CONTENT BASED IMAGE RETRIEVAL USING COLOR, TEXTURE & SHAPE

Shankar M. Patil
Department of information Technology, BVCOE,
Navi Mumbai, India
smpatil2k@gmail.com

I. **ABSTRACT- Images contain information in a very dense and complex form, which a human eye, after years of training, can extract and understand. The main goal is to extract form an image a set of composing objects or real life attributes. This information is inferred from low-level physical and mathematical properties of the image using a complex model of the reality the image reproduces. The existing systems based on domain specific models are incapable of retrieving images from heterogeneous collections or in the case they are only based on some abstract perceptually semantic feature like texture. Most systems use this lower level approach to retrieve images from heterogeneous collections** [4].

Besides containing a large quantity of complex data, images are also of very large dimensionality. Methods like comparing and correlating pixels that operate on the image directly are seldom powerful (costly in terms of time complexity) enough to full the users requirements. The usual approach to overcome this problem is to extract from the image a certain number of relevant features, which reduces the dimensionality, yet preserving useful information. These features are then considered as components of a feature vector, which makes the image, correspond to a point in a much lower but still high dimensional abstract space called the feature space. Two images are then considered similar if their feature vectors lay close in the feature space. The features that are extracted usually fall in three general categories color, shape and texture.

Keywords: color, shape, texture, LSI, CBIR, Relevance Feedback, histogram.

II. INTRODUCTION

The content-based image retrieval system retrieves the stored images from the database by comparing the features of the query image against the images in the collection. The system first extracts and stores the features of the query image then it go through all images in the database and extract the features of each image. The results are the images that its features are most similar to the query image.

Image Retrieval is a domain of increasing and crucial importance in the new information based society. Large and distributed collections of scientific artistic technical and commercial images are becoming a common ground thus requiring sophisticated and precise methods for users to perform similarity and semantic based queries.

A. LSI in Image Retrieval Selecting Features

The proposed a method to give an idea of occurrence count for low level features like color histogram bins or texture contrast by the following steps.

1. Pick a representative training set of documents for each feature or simply the whole collection if it is possible
2. Compute the $_j$ and standard deviation $_j$ of the feature s value across the training set.
3. For each image j in the collection define the occurrence count O_{ji} of feature f as

$$O_{fi} = \left\{ \begin{array}{ll} \left\lceil \dfrac{Val\,fj - \mu f}{\sigma f} \right\rceil & if\ Val\,fj > \mu f \\ 0 & Otherwise \end{array} \right\}$$

Where, value j is the value of the feature in image j.

If the feature value val has a normal distribution an assumption justified for most features usually encountered then the central limit theorem allows us to measure the difference between the realization value j of value and the mean by the previous formula and to use this as the “occurrence count” [4].

III. HOW THE SYSTEM WORKS

CBIR system is equipped with database of images with different set of features. User has interface to add image in the image database and retrieve images from the database.

A database stores all the images from which the image similar to the query instance is to be retrieved. When a query instance is entered in the input panel, the image feature for that image is derived and this is then compared with the all image features of the images stored in the database. The image whose feature matches closest to that of the query instance will be the output. The larger the image database slower is the retrieval process [2].

IV. THE SYSTEM ARCHITECTURE

This section presents the components and the interaction necessary to offer a flexible and distributed functionality to the LSI Image Retrieval system. The important consideration was that a user-friendly interface was needed for formulating complex queries and presenting data in a visual and straightforward way. Thus decided that a MATLAB 6.5 would be the easiest and in fact only solution satisfying our needs. The system is composed of three main blocks [4].

1. Feature extraction.
2. Program/Server that runs the queries.
3. Clients construct the queries.

Feature Extraction & Construction of the Index

This part of the system corresponds to the parser in a TR system; it is composed of several stand alone programs written in MATLAB 6.5. The main tasks of these programs are to.

1. Extract numerical feature values.
2. Calculate and apply weighting and create Feature index.
3. Compute the singular value of the Features.
4. Construct the projections of the documents to create the index

The functionality of the first subcomponent is to calculate the feature values of the image database. Several weighting schemes have been implemented in the second component, but the empirically best results, as expected where given by the singular value scheme which defines L(i, j) and G(i) [4].

The correlation of rows (terms) is evident because neighboring histogram bins, for example, contain similar values. The small yet surprising correlation of columns (documents) is due to the fact that the images were introduced in a hard disk and neighboring images on one file were often in fact similar.

The quality measure of the approximation is calculated by precision and call, were we can see that with only about 50 singular triplets the difference between the original and the approximation becomes very small Several different matrices have been also computed during the work and display very similar decay of the curves.

