Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.379 – 386

RESEARCH ARTICLE

Baseline JPEG Image Compression with K-Means Clustering Based Algorithm

Karri Tata Rao[1], Ch. Ramesh[2]
[1]Final M.Tech student, Dept .of CSE, AITAM, Tekkali, A.P, India
karritatarao934@gmail.com
[2]Professor, Dept .of CSE, AITAM, Tekkali, A.P, India
chappa_ramesh01@yahoo.co.in

Abstract: Applications of Digital Image Communication have increased drastically in our day to day activities. Baseline Joint Photographic Experts Group (JPEG) is the common image compression method for reducing bandwidth. Baseline JPEG Compression system comprises of a DCT unit followed by a normalizer and encoder. At the receiving end, image is reconstructed by inverse DCT. In this paper, we develop a JPEG Compression algorithm that joins K-Means clustering technique and DCT to further reduce the bandwidth requirements. Experiments are carried out with many standard images. Our algorithm proved to be giving approximately same Peak Signal-to-Noise Ratio (PSNR) as that of Base line conventional JPEG algorithm.

Keywords:- DCT, IDCT, PSNR, Baseline JPEG, K-Means clustering

I. INTRODUCTION

In the recent years, data intensive multimedia-based web applications have increased by many folds. Many remote applications such as remote monitoring, surveillance, automatic navigation systems often involves communication of captured images for further processing. In these applications, it is inevitable to conserve the available bandwidth in order to reduce consumer side bills. In order to reduce the bandwidth consumption, communicating compressed images is a classical solution in communications and related fields. Baseline JPEG compression is the standard image compression for still images.

Image based recognition systems that are used in production industries such as IC manufacturing, fruit processing systems, automatic welding, etc., will be using original images. However, there are some recent applications such as remote monitoring, surveillance, remote surgery, etc., may involve communication of captured images to a server for processing, recognition and control. For example, [7] used compressed face images in JPEG format for the development of their face recognition system that extracts edge based features from the JPEG images and uses with a neural network. Certainly, performance of recognition system based on original images will be giving better results than compressed images as compressed images loses some details which are otherwise useful for recognition. It is reported elsewhere JPEG compression induces some artifacts such as noise around edges, blurring, a smeared appearance, color distortion, and/or checkerboard-like blocking in busy regions. However, it consumes very little bandwidth. Thus, scientific community may be interested in studying about recognition system performance when it is designed to used compressed images rather than original images.

The paper is organized as follows. In section II, a overview of the standard Baseline JPEG is provided. The proposed K-means clustering based Baseline JPEG compression system is described in section III. Experimental results are presented in section IV. Finally, conclusions are reported in section V.

II. OVERVIEW OF BASELINE JPEG SYSTEM

JPEG is a well known standardized image compression technique. JPEG loses information so the decompressed picture is not the same as the original one. The main reason for the use of Baseline JPEG is to reduce the size of image files. Reducing image files is an important procedure for transmitting files across networks or archiving libraries. Usually JPEG can remove the less important data before the compression; hence JPEG is able to compress images meaningfully, which produces a huge difference in the transmission time and the disk space. Fig 1 shows the basic Architecture of JPEG compression system. Here is a brief overview of the JPEG compression system [8]

The image is first subdivided into pixel blocks of size 8X8, which are processed from left to right, top to bottom. As each 8X8 block or sub image is encountered, its 64 pixels are level shifted by subtracting the quantity L/2, where L is the Gray level resolution of the image . The 2-D Forward Discrete Cosine Transform (FDCT) (Eq 1)[5]of the block is then computed, quantized using 64 corresponding step size values from the quantization table in Fig.2[3]. After quantization the DCT coefficients are rearranged in a zigzag sequence order as shown in the Fig.3.[3]

Since the one-dimensional reordered array generated under the zigzag pattern of Fig.3 is qualitatively arranged according to increasing spatial frequency, the JPEG coding procedure is designed to take the advantage of the long runs of zeros that normally result from the reordering. In particular, the nonzero AC coefficients (the term AC denotes all transform coefficients with the exception of the zeroth or DC coefficient) are coded using a variable-length code that defines the coefficient’s value and number of preceding zeros. The DC coefficient is difference coded relative to the DC coefficient of the previous sub image.

