Improved Adaptive Block Truncation Coding for Image Compression in LCD Overdrive

Poonam A. Arbat, P.D. Gawande

Abstract-Block truncation coding (BTC) technique has attracted much attention during the last few years as a simple source coding method to achieve good quality image reproduction by preserving the first two statistical moments of an image: the mean and the variance. It significantly reduces the amount of computation required by the traditional coding methods, such as transform. In this paper, we have proposed a method called the Improved Adaptive Block Truncation Coding (IABTC) based on Adaptive Block Truncation Coding (ABTC).And we are implemented it on LCD overdrive for image compression. The feature of inter-pixel redundancy is exploited to reduce the bit-rate further by retaining the quality of the reconstructed images. The proposed method outperforms the existing BTC based methods both in terms of bit-rate and PSNR values.

Keywords: Compression, BPP, Bit plane, PSNR

I. INTRODUCTION

With the advanced development in internet, teleconferencing, multimedia and high-definition television technologies, the amount of information that is handled by computers has grown exponentially over the past decades. Hence, storage and transmission of the digital image component of multimedia systems is a major problem.

The amount of data required to present images at an acceptable level of quality is extremely large. High quality image data requires large amounts of storage space and transmission bandwidth, something which the current technology is unable to handle technically and economically.

One of the possible solutions to this problem is to compress the information so that the storage space and transmission time can be reduced and the main goal for image data compression is to reduce redundancy in the image block as much as possible.

BTC has the advantage of being easy to implement. BTC achieves 2 bits per pixel (bpp) with low computational complexity. In this paper, we extending the basic block truncation coding method and we have presented a method named Improved Adaptive Block Truncation Coding (IABTC) which gives a better result than existing BTC methods.

Manuscript received September, 2013.

Poonam A. Arbat Department of Electronics & Telecommunication, Sipna C.O.E.T, Amravati, Maharashtra, India.

Prof. P.D. Gawande Department of Electronics & Telecommunication, Sipna C.O.E.T, Amravati, Maharashtra, India.

II. LITERATURE REVIEW

To improve the transmission rate and storage space the compression of image is very necessary. For compression of image many tools and methods are applied. The basic block truncation coding (BTC) was proposed by Delphi and Mitchell [4] which is easy to implement. The bit rate obtained by BTC is 2 bpp but the quality of reconstructed image is very low. The quality of reconstructed image is improved by extending the basic block truncation coding as ABTC which is proposed by Lema and Mitchell [1] also AMBTC which is proposed by K.Somasundaram and Ms.S.Vimala [2] These methods gives better PSNR which measures the quality of image but not gives better bit rate. So we in this paper we proposed a method Improved Adaptive Block Truncation Coding (IABTC) which gives better PSNR with lower bit rate. we used this method on LCD overdrive for image compression and got effective results in terms of BPP and PSNR.

III. PROPOSED WORK

Improved Adaptive Block Truncation Coding which is our proposed method is the lossy image compression method. [1] It is based on Adaptive Block Truncation Coding. In this we preserved the statistical moments: means and variance along with the bit plane for reconstructing the image. The compressed image is transmitted or stored as a set of bit plane B and statistical moments.

Firstly we calculate the mean and variance by using following formulas as

$$\bar{X} = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad (1)$$

Where m is the number of pixels in each block, and Xi is the original pixel value of the block.

$$\sigma = \sqrt{\frac{\sum(y_i - x_i)^2}{m}} \qquad (2)$$

All the input image blocks are categorized into three groups based on sum value (S), which is calculated using the equation (3)

$$S = \sum_{i=1}^{m} \mathrm{abs}(x_i - \overline{X}) \quad (3)$$

The blocks are categorized into three groups as follows:
1. Low detailed block, if S <= t1,
2. Medium detailed block, is S>t1and S<=t2 (4)
3. High detailed block, if S > t2 and S <= t3

The threshold values t1, t2 and t3 are 50,170 and 256 respectively.

$$thr = min + ((max-min)r/n) \qquad (5)$$

Where, thr represents the rth value of threshold and n is the number of quantization levels and the four quantizing levels *a, b, c* and *d* are computed using the equation (6). *min* and *max* are the minimum and maximum intensities of the block respectively.

a = min.
b= (2min+max)/3.
c = (min+2max)/3. (6)
d = max.

