Journal of Mobile Multimedia, Vol. 5, No.2 (2009) 125-139

A NEW LOSSY AND LOSSLESS IMAGE REPRESENTATION BY USING NON-SYMMETRY AND ANTI-PACKING MODEL WITH RECTANGLES FOR GRAY IMAGES

MUDAR SAREM

School of Software Engineering,HuazhongUniversity of Science and Technology, China
mudar66@hotmail.com

YUNPING ZHENG, CHUANBO CHEN

School of Computer Science and Technology,HuazhongUniversity of Science and Technology, China
{zypdragon, chuanboc}@163.com

Received March 10, 2008
Revised November 6, 2008

With the rapid development of mobile communication systems, demands for the transmission of multimedia information are increasing day by day. The effective transmission of images can be increased by getting smaller image file that is obtained by compression. However, image quality is often sacrificed in the compression process. Therefore, there is a need to represent images with less data storage without sacrificing the image quality. In this paper, inspired by the concept of the packing problem, we present a new Non-symmetry and Anti-packing Model with Rectangles (NAMR) for lossy and lossless image representation in order to represent the pattern more effectively and flexibly. Also, in this paper, we propose an algorithm of NAMR and analyze the data amount of this algorithm. The theoretical analyses and experimental results presented in this paper show that when the representation method of NAMR is compared with that of the popular linear quadtree, not only can the former reduce the data storage much more effectively than the latter in lossless case, but also the former has a better reconstruction quality than the latter in lossy case.

Key words: Image representation, anti-packing problem, linear quadtree, non-symmetry and anti-packing model, rectangle subpattern
Communicated by I. Khalil

1 Introduction

Designing efficient image representation is an important issue in different fields of multimedia information. The hierarchical quadtree representation methods for an image have been widely applied in computer visualization, robotics, computer graphics, image processing, and pattern recognition [8, 11, 10]. In the traditional quadtree model, 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 subquadrants until all subquadrants include either all black or all white pixels. In a two-dimensional space, data points can be partitioned by the same regular hierarchical decomposition method so that the number of points inside any quadrant and subquadrant is within a threshold. The quadtree can be stored entirely as a region quadtree.

Alternatively, we can only store, in a linear quadtree structure, the encodings of paths from the root quadrant to individual subquadrants that contain all black pixels or have at least one data point [6, 12, 14, 16]. Maximal quadtree blocks or quadrants can be obtained through the regular hierarchical decomposition to save the data storage. Therefore, the hierarchical representation of images, which has not only good compatibility but also fast processing speed, is an effective representation method. More theories and applications based on the hierarchical quadtree representation can be found in [7, 9, 5, 15].

However, although the hierarchical representation methods stated above have many merits, they put too much emphasis upon the symmetry of image segmentation. Therefore, they are not the optimal representation methods.

Packing problems are optimization problems that are concerned with finding a good arrangement of multiple objects in a larger containing region without overlapping [13]. The usual objective of the allocation process is to maximize the material utilization and hence to minimize the wasted area. Packing problem includes various kinds of issues which have yielded many significant theories and valuable applications such as loading of tractor trailer trucks and cargo airplanes and ships [2, 3, 8, 1]. However, the subject of the pattern representation problem is to find the representation methods of information entities. It seems that the packing problem has no inevitable relations with the pattern representation problem. Although the relations between two things apparently do not exist, from the point of view of the philosophy, it is not necessary that the two things don't have intrinsic relations with each other. As far as the packing problem and the pattern representation problem are concerned, the similarities between them are embodied at least in three aspects. Firstly, both of them are the formalization systems of the information representation. Secondly, they have similarities in describing the formalization systems. The packing problem is a NP-hard problem in computational theory. In a simplified view, the packing problem is to fill in a container with some given objects according to some rules and objectives. The pattern representation is to represent a given information framework such as the image, text, voice, and video by using the appropriate subpatterns according to some rules and objectives. Finally, the entities of the two formalization systems can correspond one by one. For example, the container and the object in the packing problem correspond to the pattern and the subpattern in the pattern representation problem, respectively. Therefore, the similarities between the packing problem and the pattern representation problem do exist, and the ideas of the two problems can be used to reference each other at least.

In our previous work [4], we proposed a novel representation algorithm based on Binary-bit Plane Decomposition by using the Non-symmetry and Anti-packing pattern representation Model (BPD-based NAM) for gray images. Also, in another work [17], a novel algorithm was put forward for the Triangle Non-symmetry and Anti-packing pattern representation Model (TNAM) for gray images that have no obvious characteristics of blocks where the predefined subpatterns are triangles.

In this paper, inspired by the concept of the rectangle packing problem, we present a new Non-symmetry and Anti-packing Model with Rectangles (NAMR) for lossy and lossless image representation. Also, we propose an algorithm of NAMR. By comparing the proposed algorithm with that of the popular linear quadtree, the theoretical and experimental results presented in this paper show that the former not only can reduce the data storage much more effectively than the latter in lossless case, but also it has a better reconstruction quality than the latter in lossy case.

The remainder of this paper is organized as follows: In section 2, we first illustrate the idea of the NAMR, and then present a new Non-symmetry and Anti-packing Model with Rectangles (NAMR) for lossy and lossless image representation. In section 3, we first introduce a method of binary Bit-Plane Decomposition (BPD), which can decompose a gray image to several binary images. Then, an algorithm of NAMR is proposed. In section 4, we analyze the total data amount of the proposed algorithm. Next, we test our algorithm with the Matlab 6.5 software and analyze experimental results in section 5. Finally, in section 6, we conclude our study and discuss the future work.

2 Non-symmetry and Anti-packing Pattern Representation Model with Rectangles (NAMR)

2.1 Idea of the NAMR

The packing problem includes various kinds of issues which have yielded many significant theories and valuable applications. It can be briefly described as follows: Giving a container and *n* objects with different shapes, when putting these objects in the container, if these objects can not be held by the container, then a negative answer is given; otherwise, if these objects can be held by the container, then a positive answer is given and the concrete coordinates of where these objects should be put are designated.

