A Direct Non-symmetry and Anti-packing Model for Color Images

Chuanbo Chen
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China
chuanboc@163.com

Yunping Zheng
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China
zypdragon@163.com

Mudar Sarem
School of Software Engineering, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China
mudar66@hotmail.com

Abstract

The representation methods of the hierarchical data structures have been widely applied in many important fields, such as computer visualization, robotics, computer graphics, image processing, and pattern recognition. Since these methods put too much emphasis upon the symmetry of segmentation, they are not the optimal representation methods. In this paper, we present a Direct Non-symmetry and Anti-packing Model (DNAM) for color images. Also, we propose a novel algorithm of the DNAM for color images and analyze the data amount of the proposed algorithm. By taking a rectangle subpattern for example, we implement the proposed algorithm and make a comparison with the algorithm of the popular linear quadtree. The theoretical and experimental results presented in this paper show that our proposed algorithm can reduce the data storage much more effectively than that of the linear quadtree and it is a better method to represent color images.

1. Introduction

The hierarchical quadtree-based representation methods have been widely applied in computer visualization, robotics, computer graphics, image processing, and pattern recognition. Klinger [1] proposed a quadtree representation method for a binary image. However, the earlier quadtree representation is based on the pointers. Later, in order to reduce the storage room further, Gargantini [2] removed the pointers and put forward a linear quadtree representation method. The establishment of the linear quadtree representation is an important milestone for the history of the quadtree representation. Until recently, the linear quadtree method is still the most popular representation and is widely studied and applied in many fields [3-8]. However, although the representation methods of the hierarchical data structures stated above have many merits and applications, they put too much emphasis upon the symmetry of segmentation. Therefore, they are not the optimal representation methods. Inspired by the concept of the rectangle packing problem, Zheng and Chen presented a color image representation method based on Non-symmetry and Anti-packing Model (NAM) by using the method of Binary-bit Plane Decomposition (BPD) in [9] where the predefined subpatterns are rectangles, since the traditional representation methods of the hierarchical data structures have their limitations in the representation efficiency. Also, a novel algorithm was put forward for the triangle non-symmetry and anti-packing pattern representation model (TNAM) of gray images in [10] where the predefined subpatterns are triangles.

In this paper, without using the method of BPD [9] and the triangle subpatterns [10], we present a Direct Non-symmetry and Anti-packing pattern representation Model (DNAM) [11] for color images. Also, we propose a novel algorithm of the DNAM for color images and analyze the data amount of this algorithm. The theoretical and experimental results presented in this paper show that our proposed algorithm of the DNAM can reduce the data storage much more effectively than that of the popular linear quadtree and it is a better method to represent color images.

2. The idea of Direct Non-symmetry and Anti-packing Model (DNAM)

A detail description about the DNAM can refer to [11]. In this paper, the idea of the DNAM for color images can be described as following: Giving a packed color image pattern and n predefined subpatterns with different shapes, pick up these subpatterns from the packed pattern and then represent the pattern with the combination of these subpatterns.

We know that a color image can be represented by using three gray images. By using the method of BPD [9], a gray image with a gray level m can be decomposed to m binary images. The method of BPD can effectively reduce the complexity of a gray image, and can make the total subpatterns number of all of the decomposed binary images much less than that of the gray image.

However, in this paper, as far as the DNAM for color images with a gray level m is concerned, we do not treat

978-0-7695-3304-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICNC.2008.72

each gray image as *m* binary images. On the contrary, we treat each gray image just as an indecomposable gray image.

3. A novel algorithm of the DNAM for color images

3.1. Rules of K-Code transform

Before describing the encoding and decoding algorithm of the DNAM for color images, we first reintroduce the rules of K-Code transform [9,10] since our algorithm strongly depends on them.

Suppose a binary image of size $2^n \times 2^n$ is denoted by $F=\{f(x,y)\}$, i.e., $f(x,y)\in\{0,1\}$. Without loss of generality, we assume that 0 and 1 stand for the black and white pixels, respectively, and that the white pixels denote the backgrounds, whereas the black pixels denote the objects. Let (x,y) be a two-dimensional coordinate, 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, the 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 called as K-Code.

The transform from a two-dimensional coordinate (x,y) to a one-dimensional coordinate k is called as K-Code transform, i.e., $k=\mathrm{K}(x,y)$. On the contrary, the transform from k to (x,y) is called as Counter-K-Code transform, i.e., $(x,y)=\mathrm{K}^{-1}(k)$.

