Image Retrieval from Compressed Images

*Yuk Ying CHUNG, Man To WONG
*Basser Department of Computer Science,
The University of Sydney,
NSW 2006 , Australia

Abstract

Digital images are useful media for storing spatial, spectral and temporal components of information. Many large image databases have been generated for different kinds of purposes. These images are often stored in compressed format such as JPEG. This paper examines the algorithms of direct extraction of low level features from compressed images. Also a cluster based indexing technique will be used in this retrieval system. The query image will only compare with small subset group instead of all the images in the large database. The content-based retrieval system is based on DCT coefficient energy histogram and clustering algorithms. It can offer a fast response time with high retrieval accuracy.

Keywords: Image compression, DCT domain, clustering, Image retrieval

1. Introduction

Content-based image and video retrieval has become an important research area recently. The growing popularity of the internet, the introduction of new consumer products for image and video creation have resulted in great demand of multimedia storage and retrieval systems. Content-based retrieval systems for images based on various image features such as colour, shape and texture have been reported [1-3]. However, most images and video are stored in compressed format in order to solve the storage problem. To search and locate the database in compressed domain is still a difficult problem.

This work is based on the discrete cosine transform (DCT) which is the heart of current image and video compression standards such as JPEG[4], MPEG and H.261. The features will be directly extracted from compressed data and these features are the characteristics of image and video contents for content-based retrieval.

Most image retrieval systems extract specific features from a query image and compare these features with the corresponding pre-computed features of all the images in the database. The search time will increase linearly with the size of database. The cumulative time needed to compare the query image with all database images will become very long even the time required to compare two images is very short. In order to solve this problem, we can create an indexing scheme by grouping the images based on their content values so that only the relevant set of cluster images will be compared. The clustering algorithms must retain the original retrieval accuracy. Indexing techniques such as R* and K-D-B trees [5,6] have been proposed for fast range search but they cannot solve the problem of large dimensionality of vectors greater than ten. Therefore, we choose clustering techniques in our retrieval system to overcome the feature dimension problem.

In this paper, section 2 gives the introduction of DCT/ JPEG compression. Section 3 shows how we can extract the DCT coefficients for feature extraction and retrieve the required image from an image database. The algorithm will retrieve the required image in DCT transform domain and it does not require the decompression of database image. Section 4 presents an introduction to image clustering. Section 5 and 6 present our preliminary result, future work and conclusion.

2. DCT/JPEG compression

We first give an introduction of DCT/JPEG[4] compression. The block diagrams of JPEG compression and decompression are shown in Figure 1. The original image is sub-divided into 8x8 blocks, each of which is independently processed. Each block is transformed into frequency domain by using forward DCT, resulting in an 8x8 block of DCT coefficients. These coefficients are then quantized by integer division by constants. The quantizing constant for each DCT coefficient is chosen to produce minimal visual artifacts, while maximally reducing the representational entropy of the coefficients. The quantized coefficients are then entropy coded into a compressed data stream. The reduced entropy of the quantized coefficients are reflected in the higher compression ratio of the data. For decompression of JPEG images, the reverse process is performed to construct the image from compressed stream.

