 ICCES, vol.12, no.4, pp.137-143

A New Quadtree-based Image Compression Technique using Pattern Matching Algorithm

F. Keissarian[1]

Summary

In this paper, a new image compression technique is proposed in which variable block size technique is adopted, using quadtree decomposition, for coding images at low bit rates. The proposed algorithm decomposes the host image into blocks of variable sizes according to histogram analysis of the block residuals. Variable block sizes are then encoded at different rates based on their visual activity levels. To preserve edge integrity, a high-detail block is coded by a set of parameters associated with the pattern appearing inside the block. The use of these parameters at the receiver together with the quadtree code reduces the cost of reconstruction significantly and exploits the efficiency of the proposed technique.

keywords: Image compression, Quadtree decomposition, Pattern classification.

Introduction

Since natural images are mostly smooth with a few highly detailed regions, a good way to compress them is to divide them into smooth segments and encode each with some technique. Since high detail regions are divided into smaller segments, they are encoded with more bits per pixel than low detail regions. An efficient way to perform segmentation is with the use of quadtrees. Many image coding algorithms based on quad tree segmentation have been developed in the past several years [1]-[2]. In a typical algorithm, the image is quadtree segmented into different sized square blocks and those blocks are either coded by a single gray level or by some method such as transform coding or vector quantization. Due to the importance of edges to a human observer in the perception of image quality, variant classified vector quantization algorithms have been developed. The design of an edge-oriented classifier based on edge detection in the spatial domain is not a simple task, it is usually necessary to employ several threshold in the gradient comparison process.

In this paper, we introduce a new quadtree based image compression to achieve high compression ratios and preserve edge integrity. In the proposed approach, low-activity regions, which usually occupy large areas in an image, are coded with a larger block size. A novel classification scheme, which operates based on the distribution of the block residuals, is employed to determine whether the processed block is a low-detail (smooth) or a high-detail (edge) block. To preserve edge integrity, the block pattern matching coding technique, which we presented earlier in

[1]UOWD, U.A.E.

[3] is used to code high-activity regions. The collection of patterns, which is constructed over edge blocks of 4 x 4 pixels, together with the quadtree code reduces the cost of reconstruction significantly and exploits the efficiency of the proposed coding scheme. The organization of this paper is as follows. Section 2 reviews the quadtree decomposition and introduces the block pattern matching algorithm. This is followed by simulation results, presented in section 3.

Coding System

In this section, the formal description of the proposed coding algorithm is given. The quad-tree decomposition algorithm is first presented. The coding of the image blocks through the proposed pattern matching technique is then introduced.

Quad-tree Decomposition

A main point of quadtree segmentation is the evaluation criterion of image segmentation. In quadtree decomposition, a judgment is first made to see whether a block can be represented by a single gray value or whether it must be divided into four subblocks. In this paper, we present a method that operates based on the distribution of the block residuals and determines whether the processed block needs further divisions. This is accomplished by classifying a block either as a low-detail (uniform) or as a high-detail (edge) block. The classifier employs the residual values of a block and classifies the block according to the shape of the histogram of the residuals. The classification is carried out through a peak detection method on the block histogram. A brief description of the classifier is as follows.

Each block of $n \times n$ pixels is converted into a residual block by subtracting the sample mean from the original pixels. 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 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 quantized residual histogram (QRH) is then analyzed by simply detecting its peaks. A minimum score, $Score_{\min}$ can be defined below which a peak at index j is not detected. Based on the formed histograms of test images, we have chosen to use a $Score_{\min}$ of 80% of the total number of samples n^2 for the block size of $n \times n$ pixels. 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. According to the number of detected peaks on the histogram, image blocks can be classified as either low-detailed or high-detailed blocks. A histogram with a unique peak at its center (uni-modal histogram) identifies a low detailed (uniform) block. The existence of two peaks or more implies that the processed block is a high detailed block.

In the decomposition approach, an image to be coded is first divided into blocks of 16x16 and then each block is repeatedly divided into four equal quadrants, if its residual histogram is not a uni-modal type. On the other hand, the decomposition process will stop if the residual histogram of the block has a dominant peak at its center. This block is regarded as a uniform block and all the pixels in the block will be represented by the block mean. If the smallest block size of 4x4 is reached and its residual histogram is still not a uni-modal type, it is regarded as an edge block. Fig. 1 depicts an example of a 4x4 uniform block and its histogram.

Since variable block sizes are used in quadtree segmentation, decoding of transmitted images requires the information about the size and location of each block. That is, if a block is divided into smaller blocks, the quadtree code is "1." Otherwise, the quadtree code is "0." This amounts to too much overhead information needed for transmission. To overcome this problem, we use the method presented in [4] which introduces 17 possible combinations within a 16x16 image block. Only a 6- bit binary sequence ($D^0D^1d^1d^2d^3d^4$) is required to represent each splitting mode as shown in Fig 2.

