ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 4 April, 2013 Page No. 997 -1006

Performance Analysis of Content Based Image Retrivel

Arul Murugan A, Jothi Vignesh C, Nijanthan S, Rajasekar R, Vignesh E

Final Year, Department of Computer Science and Engineering
K.S.R College of Engineering.

Mr.G.Karthick,M.E.,
Assistant professor, Department of Computer Science and Engineering
K.S.R College of Engineering

Abstract**:**

This report provides a comprehensive justification on the architectural design of the content based image retrieval system, which adopted the plug-in framework. It presents a thorough explanation of how diverse image feature extraction algorithms were implemented and adopted seamlessly in the system; moreover, it also describes how these algorithms will be dynamically applied in the context of user modifying the query parameters. Furthermore, it illustrates the prominent performance along with accurateness of the system by given that performance metrics and outcome of comparing with other systems. As a final point, it reports the work contribution of team members, and the project management disciplines used to achieve the success of the project.

I.INTRODUCTION

As digital images bring impressive moments to our daily life, there is an ever increasing need to ensure effectively retrieve multimedia content in a wide range of environment. The massive volume of images has challenging many great researchers to investigate on the feasible methods for content-based image retrieval applications; such applications could be used in commerce, medicine, education, and crime prevention.

The conventional image database search based on semantic annotation or keywords, editing keywords or labelling images are time-consuming tasks, and sometimes semantic views are normally different for each user. Content Based Image Retrieval (CBIR), aims to solve those problems by representing images with feature vectors (colour, texture, shape) in the database, those feature vectors are extracted from images without human intervention.

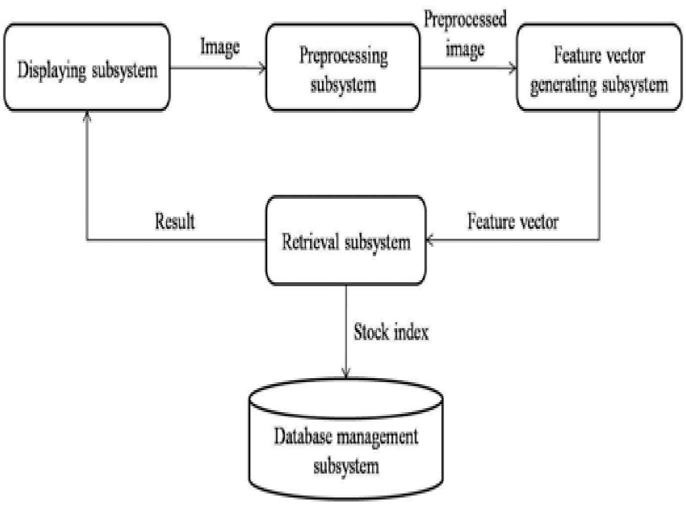
CBIR image retrieval system presented in this paper is called Images Management Smart this system consists of three main phase: 1) features extraction, 2) retrieving methods, and 3) ranking results and present images. The feature extraction is the essential process of a CBIR system. First, the CBIR retrieval systemselects appropriate feature spaces and explores various visual features to represent an image. Thus, the system can find the "preeminent" imagery representation for an image in database. Second, based on the selected features, the images are represented by feature vectors. Aretrieval system searches the nearest neighbours in the feature space by weighting different feature vectors and computing a similarity measurement for these feature vectors. The special measuring algorithms are designed to search the 'most similar' image from a database. Has more than one feature extraction function which means it allows user trying different features in image querying while deciding which features provide better result than others in some particular conditions.

CBIR has become an active and fast-advancing research area in image retrieval in the last decade. By and large, research activities in CBIR have progressed in four major directions: global image properties based, region-level features based, relevance feedback, and semantic based. Initially, developed Algorithms exploit the low-level features of the image such as color, texture, and shape of an object to help retrieve images. They are easy to implement and perform well for images that are either simple or contain few semantic contents. However, the semantics of an image are difficult to be revealed by the visual features, and these algorithms have many limitations when dealing with broad content image database. Therefore, in order to improve the retrieval accuracy of CBIR systems, region basedimage retrieval methods via image segmentation were. These methods attempt to overcome the drawbacks of global features by representing images at object level, which is intended to be close to the perception of human visual system. However, the performance of these methods mainly relies on the results of segmentation. The difference between the user's information need and the image representation is called the semantic gap in CBIR systems. The limited retrieval accuracy of image centric retrieval systems is essentially due to the inherent semantic gap. In order to reduce the gap, the interactive relevance feedback is introduced into CBIR. The basic idea behind relevance feedback is to incorporate human perception subjectivity into the query process and provide users with the opportunity to evaluate the retrieval results. The similarity measures are automatically refined on the basis of these evaluations. However, although relevance feedback can significantly improve the retrieval performance, its applicability still suffers from a few drawbacks. The semantic-based image retrieval methods try to discover the real semantic meaning of an image and use it to retrieve relevant images. However, understanding and discovering the semantics of a piece of information are high level cognitive tasks and thus hard to automate.

