International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 2 Issue: 6
ISSN: 2321-8169
1669 – 1672

# Improved Harmony Search Algorithm For Color Image Compression

Sarang Bansod
Dept. of Computer Science & Engg.
RCOEM
Nagpur, India
*e-mail: sarangbansod5@gmail.com*

Sweta Jain
Dept. of Computer Science & Engg.
RCOEM
Nagpur, India

***Abstract***— Harmony search algorithm can be used efficiently for solving various problems but it is relatively new, especially to field of image compression. This paper tries to improve its efficiency when it is applied for compressing various images. Improved result can be obtained by combining it to variants of BTC (Block Truncation Coding). Implementation results also shows feasibility of this newly generated algorithm.

***Keywords***- *image compression; lossy compression; PSNR; HSA; BTC; meta heuristic; quantization.*

*****

## I. Introduction

Digital images today are widely used in numerous fields. As their size increases, storage may become an important issue. Storing images in less memory leads to a direct reduction in storage cost and faster data transmissions. Mostly image contains redundant data which can be removed for storage and replaced for recovery. It can be changed in many ways that are either not detectable by the human eye or do not contribute to the degradation of the image. Thus Compression algorithms which are categorized as lossy or lossless can be used depending upon application.

For e.g. in medical diagnosis and some scientific measurements, we may need the reconstructed image exactly same as the original image. This type of compression is referred to as lossless compression. In application, such as capturing images using digicam, images on web applications, motion picture, television (TV),etc. a certain amount of information loss is allowed. This type of compression is called lossy compression [1].

This paper is based on lossy image compression techniques. The goal is to represent image with reduced bit rate , while the quality of the reconstructed image satisfies a requirement for a certain application and the complexity of computation involved is affordable for the application.

Here we first discuss implementation of harmony search algorithm for compressing images along with its limitations. Next we explore various BTC algorithms along with its core concept. And finally proposed method is discussed with implementation results.

## II. HSA for Image Compression

Finding solutions of problem using HSA is analogous to finding harmony in music by musicians. This music inspired algorithm was first developed by geem [2] in 2001. Since then it has been applied successfully to solve various problems in different fields including robotics, visual tracking, web text mining etc.[3]. Hsa was first implemented to compress images by Daga and Yusiong in 2012 [4].

Image has to be preprocessed before applying HSA to it. This preprocessing involves separation of RGB components of image and variance of each component is then calculated to select component with least variance. This component is then further processed by HSA.

In HSA, size of HM (Harmony Memory) is first calculated using size of selected component. Each element in harmony is initialized by value ranging from 0 to 255. For selecting harmony HMCR (Harmony Memory Consideration Rate) is used. HMCR indicate whether to select harmony from HM or it should be randomly selected. This facility makes algorithm meta heuristic. Harmony once selected may undergo through PAR (Pitch Adjustment Rate) which modifies it before getting processed. Fig. 1 shows the idea [5].

Elements in harmony will represent 2x2 pixel block of selected component in substitution process. Selected component is optimized keeping other two components as it is. It's combined with other two components to get new image. PSNR is used to calculate the quality of this newly generated image. If this harmony produces better quality image than the worst harmony in harmony memory (HM), it will get replaced by that worst harmony in memory. This process is repeated till the termination criteria i.e. MI (Maximum Improvisation) is satisfied. The best harmony having new RGB values for the selected component is then used to produce compressed image [4].

Algorithm requires harmony memory to be initialized first with random values and operates on small block (2x2 pixels) of image to improvise quality through iterations. But, if image size is large, for both memory initialization and manipulation on small blocks, time required and quality of image will be important issues of concern.

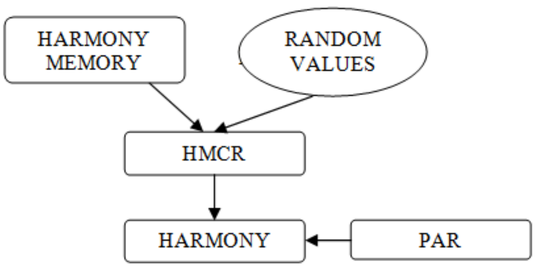



Figure 1. Harmony formation using HMCR and PAR

This algorithm works on very small pixel block (2x2 pixels) of images. Initializing memory with random values for such small block and its manipulation may require considerable time specially when image size is large. Also the amount of randomness it uses for compressing images may also affect its quality.

