. It should be anyway noticed that the time to draw the sketch is anyway negligible with respect to an automated segmentation. A further "border" layer is provided to let the user check if the shape he / she has drawn is closed. Figure 2 shows the user interface with a query sketched on the canvas and the 9 higher ranking retrieved images.

A variety of user interfaces has been recently proposed to allow users interact via web with remote image repositories. Most of them rely on simple query by example; we have already discussed the disadvantages of this presentation type. Other Java based user interfaces have been proposed for web-based image retrieval systems, e.g. Netra and VisualSeek. In our opinion they may appear complicate enough for the casual user. Furthermore, these systems adopt an active interface just to select presegmented regions, as in Netra, or to specify percentages of color present in target images, as in VisualSeek. The system proposed in [20] was not designed for the web, but it is worth of note as it adopts query by sketch. The problem is that the sketch is treated as a new "normal" image over which feature extraction is performed via multiresolution analysis. Our query by sketch interface relies on different layers; thus it overcomes the segmentation problem, quite simplifying the processing.

Other systems seem simply unsuitable for the web framework. The "El Nino" system [33], for example, proposes an interesting approach to user interaction for query refinement. The problem is that it foresees a user interface showing, for each iteration, some hundred images; quite a challenge to the average Internet performances, even adopting small thumbnails.

We tried to emphasize simplicity, asking the user basically to sketch what he/she is looking for and nothing else, and endowed the query interface of a processing capability. We believe that the initial query is always approximated and the workload needed to refine the original query is not always worth the actual retrieval accuracy improvement.

Various models have been proposed for similarity analysis in image retrieval systems. We have already stated that our representation shares many properties with the vector space model, which is widely used in textual document retrieval systems.

The vector space model [6] is based on the association of *term vectors* to documents, each vector representing a specific document by holding information about the index terms or keywords associated to it. Such information may appear simply as a set of present/not present flags, but more often it is a measure (weight) of the ability of each index term to discriminate the document within the collection. Weights are computed by considering how often a term appears in the document and in the whole document collection. Frequent terms are usually more meaningful, unless they are very common, therefore unable to provide an effective document discrimination.

Retrieval is performed by measuring the distance (or the similarity), in the n-dimensional space defined by the index terms, between the term vector of the query and the term vectors of the documents.

In our image retrieval model, feature vectors play almost the same role that term vectors play in text retrieval, holding normalized values of the image features as indexing information. However, differently from the vector space model, we do not weight the features against the whole image collection. Weights are assigned on the basis of the distribution of features in the image, independently from the collection content.

The reason for this difference comes from the different user perception of image similarity with respect to text similarity, and to the different level at which the retrieved items are evaluated: visual for the images, semantic for the text.

Image similarity is evaluated on visual properties, and may be verified at a glance. It is therefore independent from the image collection size and variety. The features are used to find similarities rather than to discriminate differences. On the other hand, text reading to verify the adequacy of the retrieved documents is a long process, therefore a text retrieval system must be provided with good discrimination capabilities, relying on the use of words to describe the document meaning and not its appearance.

Our approach associates to each image (or query sketch) two separate vectors, for the color and shape feature, respectively. Hence, two similarity functions are computed: simC*(R,Q)* and simS*(R,Q)*, respectively accounting for color and shape, both computed using the *cosine* similarity coefficient, defined as:

$$\mathrm{sim}(R,Q) = \frac{\sum r_i q_i}{\sqrt{\sum r_i^2 \times \sum q_i^2}} \qquad (1)$$

with $R = (r_0, r_1, ..., r_n)$ a database image feature, and $Q = (q_0, q_1, ..., q_n)$ the query image feature. The number of terms in the tuples is n = 48 for *SimC* and n = 2 $\times$ Nc for *SimS*.

The resulting coefficients are merged to form the final similarity function as a linear combination:

$$\mathrm{sim}(R,Q) = \alpha \times \mathrm{simS}(R,Q) + \beta \times \mathrm{simC}(R,Q) \qquad (2)$$

where α and β are weighting coefficients. The weights obviously lead to increase or decrease the contribution of a feature with respect to the other. To better characterize the user's information need, these coefficients are dynamically modified during the relevance feedback stage.

4. THREE-LAYER RELEVANCE FEEDBACK ARCHITECTURE

In information retrieval systems, the retrieved documents do not match exactly the user expectations. Indicators like *recall* and *precision* are introduced to evaluate the quality of the retrieval process with respect to an ideal exact match.