II .OUR PROJECT

In this section the goal and the global structure of our system

is presented. The components and their communications



III.SYSTEM OVERVIEW

Content-Based Image Retrieval system, its goal is to operate on collections of images and, in response to visual queries, extract relevant images. It has assorted visual feature extraction capabilities with portable, high-speed embedded database, which lead the high performance of IMSmart. Its friendly user interface allows users interact with the system straightforwardly. The ranking feature enables the system to rank relevant images based on the percentage of similarity, in addition, users can give feedback to the system by voting the relevant images that not detected by the system. An additional notable feature is painting board, which allows users paint a picture, and then make a query based on the painting.

IV *THE PURPOSE OF THE SYSTEM*

Even though the measure of research in sketch-based image retrieval increases, there is no widely used SBIR system. Our goal is to develop a content-based associative search engine, which databases are available for anyone looking back to freehand drawing. The user has a drawing area, where he can draw all shapes and moments, which are expected to occur in the given location and with a given size. The retrieval results are grouped by color for better clarity. Our most important task is to bridge the information gap between the drawing and the picture, which is helped by own preprocessing transformation process. In our system the iteration of the utilization process is possible, by the current results looking again, thus increasing the precision

V THE GLOBAL STRUCTURE OF OUR SYSTEM

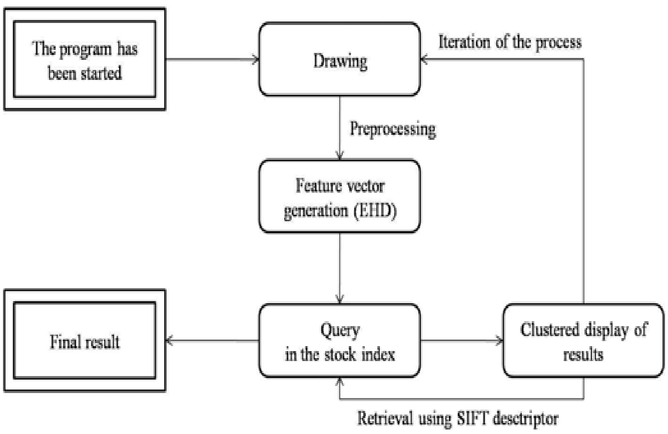
The system building blocks include a preprocessing subsystem, which eliminates the problems caused by the diversity

of images. Using the feature vector generating subsystem our

image can be represented by numbers considering a given

property. The database management subsystem provides an

interface between the database and the program. Based on the feature vectors and the sample image the retrieval subsystem provides the response list for the user using the displaying



Subsystem (GUI).The content-based retrieval as a process can be divided intotwo main phases. The _rst is the database construction phase, in which the data of preprocessed images is stored in the form of feature vectors – this is the off-line part of the program. This part carries out the computation intensive tasks, which has to be done before the program actual use. The other phase is the retrieval process, which is the on-line unit of the program. Examine the data _ow model of the system from the user's point of view. It is shown in Fig. 2. First the user draws sketch or loads an image. When the drawing has been finished or the appropriate representative has been loaded, the retrieval process is started. The retrieved image is preprocessed.

