Integrated CLUE with Genetic Algorithm An Efficient Image search Engine, 2013

Ch.Satish
Assistant Professor
Department of Electronics and Communication Engineering
Pragati Engineering College., Surampalem , AP, India.

M.Vijaya Bhaskar
Department of Electronics and Communication Engineering
Pragati Engineering College., Surampalem , AP, India.

S. Prabhu Das
Associate Professor
Department of Electronics and Communication Engineering
Srinivasa Institute of Engg;&Tech., Cheyyeru,Amalapuram , AP, India.

Abstract - In typical content-based image retrieval (CBIR) system, target images (images in the database) are sorted by feature similarities with respect to the query. Among various low-level features, the color and texture information has been extensively studied because of their invariance with respect to image scaling and structure.
Individual features such as color, texture and shape are having their own advantages and limitations. And Individual features are not enough for best retrieval. If multi features are used alternatively for retrieval, it takes long time to get the matched image for large sized databases.

So, in order to improve the system's retrieval performance and to reduce the retrieval time ,this paper introduces a new technique, to obtain the unique multi feature fusion similarity score using genetic Algorithm by unsupervised learning. In addition It improves user interaction with efficient image retrieval systems by fully exploiting the similarity information. It retrieves image clusters by applying a Multidimensional Indexing to a collection of images in the vicinity of the query. Clustering is dynamic. In particular, clusters formed depend on which images are retrieved in response to the query. The performance of an experimental image retrieval system using proposed algorithm is evaluated on a database of around 60,000 images from COREL.

Keywords — Content-based image retrieval (CBIR), image classification, similarity measure, clustering, unsupervised learning. Image Fusion, Genetic Algorithm

I. INTRODUCTION

In the last few years, the rapid growth of the Internet has enormously increased the number of image collections available. Early image retrieval methods locate the desired images by matching keywords that are assigned to each image manually. However, as a result of the large number of images in collections, manual processing has become impractical. As well, because we are unlikely to foresee all the query keywords that will be used in a retrieval process, it is impractical to assign keywords to every image, so the effectiveness of classic image retrieval is very limited.

In Integrated CLUE shown in Figure 1, each image that is stored in the database as a cluster has its features extracted and compared to the features of the query image. It involves three steps:

- **Feature Extraction:** The first step in the process is extracting image features to a distinguishable extent.
- **Fusion with Genetic Algorithm:** The second step involves that the distinguish features of both color and texture are fusion with Genetic Algorithm to get similarity score.
- **Matching:** The third step involves matching these similarity score vector to yield a result that is visually similar.
- **Clustering:** In clustering[1] the similar image features are grouped from the matched images to query image.

Motivation:

The goals for this paper have been the following.

The primary goal the paper is to reduce the computation time and more user interaction. In our proposed system we compute texture feature and color feature for compute the similarity between query and database images. This integrated approach will reduce the output results by the formation of Clusters[7].

The secondary goal is to reduce semantic gap between high level concepts and low level features.