3.2. Description of our proposed algorithm of the DNAM for color images

In this section, by taking a single rectangle subpattern $p=\{\text{rectangle}|\text{rectangle}=L\times W\}$ for example, we propose a novel algorithm of the DNAM for color images. The following procedures Rect_encoder ( ) and Rect_decoder ( ) describe the encoding and decoding parts of the proposed algorithm of the DNAM for color images, respectively. The K-Code transform rules are adopted to reduce the total subpatterns number.

```
procedure Rect_encoder (n, m, {CI(x,y)}, Q)
/* Given a color image {CI(x,y)} of size 2^n×2^n×3 with
a gray level m and a resolution n, an encoding result is
stored into a queue Q. */
begin
  color image {CI(x,y)};
  gray image {MP[1](x,y)}, {MP[2](x,y)};
  gray image {MP[3](x,y)};
  K-code k0=0,{g(k)}; /* K-Code array of size 4^n */
  DNAM table Q={Q[1], Q[2], Q[3]};
  integer n, m, gray, MAXarea, area, j, k, x, y, t;
  integer i, j, length, width, length0, width0;
  for t←1 to 3
  begin
    MP[t]←rgb2gray(CI, t);
    /* Decompose a color image CI to its three gray
    images by three-color components r, g, and b, i.e.
    MP[1], MP[2], and MP[3], respectively. */
    i=0;
    while(i<4^n)
    begin
      (x,y)←K^-1(i);
      if MP[t](x,y) ≠-1 then
      begin
        gray←MP[t](x,y);
        length←1; width←1; MAXarea←1;
        for j←y to 2^n-1 do
        begin
          for k←x to 2^n-1 do
          begin
            if MP[t](k,j) ≠gray then break;
            length0←length0+1;
          end
          area←length0× width0;
          if MAXarea<area then
          begin
            MAXarea←area;
            length←length0;
            width←width0;
          end
          if MP[t](x,j) ≠gray then break;
          width0←width0+1
        end
        Q[t]←{gray, length, width};
        marking the rectangle as -1;
      end
      i←i+1;
    end
  end
  output(Q);
end
procedure Rect_decoder (n, m, Q, {CI(x,y)})
/* Given a queue Q, a resolution n, and a gray level m
of a color image {CI(x,y)}, a decoding result is stored
into {CI(x,y)}. */
begin
  color image {CI(x,y)};
  gray image {MP[1](x,y)}={-1}, {MP[2](x,y)}={-1};
  gray image {MP[3](x,y)}={-1};
  K-Code k0=0;
  DNAM table Q={Q[1], Q[2], Q[3]};
  integer max, ptr, n, i, gray, length, width, x, y, j;
  for i←1 to 3
  begin
    ptr=0;
    max←size(Q[i]);
    while (ptr<max)
    begin
```

```
          {gray, length, width}←Q[i](ptr);
          (x,y) ←top_left_coordinate(gray);
          for j←0 to width-1 do
          begin
             for k←0 to length-1 do
             begin
               MP[i](x+k,y+j)=gray;
             end
          end
       ptr←ptr+1;
       end
    end
    CI←gray2rgb(MP[1], MP[2], MP[3]);
end
```

3.3. Storage structure of the DNAM for color images

In this section, a storage structure of the DNAM for a color image of size $2^n \times 2^n \times 3$ is analyzed with respect to a typical rectangle subpattern.

The output of the procedure Rect_ encoder ( ) is a queue Q. As far as a color image *CI* with a gray level *m* is concerned, *m* bits are needed to represent the gray value of *CI*. In addition, a rectangle subpattern has two parameters, i.e., the length *L* and the width *W*. According to the definition of K-Code, the maximal lengths of both *L* and *W* are *n*/2 bits. Thus, the storage structure of the DNAM for color images can be denoted as shown in Fig.1. Therefore, storing a rectangle subpattern takes up *m*+*n* bits.

gray	{ *L*, *W* }

Figure 1. A storage structure of the DNAM

3.4. Data amount analysis of the proposed algorithm of the DNAM for color images

In this section, the representation efficiency of the DNAM for color images is fully embodied. Before analyzing the data amount of the proposed algorithm of the DNAM for a color image *CI* of size $2^n \times 2^n \times 3$ with a gray level *m*, we first introduce the following definition [10] which describes the complexity of a color image in order that the DNAM can be conveniently compared with the linear quadtree.