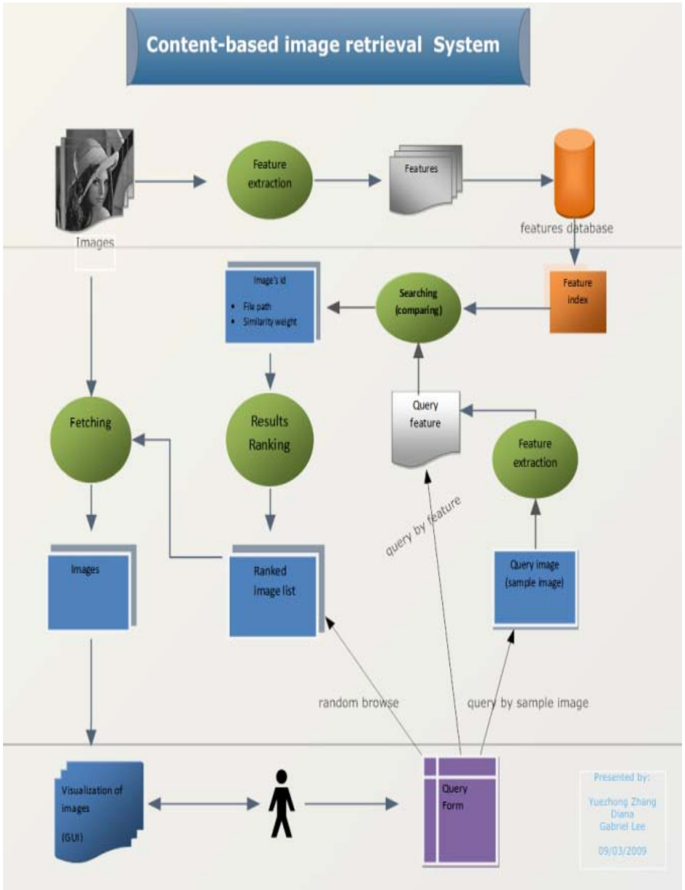
After that the feature vector is generated, then using the retrieval subsystem a search is executed in the previously indexed database. As a result of searching a result set is raised, which appears in the user interface on a systematic Form. Based on the result set we can again retrieve using another descriptor with different nature. This represents one using loop.

VI THE PREPROCESSING SUBSYSTEM

The system was designed for databases containing relatively simple images, but even in such cases large differences can occur among images in _le size or resolution. In addition, some images may be noisier, the extent and direction of Illumination may vary (see Fig. 3), and so the feature vectors Cannot be effectively compared. In order to avoid it, a multistep preprocessing mechanism precedes the generation of descriptors. The input of the preprocessing subsystem is one image, and the output is the respective processed result.

1. Architectural Strategies

Architecturally largely similar to other content-based image retrieval systems; nevertheless, it follows the multi-tire (layer) architecture design, even though it doesn't involves any client-server communication. The presentation layer, processing (retrieval) layer, feature extraction layer are completely separated. This lead to easier implementation by separating the system to layers, because each layers can be implemented independently, it only expose necessary interfaces to another layer.



System Architecture

As we mentioned above that system is separated into layers, but each layer has modules (*components*) that provides actually services to another layer. Next, we will discuss the architecture of individual modules and how it is fitted in the system. Below is general overview of these modules, the elaborated discussion on these components is in detailed design section.

Feature Extraction Module

The feature extraction module is designed using plug-in framework principle; the purpose of adopting that principle is to enhance the extensibilities of which enables dynamically discovers newly added feature extraction modules. *(We will discuss more about it in detailed design)*

Persistence Module

This module (component) takes care the transaction and persistent of the image features with database. It provides a clear-cut programming interface to other components. Consequently, other module in the system will effortlessly to deal with database (such as Feature Extraction and Query module).

Query Module

The responsibilities of this module are analysis user's query (e.g. identify different parameters of the query), it retrieves the image feature vectors from database based on the query.

Result Ranking Module

The role of this component is to rank the result of query and rank the image with highest similarity to the front of result list.

Presentation Module

It consists of UI components to allow users to edit and submit queries to the Query Module. It also displays the result images to users, and interacts with users.

2. Detailed System Design

2.1. Feature Extraction Module

As we mentioned about plug-in framework for this module, it is actually done by all feature extraction module (method or algorithms) must be implemented Feature Module interface, Feature Module should be created by the factory (FeatureModuleFactory), let have a look the diagram below. The factory create and manage different feature extraction module, it will dynamically select a feature extraction module to use based on the user preferences. It has the ability to add new feature extraction module to the manager at run-time.

