ISSN:0975-9646

Sruthi Chandran et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (4) , 2013, 614 - 618

A New Predictive Image Compression Scheme using Histogram Analysis and Fractal Method

Sruthi Chandran, Anitha

Department Of CSE, Calicut university, Kerala

***Abstract*— In this paper, we introduce a new predictive image compression scheme that compresses an image by a set of parameters computed for individual blocks of different types. These parameters include the average and difference of the representative intensities of an image block, together with the index of a pattern associated with the block visual activity. The block representative gray values are computed through a histogram analysis of the block residuals and fractal technique is employed to find the best match for the block bit-pattern. To further reduce the bit rate, a predictive technique selectively predicts the parameters based on the corresponding values in the neighboring blocks. The simulation results confirm that the proposed technique can provide a high compression ratio with acceptable image quality of the compressed images.**

***Keywords*— component; image compression; block patterns; block histograms; fractal; predictive coding.**

I. Introduction

Efficient image compression solutions are becoming more critical with the recent growth of data intensive, multimedia-based web applications. There are numerous methods for compressing digital image data and each has its own advantages and disadvantages [1]. Vector quantization (VQ) [2], block truncation coding (BTC) [3]-[4], and transform coding [5] are three commonly used schemes for image compression. VQ is widely used because it has a simple decoding structure and requires a low bit rate. However, a lot of computational costs are needed in the codebook design procedure and the image coding procedure. In general, BTC has the advantages of requiring very low computational cost and providing good reconstructed image quality. However, a high bit rate is needed in BTC. Predictive coding scheme is exploited due to existing the similarity among neighboring image blocks and is adopted to image compression techniques. The study in [6] presents a predictive image compression scheme that combines the advantages of vector quantization and moment preserving block truncation coding. The prediction technique employs the squared Euclidean distance between two blocks to find the similarity of the current block with a neighboring block. If a similar compressed image block can be found in the neighborhood of current processing block, it is taken to encode this block. Otherwise, this image block is encoded either by vector quantization or moment preserving block truncation coding. A hybrid images coding scheme that combines the advantages of AMBTC, predictive coding scheme, VQ, and DCT coding algorithms is presented in [7]. In this method, the predictive coding scheme combined with VQ is used in encoding the bit-map blocks generated by AMBTC. In this study, Hamming distance is used to measure the difference rather than Euclidean distance. Hamming distances of the current block with the neighboring blocks are compared with a pre-defined threshold, and the current block is replaced with the neighboring block with the minimum Hamming distance.

In this paper, we present a modified scheme of the image compression technique that we reported earlier in [8], by incorporating a predictive technique. The proposed technique compresses an image block by transmitting a set of parameters computed through a histogram analysis of residual blocks and a pattern matching scheme. Moreover, for exploiting the correlation between the adjacent image blocks, these values are selectively predicted based on the corresponding values in the neighboring blocks. The rest of the paper is organized into four sections. The concept of the proposed compression algorithm is introduced in Section 2. In section 3, the predictive coding scheme is presented. Experimental results are given in Section 4.

II. Literature Survey

In the proposed image compression algorithm in [8], an image is block coded according to the type of individual blocks. A novel classifier, which is designed based on the histogram analysis of residual blocks, is employed to classify the blocks according to their level of visual activity. The classifier places each block into one of the two categories of uniform or edge block. A uniform block is coded by the block mean, whereas an edge block is coded by a set of parameters associated with the pattern appearing inside the block. Like the original BTC algorithm [3], our method encodes an edge block by initially computing two gray values and constructing a bit-map. However, in the proposed method the computation of the gray values, namely the low and high representative intensities are carried out through analysis of the block residuals' histogram. Moreover, instead of transmitting the two gray values, their average and difference will be sent to decoder. Finally, instead of transmitting the whole bit-map for the processed edge block, an optimum bit-pattern is selected from a set of pre-defined patterns, and its index will be transmitted. The use of these parameters at the receiver reduces the cost of reconstruction significantly and exploits the efficiency of the proposed technique. A brief description of the block classifier and the coding scheme are given in the next two sub-sections.

A. Block Classifier

A novel histogram-based classification scheme has been developed for classifying the image blocks [6]. The method operates based on the distribution of the block residuals and classifies block either as a low-detail (uniform) or as a highdetail (edge) block. The classifier employs the block residuals and classifies the block according to their histogram. The classification is carried out through a peak detection method on the histogram. A brief description of the classifier is as follows: Each block of 4x4 pixels is converted into a residual block by subtracting the sample mean from the original . The residual samples are less correlated than the original samples within a block. Here, two of the most important local characteristics of the image block are considered*: central tendency*, represented by the mean value and the *dispersion* of the block samples about the mean, which is represented by the residual values. The challenge here is to analyze the dispersion of the residual values about the mean. One way of achieving this is to sort the histogram of the block residual samples. As the neighboring pixels in the original block are highly correlated, the residual samples will tend to concentrate around zero. One can then quantize the residual samples prior to forming the histogram. The histogram of the quantized residuals may then be formed and analyzed by simply detecting its peaks. Based on the distribution of the residual samples within the test images, we choose to apply a coarse quantization, in particular a 15-level non-uniform quantizer. We now define q_j as the output of the quantizer with index *j*, as shown in "Fig. 1". The histogram of the quantized values $h(q_j)$ may then be formed to provide the occurrence of q_j. The quantized residual histogram (QRH) is then analyzed by simply detecting its peaks. According to the number of detected distinct peaks on the histogram, image blocks can be placed into two major categories of uniform and edge blocks. A histogram with a unique peak at its centre (uni-modal histogram) identifies a uniform block. Whereas, the existence of two distinct peaks implies that the processed block is an edge block. "Fig. 2" and "Fig. 3" show examples of both types of blocks. A peak on the histogram indicates a high score of residual values; therefore it is fair to conclude that there is a considerable number of pixels that have the same dispersion about the block mean. This, in turn will lead us to conclude that the gray level values of these pixels are very close to one another. Hence, this group of pixels can be represented by a single gray value. In this analysis, a distinct peak on the QRH of the processed block represents a gray value *X*

j	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
q_j	-67	-52	-39	-28	-17	-10	-5	0	5	10	17	28	39	52	67

Figure 1. The quantizer output q_j with index *j*

185	182	182	178
185	182	182	178
185	182	182	178
185	183	182	175

(a)

4	1	1	-3
4	1	1	-3
4	1	1	-3
4	2	1	-6

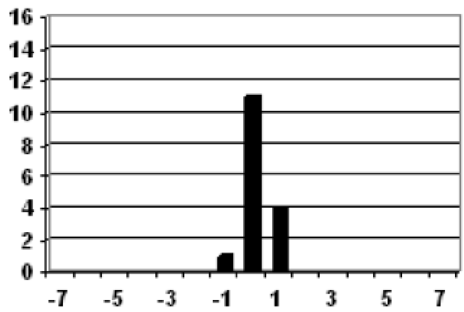
(b)

5	0	0	0
5	0	0	0
5	0	0	0
5	0	0	-5

(c)

1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	-1

(d)



(e)

Figure 2. (a) Original uniform block; sample mean = 181, (b) Residuals (c) quantized residuals, (d) quantizer's indexes, (e) block QRH

$$X_j = X_{mean} + q_j \quad (1)$$

where X_{mean} is the block mean. For a uniform block, since the single peak occurs at the center of the histogram, where $q_j = 0$, then from Eq.1 the representative intensity X_j will be the same as the block mean. For an edge block, the two peaks of the *QRH*, which are positioned on the left and right hand side of the centre (*j=0*) represent the low representative intensity *XL* and the high representative intensity *XH* , respectively. If the two peaks are positioned at indexes j' and j'' , the two representative intensities are calculated as:

$$X_L = X_{mean} + q_{j'}$$
$$X_H = X_{mean} + q_{j''} \quad (2)$$

In "Fig. 3", *Xmean* = 125, and *XL* and *XH* are computed using Eq. 2 ; X_L = 125 + (-39) = 86 and *XH* =125 + 28 = 153.

B. *Coding Of Image Blocks*

This method is also called "*statistics aware embedding*" or "*masking*". Once the image blocks have been classified, the coder, switches between a one-level (uniform block) and a bilevel(edge block) representation. A uniform block is encoded by transmitting the block mean plus an indicator to inform the decoder that the block is uniform. By forcibly clustering all pixels in an edge block into two groups, a bi-level approximation of the block is obtained. The clustering partitions a block *W* into two sets of pixels, *W*0 and*W*1, suchthat $W = W0 \cup W1$ and $\cap = \Phi$ 0 1 *W W* . The clustering is carried out by marking the pixels of set *W*0 and *W*1 by '0' and '1', respectively. Thus the clustering can be represented as a bit-pattern, $B = \{b1, b2 ,...b16 \; bi \in (0,1)\}$. By selecting the block mean as a threshold, the bit-pattern can be generated as :

$$b_i = \begin{cases} 1 & \text{if} \quad x_i > X_{mean} \\ 0 & \text{if} \quad x_i \le X_{mean} \end{cases} \quad (3)$$

where, $x_i \in W$ are the intensities of the pixels of the edge block. It is noted that, X_L and *XH* are the representative intensities of the set *W*0 and *W*1 , respectively. Like the original BTC, an edge block can be coded by transmitting the representative intensities and the bit-pattern. However,

in our method, we transmit the average, M and difference, l of
the representative intensities, defined by :

$$M = \frac{X_H + X_L}{2}$$

$$l = \frac{X_H - X_L}{2}$$

The values M and l represent the low and high frequency components, respectively. It is evident fro eq.3 that $XH = M + l$ and $XL = M - l$. During the reconstruction, the coded block can be constructed by :

$$b_i = \begin{cases} M + l & \text{if} \quad b_i \in W_1 \\ M - l & \text{if} \quad b_i \in W_0 \end{cases}$$

l

It should be noted that for a uniform block, since both representative intensities are the same as the block mean, therefore, $M = Xmean$ and $l = 0$. Instead of transmitting the whole bit-pattern of an edge block, further bit reduction can be achieved by finding the best match for the block bit pattern from a set of pre-defined patterns, Pk , $k = 0,1,2,\ldots, N$. A set of 32 patterns shown in "Fig. 4", which preserve the location and polarity of edges in four major directions and their complements making $N=64$ is used in our method. The pattern matching stage is carried out by performing a logical exclusive NOR operation on the block bit-pattern and each pattern from the set to calculate a matching score, *ms* ,given as :

$$ms = \sum_{i=0}^{3} \sum_{j}^{3} \overline{(P_{ij} \oplus b_{ij})}$$

The pattern with the highest *ms* is selected and its index k will be transmitted. Since, the proposed method sends k instead of the whole block bit-pattern, only 64 log2 = 6 bits are transmitted. Each image block is therefore encoded by generating a triple (M, l, k) . "Fig. 5" illustrates an edge block with block mean = 125, its bit-pattern, the selected pattern from the set *(k=23)* as well as the reconstructed block. The reconstructed values were calculated in the previous sub-section, from Eq. 2. Using Eq. 4, M and l are computed as (120, 34, 23) to generate the compression code.

It should be also noted that, since for a uniform block, no pattern index is transmitted, therefore the compressed code for such a block is the pair (M,l) , where $l = 0$. The value M in the triple (M, l, k) , can be coded by 8 bits, whereas coding l requires only 6 bits, as its standard deviation is smaller than that of $X L$ and XH . Therefore, the compression code of an edge block requires 20 = (8+6+6) bits to be transmitted. For a uniform block, the number of bits required to code the pair *(M, 0)* is 9 = 8+1. The bit-rate in bits per pixel (bpp) for an 8-bit gray level image size of *w x h,* with *u N* uniform blocks, and *Ne* edge blocks is given by :

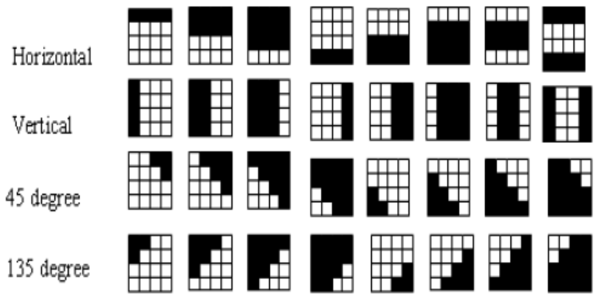


Figure. 4: Set of 32 pre-defined patterns.

150	91	89	85
149	122	90	92
152	148	135	89
156	142	149	147

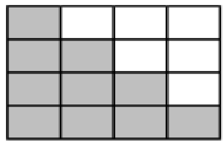
(a)

1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1

(b)

153	86	86	86
153	153	86	86
153	153	153	86
153	153	153	153

(d)



(c)

Figure 5. (a) Original edge block, (b) Bit-pattern
(c) Matched pattern, (d) Reconstructed block

III. PREDICTIVE CODING

As mentioned earlier that M is bias or average intensity for both types of blocks, it has smaller variation across adjacent blocks. In particular, the M value of an edge block has a stronger correlation with that of its neighboring blocks compared to $X L$ and XH , because of spatial homogeneity. This suggests that if we use some form of selective prediction on the M value of a block based on the M values of the previous blocks then a further reduction in bpp can be achieved. "Fig.6" shows the blocks that are used for prediction of M. These are the neighboring left, upper, upper left and upper right blocks for the current block. Suppose M values of the blocks numbered 1, 2,3,4, C are $M1$, $M2$, $M3$, $M4$, and Mc , and are known. In order to check the closeness of the M value of the current block with the M values of the neighboring blocks, we define the correlation factor R given as :

$$R_\alpha = \frac{M_c - M_\alpha}{M_c}, \quad \alpha \in \{1,2,3,4\} \tag{8}$$

Used for Prediction 2	Used for Prediction 3	Used for Prediction 4
Used for Prediction 1	Current Block C	

Figure 6. Blocks used in prediction of M in current block

The factor R for each neighboring block is computed from Eq. 8 and the smallest one is compared against a predefined threshold *Th*. If it differs below the threshold then *M* value of the block having the minimum *R* is used for the current *M* value. The steps involved in predicting the *M* values are as follows :

1. Encode the blocks starting from left to right and top to bottom sequence.
2. If the block is in the first row or first column then go to step 5.
3. Obtain the value of *M* for the processed block and calculate R_α for the neighboring blocks.

Fractal Image Coding

Principle of Fractal Coding

In the encoding phase of fractal image compression, the image of size N x N is first partitioned into no overlapping range blocks *Ri* , { *R* , *R* ,...*Rp* 1 2 } of a predefined size BxB. Then, a search codebook is created from the image taking all the square blocks (domain blocks) *j D* , { *q D* ,*D* ,...*D* 1 2 } of size 2Bx2B, with integer step L in horizontal or vertical directions. To enlarge the variation, each domain is expanded with the eight basic square block orientations by rotating 90 degrees clockwise the original and the mirror domain block. The range-domain matching process initially consists of a shrinking operation in each domain block that averages its pixel intensities forming a block of size BxB. For a given range R*i* , the encoder must search the domain pool Ωfor best affine transformation w*i* , which minimizes the distance between the image R *i* and the image w*i* (D*i*), (*i.e.* w*i* (D*i*) ≈ R*i*). The distance is taken in the luminance dimension not the spatial dimensions. Such a distance can be defined in various ways, but to simplify the computations it is convenient to use the Root Mean Square RMS metric. For a range block with n pixels, each with intensity r *i* and a decimated domain block with n pixels, each with intensity d *i* the objective is to minimize the quality

The parameters that need to be placed in the encoded bit stream are *i s* , *i o* , index of the best matching domain, and rotation index. The range index *i* can be predicted from the decoder if the range blocks are coded sequentially. The coefficient s *i* represents a contrast factor, with | *i s* |≤ 1.0 , to make sure that the transformation is contractive in the luminance dimension, while the coefficient o *i* represents brightness offset.

At decoding phase, Fisher [5] has shown that if the transforms are performed iteratively, beginning from an arbitrary image of equal size, the result will be an attractor resembling the original image at the chosen resolution.

Baseline fractal image coding algorithm

The main steps of the encoding algorithm of fractal image compression based on quadtree partition [5] can be summarized as follows:

Step 1: Initialization (domain pool construction)
Divide the input image into N domains, $_jD$
For (j =1; j ≤ N; j ++)
Push $_jD$ onto domain pool stack Ω
Step 2: Choose a tolerance level c;
Step 3: Search for best matches between range and domain blocks

```
For ( i =1 ; i ≤ num_range ; i ++ ) {
min_error = c;
For ( j =1 ; j ≤ num_domain; j ++ ) {
Compute s, o;
If (0 ≤ s< 1. 0)
If ( E( iR , jD ) < min_error) {
min_error = E( iR , jD );
best_domain[i] =j ; }
}
If (min_error = = c)
Set R i uncovered and partition it into 4 smaller blocks;
Else
Save_coefficients(best_domain, s, o);
}
```

In this algorithm, parameter _ *C* settles the fidelity of the decoded image and the compression ratio. By using different fidelity tolerances for the collage error, one obtains a series of encodings of varying compression ratios and fidelities. For a range block if _ *C* is violated for all the domain blocks, that is the range block is uncovered, the range block is divided into four smaller range blocks, and one can search for the best match domains for these smaller range blocks. At the end of **step 1** the domain pool .

