Comparative Study of Predictors Used in Lossless Image Compression

G.Shanmathi[1] and Shima Ramesh Maniyath[2]

[1]UG Student, Department of ECE, MVJ College of Engineering, Bangalore, India. Email: shanmathig16@gmail.com
[2]Assistant Professor, Department of ECE, MVJ College of Engineering, Bangalore, India. Email: ramesh.shima86@gmail.com

Article Received: 12 May 2017 *Article Accepted: 01 June 2017* *Article Published: 04 June 2017*

ABSTRACT

Lossless image compression has a wide variety of applications, especially when data acquisition is expensive such as aerial, medical and space imaging. Compression techniques used should be fast and efficient apart from providing high compression ratios. So, predictive methods of compression are used rather than transform based methods. The first step in the predictive method is to use a predictor which removes spatial redundancy in an image which is followed by context modelling and entropy coding techniques. This paper gives a comparative analysis of the commonly used predictors in terms of entropies after prediction. The predictors discussed in this paper include median edge detection (MED) predictor used in JPEG-LS standard, gradient adjusted predictor (GAP) used in CALIC and gradient edge detection (GED) predictor. Analysis shows that GED, which combines the simplicity of MED and efficiency of GAP, provides better entropy than the other predictors. A brief of context modelling and entropy coding is also explained in this paper to give a complete overview of predictive lossless image compression.

Keywords: Lossless image compression-Predictive coding, MED predictor, GAP, GED predictor, context modelling, entropy coding.

1. INTRODUCTION

Although there are multiple new techniques for data storage and transmission, more efficient compression techniques are required as the data generation rates are increasing rapidly. Images and videos carry certain information. Redundancies occur when information is carried by more data than required. Compression is a data transform that reduces the amount of data while preserving the information that is carried by the image. Image compression techniques are of two types [1] - Lossy image compression and Lossless image compression. Lossy image compression is a data encoding method which discards data that carry information which are unnoticeable to the human eye. This technique provides a better compression ratio than lossless compression and reduces the memory requirement but results in loss of data, thus preventing the reconstruction of the original image. Lossless image compression encodes an image without loss of data, which produces an exact replica of the original image when decompressed.

There are three types of lossless image compression – dictionary coding, entropy coding and predictive coding. In dictionary coding, the values in the database (dictionary) are used to represent the actual values present in the image during transmission. In entropy coding, symbols are represented by average number of bits. In Predictive coding, the difference between the predicted value and the actual value is transmitted. There are few disadvantages in dictionary and entropy coding. Dictionary coding is well suited for long files, encoding short files results in long codes causing transmission or storage of more number of bits. A major problem in entropy coding is that if there are some codes that have the same ending as the beginning of some other codes, they cannot be differentiated. Hence we go for predictive coding technique which is simple and efficient.

Predictive coding for lossless image compression comprises of prediction, context modelling and entropy coding. Simple and efficient predictor removes the spatial redundancy (correlation between neighbouring pixels). Context modelling improves the output of the predictor by including information about the pixels such as horizontal or vertical edges. Finally entropy coding reduces statistical redundancy (inter-pixel redundancy and coding redundancy) resulting in code stream.

In this paper, Section II gives a brief about the performance parameters and a generalized predictive lossless compression scheme. Section III explains the different predictors used for predictive lossless compression – MED, GAP and GED predictors. Section IV gives an overview of context modelling and entropy coding. Section V analyses the predictors based on their entropy after prediction. Finally Section VI concludes the paper.

2. PERFORMANCE PARAMETERS AND LOSSLESS COMPRESSION SCHEME

PERFORMANCE PARAMETERS

Lossless image compression must preserve every pixel value present in an image irrespective of whether it is a valid pixel or noise. The performance of any compression algorithm can be expressed in terms of compression efficiency and complexity. Compression efficiency can be measured by either compression ratio or bit rate. Compression ratio is the ratio of the original size of the image to the size of the compressed image (code stream). Bit rate is the number of bits required to store a single pixel value of the image. Compression ratio can also be estimated using the measure of entropy. Entropy is the minimum bit rate required to encode an image, assuming that images belong to Markov processes. If we denote an image as a random variable X, with an alphabet A = (a0, a1, a2... aN-1), which mean we have an N-bit image, entropy can be calculated as follows:

$$H(X) = -\sum_{x \in A} p(x) \log_2 p(x)$$

Where p(x) is associated probability of a symbol x.

Complexity depends on the coding algorithm used and is measured by the number of arithmetic operations required to perform encoding and decoding operation. Complexity plays an important role when online compression is performed using a software where speed is a matter of concern.

LOSSLESS COMPRESSION SCHEME

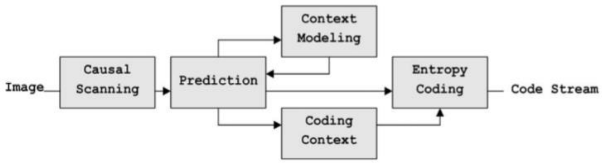

Fig.1. General predictive image lossless compression scheme

General lossless compression scheme consists of a predictor, context modelling block, coding context block and an entropy coder (Fig 1). The predictor predicts the value of the current pixel from a finite number of causal pixels. The predictor output is the difference between the original intensity and the predicted value which is coded using entropy coder, thus removing spatial redundancy. Context modelling block improves the output of the predictor with feedback by including information about the texture of an image. The coding context block ensures efficient coding by pre-processing the output of the predictor which may include alphabet reduction, prediction error remapping and conditional coding.

3. PREDICTORS USED FOR LOSSLESS COMPRESSION

3.1 MEDIAN EDGE DETECTION (MED) PREDICTOR

Median edge detection (MED) Predictor is used in Low Complexity Lossless Compression for Image (LOCO-I) [2] as it provides both simplicity and efficiency. It can be classified as a switching predictor based on its local characteristics as it selects one of the three sub-predictors based on the type of area it selects. MED [3] uses only three causal pixels to the type of area the predicted pixel belongs to i.e. either horizontal edge, vertical edge or smooth area. For each image pixel, the MED predictor generates predictive values which is called the predictive image. The entropy coder codes the error values obtained by the difference between the original pixels and the predicted values.

The predictive template is shown in Fig. 2 which uses the causal neighbor pixels to determine the predicted value. 'x' is the predicted pixel, 'a', 'b' and 'c' are the neighboring pixels. The MED uses past data (a, b and c) to detect horizontal or vertical edges in the predictive template. When a vertical edge appears to the left of 'x' MED uses 'b' as the predicted value. When a horizontal edge appears on top of 'x' MED uses 'a' as the predicted value. When no edge appears, MED uses 'a+b-c' as the predicted value.

If x' is the predicted value of x, the predictive rule can be expressed as

$$x' = \begin{cases} \min(a,b) & if\ c \geq \max(a,b) \\ \max(a,b) & if\ c \leq \min(a,b) \\ a+b-c & otherwise \end{cases}$$

At the recovery stage, the original image can be obtained by adding the error values with the predicted values obtained using the same MED predictor. Although the compression techniques that use MED have a local gradient, MED does not use them. Another drawback is that MED cannot be used on high noise areas.

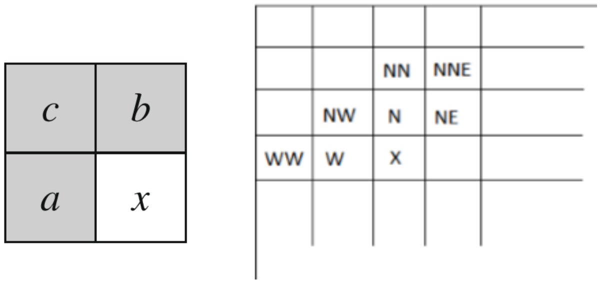

Fig. 2. Predictive template for MED (left) and Predictive template for GAP (right)

3.2 GRADIENT ADJUSTED PREDICTOR (GAP)

GAP is a simple, adaptive, nonlinear predictor which was first used in CALIC [4] (Context-based adaptive lossless image codec). GAP is based on gradient estimation of current pixel and is characterized by high flexibility to different regions. It is highly adaptable as it can identify sharp, regular and smooth vertical and horizontal edges as well as smooth areas. Hence it is more robust than traditional DPCM like linear predictors especially in strong edge areas.

[5] The predictive template with labelled causal pixels is shown in Fig. 2. It uses seven neighboring pixels to determine the predicted value. The first step is to determine the horizontal and vertical gradients which is given by the following equations:

$$g_h = |W - WW| + |N - NW| + |N - NE|$$
$$g_v = |W - NW| + |N - NN| + |NE - NNE|$$

For prediction, the predictor uses the estimated gradient and three heuristic thresholds which are assigned for 8-bit data. The predicted value P is estimated as follows:

Sharp horizontal edge:

$$if\ (g_v - g_h) > 80, P = W$$

Sharp vertical edge:

$$elseif\ (g_v - g_h) < -80, P = N$$
$$else, P = (W+N)/2 + (NE - NW)/4$$

Horizontal edge:

$$if\ (g_v - g_h) > 32, P = (P+W)/2$$

Weak horizontal edge:

$$esleif\ (g_v - g_h) > 8, P = (3P+W)/4$$

Vertical edge:

$$elseif\ (g_v - g_h) < -32\ , P = (P + N)/2$$

Weak vertical edge:

$$esleif\ (g_v - g_h) < -8\ , P = (3P + N)/4$$

GAP provides better efficiency than MED but its complexity increases as it uses three heuristic threshold levels for prediction.

3.3 GRADIENT EDGE DETECTION (GED) PREDICTOR

Gradient edge detection (GED) [6] predictor combines the simplicity of MED and the efficiency of GAP. It uses five causal neighboring pixels to estimate the gradient which is a compromise between GAP which uses seven and MED that uses three neighbor pixels. Local gradient is estimated similar GAP, but here a single threshold value is used for prediction.

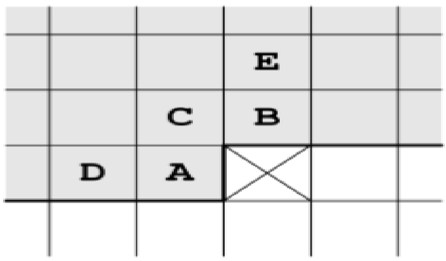

Fig. 3. Predictive template for GED predictor

Fig. 3 shows the labels for causal pixels used in prediction. Local gradient can be estimated using the following expression:

$$g_h = |D - A| + |C - B|$$
$$g_v = |C - A| + |E - B|$$

The prediction value, P is determined using:

$$if\ (g_v - g_h) > T\ , P = A$$
$$elseif\ (g_v - g_h) < -T, P = B$$
$$else\ P = 3(A + B)/8 + (C + D + E)/12$$

Where T is the predefined threshold which can be fixed or user defined for every image.

4. CONTEXT MODELLING AND ENTROPY CODING

CONTEXT MODELLING

Context modelling further improves the output of the predictor with the help of repeating neighboring pixels of the current pixel. Often images contain textures that are represented by repeating pixel values. Each time a scheme is detected, context model is updated which helps in understanding the probability distribution of pixels in that scheme. If we consider ‘n’ previous pixels of binary image, there can be 2^n different schemes. To avoid this bottleneck, similar schemes can be grouped in one context and its determination is important for prediction correction.[7,8] Context is determined by applying detection rules on neighboring pixels. Each context is provided with an accumulator and a counter which are constantly updated. Accumulator ‘A’ contains the sum of prediction errors each time a context is updated. Counter ‘N’ is incremented for every context detection. After prediction and context detection, the predicted value is corrected as

$$P_c = P + [N/A]$$

Large number of contexts can lead to context dilution, a phenomenon when number of contexts are so high that they cannot be learned from scanning. Very less contexts is also a problem because, conditional probability of the following pixel values may not determine an optimum correction for prediction. Another problem is the presence of areas with different textures in an image. A solution to these problems is offered by setting a limit to the number of contexts. When the counter exceeds this limit, the counter value and the context accumulator is halved.

ENTROPY CODING

Entropy coder is the final step in lossless image compression which actually performs the compression of the image. Previous steps were used to remove redundancy, hence reducing the transmission bandwidth and storage space required. Variable length code words are preferred over fixed length code words to remove statistical redundancy. Removing statistical redundancy exploits the fact that some pixel values occur in high frequencies, thus coding them with minimum length code words reduces the storage space. When encoding gray scale images, the input pixels are values of grayness. Most commonly used entropy encoders are Huffman coder, Golomb-Rice coder and arithmetic coder.

5. COMPARATIVE ANALYSIS

Entropies after prediction for MED, GAP and GED with threshold equal to 8 is given in Table 1. Examples for test images are shown in Fig.4

Fig. 4. Examples of test images, from left to right and from top to bottom: *plane, milkdrop, lake, peppers, cameraman, Lena.*

Image	Entropies after prediction [bpp]		
	MED	GAP	GED8
Plane	4.20	4.15	4.23
Milkdrop	3.80	3.76	3.77
Lake	5.38	5.26	5.37

Peppers	4.93	4.72	4.81
Cameraman	4.74	4.72	4.68
Lena	4.54	4.39	4.68

Table.1. Entropies of prediction error images for MED, GAP and GED (with threshold 8) predictors.

MED predictor is simple but cannot be used on high noise areas. GAP uses smaller error entropy than MED but is complex as it uses three heuristic thresholds. GED predictor combines the advantage of simplicity of MED and efficiency of GAP to give a compromising bit rate between MED and GAP. Another point to be noted is computationally expensive scaling does not guarantee less scaling. Thus with an optimum threshold, GED predictor produces comparable bit rates as with a more complex GAP.

6. CONCLUSION

Basic concept of predictive lossless image compression has been discussed in this paper. The commonly used predictors namely Median Edge Detection (MED) predictor, Gradient Adjusted Predictor (GAP) and Gradient Edge Detection (GED) predictors are explained and their entropies are compared. GED predictor combines the advantages of MED and GAP predictors. Comparison shows that simple GED predictor can achieve a comparable bit rate and can be used for high resolution images by selecting appropriate threshold value.

7. FUTURE SCOPE

In future, these methods can also be applied to live photos for lowering storage space.

REFERENCES

[1] Rajesh Mandale, Akshay Mhetre, Rahul Nikam & Prof.Dr.V.K.Bairagi “Image Compression Based On Prediction Coding”.

[2] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,” *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[3] Hsien-Chu Wu, Hao-Cheng Wang, Chwei-Shyong Tsai, Chung-Ming Wang “Reversible image steganographic scheme via predictive coding”.

[4] X. Wu, N. Memon, “CALIC – A Context Based Adaptive Lossless Image Codec”, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP-96,* vol. IV, pp. 1890-1893, May 1996.

[5] Vikas Bajpai, Dushyant Goyal, Soumitra Debnath, Anil Kumar Tiwari “Multidirectional Gradient Adjusted Predictor” *IEEE International Conference on Signal and Image Processing* 978-1-4244-8594-9

[6] Aleksej Avramović, Branimir Reljin “Gradient Edge Detection Predictor for Image Lossless Compression”, *52nd International Symposium ELMAR-2010, 15-17 September 2010, Zadar, Croatia.*

[7] Aleksej Avramović, Goran Banjac, “On Predictive-Based Lossless Compression of Images with Higher Bit Depths”.

[8] M.A. Mohammed, et al., Artificial neural networks for automatic segmentation and identification of nasopharyngeal carcinoma, *J. Comput. Sci. (2017), http://dx.doi.org/ 10.1016/j.jocs.2017.03.026.*