Feature Integration and Relevance Feedback Analysis in Image Similarity Evaluation

Augusto Celentano
Università Ca' Foscari di Venezia, Dipartimento di Matematica Applicata e Informatica
Via Torino 155, I-30173 Mestre (VE), Italia

Eugenio Di Sciascio
Politecnico di Bari, Dipartimento di Elettrotecnica ed Elettronica
Via Orabona 4, Bari I-70125, Italia

Abstract

In this paper we describe the results of a study on similarity evaluation in image retrieval using color, object orientation and relative position as content features, in a framework oriented to image repositories where the semantics of stored images are limited to a specific domain.

The focus is not on a complete description of image content, which is supposed to be known to some extent, but on the extraction of simple and immediate features that can assure, through their combination, automated image analysis and efficient retrieval. Relevance feedback is introduced as an effective way to improve retrieval accuracy.

A simple prototype system is also introduced that computes feature descriptors and allows users to put queries, browse the retrieved images, and refine the results through relevance feedback analysis.

Keywords: image feature computation, retrieval by image content, vector space model, Hough transform, relevance feedback.

1 Introduction

Traditional approaches to database content modeling use alphanumeric data to represent documents. Multimedia documents, containing image, audio and video components, can be described by attaching textual descriptors to non textual content, and base indexing and retrieval on such descriptors.

Limiting our discussion to image databases, an ideal situation would be the one in which queries to an image database should refer to the image content, and returned images should be ranked according to the degree of content matching.

Most DBMS, when used to hold image repositories, are based on descriptive textual legends and structured parameters. A realistic approach has to cope, at least in the next years, with these kind of systems.

Recently 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.

Content based information retrieval (CBIR) is now a widely investigated issue that aims at allowing users of multimedia information systems (MMIS) to retrieve images coherent (to some extent) with a graphic query or with a reference image [1-3]. A way to achieve this goal is the automatic computation of features such as color, texture, shape, and position of objects within images, and the use of the features as query terms. Simple systems tend to rely on a representation that is based on numerical feature vectors, and can use retrieval and ranking methodologies taken from the well assessed text retrieval framework, such as the vector space model [4].

The approach we consider in this paper is oriented to retrieving images from a thematic database, where the semantic content of the images is limited to a specific domain. 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. While retrieving images for professional applications like publishing, medical care, environment sciences, education (as a few examples) the need for a pre-selection of the relevant collection limits in no way the generality of the approach.

However, the approach can be extended to a generic database by combining queries based on textual legends or other objective attributes in order to retrieve a subset of the images, and by completing the content oriented search on a coherent set of data.

We assume that the layout and the color distribution of the image content are key attributes in defining the similarity between images. Layout can be defined in terms of the orientation and position of objects in the image. The visual coherence of a set of similar images should strongly depend on these features.

An aim of this work is to provide a set of extremely simple features that can allow an automated image analysis, hence without user intervention, and fast retrieval. In [5] the authors have widely discussed issues related to the orientation features. Here we extend that approach to include color distribution and object position, and introduce relevance feedback as an effective method to refine query results. We also tried to emphasize the extraction of features that lend themselves to be simply embedded into currently available DBMS.

As a matter of fact, other, more precise, features and methodologies have been recently proposed for image retrieval, but they rely on heavy computational burden, and hence require time and resources. The workload is not always worth the actual retrieval accuracy improvement

A prototype system has been built to demonstrate image feature computation and image retrieval. The proposed approach has been tested on a number of public domain image collections, each collection addressing a uniform application domain, e.g., aircraft, cars, and so on.

The paper is organized as follows: in Section 2 we overview the relevant literature. Section 3 addresses the basic techniques that have been used during feature extraction. Sections 4 and 5 describe respectively the image analysis phase and the query processing. Relevance feedback analysis is discussed in Section 6. Section 7 evaluates the approach by discussing the results obtained with a prototype system. The conclusions are summarized in Section 8.

2 Related work

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 different approaches, mainly differing in terms of number and type of extracted features, degree of automation and domain independence, feature extraction algorithms and processing complexity in database population and query.

The QBIC system [6,7] allows queries to be performed on shape, texture, color, directly, by example and by sketch using as target media both images and shots within videos. Anyway it appears to require a substantial level of human interaction during the database population for features that require the interpretation of the image semantics, like shapes and foreground–background identification. The system is currently embedded as a tool in a commercial product, Ultimedia Manager.

In the Candid system [8] each image stored in the database has associated a global signature including color, texture and shape. Queries are asked by example.

The Chabot system [9] is also based on interactive feature interpretation. In its current version 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. While QBIC produces a ranking of retrieved images, Chabot returns a flat set of images that the user can browse.

The Virage Image Search Engine [10] 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 [11] an object-oriented database is provided with domain knowledge appearing in form of classes, that manage image features and operators semantics during query interpretation.