IV Performance Evaluation

Peak Signal to Noise Ratio (PSNR). Performance measurement for image distortion is well known as peak signal to noise ratio (PSNR) which is classified under the difference distortion metrics can be applied on stego images. We use Peak signal to noise ratio (PSNR) to evaluate quality of stego image after embedding the secret message .This is basically a performance metric and use to determine perceptual transparency of the stego image with respect to cover image.It is measured in terms of decibel(db).Higher the PSNR higher the quality of the image(which means there is a little difference between cover image and stego image)..Quality of the image is more when it is greater than 40db and less when PSNR is 30db or low.ie PSNR is measured in terms of MSE(Mean Square Error).Thus performance can be measured. PSNR is defined by using the following equation.

$$PSNR = 10\log_{10}(255^2 / MSE)$$

V Conclusions

In this paper a parameterized and non-adaptive version of domain pool reduction is proposed, by allowing an adjustable number of domains to be excluded from the domain pool based on the value of the domain block, which in turn reduced the encoding time. Experimental results on standard images showed that removing domains with high from the domain pool has little effect on the image quality while significantly reduce the encoding time. The proposed method is highly comparable to other acceleration techniques. Next step in our research is to use the proposed method to improve the speed of hybrid coders (gaining better results than JPEG) that are based on fractal coders and transform coders so as to improve their performance.

REFERENCES

[1] M. Barnsley and A. Jacquin. Applications of Recurrent Iterated Function Systems to Images. SPIE Visual Communications and Image Processing, Vol. 1001,

[2] A. E. Jacquin. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transform. IEEE trans. on Image Processing, Vol. 1, N

[3] M. Barnsley and L. Hurd. Fractal Image Compression. on Image Processing: Mathematical Methods and Applications. pp. 183-210, Clarendon Press, Oxford, 1997.

[4] B. Ramanurthi and A. Gersho. Classified Vector Quantization of Image. IEEE Trans. Communication, COM-34, Vol.11, pp. 1105-1115, 1986.

[5] Y. Fisher. Fractal Image Compression: Theory and Applications. Springer-Verlag, New York, 1994.

[6] R. Hamzaoui. Codebook Clustering by Self-Organizing Maps for Fractal Image Compression. In NATO ASI Conf. Fractal Image Encoding and Analysis, Trondheim, July 1995. Fractals, Vol. 5, Supplementary issue, April 1997.

[7] R. Hamzaoui and D. Saupe. Combining Fractal Image Compression and Vector Quantization. IEEE Trans. on Image Processing, 9(2), pp.197-208, 2000.

[8] B. Hurtgen and C. Stiller. Fast Hierarchical Codebook Search for Fractal Coding of Still Images. Proceeding of SPIE, Vol. 1977, pp. 397-408, 1993.

[9] L.M. Po and C.K. Chan. Adaptive Dimensionality Reduction Techniques for Tree-Structured Vector Quantization. IEEE *Trans*. on Communications, Vol. 42, No. 6, pp. 2246-2257, June 1994.

[10] G. Caso, P. Obrador and C.-C. Kuo. Fast Methods for Fractal Image Encoding. Proc. SPIE Visual Communication and Image Processing 95, Vol. 2501, pp.583-594, 1995.

[11] B. Bani-Eqbal. Enhancing the Speed of Fractal Image Compression. Optical Engineering, Vol. 34, No. 6, pp.1705-1710, June 1995.

[12] D. Saupe. Accelerating Fractal Image Compression by Multi-dimensional Nearest Neighbor Search. In Proc. Data compression Conference, March 28-30, 1995.

[13] D. Saupe. Fractal Image Compression via Nearest Neighbor Searching. In Conf. Proc. NATO ASI, Fractal Image Coding and Analysis Trondheim, July 1995.

[14] C.S. Tong and W. Man. Adaptive Approximation Nearest Neighbor Search for Fractal Image Compression. IEEE Trans. on Image Processing, 11(6), pp.605-615, 2002.

[15] E.W. Jacobs, Y. Fisher, and R.D. Boss. Image Compression: A study of the Iterated Transform Method. Signal Process, Vol. 29, pp. 251-263, 1992.

[16] D.M. Monro and F. Dudbridge. Approximation of Image Blocks. in Proc. Int. Conf. Acoustics, Speed, Signal Processing, Vol. 3, pp.4585-4588, 1992.