An Efficient Image Compression Algorithm Based on Histogram Based Block Optimization and Arithmetic Coding

Subarna Dutta, Aditya Abhinav, Partha Dutta, Purushottam Kumar, and Amiya Halder

***Abstract*—The need for an efficient technique for compression of Images is ever increasing because the raw images need large amounts of disk space seems to be a big disadvantage during transmission and storage. In this paper, an efficient algorithm has been proposed for lossy image compression/decompression scheme using histogram based block optimization and arithmetic coding. Experimental result shows that the proposed algorithm gives better compression ratio as that of JPEG and some other techniques and PSNR value of decompressed image is comparable to JPEG.**

***Index Terms*—Lossless compression, lossy compression, block optimization, histogram, arithmetic coding.**

I. Introduction

Images are being used day to day in various fields and the periphery of application is also increasing. It has got applications in field of medicine, astronomy, biological research, photography and many more. With the advent of internet and WWW, there was a need to transmit images and other multimedia objects over the network and for this came various compression techniques to achieve better throughput. These techniques aimed to reduce the size of image in a way that it can be easily regenerated at the receiver's end [1]. Some of these techniques focused on high compression ratio while other on better quality and appreciable compression ratio. With ever-increasing growth of multimedia applications over the internet, data compression has become a necessity. Compression hardware having throughput that can match the capacity of high-speed data communication channel is the backbone of such on-line communication. This can be achieved by designing the compression /decompression algorithm [2]-[4].

An image is a two dimensional object that provides visual information such as colors, depth, contrast and so on to the human eyes. A photograph taken from a 35mm camera is an example of an analog image. Information stored is continuous in either spatial direction. A digital image is obtained by sampling the information at regular intervals(pixel) in either direction. The information for each pixel is stored numerically in the binary base. The number of bits in each pixel represents the precision of the sample. Higher the precision value better would be the picture quality.

In case of digital images adjacent pixels are often related with each other as the pixel values have very less variation in between them. These sorts of redundancies are generally exploited to get compression. The redundancies can either be spatial or spectral [4]. The efficiency of image compression algorithm depends on how fast and how best these redundancies can be exploited. The main objective behind this clever exploitation of redundancies is to reduce the image size without any appreciable loss of quality.

Images typically use either 8-bit or 24-bit color. In case of 8-bit the range of pixel value is from 0-255 (means 256 different colors). In case of 24-bit scheme, each pixel uses 24-bit and each 8-bit is used for representing three different primary colors red, green, blue (R, G, B). Image compression algorithms are divided into two groups: Lossless and lossy compression [5]-[7].

Lossless image compression techniques compress image in a way that there is no data loss in decompressed image. Lossless compression technique generally doesn't give high compression ratio and there is no quality loss at all.

In case of lossy image compression techniques, the decompressed image suffers from some amount of data loss. All the data is not retrieved in case of lossy compression. In this case the compression ratio achieved is generally higher but at the expense of image quality.

Various lossy and lossless compression techniques have been developed for image compression [11]-[18]. Here, the proposed efficient image compression algorithm that work in two steps: first, it divides the images block by block sub images (2×2, 3×3, 4×4 etc.), and then it is replaced by mode value of the block. In second step, the arithmetic coding is applied. It should yield higher level of compression as compared to JPEG standard and also giving comparable PSNR value to that of JPEG [8]-[10]. Main features of the proposed algorithm is that it is very simple in implementation and gives better compression ratio for almost the same image quality.

The rest of paper is organized as follows: in Section II, gives the contrast finding and block size determination. Section III gives the histogram based block optimization. Section IV describes about Arithmetic coding. Section V gives the proposed algorithm and Section VI describes the experimental results and Section VII concludes the paper.

Manuscript received August 9, 2012; revised October 2, 2012.

S. Dutta, A. Abhinav, P. Dutta, and P. Kumar are with the Department of Information Technology, St. Thomas' College of Engineering and Technology, Kolkata, India (e-mail: subarnadutta11@gmail.com, aditya.abhinav.ankur@gmail.com, partha.dutta1988@gmail.com, kumar.stcet@gmail.com).