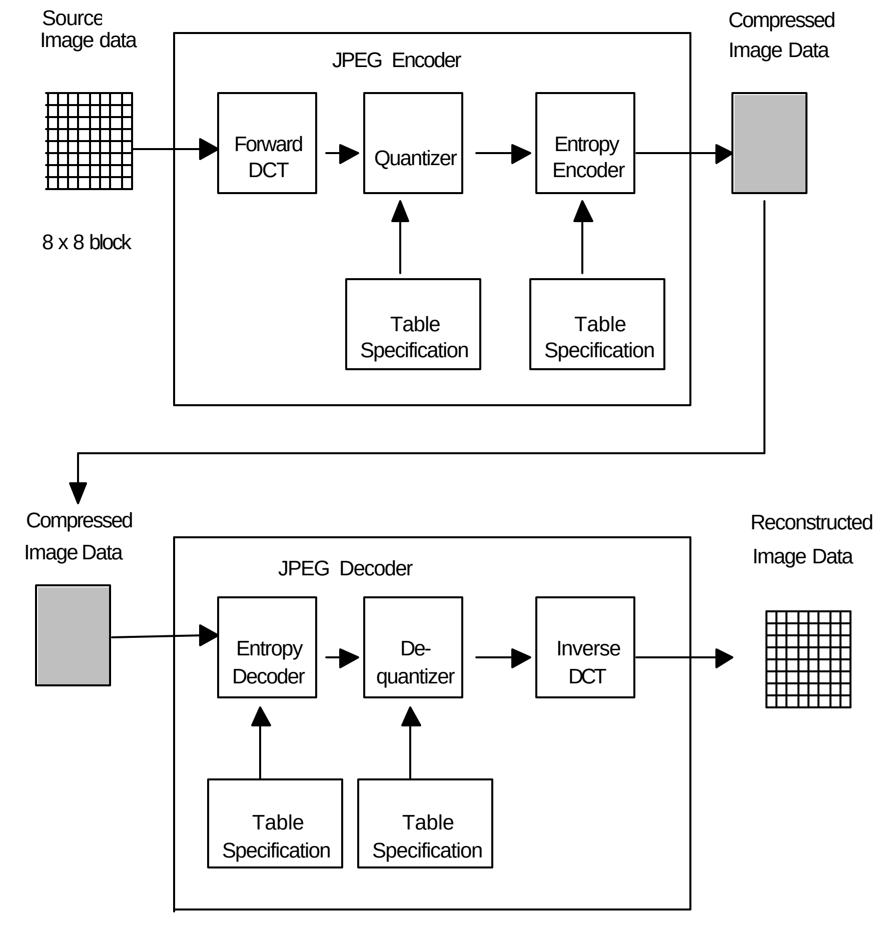

Figure 1. JPEG compression and decompression algorithms

Discrete Cosine Transform (DCT) is the heart of the JPEG/DCT based compression techniques. The two dimensional DCT is defined as follows:

The 2-D forward DCT and inverse DCT can be constructed from products of the terms of a horizontal 1-D DCT (using *u* and *x*) and a vertical 1-D DCT (using *v* and *y*, where *v* represents vertical frequencies and *y* represents vertical displacements).

Forward DCT:

$$S(v,u) = \frac{C(v)}{2}\frac{C(u)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} s(y,x)\cos(\frac{(2x+1)up}{16})*\cos(\frac{(2y+1)vp}{16})$$

……….[eq 4.4]

Inverse DCT:

$$s(y,x) = \sum_{v=0}^{7}\frac{C(v)}{2}\sum_{u=0}^{7}\frac{C(u)}{2} S(v,u)\cos(\frac{(2x+1)up}{16})*\cos(\frac{(2y+1)vp}{16})$$

……….[eq 4.5]

where:

$C(u) = \frac{1}{\sqrt{2}}$	for u = 0
$C(u) = 1$	for u > 0
$C(v) = \frac{1}{\sqrt{2}}$	for v = 0
$C(v) = 1$	for v > 0
$s(y,x) = 2D$	sample value
$S(v,u) = 2D$	DCT coefficient

3. Image retrieval using DCT coefficients

Most images in database are stored in JPEG[4] format. It can reduce much computational complexity if we process the image retrieval in compressed domain and avoid the full data decompression process.

In this work, we use various sets of DCT coefficients extracted from JPEG compressed images. The DCT coefficient values can be regarded as the relative amount of the 2D spatial frequencies contained in the 8x8 block input data. The coefficient with zero frequency is called "DC coefficient" and the remaining 63 coefficients are called "AC coefficients". Figure 2 shows the zig zag sequence of 8x8 DCT block with DC and 63 AC coefficients. The forward DCT process can concentrate most of the signal in the lower spatial frequency. For 8x8 block from compressed image, we can only consider the low spatial frequency coefficients to construct the energy histogram. The following S0, S1 and S2 are three sets of the DCT coefficients that we have used for testing.

S0 = [DC];
S1 = [DC, c(0,1), c(1,0), c(1,1)]
S2 = [DC, c(0,1), c(1,0), c(1,1), c(2,0), c(2,1), c(1,2), c(2,2)]

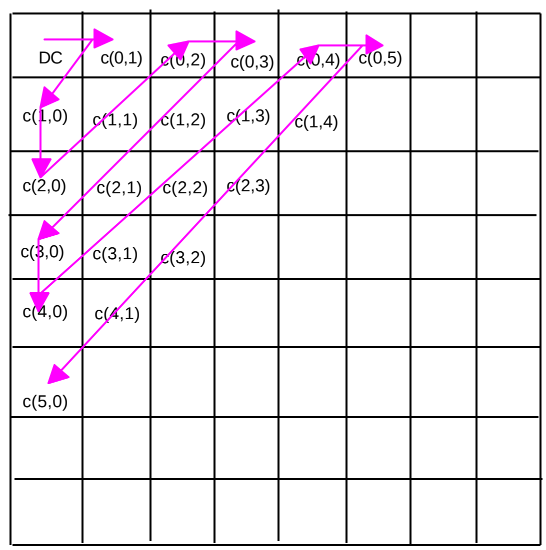

Figure 2. The zig zag sequence of DCT coefficients

The DCT energy histogram is constructed by counting the number of times of an energy level that occurs in 8 x 8 DCT blocks. After constructing the DCT energy histogram, the histogram intersection method will be used to perform matching. In matching process, the reference image DCT energy histogram will compare with different groups in the databases. The matched images will have the higher matching values. Histogram intersection is an efficient way to perform matching. The computation complexity is low and can be implemented by most of the computers. The matching algorithm is as follow:

