# Similarity Evaluation in Image Retrieval Using Simple Features

Eugenio Di Sciascio

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

Augusto Celentano

Università Ca' Foscari di Venezia
Dipartimento di Matematica Applicata e Informatica
Ca' Dolfin, Dorsoduro 3825/e, Venezia I-30100 Italia

## ABSTRACT

The development of increasingly complex multimedia applications calls for new methodologies for the organization and retrieval of still images and video sequences. Query and retrieval methods based on image content promise good results, are currently widely investigated and, to some extent, already commercially available. Yet a large number of issues remain unsolved.

In this paper we describe some results of a study on similarity evaluation in image retrieval using color, object orientation and relative position as content features. A simple prototype system is also introduced that computes the feature descriptors and performs queries. Although not trivial, the features extraction process is completely automated and requires no user intervention.

The system is admittedly not a general purpose tool, but is oriented to thematic image repositories where the semantics of stored images are limited to a specific domain.

**Keywords:** Image databases, image feature computation, retrieval by image content, vector space model, Hough transform.

## 1. INTRODUCTION

Current technology allows the acquisition, transmission, storing, and manipulation of large collections of images. Yet conventional archiving systems rely on well defined text documents, describing the image content. The texts are accessed by database queries according to the conventional text retrieval rules. Typically such descriptions result inadequate; therefore queries performed this way often lead to unsatisfactory answers.

In contrast with this approach an ideal situation would be the one in which queries to an image database should refer to the images content, and returned images should be ranked according to the degree of content matching. Content based information retrieval (CBIR) systems apply image processing techniques on raw (i.e. bitmap) images striving to generate an internal representation of the image.

Recently a number of methodologies, techniques and tools have been studied for identification and comparison of images features in order to develop classification and retrieval systems based on (almost) automatic interpretation of image contents.

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 sample image [1-4]. 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 features vectors, and can use retrieval methodologies taken from the textual retrieval framework, such as the vector space model [5].

The approach considered 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 preceding identification of the relevant collection theme is not a limiting requirement.

The approach can be however extended to a generic database by pre-selecting on legends or other objective attributes a subset of images, in order to complete the content oriented search on a semantically coherent set of data.

In this paper it is argued that the orientation of objects within an image, their position and the color distribution are key attributes in the definition of the similarity in retrieval by content. In particular, the visual coherence of a set of similar images should strongly depend on these features. A further 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.

As a matter of fact, other, more precise, features and methodologies have been recently proposed, but the heavy related computational burden, and hence the time required to process them, mainly in the retrieval stage, aren't always worth the actual retrieval accuracy improvement.

The case study introduced in this paper is a small database of eighty images picturing aircrafts. Scenes are variable, as the image quality is, although all the images refer to the same subject. Figure 1 shows a sample of the images used.

In Section 2 we describe the related work. Section 3 addresses the basic techniques that are used during feature extraction. Sections 4 and 5 describe respectively the image analysis stage and the query processing. Some experiments and a prototypal system are presented in Section 6. Obtained results and a conclusion about further investigations are outlined in the final section.

## 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.

The Candid system [8] is also a content based storage and retrieval systems. 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 features 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.

Domain knowledge is also used as a basis for image interpretation. In ref. [10], 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 [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.


Figure 1. Sample of the images used

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

An interesting reading is in ref. [15] 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, hence trying to be more fair towards conventional retrieval systems.

## 3. BASICS

### 3.1. The Hough Transform

The Hough Transform [16,17] 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 dominion.

A straight line in the starting dominion 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 dominion. 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:

$$r = x_i \cos(a_i) + y_i \sin(a_i) \tag{2}$$

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 Hough (radial) space by the transformation:

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

For every edge point (x,y), the corresponding $(r,a)$ coordinates are computed. Then, the value associated to $(r,a)$ 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 it results 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 [18] the transformation is obtained through the CIE L*a*b* model [19].

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. IMAGE ANALYSIS

The approach of this work is based on the assumption that images content is to some extent homogeneous, hence we concentrate our attention on the features, simple and immediate, that may characterize 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 scenes it should not be assumed as a generally valid rule. Figure 2 shows a simple counter-example, where the focus of the image is far from the center of the shot.

Orientation is the other feature we concentrated on. We argue that orientation, in the described framework, is a key attribute in defining a perceptual similarity.

The other key point we tried to emphasize in the design of our system was 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 and that queries must be processed extremely quickly.

Figure 2. Example image showing the object displaced with respect to the shot center

Rather than putting much processing effort in answering a query in the sharpest way, it is often reasonable to give a fast reply with a rougher retrieval method, and give the user the ability to interact easily with the system by tuning the response through relevance feedback analysis. 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 image blocks.

