An Optimized CBIR Using Particle Swarm Optimization Algorithm

Subhakala S.
Sri Krishna college of technology,
Coimbatore,India.

Bhuvana S.
Sri Krishna college of technology,
Coimbatore,India.

Radhakrishnan R.
Sri shakthi institute of Engineering and technology,
Coimbatore,India.

ABSTRACT
Storage and retrieval of images over a large database is an important issue. Content Based Image Retrieval system provides solution for this issue. In Content Based Image Retrieval(CBIR) similar images are retrieved using low level features such as color, texture, edge, etc that are extracted both from the query image and the database. In CBIR less amount of retrieval time with high accuracy is desired property. The proposed system achieves this property by using Particle Swarm Optimization algorithm. The proposed system consists of the following phases (i) Color feature extraction using (luminance(y), blue chrominance (u), red chrominance (v)) method (ii) Texture feature extraction using Grey Level Co-occurrence Matrix (iii) Edge feature extraction using Edge Histogram Descriptor (iv) Measurement of Similarity between Query image and the Database image using Euclidean Distance. (v) Optimization of retrieved result using Particle Swarm Optimization. In comparison with the existing approach, the proposed approach significantly improves the precision and recall of the retrieval system.

Keywords
Accuracy, Particle Swarm Optimization, Luminance, Chrominance, Edge Histogram Descriptor.

1. INTRODUCTION
The Content Based Image Retrieval is a method which uses visual contents to search images from large image repositories. According to user's interest, has been an active research area over the last few years. Users are exploiting the opportunity [1] to access remotely stored images in all kinds of new and exciting ways. However this has the problem of locating a desired image in a large and varied collection. This leads to the rise of new research and development area CBIR, the retrieval of images in the basis of features automatically extracted from the image themselves.

The increase in computing power and electronic storage capacity has lead to an exponential increase in the amount of digital content available to users
in the form of images and video, which form the bases of much entertainment, educational and commercial applications. Consequently, the search for the relevant information in the large space of image and video databases has become more challenging. How to achieve accurate retrieval results is still an unsolved problem and an active research area.

The currently available CBIR technique retrieves stored images from a collection of given images by comparing features [6] [7] and automatically extracted from the image themselves. The most commonly used features are color, shape, texture etc.

The proposed system uses color, texture [11] and edge feature extraction. From the extracted features similarity is measured using Euclidean distance. The results are optimized using Particle Swarm Optimization. This system will achieve the better retrieval accuracy.

1.1 Feature Extraction
Feature extraction is a form of dimensionality reduction. It reduces the input size. This approach is helpful when images are in larger size. Reduced feature representation is required for tasks such as query matching and similarity retrieval. Feature extraction [3] is very different from feature selection. Feature extraction consists in transforming arbitrary data, such as text or images, into numerical features usable for machine learning.

1.2 Similarity Measurement
To compare the similarity between images, distance between that images are measured. Example for a similarity measures such as Euclidean distance, histogram intersection etc.

1.3 Optimization
Optimization [5] is defined as a set of methods and techniques for the design and use of technical systems as fully as possible within the parameters. Optimization can be classified into two categories: local and global. The basic difference between local and global optimization is the size of region where optimization conditions hold. A local optimum has an extreme function value as compared to the points contained in the small neighbourhood. However, the global optimum has the extreme function value amongst all the points in the whole design space. Even though clustering algorithms are simple and effective, they are sensitive to initialization and easily trapped in local optima. There are many optimization algorithms like Particle Swarm Optimization to overcome the drawbacks of clustering techniques.
2. RELATED WORKS

In recent years many studies have been performed in Content Based Image Retrieval (CBIR). Lu et.al, proposed image retrieval technique based on features of image using color and bitmap of an image. For the purpose of retrieving more similar images effectively from the digital image databases, this proposed system uses the color distributions. Mean and standard deviations are the global characteristics of the images. Color, shape and texture are the local characteristics of the image. To improve the retrieval accuracy bitmap is used. This technique outperforms in terms of image retrieval accuracy and category retrieval ability. This system uses RGB for color extraction. RGB is not very efficient when dealing with real world images. Creation of keywords for each image is time consuming because of the size of database.