Uncertainty is even more present in image retrieval, due to the weaker correspondence between the computed features and the image content perceived by the user. In other words, the system may not match the user perception of similarity.

For this reason there is a growing interest in the image database community towards methodologies and techniques for relevance feedback [21-26]. Relevance feedback is the mechanism, widely used in textual information systems [28], which allows improving retrieval effectiveness by incorporating the user in the query-retrieval loop. Depending on the initial query the system retrieves a set of documents that the user can mark as relevant or not-relevant. The system, based on the user preferences, refines the initial query retrieving a new set of documents that should be closer to the user's information need.

Text-based information retrieval systems rely on techniques such as relevance feedback to refine results of a query through the interaction with the user. Assuming a text retrieval system based on the vector space model, the documents and the query are represented by term vectors, whose elements hold information about the presence or the relevance (weight) of index terms. Relevance feedback analysis is usually done in six steps:

1. Given the generic *k-th* query, a term vector $\overline{Q}^{(k)}$ associated to the query is computed.
2. $\overline{Q}^{(k)}$ is compared with the term vectors of the documents in the database.
3. Resulting documents more similar to the query are ranked according to a suitable metric.
4. The user marks some selected documents as relevant or not relevant.
5. The term vector $\overline{Q}^{(k)}$ is modified using information provided by the user: the weight of terms present in the vectors of relevant documents is increased, while the weight of terms present in the documents marked as not relevant is decreased.
6. A new query is submitted through the modified query vector $\overline{Q}^{(k+1)}$.

In our prototype system we allow the user to improve retrieval results by selecting, among the topmost ranked retrieved images, the ones he/she considers relevant and those considered not relevant. Leaving an image unselected marks it as "don'tcare" and it does not contribute to the relevance feedback process.

Our model adopts a three-layer relevance feedback architecture. The first one basically follows the approach typical of textual documents. A new query is computed by combining the feature vectors of the original query with the ones of the relevant and not relevant images.

In practice, the modified query is computed by adding to the feature vector $\overline{Q}^{(k)}$ associated to the *k-th* query, the N_{rel} feature vectors $\overline{X}^{(k)}$ associated to relevant images and subtracting the M_{notrel} not relevant ones $\overline{Y}^{(k)}$, respectively weighted with suitable δ and ε coefficients:

$$\overline{Q}^{(k+1)} = \overline{Q}^{(k)} + \delta \sum_{i=1}^{N_{rel}} \overline{X}_i^{(k)} - \varepsilon \sum_{j=1}^{M_{notrel}} \overline{Y}_j^{(k)} \quad (3)$$

The modification is performed separately on the two components, color and shape. The vector values are normalized to a sum of 1, in order to retain compatibility with the feature vectors associated to the image collection. The modified query is then used in a new retrieval step, performed as described in Section 4.

It is worth noticing that, though the application of relevance feedback in our framework has proved to be effective, a conceptual difference exists with respect to the same approach applied on textual documents. In that case, an index term (i.e., a word) either appears in a document, with some weight, or it does not appear. It is usual to set up a query with a very small subset of the words appearing in the whole document collection. By applying relevance feedback, new index terms may be considered, or some terms may be excluded from the query.

In image retrieval, basically all index terms (the feature vectors components) are present in the whole collection, though with different contributions. The relevance feedback operates by changing the amount of contribution (i.e., the weight) of the feature components.

The above-described modifications of the query operate on the single feature. The second relevance feedback layer tries to combine different features.

The overall computed similarity coefficients may differ from the user expectation, which may tend to concentrate on a particular feature. The weights used in the linear combination that provides the final similarity value identify the importance a feature has. When the query is first posed we assume both features have the same relevance: $\alpha = \beta = 0.5$; this assumption is not precise, anyway experiments carried out slightly modifying these initial values did not lead to a substantial modification of retrieval results.