Given a pair of histograms which consist of a reference image (I) and a database image (J), each contains N DCT energy levels,

$$match\_value = \sum_{n=1}^{N} \min(I_n, J_n)$$

In order to normalize the matching result between 0 and 1, the measure is finally divided by the total number of coefficients used in the reference image. The histogram intersection technique proposed by [7] is defined as follows:

$$H(I,J) = \frac{\sum_{n=1}^{N} \min(I_n, J_n)}{\sum_{n=1}^{N} I_n}$$

4. Image Clustering

Many retrieval systems calculate the features similarity between the query image and all images in the database and rank the images by sorting their similarities. The problem of this full search approach is very time consuming for large scale database.

The retrieval time of this full search is the sum of the time to calculate similarity time $\mathbf{T}_{sim}$ and the time to sort the images in the database according to their similarity $\mathbf{T}_{sort}$. Time for the full search $\mathbf{T}_{full}$:

$$\mathbf{T}_{full} = \mathbf{T}_{sim} + \mathbf{T}_{sort}$$

$$\mathbf{T}_{full} = \mathbf{n}\mathbf{T}_{sim(1)} + O(\mathbf{n}\log \mathbf{n})$$

Where
n : number of images in the database

$\mathbf{T}_{sim}$: total time to calculate the similarity

$\mathbf{T}_{sort}$: total time to rank n images

$\mathbf{T}_{sim(1)}$: time to calculate similarity between two images

$O(\mathbf{n}\log \mathbf{n})$: time to sort n elements

If the images in the database are clustered, the retrieval time is the sum of:

1) The time to calculate the similarity between the query and the cluster centers.
2) The time to calculate the similarity between the query and the images in the nearest clusters.
3) The time to rank the images in step 2.

Time for cluster search $\mathbf{T}_{cluster}$:

$$\mathbf{T}_{cluster} = k\,\mathbf{T}_{sim(1)} + l\,\mathbf{T}_{sim(1)} + O(l \log l)$$

Where *k*: number of clusters
l: number of images in the clusters nearest to the query

Since $k \ll \mathbf{n}$ and $l \ll \mathbf{n}$, therefore the clustering search time $\mathbf{T}_{cluster}$ should be much smaller than the full search time $\mathbf{T}_{full}$ (i.e. $\mathbf{T}_{cluster} \ll \mathbf{T}_{full}$).

In our retrieval system, we choose k-means clustering [8]. The k-means clustering algorithm is as follow:

Step 1: Begin with k clusters, each consisting of one of the first k samples. For each of the remaining (n-k) samples, find the centroid nearest it. Put the sample in the cluster identified with this nearest centroid. After each sample is assigned, recompute the centroid of the altered cluster.

Step 2: Go through the data a second time. For each sample, find the centroid nearest it. Put the sample in the cluster identified with this nearest centroid. Do not recompute any centroid in this step.

Step 3: If no samples change cluster in step 2, stop.

Step 4: Go to Step 2.

5. Experimental Results:

RGB colour space

Digital images are normally represented in RGB space. The original image lenna is shown in figure 3. The RGB colour distribution of the colour image lenna is shown in figure 4.

Figure 3 Lenna (512x512 pixels)

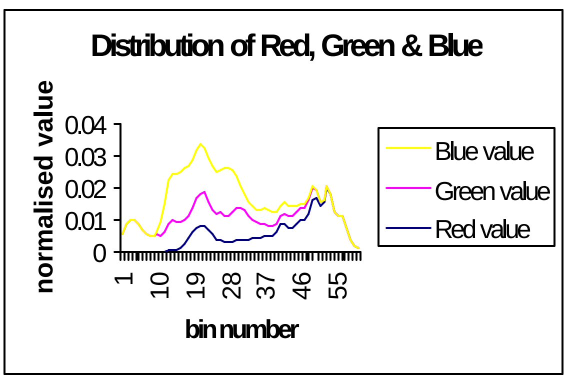

Figure 4 RGB colour distrubution of Lenna in figure 3

YUV colour space

Y represents the luminance of a color, U and V represent the chromaticity of colour. The advantage of YUV space is that the luminance is separated from chrominance which is very useful in compression and image processing applications. The YUV distribution of the colour image lenna is shown in figure 5.