S.-B. Cho, et.al presented image retrieval process that dealt human preference and emotion by using an interactive genetic algorithm (IGA). In this method features are extracted from images using wavelet transform and provide the means to retrieve the image from large database. This works by creating a population of individuals that are represented by chromosomes. Here crossover and mutation operators are used to induce variations in the population. It requires several genetic operators such as cross over, mutation for better performance. It is difficult to retrieve an image that cannot be explicitly specified because it deals with emotion.

X.S. Zhou et al. proposed the Genetic Programming framework to discover a combination of descriptors. Color, shape, and texture descriptors are used to represent the content of database images. Local Similarity pattern (LSP) is used in retrieval process. Image descriptors are used for image searching process. Feature extraction and similarity computation are characterized as descriptors. In this method a new relevance feedback method for interactive image search is proposed. This method adopts generic programming approach to learn user preferences in a query session. This system allows only a linear combination of similarity values however, more complex combination may be needed to express the user needs moreover this system requires more iteration for user satisfaction.

James et al. proposed Wavelet-Based Image Indexing and Searching. In this method a new technique for image indexing is used. New algorithm uses partial sketch image searching capability for large image databases. This algorithm characterizes the color variations. Features extracted from the image are wavelet co-efficient and their variances. To improve the retrieval accuracy two stage algorithm is proposed. In first stage, crude selection is performed for query image. In second stage search is refined by performing match between feature and the selected image query. For better accuracy in
searching, two-level multi resolution matching may also be used. Masks are used for partial-sketch queries.

2.1 Motivation

In user oriented CBIR system, color feature is extracted using HSV method which requires more retrieval time. To improve on this problem YUV method is used which reduces the retrieval time compared to RGB and HSV. YUV method uses mean and standard deviation values to extract Y component, U component and V component. Interactive Genetic Algorithm uses genetic operators such as mutation and crossover which requires more computational cost for retrieving the images. To overpass this problem Particle Swarm Optimization technique is used.

3. PROPOSED WORK

In The proposed system applies a user oriented CBIR approach to image retrieval which is performed by extracting the feature [2] from the image as well as the database. Low level features such as color, texture, edge that are extracted both from the query image and the database. In order to improve the retrieval performance and accuracy, this system uses YUV (luminance(y), blue chrominance (u), red chrominance (v)) method for color feature extraction. Texture feature is extracted using Grey Level Co-occurrence Matrix method and edge feature is extracted using Edge Histogram Descriptor method. After extracting these features similarity is computed between the query image and the images in the database using Euclidean distance and images are retrieved which are optimized further using Particle Swarm Optimization. The proposed system consists of following phases: (i) Color feature extraction (ii) Texture and edge feature extraction (iii) Similarity Computation (iv) PSO Optimization

3.1 Color Feature Extraction

In this module, YUV [10] color space is used to extract the color feature from the image as well as in the database. After extraction mean and standard deviation is also calculated for YUV image. The mean of pixel color states the principal color of the image, and the standard deviation of pixel colors can depict the variation of pixel colors.

The "mean" is the "average" where adding up all the numbers and then divide by the number of numbers. Mean of pixel color states the principal color of the image. Sample mean and standard deviation are given in eqn (1) and (2) respectively.

\[
\text{Sample mean} = \frac{\sum X}{N} \quad (1)
\]
Standard deviation \( = \sqrt{\frac{\sum(x_1 - x_2)^2}{n-1}} \) \( (2) \)

Where

\( \sum x \) is the sum of all data values

N is the number of data items in population

n is the number of data items in sample

Algorithm:

Input: Query image

Output: YUV image

Steps:

- Query image is given as input
- Color image is converted into RGB and then into YUV using `rgb2yuv` function in MATLAB.
- For each row and column constant matrix value is multiplied with rgb matrix values to get YUV components of an image. The formula is given by
  \[
  \begin{pmatrix}
  Y \\
  U \\
  V \\
  \end{pmatrix} = \begin{pmatrix}
  0.30 & 0.59 & 0.11 \\
  -0.15 & -0.29 & -0.44 \\
  0.62 & -0.51 & 0.10 \\
  \end{pmatrix} \begin{pmatrix}
  R \\
  G \\
  B \\
  \end{pmatrix}
  \]

- Mean and standard deviation is calculated by using an inbuilt function in MATLAB for an YUV image.

3.2 Texture Feature Extraction

In this module [9], Grey Level Co-occurrence matrix method is used to extract the texture feature from an image. The texture feature such as energy which is used to compute the energy of grey scale images, entropy which is used to capture the textural information in an image, auto-correlation, and homogeneity are extracted from the image.