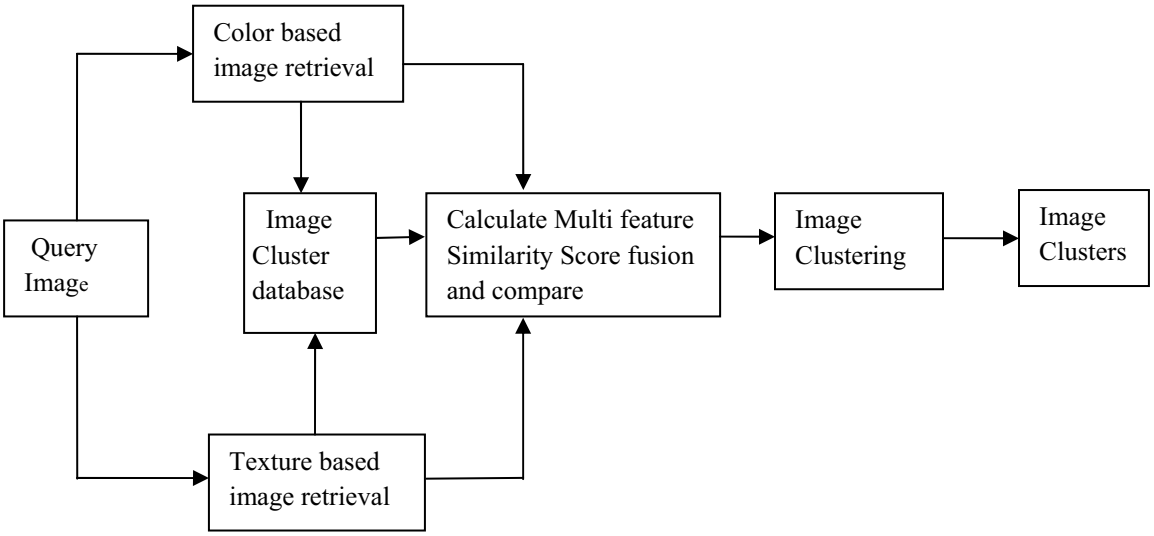

Figure 1.Block Diagram of Integrated CLUE

A third goal is to evaluate their performance with regard to speed and accuracy.

Organization of the paper:

This paper is organized as follows: Chapter 2 presents different Image retrieval system techniques that are used for querying by texture and color content of images. Not only the implemented features but also a general framework for proper extensions of the system is also discussed. Clustering and indexing in Database in Chapter 3 and the performance experiments of the content-based retrieval system are given in Chapter 4 and finally, Chapter 5 concludes with Future scope.

II. PROPOSED ALGORITHM

Several systems extracting features currently exist are COREL,QBIC,VIR Image Engine Visual SEEK and Web SEEK, NeTra ,MARS or Multimedia Analysis and Retrieval System enables image retrieval based on primitive features such as colour, texture and structure.

Color and its Representation[2,6]:

Color Histogram is the low level feature to represent the RGB color contents of the image. More formally, the color histogram is defined by,

$$h_{A,B,C}(a,b,c) = N \cdot \mathbf{Pr}ob(A = a, B = b, C = c) \quad (1)$$

Where *A, B* and *C* represent the three color channels (R,G,B or H,S,V) and *N* is the number of pixels in the image. Computationally, the color histogram is formed by discretizing the colors within an image and counting the number of pixels of each color. The developments of the extraction algorithms follow a progression as:

(1) Selection of a color space,
(2) Quantization of the color space,
(3) Computation of histograms,
(4) Derivation of the histogram distance function,

(5) Identification of indexing shortcuts. Each of these steps may be crucial towards developing a successful algorithm.

An example of a color histogram in the HSV color space can be seen with the following Figure 2:

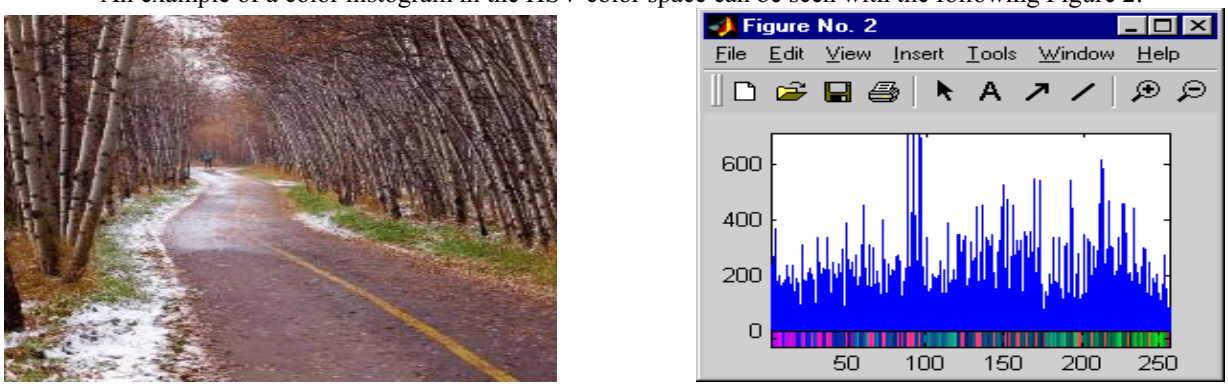

Figure. 2 Sample Image and its Corresponding Histogram

To view a histogram numerically one has to look at the color map or the numeric representation of each bin.