## III. BTC ALGORITHMS FOR IMAGE COMPRESSION

As the name indicate Block truncation coding (BTC) works on blocks of image. It's simple and fast lossy compression technique developed by delp and Mithchell [6] for grayscale images. Since then, many techniques with modifications to BTC have been proposed.

In basic form, BTC divides the whole image into small blocks and then try to reduce the number of gray levels within each block. This is done using quantizers or statistical moments. Standard BTC uses two level quantizers which are based on mean and standard deviation of the block. While compressing the image, bit plane is constructed such that, when pixel value in the block is greater than or equal to mean, it is represented by 1 and if it's less, it is represented by 0 in bit plane. This allows single bit to represent a pixel. Compressed image then can be stored or transmitted as a set consisting of bit plane and two quantizers, rather than a bitmap. Decompression method then uses this set to reconstruct image containing reduced number of level in each block.

Absolute Moment Block Truncation Coding (AMBTC) uses lower and higher mean as two statistical moments and doesn't involve square toot calculation (std. deviation), in order to reduce time required for coding and decoding process [7]. Iterative process of AMBTC is Minimum Mean Square Error (MMSE) which reduces MSE value. Here the threshold value, which is initialized by average of minimum and maximum values of block, is optimized using iterations. Process terminates when threshold values of consecutive iteration converges [8]. Multi- level quantizers can also be used to encode and decode blocks. Adaptive Block Truncation Coding (ABTC) uses multiple quantizers depending on nature of blocks [9]. Improved Adaptive Block Truncation Coding (IABTC) is based on ABTC and is used in further reducing the bit rate and to improve image quality [10].

A single-bit-map BTC (SBBTC) method, developed by Wu and Coll [11], exploits the inter-color correlation by using a single bit map to quantize all three color planes. Here two color vector and a bit pattern are needed to reconstruct the block. Kurita and Otsu [12] suggested another SBBTC method which generates the single bit-map by employing an adaptive one-bit vector quantization based on the principle score of each pixel in the block. The SBBTC method proposed by Tai et al. [13] uses Hopfield Neural Network (HNN) to generate the single-bit-map of a block. Yang et al. [14] suggests another single bit map method, called CICMPBTC method, based on moment preserving principle. Various other methods for color image compression using BTC are proposed [14, 15].

## IV. THE PROPOSED APPROACH

HSA for image compression [4] requires harmony memory to be initialized first with random values for selected RGB component and operates on small block (2x2 pixels) of image. For manipulation of large images time require to perform this operation increases. To minimize this, instead of initializing harmony memory with random values, we can initialize it with mean value of the block. Also, we can increase the size of block on which it operates to 4x4, 8x8 and so on.

Also before compression, Image is preprocessed which involves separation of RGB components of image and variance of each component is then calculated to select component with least variance. This component is then further processed. Here we can increase the compression ratio by increasing the number of components on which to operate.

Size of HM (Harmony Memory) is first calculated using one of the components of image. Components are then optimized by operating on its fixed sized blocks. In this we can use the concept provided by variants of BTC. Here we make use of Improved Adaptive Block Truncation Coding (IABTC) [13] that uses multi level quantizers that adapt according to blocks. These blocks are

- Low activity blocks - Contains visually same color area where pixel values are approximately same.
- Medium activity blocks - There is small transitions between pixels but do not represent any contrasting edges, and
- High activity blocks - Contain big pixel value changes with contrasting edge.

These blocks are categorized based on their absolute sum value (S).

$$S = \sum_{i=1}^{m} abs(x_i - \mu) \quad (1)$$

Where (μ) is mean value of the block.

Thresholds t1 and t2 are then used to identify the blocks as, low detailed block, if S <= t1, medium detailed block, if S > t1 and S <= t2, high detailed block, if S > t2. Generally t1 and t2 are 50 and 170, but are adjustable.

While compressing the image component, bit plane is constructed that stores number of quantizers for respective blocks. Compressed image then can be stored or transmitted as a set consisting of bit plane and quantizers, rather than a bitmap. Due to this facility, this algorithm can also be used in real time transmission of images.

### A. *Low activity block*

In this block there is very little variation in color. Pixel values of these blocks are approximately same, see fig 2. Low activity blocks are coded by using values form the harmony memory. This allows single value to represent the entire block.