The Non-symmetry and Anti-packing Model with Rectangles (NAMR) is an anti-packing problem. The idea of the NAMR can be described as follows: Giving a packed pattern (a packed container) and *n* predefined rectangle subpatterns (*n* predefined rectangle objects) with different shapes, pick up these rectangle subpatterns (rectangle objects) from the packed pattern (the packed container) and then represent the packed pattern (the packed container) with the combination of these rectangle subpatterns (rectangle objects).

The concept of the non-symmetry in this model means that the structure of anti-packing is asymmetrical. The reason why the non-symmetry, which is relative to the symmetry of the hierarchical structure, is put forward is that the non-symmetry has a capability of representing a packed pattern with the least number of subpatterns from the point of view of the packing problem. Therefore, the representation method of the NAMR has the capability of making a pattern achieve the best representation efficiency which cannot be achieved by the traditional hierarchical quadtree representation methods.

2.2 Description of the NAMR

The new NAMR for lossy and lossless image representation can be described as follows:

Suppose an original pattern is Γ, two reconstruction non-distortion and distortion patterns are Γ' and Γ'', respectively. Then, the NAMR 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:

$$\Gamma' = \mathrm{T}(\Gamma)\,,\, \Gamma'' \approx \mathrm{T}(\Gamma)\,, \tag{1}$$

where T( ) is a transform or coding function.

The procedure of the non-distortion encoding can be obtained by the following expression.

$$\Gamma' = \bigcup_{j=1}^{n} p_j(v, A \mid A = \{a_1, a_2, \cdots, a_{m_i}\}) + \varepsilon(d)\,, \tag{2}$$

where Γ' is the reconstruction pattern; $P = \{p_1, p_2, \cdots, p_n\}$ is a set of some predefined rectangle subpatterns; n is the type number of the rectangle subpatterns; p_j is the j^{th} subpattern $(1 \le j \le n)$; v is the value of p_j; A is a parameters set of p_j; $a_i (1 \le i \le m_i)$ denotes the shape property of p_j; m is the serial number of p_j; $\varepsilon(d)$ is a residue pattern; and d is a threshold of $\varepsilon(d)$.

If the residue pattern $\varepsilon(d)$ is removed from the non-distortion pattern, then the distortion pattern can be obtained as follows:

$$\Gamma'' = \bigcup_{j=1}^{n} p_j(v, A \mid A = \{a_1, a_2, \cdots, a_{m_i}\}) \,. \tag{3}$$

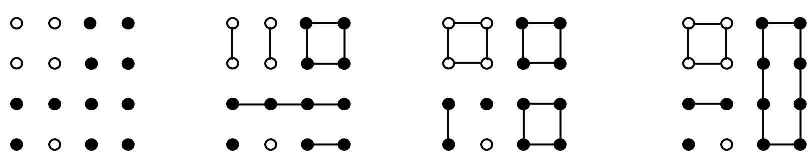
It is obvious that the following expression is true.

$$\Gamma \propto \Gamma' = \Gamma'' + \varepsilon(d)\,. \tag{4}$$

The representation of the NAMR is suitable for the image, text, audio, and video patterns and is a general, abstract, and pioneer representation method. However, in this paper, we focus on the image pattern which can be defined as follows: If an image can be denoted by a two-dimensional array f of size $2^n \times 2^n$ and $f = \{f(x,y)\}$, where $f(x,y)$ is the gray value of a pixel and (x,y) is a coordinate of the pixel, then $f = \{f(x,y)\}$ is called image pattern.

2.3 Example of the NAMR

By taking a rectangle subpattern as an example, the idea of the NAMR is illustrated by the following Fig.1. Without loss of generality, suppose that Fig.1 (a) is a binary image of size $2^2 \times 2^2$, i.e., the binary image includes sixteen black and white pixels, where objects are marked with black pixels. By applying different anti-packing strategies on Fig.1 (a), we can obtain different anti-packing results, some of which can be shown in Fig.1 (b), (c), and (d). As a matter of fact, the number of the anti-packing results is far more than three; we only select the three representative results.



(a) (b) (c) (d)

Figure 1 Idea of the NAMR.

In fact, the total subpatterns numbers of Fig.1 (b), (c), and (d) are seven, six, and five, respectively. Thus, Fig.1 (d) is the best anti-packing result among them from the point of view of the subpatterns number.

By this example, we know that the anti-packing algorithm is not exclusive, and that different anti-packing algorithms can achieve different representation efficiencies. Therefore, it is imperative that the anti-packing algorithm should be optimized in order to achieve the best representation efficiency.

As a matter of fact, in order to achieve the least rectangle subpatterns number for a binary image, we must make the found rectangle subpatterns as big as possible according to the area of the rectangle sub-patterns. In our algorithm of NAMR, we adopt this rule in order to achieve the best representation efficiency.

3 An algorithm of NAMR for Lossy and Lossless Image Representation

In this section, we first introduce a method of binary Bit-Plane Decomposition (BPD), which can decompose a gray image to several binary bit-plane images. Then, the K-Code transform rules are introduced since our algorithm of NAMR strongly depends on them. Then, the description of our algorithm (the encoding and decoding parts) of NAMR for lossy and lossless image representation is presented. And finally, we analyze the storage structure of NAMR for the gray images.

3.1 Method of binary bit-plane decomposition

A binary bit-plane image is defined as follows: It is a binary image which is made up of i^{th} bit-plane of a gray image with a bit depth *m*, where i=0,1,…,m-1. In this paper, BP_i refers to i^{th} bit-plane and BP_i (x,y) is the color of the address (x,y), i.e., 0 or 1 (*black* or *white*). Although the meaning of 0 or 1 for a binary bit-plane image is different from that for an ordinary binary image, we still think their meaning is the same since 0 or 1 represents *black* or *white* in both a binary bit-plane image and an ordinary binary image. The complexity of a gray image is defined as: $Cp = N_{LQT}/N_f$, where N_{LQT} is the total nodes number of the gray image which is represented by the method of the linear quadtree and N_f is the total pixels number of the gray image.

Without loss of generality, suppose *MP* denotes a gray image with a bit depth *m*. The following two theorems give some properties and a concrete method of BPD for a gray image.

Theorem 1: Let *MP* denote a gray image with a bit depth *m*. $BP_i(x,y)$ denotes the color of the address (x,y), i.e., 0 or 1. Stipulate that if $BP_i(x,y)$ =0, then the pixel value of $BP_i(x,y)$ is 0, and that if $BP_i(x,y)$ =1, then the pixel value of $BP_i(x,y)$ is 1. Then, the following expression is true.

$$MP = \sum_{i=0}^{m-1} 2^i BP_i \ , \tag{5}$$

where BP_i denotes i^{th} bit plane of *MP*, with i=0,1,…, m-1.

Proof: Given a pixel $MP(x,y)$ of *MP*, let the gray value of $MP(x,y)$ be g. We can write a binary form of g as follows:

$$g = (b_{m-1}b_{m-2}\cdots\cdots b_0)_2 \ , \tag{6}$$

i.e.,

$$\begin{aligned} g &= 2^{m-1}b_{m-1} + 2^{m-2}b_{m-2} + \cdots + 2^0 b_0 \\ &= \sum_{i=0}^{m-1} 2^i b_i = MP(x,y) \, . \end{aligned} \tag{7}$$

According to the stipulation of theorem 1, the value of b_i is just that of $BP_i(x,y)$. Since $MP(x,y)$ is any pixel of *MP*, and b_i, i.e., $BP_i(x,y)$ is also any pixel of BP_i, according to (7), the expression (5) must be true. .

Proof is finished hereon.

After we have known the relations between *MP* and BP_i, we can easily decompose a gray image to *m* binary-bit plane images. The following theorem 2 describes the method of BPD.

Theorem 2: Given a binary-bit plane number *i* and a pixel *MP*(*x*,*y*) of *MP*, let the gray value of *MP*(*x*,*y*) be *g*. If the following expression is true:

$$2^i \le g \textbf{ mod } 2^{i+1} < 2^{i+1},\ 0 \le i < m. \tag{8}$$

Then, $BP_i(x,y)=1$, otherwise, $BP_i(x,y)=0$.

Proof: Suppose that a binary form of *g* is $g=(b_{m-1}b_{m-2}\ldots b_0)_2$. Three cases are considered as follows:

Case 1: $2^i \le g < 2^{i+1}$

In this case, it is obvious that $g \textbf{ mod } 2^{i+1} = g$. Therefore, the expression (8) is true and at the same time $b_{m-1}, \ldots, b_{i+1}$ are all 0, b_i is 1, and $b_{i-1}, \ldots, b_0$ are either 0 or 1. Since $b_i=1$, according to theorem 1, we can deduce $BP_i(x,y)=1$;

Case 2: $0 \le g < 2^i$

In this case, it is also obvious that $g \textbf{ mod } 2^{i+1} = g$. However, the expression (8) is not true and at the same time $b_{m-1}, \ldots, b_{i+1}$ are all 0, b_i is 0, and $b_{i-1}, \ldots, b_0$ are either 0 or 1. Since $b_i=0$, according to theorem 1, we can deduce $BP_i(x,y)=0$;

Case 3: $2^{i+1} \le g < 2^m$

In this case, $g \textbf{ mod } 2^{i+1} = (0\ldots0b_ib_{i-1}\ldots b_0)_2$. If $b_i=1$, then it is the case 1, i.e., the expression (8) is true and $BP_i(x,y)=1$. If $b_i=0$, then it is the case 2, i.e., the expression (8) is false and $BP_i(x,y)=0$.

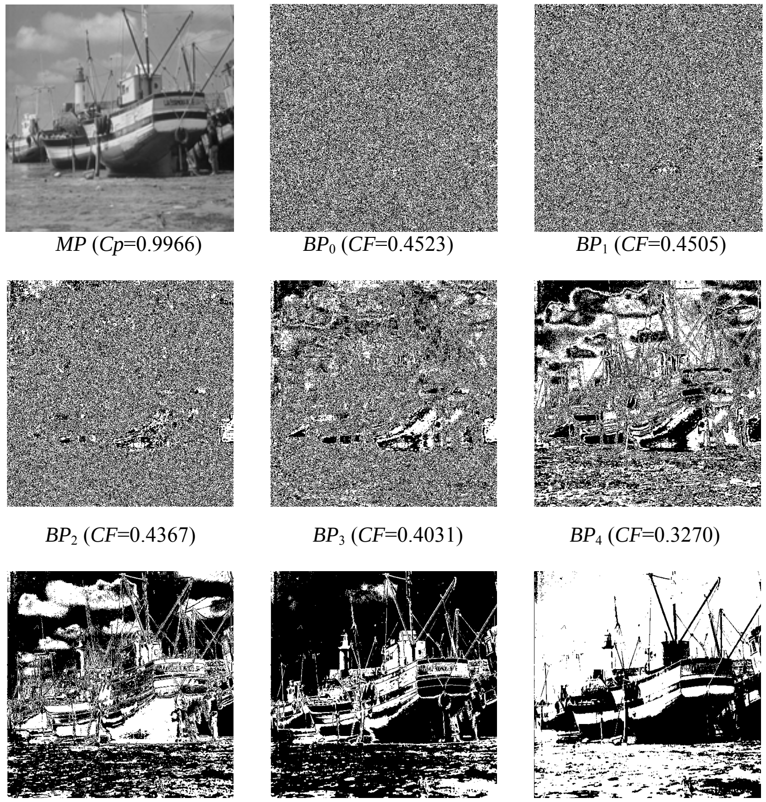
Since theorem 2 is always true in the above three cases, this shows that theorem 2 is true.

Proof is finished hereon.

Fig.2 shows an original gray image *MP* with a bit depth *m*=8 and eight binary bit-plane images $BP_i(0 \le i \le 7)$. Also, the image complexity *Cp* and *CF* for these images are given. The complexity of a binary image is defined as: $CF = N_{LQTB}/N_{fb}$, where N_{LQTB} is the total number of the black nodes of the binary image which is represented by the method of the linear quadtree and N_{fb} is the total pixels number of the binary image.

The following Fig.3 is a plot of the image complexity against the gray image and eight binary bit-plane images which are presented in Fig. 2. It can be easily seen that the bigger the bit-plane number *i* $(0 \le i \le 7)$ is, the smaller the image complexity *CF* is. In other words, the bigger the bit-plane number is, the better the character of retaining or even expanding the block of *MP*. The reason is that the lower bit-plane reflects detail or noise information of *MP*, whereas the higher bit-plane reflects energy information of *MP*. In addition, the complexity of any binary bit-plane image $BP_i\ (0 \le i \le 7)$ is far below that of the gray image *MP*.

Therefore, the method of BPD can greatly reduce the complexity of a gray image, and make the total subpatterns number of all the decomposed binary images much less than that of the gray image.



MP (*Cp*=0.9966) BP_0 (*CF*=0.4523) BP_1 (*CF*=0.4505)

BP_2 (*CF*=0.4367) BP_3 (*CF*=0.4031) BP_4 (*CF*=0.3270)

BP_5 (*CF*=0.2357) BP_6 (*CF*=0.1427) BP_7 (*CF*=0.0834)

Figure 2. Original image *MP* and binary images BP_i ($0 \le i \le 7$)

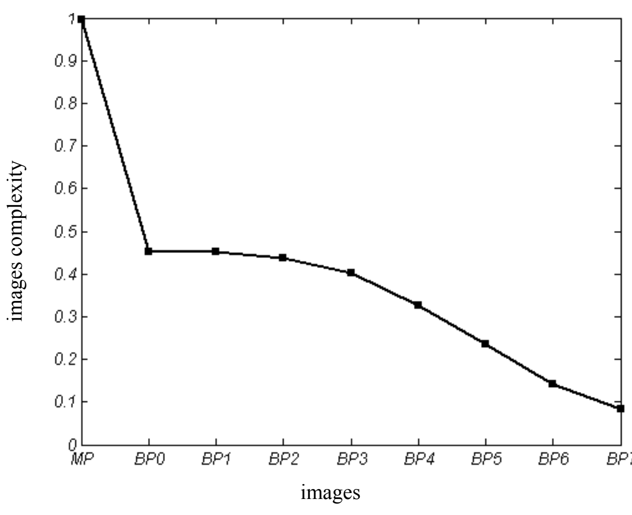


Figure 3. Curve of the images complexity

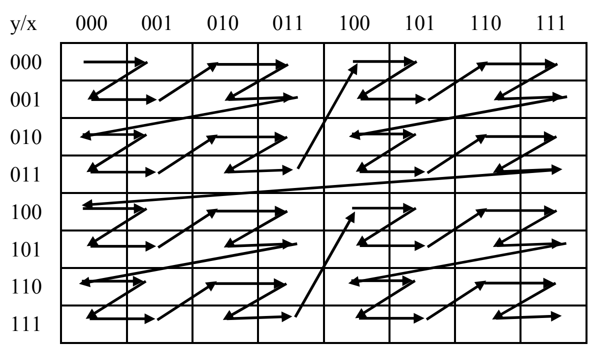
3.2 K-Code transform rules

Suppose an image of size $2^n \times 2^n$ is denoted by $F=\{f(x,y)\}$. Let $F=\{f(x,y)\}$ be a binary image, where $x=(x_{n-1}x_{n-2}\ldots x_1x_0)_2$ and $y=(y_{n-1}y_{n-2}\ldots y_1y_0)_2$. By rearranging the binary bits of x and y, we can obtain a new coordinate variable k, i.e., $k=(y_{n-1}x_{n-1}y_{n-2}x_{n-2}\ldots y_1x_1y_0x_0)_2$. Herein, a two-dimensional image is converted into a one-dimensional sequence, i.e., $F=\{f(x,y)\}=\{g(k)\}$. This one-dimensional representation for the image is named as *K*-Code. The merit of *K*-Code is that it can make the subpattern have a good character of retaining a block. Thus, the *K*-Code can greatly reduce the subpatterns number and enhance the representation efficiency of the image.

The following Fig. 4 is a sketch of *K*-Code. Fig. 4 (a) denotes how a two-dimensional array can be converted into a one-dimensional sequence, which is denoted by the data from 0 to 63. Fig. 4 (b) is an orientation sketch of *K*-Code, which is denoted by the arrows. By the orientations of the arrows, the transform of *K*-Code can be easily understood. The transform from a two-dimensional coordinate (x,y) to a one-dimensional coordinate k is called as *K*-Code transform, i.e., $k=K(x,y)$. On the contrary, the transform from k to (x,y) is called as Counter-*K*-Code transform, i.e., $(x,y)=K^{-1}(k)$.

y/x	000	001	010	011	100	101	110	111
000	0	1	4	5	16	17	20	21
001	2	3	6	7	18	19	22	23
010	8	9	12	13	24	25	28	29
011	10	11	14	15	26	27	30	31
100	32	33	36	37	48	49	52	53
101	34	35	38	39	50	51	54	55
110	40	41	44	45	56	57	60	61
111	42	43	46	47	58	59	62	63

(a) The data sketch of *K*-Code



(b) The orientation sketch of *K*-Code

Figure 4. The sketch of K-Code transform

3.3 Description of Our Algorithm of NAMR for Lossy and Lossless Image Representation

In our algorithm of NAMR, the shape of the subpattern is a rectangle p = {rectangle | rectangle = $L \times W$}, which has two parameters, i.e., the length L and the width W. A threshold d is given beforehand, which is used to control the size of the area of a rectangle pattern. Whether the algorithm is in lossy case or not, it is differentiated by the threshold d. In other words, when d equals zero, the algorithm is a lossless one, whereas when d is greater than zero, the algorithm is a lossy one.

Suppose MP is a gray image of size $2^n \times 2^n$ and m is the bit depth of it. Let BP_i ($0 \le i \le m$-1) denote binary bit-plane images after decomposing the gray image MP by the method of BPD.

The algorithm of NAMR for lossy and lossless image representation consists of the following encoding and decoding parts.

The encoding part of the algorithm can be described as follows: For a given gray image MP with a bit depth m and a threshold d, an encoding result of MP is stored into a queue set $Q=\{Q_0,Q_1,\ldots,Q_{m-1}\}$, where Q_i denotes the i^{th} encoding result for any binary bit-plane image BP_i ($0 \le i \le m$-1). The encoding part of our algorithm has the following steps:

1. Decompose a gray image MP to m binary bit-plane images BP_i ($0 \le i \le m$-1) by the method of BPD and let i=0.

2. Find out an unmarked black rectangle sub-pattern and mark it in BP_i.

2.1. Establish a start point of an unmarked rectangle sub-pattern from the first entrance of BP_i according to the K-Code scanning method and trace the corresponding rectangle subpatterns.

2.2. Find out the biggest rectangle sub-pattern in terms of the area of the rectangle sub-pattern.

2.3. If the area of the biggest rectangle sub-pattern is greater than the threshold d, then mark the found rectangle sub-pattern in the pattern BP_i so that the next start point of the sub-pattern can be easily found.

3. Record the parameters of the marked rectangle subpattern, such as the start point (x,y), the length L, and the width W.

3.1. Obtain a variable SP by using the K-Code transform for (x,y), i.e., $SP = K(x,y)$.

3.2. Increase the variable *Rect_num* by one. *Rect_num* is used to count the found rectangle subpatterns.

3.3. Store SP, L, and W into a queue Q_i, i.e., Q_i {*Rect_num*} = {SP, L, W}.

4. Repeat steps 2 and 3 until there is no unmarked rectangle subpattern in the pattern BP_i.

5. Increase the variable i by one, i.e., $i=i+1$. If $i \le m$-1, then go back to step 2.

6. Output the encoding result $Q = \{Q_0,Q_1,\ldots,Q_{m-1}\}$.

The decoding part of the algorithm can be described as follows: For a given encoding result $Q=\{Q_0,Q_1,\ldots,Q_{m-1}\}$ of a gray image MP of size $2^n \times 2^n$ with a bit depth m, output the reconstruction image g of MP, where $MP=\{MP(x,y)\}$ and $g=\{g(x,y)\}$. The following steps are the decoding part of our algorithm:

1. Obtain Q_i according to the given queue set $Q = \{Q_0, Q_1, \ldots, Q_{m-1}\}$, where Q_i denotes the i^{th} encoding result for the binary bit-plane image BP_i ($0 \le i \le m-1$).

2. Initiate all values of the BP_i of size $2^n \times 2^n$ with white pixels and let i=0.

3. Work out the total subpatterns number *Rect_num* according to Q_i, and let *num*=0.

3.1. Retrieve the three parameters of a rectangle subpattern, i.e., $\{SP, L, W\} = Q_i\{num\}$.

3.2. Obtain a coordinate (x,y) of the start point by using the Counter-*K*-Code transform for *SP*, i.e., $(x,y) = K^{-1}(SP)$.

3.3. Decode the rectangle subpattern of BP_i according to a parameters set $\{(x,y), L, W\}$ and mark it in BP_i.

3.4. Increase the variable *num* by one, i.e., *num* =*num*+1.

4. Compare the variable *num* to *Rect_num*. If *num*<*Rect_num*, then repeat step 3, else go to next step 5.

5. Increase the variable *i* by one, i.e., i=i+1. If $i \le m-1$, then go back to step 3.

6. Obtain the decoding result *MP* according to the expression (5), and compute the Mean Square Error (*MSE*) and the Peak Signal-to-Noise Ratio (*PSNR*) according to the following two formulas:

$$MSE = \frac{1}{2^n \times 2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} \left[g(x,y) - MP(x,y) \right]^2 , \tag{9}$$

$$PSNR = 10 \log_{10} \left(\frac{(2^m - 1)^2}{MSE} \right). \tag{10}$$

3.4 Storage Structure of NAMR

As far as a rectangle subpattern $p = \{\text{rectangle} \mid \text{rectangle} = L \times W\}$ is concerned, in this subsection, we analyze the storage structure of NAMR for the gray images.

As for a gray image, the output of NAMR is a queue set $Q = \{Q_0, Q_1, \ldots, Q_{m-1}\}$. Each Q_i ($0 \le i \le m-1$) corresponds to an encoding result for each binary bit-plane image BP_i after decomposing the gray image by the method of BPD. In fact, Q_i is made up of two elements. One is the start point *SP* of the rectangle subpattern which has a coordinate (x,y); the other is the parameters of the rectangle subpattern which are denoted by the length *L* and the width *W*. Since recording the black pixels is needed, the start point *SP* cannot be omitted. In addition, the encoding lengths for both *x* and *y* are *n*. The start point *SP* is represented by the value of *K*-Code and stored as a relative value ΔK, i.e., $\Delta K = K_i - K_{i-1}$. Therefore, the length of ΔK is *n* in the sight of the statistical concept. In fact, if the length of ΔK is greater than *n*, then one rectangle can be divided into two rectangles, which can be represented by two records. According to the definition and the storage of the value of *K*-Code, the maximal length of *L* or *W* is *n*/2. Thus, the storage structure of the NAMR for a binary image can be denoted by the following Fig. 5. Therefore, storing a rectangle subpattern needs 2*n* bits.

Q_i:	*SP*	$\{L,W\}$

Figure 5. The storage structure of NAMR

4 Data Amount Analyses of NAMR

In this section, by considering a typical rectangle subpattern p = {rectangle | rectangle = $L\times W$}, the total data amount of NAMR is analyzed. Suppose the sizes of a gray image G and a binary image F are both $2^n\times 2^n$. The total pixels numbers of G and F are N_f and N_{fb}, respectively, where $N_f = 4^n$, $N_{fb} = 4^n$.

As far as the rectangle subpattern is concerned, storing a record needs $2n$ bits. Suppose that the bit depth of the gray image G is m. By the method of BPD, we can easily obtain m binary bit-plane images. Let the total data amount of the image G be H_{bpd} when G is represented by NAMR. We can obtain the following expression:

$$H_{bpd} = \sum_{i=0}^{m-1} H_r(i) = \sum_{i=0}^{m-1} 2nN_r(i) = \sum_{i=0}^{m-1} 2nN_f CF(i)\frac{N_r(i)}{N_{LQTB}(i)} = 2n4^n \sum_{i=0}^{m-1} CF(i)\frac{N_r(i)}{N_{LQTB}(i)}. \tag{11}$$

Where $H_r(i)$, $N_r(i)$, $CF(i)$, and $N_{LQTB}(i)$ denote the total data amount, the total subpatterns number, the image complexity, and the total black nodes number of the i^{th} binary bit-plane, respectively.

As for the method of the linear quadtree, storing a record of the black node takes up 3(n-1)+2 bits for a binary image (6, 1982), and takes up 3(n-1)+2+m bits for a gray image since the bit depth m needs to be stored. Suppose that the total data amount of the image G is H_{LQT} when G is represented by the linear quadtree. We can write H_{LQT} as follow:

$$H_{LQT} = (3n-1+m)N_{LQT} = (3n-1+m)N_f Cp = (3n-1+m)4^n Cp. \tag{12}$$

Where N_{LQT} is the total nodes number of G when G is represented by the linear quadtree.

Then, let φ_{LQTbpd} denotes the ratio of H_{LQT} to H_{bpd}. We can easily deduce the following expression.

$$\varphi_{LQTbpd} = \frac{(3n-1+m)4^n Cp}{2n4^n \sum_{i=0}^{m-1} CF(i)\frac{N_r(i)}{N_{LQTB}(i)}} = \frac{(3n-1+m)\frac{N_{LQT}}{4^n}}{2n\sum_{i=0}^{m-1}\frac{N_r(i)}{4^n}} = \frac{(3n-1+m)N_{LQT}}{2n\sum_{i=0}^{m-1} N_r(i)}. \tag{13}$$

By (11) and (12), we can know that the total data amounts of the two methods are both relative to the image complexity. The image complexity, which refers to the nodes number of the linear quadtree, is defined so that our novel algorithm can be compared with the linear quadtree.

The expression (13) shows the ratio of the total data amount of the linear quadtree to that of NAMR. By this expression, we can judge which method is better to represent an image pattern. Since the quadtree segmentation is symmetrical, the segmentation methods suffer from great confine. However, since the NAMR segmentation is asymmetrical, the segmentation methods are unrestricted. The purpose of the NAMR segmentation is to construct the subpatterns as big as possible and yield the least subpatterns number for a packed pattern. Therefore, generally speaking, the total subpatterns number of the NAMR is less than the total nodes number of the linear quadtree, i.e., $\sum_{i=0}^{m-1} N_r(i) < N_{LQT}$. Thus, by (13), we can easily deduce the inequation: $\varphi_{LQTbpd} > (3n-1+m)/(2n)$. Therefore, the ratio of the total data amount of NAMR to that of the linear quadtree is greater than $(3n-1+m)/(2n)$ in lossless case. For example, when *m* and *n* equal 8 and 9, respectively, φ_{LQTbpd} is greater than 1.89 in lossless case. Therefore, our representation method of NAMR can reduce the data storage much more effectively than that of the popular linear quadtree in lossless case.

Further, in lossy case, when compared to traditional lossy representation methods, the representation method of NAMR has a new meaning since it only removes noise information from the image but almost completely retains energy information of the image. Thus, our lossy representation method has a better reconstruction quality.

In a word, the theoretical analyses in this section show that when the representation method of NAMR is compared with that of the popular linear quadtree, not only can the former reduce the data storage much more effectively than the latter in lossless case, but also the former has a better reconstruction quality than the latter in lossy case.

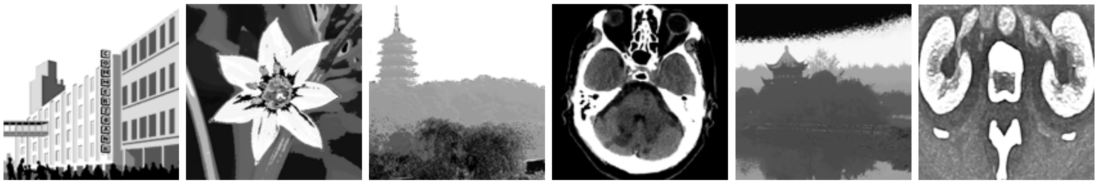
5 Experimental Results

In this section, we tested our algorithm of NAMR with the Matlab 6.5 software and made a comparison with the algorithm of the popular linear quadtree.

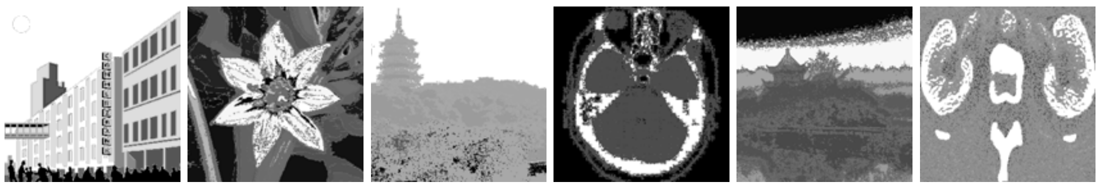
Some representative gray images of size $2^9 \times 2^9$, i.e., n=9, were analyzed. The bit depth of these images is 8, i.e., m=8. The following Fig.6 shows the subjective sense of the image distortion when the threshold *d* equals one. Hereon, 'd=1' means that the subpatterns or nodes whose area equals one are removed from the packed image. Fig.6 (a) is the original six test images, i.e., 'Building', 'WaterLily', 'Tower', 'Brain', 'Gloriette', and 'Liver'. Fig.6 (b) and Fig.6 (c) are the reconstruction images by using representation methods of Quadtree and NAMR, respectively. From the point of view of the subjective sense, the reconstruction image quality of NAMR is much better than Quadtree. The following Table 1 shows the experimental results of NAMR and Quadtree. It contains *Cp* and the number of the subpatterns or nodes in lossy and lossless cases. From the value of *Cp*, we can know that the image complexity of 'Building' is relatively lower, whereas the image complexities of the rest are relatively higher. It can be easily seen from Table 1 that in lossless case, the subpatterns number in NAMR is much less than the nodes number in Quadtree. However, in lossy case, the nodes number in Quadtree may be less than the subpatterns number in NAMR, but from the values of *MSE* and *PSNR*, we know that the corresponding reconstruction image quality of Quadtree is much worse than that of NAMR.

From Table 1, we also know that in lossless case, as far as the six test images are concerned, the total data amount of the Quadtree is 2.36 to 9.39 times of that of NAMR, which corroborates the theoretical obtained result, i.e., $\varphi_{LQTbpd} > 1.89$ in lossless case.

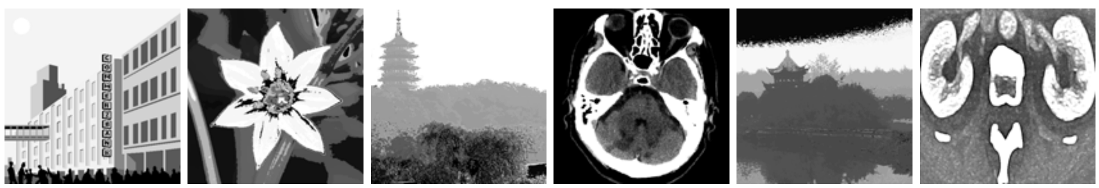
In addition, in lossy case, the total data amount of the Quadtree is 0.20 to 3.80 times of that of NAMR. Since the values of *Cp* are different for different gray images, the total data amount of Quadtree may be less than that of NAMR in lossy case. However, we know that the less data amount of Quadtree is obtained at the cost of sacrificing its reconstruction image quality. As far as *MSE* and *PSNR* are concerned, the algorithm of NAMR is always much better than that of Quadtree. The reason why the reconstruction image quality of the former is much better than the latter is that the former only removes noise information from the image, whereas energy information is almost completely retained.



(a) Original test images: ‘Building’, ‘WaterLily’, ‘Tower’, ‘Brain’, ‘Gloriette’, and ‘Liver’



(b) Reconstruction images by using Quadtree



(c) Reconstruction images by using NAMR

Figure 6. The subjective sense of the image distortion

Table 1. Comparison of performance between NAMR and Quadtree

Image	*Cp*	lossless			lossy (*d*=1)						
		N		φ_{LQTbpd}	*N*		φ_{LQTbpd}	*MSE*		*PSNR*	
		NAMR	Quadtree		NAMR	Quadtree		NAMR	Quadtree	NAMR	Quadtree
Building	0.1422	7477	37264	9.39	6423	12924	3.80	18.36	709.3	35.49	19.62
WaterLily	0.3199	40433	83869	3.89	31507	19293	1.16	110.24	1729.7	27.71	15.75
Tower	0.4191	85728	109873	2.43	59980	10773	0.34	179.59	3150.9	25.59	13.15
Brain	0.5475	114514	143521	2.36	81572	8421	0.20	47.93	3326.8	31.32	12.91
Gloriette	0.3940	63368	103273	3.07	44022	16661	0.72	189.13	1381.0	25.36	16.73
Liver	0.6828	142696	178993	2.37	105324	14417	0.26	50.69	1979.9	31.08	15.16

Note: *Cp*: complexity of a gray image; *N*: number of subpatterns or nodes; Quadtree: linear quadtree; φ_{LQTbpd}: ratio of the total data amount of Quadtree to that of NAMR; *MSE*: Mean Square Error; *PSNR*: Peak Signal-to-Noise Ratio.

As stated above, our experimental results corroborate obtained theoretical results. Therefore, either from theoretical analyses to experimental results or from personality standard to impersonality standard, our representation method of NAMR is much more effective than that of the popular linear quadtree with respect to the data storage in lossless case and the reconstruction image quality in lossy case.

6 Conclusions and Future Work

Image representation is very important in many multimedia applications such as mobile and portable devices. In this paper, inspired by the concept of packing problems, we present a new Non-symmetry and Anti-packing Model with Rectangles (NAMR) for lossy and lossless image representation. The theoretical analyses and experimental results presented in this paper show that when the representation method of NAMR is compared with that of the popular linear quadtree, not only can the former reduce the data storage much more effectively than the latter in lossless case, but also the former has a better reconstruction quality in lossy case. Therefore, our representation method of NAMR presented in this paper is a better method to represent the image pattern and it is valuable for the theoretical research and potential business foreground, such as reducing the storage room, increasing the transmission speed, quickening the process procedure, matching pattern, and so forth.

However, further researches related to our new conception of NAMR need to be done. We list them in the following.

1) The implementation of the algorithm of NAMR for a binary image is based on a single rectangle subpattern, and is not sophisticated, yet it yields excellent encoding and decoding results. It seems extremely likely that well-devised improvements of this algorithm could still increase encoding performance dramatically. For example, as far as the multi-subpattern or arbitrary subpattern is concerned, we can expect a much better result than the single rectangle subpattern's.

2) Unlike a binary image pattern, there are two representation methods of NAMR for a multi-valued image. One is the direct implementation by a similar method for a binary image pattern. The other is the indirect implementation by the method of the binary-bit plane decomposition which can decompose a gray image to m binary images when the bit depth of the multi-valued image is m. However, the implementation of the two representation methods for a multi-valued image still needs to be studied, which is a very important research content.

3) The representation method of NAMR has two purposes: One is to improve the representation efficiency of the data, which has been embodied in our paper. The other, which has not been included in our study, is to improve the computation efficiency of some common atomic operations, such as adjacent finding, set operations, area computing, searching and so on. As far as an image pattern is concerned, since the subpatterns number is much less than the total pixels number after the image pattern is anti-packed, we believe that the computation efficiency of NAMR must be much higher than that of the pixels-based model. Therefore, the study on these operations by the representation method of NAMR is also very significant.

4) NAMR, which is a general, abstract, and pioneer representation method, is suitable for many patterns, such as the image, text, voice, and video, but only the image pattern is studied in this paper. For other patterns, there are still many concrete research contents that need to be studied by a similar method.

In a word, we strongly believe that the further study on NAMR will also yield some very important and valuable theories and results.

Finally, inspired by the new Non-symmetry and Anti-packing Model with Rectangles (NAMR) for lossy and lossless image representation research done in this paper and our previous Triangle Non-symmetry and Anti-packing pattern representation Model (TNAM) of gray images research, in future work, we will consider developing an improved gray image representation method based on Binary-bit Plane Decomposition by using the Non-symmetry and Anti-packing pattern representation Model with Triangles and Rectangles (BPD-based NAMTR).

Acknowledgements

The authors wish to acknowledge the support of the National High Technology Development 863 Program of China under Grant No. 2006AA04Z211.

References

1. A. Bortfeldt and H. Gehring, New large benchmark instances for the two-dimensional strip packing problem with rectangular pieces. in Proceedings of the 39th Hawaii International Conference on System Sciences, (Hawaii, USA, 2006), 1-7.
2. C.B. Chen and D.H. He, Heuristic method for solving triangle packing problem. Journal of Zhejiang University, 2005, 6(6). 565-570.
3. C.B. Chen, D.H. He, and W.Q. Huang, An approximation algorithm for solving the problem of packing unit equilateral triangles in a square. Chinese J. of Computers, 2003, 26(2). 212-220.
4. C.B. Chen, Y.P. Zheng, and M. Sarem, A novel algorithm for multi-valued image representation. in Proceedings of the 3rd International Conference on Natural Computation and the 4th International Conference on Fuzzy Systems and Knowledge Discovery, (2007) 3: 84–89.
5. B.S. De, W.W. De, and Q.J. Van, Using quadtree segmentation to support error modelling in categorical raster data. Intl J. of Geographical Information Science, 2004, 18(2). 151-168.
6. I. Gargantini, An effective way to represent quadtrees. Comm. ACM, 1982, 25(12). 905-910.
7. R. Injong, R.G. Martin, and S. Muthukrishnan, Quadtree-structured variable-size block-matching motion estimation with minimal error. IEEE Trans. Circuits Syst. Video Technol., 2000, 10(1). 42-50.
8. J.J. Laguardia, E. Cueto, and M. Doblare, A natural neighbour galerkin method with quadtree Structure. Intl. J. for Numerical Methods in Engineering, 2005, 63(6). 789-812.
9. M. Manouvrier, M. Rukoz, and G. Jomier, Quadtree representations for storage and manipulation of clusters of images. Image and Vision Computing, 2002, 20(7). 513-527.
10. G. Minglun and Y. Yee-Hong, Quadtree-based genetic algorithm and its applications to computer vision. Pattern Recognition, 2004, 37(8).1723-1733.
11. R. Pajarola, M. Antonijuan, and R. Lario, QuadTIN: Quadtree based Triangulated Irregular Networks. Proc. of 12th IEEE Visualization, (2002), 395-402.
12. H. Samet, Design and Analysis of Spatial Data Structures. Addison-Wesley, 1990.
13. C. Shiwen, G. Oktay, and B. Yener, The multicast packing problem. IEEE Trans. Networking, 2000, 8(3). 311-318.
14. T. Tzouramanis, M. Vassilakopoulos, and Y. Manolopoulos, Overlapping linear quadtrees and spatio-temporal query processing. Computer Journal, 2000, 43(4). 325-343.
15. C.L. Wang, S.C. Wu, and Y.K. Chang, Quadtree and statistical model-based lossless binary image compression method. Imaging Science Journal, 2005, 53(2). 95-103.
16. C. Yung-Kuan and C. Chin-Chen, Block image retrieval based on a compressed linear quadtree. Image and Vision Computing, 2004, 22(5). 391-397.
17. Y.P. Zheng, C.B. Chen, and M. Sarem, A novel algorithm for triangle non-symmetry and anti-packing pattern representation model of gray images. Proc. of 3rd Intl Conf. on Intelligent Computing, (Qingdao, China, 2007), LNCS 4681, 832–841.