Computational Complexity Reduction of JPEG Images

Mahmud Hasan, Kamruddin Md. Nur, Tanzeem Bin Noor

Abstract— The JPEG Images play a significant role in present multimedia based computing industry. Being a popular lossy mode of image compression, The JPEG has extensively been being used in almost all sorts of digital device including the mobile phones, tablet and handheld computers. Although the popularly used Baseline JPEG Algorithm is an easy one to be performed by the powerful processors, still the small devices of less capable processors suffer a lot from encoding or decoding a JPEG image by the Baseline JPEG Algorithm. This is due to some complex computations required by Baseline JPEG. This paper discovers the computational cost currently needed by Baseline JPEG and suggests an efficient way to encode or decode the JPEG images so that the overall computational cost of the Baseline JPEG Algorithm is reduced with less affecting the obtainable Compression Ratio and Peak Signal to Noise Ratio (PSNR). The suggested cost reduction technique has been tested upon some small computing devices and comparative cost analysis is presented.

Index Terms— Baseline JPEG, Compression Ratio, Computational Cost, Default JPEG Quantization, Element-wise Division, PSNR, Psychovisual Redundancy.

◆

1. INTRODUCTION

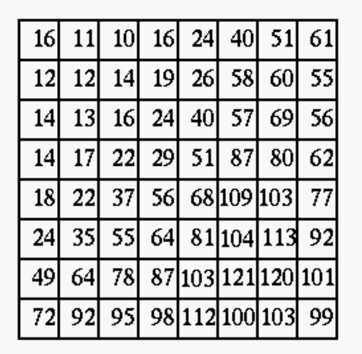
AT the beginning of 90th decade, the Joint Photographic Experts Group introduced The JPEG Image Format as an image compression standard [1-5]. This standard has four different modes of operation, though; the Baseline JPEG is most popular among them. It had been very much popular to the end users due to its easy compression-decompression (CODEC) technique, more compression ratio, higher PSNR and portability. Specially, due to its less requirement of bandwidth [4, 5], JPEG images had been a common platform for sharing images through network and Internet. Since the number of erstwhile computing devices was less as compared to that of present industry and most of them were desktop devices, the Joint Photographic Experts Group did not have to think more about the computational cost of JPEG images. However, as the production and usage of tablet and handheld computing devices with less capable processors increased, the necessity of computational cost reduction of JPEG images has been appeared. Although, the Baseline JPEG has observed a number of improvements since its appearance to the industry [6-18], most of them dealt with either quality improvement or further compression of JPEG images. The necessity of computational cost reduction of JPEG images has ever been overlooked by the researchers of this arena.

Generally, the small computing devices have less capable processors that are designed to do simple processing tasks as usually done by those devices. The Baseline JPEG Algorithm is although very simple to be carried out, it has severals steps that practicably do overhead computation in some cases. Though, the powerful processors do not seem to be bothered by those extra computation, it is obviously a less efficient and time consuming task for the low capacity processors of small computing devices. Digital Cameras, the principal producers of JPEG images, are not even free from such overhead computation and therefore the users have to wait for a certain while after taking a snap so that the captured raw image can be saved by the digital camera in JPEG format. Although for today's modern cameras this waiting time is usually less than a second, when we consider the time in mili, micro or nano second units; it is a big deal yet. However, this time can be substantially reduced if we can indicate the step for which the Baseline JPEG consumes more time and reduce time complexity of that step. This paper explores the Baseline JPEG steps that consume much time of the processors due to a number of overhead computations. After a careful explanation of the Baseline JPEG Algorithm and its overhead computing cost, this paper suggests a way to reduce the computational complexity of the JPEG images. The rest of the paper is organized as follows. Section 2 describes in detail how the Baseline JPEG compresses a raw image. Section 3 discusses the computational complexity of the Baseline JPEG followed by the proposed modified approach in section 4. The experimental results are shown in section 5 and an overall brief of the paper is given in a nutshell in section 6.

Mahmud Hasan is currently teaching at the Dept. of Computer Science & Engineering as a full-time Lecturer in Stamford University Bangladesh.
E-mail: hasanpoet@gmail.com
Kamruddin Md. Nur is currently teaching at the Dept. of Computer Science & Engineering as Assistant Professor in Stamford University Bangladesh.
E-mail: kamruddin.nur@gmail.com
Tanzeem Bin Noor is currently teaching at the Dept. of Computer Science & Engineering as a full-time Ltecturer in Stamford University Bangladesh.
E-mail: tanzeem.noor@gmail.com

