Segmentation and Reassembly of Images using Biplane Slicing in Adaptive Lossless Dictionary based Compression

Deepa Raj
Computer science Department
BBAU,Lucknow

Seema Gupta
Computer science Department
BBAU,Lucknow

ABSTRACT
A digital discrete signal corrsepond to a specific pointer is termed as bitplane image which represents position of bit in binary number.The lossless image compression is a combination of bitplane slicing and adaptive coding.This paper will discuss adaptive lossless image compression using bitplane slicing technique and derived the compression ratio.

Keywords
Image compression, Adaptive compression, bit plane slicing, compression ratio.

1. INTRODUCTION
A digital discrete signal[1] either image or sound produces a set of bits which corresponds with the specific bitpointer in binary number is termed as bitplane i.e. 8 bit data representation will produce 8 bitplanes,16 bit data representation will produce 16 bitplanes[6] and so on.

Table 1. 8 Bitplane Representation

BITPLANE	BINARY VALUE	CONTRIBUTION	TOTAL RUN
First	1	1*2^7=128	128
Second	0	0*2^6=0	128+0 =128
Third	1	1*2^5=32	128+32 =160
Fourth	1	1*2^4=16	160+16 =176
Fifth	0	0*2^3=0	176+0 =176
Sixth	1	1*2^2=4	176+4 =180
Seventh	0	0*2^1=0	180+0 =180
Eight	1	1*2^0=1	180+1 =181

Starting from most significant bit i.e first bitplane and Least significant bit to the last bitplane[2].First bitplane which contains the most significant bit gives the most near approximate values compare to last bitplane hence for better approximation value adding of a bitplane gives better results.Bitplane and bitmap are synonymous except that a bitplane refers to the data location in memory and bitmap denotes dataitself.Let us take the example of 8 bit plane for one decimal value suppose 181.The value of 181 can be treated as pixel value.Firstly,we convert into binary value i.e.10110101.It can be numbered from left to right.The contribution of first number is highest and contribution of last number is lowest.A bit-plane contains significant information or random information.One method for finding is, compare each pixel (m,n) to three neighbouring pixels (m-1,n), (m,n-1) and (m-1,n-1).If two pixel values out of three are same then it has no noise otherwise it has a noise. Approximation of noisy bit-plane will have 49% to 51% pixel.

A digital image[3] is represented in terms of pixels. These pixels can be expressed further in terms of bits. Given an X-bit per pixel image, slicing the image at different planes (bit-planes) plays an important role in image processing. An application of this technique is data compression. In general, 8-bit per pixel images are processed. We can slice an image into the following bit-planes. 0 is the least significant bit (LSB)[5] and 7 is the most significant bit (MSB).

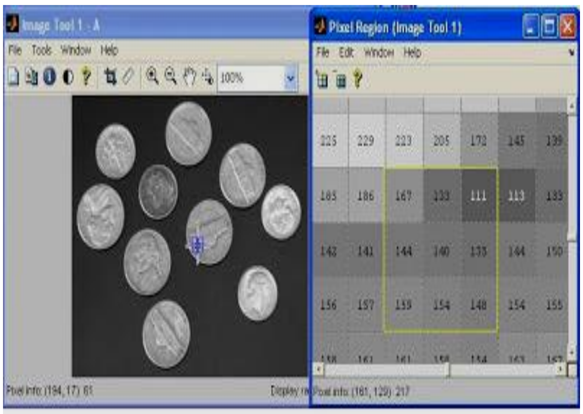

Fig 1: Take One Block(3*3) Of Images

Let us take one example as shown below is an 8-bit per pixel image and the different bit-planes after slicing.Martix can be converted into binary value:

Table 2. Convert from pixel to binary

10100111	**10000101**	**01101111**
10010000	**10001100**	**10000111**
10011111	**10011010**	**10010100**

This 8-bit image is composed of eight 1-bit planes. Plane 1 contains the lowest order bit of all the pixels in the image and shown in blue color,plane2 contains green color, plane3,plane4,plane5,plane6 and plane 7 contains different color . Plane 8 contains the highest order bit of all the pixels in the image and shown in red color.

Table 3. bitplane 8 i.e MSB

10100111	**10000101**	**01101111**
10010000	**10001100**	**10000111**
10011111	**10011010**	**10010100**

2. IMAGE SEGMENTATION

A= [167 133 111

144 140 135

159 154 148]

B=bit get (A, 1); %Lowest order bit of all pixels

'Bit get' is a MATLAB function used to fetch a bit from the specified position from all the pixels.