Other approaches are based on fuzzy searching, taking into account the subjective interpretation of image features [12] and domain specific image distinctive landmarks [13]. 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 [14,15] segmenting techniques of video clips are based on content analysis for identifying the shots and the transitions between different scenes.

An interesting reading is [16] 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.

3 Basics

3.1 The Hough Transform

The Hough Transform [17,18] has been widely used in pattern analysis and recognition, and automated lineament detection. It basically transforms points of a two-dimensional space into a sinusoidal curve in the transformed domain.

A straight line in the starting domain corresponds to an accumulation point crossed by a number of sinusoidal curves. It is then possible to find main directions of the image and/or of objects within the image by looking for maximum points within the Hough Transform domain. Its algorithmic fundamentals are hereafter outlined.

An arbitrary straight line in the two-dimensional Cartesian space (x,y) can be represented as a point in a parameters space (Hough space) where a line is identified by the angle a and the distance r, and the equation of a straight line is defined as:

$$r = x \cos(a) + y \sin(a) \tag{1}$$

Considering its direction within the interval [0°,179°] the line can be uniquely identified as a point in the Hough space.

We can then transform the points (x_i,y_i) belonging to the line into sinusoids in the Hough space defined as:

$$r = x_i \cos(a_\mathrm{i}) + y_i \sin(a_\mathrm{i}) \tag{2}$$

Sinusoids corresponding to collinear points have a common point of intersection. This point of coordinates (r,a) in the Hough space defines a straight line in the Cartesian space as in (1).

The implementation is based on the conversion between two spaces: the *line space* (x,y) where the image is, and the *Hough space* (r,a). Both are implemented as 2D arrays, where indices represent the coordinates, and values represent respectively the image points and the number of converted points sharing the same coordinates.

Each point in the line (Cartesian) space is converted into the Hough (radial) space by the transformation:

$$x = r \cos(a), \;\; y = r \sin(a) \tag{3}$$

For every edge point (x,y), the corresponding (r,a) coordinates are computed. Then, the value associated with (r,a) point in the Hough space is incremented by one.

Once this procedure has been applied for all points in the line space, the Hough space is scanned to find local maxima, each maximum corresponding to a line. Then, that line is taken out of the Hough space, and the next highest value is found.

The procedure is repeated until all lines are found, within a threshold value that filters low values, corresponding to short segments and isolated points of the original image.

3.2 The HVC color space

The detection of regions matching a given color feature is a frequently required task in image processing applications. Various color identification schemes have been proposed and used. The RGB (Red, Green, Blue) 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.

The HVC (Hue, Value, Chroma) color model completely separates the luminance and chromatic components representing with Hue the color type, with Value the luminance, and with Chroma the color purity.

The transformation from RGB model to HVC can be performed in several ways; in this work, following the approach in [19] the transformation is obtained through the CIE L*a*b* model [20].

Assuming a 24 bit per pixel (8 bit each color) context, the RGB components are transformed into the CIE xyz components using the following formulas:

$$X = 0.607*R + 0.17*G + 0.201*B \tag{4}$$
$$Y = 0.299*R + 0.587*G + 0.114*B \tag{5}$$
$$Z = 0.066*G + 1.117*B \tag{6}$$

then transforming through CIE L*a*b*, the HVC values are finally obtained:

$$H = arc\tan\left(\frac{200\times\left[\left(\frac{Y}{Y_0}\right)^{\frac{1}{3}}-\left(\frac{Z}{Z_0}\right)^{\frac{1}{3}}\right]}{500\times\left[\left(\frac{X}{X_0}\right)^{\frac{1}{3}}-\left(\frac{Y}{Y_0}\right)^{\frac{1}{3}}\right]}\right) \tag{7}$$

$$V = 116\times\left(\frac{Y}{Y_0}\right)^{\frac{1}{3}}-16 \tag{8}$$

$$C = \sqrt{\left(500\times\left[\left(\frac{X}{X_0}\right)^{\frac{1}{3}}-\left(\frac{Y}{Y_0}\right)^{\frac{1}{3}}\right]\right)^2+\left(200\times\left[\left(\frac{Y}{Y_0}\right)^{\frac{1}{3}}-\left(\frac{Z}{Z_0}\right)^{\frac{1}{3}}\right]\right)^2} \tag{9}$$

where X_0, Y_0, Z_0 are the reference values for pure white.

4 Feature extraction

The approach of this work is based on the assumption that the content of the image collection is to some extent homogeneous, hence we concentrate our attention on the features, simple and immediate, that may differentiate images related to a common subject.

Color is a typical, well acquainted feature; almost all similarity based systems include, as a relevant feature, the color distribution.

The relative position of objects within the image is a far less investigated feature. In many systems, e.g. QBIC, it is empirically assumed that objects tend to be in the center of the image. While this is true in many cases, it should not be assumed as a generally valid rule. Rather, the objects position should be taken as one of the elements capable of discriminating different images.