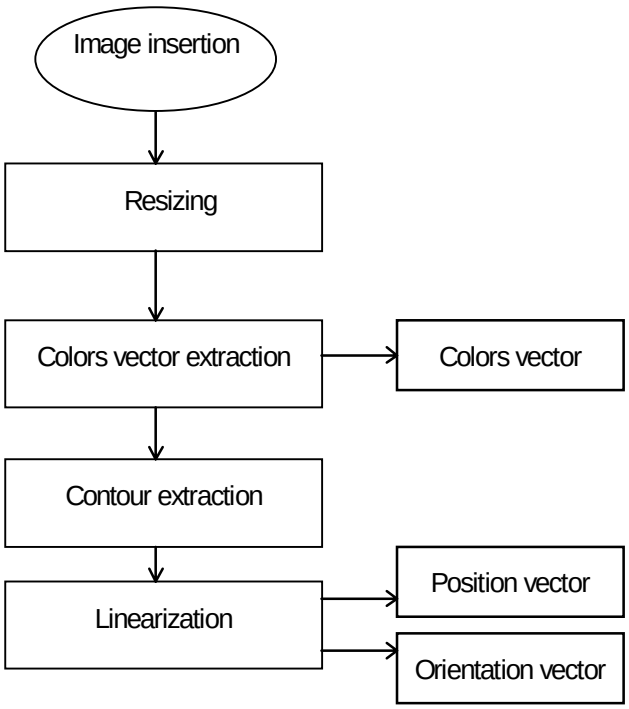


Figure 3. Image analysis stages

The proposed system image analysis operations are summarized in figure 3. Each image to be stored during the population stage is processed in order to extract values related to the features that are candidates for

successive similarity evaluation: orientation, color distribution, and objects position.

The procedure that has been adopted to extract the named features of an image during the population stage 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 space is converted as previously described. Only the Hue component is then used in the subsequent steps.

Although a complete color evaluation would obviously be more precise, the Hue component, in the present prototype, provides a sufficient description of the color distribution. The experiments performed so far show that processing efficiency can be well traded off with accuracy.

The image is segmented into 16 blocks, and the average Hue component is computed for each block. The resulting data are normalized and grouped into a 16 elements array

In the subsequent step, contours of the image objects are extracted. The procedure used in the edge detection stage is the one described in ref. [20] 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 one decomposes the edges in straight segments, and computes for each edge segment its length and the corresponding slope, in the 0º to 179º range. The computed data are grouped and sorted into a 18 elements array, each element representing the integral over a range of 10º of the weights corresponding to line directions. The values are then normalized so the integral of all the values adds up to 1.0.

The second one considers the linearized image segmented in 16 blocks. We aim at computing how lines are distributed in the image. The hypothesis is that this distribution shows, to some extent, where objects are in the image. All straight segments within each block are added up, this time regardless of their orientation, and results are normalized and stored into a 16 elements array. A threshold imposes that segments to be added must be at least 4 pixels long, an empirically determined value imposed to avoid adding up ineffective image details. In this way we try to roughly identify areas were objects are within the images.

It is worth to note that, while not trivial, the whole process is automatic and not driven by the user interpretation of the image meaning. It is however obvious that the initial quality of the image influences the final result. Human intervention is required only if images include captions or frames that could modify the feature interpretation, and therefore must be removed.

At the end of the various steps each image is endowed of three vectors that represent to some extent how the color is distributed, how the lines that form objects are oriented, and how objects are distributed.

## 5. QUERY PROCESSING

Once features values associated to images have been computed and stored, queries may be done. Various models have been proposed for similarity analysis in image retrieval systems. In this work we use the vector space model[5], that is widely used in textual document retrieval systems.

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

Three similarity functions *simH(D,Q)*, *simO(D,Q)* and *simS(D,Q)*, respectively accounting for Hue, orientation and segments distribution, are computed. Each function *simX*(*D,Q*) between a database image feature, defined by the tuple $D = (d_0, d_1, ..., d_n)$, 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:

$$sim(D,Q) = \frac{\sum d_i q_i}{\sqrt{\sum d_i^2 \times \sum q_i^2}} \tag{10}$$

Other coefficients, namely the Dice and Jaccard coefficients have been tested [5]. 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:

$$sim(D,Q)=a \times simH(D,Q) + b \times simO(D,Q) + c \times simS(D,Q) \quad (11)$$

where $a$, $b$ and $c$ are weighting coefficient empirically set by default to $a$=0.3, $b$=0.35, $c$=0.35. They can be changed should the user want to stress a single feature.

In order to better characterize the orientation comparison a heuristic modification has been introduced in the simO ranking computation. Whenever an image in the database has a main orientation index (i.e. the orientation whose contribution is a maximum in the image associated vector) differing for more than 30º in either direction, its similarity score is reduced of 50 %. Although admittedly rough, this can be considered as a sort of a priori relevance discrimination. As a further remark, as experiments showed that horizontal components have always a considerable presence within an image, their weight is reduced by 30 % in order to avoid the measure biasing due to this component.