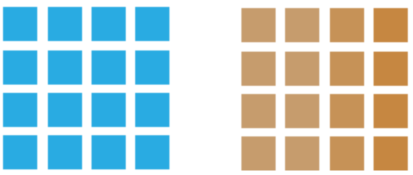


Figure 2. Low activity blocks

### B. *Medium activity block*

Medium activity blocks can be coded using two quantizers. In medium activity block, when pixel value in the block is greater than or equal to mean, it is represented by 1 and if it's less, it is represented by 0 in bit plane. Two statistical moments a (lower mean) and b (higher mean) calculated using the

ISSN: 2321-8169
1669 – 1672

equations (2) and (3) respectively, are preserved along with the bit plane which are used while decompressing image.

$$a = \frac{1}{p} \sum_{x_i < \mu} x_i \tag{2}$$

$$b = \frac{1}{q} \sum_{x_i \geq \mu} x_i \tag{3}$$

Where p and q is the number of 0s and 1s respectively in the compressed bit plane and μ is mean of the block.

### C. High activity block

For coding high activity block four quantizers are used. Two values are the minimum and the maximum values of the block. Other two values are randomly chosen based on the mid value (4) which is average of minimum and maximum value of block.

$$Mid = \frac{maxValue + minValue}{2} \tag{4}$$

First random value is selected from minimum and the average or mid value of block. Thus minimum and the average value provide the range from which the first random value gets selected. The range for second random value is from maximum and mid value of block (5).

$$\begin{aligned} rValue1 &= random(minValue; mid) \\ rValue2 &= random(mid; maxValue) \end{aligned} \tag{5}$$

To encode this block we thus require four values, If the pixel value is equals to minimum or the maximum value of block then 0 and 1 is stored in bitplane, 2 can be stored for first random value and 3 can be stored for second random value. Four quantizers which are preserved along the bitplane then replaces these values in decompression process. This algorithm uses some amount of randomness to provide the solution preserving meta heuristic property of HSA.

Fig. 3 shows an example of encoding high activity block. Mean value (μ) of block is 51.38. Absolute sum value (S) is 295.5. Mid is (30+78)/2=54.first random value (rValue1) is found as 45 and second random value is 68. Block decoded using these four values is shown in fig 3(b). Decoded block thus consist of four levels.

| | | | |
|---|---|---|---|
| 30 | 30 | 40 | 40 |
| 30 | 40 | 48 | 78 |
| 30 | 48 | 72 | 78 |
| 30 | 72 | 78 | 78 |

(a)

| | | | |
|---|---|---|---|
| 30 | 30 | 45 | 45 |
| 30 | 45 | 45 | 78 |
| 30 | 45 | 68 | 78 |
| 30 | 68 | 78 | 78 |

(b)

Mid = (30+78)/2 = 54

rValue1 = random(30; 54) = 45

rValue2 = random(54;78) = 68

Figure 3. Encoding and decoding of high activity block

Quality of the decompressed image is then evaluated using PSNR.

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{6}$$

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{7}$$

Where MAX is equal to 255 i.e. the maximum value a pixel can have. To calculate PSNR, Mean Square Error loss (MSE) is calculated over block of size m x n pixel. Parameters like Compression Ratio (CR) and Bit Per Pixel (Bpp) are also used for evaluation of final image.

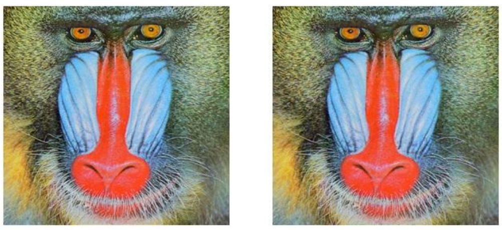
## V. EXPERIMENTAL RESULTS

Algorithm is implemented with harmony memory (HM) initialized with mean values of block. Block are categorized based on its S value into three categories viz. low, medium and high. These blocks are encoded as explained earlier. Image can be stored or transmitted inform of set containing bitplane and preserved values. Decompression algorithm operates on this set to reconstruct the image.

Algorithm is implemented on all the RGB components of image. Following is the result of implementation on various images.

Figure 4. Lenna (a)original & (b)decompressed

Figure 5. Peppers (a)original & (b)decompressed



ISSN: 2321-8169
1669 – 1672

Figure 6. Mandrill (a)original & (b)decompressed

Figure 7. Airplane(F-16) (a)original & (b)decompressed

This technique is also compared with existing techniques. Table.1 shows detail description and comparison with original HSA for image compression and one of the existing BTC technique called Absolute Moment Block Truncation Coding (AMBTC). Results shows improvisation over original HSA and also size of decompressed image in case of AMBTC and Improved HSA is nearly same but quality of image (PSNR) in case of improved HSA is much better.