Orientation of image objects is the third feature we consider. It is computed by analyzing the directions of the image edges, and computing an angular histogram.

Besides the identification of simple features, the other key point we emphasize in the design of the system is simplicity. For simplicity we mean that, even if it may have a cost in terms of efficiency, the degree of human interaction must be kept at the lowest level, and that queries must be processed quickly.

Rather than putting much processing effort into answering a query in the sharpest way, in an image retrieval system it is 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 image collections is fast and easy since the relevance of the content can be stated at a glance, rather than by reading.

To this aim, for example, we do not build a color histogram, like other systems do, but limit the computation to the average value within predefined image blocks.

The image analysis operations are summarized in Figure 1. Each image to be stored during the archive population stage is processed in order to extract values related to the features that are candidates for successive similarity evaluation: color distribution, orientation and objects position. The process involves various steps.

First, the images are scaled to 320 pixels in the horizontal direction, and the number of colors is normalized to 24-bit in a BMP representation format. The adopted color model is the HVC, therefore the original RGB space is converted as described in Section 3.2.

Then, the image is segmented into 16 blocks (in a 4 by 4 arrangement), and the average Hue, Value and Chroma components are computed for each block. The resulting data are normalized to a sum of 1, and arranged into a 16 elements array. The blocks are ordered by row. The *color* array C is obtained by a linear combination of the three components:

$$C_i = 0.5 \times \mathrm{Hue}_i + 0.25 \times \mathrm{Value}_i + 0.25 \times \mathrm{Chroma}_i \tag{10}$$

Computing the color feature in this way is obviously not very precise; but it allows us, with limited computational effort, to represent the color distribution of the most immediately visible components like large objects or background. The coefficients in (10) can be modified by the user, as will be described in Section 7.

In the subsequent step, the contours of the image objects are extracted. The procedure used in the edge detection stage is the one described in [21] adopting the zero crossing of second derivative. The edge lines are used as an image sketch for computing the linearization of the image using the Hough Transform. Two procedures are then executed on the linearized image.

The first procedure decomposes the edges in straight segments, and computes the length of all the edge segments for each slope in the 0º to 179º range. The computed data are grouped and sorted into an *orientation* array O composed of 18 elements, each element representing the integral over a range of 10º of the weights corresponding to line directions.

Assuming l_s the cumulative length of the segments with slope s, computed by analyzing the Hough transform, the i-th element of the array O is:

$$O_i = \sum_{s \in \Theta_i} l_s \qquad \Theta_i = \{\theta | i \times 10 \le \theta < (i+1) \times 10\} \qquad i = 0, ..., 17 \qquad (11)$$

The values are then normalized so the sum of all the values adds up to 1.

The second procedure aims at computing how lines are distributed in the image. The underlying hypothesis is that the distribution of edges shows, to some extent, where objects are located in the image. The procedure considers the linearized image segmented in 16 blocks (as in the color feature computation). All straight segments within each block are added up, regardless of their orientation, and the results are stored into a *distribution* array D with 16 elements.

Assuming l_e the length of a generic segment e, the i-th element of the array D is

$$D_i = \sum_{e \in E_i} l_e \qquad i = 0, ..., 15 \qquad (12)$$

with E_i denoting the set of all segments contained in image block i or clipped by its boundaries.

A threshold ensures that segments to be added must be at least 4 pixels long, an empirically determined value imposed to avoid adding up ineffective image details. Also in this case, the array values are normalized to a sum of 1 for the whole array.

At the end of the processing steps each image is described by a tuple of three feature vectors $F(C,O,D)$.

Figure 2 shows an example image picturing an airplane, with the 16 segments identified. The feature vectors describing the image according to the extracted features are shown in Figure 3.

It is worth noting that, while not trivial, the whole process is automatic and not driven by the user interpretation of the image meaning, nor it requires specific image areas to be marked. It is however obvious that the initial quality of the image influences the final result, and some human intervention can be required to “clean” the image or remove unwanted elements like frames, captions, etc., that are not part of the true content.

5 Query processing

Once features values associated with images have been computed and stored, queries may be processed.

Various models have been proposed for similarity analysis in image retrieval systems. Our representation shares many properties with the vector space model [4], that is widely used in textual document retrieval systems.

The vector space model 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 as a means of describing the document meaning and not its appearance.

Queries can be formulated in two ways, either by example or by sketch. The former assumes the query is an image to which the database content is compared, the latter uses as a query a drawing made by the user, sketching the color and the lines distributions of the requested image.

