RESEARCH ARTICLE OPEN ACCESS

ROI Based Image Compression in Baseline JPEG

M M M Kumar Varma[#1], Madhuri. Bagadi[#2]
Associate professor[1], M.Tech Student[2] Sri Sivani College of Engineering, Department of Computer Science Srikakulam, Andhrapradesh,, India

ABSTRACT
To improve the efficiency of standard JPEG compression algorithm an adaptive quantization technique based on the support for region of interest of compression is introduced. Since this is a lossy compression technique the less important bits are discarded and are not restored back during decompression. Adaptive quantization is carried out by applying two different quantization to the picture provided by the user. The user can select any part of the image and enter the required quality for compression. If according to the user the subject is more important than the background then more quality is provided to the subject than the background and vice- versa. Adaptive quantization in baseline sequential JPEG is carried out by applying Forward Discrete Cosine Transform (FDCT), two different quantization provided by the user for compression, thereby achieving region of interest compression and Inverse Discrete Cosine Transform (IDCT) for decompression. This technique makes sure that the memory is used efficiently. Moreover we have specifically designed this for identifying defects in the leather samples clearly.

I. INTRODUCTION

Images are being used very extensively to depict, analyze and share information in the world today. With the advent of new age gadgets like Smartphones and social networking websites like Facebook the use of images has increased even more. So, with the ever increasing usage of images, the need for efficient compression techniques without the loss of vital information is necessary. JPEG (Joint Photographic Experts Group) is a standardized compression method for organizing and storing digital images [1].
JPEG uses a lossy form of compression based on the discrete cosine transform [2]. There has been always a selectable tradeoff between storage size and image quality using JPEG. In Baseline Sequential Quantization the whole image can be compressed with desired quality,But the Region of Interest compression is not achieved. Adaptive Quantization should be used to obtain the desired compression for the subject and the background separately.
Kakarala and Bagadi described that the traditional JPEG compression standard, unlike JPEG2000, does not prescribe a means for spatially-variable quantization of transform coefficients. This prevents useful features such as region-of-interest (ROI) coding, where, for example, the subject in a photograph may be lightly compressed while the background is heavily compressed to hide irrelevant details [3].

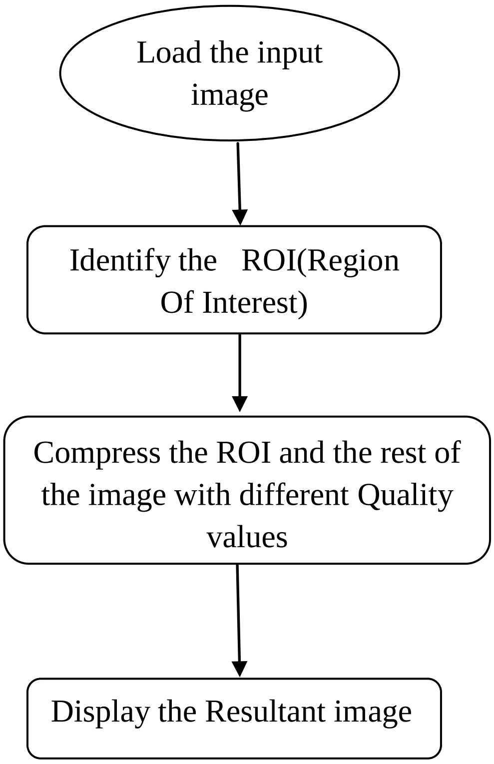
II. LITERATURE REVIEW

The American Mathematical Society described the process of Baseline compression algorithm (lossy compression) taking an example of image whose dimensions are 250x375 giving a total of 93750 pixels requiring 281250 bytes of storage. However, that image is finally compressed such that it takes 32414 bytes for storage i.e., the image has been compressed by a factor of roughly nine. This article describes how the image can be represented in such a small file (compressed) and how it may be reconstructed (decompressed) from this file. The color space transformation used in preparing the image for compression is explained in detail [5].

Aruna and Ramesh demonstrated a simple method for signaling adaptive quantization to the decoder, using the empty slots in standard baseline Huffman table. This is useful for ROI coding, which has the advantage of improving image quality in selected regions. They also showed how the encoder may be modified to produce an output that is in compliant with the standard and how the decoder may be modified to correctly recover the adaptively quantized image [6].The basic concepts and definitions involved in simple JPEG compression are referred. Description of differences between lossless and lossy compressions, Baseline sequential and progressive formats etc. Overview of the Baseline sequential JPEG encoding and decoding processes as a whole using flowcharts and various other figures [2]. Kakarala and Bagadi showed how the quantization may be adapted in each block and how the adaption may be signaled to the decoder in a memory efficient manner. They also explained how this adaptation allows ROI (Region of Interest) coding of subject and background. Examples are shown to illustrate adaptive quantization, signaling and decoding [3].