2. The Baseline JPEG Working Procedure

The Baseline JPEG Algorithm is simple and straight forward [1-5]. It takes a raw image represented by a pixel grid and then divides it into a number of non-overlapping 8×8 blocks. Then it performs a shifting operation upon each of the pixels of those blocks. The shifting is performed by adding -128 with each pixel. For color images, the color planes are first separated and then each plane is individually divided into non-overlapping 8×8 blocks. The scaling or shifting is performed in the same way as just described. The Baseline JPEG provides an optional transformation from RGB color domain to YC_bC_r color domain while separating the color planes [6, 7]. After level shifting, the Baseline JPEG performs a Two Dimensional Fast Discrete Cosine Transform (2D-FDCT) upon each 8×8 block. This transforms the image block from spatial domain to DCT domain that facilitates the reduction of psychovisual redundancy. The Baseline JPEG standard provides a Default Quantization Matrix as shown in figure 1 for quantizing the blocks. Each pixel value of the DCT transformed image block is then divided by the corresponding quantization factor given by the Default Quantization Matrix. That is, an element-wise division is carried out in this step and the fractional parts of the result are rounded up. A zigzag scan is then performed upon each quantized block. The order of zigzag scan is shown in figure 2. The first coefficient of each block obtained as a result of zigzag scan is called the DC Coefficient while the other coefficients are called AC coefficients [1-5]. An End-of-Block (EOB) at the end of each block indicates the rest of the coefficients of the block are all zero. Figure 3 shows an example block after zigzag scan is performed where DC coefficients, AC coefficients and EOB are indicated. The last step of Baseline JPEG Encoding Algorithm is Optimal Coding. Although any optimal coding can be used at this stage without affecting the resulting number of bits, the Baseline JPEG provides a Default Huffman coding Table for coding the DC and AC coefficients. It also provides a Category Table so that the coding stage is facilitated. The Coefficients Coding Category Table is given in table 1 and the Default DC Coding Table is given in table 2 for convenience. First, the Baseline JPEG encoder looks for the category of a Differenced DC Coefficient. Then, from table 2, the corresponding Base Code is chosen. If the Base Code Length is less than the Total Code Length, the remaining bits are chosen from the least significant



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. 1. The Default JPEG Quantization Matrix

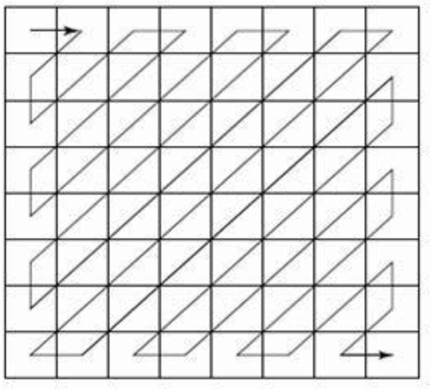


Fig. 2. ZigZag Scan Ordering of Baseline JPEG

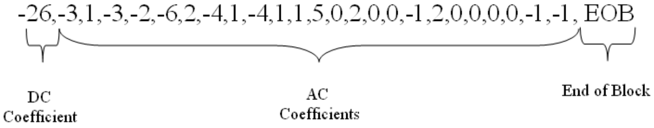


Fig. 3. Sequence of DC-AC Coefficients and EOB

TABLE 1: JPEG COEFFICIENTS CODING CATEGORIES

Range	DC Difference Category	AC Category
0	0	N/A
-1,1	1	1
-3,-2,2,3	2	2
-7,…,-4,4,…7	3	3
-15,…,-8,8,…,15	4	4
-31,…-16,16,…,31	5	5
-63,…,-32,32,…,63	6	6
-127,…,-64,64,…,127	7	7
-255,…,-128,128,…,255	8	8
-511,…,-256,256,…511	9	9
-1023,…,-512,512,…1023	A	A
-2047,…,-1024,1024,…,2047	B	B
-4095,…,-2048,2048,…,4095	C	C
-8191,…,-4096,4096,…,8191	D	D
-16383,…,-8192,8192,…,-16383	E	E
-32767,…,-16384,16384,…,32767	F	N/A

bits of the Differenced DC coefficient. If the Differenced DC Coefficient is negative, a binary 1 is subtracted from it and the resulting least significant bits are taken to complete the code. If, on the other hand, the Differenced DC Coefficient is positive, it is simply transformed to binary, and the least significant bits are taken to complete the code. AC Coefficients are coded using the same way, but a different default AC coding table is used in this case and the run of the Zero Coefficients are taken into consideration [2]. As the Huffman coding is uniquely decodable, the resulting binary stream of Baseline JPEG Encoding Mechanism can be decoded without creating any confusion. The above mentioned steps are performed in a reverse manner for decoding, and, finally, an approximation of the original raw image is produced. Since the quantization steps round up the fractional parts of the pixel values, information is lost that can never be recovered. Hence the process is called lossy.