Three similarity functions simC(R,Q), simO(R,Q) and simD(R,Q), respectively accounting for color, orientation and segments distribution, are computed. Each function simX(R,Q) representing the similarity between a database image feature, defined by the tuple $R = (r_0, r_1, ..., r_n)$, and the query image feature, also defined by a tuple $Q = (q_0, q_1, ..., q_n)$ is computed using the *cosine* similarity coefficient, defined as:

$$\mathrm{sim}(R,Q) = \frac{\sum r_i q_i}{\sqrt{\sum r_i^2 \times \sum q_i^2}} \tag{13}$$

Our approach basically does not put any constraint on the similarity evaluation function. In fact other coefficients, namely the Dice and Jaccard coefficients [4] have been tested. They provide basically the same results, as far as higher ranking retrieved images are concerned, but with different absolute values. The resulting coefficients are merged to form the final similarity function as a linear combination:

$$\mathrm{sim}(R,Q) = \alpha \times \mathrm{simC}(R,Q) + \beta \times \mathrm{simO}(R,Q) + \gamma \times \mathrm{simD}(R,Q) \tag{14}$$

where α, β and γ are weighting coefficients. The weights defined by the coefficients obviously lead to increase or decrease the contribution of a feature with respect to the others.

In order to better characterize the orientation comparison a heuristic modification has been introduced in the orientation (simO) computation. As experiments showed that horizontal components have always almost a non-negligible presence in an image, their weight is reduced by some amount (actually 30%) in order to avoid a measure biasing.

Our work focused on the feature extraction and evaluation and on the relevance feedback analysis that will be described in next section, with little concern about time performance. Therefore we did not elaborate on a database engine in building the prototype, and based our experiment on a sequential scan of the feature vectors. For collections of limited size (some hundreds of items) the system response is reasonable. The distance between the query features and the stored features is computed during the scan, and the similarity indexes that result rank images in decreasing order.

Retrieval through graphic query formulation is approached by asking the user to draw a sketch of the desired images aspect. In this case, the sketch has only to make evident the distribution of lines and colors in the images. In general similarity scores are lower than when comparing real images, because of the lack of several contributions in the feature vectors due to the low level of detail of the sketch.

6 Relevance Feedback

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.

Furthermore our algorithm tries to combine different features, hence the overall computed similarity coefficients may differ from the user expectation, that may tend to concentrate on some particular feature.

Text-based information retrieval systems may 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. 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 $\overline{Q}^{(k+1)}$.

Our model for relevance feedback analysis follows the same approach. 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. Leaving an image unselected marks it as not relevant. 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 $\overline{Q}$ associated to the query image the feature vectors $\overline{X}$ of relevant images and subtracting the not relevant ones $\overline{Y}$, respectively weighted with suitable δ and ε coefficients:

$$\overline{Q}^{(k+1)} = \overline{Q}^{(k)} + \delta \sum_{i=1}^{N_{rel}} \overline{X}_i - \varepsilon \sum_{i=1}^{M_{notrel}} \overline{Y}_i \qquad (15)$$

The modification is performed separately on the three components: color, orientation and distribution, and 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 5.

It is worth noting 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.

7 Results and discussion

In this section we evaluate the results obtained with a prototype system implementing the described approach. The discussion focuses on the results obtained with two image collections picturing aircraft and cars, taken from repositories available in the public domain. Feature extraction was performed on the images in BMP format, while the images were stored in GIF format for browsing and display of results. The display of query results involves thumbnails that may make some features less visible.

The prototype is endowed with a graphical user interface written in Tcl-Tk. The system has two main parts: the one controlling the image database population and the other performing queries, implementing the operations described in Sections 4, 5 and 6. Computationally intensive image analysis stages like color processing, edge detection and Hough Transform were coded in Pascal, while the remaining part was coded in Tcl-Tk. As we said in Section 5, the prototype does not rely on a database engine. Images and associated feature vectors are stored as plain files, since the prototype aims at testing the effectiveness of the retrieval approach without time performance concern.

To perform a query, the user can either select an image in the database or draw a sketch including the features he/she considers relevant. Retrieved images are ranked in decreasing order of similarity. The system displays the five highest ranking images.

The user can tune his/her query, stressing the relevance of one of the computed features. The result is a modification of the weighting coefficients α, β and γ discussed in Section 5 (formula 14).

Figure 4 shows an example retrieval by pictorial example on the cars image collection: the query image is shown on the top left while retrieved images are displayed in the lower part, ranked in decreasing order of similarity.

The user can tune his/her requests in terms of percentage of Hue, Value, and Chroma components, which are set by default to 0.5, 0.25 and 0.25, respectively. In Figure 4 the higher contribution of the Hue component accounts for the high ranking of image *car06.gif* with respect to image *car23.gif*, even if the latter seems closer to the query.

The user can also stress the relative relevance of one out of the three features that, in our approach, describe the image. This can be done manually adjusting the weight α, β and γ coefficients have in the similarity function evaluation, as discussed in Section 6. These are all set by default to 1/3.

It is interesting to notice how things change considering only one of the features. Figure 5 presents a result for a query with $\alpha = 1$, $\beta = 0$ and $\gamma = 0$ values, i.e. only considering lines distribution. Figure 6 shows the retrieval results obtained, for the same query, with $\alpha = 0$, $\beta = 0$ and $\gamma = 1$, i.e., by considering color distribution only.