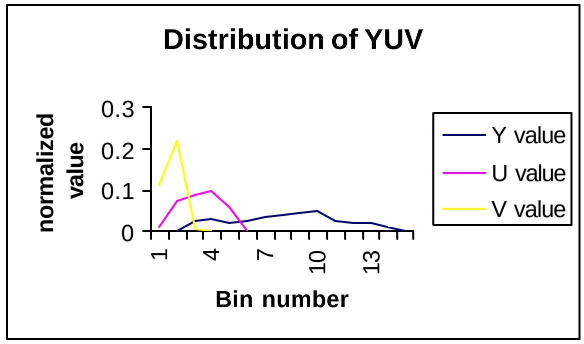

Figure 5. YUV distribution of Lenna in figure 3

DCT energy histogram

S0, S1 and S2 have been explained in section 3. The Y-Cr-Cb of S0,S1 and S2 of the colour image lenna are shown in figure 6, 7 and 8 respectively.

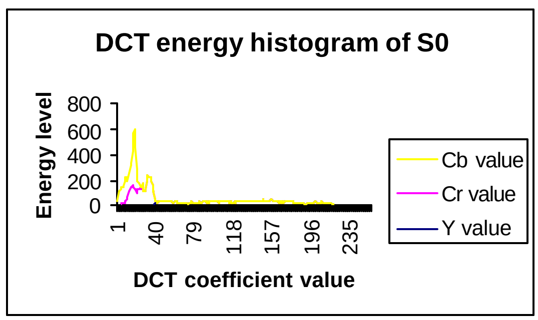

Figure 6 The Y-Cr-Cb distribution of S0

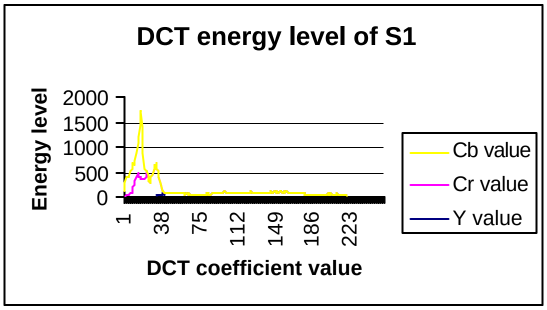

Figure 7 The Y-Cr-Cb distribution of S1

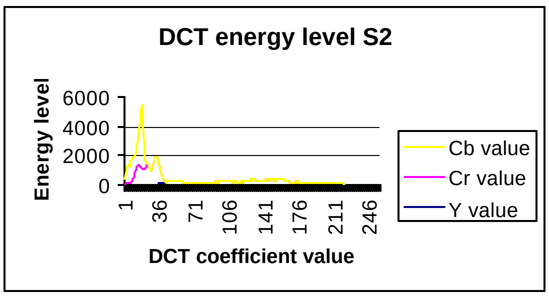

Figure 8 The Y-Cr-Cb distribution of S2

In our testing result, different colour distributions based on different colour spaces such as RGB and YUV have been studied. In our retrieval system, DCT energy values are used as the contents. These values are used by the clustering system to divide the images into different groups. Hence, the query image only compares with small subset group instead of all the images in the large database.

5 . Conclusion

In this paper, a new content based retrieval system based on DCT coefficient energy histogram and clustering algorithm has been proposed. This new retrieval system can offer a fast response time with high retrieval accuracy. This work only presents some results of our preliminary study. We will further test our system with large image database. Parallel algorithms will be further developed and run on super computer provided by VPAC organization.

6. Acknowledgements

This research is supported by the VPAC (Victorian Partnership for Advanced Computing Expertise Program) Grants Scheme.

7. References

[1] W.Niblack, et.al.,"The QBIC Project: querying images by content using color, texture and shape," In Storage and Retrieval for Image and Video Databases, Vol. 1908, SPIE Proceedings, Feb, 1993.

[2] J.Smith and S.F. Chang, "A Fully Automated Content-based Image Query System", Proc. of ACM Intl. Multimedia Conference, Nov. 1996.

[3] H.Zhang, Y.Gong, C.Y.Low and S.W. Smoliar, "Image retrieval based on color features: an evaluation study", Proc. of SPIE vol 2606, pp.212-220, 1995.

[4] Gregory K. Wallace, "The JPEG Still Picture Compression Standard.

[5] N.Beckman, et. Al., "The R*tree: An efficient and robust access method for points and rectangles", In Proc. Of ACM SIGMOD, Atlantic City, May 1990.

[6] J.T. Robinson, "The K-D-B tree: a search structure for large multidimensional dynamic indexes", In Proc of ACM SIGMOD, Ann Arbor, April 1981.

[7] M.J. Swain and D.H. Ballard, "Color Indexing", International Journal of Computer Vision, Vol 7, No. 1, 1991, pp 11-32.

[8] E.Gose, R.Johnsonbaugh, S.Jost, "Pattern Recognition and Image Analysis" Prentice Hall, 1996.