Ken cabeen and Peter gent explained the need for image compression and also illustrated the various equations of the DCT and its uses with image compression. The JPEG method is described step by step in detail. They also showed how the DCT (Discrete Cosine Transformation) and the IDCT (Inverse DCT) are applied to each block. This whole process is demonstrated taking a matrix as an example which represents the block of image-pixel values [4].Tutorial-Po-Hong Wu, Ting-Yu Chen describes the popular JPEG still image coding format. The purpose is to compress images while maintaining acceptable image quality. Also the focus is on the DCT and quantization. The entropy coding is also briefly discussed [7].

III. PROPOSED ALGORITHM



IV. DETAILED ALGORITHM OF COMPRESSION

STEP 1: The image is broken into 8x8 Blocks of Pixels.
STEP 2: Working from left to right,top to bottom DCT(DiscreteCosine Transform) is applied to each block.
STEP 3: Each block is compressed through Quantization.
STEP 4: The array of compressed blocks that constitute the image is stored in drastically reduced amount of space.
STEP 5: When desired the image is re-constructed through Decompression, a process that Uses Inverse DCT.
STEPS 1-5 above use DCT based image Compression[8].

V. HOW JPEG ALGORITHM WORKS FOR A GIVEN IMAGE

First, the image is broken into 8x8 blocks of pixels as shown in the image below.

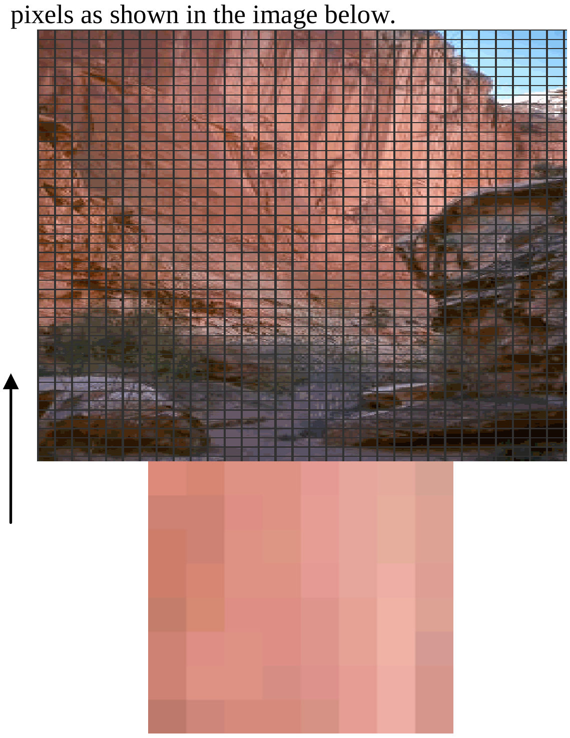


Fig: Shows an image with 8x8 blocks and inset image [image source[5]].

Now we convert this image into a matrix representation of pixels. Let each pixel is represented by p bits, the pixel value is in the range of (0, 2^p-1). We represent this matrix as Original as shown below.

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

Fig: Matrix representation of the pixels [4]

Pixel values are shifted from (0,2^p-1) to (-2^p-1,(2^p-1)-1).Since our p=8 so shift the values from(0,255) to (-12,127) because DCT requires range be centered around 0.Each value of the matrix from left to right and top to bottom is leveled off by subtracting 128.We represent the matrix by M.

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

Fig: Levelled of matrix [4]

1) Discrete Cosine Transform (DCT):

DCT converts intensity function into weighted sum of periodic basis (cosine) functions. It identifies band of spectral information which can be thrown away without loss of quality .So DCT converts spatial domain to frequency domain.

The DCT equation computes the i,jth entry of the DCT of an image

$$D(i,j) = \left(\frac{1}{\sqrt{2N}}\right) C(i)C(j)\sum\sum p(x,y)\cos\left[\frac{(2x+1)i\pi}{2N}\right]\cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$C(u) = \frac{1}{\sqrt{2}} \, if u = 0$$
$$= 1 \quad if u > 0$$

Where x,y range from 0 to N-1

To get the matrix form of the above equation, we will use the following equation

$$Ti,j = \frac{1}{\sqrt{N}} if i = 0$$
$$= \frac{\sqrt{2}}{\sqrt{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right] if i > 0$$

N is the size of the block.

P(x, y) represents x y^{th} element of the image represented by matrix P.