Quantization in terms of color histograms refers to the process of reducing the number of bins by taking colors that are very similar to each other and putting them in the same bin. For the purpose of saving time when trying to compare color histograms, one can quantize the number of bins. Obviously quantization reduces the information regarding the content of images.

There are several distance formulas for measuring the similarity of color histograms. Three distance formulas that have been used for image retrieval including histogram Euclidean distance, histogram intersection and histogram quadratic (cross) distance.

Let h and g represent two color histograms. The Euclidean distance between the color histograms h and g can be computed as:

$$d^2(h,g) = \sum_A \sum_B \sum_C (h(a,b,c) - g(a,b,c))^2 \qquad (2)$$

Texture and its Representation[2,3]*:*

Texture is one of the most important defining features of an image that describes visual patterns shown in Figure 3, each having properties of homogeneity. It contains important information about the structural arrangement of the surface. In short, it is a feature that describes the distinctive physical composition of a surface. Texture properties include Coarseness, contrast, Directionality, Line-likeness, Regularity, Roughness.

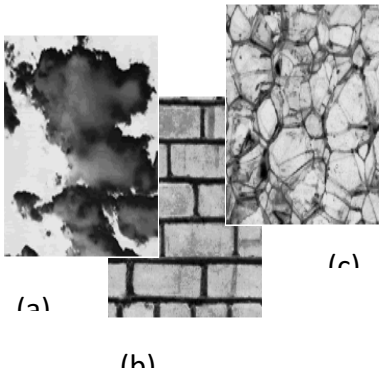

Figure.3 Examples of Textures

There are three principal approaches used to describe texture; statistical, structural and spectral. For optimum classification purposes, what concern us are the statistical techniques of characterization. The most popular statistical representations of texture are:

- Co-occurrence Matrix
- Tamura Texture
- Wavelet Transform

Co-occurrence Matrix Originally proposed by R.M. Haralick , the co-occurrence matrix representation of texture features such as Angular Second Moment, Contrast, Correlation, Variance, Inverse Second Differential Moment, Sum Average, Sum Variance, Sum Entropy, Entropy Difference Variance Difference Entropy Measure of Correlation Measure of Correlation Local Mean explores the grey level spatial dependence of texture .

Tamura explored the texture representation using computational approximations to the three main texture features of: coarseness, contrast, and directionality. Each of these texture features are approximately computed using algorithms.

Textures can be modeled as quasi-periodic patterns with spatial/frequency representation. The wavelet transform transforms the image into a multi-scale representation with both spatial and frequency characteristics. According to this transformation, a function, which can represent an image, a curve, signal etc., can be described in terms of a coarse level description in addition to others with details that range from broad to narrow scales. Examples of wavelets are Coiflet, Morlet, Mexican Hat, Haar and Daubechies. Haar is the simplest and most widely used, while Daubechies have fractal structures and are vital for current wavelet applications.

The standard Pyramid Wavelet Transform is shown in the Figure 4. The first step is to resize the image size into 256X256 in a matrix format. Then the pyramid wavelet transform is applied to get the sub bands of the image. To find the energy measures of the image Daubechies filter is applied.

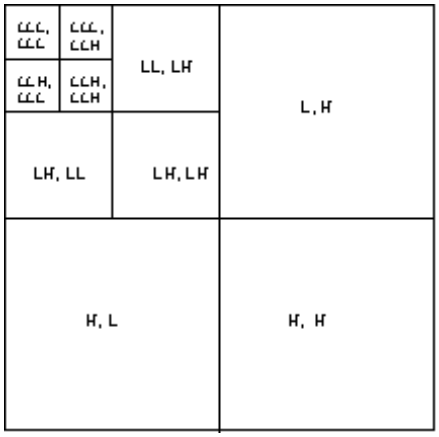

Figure.4. Pyramid Wavelet Transform

The decomposition is applied to 6 levels so that we can able to get the low frequency contents in the LL sub band and other frequencies in LH, HL and HH bands separately. Finally we will get the 4X4-sized image.