Relevance feedback is 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. The system modifies the query as described in Section 6 by increasing the query feature vectors with the contribution of the selected images, and decreasing the contribution of the other displayed images, assumed to be not relevant.

Figure 7 shows a query and the five higher ranking retrieved images without considering relevance feedback. As can be noticed, retrieval results do not appear satisfactory; although two images are present in the retrieved set, that are visually similar to the query, the other three are quite different, and their good score comes only from the cumulative effect of combining the different features.

By selecting as relevant the images that appear more similar to the query (the ones with check buttons marked in Figure 7), relevance feedback is applied, obtaining the results displayed in Figure 8. Even if the relevant images were the top ranking ones, the evenly distributed weight of the different features did not allow stressing of the specific similarities among them and the query. By explicitly marking them as relevant, thus marking the other ones as not relevant, their distinguishing features are strengthened.

Another example is in Figure 9, which shows a tuned result obtained by marking as relevant the first retrieved image in Figure 4.

In order to evaluate the results against human perception, a small experiment was undertaken. Despite its toy size and limited scope, it showed how this approach is promising in performing a reliable image clustering.

A subset of twenty images picturing aircraft was submitted to ten volunteers, asking them to cluster the images according to their visual appearance, neglecting any consideration about the type of plane, e.g. pistons or jet propulsion, bomber or fighter. The responses where homogeneous, and were merged obtaining six clusters.

A set of retrieval sessions was then executed, considering as database only the previously selected twenty images, and as query images one for each of the selected clusters. Results proved a good correspondence of the retrieval system with the human evaluation.

Within the five highest ranking images were the images of the selected clusters with a percentage of 88%. This percentage raised to 96% after relevance feedback performed selecting retrieved images included in the cluster.

As a general comment, the scores allow a good partitioning of images in classes exhibiting coherent visual properties and similar aspects. Local ranking within classes is sometimes biased by other image features, the most notable being the influence of contrast, resolution and foreground/background relationships, and the general image quality.

8 Conclusion

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

We have presented a system for similarity evaluation based on the extraction of simple features such as color, object orientation and object position within images. A simple prototype system implementing image analysis and retrieval has also been introduced. Although not examining other features typically included in similar systems, e.g. shape, we consider these features as a simple set useful in the retrieval from thematic databases, i.e. databases limited to a common domain. Most image collections are actually subdivided in homogeneous sub-collections.

Other limitations include the fact that the system, requiring no user intervention, is strongly dependent on the original image quality, and that it suffers in highly textured images. The contour extraction and Hough Transform procedures may be considered computationally intensive although, as the processing is performed on reduced size images, their requirements are limited; anyway this processing has to be done just once during the population stage.

Turning to advantages, there are several, namely: the absence of human interaction, the overall simplicity of features used and of the retrieval technique, which in the authors opinion is an extremely important issue, and the use of a technique similar to the vector space approach used in text-based information retrieval, that allows us to increase relevant image accuracy using relevance feedback analysis.

Acknowledgments

The authors acknowledge one of the referees for suggesting a number of language improvements.

This work has been supported by the Italian Ministry for University and Scientific Research (MURST), in the framework of the project "Basi di dati evolute: modelli, metodi e strumenti".

References

1. *ACM Multimedia Systems*, special issue on Multimedia Databases, 3, 5/6, 1995.
2. V. N. Gudivada, V. V. Raghavan, Picture Retrieval Systems: A Unified Perspective and Research Issues, *Computer Science Technical Report CS9503*, Ohio University, 1995, ftp://ftp.cs.ohiou.edu/tech-reports/cs9503.ps.Z
3. *IEEE Computer*, special issue on Content Based Image Retrieval, 28, 9, 1995.
4. G. Salton, *Automatic Text Processing*, Addison Wesley, 1989.
5. E. Di Sciascio, A. Celentano, Similarity Evaluation in Image Retrieval Using the Hough Transform, *Journal of Computing and Information Technology*, 4, 3, 199-204, 1996.
6. W. Niblak et al., 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, 1993.
7. M. Flickner et al., Query by Image and Video Content: The QBIC System, *IEEE Computer*, 28, 9, 23-31, 1995.
8. P. M. Kelly, T. M. Cannon, D. R. Rush, Query by image example: the CANDID approach, in *Storage and Retrieval for Image and Video Databases*, SPIE, vol. 2420, 238-248, 1995.
9. V. E. Ogle, M. Stonebrakes, Chabot: retrieval from a relational database of images, *IEEE Computer*, 28, 9, 40-56, 1995
10. R. Bach et al., 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, 1996.