Figure 4. Result of query by example using color, orientation and object position: higher three scores. Retrieved images captions refer to the image file name and corresponding ranking.

The retrieval engine is based, at the current development stage, on a sequential scan of the features associated vectors and the consequent vector distance measure between the query features and the stored features. The similarity index that results ranks images in decreasing order.

In general, by comparing ranked results with the actual image contents, it can be seen how scores from about 0.6 up constitute a good measure of image similarity, while lower scores correspond to visible differences, and with very low scores (i.e., less than 0.4) almost no visual similarity with the original image can be found. Figure 4 shows an example query and the three higher ranking retrieved images.

Extremely high ranking is an index of similarity that is rarely available, but in cases like the one in figure 5 where the highest ranking photo has been evidently shot in the same place with an identical scene situation, just varying the subject.

Retrieval through graphic query formulation can be 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 scores are lower than when comparing real images, due to lack of several contributes in the vectors, at least for the similarity measures involving the line distribution. Figure 6 shows a simple query by sketch and best ranking retrievals.

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


Figure 5. Example of query by example showing, as best ranking retrieval, an image extremely similar to the query.

## 6. A PROTOTYPE SYSTEM

A prototype system that allows the automatic images analysis and retrieval has been developed, endowed of a graphical user interface written in MS-Visual Basic. It is currently being ported to Tcl-Tk for wider use. The system has two main parts: the one controlling the image database population and the one performing queries, and basically implements operations described in sections 4 and 5.

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 user can

also use one of the resulting images to run a new more refined query. Possibility of running an automated relevance feedback is under development.

The user can also tune his/her query, stressing the relevance of one of the computed features; the result is a modification of the weighting coefficients *a*, *b* and *c* previously described.

More computationally intensive stages like contour extraction and Hough Transform are coded in Pascal, while the remaining part is coded entirely in Visual Basic.

The prototype does not rely on a real database. Images and feature vectors are represented as files, since at the current development stage the prototype aims at testing the retrieval approach without performance concern.