Once the wavelet coefficients of an image are available, features are computed from each sub-band, resulting in 19 features for each image. The mean μ is the energy measure used to compute the features, then the feature vector f, for a particular image is calculated using the given formula.

$$f = [\mu_{mn}], \quad n \neq 1 \text{ except for the coarsest level, } m=6. \qquad (3)$$

Where μ_{mn} is the energy measure for the decomposition level and the sub bands. Now we get the energy coefficients and stored in the database.

When the user gives the query image then it will be converted into the same above operations and finally gives the energy measure coefficients.

The distance between the two images is calculated using Euclidean distance classifier and the retrieved images are fusion with genetic algorithm.

III MULTI-FEATURE SIMILARITY SCORE FUSION GENETIC ALGORITHM

Since the physical meanings of different features are different, and value ranges are totally different, similarity scores of different features cannot be compared. So, before multi-feature similarity score are fused, they should be normalized. Similarity scores can be normalized through the following ways. Let Q be the query image. By calculating distances between the query image and images in database, similarity score set $\{ S_i\}$ can be gotten, where i = 1,L,… N , N is the number of images in database. Thus, similarity score normalization can be implemented as

$$S_{Ni} = \frac{S_i - \min\{S_i\}}{\max\{S_i\} - \min\{S_i\}} \quad (4)$$

The results of multi-feature similarity scores is

$$S_{Fi} = \frac{S_{NCi} \cdot W_C + S_{NTi} \cdot W_T}{W_C + W_T}, \quad (5)$$

Where S_{Fi} is the fused similarity score, S_{NCi} is the normalized color feature similarity score, S_{NTi} is the normalized texture feature similarity score, W_c is the weight of color feature similarity score, and W_T is the weight of texture feature similarity score. By assigning appropriate values to W_C and W_T , a fine similarity score fusion can be gained.

During the course of similarity score fusion, a key problem is how to assign the weights of similarity score. It affects directly the retrieval performance of the system. It can be considered as an optimization problem to assign reasonably the weights of color feature similarity score and texture feature similarity score. That is to find the optimum in weight value space. So, this problem can be resolved by genetic algorithm. This paper proposed a similarity score fusion method using genetic algorithm. With genetic algorithm the weights of color feature similarity score and texture feature similarity score are assigned optimally.

A. Determination of solution space

The aim of fusing similarity scores is to assign the weights of color feature similarity score and texture feature similarity score to gain a better image retrieval performance. With the consideration of (5), the weight of color feature similarity score W_C can be a integer between 0 and I , where I is a positive integer. Without loss of generality, the weight of texture feature similarity score can be assigned to $I - Wc$. The positive integer I determined the accuracy of solution. The bigger the value of I is, the higher the accuracy of solution is. But this may take a long time to resolve, and vice versa. To resolve using genetic algorithm, the weights should be encoded. The solution should be expressed as a binary number. So generally the value I is taken as 2^L , where L is a positive integer, the encoding length of the solution.

B. Population Initialization

In genetic algorithm, the number of individuals in population and the initial values of the individuals will influence the solution greatly. In this paper, the number of individuals in population N is taken as $\sqrt{I}$. N is set a bigger value, the aim of which is to gain the optimal solution quickly. The individuals are initialized as follows. The solution space is divided into N equal portions, the centers of which are taken as the initial values of the individuals

C. Determination of fitness function

The fitness of individuals can be evaluated as follows. According to the weights W_C and W_T of N individuals, we can get N groups of image retrieval results. For every group, the top M images are considered. Total number of images is MN . By calculating occurrence frequency of images of every group in all images, the fitness of every individual is evaluated. Specific operations are as follows. Let N_{ikj} denote if kth image A_{ik} of ith group G_i is in jth group G_j or not. That can be formulated as

$$N_{ikj} = \begin{cases} 1, \mathrm{A}_{ik} \in G_j \\ 0, \mathrm{A}_{ik} \notin G_j \end{cases} \quad (6)$$

