

2010

A new quad-tree segmented image compression scheme using histogram analysis and pattern matching

F. Keissarian

University of Wollongong in Dubai, farhadk@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/dubaipapers>

Recommended Citation

Keissarian, F.: A new quad-tree segmented image compression scheme using histogram analysis and pattern matching 2010.
<https://ro.uow.edu.au/dubaipapers/254>

A New Quad-tree Segmented Image Compression Scheme using Histogram Analysis and Pattern Matching

Farhad Keissarian

University of Wollongong in Dubai
farhadkeissarian@uowdubai.ac.ae

Abstract—In this paper, a new variable block-size image compression scheme is presented. A quadtree segmentation is employed to generate blocks of variable size according to their visual activity. Inactive blocks are coded by the block mean, while active blocks are coded by the proposed matching algorithm using a set of parameters associated with the pattern appearing inside the block. Both the segmentation and the pattern matching are carried out through histogram analysis of block residuals. The use of pattern parameters at the receiver together with the quadtree code reduces the cost of reconstruction significantly and exploits the efficiency of the proposed technique.

Keywords—component; image compression; quadtree decomposition, histogram analysis, block patterns.

I. INTRODUCTION

Natural images can be segmented into regions having widely different perceptual importance. Certain regions are critical to subjective evaluation quality, and relatively small quantization errors can perceptually have major degrading effect on the overall reproduction quality. Such segmentation of the image is useful for efficient coding of image data [1]. Traditional block-based image coding algorithms, such as vector quantization, transform coding, and block truncation coding techniques require the partitioning of the original image into a number of, usually square blocks of pixels which are then encoded as separate entities. In all these schemes, the block size is a fundamental design parameter. Variable-rate image coding that changes the coding resolution (in bits used per unit area) according to the local character and importance of the region to be coded, has become a new direction in image coding.

Quadtree decomposition is a simple technique for image representation at different resolution levels, which partitions an image into variable block size regions based on a quadtree structure. Studies have demonstrated that quadtree-based image segmentation can be effective and efficient mechanism for isolating blocks of distinct perceptual significance and thereby allowing different coding strategies that are perceptually suited to the individual segment categories [2]-[4]. It provides an effective compromise between the accuracy with which the region boundaries are determined and the overhead required to specify the segmentation information.

In this paper, we introduce a new quadtree-based image compression to achieve high compression ratios and preserve edge integrity. A novel classification scheme, which operates through histogram analysis of block residuals, is employed to determine whether the processed block requires further division. To preserve edge integrity, the block pattern matching coding technique, which we presented earlier in [5] is used to code high-activity regions. The collection of patterns, which is constructed over active blocks of 4×4 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.

II. THE PROPOSED COMPRESSION ALGORITHM

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.

A. 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. 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 and requires further segmentation. “Fig. 2” shows the histogram analysis of a 4x4 uniform 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. 3 depicts the histogram analysis of a 4x4 edge block.

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 [6] which introduces 17 possible combinations within a 16x16 image block. Only a 6-bit binary sequence $(D^0 D^1 d^1 d^2 d^3 d^4)$ is required to represent each splitting

mode as shown in Fig 4. The first bit D^0 indicates whether or not the 16x16 block is partitioned into four 8x8 blocks. If $D^0 = 1$, then the second bit D^1 indicates whether at least one 8x8 block is partitioned into four 4x4 blocks. If $D^1 = 1$, then the other four bits d^1, d^2, d^3, d^4 are required to indicate whether to split each 8x8 block into

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

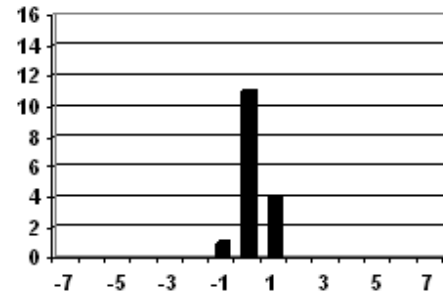
(b)

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

four 4x4 blocks or not. The amount of side information is

calculated as $\frac{wxh}{16 \times 16} \times 6$ bits for a $w \times h$ image size. The

uniform blocks of variable size are coded by the block mean, whereas a 4x4 edge block is coded by a set of parameters associated with the pattern appearing inside the block. Like the original BTC algorithm [7], 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.

B. Pattern Marthing Coding

A peak on the QRH 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

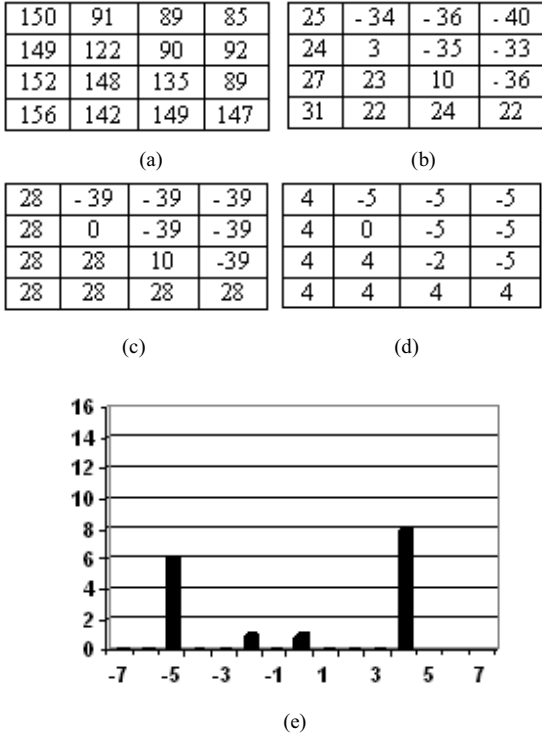


Figure 3. (a) Original edge block: sample mean = 125 (b) Residuals (c) quantized residuals, (d) quantizer's indexes, (e) QRH

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 , given as :

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

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 X_L and the high representative intensity X_H , respectively. If the two peaks are positioned at indexes j' and j'' , the two representative intensities are calculated as:

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

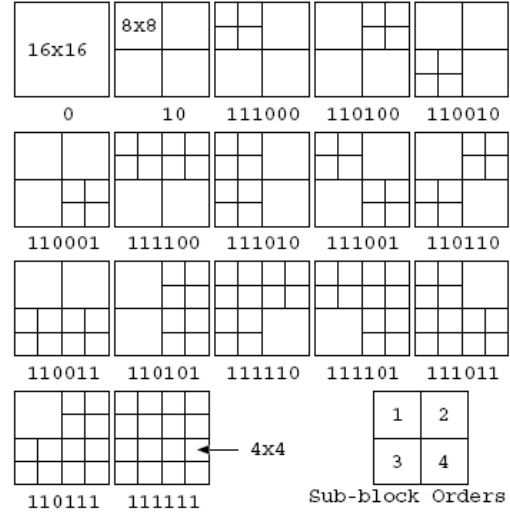


Figure 4: All possible 16 x 16 block partitioning modes and their associated binary representation

In "Fig. 3", $X_{mean} = 125$, and X_L and X_H are computed using Eq. 2 ; $X_L = 125 + (-39) = 86$ and $X_H = 125 + 28 = 153$. 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 , such that $W = W_0 \cup W_1$ and $W_0 \cap W_1 = \Phi$. 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, $B = \{b_1, b_2, \dots, b_{16} | b_i \in (0,1)\}$. By selecting the block mean as a threshold, the bit-pattern can be generated as :

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

where, $x_i \in W$ are the intensities of the pixels of the edge block. It is noted that, X_L and X_H 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 :

$$\begin{aligned} M &= \frac{X_H + X_L}{2} \\ l &= \frac{X_H - X_L}{2} \end{aligned} \quad (4)$$

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

$$b_i = \begin{cases} M + l & \text{if } b_i \in W_1 \\ M - l & \text{if } b_i \in W_0 \end{cases} \quad (5)$$

It should be noted that for a uniform block, since both representative intensities are the same as the block mean, therefore, $M = X_{mean}$ 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, P_k , $k = 0, 1, 2, \dots, N$. A set of 32 patterns shown in “Fig. 5”, 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 :

$$ms = \sum_{i=0}^3 \sum_{j=0}^3 \overline{(P_{ij} \oplus b_{ij})} \quad (6)$$

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 $\log_2^{64} = 6$ bits are transmitted. Each image block is therefore encoded by generating a triple (M, l, k) . “Fig. 6” 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 to generate (120, 34, 23) as 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 X_H . Therefore, the compression code of a 4x4 edge block requires 20 = (8+6+6) bits to be transmitted. For a 4x4 uniform block, the number of bits required to code the pair $(M, 0)$ is 9 = 8+1.

C. Post Processing

Since the pixels in each smooth block are represented by the block mean, the blocking effect between the boundaries of two blocks occurs unavoidably. To remove the blocking effect, a simple smoothing filter is provided. Since only smooth blocks are filtered, the edge blocks will not be blurred and edges will be preserved. The smoothing filter uses three

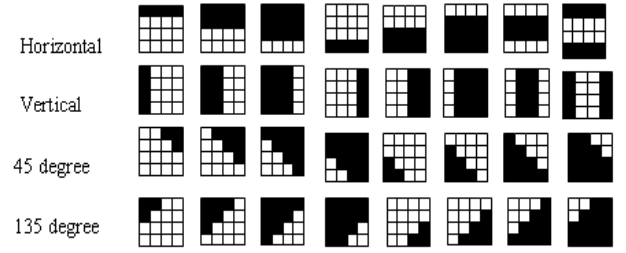


Figure 5: Set of 32 pre-defined patterns.

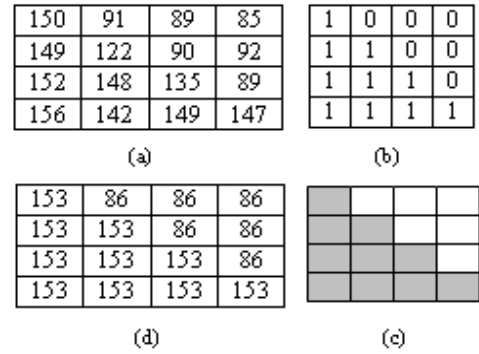


Figure 6. (a) Original edge block, (b) Bit-pattern (c) Matched pattern, (d) Reconstructed block

various masks sizes (3x3, 5x5, and 9x9) for the post processing of three different block sizes (4 x 4, 8 x 8, and 16 x 16). It adopts a simple average operation over the pixels in the area of the mask. The response of the smoothing operation is given by

$$x' = \frac{1}{W} \sum_{i=1}^l \sum_{j=1}^l (w_{i,j} * x_{i,j}) \quad (7)$$

where x' represents the smoothed gray value of the present pixel at which the center of the mask is located, $x_{i,j}$ denotes the gray level of (i, j) th pixel in the mask, $w_{i,j}$ denotes the weight of (i, j) th pixel defined as follows :

$$w_{i,j} = \begin{cases} 1 & \text{if } x_{i,j} \in \text{uniform} \\ 0 & \text{if } x_{i,j} \in \text{edge} \end{cases} \quad \text{and} \quad W = \sum_{i=1}^l \sum_{j=1}^l w_{i,j} \quad (8)$$

With the above operation, the gray level of each pixel in smooth blocks only changes slightly and is not affected by any edge pixels.

III. SIMULATION RESULTS

We have evaluated the performance of the proposed coding scheme through a computer simulation on a set of gray-level images including the image of “Lena” shown in “Fig.7a”. The test images are 8 bits per pixel, and the proposed technique was tested on images of 256 x 256 pixels and 512 x 512 pixels in size. The largest block size for 512x512 and 256x256 image sizes are 32x32 pixels and

16x16 pixels, respectively. The simulation platform is Microsoft Windows XP, Pentium III, and the proposed scheme is implemented using Matlab. Two performance matrices are used to measure the performance of the proposed compression schemes : compression ratio (bpp), and image quality. The peak signal-to-noise ratio (PSNR) is used to evaluate image quality of a compressed image generated by the proposed scheme.

“Fig. 7.b” and “Fig. 7.c” show the quadtree segmented images of ‘Lena’ for the image size of 256x256 and 512x512, respectively. “Fig. 7.d” and “Fig. 7.e” show magnified portions of the coded images for image size of 256x256 and 512x512, respectively. The quadtree overhead was computed as 0.023 and 0.010 for 256x256 and 512x512 image sizes, respectively. A compression ratio of 0.32 bpp at 30.15 dB, and a compression ratio of 0.28 bpp at 29.57 were achieved for the image size of 256x256, and the image size of 512x512, respectively.