While decompression, the pixels in the low detailed blocks are reconstructed using the mean of the block. The pixels in the medium detailed blocks are reconstructed using a two level quantizer with the two statistical moments being Min and Max which are optimized as in MMSE and the pixels in the high detailed blocks are reconstructed using the four quantizing levels. The compression of image is done in two levels in IABTC:

i)By making slight modifications to ABTC and
ii) Dividing the statistical moments by 4.

A. *First level compression*

Step1: Input the image size n x n pixels
Step2: Divide the image into N blocks.
Step3: Compute the mean for each block using the equation (1).
Step4: Compute S for each block using the equation (3)
Step5: Categorize the block (Low or Medium or Detailed block) using the equations (4).
Step6: Store the set {Bit plane B, statistical moments}

B. *Second level compression*

In the second stage, all the statistical moments are divided by 4. Generally a statistical moment requires a maximum of 8 bits (log2256) to get stored. But, when divided by 4, it requires only a maximum of 6 bits (log264). While reconstructing the image, the statistical moments are multiplied back by 4 to get the approximate original values ranging from 1 to 256.

C. *Bit Rate calculations:*

While encoding the image, for a low detailed block, only the mean (8 bits) alone is stored. For a medium detailed block, the Bit plane (size 16 bits) and the two quantizing values Max (8 bits) and Min (8 bits) are stored. Hence it requires 32 bits to store a medium detailed block. In the bit plane B of medium detailed block, if a pixel's intensity value is greater than or equal to mean, it is coded as 1 otherwise 0. For a high detailed block, the bit plane B requires 32 bits, i.e. 2 bits per pixel. The bit plane is generated by coding the individual pixel as

00, if Xi <= t1
01, if Xi > t1 and <=t2
10, if Xi > t2 and <= t3 and
11, if Xi > t3

After the second level of compression, only 6 bits are required to store the quantizing levels. Hence the bpp of the image compressed using IABTC method is calculated as

n=n1*6+n2*(16+6+6) + n3*(32+6+6) (7)
bpp = n / (256*256) (8)

IV. RESULT AND COMPARISON WITH ABTC METHOD:

Before applying IABTC method to LCD overdrive we compare our results with ABTC method as in figure. 1.

As shown in figure. we got a good quality of reconstructed image as compare to ABTC method with PSNR of 22.52 "db".