V. IMPLEMENTATION

A. Image database

The Image collection is obtained from several World Wide Web sites. The database of 700 images is being used. Images are divided into different categories and each category contains similar type of images. This will become useful when the program is being tested because when the user select a query image from one category in the database and the program returns the retrieved images, the user will know good is the program by comparing the number of retrieved images and the number of images in the same category as the query image [7, 8, 9].

To reduce the number of calculations at run-time, every image in the database should be pre-computed. The information needed to be pre-computed is Image feature, e.g. color, Texture and shape.

Image database stores all the images from which the images similar to the query instance will be retrieved. The images will be stored in JPEG format. The larger the database, slower will be the retrieval process.

B. Feature Extraction

a) Color

A content-based image retrieval system is presented that computes color similarity among images i.e. it supports querying with respect to color. Color is one of the most important features of objects in image. Each pixel in an image has a three-dimensional color vector and different color space approaches exist to represent color information.

Each image in the database is computed to obtain the color histogram, which shows the proportion of pixels of each color within the image. The color histogram of each image is then stored in the database. When the user does the search by specifying the query image, the system registers the proportion of each color of the query image and goes through all images in the database to find those whose color histograms match those of the query most closely.

This method is about evaluating the color similarity between the query image and other images in the database [2, 4].

b) Color histograms

The color histograms are used to represent the color distribution in an image. Mainly, the color histogram approach counts the number of occurrences of each unique color on a sample image. Since an image is composed of pixels and each pixel has a color, the color histogram of an image can be computed easily by visiting every pixel once. By examining the color histogram of an image, the colors existing on the image can be identified with their corresponding areas as the number of pixels.

Histogram search characterizes an image by its color distribution, or histogram. Many histogram distances have been used to define the similarity of two color histogram representations.

The color histogram is implemented using a one-dimensional array. The index of the array represents the color frequency. An element of the array represents the number of pixels with the color frequency represented by the array index.

Formula for Histogram comparison

$$D = \sum_{i=1}^{256} \left(\left| count1[i] - count2[i] \right| \right)$$

Where, count1 – histogram for the query image

count2 – histogram for the image in the database

D= DISTANCE

The image in the database whose histogram when compared with the histogram of the query image gives minimum distance will be displayed at the output panel.

C. Texture

Texture is a feature that is quite difficult to describe, and subjected to the difference of human perception, and it is hard to extracted by segmentation, because segmentation unable to extract the whole texture but the texture element.

Texture Descriptors

Another set of features computed are the texture descriptors, only out of the proposed descriptors were retained mainly due to computation costs. Let for instance P_d be the gray level (g g) co occurrence matrix corresponding to the 45 anti-causal direction. g is the number of distinct gray levels in the block and p_{ij} is the probability of having gray levels i and j at the distance d in this direction The descriptors all repose on this matrix and are [1, 4].

1. Mean
2. Standard deviation
3. Angular second moment
4. Inverse difference moment
5. Sum average
6. Contrast
7. Correlations
8. Sum variance

These descriptors have mainly been used to decide on the number of uniform or textured or coarse blocks in the image. Each image is in fact subdivided into a regular grid of non-overlapping blocks. The dimensions of the grid are 16 by 16 and for the moment all images are scaled to the same square size once again 16 by 16. Using fixed size blocks and variable size images would probably yield different results. To decide whether a given block is uniform or textured has low or high contrast correlation etc the mean and variance of the descriptor are computed and a similar formula to is applied. In contrast to color histogram bins where the absence of a color was not considered to be a feature in this case if the value is significantly lower than the mean it is also considered as a feature.

Texture has significantly richer information than color histograms and corresponds to human perception rather well the problem is that it is very sensitive to transforms such as scaling illumination and view angle. Another drawback is that the number of texture related features is only 32 whereas for color and luminance we total 1200 features. This difference is alleviated by the log-entropy weighting scheme because texture information has higher discriminating power.

ALGORITHM

1. Read the Query input image name.
2. Convert color image into grayscale image.
3. Divide the Query image into 16 by 16 blocks.
4. Calculate the query image feature as mean, max and min.
5. Finally compute the covariance of the Query image.
6. Load Covariance features of the image database.
7. Compare the difference between Query image feature and the features of the image database.
8. Sort the distance of the features of the image database.
9. Retrieves the closest distance images from the image database.
10. Display the retrieval images in the output panel.

D. Shape

Shape is also an important low-level feature in image retrieval system; since an object, in most case, can form by a set of shape (e.g. a car is consisted of a few rectangles and a few circles), most similar objects have a high correlation in the set of shapes.

Shape-based image retrieval should extract the shapes from images by segmentation, and classify the shape, where each shape should have their own representation and should variant to scaling, rotation, and transition [2, 3].

