

Image Retrieval Based On DWT and Clustering Algorithm

Asmita Shirsath, M.J.Chouhan, N.J.Uke

Abstract— Content-based image retrieval has been an interesting subject of many researchers in recent years, and image classification and retrieval is also an important issue in pattern recognition and artificial intelligence. For large and high-dimension image databases, it is obvious that traditional CBIR retrieval method has been unable to meet efficiency. In this paper, in order to improve the retrieval efficiency and retrieval result, we first preprocess the image database using DWT and K-means algorithm and then using hierarchical clustering algorithm we retrieve the images. Currently we have formed initial clusters results by using preprocessing algorithm i.e by using DWT and K-means algorithm.

Index Terms—Color Histogram, DWT, Haar Discrete Wavelet Transforms, Hierarchical Clustering Algorithm, K-Means Algorithm.

1 INTRODUCTION

There has been an intense work carried out in development of image retrieval technique based on image content in recent years [1]. At present, clustering has achieved great success in many fields including pattern recognition, system modeling, image processing, data mining and etc. The purpose of clustering is grouping a set of physical or abstract objects into classes of similar objects based on certain rules. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Clearly, the thought of clustering is very useful reference to content-based image retrieval [2].

In order to get faster and more accurate retrieval result from large-scale image database, in this paper, a improved hierarchical clustering algorithm is proposed for hierarchical indexing to image database based on the classical hierarchical clustering algorithm [2], K-means algorithm and the DWT method [3]. The rest of the paper is organized as follows: in section 2, overview of the system is explained, in section 3, a image clustering method is explained. The section 4 describes hierarchical clustering algorithm and section 5, presents image retrieval.

2 OVERVIEW OF THE SYSTEM

Through clustering algorithm we classify large image database according to a similarity principle, similar images can be gathered together and thus the scope of image searching can be

greatly reduced, and so the target image can be found quickly and accurately[2]. The CBIR systems has widely been used to describe the process of retrieving desired images from a large collection of database, on the basis of syntactical image features (color, texture and shape).

In every simple and complex CBIR systems, the process involves extraction of a features for every image based on its pixel values and to define a rule for comparing images. One advantage of features over the original pixel values is the significant compression of image representation [1]. The color histogram for color feature and wavelet representation for texture and location information of an image reduces the processing time for retrieval of an image with more promising representatives. The proposed system mainly consists of three steps:

- Step 1: To create initial cluster results by Pre-processing algorithm.
- Step 2: To establish hierarchical indexing based on the results of step one.
- Step 3: Image retrieval.

3 IMAGE CLUSTERING METHOD

3.1 Preprocessing Algorithm

Pre-processing Algorithm consists of three steps:

3.1.1 Feature Extraction Method

3.1.2 Image Clustering

3.1.3 Pre-processing Algorithm Steps

3.1.1 Feature Extraction Method

3.1.1.1 Color Feature

The color feature is one of the most widely used visual features in image retrieval. Images characterized by color features have many advantages such as Robustness, Effective

- Asmita Shirsath is currently pursuing masters degree program in Information Technology engineering in Pune University, India, PH-7588077082. E-mail: shirsathasmita@gmail.com
- Mrinalini J. Chouhan is currently Assistant Professor in SCOE, Pune in Pune University, India, PH- 9881913374. E-mail: mrinal_rajput@yahoo.com
- Nilesh J Uke is currently Assistant Professor in SCOE, Pune in Pune University, India, PH-98500 35805. E-mail: nilesh.uke@gmail.com

tiveness, Implementation simplicity, Computational simplicity, Low storage requirements [6]. The color feature has widely been used in CBIR systems, because of its easy and fast computation. Typically, the color of an image is represented through some color model. There exist various color model to describe color information.

A color model is specified in terms of 3-D coordinate system and a subspace within that system where each color is represented by a single point. The more commonly used color models are *RGB* (red, green, blue), *HSV* (hue, saturation, value) and *Y,Cb,Cr* (luminance and chrominance). Thus the color content is characterized by 3-channels from some color model. One representation of color content of the image is by using color histogram. Statistically, it denotes the joint probability of the intensities of the three color channels. Color is perceived by humans as a combination of three color stimuli: Red, Green, and Blue which forms a color space. This model has both a physiological foundation and a hardware related one. *RGB* colors are called primary colors and are additive. By varying their combinations, other colors can be obtained. The representation of the *HSV* space is derived from the *RGB* space cube, with the main diagonal of the *RGB* model, as the vertical axis in *HSV*. As saturation varies from 0.0 to 1.0, the colors vary from unsaturated (gray) to saturated (no white component). Hue ranges from 0 to 360 degrees, with variation beginning with red, going through yellow, green, cyan, blue and magenta and back to red. These color spaces are intuitively corresponding to the *RGB* model from which they can be derived through linear or non-linear transformations [6].