TABLE 2: THE JPEG DEFAULT DC CODING TABLE

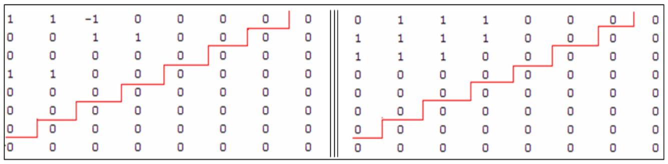
Category	Base Code	Length
0	010	3
1	011	4
2	100	5
3	00	5
4	101	7
5	110	8
6	1110	10
7	11110	12
8	111110	14
9	1111110	16
A	11111110	18
B	111111110	20

3 COMPUTATIONAL COST ANALYSIS OF BASELINE JPEG

As described in section 2, the quantization step of the Baseline JPEG produces a quantized two dimensional image signal, where most of the pixel values are zero. If we have a careful look at the default Baseline JPEG Quantization Matrix, we can say that the Resulting Quantized Matrix will have less possibility of having Non-Zero Coefficients in the lower triangular portion as the Quantization Matrix itself contains higher degree values in lower triangular portion and the Discrete Cosine Transform itself produces lower values in lower triangular portion of the image block. Practically, almost no Non-Zero Coefficient is produced in the lower triangular portion of the quantized matrix. Thus, dividing the lower triangular portion of the image block by quantization matrix is simply wastage of time. For the modern high capacity processors, this division may seem to be trivial; however, for the low capacity processors of small computing devices, it is a huge operation. This is because the division operation generally takes more time as compared to other arithmetic operations since the complexity of two n digit numbers' division is $O(n^2)$ [19]. Figure 4 illustrates some practical rounded DCT quantized blocks; it is clear from these example blocks that the lower triangular portions should almost always produce zero coefficients.

4. Complexity Reduction by Modified Quantization

Since the Quantization Step of Baseline JPEG suffers from some overhead computation of complexity $O(n^2)$, we took this step into our prime consideration. Each division of the lower triangular coefficients of the transformed image block must deal with the complexity $O(n^2)$ even when it is known that they will result in zero. However, we can keep the Quantization Step away from dividing the lower triangular coefficients and simply fill those coefficients with zero so that computational complexity $O(n^2)$ is reduced to linear O(n). The filling up of lower triangular coefficients with zero can be carried out by a simple AND operation which is much lower costly than the division operation. Hence, we quantize only the upper triangular coefficients and reduce the computational cost thereby. Figure 5 shows which portion of an image block should merely be quantized by the proposed modification.



1	1	-1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0
1	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
0	0	0	0	0	0	0	0

0	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	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	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

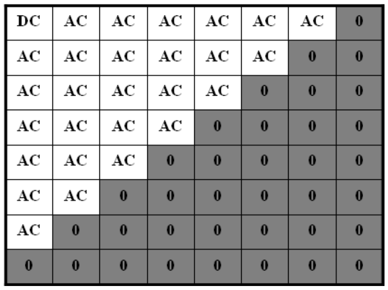
Fig. 4. Two Rounded DCT Quantized Blocks

5 PERFORMANCE ANALYSIS

We tested the proposed Reduced Cost Quantization upon 100 test images in a Simulated Less Capable Processor Computing Environment. As the major improvement suggested by this paper is computational time reduction, we computed the time required by the proposed improved quantization technique and by Baseline JPEG Algorithm. Table 3 shows the portion of our study for 10 different images. However, PSNR and compression ratio have been found more or less unchanged which means the amount of data loss introduced due to Reduced Cost Quantization is ignorable.

6 CONCLUSION

This paper presented a modified approach for Baseline JPEG Algorithm with a view to reducing its redundant computational



DC	AC	AC	AC	AC	AC	AC	0
AC	AC	AC	AC	AC	AC	0	0
AC	AC	AC	AC	AC	0	0	0
AC	AC	AC	AC	0	0	0	0
AC	AC	AC	0	0	0	0	0
AC	AC	0	0	0	0	0	0
AC	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 5. Only Upper Triangular Coefficients are Quantized

cost caused by the standard quantization process. Describing the popular Baseline JPEG Algorithm first, it took into consideration its Quantization Step as this process can be reorganized to minimize the Computational Complexity. The paper then proposed a model for Reduced Cost Quantization and proved the efficiency of the proposed method through proper discussion of the experimental results. The performance analysis finally showed that the proposed method can efficiently be used in case of less capable processors of small computing devices without considerable distortion of the resulting images.