Tables I and II show the representation of the splitting mode for both image sizes, and table III shows the decomposition results.

TABLE I
Splitting Mode for a 256 x 256 image

Splitting Blocks	D^0	D^1	d^1	d^2	d^3	d^4
16 x 16 block	0					
Four 8 x 8 blocks	1	0				
≥ 1 four 4 x 4 blocks	1	1	x	x	x	x

TABLE II
Splitting Mode for a 512 x 512 image

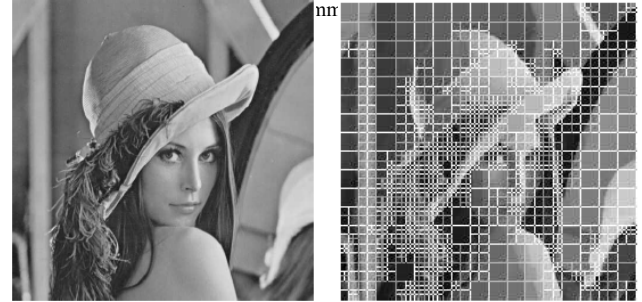
Splitting Blocks	D^0	D^1	D^2	d^1	d^2	d^3	d^4	d^5	d^6	d^7
32 x 32 block	0									
Four 16x16 blocks	1	0								
≥ 1 four 8 x 8 blocks	1	1	0							
≥ 1 four 4 x 4 blocks	1	1	1	x	x	x	x	x	x	x

TABLE III
Quad tree decomposition for the image of “Lena”

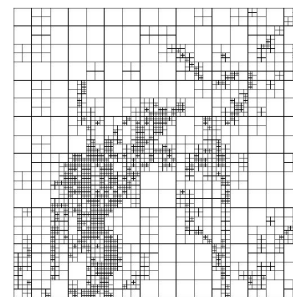
Image Size	Total no. of blocks	32 x 32	16 x 16	8 x 8	4 x 4 (uniform)	4 x 4 (edge)
256x256	1483	0	99	376	271	737
512x512	4340	95	280	641	1358	1966

Technologies and Engineering Science, Publisher Springer Netherlands, pp. 123-138, 2009.

- [3] El-Harby, G.M. Behery, “Qualitative image compression algorithm relying on quadtree”, ICGST-GVIP, vol. 8, no. 3, pp. 41-50., 2008.
- [4] K. sasazki, S. saga, J. Maeda, Y. Suzuki, “Vector quantization of images with variable block size”, Applied Soft Computing, vol. 8., no. 1, pp. 634-645, Jan. 2008.
- [5] F. Keissarian, “Novel quad-tree predictive image coding technique using pattern-based classification, *Proc. SPIE, Visual Communications and Image Processing (VCIP-2003)*, vol. 5150, pp. 1481-1490, June 2003, Lugano, Switzerland.
- [6] W. Dai, *et al*, “Adaptive block-based image coding with pre-/post-filtering, *Proceeding of the Data Compression Conf.*, March 2005, pp. 73-82.
- [7] E.J. Delp and O.R. Mitchell, “Image compression using block



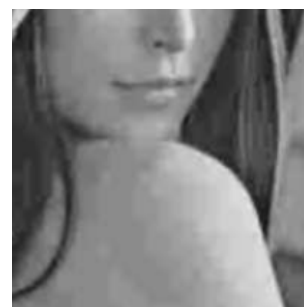
(a) Original Image (b) Quadtree segmented image (256x256)



(c) Quadtree segmented image (512x512)



(d) Coded image at 0.32 bpp



(e) Coded image at 0.28 bpp

Figure 7. The compression and embedding results for the image of “Lena”

REFERENCES

- [1] J. Vaisey, A. Cersho, “Image compression with variable block size segmentation”, IEEE Trans. on Signal Processing, vol. 40, no. 8, pp. 2040-2060.
- [2] Y. 8. T. Y. Sung, H. C. Hsin, “Quad-Tree based adaptive wavelet packet image coding”, Book chapter, Trends in Communication