Color descriptors of images can be global or local and consist of a number of histogram descriptors and color descriptors represented by color moments, color coherence vectors or color correlograms [6]. Color is also an intuitive feature and plays an important role in image matching. The extraction of color features from digital images depends on an understanding of the theory of color and the representation of color in digital images. The color histogram is one of the most commonly used color feature representation in image retrieval [3].

3.1.1.2 Color Space Selection and Color Quantization

The color of an image is represented, through any of the popular color spaces like *RGB*, *XYZ*, *YIQ*, $L^*a^*b^*$, $U^*V^*W^*$, *YUV* and *HSV*. It has been reported that the *HSV* color space gives the best color histogram feature, among the different color spaces. In *HSV* color space the color is presented in terms of three components: Hue (*H*), Saturation (*S*) and Value (*V*) and the *HSV* color space is based on cylinder coordinates. Color quantization is a process that optimizes the use of distinct colors in an image without affecting the visual properties of an image. For a true color image, the distinct number of colors is up to $2^{24} = 16777216$ and the direct extraction of color feature from the true color will lead to a large computation. In order to re-

duce the computation, the color quantization can be used to represent the image, without a significant reduction in image quality, thereby reducing the storage space and enhancing the process speed. The effect of color quantization on the performance of image retrieval has been reported by many authors [3].

3.1.1.3 Color Histogram

Color histograms are used to represent image color information in various CBIR systems. Color histogram describes the distribution of colors within a whole or within a interest region of image. The histogram is invariant to rotation, translation and scaling of an object but the histogram does not contain semantic information, and two images with similar color histograms can possess different contents [6]. A color histogram is a type of bar graph, in which each bar represents a particular color of the color space being used. The bars in a color histogram are referred to as bins and they represent the x-axis. The number of bins will be totally dependent on the number of colors in an image. Color histogram y-axis denotes the numbers of pixels of each bin. There are two types of color histograms, Global color histograms (*GCHs*) and Local color histograms (*LCHs*). A *GCH* represents one whole image with a single color histogram while the *LCH* divides an image into fixed blocks and takes the color histogram of each of those blocks [5].

The color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set. The color histogram is easy to compute and effective in characterizing both the global and local distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle. Since any pixel in the image can be described by three components in a certain color space (for instance, red, green, and blue components in *RGB* space, or hue, saturation, and value in *HSV* space), a *histogram*, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains, the more discrimination power it has. However, a histogram with a large number of bins will not only increase the computational cost, but will also be inappropriate for building efficient indexes for image databases [7]. A color histogram represents the distribution of colors in an image, through a set of bins, where each histogram bin corresponds to a color in the quantized color space. A color histogram for a given image is represented by a vector: $H = \{H[0], H[1], H[2], \dots, H[i], \dots, H[n]\}$ Where *i* is the color bin in the color histogram and $H[i]$ represents the number of pixels of color *i* in the image, and *n* is the total number of bins used in color histogram. Typically, each pixel in an image will be assigned to a bin of a color histogram. Accordingly in the color histogram of an image, the value of each bin gives the number of pixels that has the same

corresponding color. In order to compare images of different sizes, color histograms should be normalized. The normalized color histogram H' is given as:

$$H' = \{H'[0], H'[1], H'[2], \dots, H'[i], \dots, H'[n]\}$$

Where $H'[i] = \frac{H[i]}{p}$, p is the total number of pixels of an image [3].

3.1.1.4 Texture Feature

Like color, the texture is a powerful low-level feature for image search and retrieval applications. Much work has been done on texture analysis, classification, and segmentation for the last four decades, still there is a lot of potential for the research. The common known texture descriptors are Wavelet Transform, Gabor-filter, co-occurrence matrices and Tamura features. We have used Wavelet Transform, which decomposes an image into orthogonal components, because of its better localization and computationally inexpensive properties [3].

Wavelet transform is a relatively recent signal processing tool that has been successfully used in a number of areas. Wavelet becomes more popular tool for image compression. Wavelets provide multiresolution capability, good energy compaction and adaptability to human visual system characteristics. The conventional Discrete wavelet transform (DWT) may be regarded as equivalent to filtering the input signal with a bank of filters whose impulse response are all approximately given by scaled version of a mother wavelet. The images in a database are likely to be stored in a compressed form. Superior indexing performance can therefore be obtained if the properties of the coding scheme are exploited in the indexing technique. Recently, discrete wavelet transform (DWT) has become popular in image coding applications. Wavelet transform represents a function as a superposition of a family of basic functions called wavelets. A set of basic functions can be generated by translating and dilating the mother wavelet corresponding to a particular basis. The signal is passed through a lowpass (LPF) and a highpass filter (HPF) and the outputs of the filter are decimated by two. Thus, wavelet transform extracts information from the signal at different scales [4].