Then the occurrence frequency of *k*th image A_{ik} of *i*th group G_i in all *MN* images is

$$N_{ik} = \sum_{j=0}^{N} N_{ikj} \quad (7)$$

The occurrence frequency of all images of *i*th group G_i in all *MN* images is

$$N_i = \sum_{k=1}^{M} N_{ik} \quad (8)$$

The normalized version of it is

$$P_i = \frac{N_i}{\sum_{l=0}^{N} N_l} \quad (9)$$

The bigger P_i indicates that the images in *i*th group G_i possess a high proportion in all *MN* images, and the Solution is considered a good one. In this paper, it is taken as fitness function

D. Solving for optimal solution

The genetic algorithm is implemented in classic mode. The condition for ending the iteration is that the number of iteration is equal to 3. When the iteration is ended, the maximum *P** = max *Pi* is taken as the optimal solution. According to the optimal solution the weights W^*_C and W^*_T are assigned, then the image retrieval results with these two weights are taken as the ultimate retrieval results.

IV. INDEXING AND CLUSTERING IN DATABASE[4,5,7]:

For a large image collection, the storage space required for the metadata is usually non-trivial. The system must possess efficient techniques to compress the metadata. The MPEG-7 standard is becoming the most important standard to describe all kinds of metadata for both images and video data.

When a query is processed against a large image database, it is often unacceptable to compare the similarity between the query image and all the images, one by one. Because users only need images having high similarity to the example image, index structures, which can help to prevent sequential searches and improve query efficiency, should be used in CBIR systems.

Further, for frequently varied image databases, dynamic index structures are necessary. When the content of images is represented by low-dimension vectors and the distance between images is defined (possibly as a spatial distance calculated with the Euclidean distance), the R-tree and its families can be used to index images dynamically. When the distance is not defined as the vector space, or when the vector space is highly dimensional, or when what we have is only a distance function that can define a metric space, methods to index images based on the distance function in the metric space are available.

When a distance function can satisfy the metric properties – non-negativity, symmetry, and triangle inequality – M-trees, which are dynamically balanced trees, can also be applied to index objects. The novel clustering technique cluster the output images and select one representative image from each clusters.