In a given offset, the co-occurrence matrix is a matrix is defined over an image to be the distribution of co-occurring values. Mathematically, a co-occurrence matrix C is defined over an n × m image I, parameterized by an offset \((\Delta x, \Delta y)\), as:

\[
C_{\Delta x,\Delta y}(i,j) = \sum_{p,q=0}^{m-1} \sum_{l=1}^{n} \begin{cases} 
1, & \text{if } I(p,q) = i \text{ and } I(p+\Delta x, q+\Delta y) = j \\
0, & \text{otherwise}
\end{cases}
\] \( (3) \)

Where i and j are the image intensity values of the image, p and q are the spatial positions in the image I and the offset \((\Delta x, \Delta y)\) depends on the direction used and the distance at which the matrix is computed d. The 'value' of the image originally referred to the grayscale value of the specified
pixel, but could be anything, from a binary on/off value to 32-bit color and beyond. Note that 32-bit color will yield a 232x232 co-occurrence matrix.

Algorithm:
Input: Query image
Output: Texture features

Steps:
- Query image is given as input
- Texture feature such as entropy, auto-correlation, contrast, etc are extracted.

3.3 Edge Feature Extraction

Edge Histogram Descriptor (EHD) describes [8] five types in the image namely horizontal, vertical, two diagonal and non-directional.

Algorithm:
Input: Query image
Output: Edge features

- The image space is divided into 16 (4x4) non overlapping sub images.
- For each sub images a histogram with five edge bins are generated. Totally 80 bins for the entire image is generated.

The role of the EHD is to provide primitive information on the edge distribution in the image.

3.4 Similarity Computation

In this module, Euclidean Distance method is used which computes the similarity between the query image and database images according to the aforementioned low level visual features. This method retrieves and presents a sequence of images ranked in decreasing order of similarity. As a result the user is able to find relevant images by getting top ranked images first.

The Euclidean distance formula is

\[ d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \]  \hspace{1cm} (4)

where \( p \) and \( q \) are length of two pixels.

Input: Query image
Output: set of relevant images

Steps:
- Query image is given as input
- Similarity is computed by using Euclidean distance method.
- Display the sequence of images.

3.5 Particle Swarm Optimization

Particle swarm optimization (PSO) [5] is an optimization technique inspired by social behaviour of bird flying or fish schooling.

In PSO each particle is associated with the best solution. This value is called pBest. When a particle takes all the population as its topological neighbours, the best value is a global best and is called gBest.
The PSO method appears to adhere to the five basic principles of swarm intelligence:

1. Proximity: In simple space and time computation the swarm must be able to perform.
2. Quality: It should be able to respond to quality factors.
3. Diverse response: It should not commit its activities along excessively narrow channels.
4. Stability: It should not change its behaviour every time.
5. Adaptability: It should be able to change its behaviour.

Applications of Particle Swarm Optimization

- Non convex search spaces
  Particle swarm optimization is able to deal with local minima and able to find the global optimum.
- Integer or discontinuous search spaces
  It does not require the space to be continuous but precautions need to be taken.

In PSO, Particle is spread throughout the search space randomly. In search space the particles are assumed to be flocking. Iteratively velocity and position of each particle is updated. Each particle possesses its local memory.

In PSO, particle is considered as potential solution to the optimization problem. The position of the particle is represented by \( X_i = x_{i1}, x_{i2}...x_{iD} \) in dimensional space \( D \). The velocity of a particle is given as \( V_i = v_{i1}, v_{i2}...v_{iD} \). Each particle has a local memory (pBest) which keeps the best position of the particle. Globally shared memory is represented as gBest. It represents the global position of that particle. Flying velocity of each particle is given in eqn (5). Particle position update is given in eqn (6).

\[
V_i = V_i + \varphi_1 \times rand \times (pBest_i - x_i) + \varphi_2 \times rand \times (gBest - x_i) \quad (5)
\]

\[
x_i = x_i + v_i \quad (6)
\]

Where, \( \varphi_1 \) and \( \varphi_2 \) are constants used for calculating the relative influences of the particle. Introduction of inertia factor to eqn (5) is given in eqn (7). It improves the performance and search precision of the particle.

\[
V_i = \theta \times V_i + \varphi_1 \times rand \times (pBest_i - x_i) + \varphi_2 \times rand \times (gBest - x_i) \quad (7)
\]

Where the inertia factor and rand is denotes the random number.