Amiya Halder is with the Department of Computer Science and Engineering, St. Thomas' College of Engineering and Technology, Kolkata, India (e-mail: amiya_halder@indiatimes.com).

II. CONTRAST FINDING AND BLOCK SIZE DETERMINATION

Contrast of an image is the range of the intensity values of the pixel in the image. Higher is the range, higher will be the contrast of the image. For example, the low contrast image is an image where all the pixel values of the image are restricted within a very small range. Thus, the high correlation exists between the neighboring pixels. Here, block size that can be used for block optimization will be higher. In case of high contrast image, we consider small block size.

Mathematically it can be represented as:-

$$Range(R) = \max(K) - \min(K)$$

where K is the grey level value assigned to one or more pixel. We consider, the different block size is determined in following ways:-

$$Block\,size = \begin{cases} 2\times 2, if\ R > T_1 \\ 3\times 3, if\ R < T_1\ and\ R > T_2 \\ 4\times 4, if\ R < T_2 \end{cases}$$

where R is the range and T_1 and T_2 are the threshold values.

The reason behind doing so is to maintain the quality of image while giving high compression ratio. The main advantage of this stage is that it adds dynamic behavior to algorithm to maintain appreciable quality while giving high compression ratio.

III. HISTOGRAM BASED BLOCK OPTIMIZATION

In an image there is usually a likelihood of high correlation between pixels. Such correlations between pixels or a block of pixels are exploited to achieve image compression [12], [13]. In this paper, $M\times N$ image is considered- that is, there are M number of rows each containing N pixel values. The basic approach in block optimization is for each block in the image one modified pixel value is generated and is stored in compressed file leading to compression. But the drawback of this method is some amount of data loss in the decompressed file.

The proposed algorithm mitigated this loss to some extent by taking histogram based block optimization. This methodology takes a block under consideration and draws a histogram of the block. Histogram gives the total no. of pixels assigned to each gray level. So the new set of values be

S_i= {n_k | n_k is the frequency for the kth grey level for ith pixel in the block}. Then the peak of the histogram is found out.

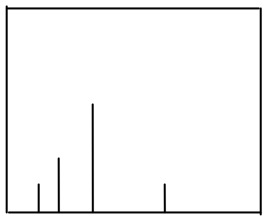
M_i=*max* ({S_i}), where M_i is the maximum value of the block.

This peak represents the pixel value that is repeated maximum number of times in the block. This pixel value is then stored for the block in compressed file leading to compression as well as reduced loss while decompression as compared to averaging based block optimization. The process can be best illustrated with an example.

For example, suppose the original 3×3 pixel neighborhoods are:

21	25	21
19	21	19
21	25	18

The histogram of the above block is:



The peak value is 4 and corresponds to pixel value 21. This 21 is stored for the block in optimized block.

This way the process is carried out. The main advantage of histogram based block optimization is that it leads to less data loss while decompression is achieved.

IV. ARITHMETIC CODING

Arithmetic coding is a variable length encoding used in lossless data compression. In this coding method encodes a specific number of values into a single value is used to replace them owing to compression and in decompression this single value gives back the original values [5-7]. This coding reduces the size without any sort of compromise in quality of image as there is no data loss.

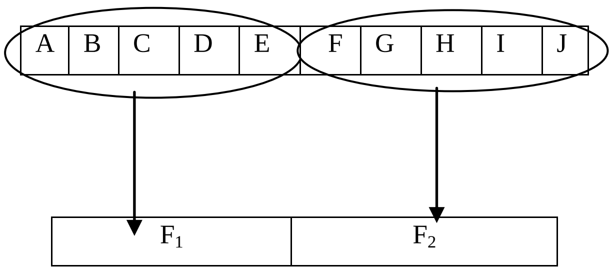


Fig. 1. This proposed algorithm, five symbols are taken at a time and these five are replaced with a single float value. Thus for five values there is a single value in compressed file thus giving 4/5=80% compression.