Definition 1. The complexity of a color image is defined as follows:

$$Cc = N_{LQT} / N_f,$$

where N_{LQT} is the total nodes number of the color image which is represented by the method of the linear quadtree and N_f is the size of the color image.

Suppose the total pixels number of the color image *CI* is N_f, i.e., $N_f = 3\times 4^n$. After the image is anti-packed with a typical rectangle subpattern, suppose the total subpatterns number and the total data amount are N_r and H_r , respectively.

As far as a typical rectangle subpattern is concerned, storing a record of the DNAM needs *m*+*n* bits. Therefore, we can write the total data amount H_r as follows:

$$\begin{aligned} H_r &= (m+n)N_r = (m+n)N_f Cc\frac{N_r}{N_{LQT}} \\ &= 3(m+n)4^n Cc\frac{N_r}{N_{LQT}} \end{aligned} \quad (1)$$

As for the method of the linear quadtree, storing a record needs 3(*n*-1)+2 bits [2] , and takes up 3(*n*-1)+2+*m* bits for a color image since the gray level *m* needs to be stored. Suppose that the total data amount of the color image *CI* is H_{LQT} when *CI* is represented by the method of the linear quadtree. We can write H_{LQT} as follows:

$$\begin{aligned} H_{LQT} &= (3n-1+m)N_{LQT} = (3n-1+m)N_f Cc \\ &= 3(3n-1+m)4^n Cc \end{aligned} \quad (2)$$

Then, let φ denotes the ratio of H_{LQT} to H_r. We can easily deduce the following expression (4).

$$\varphi = \frac{3(3n-1+m)4^n Cc}{3(m+n)4^n Cc\frac{N_r}{N_{LQT}}} = \frac{(3n-1+m)N_{LQT}}{(m+n)N_r} \quad (3)$$

The total data amounts of the DNAM and the linear quadtree can be achieved by (1) and (2), respectively. From these two expressions we can know that the total data amounts of the two methods are both relative to the image complexity. Since the image complexities are different for different images, the representation efficiencies are also various.

The last expression (3) shows the ratio of the total data amount of the linear quadtree to that of the DNAM. From this expression, we can judge which method is better to represent the color image pattern. Since the quadtree segmentation is symmetrical, the segmentation methods suffer from a great confine. However, since the DNAM segmentation is asymmetrical, the segmentation methods are unrestricted. The purpose of the DNAM segmentation is to form the rectangle subpatterns as big as possible and yield the least subpatterns number for a packed pattern. Therefore, generally speaking, N_r is much less than N_{LQT}, i.e., $N_r < N_{LQT}$. Thus, we can easily deduce the following expression:

$$\varphi = \frac{(3n-1+m)N_{LQT}}{(m+n)N_r} > \frac{3n-1+m}{m+n} > 1 \cdot$$

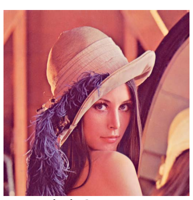
For example, when *m*=8 and *n*=9, φ is greater than 2. Thus, $\varphi > 2 > 1$.

Therefore, the theoretical analysis in this section proves that our representation method of the DNAM for color images can reduce the data storage more effectively than that of the popular linear quadtree and it is a better method to represent color images.

4. Experimental results

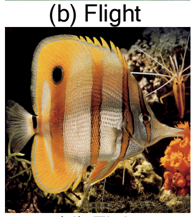
To verify the obtained theoretical results, some representative color images of size $2^9 \times 2^9 \times 3$ (see Fig.2), i.e., n=9, are analyzed. The gray level of these images is 8, i.e., m=8. By taking a single rectangle subpattern for example, we have implemented the proposed algorithm of the DNAM for color images with the Matlab 6.5 software and made a comparison with the algorithm of the popular linear quadtree. The following Table 1 shows the comparison of Cc, N, and φ between the linear quadtree and the DNAM. These representative images have different image complexities. The image complexity is defined according to the nodes number of the linear quadtree. Therefore, according to the value of Cc, we can know how complex an image is. In addition, the bigger the value Cc is, the more complex the image is. From the values of Cc, we can know that the image complexities of Fig.2(a) and Fig.2(b) are relatively higher, whereas the image complexities of Fig.2(e) and Fig.2(f) are relatively lower. The nodes number N of the Quadtree in Table 1 also explains this fact. In addition, the subpatterns number of the DNAM is much less than the nodes number of the Quadtree.