DCT matrix of an 8x8 blocks is as shown below:

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

Fig: DCT matrix for 8x8 blocks [4]

Resultant Matrix D is obtained by multiplying original matrix M with DCT matrix T and then again multiplying with transpose of DCT matrix T' so that the original value of M is retained.

The Discrete Cosine Transformation is accomplished by matrix multiplication

D=TMT'

Resultant Matrix D is obtained:

$$D = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

Fig: Resultant Matrix D. [4]

2) QUANTIZATION

Varying levels in 8x8 blocks of DCT coefficient in which compression and quantization are obtained by selecting specific quantization matrices are compressed by quantization.

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Fig: Quantized 8x8 block of DCT coefficients (level 50). [4]

For a quality level greater than 50 (less compression, higher image quality), the standard quantization matrix is multiplied by (100-quality level)/50. For a quality level less than 50 (more compression, lower image quality), the standard quantization matrix is multiplied by 50/quality level. [4]

$$Q_{10} = \begin{bmatrix} 80 & 60 & 50 & 80 & 120 & 200 & 255 & 255 \\ 55 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

Fig: Quantized 8x8 block of DCT coefficients (level 10). [4]

$$Q_{90} = \begin{bmatrix} 3 & 2 & 2 & 3 & 5 & 8 & 10 & 12 \\ 2 & 2 & 3 & 4 & 5 & 12 & 12 & 11 \\ 3 & 3 & 3 & 5 & 8 & 11 & 14 & 11 \\ 3 & 3 & 4 & 6 & 10 & 17 & 16 & 12 \\ 4 & 4 & 7 & 11 & 14 & 22 & 21 & 15 \\ 5 & 7 & 11 & 13 & 16 & 12 & 23 & 18 \\ 10 & 13 & 16 & 17 & 21 & 24 & 24 & 21 \\ 14 & 18 & 19 & 20 & 22 & 20 & 20 & 20 \end{bmatrix}$$

Fig: Quantized 8x8 block of DCT coefficients (level 90). [4]

Quantization is achieved by dividing each element in the transformed image matrix D by the corresponding element in the quantization matrix, and then rounding to the nearest integer value.

$$Ci,j = round(\frac{Di,j}{Qi,j})$$

Quantization matrix Q50 is used:

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig: Quantized Matrix [4].

In the matrix C obtained above, the lowest frequency coefficients situated in the upper –left corner are sensitive to human eye, so they are not discarded where as high frequency coefficients which are less important situated in the lower right corner are discarded leading to a lossy compression.

3) CODING

The image which is constituted of an array of compressed blocks is stored in a drastically reduced amount of space. An encoder converts all coefficients of quantized matrix C to a stream of binary data after which most of the coefficients are zeros.

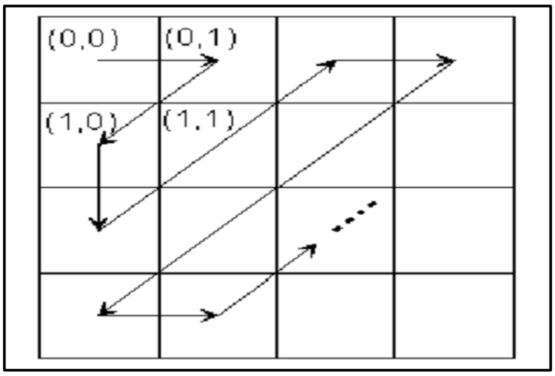


Fig: zig-zag sequence of compressed binary data [4].

4) DECOMPRESSION

Then the image is re-constructed through decompression, a process that uses the Inverse DCT.Reconstruction of our image begins by decoding the bit stream representing the quantized matrix C.

$$Ri,j = (Qi,j) * (Ci,j)$$

$$R = \begin{bmatrix} 160 & 44 & 20 & 80 & 24 & 0 & 0 & 0 \\ 36 & 108 & 14 & 38 & 26 & 0 & 0 & 0 \\ -98 & -65 & 16 & -48 & -40 & 0 & 0 & 0 \\ -42 & -85 & 0 & -29 & 0 & 0 & 0 & 0 \\ -36 & 22 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

5) INVERSE DISCRETE COSINE TRANSFORM(IDCT)

The Inverse Discrete Cosine Transform (IDCT) is applied to matrix R by multiplying transpose of DCT matrix first T' and then with DCT matrix T, which is rounded to the nearest integer. Original Matrix M is obtained by adding 128 to each entry so that the pixel value range from (0, 2^p-1).