V. PROPOSED ALGORITHM

The proposed algorithm takes input as an 8-bit image of any arbitrary dimension, (preferred a square dimension). The proposed algorithm compresses the image to lesser size compared to JPEG while giving comparable PSNR value.

A. Algorithm for Encoding

1) Read the image and its size attributes.
2) Block preparation
   - Find the maximum and minimum pixel value for the entire image.
   - Compute the Range(R) = *max* (K) – *min* (K), where K is the gray level value assigned.
   - Set block size ($R\times R$). Here, we consider as 2 when range be high (high contrast image), as 3 when medium range (medium contrast) and as4 when range is less (low contrast).

$$Mask\ size = \begin{cases} 2 \times 2, if\ R > T_1 \\ 3 \times 3, if\ R < T_1\ and\ R > T_2 \\ 4 \times 4, if\ R < T_2 \end{cases}$$

- Store this value to be used for decompression.

3) Histogram based block optimization
   - Find the histogram of the block.
   - Find the peak of the histogram.
   - Assign the pixel value corresponding to the peak for the block in the reduced image.
4) Arithmetic coding
   - Read ‘n’ symbols and find probabilities of each distinct symbol.
   - Assign a range to all symbols as per their probabilities on the probability line (from 0-1)
   - Read the first symbol from the ‘n’ symbols.
   - Read the corresponding range.
   - On this new range, again assign all symbols and their corresponding new range as per their probabilities.
   - Upper limit of the symbol is modified as = (lower limit of the symbol + new range) $\times$ probability of the symbol.
   - Lower limit of the ith symbol becomes the upper limit of the (i+1)th symbol.
   - Continue the process till the nth symbol is processed.
   - Store the lower limit of the last processed symbol and this value is the code for the block of ‘n’ symbol.

This gives the final compressed file.

B. Algorithm for Decoding

1) Read the decompressed file.
2) Arithmetic decoding
   - Read the encoded symbol.
   - Check the stored range values and identify the range within which this encoded value falls.
   - Read the symbol corresponding to that range and store this value in the output file.
   - Modify the encoded value as:
   - (Encoded value - lower limit)/probability of the corresponding symbol.
   - Continue till all the symbols are decoded.
3) Read the block size that was stored during the encoding process.
4) Size expansion/creation of decompressed image
   - Read a byte.
   - Create a block of size (R $\times$ R) for the byte and fill it with the byte.
   - Repeat the process till the last byte and creating blocks at the respective position.
   - Write the matrix so obtained in a file.

This gives the final decompressed file.

VI. Experimental Results

The proposed algorithm has been simulated using MATLAB and DEV C++. The input images are generally taken in portable gray map (.pgm) format. Various sizes of images are taken into consideration. These images are of 8-bit precision. The compression ratio achieved is better than JPEG and other compression techniques. The algorithm is also able to compress very large images which JPEG failed to do. All the decompressed images reconstructed after compression by the proposed algorithm exhibit comparable PSNR to that of JPEG and that given by other techniques. All the experimental results are illustrated at the end of paper and a comparison with various techniques is also given. The performances of the proposed methods are compared with that of the BTC [15], BTC-VQ[16], BTC-DCT[17], BTC-VQ-DCT[17], BTC-PVQ-DCT I[18] , BTC-PVQ-DCT II [18] method in terms of bit per pixel rate and coding quality for the average values of four images as listed in Tables I. Table II shows the compression ratio is better than JPEG and PSNR value of decompressed image is comparable to JPEG.

Here compression ratio measured in terms of bpp and the image quality in terms of PSNR. The bpp and PSNR may be defined, respectively, as

$$bpp(bit\ /\ pixel) = \frac{8 \times compressed\ \ file\ \ size}{Actual\ \ file\ \ size}$$

and

$$PSNR = 20 \times \log \frac{255}{RMSE}$$

where

$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [\, f(i,j) - \hat{f}(i,j)]^2}$$

where M: number of row, N: number of column, $f(i,j)$: original image matrix, $\hat{f}(i,j)$: output image matrix. Fig. 2 shows the various resultant decompressed images.