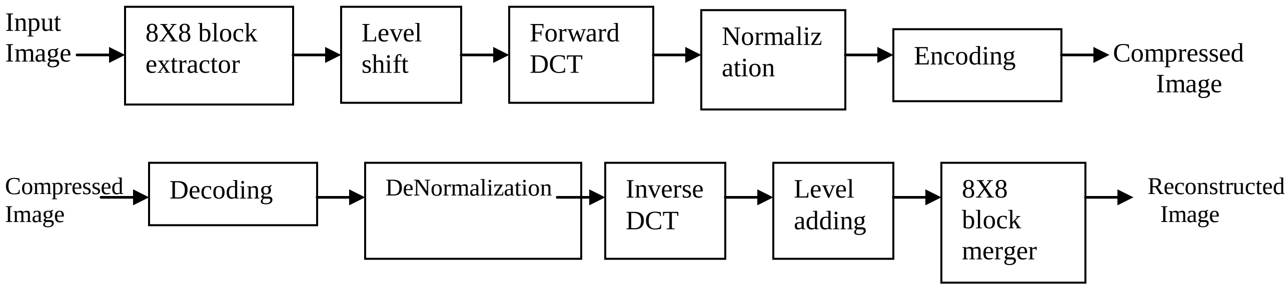

Fig. 1 Basic Architecture of Base line JPEG Compression

The 2-D DCT is

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (1)$$

for u, v = 0, 1, 2, ... , N -1

$$\alpha(u) = \begin{cases} \sqrt{1/N} & for \quad u = 0 \\ \sqrt{2/N} & for \quad u > 0 \end{cases} \quad (2)$$

$$\alpha(v) = \begin{cases} \sqrt{1/N} & for \quad v = 0 \\ \sqrt{2/N} & for \quad v > 0 \end{cases} \quad (3)$$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Fig.2 Quantization Matrix [3]

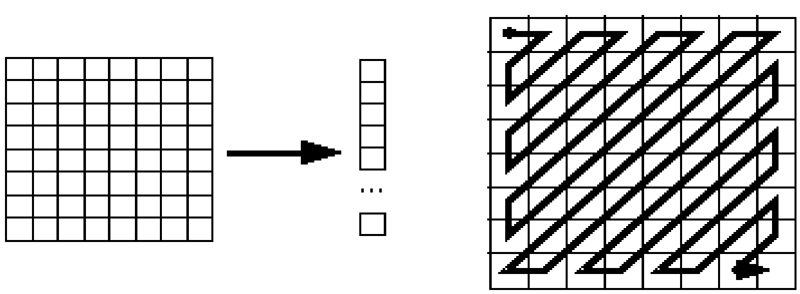

Fig.3 Zig Zag Sequence [3]

The decompression process performs an inverse procedure. It decodes the Huffman codes. Then, it makes the inversion of the Quantization step. In this stage, the decoder raises the small numbers by multiplying them by the quantization coefficients. The results are not accurate, but they are close to the original numbers of the DCT coefficients. An Inverse Discrete Cosine Transform (IDCT) (Eq.4) [6] is performed on the data received from the previous step. Finally add L/2 to each sub image. Place the sub images in their correct positions.

$$\hat{f}(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \qquad (4)$$

The error between the original image and reconstructed image is calculated in terms of Peak signal to noise ratio (PSNR) = 10 $\log_{10}$ (L^2/MSE) (5)

$$MSE=\frac{\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}\left[\hat{f}(x,y)-f(x,y)\right]^2}{mXn} \qquad (6)$$

MSE – Mean Squared Error

$\hat{f}(x,y)$ - Reconstructed Image

f (x, y) – Original Image

m x n – Size of the Image

III. K-MEANS CLUSTERING BASED JPEG ALGORITHM

Clustering is the process of grouping similar objects into groups. This is extensively used in image compression under the name hood of LBG algorithm [1]. In LBG algorithm, original 8x8 image pixels are clustered at the sender end and code book (clusters) is designed that minimizes error or noise. That is, image blocks that are having similar intensity distributions are grouped as clusters. These cluster's centers are called as code vectors. These set of vectors is called as code book. This codebook is sent to the receivers. During the coding phase of the image, image is divided into non-overlapping images of size 8x8 and the nearest code vector is found. That is, we will find to which cluster (code vector) or group they are near or similar. This code vector's index is communicated to the receiver which replaces the code vector in place of the sub image. Thus, instead of sending a sub image we will be communicating only indexes. If we assume that there exist 256 code vectors (clusters) in the code book then we need 8 bits to communicate index or cluster number. This is, very negligible when compared to communicating 8x8 sub image with 8 bit pixels. Thus, compression is achieved in the approach. For detailed explanation of LBG algorithm, readers are advised to refer [1].