11. A. Yoshitaka, S. Kishida, M. Hirakawa, T. Ichikawa, Knowledge-Assisted Content-Based Retrieval for Multimedia Databases, *IEEE Multimedia*, 1, 4, 12-21, 1994

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

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

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

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

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

17. J. Wang, P.J. Howarth, Use of The Hough Transform in Automated Lineament Detection, *IEEE Trans. on Geoscience*, 28, 4, 561-566, 1990.

18. D. W. C. Pao, H.F. Li, R. Jayakumar, Shapes Recognition using the straight line Hough transform: theory and generalization, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 14, 11, 1076-1082, 1992.

19. Y. Gong, M. Sakauchi, Detection of regions matching specified chromatic features, *Computer vision and image understanding*, 61,2, 263-269, 1995.

20. G. Wyszechi, W. S. Stiles, *Color science: concepts and methods, quantitative data and formulas*, John Wiley and Sons, NewYork, 1982.

21. V. Torre, T. Poggio, On edge detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8, 147-163, 1986.

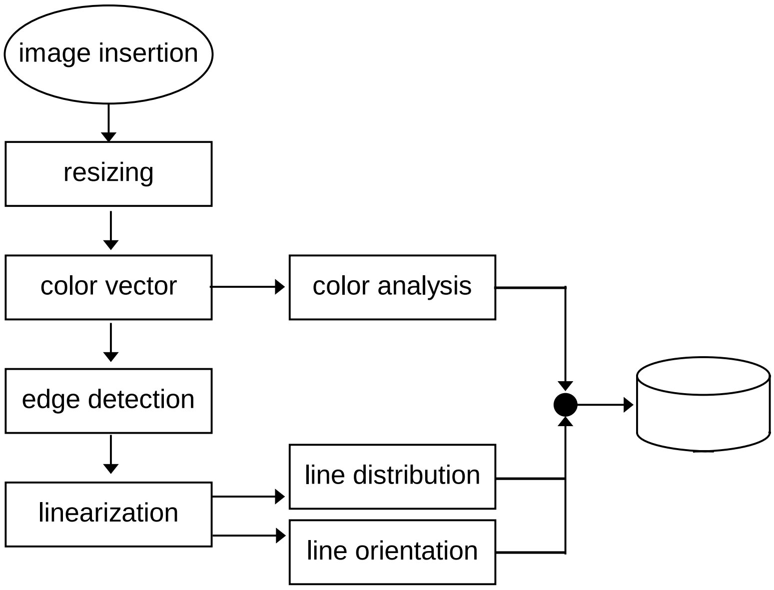

Figure 1. Image analysis operations

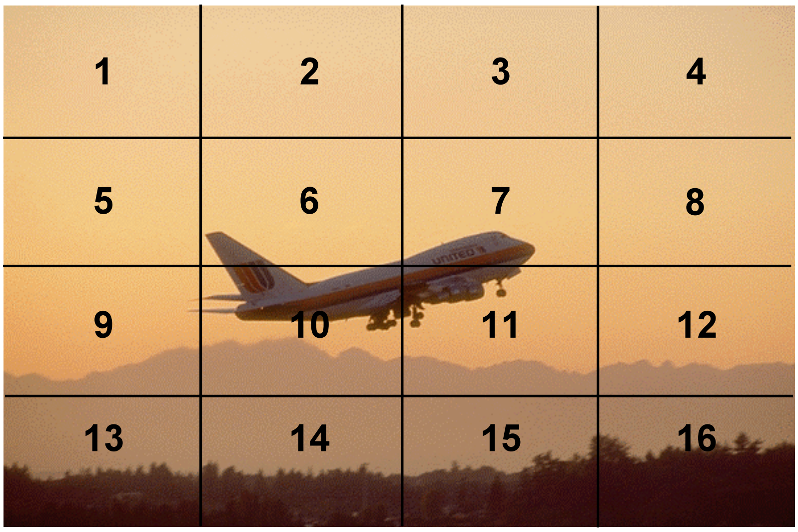

Figure 2. A sample image segmented for color and position analysis

Orientation	
Angle range	Value
0°-9°	0.468
10°-19°	0.317
20°-29°	0.024
30°-39°	0
40°-49°	0
50°-59°	0.014
60°-69°	0.006
70°-79°	0
80°-89°	0
90°-99°	0
100°-109°	0
110°-119°	0.019
120°-129°	0
130°-139°	0
140°-149°	0.044
150°-159°	0.030
160°-169°	0.028
170°-179°	0.049

Distribution	
Segment	Value
1	0
2	0
3	0
4	0
5	0
6	0.051
7	0.078
8	0
9	0.070
10	0.280
11	0.148
12	0.061
13	0.074
14	0.083
15	0.096
16	0.058