(a) (b) (c) (d)

Fig. 2. (a), (c) Different types of original images, (b), (d) Decompressed image by proposed method.

TABLE I: THE COMPARISON OF PSNR AND BIT RATE USING VARIOUS HYBRID CODING SCHEMES FOR FOUR IMAGES.

Method	Lena	Pepper	Airplane	Zelda	Avg. PSNR	Avg. Bpp
BTC [15]	33.24	34.11	33.3	36.74	34.35	2
BTC-VQ [16]	30.36	31.33	29.93	34.27	31.47	1.5
BTC-DCT [17]	29.24	29.66	28.15	34.41	30.37	1.1315
BTC-VQ-DCT [17]	27.85	28.48	26.45	31.24	28.51	0.6315
BTC-PVQ-DCT I[18]	27.67	28.19	26.15	31.03	28.26	0.5865
BTC-PVQ-DCT II[18]	27.80	28.33	26.32	31.15	28.40	0.5846
Proposed Method	30.27	29.74	24.30	30.52	28.70	0.3442

TABLE II: THE COMPARISON OF COMPRESSED SIZE BETWEEN JPEG AND PROPOSED METHOD FOR DIFFERENT IMAGES.

Original Image	Original Image Size	Compressed size by Proposed Algorithm	Size by JPEG Compression	PSNR values by proposed algorithm
Mystery bridge	3.51MB	180 KB	183 KB	27.09
Canopy effect	1.37MB	31.2 KB	31 KB	40.95
Epic falls	1.37MB	70.3 KB	80 KB	30.54
Girl	256 KB	12.8 KB	18.6 KB	34.22
sunset	3.51MB	180 KB	235 KB	23.79
Lena	254 KB	12.7 KB	35.7 KB	30.27
Zelda	254 KB	5.64 KB	27.6 KB	30.52
Peppers	256 KB	12.8 KB	32.8 KB	29.74
Camera man	256 KB	12.8 KB	26.0 KB	34.79

VII. CONCLUSIONS

In this paper, a new hybrid algorithm for image compression/decompression has been proposed. The main advantage of this algorithm is that this algorithm is applicable to a variety of image file types some of which cannot be compressed with that of JPEG and some other file types. It also works well in case of images having huge size which some of other algorithms failed to give proper results. The algorithm gives compression ratio always greater than 95% in case of any image which other algorithm and JPEG failed to give. The PSNR values for various images obtained are found to be comparable to that of JPEG standards. The algorithm is applied on 8-bit grey level image but the algorithm can be applied to 24-bit colored image also. This application to 24-bit colored image is one of our future concerns.

REFERENCES

[1] S. Khuri and H.-C. Hsu “Interactive packages for learning image compression algorithms,” *ITiCSE*, Helsinki, Finland. 2000

[2] S. Bhattacharjee, S. Das, D. R. Choudhury, and P. P. Chouduri, “A pipelined architecture algorithm for image compression,” *Proc. Data Compression Conference*, Saltlake City, USA, March 1997.

[3] J. Ritter and P. Molitor, “A pipelined architecture for partitioned DWT based lossy image compression using FPGA's,” *International Symposium on FPGA*, pp. 201-206, 2001.

[4] A. Halder, D. K. Kole, and S. Bhattacharjee, “On-line colour image compression based on pipelined architecture,” *ICCEE*, Dubai, UAE, pp. 28-30, Dec. 2009.

[5] Pratt and K. William, *Digital Image Processing*.

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson Education, 2002.

[7] D. Saloman, *Data Compression*.

[8] G. K. Wallace, “The JPEG still picture compression standard,” *Commun. ACM*, vol. 34, pp. 31-44, April 1991.

[9] T. Acharya and P.-S. Tsai, *JPEG Standard for Image Compression*, 2000.