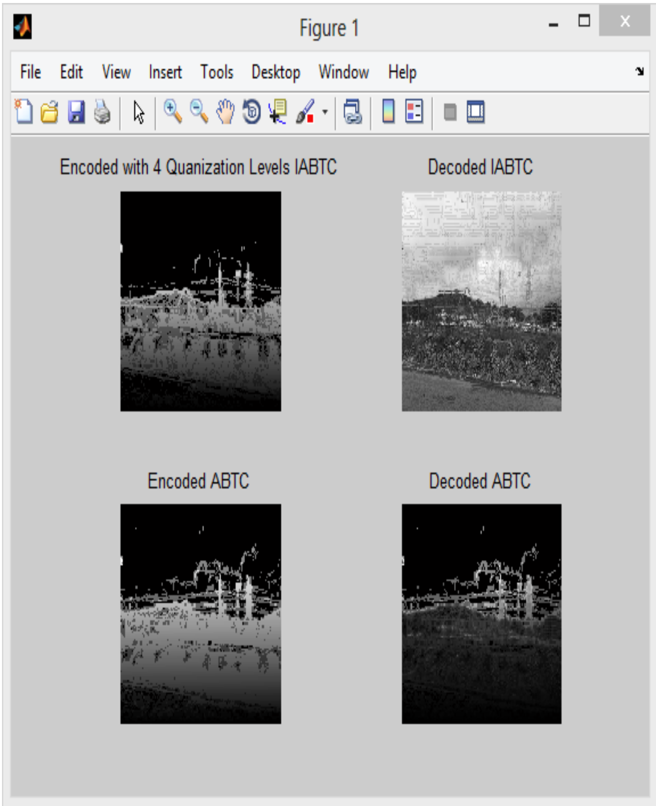


Figure 1. Comparison of IABTC & ABTC

We applied improved Adaptive block truncation coding to LCD overdrive as shown in figure 2. where the images get compressed and stored in a buffer and decompressed them when they needed Experiments were carried out with standard videos like VIP Traffic, xylophone etc. We can take a block size of 2,4,8. For following VIP Traffic video we had taken a block size of 4. We got a improved quality of reconstructed video with bit rate (bpp) of 0.25 and PSNR varying between 53"db" to 54"db" because of the frames in the video. The MMSE which gives the minimum error between original and decoded images is 0.01846 as shown in figure 3. It is observed that when we take a block size of 2, the reduction video is clearer than the block size of 4 & 8. The output of block size 4 is more clear than 8. The algorithm is implemented using MATLAB R 2011b on Windows Operating System.

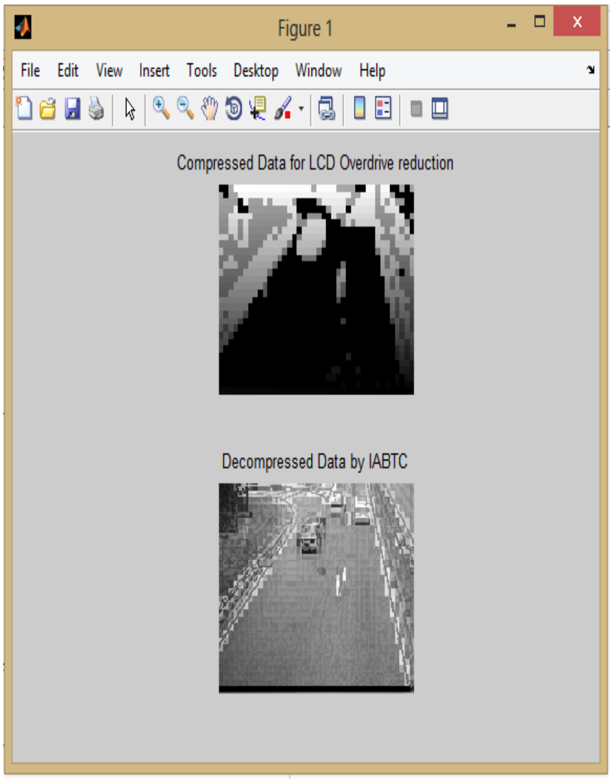
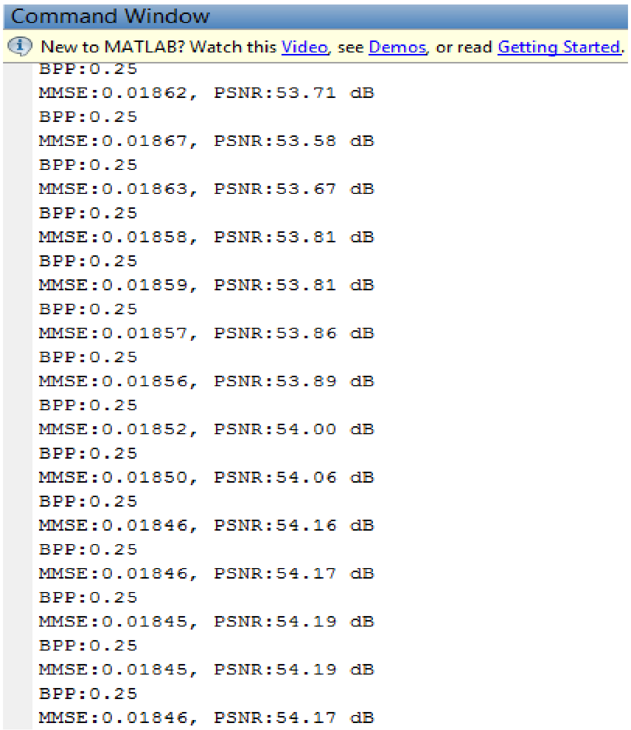


Figure 2: Result of LCD Overdrive.



Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
BPP:0.25
MMSE:0.01862, PSNR:53.71 dB
BPP:0.25
MMSE:0.01867, PSNR:53.58 dB
BPP:0.25
MMSE:0.01863, PSNR:53.67 dB
BPP:0.25
MMSE:0.01858, PSNR:53.81 dB
BPP:0.25
MMSE:0.01859, PSNR:53.81 dB
BPP:0.25
MMSE:0.01857, PSNR:53.86 dB
BPP:0.25
MMSE:0.01856, PSNR:53.89 dB
BPP:0.25
MMSE:0.01852, PSNR:54.00 dB
BPP:0.25
MMSE:0.01850, PSNR:54.06 dB
BPP:0.25
MMSE:0.01846, PSNR:54.16 dB
BPP:0.25
MMSE:0.01846, PSNR:54.17 dB
BPP:0.25
MMSE:0.01845, PSNR:54.19 dB
BPP:0.25
MMSE:0.01845, PSNR:54.19 dB
BPP:0.25
MMSE:0.01846, PSNR:54.17 dB
```

Figure 3: BPP, PSNR & MMSE for LCD overdrive.

V. CONCLUSION

The PSNR value is taken as a measure of the quality of the reconstructed image. The value of PSNR we got by applying IABTC is 54db. The bpp (bit per pixel) achieved is 0.25. Both in terms of bpp and PSNR, IABTC outperforms the other techniques.

ACKNOWLEDGMENT

I take this opportunity to express my sincere gratitude to Prof. Dr. S.A. LADHAKE principal Sipna College of engineering and technology, who have contributed in making the work towards great success. I express gratitude to my guide Prof. P.D. GAWANDE & M.E. coordinator DR. A.A. GURJAR, whose constant help and encouragement supports me to complete my work.

REFERENCES

1. S.Vimala, P. Uma, B. Abidha "Improved Adaptive block truncation coding for image compression" international journal of computer application (0975-8887) vol 19-No.7,April 2011.
2. Dr.K.Somasundara, Ms.S.Vimala "Multi-Level Coding Efficiency with Improved Quality for Image Compression based on AMBTC" International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.2, March 2012.
3. Jun Wang and Jong-Wha Chong, Member, IEEE "High Performance Overdrive Using Improved Motion Adaptive Codec in LCD" IEEE Transactions on Consumer Electronics, Vol. 55, No. 1, FEBRUARY 2009.s
4. Edward j. Delp, Robert Mitchell " Image compression using block truncation coding" IEEE transactions on communications, vol. com-27, no. 9, September 1979.
5. Jun Wang, Jong-Wha Chong "Adaptive Multi-level Block Truncation Coding for Frame Memory Reduction in LCD Overdrive" IEEE Transactions on Consumer Electronics, Vol. 56, No. 2, May 2010.
6. Yun-Ho Ko, Jin-Hyung Kim, Si-Woong Lee, and Hyun-Soo Kang "Dual Block Truncation Coding for Overdriving of Full HD LCD Driver" IEEE Transactions on Consumer Electronics, Vol. 58, No. 1, February 2012.
7. Doaa Mohammed, Fatma Abou-Chadi "Image compression using block truncation coding" Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), February Edition, 2011.
8.] Meftah M. Almrabet , Amer R. Zerek, Allaoua Chaoui , Ali A. Akash "Image compression using block truncation coding" IJ-STA, Volume 3, No 2, December 2009.
9. Maxim D. lema, Robert Mitchell "Absolute Moment Block Truncation Coding and Its Application to Color Images" IEEE transactions on communications, vol. com-32, no. 10, october 1984.
10. Jong-Woo Han, Min-Cheol Hwang, Seong-Gyun Kim, Tae-Ho You, Sung-Jea Ko" Vector Quantizer based Block Truncation Coding for Color Image Compression in LCD Overdrive" IEEE Transactions on Consumer Electronics, Vol. 54, No. 4, NOVEMBER 2008.