In shape-based image retrieval the user need to choose an reference image or sketch a desired shape, since the user may not only want the shape that exact matched, so shape based image retrieval should be able to identify similar shapes.

MOMENTS

In general, the moment's technique provides a better recognition and perimeter and area method. In this case, the moments of gray levels at each pixel in the image is calculated in the same way as mechanical moment [2].

$$M_{ij} = \sum_{(x,y)} x^i y^j f(x,y)$$

Where f(x,y) is an image function and equals to 1 if the pixel at position (x,y) is on otherwise equals to 0.

Algorithm

1. Read the Query input image name.
2. Convert color image into grayscale image.
3. Calculate the edge of the query input image.
4. Compute the moment of the query input image.
5. Integrate the moment features of given images row and column wise.
6. Load moment features of the image database.
7. Compare the difference between moment feature of the query image and moment features of the image database.
8. Retrieve the closest difference images from the image database.
9. Display retrieval images on the output panel.

Recall and Precision Evaluation

Testing the effectiveness of the image search engine is about testing how well can the search engine retrieve similar images to the query image and how well the system prevents the return results that are not relevant to the source at all in the user point of view.

The first measure is called **Recall**. It is a measure of the ability of a system to present all relevant items. The equation for calculating recall is given below:

$$\text{Recall} = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}}$$

The second measure is called **Precision**. It is a measure of the ability of a system to present only relevant items. The equation for calculating precision is given below.

$$\text{Precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}}$$

The number of relevant items retrieved is the number of the returned images that are similar to the query image in this case. The number of relevant items in collection is the number of images that are in the same particular category with the query image. The total number of items retrieved is the number of images that are returned by the search engine.

Now the values of recall and precision can be calculated. The test should be repeated by selecting other query images from other categories in the database and this will provide the user with many values of recall and precision. These values are then plotted in the **Recall-Precision Graph**.

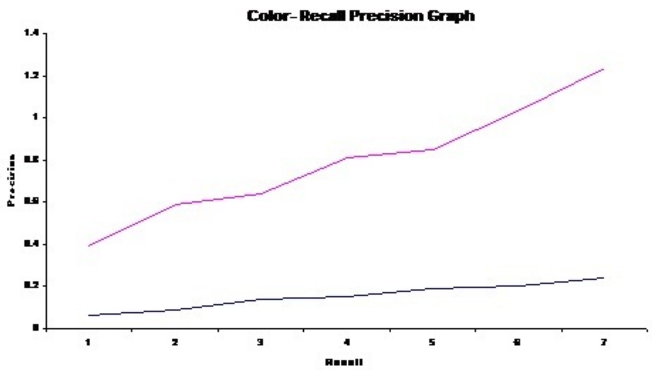

Figure 1. Recall Precision Graph for Color

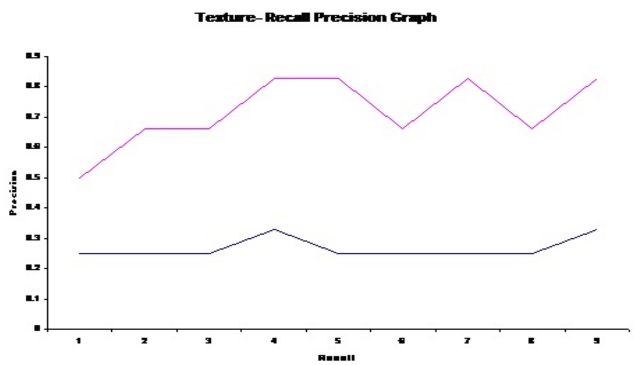

Figure 2.Recall Precision Graph for Texture

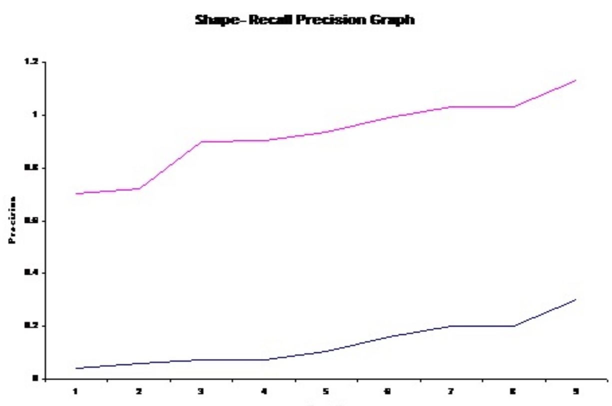

Figure 3. Recall Precision Graph for Shape

The graph above figure 7.10.1, 7.10.2, and 7.10.3 shows typically Color, Texture and shape of the recall-precision graph. As recall increases (more relevant items are retrieved), precision decreases (more non-relevant items are retrieved). This graph can be used to compare the effectiveness between two image search engines by plotting recall-precision curve of each system on the same graph. The system's curve that is closest to the top right-hand corner of the graph is where the recall and precision are maximized and therefore the best performance [2].