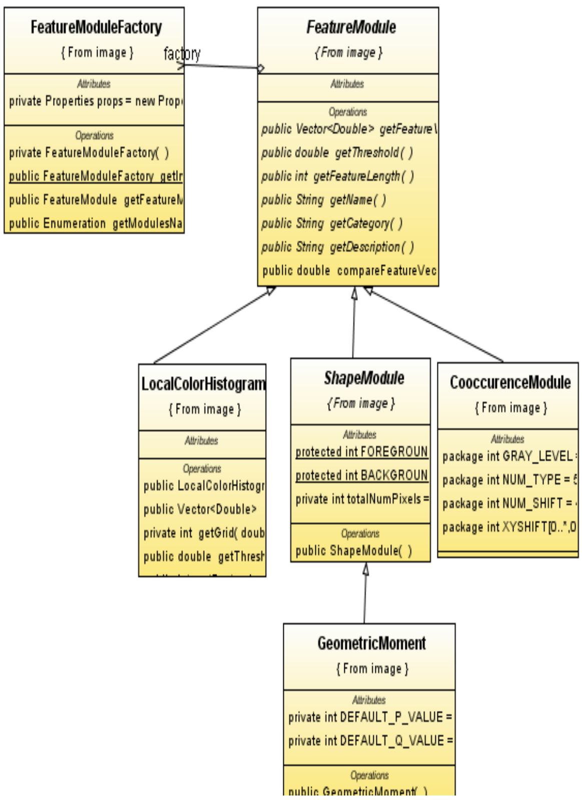


Figure 2: Feature Extraction Module Class Diagram

2.2. Persistence Module

This module handles the database transaction for store feature vectors to the database. Each feature vector as defined as an object Feature Info with four attributes. These attributes will be store in the database, each Feature Info is one row.

Feature Info

Id	Feature name	file path	vector

It provides following interface to other components

Return	Method
FeatureInfo	getFeatureById(int id)
boolean	insert(FeatureInfo info)
boolean	delete(FeatureInfo info)
boolean	exists(FeatureInfo info)
FeatureInfo	getFeatureByImage(String filepath)

2.3. Query Module

This module has a Query Manager, which handles user's image query, it provides a basic interface to other components; for example, any components required to make a query, it need to get a references of query Manager, and then call the query method , which will return a list of relevant images

```
Code Example:
QueryManager manager = QueryManager. Instance();
List resultList = manger.query (sampleImage);
```

Query module doesn't provide function to compare the distance between two feature vectors, it just delegate the job to Feature Module to compare two feature vectors. Whichever module extends the feature Module can override the default compare feature vector method.

The Euclidean distance is used as default implementation for comparing two feature vectors.

The Euclidean distance

$$\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

2.4Result Ranking Module

When the query finish, it returns a list of QueryResult, which has a value named distance between two feature vector. This module takes these data and then sorts this data in an ascending order (the greater distance will be at back of the list). It should provide a simple interface to other module; it accepts a list of Query Result, and returns a sorted Query Result list.

2.5Presentation Module

Basically, this module applying the MVC (Model/View/Control) design pattern. The UI, action handling and data should be separated, because this provides better mechanisms for handling UI has great number of actions, as well as ensure the consistency of data.

At lease following items should be considered when implementing the user interface

- A button allows user select folder or image to index it to database
- A button allows user select different features for query
- A button allows user select query sample image
- A painting board allows user draw an image

- A table or list to display the result images.

3.Feature Extractions and Methodology

The following sections will mainly discuss on the feature extraction that implemented in IMSmart v 1.0

Gray Level Co-occurrence Matrix

Gray Level Co-occurrence Matrix (GLCM), one of the most known texture analysis methods, estimate image properties related to second-order statistics. We used GLCM techniques for texture description in experiments with 14 statistical features extracted from them but for

1. ComputeCo-occurrence Matrixes for images in the database and also the query image.

   Four matrices (0, 45,90and 145) will be generated for each image. (Refer to detailed algorithm see [1])

2. Built up 4×4 features form previous Co-occurrence Matrixes (Figure 1)

Feature	Formula
Energy	$\sum_i \sum_j P^2(i,j)$
Entropy	$\sum_i \sum_j P(i,j) log P(i,j)$
Contrast	$\sum_i \sum_j (i-j)^2 P(i,j)$
Homogeneity	$\sum_i \sum_j \frac{P(i,j)}{1+\|i-j\|}$

Figure 3: four main features used in feature extraction

Colour Histogram

Color is the most widely used feature because it is the intuitive feature compared with other features and easy to extract from image. However, CBIR system based on color feature often result in disappointment, because it uses global color feature which cannot capture color distributions or textures within the image sometimes.

To improve the preferment of the color extraction IMSmart divides color histogram feature into global and local color extraction. Local color histogram can give some sort of spatial information, however the cons with that it use very large feature vectors.

Geometric Moments

This feature use only one value for the feature vector, however, the performance of current implementation isn't well scaled, [2] which means when the image size become large, it takes very long time to computer the feature vector. The pros of using this feature combine with other features such co-occurrence, which can provide a better result to user.

Average RGB

The objective of use this feature is to filter out images with larger distance at first stage when multiple feature queries involves. Another reason of choosing this feature, because it uses a small number data to represents the feature vector and it also use less computation compare to others. However, the accuracies of query result could be significantly impact if this feature isn't combined with other features.

Colour Moments

This feature has very reasonable size of feature vector, and the computation isn't expensive, [4] Colour Moments are measures that can be differentiate images based on their feature of colour, however, the basic of colour moments lays in the assumption that the distribution of colour in an image can be interpreted as a probability distribution. On pros of it is its skewness can be used to measure of the degree of asymmetry in the distribution.

4. Retrieval Experiments and results

4.1 Texture Database

Totally 1000 images which classified into 10 different categories, antiques, cars, desert, dogs, fashion, lizard, skiing, sunsets, waterfall, and workshop. Each category has 100 images. Each image is 384*256 pixels.

The similarity between images is estimated by summing up Euclidean distances between corresponding features in their feature vectors. Images having feature vectors closest to feature vector of the query image are returned as best matches

4.2 mage retrieval using multiply methods

Each image in the database is indexed before query; they are represented by the indices of features. In the retrieval, images in the database, called target images, are ranked in descending order of similarityto the query image; the ranks are presented as stars beyond every image.

The similarity between images is estimated by summing up Euclidean distances between corresponding features in their feature vectors. Images having feature vectors closest to feature vector of the query image are returned as best matches.

4.3 Experimental Results

Ten images randomly selected from the database as queries, for each query, the precision of the retrieval at each level of recall are obtained.

Three main features, global color histogram, co-occurrence and Geometric Moments are separately used each time in query.

As the results (see the precision-recall chart below) we see that colour histogram is more sensitive in colour as other colour features, they consider more about colour and a lot of irrelevant images obtained just because they have similar color style. The shape feature Geometric Moments has the best values of recall; in every query this method found more relevant images than other features while the precision is poor because a lot of irrelevant images also presented. When combine more than one features together the performance is better in both precision and

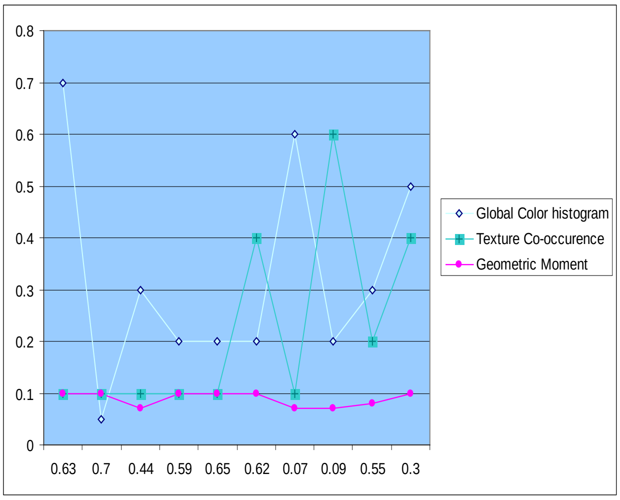


Figure 4: Experiments results

4.4 Comparison with peer group

After compare with C++ group, we figure out the result

1: Index more efficient

Our system index 1000 sample images need 12 minutes, but C++ group index 1000 sample images need more than 30 minutes.

2: Statable

Our system is more statable, already all the bugs

3: reusable
Compare with C++ group, they provide limited sample image, query from limited image database, but our group can query any sample image, can index any image folder, more reusable

4: convince GUI
Compare with C++ group, we provide more convince user interface, easy for user to use our system.

5: Compare with C++ group, we provide more searching features, they just provide HSV intersection and MTM Euclidean distance.

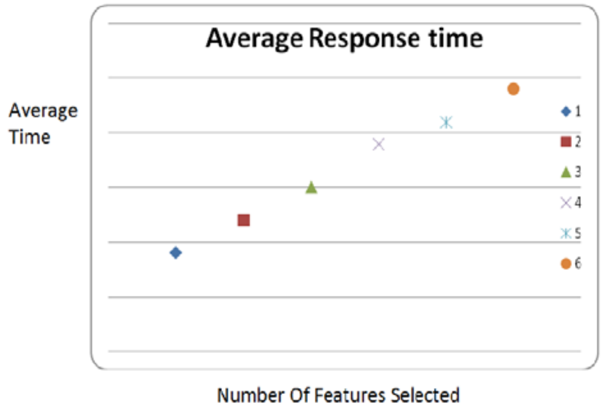
6: Feedback query
We provide User feedback Query, user can research from result, increase the accuracy.

7: Better User feature.

5. Performance analysis

This section provides detailed performance evaluation of IMSmart content-based image retrieval system; we adopted both the single value measure method, which mentioned by Berman & Shapiro (1990) measure whether the "most relevant" image is in either the first 50 or first 500 images retrieved. 50 represents the number of images returned on screen and 500 is an estimate of maximum number of images user might look at when browsing.

We also calculated the error rate, as Hwang et al. (1999) states that the error rate is the number of non-relevant images retrieved divide by the number of total retrieval imaged. Furthermore, we also evaluate the retrieval efficiency, as defined by Muller &Rigoll (1999), if the number of image retrieved is lower than or equal to the number of relevant images this value is the precision, otherwise it is the recall of a query. (Please refer to figure 4).Another performance analysis with response time,



In this diagram, the x-axis represents the number of features selected, and the y-axis represents the average time IMSmart token to search over 1000 images in the database. During our experiments, we realized that the lower-features take lesser time than the high level features, that possibly due to the expensive computation of feature extraction for texture or shape.

5.1 Highlight system features

Some notable features that supported by IMSmart

- High throughput lead to short response time
- Result images are ranked based on their percentage of similarity with the sample image. Friendly user interface to display the ranking result in stars.
- Users can provides feedback information about query result
- Support painting customized images

- Frequently used images are cached in memory for speedup display
- Plug-in framework for feature extraction module, new features can be added dynamically.

6.REFERENCES

1] Jerzy Bala, "Combining structural and statistical features in a machine learning technique for texture classification" 1990 ACM 089791-372-8/90/0007/0175

[2] Ying Liu, DengshengZhangt, Guojun Lu, Wei-Ying Ma, "Study on Texture Feature Extraction in Region-Based Image Retrieval System" 1-4244-0028-7/06 02006 IEEE

[3] Yo-Ping Huang, Tsun-Wei Chang, and Chi-Zhan Huang "A Fuzzy Feature Clustering with Relevance Feedback Approach to Content-Based Image Retrieval" 0-7803-7785-0/03/ 2003 IEEE

[4] Sagarmay Deb, Yanchun Zhang, "An Overview of Content-based Image Retrieval Techniques" 0-7695-2051-0/04/ © 2004 IEEE

[5] Xiang-Yu Huang, Yu-Jin Bang, Dong Hu, "Image Retrieval Based on Weighted Texture Features Using DCT Coefficients of JPEG Images" 0-7803-8185-8/03/ 2003IEEE

[6] KinhTieu, Paul Viola "Boosting Image Retrieval" Published in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2000

[7] Dengsheng Zhang, "Improving Image Retrieval Performance by Using Both Color and Texture Features" 0-7695-2244-0/04 2004 IEEE

[8] B.S.Manjunathi, W.Y. Ma "Texture Features for Browsing and Retrieval of Image Data" IEEE transaction on pattern analysis and machine intelligence, VOL. 18, NO. 8, AUGUST 1996"

[9] Yong Rui, Thomas S. Huang, SharadMehrotra, "comtent-based image retrieval with relevance feedback in mars" 0-8186-8183-7/91997IEEE

[10] Yong Rui, Thomas S. Huang, Shih-Fu Chang, "Image retrieval: past, present, and future"

[11] Tao Dacheng, Li Xuelong, Yuan Yuan, Yu Nenghai, Liu Zhengkai, Tang Xiao-ou, "A set of novel texture features based on 3D co-occurrence matrix for content-based image retrieval" 2002 ISIF

[12] Henning Muller, David McG. Squire "Performance Evaluation in Content-based image retrieval: overview and proposeals"

[13] DISCOVER http://www.cse.cuhk.edu.hk/~miplab/ last access 12/04/2007