Color			
Segment	Hue	Value	Chroma
1	0.067	0.051	0.076
2	0.068	0.048	0.076
3	0.067	0.049	0.075
4	0.066	0.049	0.073
5	0.070	0.061	0.077
6	0.068	0.058	0.075
7	0.065	0.057	0.070
8	0.067	0.059	0.075
9	0.060	0.069	0.071
10	0.058	0.065	0.054
11	0.058	0.068	0.063
12	0.058	0.065	0.068
13	0.041	0.076	0.039
14	0.045	0.069	0.040
15	0.053	0.072	0.038
16	0.089	0.085	0.029

Figure 3. Feature vectors related to image in Figure 2

Figure 4. An example of image retrieval

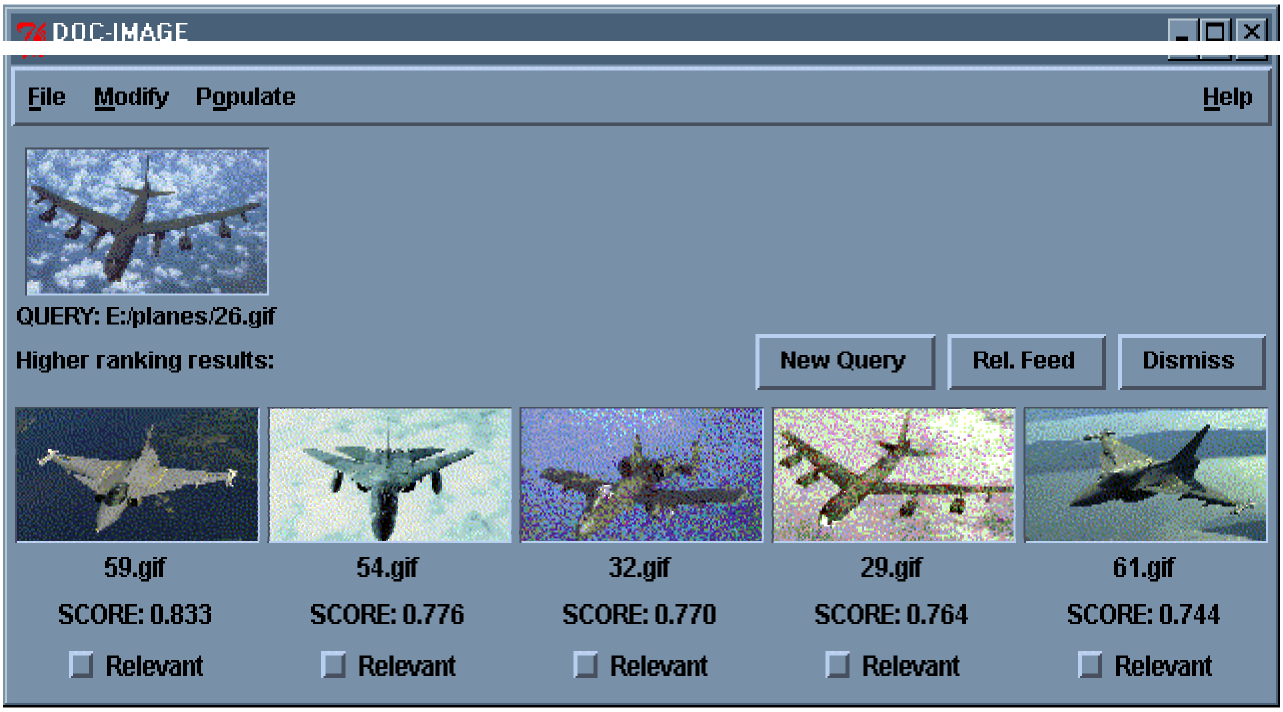

Figure 5. Query and results, retrieval based on lines distribution

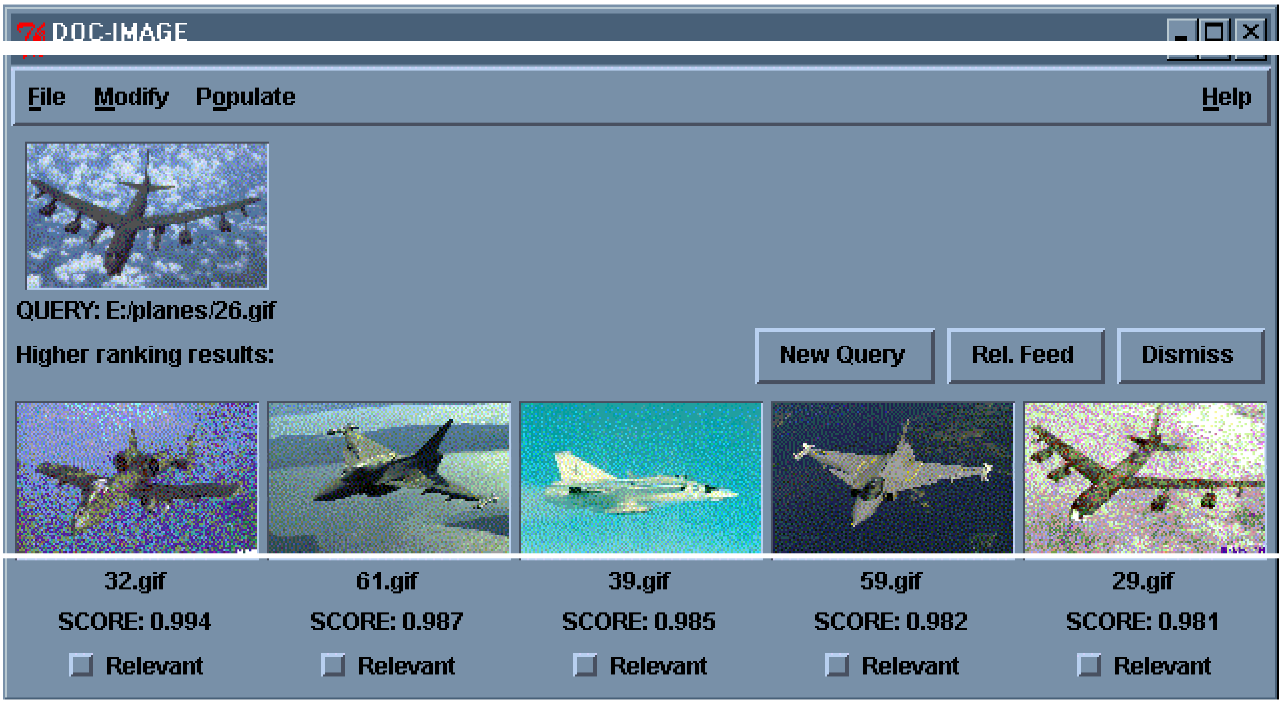

Figure 6. Same query as Figure 5, retrieval based on color distribution

Figure 7. Another query and the retrieved images

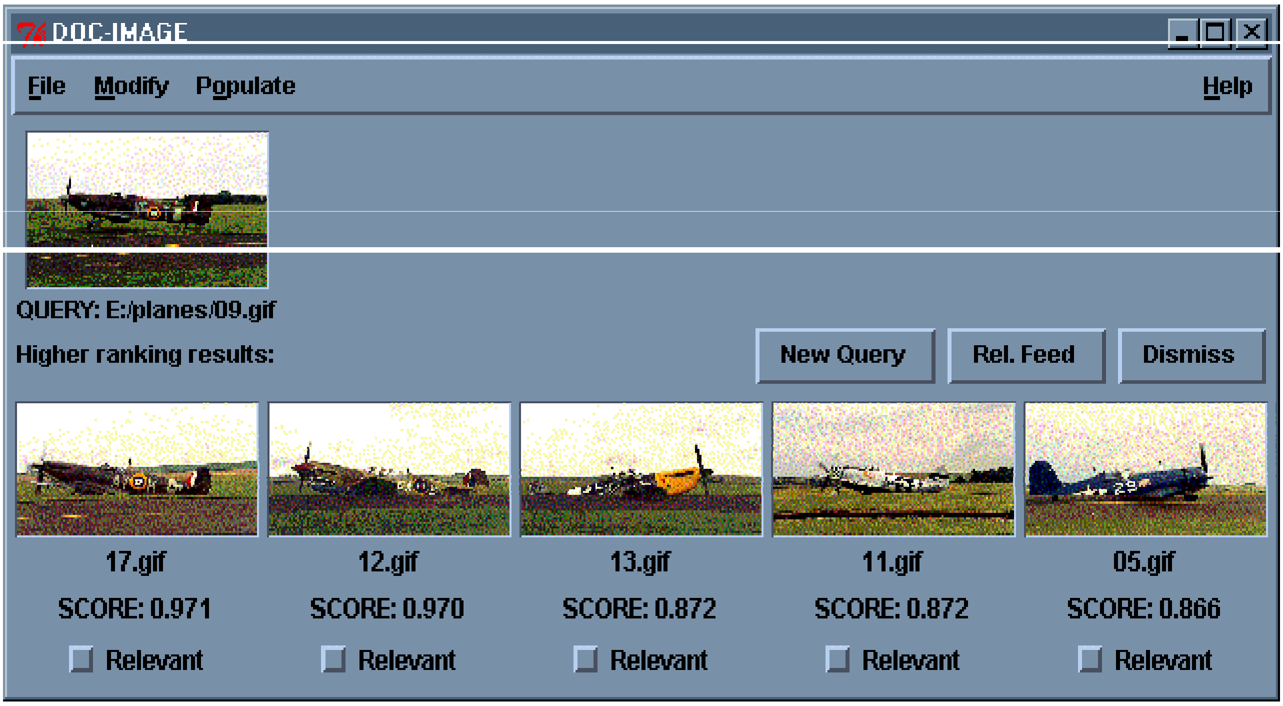

Figure 8. Same query as Figure 7, relevance feedback results

Figure 9. Another example of relevance feedback result, with query as in Figure 4