As the user refines his/her query with relevance feedback weighting coefficients are modified according to the following expression:

$$\alpha = \begin{cases} 0.6 & \textbf{\textit{if}} \;\; S^{(k)} > C^{(k)} \\ 0.4 & \textbf{\textit{if}} \;\; C^{(k)} > S^{(k)} \end{cases} \qquad \beta = \begin{cases} \mathbf{0.4} & \textbf{\textit{if}} \;\; S^{(k)} > C^{(k)} \\ 0.6 & \textbf{\textit{if}} \;\; C^{(k)} > S^{(k)} \end{cases} \tag{4}$$

Where $S^{(k)}$ is the number of relevant and retrieved images having simS*(R,Q)* > simC*(R,Q)* and $C^{(k)}$ is the number of relevant and retrieved document having simC*(R,Q)* > simS*(R,Q)*, both at the *k-th* relevance feedback iteration.

In other words, if the user tends to concentrate on a single feature, i.e. he/she selects as relevant only images having a higher contribution from one of the features, the systems strengthens the contribution of that feature in the similarity measure.

The third layer operates on the domain level. Most image collections available in the public domain or through the commercial and professional distribution channels are organized in sub-collections (directories), each covering a separate theme. Hence, selection of the relevant collection limits in no way the generality of the approach.

Therefore we operate as follows: if the user concentrates his/her interest on images of a single domain, i.e. all relevant images belong to a single category, we retrieve in the following relevance feedback stages only images belonging to that domain. Otherwise no action is taken.

Relevance feedback has been introduced to improve retrieval accuracy. If the results of the query are not satisfying, the user can select, within the retrieved subset, the images he/she considers relevant. Other approaches introduce a score to represent the degree of relevance a retrieved picture has; we limit here the choice to a relevant/not-relevant one, in order to ease user's interaction. An image can be considered "don't care" if the user is not certain about its degree of relevance and it is not considered in the relevant feedback stage.

Figure 3 tries to emphasize how relevance feedback works. With reference to the query in figure 3.a and the corresponding retrieval results of figures 3.b, it can be noticed how images 1, 3, and 5 were considered by the user relevant, images 4, 6, 7, and 9 were marked as not-relevant and 2 and 8 were marked "don'tcare". Retrieval results after the relevance feedback processing are shown in figure 3.c. It can be noticed how images 7, 8, and 9 were discarded, image 3 became the highest ranking and image 4 in figure 3.c was ranked as 8. New images, more similar to the query were retrieved. It is interesting to notice that image ranked as 6 in figure 3.b not only was not discarded, but received a higher rank in the following step. Yet it is clear that this image is quite similar in shape to the originally highest in rank, selected as relevant, in figure 3.a. As the systems tends in this case to privilege the shape feature the image was not withdrawn. Figure 3.d shows results again applying relevance feedback on the retrieved set. It is noteworthy that other images more visually similar to the original query appear.

Figure 4 shows the behavior of the system using the third relevance feedback layer: the user sketch in figure 4.a portrays a drawing of a red car on a light-blue landscape. Retrieval results in figure 4.b show that though 6 out of 9 topmost relevant images are red cars, 3 retrieved images are planes on a light-blue landscape. Figure 4.c shows retrieval results after relevance feedback obtained by selecting as relevant only images picturing cars.

5. EXPERIMENTS AND RESULTS

DrawSearch, the system described in this paper, is currently still a prototype being evaluated and improved. It is accessible at http://deecom03.poliba.it/DrawSearch/DrawSearch.html.

To evaluate the retrieval effectiveness and search efficiency of a CBIR system we believe it is necessary to revert to typical information retrieval measures, though keeping into account differences in the document types. We adopted the evaluation criteria presented by Salton and Mc Gill in [34], namely recall, precision, time, effort, presentation and coverage.

The presentation, i.e. the user's interface, has been already described in a previous section. It is subdivided into two parts. It is based on a Java applet where the user can initially draw a sketch in terms of color distribution and shape of an object. Features are computed and transmitted to the server that performs indexing and retrieves the best matching images in decreasing order of similarity. Figs. 1 and 2 show the Java applet user interface and two examples of query by sketch. Retrieved images are displayed in a standard html page in a ranked order, upper left is the most relevant, lower right the least one (Please notice, the server writes on its own disk the first retrieved html page and names it with the client host name. Pages retrieved applying relevance feedback are dynamically built with a CGI. This difference is necessary to ease interface with the Java client and to reduce the burden on the client side). Checkboxes allow selecting relevant or not-relevant images. Images left unmarked are not considered in the relevance feedback process.

With reference to the effort, a user may take approximately a couple of minutes to draw an acceptable quality sketch.