In the following, we have listed the basic K-Means clustering algorithm [2].

K-means Algorithm

1) Select K random cluster centers z_1, z_2, …..z_k from the training data and call it as code vectors.
2) The training vectors x_1, x_2, …..x_N are classified into the cluster from which their distance is minimum.
3) Calculate the sum vector for every cluster by adding the corresponding components of all the training vectors that belong to the same cluster.
4) Calculate the centroid for each cluster $\hat{z}_1, \hat{z}_2 ......\hat{z}_k$ by dividing the individual components of the sum vector by the cluster strength.
5) If $z_1 == \hat{z}_1; z_2 == \hat{z}_2 ....z_k == \hat{z}_k$ then stop else take $\hat{z}_2 \quad as \quad z_2 ....\hat{z}_k \; as \quad z_k$ then go to step 3.

KDCT Algorithm

In this approach, we propose to study the possibility of designing code book using K-Means clustering algorithm by taking the quantized DCT arrays of each of the sub-images instead of original sub-images. We feel that clustering results will be good and we may get optimal code book as DCT transforms the image information into first few coefficients only while remaining coefficients are negligible. We have illustrated the steps in our algorithm which we denote as KDCT algorithm. Figure 4 a & b explains the steps involved in both coding and decoding with our algorithm.

1. The image is first subdivided into pixel blocks of size 8x8 which are processed from left to right, top to bottom.
2. For each block its 64 pixels are level shifted by subtracting the quantity L/2 where L is the gray level resolution.
3. The 2D-DCT of each block is computed.
4. Normalize the DCT blocks by standard normalization matrix.
5. The normalized blocks are grouped into clusters by using K-means Clustering described above.
6. Encode the cluster centers and indexes and send to the receiver.
7. The receiver decodes the cluster centers and indexes.
8. Based on the cluster centers and indexes the receiver costruct the output image.
9. Calculate the error between the original image and the reconstructed image.

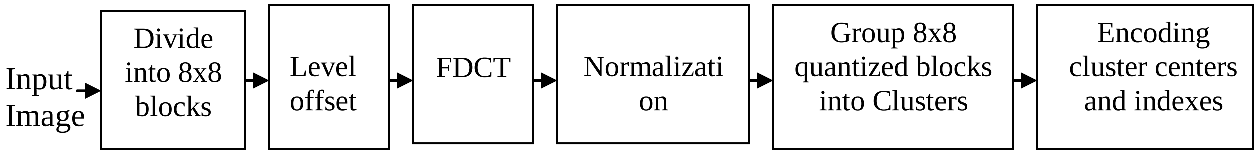

Fig 4 (a) Encoding

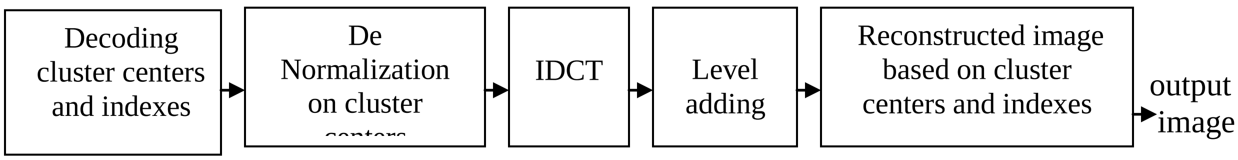

Fig 4 (b) Decoding

Fig 4 (c)

IV. EXPERIMENTAL WORK

In our study, a number of images in tiff format are used. All the above images are taken from USC-SIPI image database "http://sipi.usc.edu/database" [4] .Experiments are carried out under MS Windows XP version 2002, SP3 edition. The experimental system is equipped with Intel core 2 Duo 2.60 GHz processor with 1 GB RAM. Programs are written in C language under Micro Soft Visual Studio 2005 version 8.0. In our experiments, we have carried out observations with cluster. sizes of 64, 128, 256, 512, 1024 and 2048. Table-1 illustrates standard Base line JPEG's compression benefits and PSNR values with the selected images [4].