3.1.1.5 Haar Discrete Wavelet Transforms

Discrete wavelet transformation (DWT) is used to transform an image from spatial domain into frequency domain. The wavelet transform represents a function as a superposition of a family of basic functions called wavelets. Wavelet transforms extract information from signal at different scales by passing the signal through low pass and high pass filters. Wavelets provide multiresolution capability and good energy compaction. Wavelets are robust with respect to color intensity shifts and can capture both texture and shape information efficiently. The wavelet transforms can be computed linearly with time and thus allowing for very fast algorithms. DWT decomposes a signal into a set of Basis Functions and Wavelet Functions. The wavelet transform computation of a two-dimensional image is

also a multi-resolution approach, which applies recursive filtering and sub-sampling. At each level (scale), the image is decomposed into four frequency sub-bands, LL, LH, HL, and HH where L denotes low frequency and H denotes high frequency.

In this paper, we have used Haar wavelets to compute feature signatures, because they are the fastest to compute and also have been found to perform well in practice. Haar functions have been used from 1910 when they were introduced by the Hungarian mathematician Alfred Haar. The Haar transform is one of the earliest examples of what is known now as a compact, dyadic, orthonormal wavelet transform. The Haar function, being an odd rectangular pulse pair, is the simplest and oldest orthonormal wavelet with compact support [8]. Haar wavelets enable us to speed up the wavelet computation phase for thousands of sliding windows of varying sizes in an image [3].

3.1.2 Image Clustering

The k -means algorithm is an algorithm to cluster n objects based on attributes into k partitions, $k < n$. The algorithm starts by partitioning the input points into k initial sets, either at random or using some heuristic data. It then calculates the mean point, or centroid, of each set. It constructs a new partition by associating each point with the closest centroid. Then the centroids are recalculated for the new clusters, and algorithm repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed) [1]. The k -means algorithm is popular because it is easy to implement, and its time complexity is $O(n)$, where n is the number of patterns [10]. K -means is a numerical, unsupervised, non-deterministic, iterative method. It is simple and very fast, so in many practical applications, the method is proved to be a very effective way that can produce good clustering results [11]. The steps to be followed is as follows:

- Step 1: Choose the number of clusters, k .
- Step 2: Randomly generate k clusters and determine the cluster centers, or directly generate k random points as cluster centers.
- Step 3: Assign each point to the nearest cluster center.
- Step 4: Recompute the new cluster centers.
- Step 5: Repeat the two previous steps until some convergence criterion is met.

3.1.3 Pre-processing Algorithm Steps

In Pre-processing algorithm, we use Color Histogram for color feature, discrete wavelet transformation for texture feature and K -means algorithm to obtain group of cluster of feature vectors. The steps to be followed are as follows:

- Step 1. Extract the Red, Green, and Blue Components from an image.
- Step 2. Decompose each Red, Green, Blue Component using

Haar Wavelet transformation at 1st level to get approximate coefficient and vertical, horizontal and diagonal detail coefficients.

Step3. Combine approximate coefficient of Red, Green, and Blue Component.

Step4. Similarly combine the horizontal and vertical coefficients of Red, Green, and Blue Component.

Step5. Assign the weights 0.003 to approximate coefficients, 0.001 to horizontal and 0.001 to vertical coefficients (experimentally observed values).

Step6. Convert the approximate, horizontal and vertical coefficients into HSV plane.

Step7. Color quantization is carried out using color histogram by assigning 8 level each to hue, saturation and value to give a quantized HSV space with 8x8x8=512 histogram bins.

Step8. The normalized histogram is obtained by dividing with the total number of pixels.

Step9. Repeat step1 to step8 on an image in the database.

Step 10: Apply K-means algorithm to obtain group of cluster of feature vectors [3].

4 HIERARCHICAL CLUSTERING ALGORITHM

A hierarchical clustering is a nested sequence of partitions. This method works on both bottom-up and top-down approaches. Based on the approach hierarchical clustering is further subdivided into agglomerative and divisive. The agglomerative hierarchical technique follows bottom up approach whereas divisive follows top-down approaches. Hierarchical clustering use different metrics which measures the distance between 2 tuples and the linkage criteria, which specifies the dissimilarity in the sets as a function of the pairwise distances of observations in that sets. The linkage criteria could be of 3 types' single linkage, average linkage and complete linkage [9].

4.1 Agglomerative Algorithm

For n samples, agglomerative algorithms begin with n clusters and each cluster contains a single sample or a point. Then two clusters will merge so that the similarity between them is the closest until the number of clusters becomes 1 or as specified by the user.