Algorithm:

Steps:

- Initialize each particle.
- Calculate fitness value for each particle.
• Compare fitness values between particles and the best fitness value pBest.
• The above steps are repeated for all particles.
• The particle with best fitness value among pBest values is set as gBest.
• Calculate the velocity for each particle and also update the particle position.
• Continue the above steps until the optimized results are retrieved.

4. EXPERIMENTAL RESULTS
To show the effectiveness of the proposed system, some experiments performed on simplicity database. In our experiments, the database uses the image from different categories like natural scene, beach etc. The database is partitioned into ten categories: village, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountains and, food, etc., and each category contain 100 images. The image from the database is taken as query image. Figure2 depicts the query image.

![Figure 2. Query image](image)

Color feature are extracted using YUV method. Y component, U component and V component is extracted. Mean and standard deviation is computed. Query image is given as input. Color image is converted into RGB and then into YUV using rgb2yuv function in mat lab. Figure 3, 4 and 5 depict Y component, U component and V component respectively.
Figure 3. Y Component of query image

Figure 4. U Component of query image

Figure 5. V Component of query image
Texture feature is extracted using Grey Level Co-occurrence Matrix method. Texture features such as auto-correlation, entropy, energy, homogeneity, contrast, etc. are extracted from the query image and the images in the database. Edge feature is extracted using Edge Histogram Descriptor method. Texture feature extraction is shown in Figure 6.

Edge Feature extraction is shown in figure 7. In edge feature extraction image space is divided into 16 (4x4) non overlapping sub images. For each sub images a histogram with five edge bins are generated. Totally 80 bins for the entire image is generated.

In this image retrieval method Euclidean Distance method is used which computes the similarity between the query image and database images according to the before mentioned low level visual features. This method retrieves and presents a sequence of images ranked in decreasing order of similarity. As a result the user is able to find relevant images by getting top
ranked images first. Figure 8 and 9 shows the query image and retrieved result of a feature based image retrieval process. Figure 10 and 11 shows the query and retrieved result for a PSO optimization method.

Figure 8. query image for feature based image retrieval

Figure 9. Feature based image retrieval result

Figure 10. query image for PSO based image retrieval
The retrieved images are again optimized using PSO which uses the fitness value of each particle. The images with best fitness value are retrieved as a result of optimization.

5. PERFORMANCE EVALUATION
The performance of the proposed system is demonstrated using MATLAB 2010b platform. The retrieval efficiency, namely precision and recall were calculated using natural color images from corel image database. Precision is defined as the proportion of the number of relevant images retrieved to the total number of retrieved images and Recall is defined as the number of retrieved relevant images over the total number of relevant images available in the database. Standard formula for precision and recall is given in Eqns (8) and (9).

\[
\text{Precision} = \frac{\text{Number of relevant images retrieved}}{\text{Number of retrieved images}} \quad (8)
\]

\[
\text{Recall} = \frac{\text{Number of relevant images retrieved}}{\text{Total number of relevant images in the database}} \quad (9)
\]

The results obtained from the various categories have been tabulated in table 1. Figure 12 and 13 shows the average retrieval precision and recall of the proposed system.
Table 1. Precision and recall values for different categories of images

<table>
<thead>
<tr>
<th>Query image</th>
<th>No. Of relevant images retrieved</th>
<th>Relevant images in database</th>
<th>No. Of retrieved images</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiger</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>100%</td>
<td>80%</td>
</tr>
<tr>
<td>Bird</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>80%</td>
<td>80%</td>
</tr>
<tr>
<td>Flower</td>
<td>8</td>
<td>11</td>
<td>9</td>
<td>88.8%</td>
<td>72.2%</td>
</tr>
<tr>
<td>Road</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>70%</td>
<td>70%</td>
</tr>
</tbody>
</table>

Figure 12. Retrieval Average precision
5. CONCLUSION
Thus in proposed method color feature is extracted using YUV method. Texture feature is extracted using Grey level co-occurrence method. These features are extracted both from the query image and the image from the database. Edge feature is extracted using Edge Histogram Method. Similarity is computed between the query image and the images from the database. Results are optimized using Particle Swarm Optimization.

REFERENCES


This paper may be cited as: