Spatial Compression of color images in an Event Image Set

Dr. Hanumanthappa. M
Associate Professor
Department of Computer Science and Applications
Bangalore University, Bangalore
hanu6572@hotmail.com

S. Regina LourdhuSuganthi
Research Scholar
Department of Computer Science and Applications
Bangalore University, Bangalore
reginalsuganthi@gmail.com

Abstract: Advent of handheld devices with cameras have created huge impact in the present time. These devices have generated enormous amount of Video and Image data. Governmental, Business and Academic institutes organize various events to keep abreast with the growing trends in their area of work. These events are documented for future reference. As a result the data is piled up in the storage media occupying considerable amount of storage space. Color images need to be efficiently compressed without losing potential information of the image content in order to reduce the storage space. The proposed method in this paper considers the spatial data of the image and converts the pixel information of R-Red, G-Green, B-Blue components into an integer value that require two bytes instead of the usual three bytes namely one byte for each color component. Thus the storage requirement for each image reduces by one-third of its original storage requirement.

Keywords :*BIT Plane, JPEG-XR, Wavelets, FHT, SIFT*

1. INTRODUCTION

Image and video information captured during various events organized by the Institutions are stored in electronic vaults. These data occupy considerably more space in contrast to text data. The image and video data stored in the servers need to be maintained as an archive for future reference. The image data is generally used for organizational documentation, printing newsletters, annual magazines and the like. In this view, the primary requirement of captured images is for printing purpose, thus a minimal loss in the color information would not effect and compromise on the print quality. Image compression is an important and integral part of image processing applications. Compression procedures generally involve color space conversion, frequency transformation and entropy encoding[5]. Visually lossless compression techniques attempt to compress images ensuring that the distortions are low to human perception.

The paper is organized as follows :
Section 2 presents the literature reviewon compression techniques proposed by various Researchers. Section 3 introduces the proposed framework. Section 4 presents a detailed discussion of the bit plane compression technique. Section 5 gives the experimental results and Section 6concludes with scope of the work.

2. LITERATURE REVIEW

The two classes of image compression techniques are lossy[1] and lossless compression[6]. Lossless compression algorithms reconstruct the original data precisely in contrast to lossy compression algorithms. Lossless compression is essential in applications such as medical imaging, technical drawing and many other such fields. Integer to Integer multi-wavelet transform[7] based on lifting scheme is used for lossless image compression. Integer to Integer multi-wavelets transform is based on multi-scaling and multi-wavelets function. The important part of multi-wavelet transform is lifting scheme where integer operation is used to achieve the expected transform.

Many of the image compression algorithms aim at reducing redundancy in order to reduce storage and facilitate efficient transmission. Image compression can be achieved by addressing redundancy[3] with respect to color component or by computing pixel difference in three different directions namely horizontal, vertical and diagonal directions or by reducing any form of comparative redundancy in any direction and not restricted to the three directions alone.

In the process of development of image compression techniques, to attain perpetual quality, visual redundancy is combined with statistical redundancy[4].

These procedures compress the image by using image features. Instead of pixel values, Scale Invariant Feature Transform (SIFT) descriptors[2] are used to compress the image. A down sampled image with key SIFT descriptors are generated as one component and the rest are simplified to form a differential set. The groups formed are compressed separately. During the process of reconstruction, the decoder uses the visual correlation through SIFT based matching.

3. FRAMEWORK

The proposed procedure is using spatial information of the image based on pixel values. An RGB image has a 3 byte representation for each pixel, one byte each for Red, Green, and Blue color components. The color values range from 0 - 255.

Step 1 :Bit planes of each color component of a pixel is derived and only five of the eight planes are accounted and the remaining three bit planes are ignored

Step 2 :The bit planes of the three color components are combined in octal form

Step 3 :The five digit octal value is translated to an integer value

Step 4 :The storage requirement after this encoding is 2 bytes per pixel. This require only two-third of the space of the original data

Step 5 :Reconstruct the image using the inverse process. Though this procedure losses bit depth values, the loss is very minimal and has no significant difference in the print quality.

The following figure (Figure 1) depicts the compression scheme

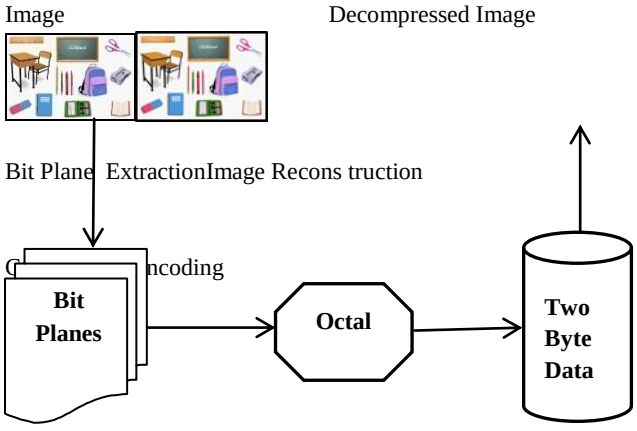

Figure 1 : Compression Scheme

4.PROCESS FLOW

4.1 Bit-Plane Extraction and Compression

The color component of each pixel is generally quantized to the range 0 – 255. For an RGB image, each pixel account for three bytes. Thus an RGB image of size 100 x 100 with three –eight bit depth require approximately 30 kilo bytes of memory. Extracting only the first five higher order bit planes of each color component, would yield bit strings of length five for each color value.

Let x be an image pixel with (R,G,B)=(r,g,b), where $r=r_8r_7r_6r_5r_4r_3r_2r_1$, $g=g_8g_7g_6g_5g_4g_3g_2g_1$ and $b=b_8b_7b_6b_5b_4b_3b_2b_1$. The bits extracted are : $r=r_8r_7r_6r_5r_4$, $g=g_8g_7g_6g_5g_4$ and$b=b_8b_7b_6b_5b_4$.It is evident that the maximum loss of brightness is just seven units for each color component, which is very low for Human Visual System and it has least significance in print quality.

The weighted sum of the extracted bit streams are computed to produce an octal value that is in the range 00000 – 77777. Thus the k^{th} octal digit is the result of

$x_k = 4 * r_j + 2 * g_j + b_j$, $j=8-k$

wherer_j, g_j and $b_j$are the j^{th} bit plane value of the Red, Green and Blue colors respectively.

4.2Encoding

This step converts the resulting octal stream to an integer value y, where y is the corresponding pixel of the compressed image. The range of values resulting from this operation is 0 – 32767 and hence require only two byte storage space for any pixel x. The image could also be represented as a single matrix of integer values as against three matrices representing each color component.

4.3Image Reconstruction

The compressed and encoded image can be reconstructed by reversing the process. The compressed image data stored in the matrix is in the range 0-32767 for each pixel.This data is first converted to a 5-digit octal number. Secondly, each digit of the octal number is transformed to a 3-bit number using modulo division. The resulting higher to lower order bits are associated with Red, Green and Blue bit data. Thirdly, the Bits corresponding to each of the three color components are combined in higher to lower order bit stream. This results in a five bit string. Finally, each bit string is converted to an integer value in the range 0-248. Thus the individual color components are arrived at with a minimal loss. Though, this procedure is a lossy compression technique, the human visual perception cannot differentiate this loss in the color values in print form.

5. EXPERIMENTAL RESULTS

Let x be a pixel with (R,G,B)=(252,129,55), then the bit-plane extraction of x would result in (11111,10000,00110)=(248,128,48).This is shownhere. For a pixel x with (R,G,B)=(252,129,55), the bit strings are :

R : 1 1 1 1 1
G : 1 0 0 0 0
B : 0 0 1 1 0

andthe compressed octal value using the weighted sum is 64554. This octal value is converted to an integer26988 say y. It is evident that any octal value from the extracted bit stream would be in the range 00000 to 77777 and the corresponding integer equivalent is in the range 0 – 32767. Thus the storage required for each pixel is just 2-bytes as against 3-bytes. The image is reconstructed by first converting the integer y = 26988 to an uniform five length octal string. Thus y is transformed to 64554 using modulo 8 division and the each digit from the octal string is converted to three bit form as :

6 : 110, 4 : 100, 5 : 101, 5 : 101, and 4 : 100.

In the next stage, the first bits of each of the 3-bit strings are combined to get the Red value : 11111, the second bits are combined to form Green value : 10000 and the third bits are combined to form the Blue value : 00110. The bit streams are added with '000' as lower order bits. Thus we get the (R,G,B) of the pixel y in the reconstructed image as (248, 128, 48).

Image Processing Toolbox of Matlab software has been used. Figure 2 is the uncompressed image of size 201 x 300.

Figure 2 : Original Image of size 201 x 300

The following table (Table 1) shows the weighted sums of the extracted bits from the R,G,B components for the portion of the image 45 x 8 to 55 x 14 and Table 2 is the corresponding integer values for the same portion of the image.

Each value of the image pixel is thus converted to a two byte integer value and hence the storage requirement is just the two-third of the original storage.

77777	77777	77777	77777	77777	77777	77777
77777	77777	77777	77777	77777	77777	77765
77420	76507	76414	65735	65702	65422	65066
07740	06564	64114	64174	64174	64174	64174
07700	00006	00421	42340	64114	64174	64174
77007	07025	00007	00043	04235	24761	64174
77777	77770	70770	00077	00000	00061	04277
77777	77777	77777	07700	00654	00007	00000
77777	77777	77777	70004	06457	42126	00656
77777	77777	77777	70004	04657	42707	42703
77777	77777	77777	70077	06453	42707	42707

Table 1 : Octal strings of the mage portion 45x8 to 55x14

32767	32767	32767	32767	32767	32767	32767
32767	32767	32767	32767	32767	32767	32757
32528	32071	32012	27613	27586	27410	27190
4036	3444	26700	26748	26748	26748	26748
4032	6	273	17632	26700	26748	26748
32263	3605	7	35	2205	17905	26748
32767	32760	29176	63	0	49	2239
32767	32767	32767	4032	428	7	0
32767	32767	32767	28676	3375	17494	430
32767	32767	32767	28676	3375	17863	17859
32767	32767	32767	28735	3371	17863	17863

Table2 : Integer values of the image portion 45x8 to 55x14

The image reconstructed by first converting this integer to an octal stream and then combining the respective color bits to a five bit format is shown in figure 3.

Figure 3 : Reconstructed Image of size 201 x 300

6. CONCLUSION

Most of the image processing techniques are application specific, and here the focus is on, using the acquired images for printing. It has been shown through the experimental results that the lossy compression discussed, has no impact in reducing print quality and it does not compromise on its visual quality. The storage requirement for each pixel is just two-third of the actual data. The procedure is very simple to implement and the compression can be done offline to save the processing time. With parallel processing, the color components can be handled simultaneously to fasten the computing time.

7. REFERENCES

[1] H Nobuhara, W Pedrycz, K Hirota, *"Fast solving method of Fuzzy relational equation and its application to lossy image compression/reconstruction",* IEEE, Transactions on Fuzzy Systems, Vol 8, Issue 3,pp 325-334, June, 2000.

[2] HuanjingYue, Xiaoyan Sun, Jingyu Yang, *"SIFT-Based Image Compression"*, IEEE, International Conference on Multimedia and Expo, 2012.

[3] Krishnan Gupta, Mukesh Sharma, NehaBaweja,*"Three different KG version for Image Compression"*, International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014, IEEE.

[4] M M Reid, R J Millar, and N D Black, "*Second Generation Image Coding : an overview*", ACM Computing Surveys, Vol. 29, No.1, March 1997.

[5] SrinivasDonapati, Harish Yagain, "A Comparative study of effects of CSC on Image Compression Ratios while using JPEG-XR", International Symposium on Electronic System Design, 2013.

[6] T Chen, K Chuang, *"A Pseudo lossless Image Compression Method*", IEEE Congress on Image and Signal Processing, Vol. 2, pp 610-615, 2010.

[7] Yu Shen, XiepingGao, Linlang Liu, Qiying Cao, *"Integer to Integer Multiwavelets for Lossless Image Compression",*Proceedings of IEEE IC-BNMT 2011.