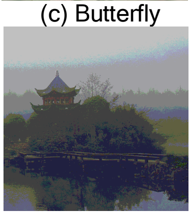
Also, from the values of φ, we know that as far as the six images are concerned, the total data amount of the Quadtree is 2.3145 to 6.9067 times of that of the DNAM, which verifies the theoretical obtained result, i.e., $\varphi > 2 > 1$.


(a) Lena


(b) Flight


(c) Butterfly


(d) Fish


(e) Gloriette

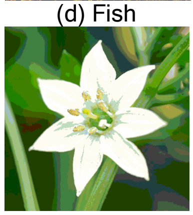

(f) Waterlily

Figure 2. The six color images

Table 1. Comparison of the performance between Quadtree and DNAM

Image	Cc	N		φ
		Quadtree	DNAM	
Lena	0.9976	677779	784545	2.3145
Flight	0.9920	630850	780159	2.4730
Butterfly	0.8895	470878	699519	2.9713
Fish	0.7847	399232	617094	3.0910
Gloriette	0.2216	62622	174309	5.5669
Waterlily	0.1953	44486	153621	6.9067

NOTE: Cc: image complexity; N: number of subpatterns or nodes; Quadtree: linear quadtree; φ: ratio of the total data amount of the Quadtree to that of the DNAM.

Therefore, the experimental results in this section show that our proposed algorithm of the DNAM for color images is much more effective than that of the popular linear quadtree with respect to the data storage and it is a better method to represent color images.

5. Conclusions and extensions

In this paper, we presented a novel Non-symmetry and Anti-packing Model (DNAM) for color images. Also, we proposed a novel algorithm of the DNAM for color images, tested the algorithm of the DNAM for color images with the Matlab 6.5 software, and compared our proposed algorithm with the popular linear quadtree's with respect to the representation efficiency. The theoretical and experimental results shown that our proposed algorithm of the DNAM for color images can reduce the data storage much more effectively than that of the popular linear quadtree.

However, although the implementation of the proposed algorithm of the DNAM for color images 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.

6. Acknowledgments

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

7. References

[1] A. Klinger, "Data structure and pattern recognition", *Proceedings of IJCPR*, Washington, DC, USA, 1973, pp. 497-498.

[2] I. Gargantini, "An effective way to represent quadtrees", *Comm. ACM*, 1982, vol. 25, no. 12, pp. 905-910.

[3] R. Injong, R.G. Martin, and S. Muthukrishnan, "Quadtree-structured variable-size block-matching motion estimation with minimal error", *IEEE Trans. Circuits and Systems for Video Technology*, 2000, vol. 10, no. 1, pp. 42-50.

[4] M. Manouvrier, M. Rukoz, and G. Jomier, "Quadtree representations for storage and manipulation of clusters of images", *Image and Vision Computing*, 2002, vol. 20, no. 7, pp. 513-527.

[5] S.D. Bruin, W.D. Wit, and J.V. Oort, "Using quadtree segmentation to support error modelling in categorical raster data", *International Journal of Geographical Information Science*, 2004, vol. 18, no. 2, pp. 151-168.

[6] S. Tasdoken and A. Cuhadar, "Quadtree-based multiregion multiquality image coding", *IEEE Signal Processing Letters*, 2004, vol. 11, no. 2, pp. 101 - 103.

[7] C.L. Wang, S.C. Wu, and Y.K. Chang, "Quadtree and statistical model-based lossless binary image compression method", *Imaging Science Journal*, 2005, vol. 53, no. 2, pp. 95-103.

[8] M. Laszlo and S. Mukherjee, "A genetic algorithm using hyper-quadtrees for low-dimensional k-means clustering", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2006, vol. 28, no. 4, pp. 533-543.

[9] Y. Zheng, C. Chen, "A color image representation method based on non-symmetry and anti-packing model", *Journal of Software*, 2007, vol. 18, no. 11, pp. 2932-2941.

[10] Y. Zheng, C. Chen, and M. Sarem, "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 (ICIC'07)*, Qingdao, China, August 21-24, 2007, LNCS 4681, pp. 832–841.

[11] C. Chen, W. Hu, L. Wan, "Direct non-symmetry and anti-packing pattern representation model of medical images", *Proceedings of the 1st International Conference on Bioinformatics and Biomedical Engineering (ICBBE'07)*, Wuhan, China, July 6-8, 2007, pp. 1011-1018.