TABLE 3: COMPARATIVE PERFORMANCE ANALYSIS

Image Name	**Computational Time**		**PSNR**		**Compression Ratio**	
	Reduced Cost Quantization	**Baseline JPEG**	**Reduced Cost Quantization**	**Baseline JPEG**	**Reduced Cost Quantization**	**Baseline JPEG**
Lena	0.75 sec	0.90 sec	34.10	33.20	11.29374	12.47572
Cameraman	0.60 sec	0.88 sec	29.86	29.02	19.118364	20.004120
Actress	0.67 sec	0.70 sec	33.24	33.14	20.40342	20.91254
Barbara	0.75 sec	0.80 sec	31.00	31.00	31.98463	31.98463
Baboon	0.86 sec	0.98 sec	32.32	32.26	19.187302	19.884820
Iris	1.23 sec	1.50 sec	30.76	30.76	21.74338	21.74338
Bridge	0.90 sec	1.14 sec	30.10	31.00	19.001889	20.632644
Tank	1.20 sec	1.45 sec	32.10	32.10	18.319325	18.319325
Airplane	0.54 sec	0.86 sec	29.89	29.14	10.66667	11.26479
Cowboy	0.57 sec	0.68 sec	33.87	33.87	16.209833	16.209833

REFERENCES

[1] Ralf Steinmetz and Klara Nahrstedt, “Multimedia: Computing, Communications and Applications”, 1st Edition, Pearson Education Inc. ISBN: 81-7808-319-1, 2005.

[2] Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, 2nd Edition, Pearson Prentice Hall. ISBN: 81-7758-168-6, 2005.

[3] Tinku Acharya and Ajoy K. Ray, “Digital Image Processing: Principles and Applications”, John Wiley & Sons, Inc. ISBN: 10 0-471-71998-6, 2005.

[4] Gregory K. Wallace, “The JPEG Still Picture Compression Standard”, *IEEE Transactions on Consumer Electronics*, 1991.

[5] Pennebaker W B and Mitchell J L, “JPEG still image data compression standard”, *Van Nostrand Reinhold*, 1993.

[6] Wikipedia: The JPEG Codec Example. http://en.wikipedia.org/wiki/JPEG# JPEG codec example.

[7] Bauschke H.H., Hamilton C.H., Macklem M.S., McMichael J.S. and Swart N.R., “Recompression of JPEG Images by Requantization”, *IEEE Transactions on Image Processing*, vol. 12, no. 7, July 2003.

[8] Takezawa M., Sanada H. and Watanabe K., “Quality Improvement Technique for JPEG Images with Fractal Image Coding”, *IEEE*, 2005.

[9] Richter, T., “Visual quality improvement techniques of HDPhoto/JPEG-XR”, *15th IEEE International Conference on Image Processing*, 2008.

[10] Gunawan I.P. and Halim A, “Haar wavelet decomposition based blockiness detector and picture quality assessment method for JPEG images”, *International Conference on Advanced Computer Science and Information System (ICACSIS)*, 2011.

[11] Zhou W., Sheikh H.R. and Bovik A.C., “No-reference perceptual quality assessment of JPEG compressed images”, *IEEE International Conference on Image Processing*, 2002.

[12] Altous S., Samee M.K. and Gotze, J., “Reduced reference image quality assessment for JPEG distortion”. *ELMAR Proceedings*, 2011.

[13] Gastaldo P. and Zunino R., “No-reference quality assessment of JPEG images by using CBP neural networks”, *International Symposium on Circuits And Systems (ISCAS)*, 2004.

[14] Stirner M. and Seelmann G., “Improved Redundancy Reduction for JPEG Files”, *Picture Coding Symposium by EURASIP*, 2007. ISBN: 978-989-8109-05-7.

[15] Bauermann I. and Steinbach E., “Further Lossless Compression of JPEG Images”, *Proc. Of Picture Coding Symposium, San Fransisco, USA*, Dec 15-17, 2004.

[16] Matsuda I., Nomoto Y., Wakabayashi K. and Itoh S., “Lossless Re-Encoding of JPEG Images using Block-Adaptive Intra Prediction”. *16th European Signal Processing Conference (EUSIPCO 2008)*, Lausanne, Switzerland, August 25-29, 2008.

[17] Zhong-Hua Z. and Wen-Yan W., “A lossless compression method of JPEG file based on shuffle algorithm”, *2nd International Conference on Advanced Computer Control (ICACC)*, 2010.

[18] Golner M.A., Mikhael W.B., Krishnan V. and Ramaswamy A., “Region based variable quantization for JPEG image compression”, *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, 2000.

[19] http://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations