Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 13 (13): 2589-2593, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2589.2593

A Novel Image Representation Model for Color Image Compression

[1]Guangwei Wang, [2]Xinggang Zhang, [1]Cenggang Xiong and [2]Yihua Lan
[1]School of Computer and Information Science, Hubei Engineering University, Xiaogan, 432000, Hubei Province, China
[2]School of Computer and Information Technology, Nanyang Normal University, Nanyang, 473061, Henan Province, China

Abstract: In this study, we propose a novel non-symmetry and anti-packing image representation model (NAM). NAM is a hierarchical image representation method and its aim is less data amount and faster operation. Based on block-wise LS linear predictor method, a color-coded bit-plane binary image NAM representation method is presented to compress the color image. The compression performance has been further improved. Experimental results show that our compression scheme is clearly better from data compression point of view than other methods.

Key words: Image representation, quadtree, image compression, gray coded, NAM mode

INTRODUCTION

The hierarchical data structures are very important region representation methods in image compression, computer vision, image processing and pattern recognition. The representation with hierarchy have not only can better space compactness, but also many operations and computations algorithms can be fast implemented (Chan, 2004; Zheng *et al*., 2007a).

The quadtree is a very typical hierarchical representation and has been widely applied in computer visualization, robotics, computer graphics, image processing and pattern recognition (Gong and Yang, 2004). The principle of the quadtree-based data structure is simple. An image region can include black and white pixels, where objects are marked with black pixels. The image region is recursively decomposed into quadrants and sub-quadrants until all sub-quadrants include either all black or all white pixels. The earlier quadtree representation was based on the pointers. Later, in order to further reduce the storage space, Gargantini removed the pointers and put forward a linear quadtree representation method. Untile now, the linear quadtree is still the most popular representation method and it is widely studied and applied in many fields. More and more theories and applications of the quadtree representation could be found in (Bruin *et al*., 2004).

Inspired by the concept of the packing problem, we present a novel image presentation, namely Non-symmetry and Anti-packing pattern representation Model (NAM) and propose a binary image representation algorithm using the NAM with rectangle sub-pattern. Until now, we have implemented many popular algorithms and applications based on NAM image (Zheng *et al*., 2007b; Wei *et al*., 2009).

In this study, we extended the content of gray-coded bit-plane binary image NAM representation method in (Chen *et al*., 2011; Wang *et al*., 2010), a block-wise LS linear predictor method (Baligar *et al*., 2003) is adapted to compression the color image and gives an error image. Then, we use image NAM representation algorithm to compression the error image. By comparing the linear quadtree and the proposed NAM representation for binary image, it is proved that NAM is much more effective than the linear quadtree(Gong and Yang, 2004; Bruin *et al*., 2004) at the aspect of data storage.

In this section, we briefly describe the Non-symmetry and Anti-packing pattern representation Model (NAM). For more details of NAM and its encoding/decoding algorithms please refer to Zheng *et al*. (2007a,b).

In addition, the theoretical analyses and experimental results show that the compression performance of our method is superior to CAPEC and JPEG-LS (Golchin and Paliwal, 1997; Weinberger *et al*., 2000).

The first aim of this study is to set out an abstract description of NAM. The second is to show that the binary image NAM representation algorithm is more effective in compression error image than linear quadtree.

Corresponding Author: Cenggang Xiong, School of Computer and Information Science, Hubei Engineering University, 432000, Xiaogan, 432000, Hubei Province, China Tel: +086 18672290737

DESCRIPTION OF NON-SYMMETRY AND ANTI-PACKING PATTERN REPRESENTATION MODEL (NAM)

Abstract description of NAM: The Non-symmetry and Anti-packing pattern representation Model (NAM) is an anti-packing problem. The idea of the NAM can be described as follows: Giving a packed pattern (container) and n predefined sub-patterns (objects) with different shapes, pick up these sub-patterns (objects) from the packed pattern (packed container) and then represent the packed pattern (container) with the combination of these sub-patterns (objects).

Suppose an original pattern is Γ and two reconstruction non-distortion and distortion patterns Γ' and Γ'', respectively. Then, the NAM is either a non-distortion transform model from Γ to Γ' or a distortion one from Γ to Γ''. The procedure of the transform can be written as follows:

$$\Gamma' = T(\Gamma),\ \Gamma'' \approx T(\Gamma)$$

where T(.) is a transform or a coding function.

The procedure of the non-distortion coding can be obtained by the following expression:

$$\Gamma' = \bigcup_{j=1}^{k} p_j(v, A \mid A = \{a_1, a_2, \cdots, a_{m_j}\}) + \varepsilon(d)$$

where, Γ' is the reconstruction pattern; P is a set of some predefined sub-patterns; k is the total number of the sub-pattern instances; p_j is the j^{th} sub-pattern instance; v is the value of p_j; A is a parameters set of the sub-pattern p_j; ε(d) is a residue pattern.

By taking a simple binary image for example, Fig. 1 illustrates the idea of the NAM. Two sub-patterns, i.e., a rectangle and a triangle shown in Fig. 1b, are used to pack the original image shown in Fig. 1a. Fig. 1c illustrates an asymmetric division of the original image and Fig. 1d shows the sub-patterns corresponding to this division. In this example, we use five instances of the rectangle sub-pattern labeled 1, 3, 4, 5 and 6 and two instances of the triangle sub-pattern labeled 2 and 7, to represent the original image.

Description of the encoding algorithm: For a given binary image I, an encoding result of I is stored into a queue Q = (p_1, p_2,... p_k), where $p_i (1 \le i \le k)$ denotes the i^{th} instances of rectangle sub-pattern and k is the total numbers of instances in Q. The encoding part of our algorithm has the following steps:

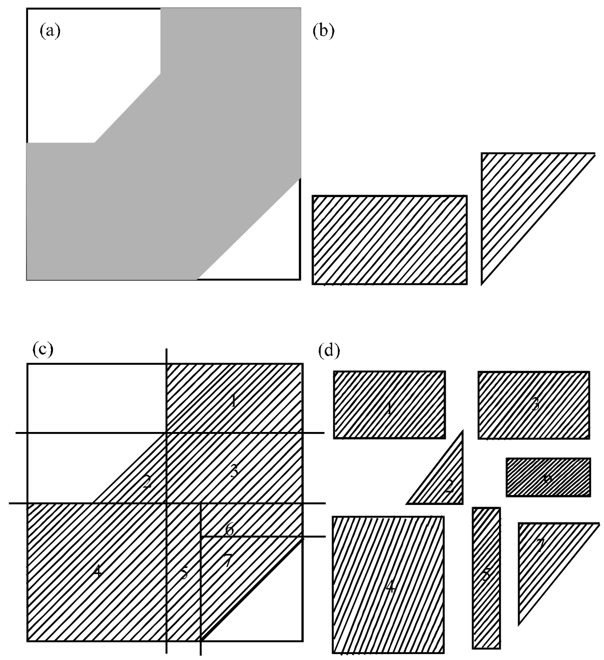

Fig. 1(a-d): Examples of NAM (a) Original image, (b) Predefined sub-pattern, (c) Non-symmetric division and (d) Sub-patterns of NAM

- **Step 1:** Set every pixel in image I unmarked
- **Step 2:** Search the first unmarked black pixel sp(x, y) of image I according raster scanning order and let sp (x, y) be the beginning point (top and left corner) of a new rectangle sub-pattern p
- **Step 3:** Find out the biggest sub-pattern p in term of the areas of the sub-pattern and mark the sub-pattern in the image I
- **Step 4:** Record the parameters of the found sub-pattern, i.e., Q←(sp, width, height)
- **Step 5:** Repeat step 3 to 5 until there are no unmarked pixel in the image I
- **Step 6:** Output the encoding result Q

Storage structure and data analysis: In this section, by taking a rectangle sub-pattern for example, the storage structure and the data amount are analyzed. Suppose the size of binary image is $2^n \times 2^n$. The storage structure of rectangle can be denoted by the expression p = (r|r = (sp, width, height)). The start point sp is stored as a relative value Δsp, i.e., $\Delta sp = sp_i - sp_{i-1}$. Therefore, the length of Δsp is n bits in the sight of statistical concept. In fact, if the length of Δsp is greater than n, then one record can be divided into two records. Also, the maximal length of width or height is n/2 bits. Therefore, storing a rectangle takes up 2n bits.

Let the total data amount of the binary image I be H_{nam} when I is represented by NAM encoding algorithm. We can write H_{nam} as follow:

$$H_{nam} = 2n\,N_{nam}$$

where N_{nam} is the total number of the sub-pattern instances.

For the method of the linear quadtree, storing a record of a node takes up 2n+1gn bits for a binary image (Zheng *et al*., 2007a, b). Suppose the total data amount of the image I is H_{lqt} when I is represented by the linear quadtree. We can write H_{lqt} as follow:

$$H_{lqt} = (2n+1gn)\,N_{lqt}$$

where N_{lqt} is the total nodes number of the linear quadtree.

Let n denote the ratio of N_{lqt} to N_{nam} and φ denote the ratio of H_{lqt} to H_{nam}. Then:

$$\eta = \frac{N_{lqt}}{N_{nam}}$$

$$\varphi = \frac{(2n+\lg n)N_{lqt}}{(2n)N_{nam}}$$

The theoretical analysis in this section proves that our representation method using the NAM can reduce the data storage much more effectively than that using the popular linear quadtree.

COMPRESSION SCHEME

Block-wise LS-based prediction: For simplicity, we first explain the LS-based prediction. Suppose we want to construct a prediction for x(t) as a linear combination of its previous N samples:

$$\hat{x}(i) = \sum_{i=1}^{K} \theta(k)x(i-k)$$

This is called N^{th} order linear predictor.

The LS approach enables us to find the optimal coefficients θ(k) such that the sum of squares of difference between x(i) and $\hat{x}(t)$ is minimized:

$$S = \sum_{i=1}^{n} [\hat{x}(n-i) - x(n-i)]^2$$

This process can be regarded as training process for the predictor to adapt itself to the characteristics of the given signal samples. In case of images, the training window should be taken from the neighboring area of current pixel. Chosen the neighboring pixels in causal order helps in decoding by constructing the same predictor without any overhead information.

Predictor order and training window size are usually difficult to determine. Here, we adapt the block-wise LS-based linear prediction scheme proposed in (Weinberger *et al*., 1999) for solving the problem. It divides the test image into a number of blocks of size B×B and finds the prediction coefficients θ (k) using Singular Value Decomposition for each block.

The SVD approach gives satisfactory results and we used the SVD method to find prediction coefficients of each block. SVD is the method of choice for solving most linear least squares problems. SVD methods are based on the following theorem of linear algebra. Any M×N matrix A whose number of rows M is greater than or equal to its number of columns N, can be written as the product of an M×N column-orthogonal matrix U, an N×N diagonal matrix W with positive or zero elements and the transpose of an N×N orthogonal matrix V:

$$(A) = (U)\begin{pmatrix} w_1 & & \\ & \cdots & \\ & & w_N \end{pmatrix}(V^T)$$

The matrices U and V are each orthogonal in the sense that their columns are orthonormal, i.e.:

$$(U^T)(U) = (V^T)(V) = (1)$$

In a block, we consider the number of pixels more than the number of prediction coefficients. The optimal coefficients θ will be satisfy:

$$(A).\ (\theta) = b$$

In such cases the least-square solution vector which is a set of prediction coefficients is given by:

$$(\theta) = (V).\ (\mathrm{diag}(1/w_j)).\ (U^T).\ b$$

The number of prediction coefficients per block is equal to the number of neighboring pixels considered to predict the value of each pixel, is also equal to the order of the linear predictor.

In this prediction scheme, the prediction for each pixel is formed by using a set of linear prediction coefficients of the block to which the current pixel belongs. Predicted value of each pixel is subtracted from the actual value of the current pixel to get an error image. An error image is processed in next section.

Gray-coded bit-plane decomposition: In this section, we will introduce a method of gray-coded bit-plane decomposition (GC-BPD) to decompose an image to several binary images. Then, these binary images will be represented by NAM representation method to decrease the image storage room.

Let f denote a gray-level image in which each pixel value is represented by L bit. The gray-level image can be represented as:

$$f = a_{L-1}2^{L-1} + a_{L-2}2^{L-2} + \cdots + a_1 2^1 + a_0 2^0$$

where, ai ($0 \le i \le L-1$) denotes i^{th} bit of f, i.e., 0 or 1.

The gray coded are defined as:

$$b_i(x,y) = \begin{cases} a_i(x,y) \oplus a_{i+1}(x,y) & \text{if } 0 \le i \le L-2 \\ a_i(x,y) & \text{if } i = L-1 \end{cases}$$

where, b_i denotes the i^{th} bit plane of the gray coded image g and $\oplus$ denotes the bit-wise exclusive OR operation.

This gray code has the unique property that successive code words differ in only one bit position. Thus, small changes in gray level are less likely to affect all L bit-planes. The method of GC-BPD can effectively reduce the complexity of binary bit plane image and can make the total sub-patterns number much fewer than original binary image. Therefore, the method can greatly improve the representation efficiency of a gray-level image.

Implementation of the scheme: First, we convert the color image into a RGB format. The RGB image is divided into three images according to its R, G and B values and each one is a 8-bit gray image. The R gray image is subtracted from the G gray-image and B gray image, respectively and two difference image (R-G image and R-B image) with flat background are obtained.

Then, the R gray image, R-G image and R-B image are processed by Block-wise LS-based prediction method and three error images are obtained.

Finally, we use Gray-coded bit-plane decomposition method to convert the three error images to several gray-coded bit-plane binary images. These gray-coded bit-plane binary images are further compressed using NAM representation method. The implementation of the proposed scheme consists of the following encoding and decoding parts.

Encoding part

- **Step 1:** Build the R gray image and two difference image R-G and G-B. Use Step 2-7 to code the three images, respectively
- **Step 2:** Divide the three images into a number of blocks of size B×B, respectively
- **Step 3:** Find the prediction coefficients using SVD for each block
- **Step 4:** Use the prediction image to find error image
- **Step 5:** Decompose the error image to several binary images using GC-BPD method
- **Step 6:** Use NAM representation method to compress these binary images
- **Step 7:** The prediction coefficients and compressed error image are stored in a file to get a compressed image

Decoding part

- **Step 1:** Decode the compressed error image
- **Step 2:** Use prediction coefficients to get the prediction image
- **Step 3:** Use the prediction image and error image to obtain the R gray image and two difference image R-G and G-B
- **Step 4:** Restore the RGB image

In practice for a 16×16 block and forth order prediction, we get the best result.

EXPERIMENT RESULT

We have implemented our scheme on 8-bit gray level images of size $2^9 \times 2^9$ and made a comparison with other methods. In the experiment, we observe that the forth order, block size of 16×16, NAM binary representation approach for the test images outperforms the gray quadtree approach, JPEG-LS method and CAPEC scheme in terms of compression ratio.

Table 1 shows that the proposed method performs better in terms of compression performance compared to CAPEC and JPEG-LS methods.

The second column of Table 1 gives the results obtained from CAPEC; the third column shows the results of the JPEG-LS method. The columns namely quadtree and NAM indicate that the absolute error image is compressed using the linear quadtree and NAM method on gray-coded bit-plane.

Table 1: Comparison of compression results (in bits per pixel), test images are 512×512, 24bits/pixel

Image	CAPEC	JEPEG-LS	Quadtree	NAM
Lena	4.060	4.245	3.688	3.461
Airplane	3.610	3.792	3.459	3.298
Barbara	4.430	5.004	4.117	3.828
Peppers	4.430	4.719	4.253	4.136
Boat	4.230	4.252	3.928	3.658
Elaine	4.820	4.898	4.871	4.572
Average	4.263	4.485	4.052	3.825

CONCLUSION

Based on the non-symmetry and anti-packing pattern representation model, a binary image representation algorithm was proposed. Here, using a block-wise linear predictor method for color level image gave us an error image. Then, the error image was transformed to gray-coded bit-plane binary image and compressed by binary image NAM representation algorithm. This application of NAM binary image representation for error image was further improve the compression performance.

The application of NAM representation method opens a new research area and is valuable for the theoretical research and business foreground, such as saving the storage room, decreasing the transmission time, enhancing processing speed, matching pattern and so forth.

ACKNOWLEDGMENTS

The authors wish to acknowledge the support of the National Natural Science Funds of China under Grant No. 61370092, Young talent program in Hubei Education Department (Grant No. Q20132709), Industry-Academia Cooperation Innovation Fund Projects of Lianyungang (Grant No. CXY1203) and Technology Research and Development Program of Lianyungang (Grant No. SH1223).

REFERENCES

Baligar, V.P., L.M. Patnaik and G.R. Nagabhushana, 2003. High compression and low order linear predictor for lossless coding of grayscale images. Image Vision Comput., 21: 543-550.

Chan, Y.K., 2004. Block image retrieval based on a compressed linear quadtree. Image Vision Comput., 22: 391-397.

Chen, C., G. Wang and M. Sarem, 2011. A new non-symmetry and anti-packing model and its application to image contrast enhancement. Comput. Electr. Eng., 37: 669-680.

De Bruin, S., A.J.W. de Wit, P.A.J. van Oort and B.G.H. Gorte, 2004. Using quadtree segmentation to support error modelling in categorical raster data. Int. J. Geog. Inform. Sci., 18: 151-168.

Golchin, F. and K.K. Paliwal, 1997. Classified adaptive prediction and entropy coding for lossless coding of images. IEEE Int. Conf. Image Process., 3: 110-113.

Gong, M. and Y.H. Yang, 2004. Quadtree-based genetic algorithm and its applications to computer vision. Patter Recognit., 37: 1723-1733.

Wang, G., C. Chen and K. Kang, 2010. Using non-symmetry and anti-packing image representation model for scale images compression. Proceedings of the 2nd IEEE International Conference on Network Infrastructure and Digital Content, September 24-26, 2010, Beijing, pp: 332-336.

Wei, H., X. Fan, C. Chen and M. Sarem, 2009. Accurate computation of zernike moments using non-symmetry anti-packing image representation. Asia-Pacific Conf. Inform. Process., 2: 156-159.

Weinberger, M.J., G. Seroussi and G. Sapiro, 1999. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Trans. Image Process., 9: 1309-1324.

Zheng, Y., C. Chen and M. Sarem, 2007a. A novel algorithm for triangle non-symmetry and anti-packing pattern representation model of gray images. Proceedings of the 3rd International Conference on Intelligent Computing, August 21-24, 2007, Qingdao, China, pp: 832-841.

Zheng, Y., C. Chen, H. Wei and S. Fang, 2007b. A new non-symmetry and anti-packing model with k-lines and its application to color image representation. Proceedings of the International Conference on Image Processing, Computer Vision, September 16-October 19, 2007, San Antonio, Texas, USA., pp: 526-532.