B= [1 1 1

0 0 1

1 0 0]

B=bit get (A, 8); %Highest order bit of all pixels

B= [1 1 0

1 1 1

1 1 1]

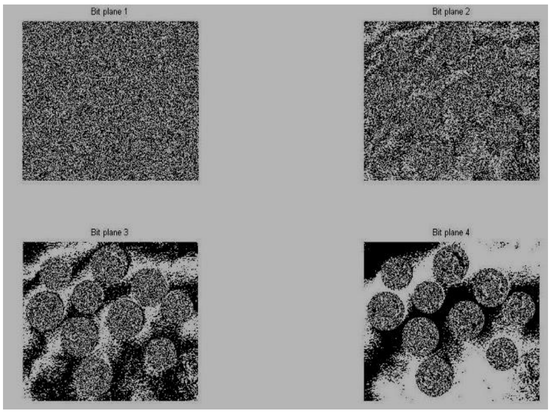

Fig 2: Construction using bitplane1,2,3,4

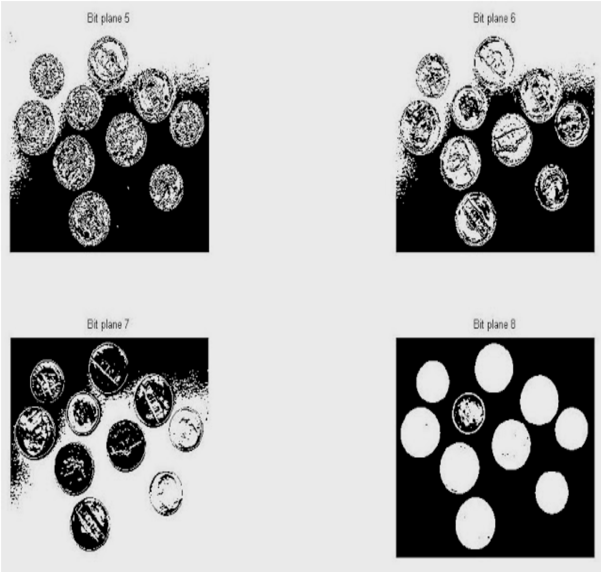

Fig 3: Construction of 5,6,7,8 bitplane

3. IMAGE RECONSTRUCTION USING N BITPLANES

The nth plane in the pixels are multiplied by the constant 2^n-1
For instance, consider the matrix

A=[167 133 111

144 140 135

159 154 148] and the respective bit format

Table 4. Convert from pixel to binary

10100111	**10000101**	**01101111**
10010000	**10001100**	**10000111**
10011111	**10011010**	**10010100**

3.1 Image reconstruction using 7,8 bitplane

For 10100111, multiply the 8 bit plane with 128 and 7 bit plane with 64.

(1x128)+(0x64)+(1x0)+(0x0)+(0x0)+(1x0)+(1x0)+(1x0)=128

Repeat this process for all the values in the matrix and the final result will be

[128 128 64

128 128 128

128 128 128]

'bitset' is used to set a bit at a specified position. Use 'bitget' to get the bit at the positions 7 and 8 from all the pixels in matrix A and use 'bitset' to set these bit values at the positions 7 and 8 in the matrix B.

3.2 Image reconstruction using 5,6,7,8 bitplane

For 10100111, multiply the 8 bit plane with 128 and 7 bit plane with 64.

(1x128)+(0x64)+(1x32)+(0x16)+(0x0)+(1x0)+(1x0)+(1x0)=160

Repeat this process for all the values in the matrix and the final result will be

[160 128 96

144 128 128

144 144 144]

Fig 4: Reconstruction of 5,6,7,8 bitplane

Image reconstructed using 8 and 7 bit planes

4. APPLICATIONS

PCM sound encoding the first bit in sample denotes the sign of the function, or in the other words defines the half of the whole amplitude values range, and the last bit defines the precise value. Replacement of more significant bits result in more distortion than replacement of less significant bits. In lossy media compression that uses bit-planes it gives more freedom to encode less significant bit-planes and it is more critical to preserve the more significant ones.

As illustrated in the image above, the early bitplanes, particularly the first, may have constant runs of bits, and thus can be efficiently encoded by run-length encoding[8]. This is done (in the transform domain) in the Progressive Graphics File image format, for instance.

Some computers displayed graphics in bitplane format most notably the Amiga and Atari ST, contrasting with the more common packed format. Many image processing packages can split an image into bit-planes. Open source tools such as Pamarith from Netpbm and Convert fromImage Magick can be used to generate bit 1 represents White and bit 0 represents Black symbol.

Table 5. A=P0+P1+P2+P3

1101	0111	1000
1100	1011	1001

A

W	W	B
B	W	W

P0

B	W	B
B	W	B

P1

W	W	B
W	B	B

P2

W	B	W
W	W	W

P3

Table illustrates how a 4,16 bit grayscale image[6] can be decomposed into 4 separate monochrome images by bitplane slicing.Image can be represented as 2*3 grayscale image.The LSB in images are P0 i.e.zeroth bitplane and The MSB in images p3 i.e.third bitplane.

5. EXPERIMENTAL RESULT

Mean square error and peak signal to noise ratio are two parameter of image compression using bitplane slicing technique.The word error in mean square error means distortion.

Formulas of mean square error and peak signal to noise ratio are given by:

$$MSE = 1/mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad \text{Equ- 1}$$

Where mn is the size of the image.

$$PSNR = 10\log_{10}[255 * 255/MSE] \quad \text{Equ- 2}$$

Table 6. Result of MSE &PSNR

BITPLANE	IMAGES	MSE	PSNR
Bitplane1	First image	40.5	5.78
Bitplane2	Second image	37.4	11.7
Bitplane3	Third image	30.9	51.6
Bitplane4	Fourth image	33.7	27.5
Bitplane5	Fifth image	33.4	29.3
Bitplane6	Sixth image	26.8	133.5
Bitplane7	Seventh image	27.48	116.01

Peak signal to noise ratio depends on the quality of images.Mean square error is inversely proportional to peak signal to noise ratio.

6. CONCLUSION

The 0th bit plane consists of the least significant bits of each sample. These bit planes contain little information and are sometimes considered to be 'noise'. The 7th bit plane consists of the most significant bits of each sample. These bits more accurately represent the structure of the grayscale source than the other planes.The 7th bit plane of an 8 bit grayscale image is equivalent to thresholding the source against 128.Compression ratio improves if there are more long runs than short runs.Consider two adjacent samples that have similar grayscale values of 127 and 128. In binary, these are given as 01111111 and 10000000. Each run on each bit plane is terminated by this small change in grayscale value.Can usually lengthen runs by using gray coding rather than binary-coded-decimal encoding. Gray codes ensure that two adjacent values differ by a single binary bit.Bitplane Slicing is very useful in lossless image compression.

7. REFERENCES

[1] Rafael Gonzalez and Richard E. Woods, Digital Image Processing, Addison-Wesley, second edition, 2002.

[2] H.Faheem Ahmed and U. Rizwan, Embedding Multiple Images in an Image Using Bit Plane, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 1, January 2013, ISSN: 2277 128X

[3] N S T Sai and R C Patil, Image Retrieval Using Bit-Plane Pixel Distribution, International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 3, June 2011

[4] OPERATIONS, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol. 4, No.3,june2014

[5] Mr. Vikas Tyagi, Data Hiding in Image using least significant bit with cryptography, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012, ISSN: 2277 128X

[6] Komal Ramteke and SunitaRawat, "Lossless Image Compression LOCO-R Algorithm for 16 bit Image", 2nd National Conference on Information and Communication Technology (NCICT), pp. 11-14, 2011.

[7] Syed Ali Hassan and Mehdi Hussain, "Spatial Domain Lossless Image Data Compression Method", International Conference of Information and Communication Technologies, 2011.

[8] Al-Wahaib and M. S. KokSheikh Wong, "A Lossless Image Compression Algorithm Using Duplication Run Length Coding",IEEE Conference on Network Application Protocols and Services, pp. 245-250, 2010.

[9] C. Saravanan and R. Ponalagusamy, "Lossless Grey-Scale Image Compression Using Source Symbol Reduction and Huffman Coding", International Journal of Image Processing, IJIP, Vol-3. Issue-5, pp. 246-251, 2009.

[10] J. Arz and J. Fischer, "LZ-Compressed string dictionaries," CORR 2013.