1. Start with n clusters, and a single sample indicates one cluster.
2. Find the most similar clusters C_i and C_j then merge them into one cluster.
3. Repeat step 2 until the number of cluster becomes one or as specified by the user.

The distances between each pair of clusters are computed to choose two clusters that have more opportunity to merge. There are several ways to calculate the distances between the clusters C_i and C_j [9].

Suppose S_i as the class vector of class i , d_{ij} , as the distance between class i and class j . In each iteration of Hierar-

chical clustering algorithm, first the minimum distance d_{ij} , will be found, and the two classes i and j will be merged into a new class m . Then we re-calculate the distances between new class m and the other classes [2].

Step 1: For each $(i, j) \in C^2$, calculate the distances between classes d_{ij} , $j=d(S_i, S_j)$;

Step 2: $(i^*, j^*) = \arg(\min(d_{ij}), (i, j) \in C^2)$

Step 3: Set $C_m \leftarrow C_i \cup C_j$

Step 4: $C \leftarrow \{C - \{i^* - j^*\} \cup \{m\}$

Step 5: Set $S_m = \frac{1}{\|C_m\|} \sum_{v \in C_m} x_v$

Step 6: Repeat (1)-(5) until it reaches the termination condition.

5 IMAGE RETRIEVAL

Image retrieval is generally conducted by searching the most similar images from databases to the query image. In this project, hierarchical index will be firstly established by Hierarchical Clustering Algorithm for image database, and then retrieval will be done based on the indexing [2]. The retrieval mainly consists of two steps:

- (1) to calculate distances between query image and class centres to obtain the most similar sub-database.
- (2) to calculate distances between query image and images in the sub-database to return N most similar images.

6 CONCLUSION

In this paper, in order to improve the retrieval efficiency, we make use of DWT method and k-Means algorithm for pre-processing the Image Database i.e we first extract the image features by using Color Histogram and Discrete Wavelet Transform method and then by applying K-means algorithm we form initial clusters and then using hierarchical index structure based on clustering we will be retrieving the images.

REFERENCES

- [1] Sanjay N. Talbar, And Satishkumar L. Varma, "Imatch: Image

- Matching And Retrieval For Digital Image Libraries," Second International Conference On Emerging Trends In Engineering And Technology, ICETET-09.
- [2] Cai-Yun Zhao, Bian-Xia Shi , Ming-Xin Zhang, And Zhao-Wei Shang," Image Retrieval Based On Improved Hierarchical Clustering Algorithm," Proceedings Of The 2010 International Conference On Wavelet Analysis And Pattern Recognition, Qingdao, 11-14 July 2010.
- [3] Manimala Singha* And K.Hemachandran," Content Based Image Retrieval Using Color And Texture," Signal & Image Processing: An International Journal (SIPIJ) Vol.3, No.1, February 2012.
- [4] N S T Sai, R C Patil," Image Retrieval Using DWT With Row And Column Pixel Distributions Of BMP Image," N S T Sai Et Al. / (IJCSSE) International Journal On Computer Science And Engineering Vol. 02, No. 08, 2010, 2559-2566.
- [5] Rahul Mehta, Nishchol Mishra, Sanjeev Sharma," Color - Texture Based Image Retrieval System", International Journal Of Computer Applications (0975 – 8887) Volume 24– No.5, June 2011.
- [6] Ryszard S. Chora'," Image Feature Extraction Techniques And Their Applications For CBIR And Biometrics Systems," International Journal Of Biology And Biomedical Engineering, Issue 1, Vol. 1, 2007.
- [7] Dr. Fuhui Long, Dr. Hongjiang Zhang And Prof. David Dagan Feng," FUNDAMENTALS OF CONTENT-BASED IMAGE RETRIEVAL,"
- [8] Radomir S. Stankovi_C A, Bogdan J. Falkowski B,*," The Haar Wavelet Transform: Its Status And Achievements," Computers And Electrical Engineering 29 (2003) 25–44.
- [9] Dr.N.Rajalingam, and K.Ranjini," Hierarchical Clustering Algorithm - A Comparative Study," *International Journal Of Computer Applications (0975 – 8887) Volume 19– No.3, April 2011.*
- [10] A.K. Jain, M.N. Murty, And P.J. Flynn," Data Clustering: A Review," ACM Computing Surveys, Vol. 31, No. 3, September 1999.
- [11] Shi Na, Liu Xumin, Guan Yong," Research On K-Means Clustering Algorithm," *IEEE Trans*, Third International Symposium On Intelligent Information Technology And Security Informatics, 2010,978-0-7695-4020-7/10.