The first bit D^0 indicates whether or not the 16x16 block is partitioned into four 8x8 blocks. If $D^0 = 1$, then the other four bitsd^1, d^2,d^3,$d^4$are required to indicate whether to split each 8x8 block into four 4x4 blocks or not. The amount of side information is calculated as $\frac{MxN}{16x16} * 6$ bits for an MxN image size. Since the total number of bits in the image is $(WxH) * 8$, the overhead will be around 0.003 bits per pixel (bpp) which is significantly small.

Pattern matching –based Coding of Image Blocks

The uniform blocks of different sizes from 4 x 4 pixels to 16 x 16 pixels are coded by the block mean. To preserve edge integrity, a 4x4 edge block is coded by one of the finite number of pre-defined patterns that best matches its features. The two distinct peaks of the QRH of a 4x4 edge block are represented by two representative intensities. These are the block low and high intensities, denoted by I_{low} and I_{high}.

By forcibly clustering all pixels in an edge block into two groups, a bi-level approximation of the block may be obtained. Only two representative intensities and certain binary bits, forming a bit-map are necessary to specify the bi-level representation. Once the representative intensities of an edge block have been determined, a bit- map may be constructed to specify the correspondence between the pixels and the representative intensities.

In such a bit-map, each pixel is represented as a '1' or a '0'. The detailed

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

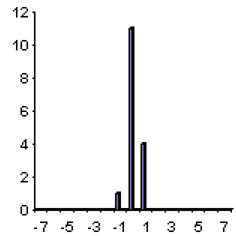
(a)

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

(b)

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

(c)



(d)

Figure 1: Classification of a 4 x 4 uniform block : a) original block (block mean=181) b) Block residuals c) quantized residuals e) uni-modal histogram of the block

description is given simply as:

$$B_{i,j} = \begin{cases} 1, & x_{i,j} \leq I_{mid} \\ 0, x_{i,j} > I_{mid} \end{cases}$$

Where, $I_{mid} = \frac{I_{low}+I_{high}}{2}$.

Once the bit-map of an edge block has been formed, the block can be coded by finding the best match for its bit-map from a set of patterns in a look-up table. A set of 30 patterns (Fig. 3) which preserve the location and polarity of edges in four major directions is used for the pattern matching stage. This process determines the index of the matched pattern selected from the set.

The transmitted information includes the index of the selected pattern in the look-up table as well as the block representative intensities I_{low} and I_{high} for the areas indicated in black and white of the selected pattern, respectively. During the decoding process, the decoder replaces the pixels of a uniform block by the block mean. Whereas, in decoding an edge block, the decoder uses the index of the selected pattern as well as the transmitted intensities to reconstruct the block.

Reconstruction of an edge block is carried out by replacing the 1's and 0's of the selected pattern by I_{high} and I_{low} , respectively. The number of bits to code an edge block *edge B* is computed as: $B_{edge} = 2B_{rep} + \log_2^P + 1$ where,B_{rep}denotes the number of bits required to code one of the representative intensity, $P$is the number of patterns used, $\log_2^P$is the number of bits required to transmit the index of P_{best} , and the 1 is the overhead to inform the decoder the block is an edge block. If $B_{mean}$is the number of bits required to code the mean value of a uniform block, the achievable bit rate in bpp for an 8-bit grey level image may then be determined according to:

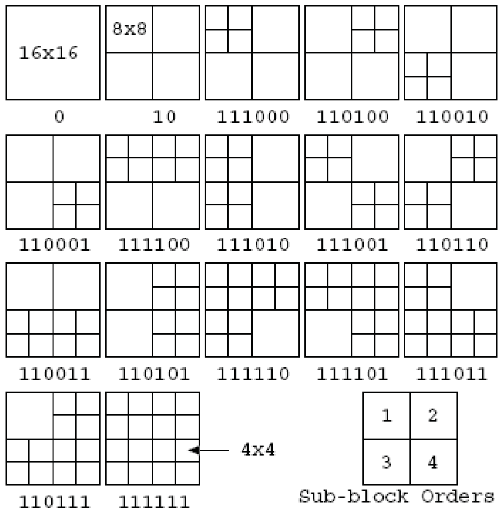


Figure 2: All possible 16 x 16 block partitioning modes

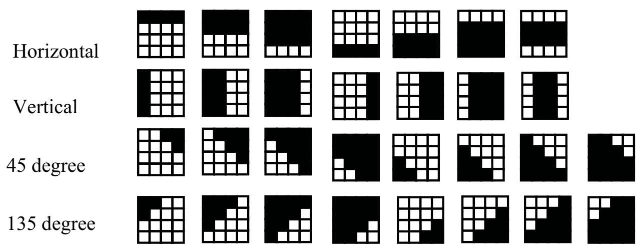


Figure 3: Set of pre-defined patterns

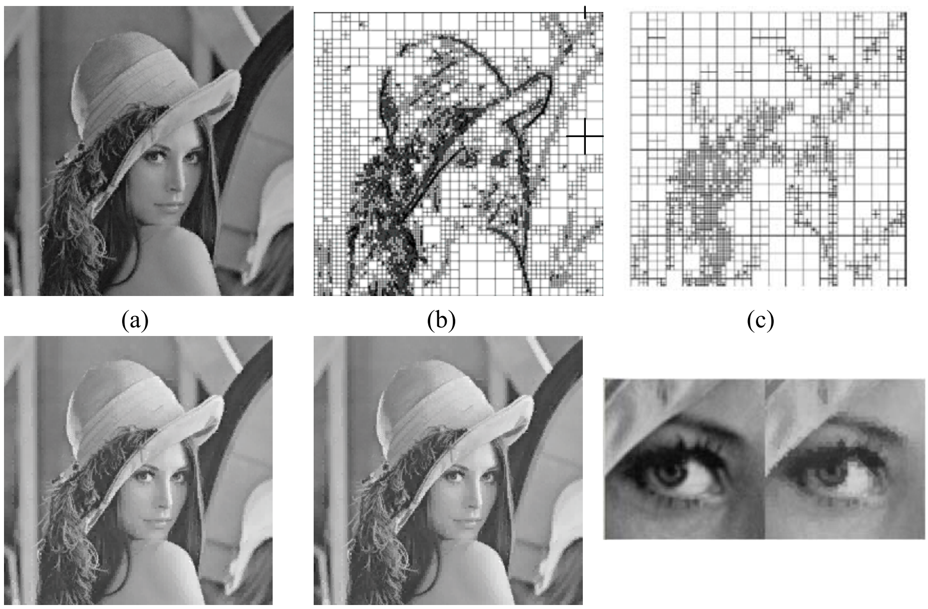
and their associated binary representation

$$bpp = \frac{(N^{16\times16} + N^{8\times8} + N^{4\times4}_{uni}) * B_{mean} + (N^{4\times4}_{edge} * B_{edge})}{M \times N * 8}$$

Where, $N^{16\times16}$: number of 16×16 blocks,$N^{8\times8}$: number of 8×8 blocks,$N^{4\times4}_{uni}$: number of 4×4 uniform blocks and $N^{4\times4}_{edge}$: number of 4×4 edge blocks. The 0.003 bpp for the quadtree overhead is added to the above calculation.

Simulation Results

We have evaluated the performance of the proposed coding scheme through a computer simulation. The computer simulation has been carried out, using a set of 256x256, 8-bit intensity, monochromatic standard images including the images of 'Lena' (Fig.4a). In the implementation used here, the $Score_{\min}$ was set to two different values: 80% and 70% of the total samples in each block. Fig. 4.b and Fig. 4.c illustrate the quadtree segmented images of ‘Lena’ for different $Score_{\min}$ values. Table 1 shows the decomposition, the compression , and the image quality (PSNR) results for the images ‘Lena’ The results show that, setting a higher value for $Score_{\min}$ will result in less number of uniform blocks being identified. This in turn leads to lower compression but a better image quality.



(a) (b) (c)

(d) (e) (f)

Figure 4: (a): Original Image of Lena (256x256), (b): Segmented Image of Lena, $Score_{\min}$ = 80%, (c): Segmented Image of , $Score_{\min}$ = 70%, (d): Compressed Image , $Score_{\min}$ = 80%, (e): Compressed Image , $Score_{\min}$ = 70%, (f): The magnified versions of (d) and (e)

Table 1: Decomposition & Compression Results

	Total					**Compression Ratio**	**PSNR**
80 %	2986	% 1.4	% 5.2	% 33.1	% 60.3	10.5 : 1	30.14
70 %	2761	% 1.9	% 6.5	% 33.7	% 57.9	11.6 : 1	29.35

References

1. **El-Harby, A. A.; Behery, G.M** (2008): Qualitative image compression algorithm relying on quadtree. *ICGST-GVIP*, vol. 8, no. 3, pp. 41-50.
2. **Morvan, Y.; Farin, D.; de With, P.H.N** (2007): Depth-Image compression based on an R-D optimized quadtree decomposition for the transmission of

multiview images. IEEE International Conference on Image Processing ICIP 2007, pp. 105–108.

3. **Keissarian, F.** (2008): Using a novel variable block size image compression algorithm for hiding secret data. IEEE International Conference on Signal Image Technology & Internet based Syatem, SITIS 2008, pp. 285–292.
4. **Dai, W., *et al*** (2005): Adaptive block-based image coding with pre-/post-filtering, Proceeding of the Data Compression Conf., March 2005, pp. 73-82.