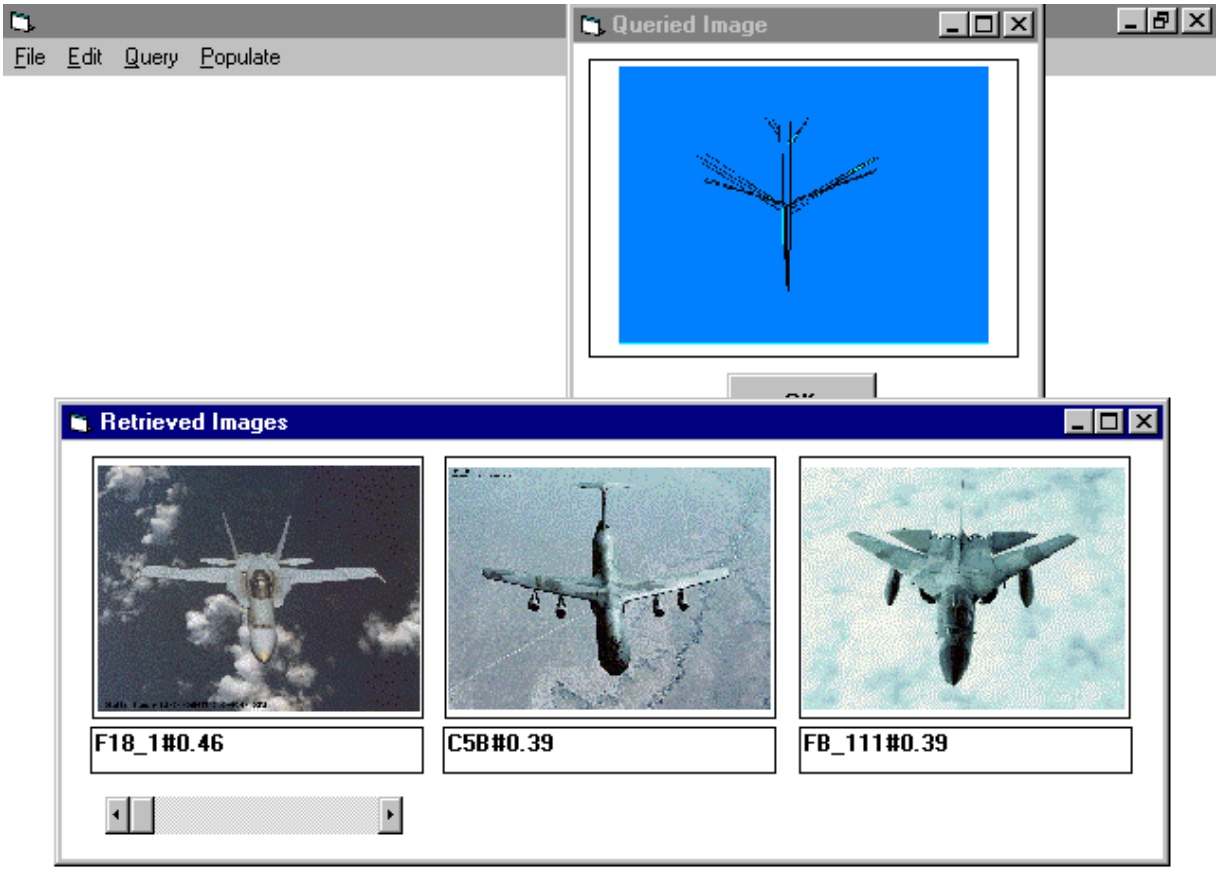


Figure 6. Example of query by sketch and three highest ranking retrieved results.

## 7. CONCLUSION

Currently available large images 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 introduced a system for similarity evaluation based on the extraction of simple features such as color, objects orientation and objects position within images. A simple prototype system implementing image analysis and retrieval has also been presented. Although giving up other features typically included in similar systems, e.g. shape, we considered these features as a simple set perceptually useful in the retrieval from thematic databases, i.e. limited to a common semantic domain , as most images collections are actually subdivided in.

Other limitations include the fact that the system, requiring no user intervention, is strongly dependent on the original images quality, and that it suffers on highly textured images. Furthermore the test bed database cannot be considered a general one, as images of planes present strong directional components, but this is not true in other frameworks. 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 various, namely: the absence of human interaction; the overall simplicity of features used and of the retrieval technique, that in the authors opinion is an extremely relevant issue and the use of a well established technique, as the vector space approach is, that calls for currently under development relevance feedback analysis.

## ACKNOWLEDGEMENTS

This work has been supported in part 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. Barber et al., Query by Content for Large On-line Image Collections, Multimedia Systems, an IEEE Tutorial, IEEE Press, 1994.
3. V. V. Gudivada and V. V. Raghavan, Guest Editors' Introduction: Content-Based Image Retrieval Systems, IEEE Computer,28, 9, 1995.
4. IEEE Computer, special issue on Content Based Image Retrieval, 28, 9, 1995.
5. G. Salton, Automatic Text Processing, Addison Wesley, 1989.
6. Niblak et al., The QBIC project: Querying images by content using color, texture, and shape, Proceedings of the SPIE: Storage and Retrieval for Image and Video Databases vol. 1908, 1993.
7. M. Flickner et al., Query by Image and Video Content: The QBIC System, IEEE Computer, 28, 9, 1995
8. P. M. Kelly, T. M. Cannon and D. R. Rush, Query by image example: the CANDID approach, Proceedings of the SPIE: Storage and Retrieval for Image and Video Databases III, vol. 2420, 238-248, 1995.
9. V. E. Ogle, M. Stonebrakes, Chabot: retrieval from a relational database of images, IEEE Computer, 28, 9, 1995
10. A. Yoshitaka, S. Kishida, M. Hirakawa and T. Ichikawa, Knowledge-Assisted Content-Based Retrieval for Multimedia Databases, IEEE Multimedia, 1, 4, 1994
11. J. K. Wu and A. S. Narisimhalu, Identifying faces using multiple retrievals, IEEE Multimedia, 1, 2, 1994.
12. J. Bach, S. Paul and R. Jain, A Visual Information Management System for The Interactive Retrieval of Faces, IEEE Trans. on Knowledge and Data Engineering, 5, 4, 1993
13. S. W. Smoliar and H. Zhang, Content-Based Video Indexing and Retrieval, IEEE Multimedia, 1, 2, 1994.
14. H. Zhang, A. Kankanhalli and S.W. Smoliar, Automatic Partitioning of Full Motion Video, ACM Multimedia Systems, 1, 1, 1993.
15. 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.
16. J. Wang and P.J. Howarth, Use of The Hough Transform in Automated Lineament Detection, IEEE Trans. on Geoscience, 28, 4, 1990.
17. D.W.C. Pao, H.F. Li and R. Jayakumar, Shapes Recognition using the straight line Hough transform: theory and generalization, IEEE Trans. On Pattern Analysis and Machine Intelligence, 14, 11, 1992.
18. Y. Gong and M. Sakauchi, Detection of regions matching specified chromatic features, Computer vision and image understanding, 61,2, 1995.
19. G. Wyszechi, W. S. Stiles, Color science: concepts and methods, quantitative data and formulas, Wiley eds., NewYork, 1982.
20. V. Torre and T. Poggio, On edge detection, IEEE Trans. on Pattern Analysis and Machine Intelligence, 8, 147-163, 1986.