The system stores images and associated feature vectors as plain files, since the initial aim of the prototype is at testing the effectiveness of the retrieval approach without time performance concern, and the code is far from being optimized. Our prototype is currently hosted on an old HP720 ws with a 68030 CPU. The response time is approximately T_c=3÷5 secs. The Java applet takes approximately T_i=30÷60 secs (on a Pentium 100 MHz CPU with 32 MB of RAM) to compute the coefficients of the features. The computation time is dominated by the shape coefficients extraction, which depends on the shape size. The total time T_q the system takes to answer a query can be expressed as:

$$T_q = T_i + T_l + T_n + T_c + T_d \tag{5}$$

where: T_l is the time to load Java classes on the client; T_n is the time to transmit query coefficients from the client to the server and T_d is the time to display the retrieved set of images. These times obviously depend on the network efficiency.

The collection coverage is currently limited to the approximately 400 images stored in the database. Considered features are shape and color distribution. Only one shape per image is assumed. The collection includes images of planes, ships and cars. Largest part of images has been taken from repositories available in the public domain, in particular from the U.S. Air Force and U.S. Navy archives. Feature extraction has been performed on the images in PPM (24-bit) format, while the images have been stored in GIF/JPG format for browsing and display of results. The display of query results involves thumbnails that may make some features less visible.

Subjectivity is always present in CBIR, and it is not a simple task to evaluate results against human perception. To reach an acceptable degree of objectivity in the further evaluations some actions were undertaken. The tests were basically sub-divided into two parts.

The first one refers to experiments based on query by example, i.e. similarity ranking against a given query image and improvements obtained with relevance feedback. We arbitrarily selected 160 images from our database. Among these we selected 50 images to be used as queries. We asked 20 volunteering students (14 males, 6 females) from an entry level Computer Science course to select, for each of the 50 query images, up to 16 images they considered relevant according to their own visual perception. We did not specify any further constraint. We received 18 compiled questionnaires. We examined them and had to discard two more questionnaires that clearly showed a completely random selection of relevant images. The remaining set was processed with standard statistical techniques to extract the higher-ranking images for each of the 50 queries. 12 queries were matched with 16 relevant images; 27 queries where matched with 15-11 relevant images; 9 queries where matched with 10-5 relevant images; 2 images were matched with 4-2 relevant images.

Recall and precision measures were computed for the 50 image queries using micro and macro averaging techniques as proposed in [35]:

$$\begin{aligned} m_{p1} &= \frac{\sum r_i}{\sum n_i} \quad & m_{p2} &= \frac{\sum (r_i/n_i)}{q} \\ m_{r1} &= \frac{\sum r_i}{\sum t_i} \quad & m_{r2} &= \frac{\sum (r_i/t_i)}{q} \end{aligned} \tag{6}$$

where:

q is the number of queries; r_i is the number of relevant and retrieved document for the query i; n_i is the number of retrieved documents for the query i; t_i is the number of relevant documents for the query i.

m_{p1} and m_{r1} represent the microaverages for precision and recall, respectively; m_{p2} and m_{r2} represent the macroaverages for precision and recall, respectively.

We then submitted the q=50 test queries to the system. Though our system ranks retrieved images, we decided not to pose a numerical threshold on the retrieved set. Instead we always retrieved 16 images, that is n_i=16 for the entire test set. Though this may appear a shortcoming, there are various reasons for our choice. Almost one quarter of our query test set had 16 corresponding relevant images. Differently from textual documents, relevance or irrelevance of a retrieved image can be evaluated at a glance. Furthermore this approach allowed a better evaluation of relevance feedback results.

Several relevance feedback evaluation measures have been proposed in the literature. Yet they all need some kind of reworking to adapt them to the CBIR framework. In this work we limited our evaluation to measuring precision and recall scores after relevance feedback. More in detail we performed, on the same group of images, two sets of experiments allowing up to 3 rounds of relevance feedback for each query. It is fair to notice that the selection of not-relevant or "don'tcare" retrieved images was performed by the authors.

The first set of experiments was performed on the system using just the one layer relevance feedback, i.e. using a fixed 0.5 weight for both shape and color distribution; the second one using relevance feedback both on the single feature weights and on their combination; the third one with all layers.

Results for recall and precision scores for the three sets of experiments are shown in Table I. It is interesting the improvement obtained with relevance feedback. It is also worth noticing that the performance increase obtained with the two-layer approach is practically due only to queries with planes and cars. We believe that this is due to the fact that our volunteering students concentrated their attention on the model of aircraft or car, neglecting the color distribution component.