[10] W. B. Pennebaker and J. L. Mitchell, “JPEG: Still image data compression standard,” *Van Nostrand Reinhold*, New York, 1993.

[11] A. Halder, S. Dey, S. Mukherjee, and A. Banerjee, “An efficient image compression algorithm based on block optimization and byte compression,” *ICISA-2010*, Chennai, Tamilnadu, India, pp. 14-18, Feb 6, 2010.

[12] E. J. Delp and O. R. Mitchell, “Image compression using block truncation coding,” *IEEE Transactions on Communications*, vol. 27, pp. 1335 -1342, 1979.

[13] C. H. Kuo, C. F. Chen, and W. Hsia, “A comression algorithm based on classified interpolative block truncation coding and vector quantization,” *Journal of Information Science and Engineering*, vol. 15, pp. 1-9, 1999.

[14] C. Amerijckx, J. D. Legaty, and M. Verleysenz, “Image compression using self-organizing maps,” *Systems Analysis Modeling Simulation* vol. 43, no. 11, November 2003.

[15] M. D. Lema and O. R. Mitchell, “Absolute moment block truncation coding and its application to color images,” *IEEE Trans. Commun*., vol. 32, no. 10, pp. 1148-1157, Oct. 1984.

[16] H. Jason and Y. C. Hu, “The study of hybrid image coding scheme based on block truncation coding and vector quantization,” in *Proc. TANET*, Taipei, vol. 29, no. 31, pp. 633-638,Oct. 2003.

[17] Y. Wu and D. C. Coll, “BTC-VQ-DCT hybrid coding of digital images,” *IEEE Trans. Commun*., vol. 39, no. 9, pp. 1283-1287, Sept. 1991.

[18] C.-H. Wei and T.-J. Kuo, “A hybrid coding scheme for image compression,” *IEEE ICSS*, pp. 457-461, 2005.

Ms. Subarna Dutta received the B.E degree in Information Technology from the St. Thomas’ Colllege of Engg. & Tech. West Bengal, India, in 2011. His area of interest is in the subject of Image Processing.

Aditya Abhinav received the B.E degree in Information Technology from the St. Thomas’ Colllege of Engg. & Tech. West Bengal, India, in 2011. His area of interest is in the subject of Image Processing.

Partha Dutta received the B.E in Information Technology from the St. Thomas’ Colllege of Engg. & Tech. West Bengal, India, in 2011. His area of interest is in the subject of Image Processing.

Purushottam Kumar received the B.E in Information Technology from the St. Thomas’ Colllege of Engg. & Tech. West Bengal, India, in 2011. His area of interest is in the subject of Image Processing.

Amiya Halder received the B. Sc degree with Honours in Physics and B. Tech. in Computer Science & Engineering from the University of Calcutta, West Bengal, India, in 1998 and 2001, respectively. He obtained the M.E. degree in Computer Science & Engineering from the Jadavpur University, West Bengal, India, in 2003. At present, he is doing his Ph.D. in the field of Image Processing from Jadavpur University, Kolkata, West Bengal. Mr. Halder is a Asst. Prof. of the Department in Computer Science & Engineering, St. Thomas’ College of Engineering & Technology, Kolkata, West Bengal, India. He has authored or co-authored over 20 International conference or journal papers in area of Image Processing. Amiya Halder and Nilavra Pathak, “An Evolutionary Dynamic Clustering based Colour Image Segmentation”, International Journal of Image Processing, Vol 4, No. 6, pp.548-555, January/February 2011. Subarna Bhattacharjee, Dipak Kumar Kole and Amiya Halder,"On-line Image Compression based on Pipelined Architecture",International Journal of Computer Theory and Engineering, Vol.2, No.6, pp. 851-859,December, 2010. Ayan Banerjee and Amiya Halder, “An Efficient Image Compression Algorithm for almost Dual-Color Image Based on K-Means Clustering, Bit-Map Generation and RLE”, ICCCT 2010, IEEE Xplore, pp. 201-205, Allahabad, India.