V. EXPERIMENT AND RESULTS

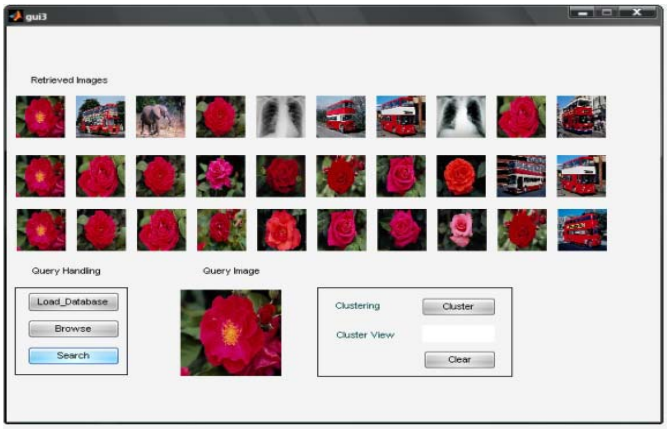

Figure. 5 Images Retrieved by Color for rose flower

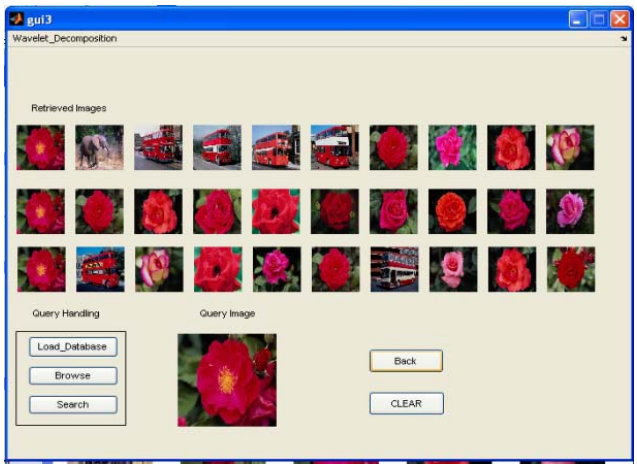

Figure. 6 Images Retrieved by Texture for same rose flower

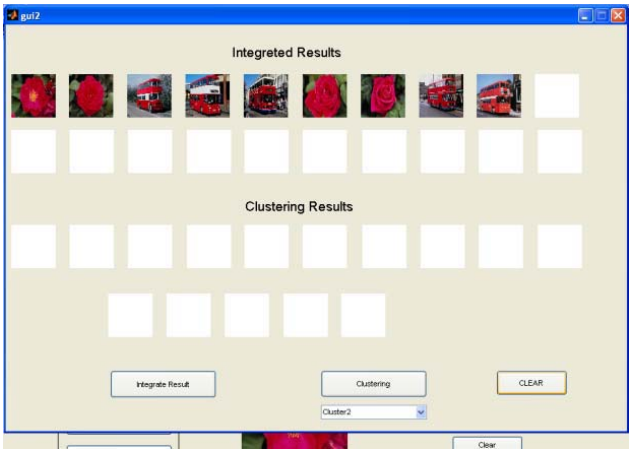

Figure. 7 Integrated and Clustered Results

Note that Images in sample Database are indexed as all flowers belong to cluster-2, Elephants to cluster-1, buses to cluster-3, Dinosaurs to cluster-4, and x-ray images to cluster-5.

V.CONCLUSION

We have designed and implemented an image retrieval system that evaluates the similarity of each image in its data store to a query image in terms of textural and color characteristics, and returns the images within a desired range of similarity. From among the existing approaches to texture analysis within the domain of image processing, we have adopted the statistical approach to extract texture features from both the query images and the images of the data store. *Energy*, *entropy*, *inertia*, *correlation* and *local homogeneity* have been selected as an optimal subset of the set of second order statistical features that can be extracted from *Spatial Grey Level Dependency Matrices*. Histogram intersection method has been used as the similarity measure between two feature vectors.

For the color content extraction, a well-known and powerful technique, Color Histograms are used. The expressiveness of this technique is accelerated via color space transformation and quantization, and the images are smoothed by the help of color median filtering, a famous method for neighborhood ranking. Hence, it also complies with texture extraction due to being related with the neighborhood property of pixels.

REFERENCES

[1] Yixin Chen*, Member, IEEE*, James Z. Wang*, Member, IEEE*, and Robert Krovetz 'CLUE: Cluster-Based Retrieval of Images by Unsupervised Learning',*IEEE Transaction on Image Processing, Vol. 14, No. 8, August* ,2005.

[2] Mianshu Chen, Ping Fu, Yuan Sun, Hui Zhang, 'Image Retrieval Based on Multi-Feature Similarity Score Fusion Using Genetic Algorithm' , *IEEE Transaction on Image Processing, Vol. 2, 978-1-4244-5586-7/10,2010*

[3] S. Belongie, C. Carson, H. Greenspan, and J. Malik. 'Color- and texture based image segmentation using EM and its application to content-based image retrieval'. In *Proceedings of the Sixth International Conference on Computer Vision*, 1998.

[4] Sharmin Siddique, "A Wavelet Based Technique for Analysis and Classification of Texture Images," *Carleton University, Ottawa, Canada, Proj. Rep. 70.593*, April 2002.

[5] Pravi Techasith, "Image Search Engine," *Imperial College, London, UK, Proj. Rep., July* 2002.

[6] D. Stan and I. K. Sethi, " Image Retrieval using a Hierarchy of Clusters," *International Conference on Advances in Pattern Recognition,* 2001.

[7] J. R. Smith and S. F. Chang, "Tools and Techniques for Color Image Retrieval," in Proceedings of the SPIE: *Storage and Retrieval for Image and Video Databases IV, 2670, pp. 381-392,* 1996.

[8] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H.-J. Zhang, " Image Classification for content-based indexing," *IEEE Trans. Image Process.*,*vol. 10, no. 1, pp. 117–130, Jan. 2001.*