$$N = round(T'\ R\ T) + 128$$

$$Decompressed = \begin{bmatrix} 149 & 134 & 119 & 116 & 121 & 126 & 127 & 128 \\ 204 & 168 & 140 & 144 & 155 & 150 & 135 & 125 \\ 253 & 195 & 155 & 166 & 183 & 165 & 131 & 111 \\ 245 & 185 & 148 & 166 & 184 & 160 & 124 & 107 \\ 188 & 149 & 132 & 155 & 172 & 159 & 141 & 136 \\ 132 & 123 & 125 & 143 & 160 & 166 & 168 & 171 \\ 109 & 119 & 126 & 128 & 139 & 158 & 168 & 166 \\ 111 & 127 & 127 & 114 & 118 & 141 & 147 & 135 \end{bmatrix}$$

Fig: Output Matrix (Decompressed) [4].

VI. EXPERIMENTAL RESULTS

The process of coding was carried out with the help of MATLAB. First with the quality value provided by the user, the pure image is compressed with the desired quality.Finally the selected portion which is important to the user is compressed by providing more quality and the portion of image which is irrelevant to the user is compressed with less quality, thus gaining Region of Interest compression. This process is implemented on defective leather samples. Here, only the defective part of the leather sample is important and thus we can use Region of interest compression by compressing the defect with high quality so that the defect is very clear. Whereas the rest of the sample can be compressed more as it contains the irrelevant information. Thus, Region of interest compression gives more efficient results(by highlighting the defects) than compressing the whole image with the same quality.

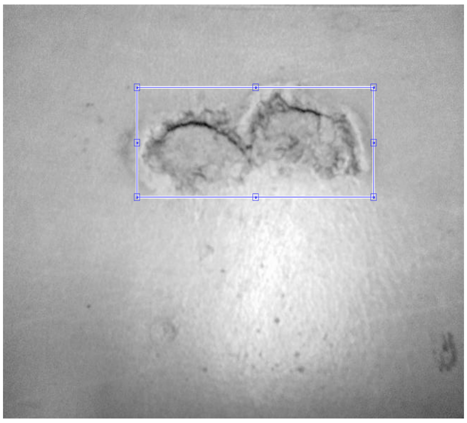
The original image of a leather sample is

Complete image compression with quality 95%

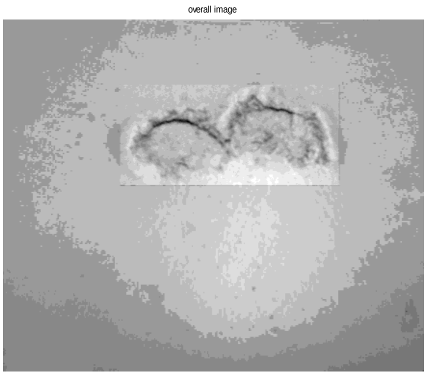
The size of this image which is compressed with the same quality throughout is 35.5MB.

VII. REGION OF INTEREST COMPRESSION

The Region of Interest is selected i.e. the defect is selected by the user



Quality of compression that is applied to the background is 1% Quality of compression that is applied to the subject(Defect) is 95%



The size of this image which is compressed based on Region of Interest is 5.64MB.

VIII. CONCLUSION

In this paper, we describe a much efficient method of compressing image based on the idea of what is required and what is not in the image data, and there by applying the required amount of compression at different parts of the image. The user can select any part of image and can enter the required quality for compression, if the subject is more important then more quality to the subject is provided by the user and less to the background. otherwise if the subject is irrelevant to user and background is important then more quality is provided to the background and less to the subject. By using this we can create images that are memory-efficient that contains only the required data from the JPEG file.

REFERENCES

[1] http://en.wikipedia.org/wiki/Image_file_formats#JPEG.2FJFIF; Wikipedia

[2] http://en.wikipedia.org/wiki/JPEG;Wikipedia

[3] *A method for signalling block-adaptive quantization in baseline sequential JPEG Kakarala*, R. ; Bagadi, R. TENCON 2009 - 2009 IEEE Region 10 Conference; 10.1109/TENCON.2009.5396160

[4] *Image Compression and Discrete Cosine Transform ;Ken Cabeen and Peter Gent,* Math 45,College of Redwoods.

[5] http://www.ams.org/samplings/feature-column/fcarc-image-compression

[6] http://www.ijmer.com/papers/Vol2_Issue4/FN2428292831.pdf

[7] *Advance Digital Signal Processing-JPEG for Still Image* Compression-Tutorial-Po-Hong Wu, Ting-Yu Chen

[8] Gonzalez, R.C. and R.E.Woods, "*Digital image processing*", New Jersey: Prentice Hall, 2002