IMAGE	SIZE	CONVENTIONAL ENCODING	
		Compression Ratio	PSNR
Tire	128x128	8.1884	32.237621
Pop	128x128	5.0327	28.423471
glad	128x128	7.1612	31.973763
chess	128x128	9.3536	33.146183
floor	128x128	9.8218	35.765728
Cameraman	256x256	9.4928	31.6130
Clock	256x256	12.5286	34.8700
Couple	256x256	10.9814	35.2177
Airplane	256x256	17.1661	37.7847
Aerial	256x256	5.4787	30.0641
Lena	512x512	12.530	35.7939
Peppers	512x512	12.319	34.7604
Girl	512x512	12.1837	33.0329
Car	512x512	10.5626	35.0132

Stream	512x512	6.3508	29.5155
Man	1024x1024	10.1373	34.1561
Fish	1024x1024	18.0811	38.388565
Sea	1024x1024	18.2175	38.158947
Test	1024x1024	16.7857	21.953581

Table 1: Compression ratios and PSNR values of selected images with conventional Base line JPEG.

We have carried out experiments with the selected images and proposed algorithm. KDCT algorithm observations on Lena and pepper images is shown in Table 2 for various sizes of clusters. We may find that the compression ratio is reducing with the increasing in the cluster size. With all the images, we found acceptable PSNR value of reconstructed images for cluster size of 1024 and above. However, for cluster size smaller than 1024, PSNR is above 25dB.

Image	No. of clusters	64	128	256	512	1024	2048
lena	Compression Ratio	98.681	81.801	65.657	48.938	32.254	18.971
	PSNR	25.590	26.243	26.877	27.733	28.938	30.740
	% Compression benefit	87.30	84.68	80.915	74.396	61.152	33.951
	% Loss in PSNR	28.50	26.683	24.911	23.03	19.15	15.06
peppers	Compression Ratio	90.371	71.099	53.606	40.589	28.400	16.822
	PSNR	25.4811	26.1853	27.1461	27.9399	29.2864	31.41933
	% Compression benefit	86.368	82.913	77.019	69.649	56.623	26.768
	% Loss in PSNR	26.695	24.669	21.905	18.297	15.747	9.611

Table-2: Compression benefits and PSNR values of KDCT algorithm with “Lena” and “peppers” image

In brief, our experiments indicated the following

1) As the number of clusters in all the observations increases, the bandwidth requirements increased.
2) As the number of clusters in all the observations increases, the Peak-signal-to-noise ratio increased.
3) All the retrieved images based on Standard Baseline JPEG system and K-means clustering based JPEG system are almost same for visual appearance.
4) All the error images based on Conventional JPEG system and K-means based JPEG system are negligible
5) K-means clustering based JPEG system is somewhat complex as compared to the Conventional JPEG system

V. CONCLUSION

In this paper, K-means clustering based base line JPEG compression algorithm is proposed. We have compared our K-means clustering based base line JPEG compression with traditional base line JPEG compression. From our experiments it is evident that our approaches gives better compression ratios compared to traditional base line JPEG. The Peak Signal-to-Noise Ratio resulting from our approach is slightly less than traditional approach. However, all the reconstructed images resulting from our approach and original images are almost the same in visual perception point of view.

REFERENCES

[1] Bormin Huang,Antonlo J.Plaza,Joan Serra-Sagrista, Chulhee Lee, Younsong LI,Shen-En Qlan Editors Proceedings of SPIE "Satellite Datacompression, communications, and ProcessingVI",.3-5 august 2010 San Dlego, California, United States.

[2] K. K. Shukla and M.V. Prasad, "Lossy Image Compression Domain Decomposition-Based Algorithms," Springer Briefs in Computer Science, 2011

[3] Darrel Hankerson, Greg A. Harris, Peter, D.Johnson Jr," Introduction to information theoryand data compression" second edition. A CRCpress company.

[4] The USC–SIPI image database (http://sipi.usc.edu/database). Signal and image processing institute Ming Hgieh Department of Electrical Engineering.

[5] Andrew B.Watson "Image Compression using the discrete cosine transform" Mathematica Journal 4 (1), 1994, p-81-88

[6] Ken cabeen and peter gent," Image compression and the discrete cosine transform", Math 45 college of redwoods

[7] Muhammad F. Sabir, Hamid Rahim Sheikh, Robert W. Heath, Alan C.Bovik "A Joint Source-Channel Distortion Model for JPEG Compressed Images" IEEE Transactions on Image Processing Vol 15, No 6, pp1349-1364 June 2006

[8] Ricke J, Maass P, Lopez Hanninen E, Liebig T, Amthauer H, Stroszczynski C, Schauer W, Boskamp T and Wolf M, "Wavelet Versus JPEG and Fractal Compression. Impact on the Detection of Low-Contrast Details in Computed Radiographs," Investigate Radiology, Vol. 33, No. 8, Aug. 1998, PP. 456-463.