There is also a combined measure of recall and precision called **van Rijsberger's measure**. The formula is given below:

$$E = \frac{1 - [(1 + b^2) \times P \times R]}{[b^2 \times P + R]}$$

where E = van Rijsbergers measure
P = Precision
R = Recall
b = relative importance to the user of recall and precision

Using this measure, the system can be tested in the case of user pays more attention to either the recall or the precision. For example if b = 2, the user is twice as interested in recall as in precision. If b = 0, the user is not interested in recall so E = 1 – P. If b =∞, the user is not interest in precision so E = 1 – R.

VI. RESULT

a) Color Result

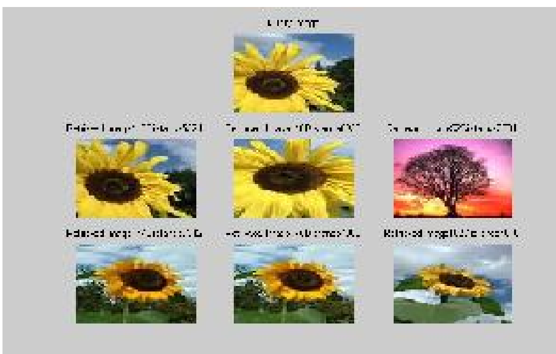

Figure 4. Result of the Color Feature

b) Shape Result

Input Query Image Number is 135.jpeg and its result is shown below.

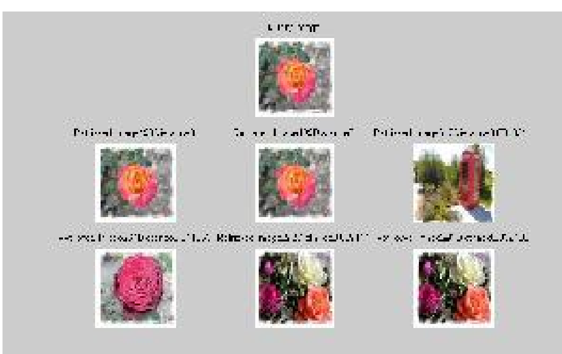

Figure 5. Result of the Shape Feature

c) Texture Result

Input Query Image Number is 58.jpeg and its result is shown below.

Figure 6. Result of the Texture Feature

VII. CONCLUSION

We have tried to mark the difference between low level and high level approaches to the problem of searching image collections. LSI cannot easily be put in one of these categories. It has proved to be a very promising method for image retrieval, because it integrates abstract and often incommensurate features into a unique framework. The gained value for the user is more natural querying and completely transparent integration of textual annotation much effort was given to improve performance by user relevance feedback because we believe it is of primary importance for efficient and powerful image retrieval systems. We have exposed our attempts to define a discriminating large and robust feature set. Finally we have shown that the field of open questions is still very vast and much interesting work can be done.

ACKNOWLEDGMENT

I take this opportunity to express my gratitude towards Dr. M. Z Shaikh, Principal, for his constant support and I would also like to thank, Dr. V. J. Kadam, Director, Bharati Vidyapeeth Educational Complex, Navi Mumbai for providing opportunity to submit this paper. I would like to thank Hon'able Dr. Pantangraoji Kadam, Founder, Bharati Vidyapeeth and Chancellor Bharat Vidyapeeth Deemed University, standing Cabinet Minister in the Govt. of Maharashtra State, and Hon'able Dr. Shivajiraoji Kadam, Vice Chancellor, Bharati Vidyapeeth Deemed University for providing an opportunity to present this paper.

REFERENCES

[1] Eakins J. P., Graham M., A report to the JISC Technology Applications Programme, 1999.
[2] Pravi Techasith *Image Search Engine Project Outsourcing Exercise*
[3] John Eakins, Margaret Graham, University of Northumbria at Newcastle.
[4] Zoran Pecenovic, *Image Retrieval using Latent Semantic Indexing*, 1997
[5] MATLAB Image Processing Toolbox, User's Guide
[6] Zoran Pe¡cenovi´c, Minh Do, Serge Ayer, Martin Vetterli, *New Methods for Image Retrieval*, *L*aboratory for Audio-Visual Communications, Swiss Federal Institute of Technology (EPFL) CH-1015 Lausanne, Switzerland
[7] http://www.wallpaperdave.com
[8] http://www.free-pictures-photos.com
[9] http://www.photodefleur.com