The second part of our experiments was devoted to test retrieval performances with the query by sketch approach. We asked five volunteering users (a graphics designer, a student, a teacher, an engineer, and a secretary) to randomly select 10 images out of our test set. For each of these we asked them to select 15 images they considered similar to the given one, without any particular ranking. The users were then asked to draw a sketch picturing the 10 selected images, which were submitted as queries. During this stage they were not allowed to watch the originally selected images. They were all allowed to perform up to five rounds of relevance feedback.

The results from this second set of tests are shown in table II.

As a general comment, the application of relevance feedback in this second set of experiments showed an improvement in the retrieval effectiveness greater than the one obtained on the first test set. Obviously, better retrieval results were obtained when users focalized their attention on visual properties, rather than on the particular model of aircraft or car.

6. CONCLUSION

Currently available large image repositories require new and efficient methodologies for query and retrieval. Content-based Image retrieval appears a promising research direction to increase the efficiency and accuracy of unstructured data retrieval.

In this paper we have discussed issues related to the use of such systems in the World Wide Web framework. We have proposed a system whose interface is a Java applet that allows to immediately portrait "what" the user is looking for. Due to the intrinsic imprecision of such a visual query we have also proposed a three-layer relevance feedback approach to let the user interactively improve retrieval effectiveness.

A performance evaluation has been carried out, and it has been presented using both well-assessed information retrieval measures and subjective evaluation criteria.

Our current work is in aimed at the integration in our model of textural features and of textual descriptors, the improvement of the relevance feedback mechanisms and the automatic indexing of images available on the web.

ACKNOWLEDGEMENTS

We wish to thank G. Mingolla for his most valuable help and A. Celentano for the helpful scientific discussions. We also wish to thank the volunteering users for their help and patience.

REFERENCES

1. Communications of the ACM. (1997) 40, 12.
2. ACM Multimedia Systems, special issue on Multimedia Databases. (1995) 3, 5/6.
3. V. N. Gudivada, V. V. Raghavan (1995). Picture Retrieval Systems: A Unified Perspective and Research Issues. Computer Science Technical Report CS9503, Ohio University.
4. IEEE Computer(1995) special issue on Content Based Image Retrieval, 28, 9.
5. M. Flickner et al. (1995) Query by Image and Video Content: The QBIC System. IEEE Computer, 28, 9, 23-31.
6. W. Niblak et al. (1993). The QBIC project: Querying images by content using color, texture, and shape. In Storage and Retrieval for Image and Video Databases, SPIE, vol. 1908, 173-182.

7. P. M. Kelly, T. M. Cannon, D. R. Rush (1995). Query by image example: the CANDID approach. In Storage and Retrieval for Image and Video Databases, SPIE, vol. 2420, 238-248.

8. V. E. Ogle, M. Stonebrakes. (1995) Chabot: retrieval from a relational database of images. IEEE Computer, 28, 9, 40-56.

9. R. Bach et al. (1996). The Virage Image Search Engine: An open framework for image management. In Storage and Retrieval for Still Image and Video Databases, SPIE, vol. 2670, 76-87.

10. A. Yoshitaka et al. (1994). Knowledge-Assisted Content-Based Retrieval for Multimedia Databases. IEEE Multimedia, 1, 4, 12-21,

11. J. K. Wu, A. S. Narisimhalu (1994). Identifying faces using multiple retrievals. IEEE Multimedia, 1, 2, 27-38..

12. J. Bach, S. Paul, R. Jain. (1993). A Visual Information Management System for The Interactive Retrieval of Faces. IEEE Trans. on Knowledge and Data Engineering, 5, 4, 619-628,

13. S. W. Smoliar, H. Zhang. (1994) Content-Based Video Indexing and Retrieval. IEEE Multimedia, 1, 2, 62-72.

14. H. Zhang, A. Kankanhalli, S.W. Smoliar. (1993) Automatic Partitioning of Full Motion Video. ACM Multimedia Systems, 1, 1, 10-28.

15. J.R. Smith, S.F. Chang (1996). VisualSEEK: a fully automated content-based image query system. In: Proc. of ACM Multimedia'96.