## VI. Conclusion

We have mainly discussed Harmony search and Block truncation coding techniques for image compression. Many algorithms are developed by performing some variation on basic idea to provide better result. HSA is relatively new, especially to field of image compression. There's only one paper [4] that describe how it is used to solve problem of image compression.

The technique we studied provides the improved performance over the original algorithm for image compression. Experimental results provided by implementation of this algorithm and its comparison with existing techniques show its feasibility.

## References


[1] Y. Q. Shi, H. Sun. "Image and video compression for multimedia engineering", Taylor & Francis Group, 2008.

[2] Z.W. Geem, "Music-Inspired Harmony search algorithm, theory and application", Studies on Computational Intelligence, vol.191, 2009, pp. 2-13.

[3] Z.W. Geem, "Recent advances in Harmony search algorithm. studies on computational intelligence", vol. 270, 2009.

[4] R. R. M. Daga, J. P. T. Yusiong, "Image compression using harmony search algorithm", IJCSI International Journal of Computer Science Issues, vol. 9, September 2012.

[5] S. Bansod, S. Jain, "Recent image compression algorithms: A survey", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol. 2 Issue 12, December 2013.

[6] E. Delp, O.Mitchell, "Image compression using block truncation coding", IEEE Transactions on Communication, vol. COM-27, pp. 1335-1342, Sept 1979.

[7] D. Maximo, O. Mitchell, "Absolute moment block truncation coding and its application to color images", IEEE Transactions on Communications, vol. COM-32, October 1984.

[8] M. Shandilya, R. Shandilya, "Implementation of absolute moment block truncation coding scheme based on mean square error criterion", SDR 03 Technical Conference and Product Exposition..

[9] L. Hui,"An adaptive block truncation coding algorithm for image compression", IEEE, pp. 2233-2235, 1990.

[10] S. Vimala, P. Uma, B. Abidha, "Improved adaptive block truncation coding for image compression", International Journal of Computer Applications (0975 8887), vol. 19, April 2011.

[11] H Y. Wu, D. Coll, "Single bit map block truncation coding for colo rimage", IEEE Trans, pp. 352-356, 1987.

[12] T. Kurtia, N. Otsu, "A method of block truncation coding for color image compression", IEEE Trans., pp. 1270-1274, 1993.

[13] S. Tai, Y. Lin, J. Lin, "Single bitmap block truncation coding of color image using a hop_eld neural network", Inform. Sci., pp. 211-218, 1997.

[14] C. Yang, J. Lin, W. Tsai, "Color image compression by moment preserving and block truncation coding techniques", IEEE Trans., pp. 1513-1516, 1997.

[15] B. Dhara, B. Chanda, "Color image compression based on block truncation coding using pattern _tting principle", Pattern Recognition, 2007.


TABLE I. Improved HSA and Other Techniques

| Technique | Image | Original Size (KB) | Decompressed Size(KB) | CR | Bpp | PSNR |
|---|---|---|---|---|---|---|
| HSA | Lenna | 462.726 | 73.1563 | 0.158099 | 2.28613 | 33.1538 |
| Improved HSA | | | 35.334 | 0.0763606 | 1.10419 | 39.1032 |
| AMBTC | | | 35.3488 | 0.0768249 | 1.1109 | 37.6968 |
| HSA | Peppers | 768.137 | 74.9961 | 0.0976338 | 2.34363 | 35.4056 |
| Improved HSA | | | 38.6377 | 0.0503005 | 1.20743 | 38.7717 |
| AMBTC | | | 38.7555 | 0.0499331 | 1.19861 | 37.5877 |
| HSA | Mandrill | 768.137 | 138.456 | 0.180249 | 4.32675 | 32.1909 |
| Improved HSA | | | 74.3359 | 0.0967744 | 2.323 | 33.8156 |
| AMBTC | | | 74.1777 | 0.0926628 | 2.2243 | 31.9839 |
| HSA | Airplane | 768.137 | 74.4531 | 0.0969269 | 2.32666 | 30.3782 |
| Improved HSA | | | 35.8486 | 0.0466696 | 1.12027 | 38.8044 |
| AMBTC | | | 37.8835 | 0.0482931 | 1.15924 | 37.8835 |