16. W.Y. Ma, B.S. Manjunath (1997). NETRA: A toolbox for navigating large image databases. In: Proc. IEEE ICIP '97.

17. R.W. Picard, T.Kabir (1993). Finding similar patterns in large image databases. In: Proc. ICASSP'93.

18. K. Hirata, T. Kato. (1992). Query by visual example, content based image retrieval. In: Lecture Notes in Computer Science, vol. 580.

19. A Del Bimbo, P. Pala. (1997). Visual image retrieval by elastic matching of user sketches. IEEE Trans. PAMI, 19, 2.

20. C. E. Jacobs, A. Finkelstein, D. H. Salesin. (1995). Fast Multiresolution Image Querying. Proc. of SIGGRAPH 95.

21. S. Mehotra et al. (1997). Supporting content-based queries over images in MARS. In: Proc. of IEEE Conf. on Multimedia Computing and Systems.

22. Y. Rui, T.S.Huang, S. Mehrotra. (1997). Content based image retrieval with relevance feedback in MARS. Proc. of IEEE ICIP'97.

23. Y. Rui, T.S.Huang, S. Mehrotra. (1997). An integrated relevance feedback architecture im multimedia information retrieval. IFP Tech. rep. YR06-97, Beckman Institute, UIUC.

24. A. Celentano, E. Di Sciascio. (1998). Features Integration and Relevance Feedback Analysis in Image Similarity Evaluation. Journal of Electronic Imaging, special section on Image and Video Databases, 7, 2.

25. E. Di Sciascio, G. Piscitelli, A. Celentano. (1998). Textural Features and Relevance Feedback in Image Retrieval, in Visual Database Systems 4,Y. Ioannidis, W. KlasEds., Chapman and Hall.

26. H.G. Stark (1996). On image retrieval with wavelets. Journal of Imaging Systems and Technology, 7, 200-210.

27. E. J. Guglielmo, N. C. Rowe (1996). Natural-language retrieval of images based on descriptive captions. ACM Trans. on Information Systems, 14, 3, 237-267.

28. G. Salton. (1989). Automatic Text Processing, Addison Wesley.

29. E. Di Sciascio, M. Mongiello. (1999). DrawSearch: a Tool for Interactive Content-Based Image Retrieval over the Net. In: Storage and Retrieval for Image and Video Databases VII, SPIE vol.3656.

30. W.K. Pratt (1991). Digital Image Processing. J. Wiley & Sons Inc., Englewood Cliffs, NJ.

31. E. Di Sciascio, C. Guaragnella, M. Mongiello (1999). Automatic segmentation and interactive retrieval for image databases. Tech. Rep. 31/s D.E.E. Politecnico di Bari.

32. Y. Rui, A.C. She, T.S. Huang (1996). Modified Fourier descriptors for shape representation-a practical approach. In: Proc. of 1 workshop on image databases and multimedia search.

33. S. Santini, R. Jain (1999). The "El Nino" Image Database System. In: Proc. IEEE Intl. Conf. Multimedia Computing and Systems.

34. G. Salton, M.J. Mc Gill. (1983). Introduction to modern information retrieval. McGraw-Hill eds.

35. J. Tague Sutcliffe (1992). The pragmatics of information retrieval experimentation, revisited. Inform. Proc. Manag., 23, 2.

List of captions:

Figure 1. DrawSearch system architecture.

Figure 2. a) User interface for query sketching with a sketch drawn on the canvas; b) retrieval results.

Figure 3. Query and retrieval results for a sketched query. a) sketch drawn on the canvas; b) retrieval results; c) retrieval results after relevance feedback; d) retrieval results after a second round of relevance feedback.

Figure 4. Query and retrieval results for a sketched query pointing out the effect of the third relevance feedback layer. a) query sketch; b)retrieval results; c) results obtained selecting as relevant only images belonging to a single domain.

TableI. Precision and recall scores

Table 2. Retrieval results with query by sketch

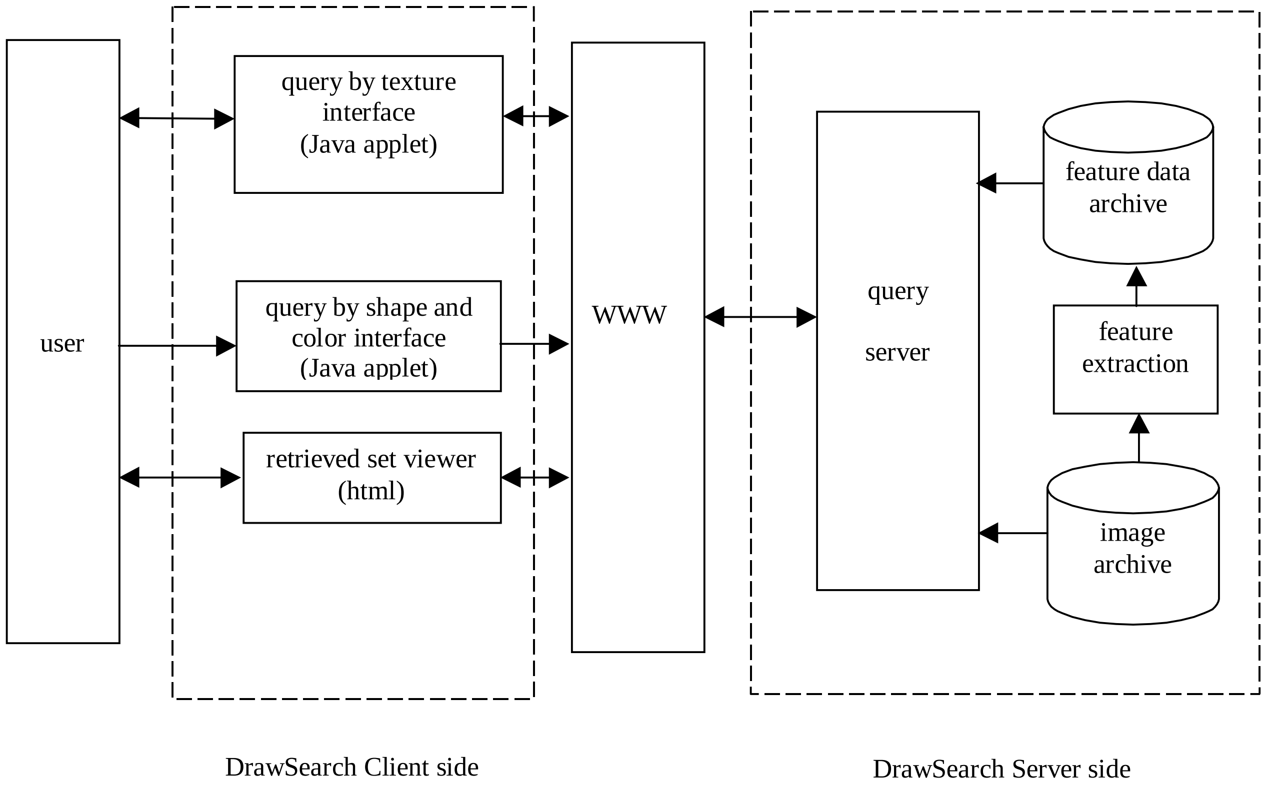
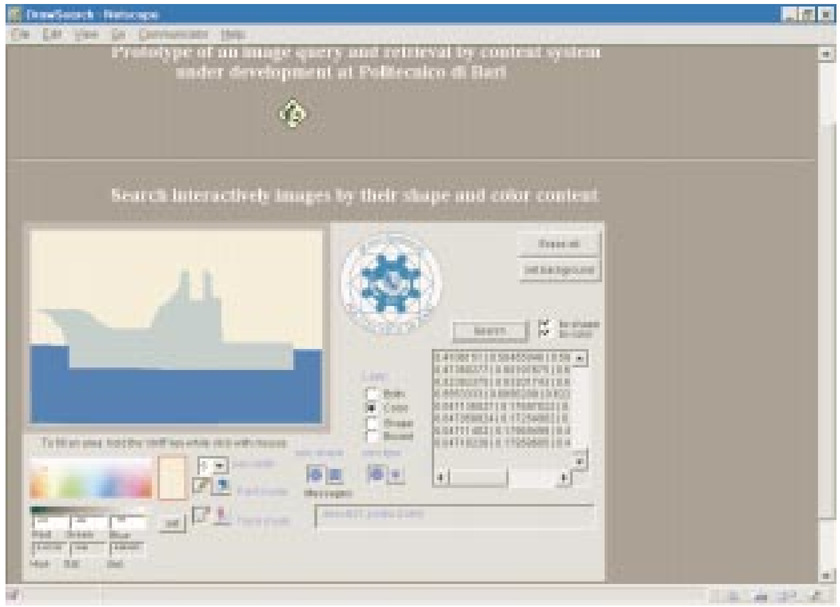
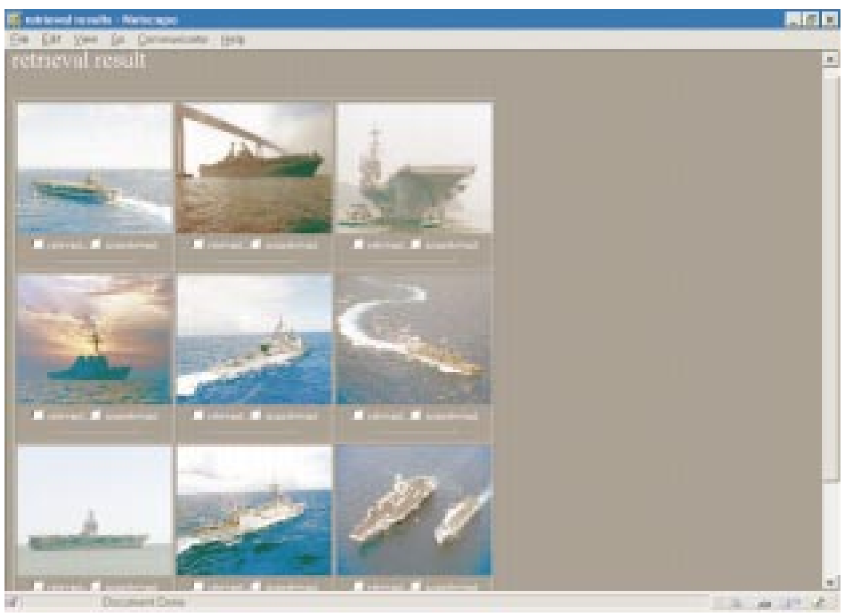

user
query by texture interface (Java applet)
query by shape and color interface (Java applet)
retrieved set viewer (html)
WWW
query server
feature data archive
feature extraction
image archive
DrawSearch Client side
DrawSearch Server side

Figure 1


Prototype of an image query and retrieval by content system
under development at Politecnico di Bari
Search interactively images by their shape and color content

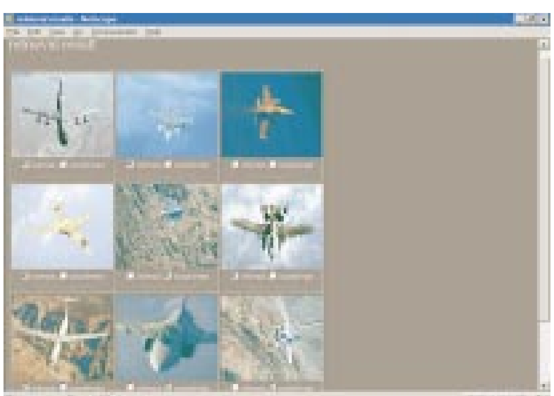
a)


retrieval result

b)

Figure 2

a) b)



c) d)

Figure 3


Prototype of an image query and retrieval by content system
under development at Politecnico di Bari
Search interactively images by their shape and color content
retrieval result
retrieval result

a)

b)

c)

Figure 4

TableI

Queries	$m_{p1} \equiv m_{p2}$	m_{r1}	m_{r2}
results without relevance feedback	0,495	0,649	0,659
results with single layer relevance feedback	0,593	0,778	0,787
results with two-layer relevance feedback	0,637	0,836	0,842
results with three-layer relevance feedback	0,653	0,857	0,871

Table 2

User	R-Avg.	RF-Avg.	Q-Pres.	min-MAX	Rmin-MAX
designer	6,9	10,2	8	4-13	7-14
student	5,8	10,5	8	2-10	6-13
teacher	6,5	9,3	6	3-10	5-13
engineer	4,9	8,5	7	0-11	0-12
secretary	5,7	10,4	9	3-12	6-15

- R-Avg. Average relevant and retrieved images w/o rel. feed.
- RF-Avg. Average relevant and retrieved images after rel. feed.
- Q-Pres. Presence of the query within the retrieved set.
- min-MAX minimum and maximum scores in the retrieved set w/o rel. feed.
- Rmin-MAX minimum and maximum scores in the retrieved set after rel. feed.