1st IEEE International Symposium on Signal Processing and Information Technology, December 28-30, 2001, Cairo, Egypt (338-342).

Frsq: A Binary Image Coding Method

Peter L. Stanchev[1], William I. Grosky[2], John G. Geske[1]

[1] Kettering University, Flint, MI 48504, {pstanche, jgeske}@kettering.edu

[2] University of Michigan-Dearborn, Dearborn, MI 48128, wgrosky@umich.edu

ABSTRACT

A new region representation method (FrSq method) is proposed. The idea of the method is taken from the assumption, that often the essential part of the image is placed in its center. We recursively subdivide the image in one central square and a frame around it. The frame is investigated like 12 squares [6]. The algorithms for raster to FrSq representation and for FrSq to raster visualization are described. The advantages of the FrSq representation are shown by the comparisons between the FrSq space efficiency and other representation methods.

1. INTRODUCTION

Region representation is an important issue in several applications such as image processing, computer graphics and cartography [6]. There has been much interest in the quadtrees, a compact hierarchical representation which, depending on the nature of the image. Quadtrees represent two-dimensional (spatial) data in a way, which takes advantage of spatial coherence in the phenomenon being represented. One major advantage of quadtrees is that they are very effective for Boolean operations; perhaps more important is the fact that quadtrees are compatible with digital elevation models in grid form. A number of algorithms have been developed for managing images using their quadtree representations. Currently, variants of quadtrees are used to store point data, regions, curves, surfaces and volumes in which the decomposition may be into equal-sized points (regular decomposition) or it may be governed by the input. A valuable review of the quadtrees is given by Samet [6]. The use of quadtree technique in MPEG-4, the first international standard allowing transmitting arbitrarily shaped video objects is described in [1, 5], the use in geographical information system is given in [2].

In this paper we present a new method for region representation. The method is extension of the method described in [9]. It has common properties with the quadtree method (Qt method), because it fits in the same class of methods that are based on the principle of the recursive decomposition of space. The new idea of the proposed method is taken from the assumption, that often the essential part of the image is placed in its center. The paper includes: the informal description of the FrSq method; the formal description of the FrSq method; FrSq to raster and raster to FrSq conversion algorithms; the space efficiency of the FrSq method in comparison with the Qt method (the worst and the best cases for the two methods are discussed).

2. DEFINITIONS AND NOTATIONS

Let a bit map image is transformed into 2^n by 2^n binary array form. In our approach we divide the image in one central square and a frame around it. The same technique is repeated for the central square. The frame is investigated like 12 squares. Let k is the step number of the image division in a square (*sq*) and a frame (*fr*) ($k=1,2, ..., n-1$). The pairs (X^k_{sq},Y^k_{sq}) and (X^k_{fr} ,Y^k_{fr}) denote the coordinates of the south-west corners of the square and the frame respectively at the k-th step. We can number the frames from 0 to $n-2$, according to their widths: 2^0, 2^1, …, 2^{n-2}. The central squares can be numbered from 1 to n-1, according to their sizes 2^1, 2^2, …, 2^{n-1}. Thus, at the step i we have the central square number $n-i$ and a frame number $n-(i+1)$. We say that the region is BLACK if it consists entirely of 1's. It is WHITE if it consists entirely of 0's, and it is GRAY in the other case. In our method we use a data structure that fits in the class of hierarchical data structures. Its nodes contain six fields. Five fields contain pointers to a node's parent and maximum four children, and the sixth field describes the block of the image associated with the node. Its values are BLACK, WHITE or GRAY. We shall denote the number of black nodes in the tree with B, the number of white nodes with W, and the number of gray nodes with G.

3. INFORMAL DESCRIPTION OF FrSq METHOD

The FrSq method belongs to a class of methods whose common property is the recursive decomposition of space. Our example of FrSq representation of data is concerned with the representation of region data. The approach is based on the successive subdivision of the image array into 2^{n-k} by 2^{n-k}

central square with coordinates (X^k_{sq},Y^k_{sq}), and a frame around it, whose width is $2^{n-(k+1)}$ and whose coordinates are (X^k_{fr}, Y^k_{fr}), where k is the step number during the decomposition process $(k=1, ..., n-1)$. If the central square does not consist entirely of 1's or entirely of 0's, k is incremented and the next central square subdivision takes place. If the frame is not entirely black or entirely white, it is subdivided in twelve equal squares, and for every one of them the quadtree representation method [6] is used. The process stops when blocks are obtained (possibly single pixels) that consist entirely of 1's or entirely of 0's. For example, consider the region shown in Figure 1a, which is represented by the 2^3 by 2^3 binary array in Figure 1b. The 1's correspond to picture elements that are in the region and the 0's correspond to picture elements that are outside the region. The resulting blocks for the array of Figure 1b are shown in Figure 1c. This process is represented by a tree, in which the root node corresponds to the entire array. There are two types of node children: (1) the set of twelve squares forming the frame, and (2) the central square. The frame's square has 4 children, corresponding to the successive subdivision of this square into four quadrants (labeled in order SW, SE, NE, NW). The leaf nodes of the tree correspond to those blocks for which no further subdivision is necessary. The FrSq representation for Figure 1a is shown in Figure 1d.

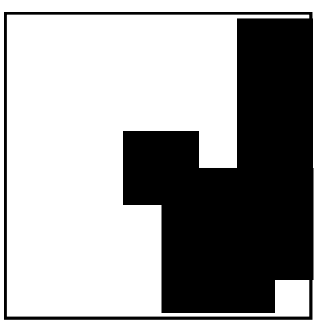


Figure 1.a. An example image

0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	1	1	0	1	1
0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	0

Figure 1.b. The matrix corresponded to the image in the Figure 1a.

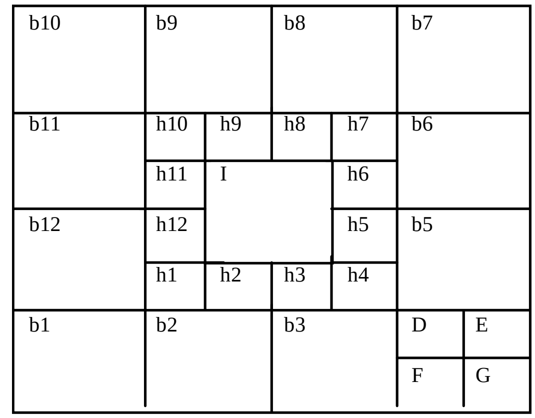


Figure 1.c. The resulting blocks for the array of Figure 1b

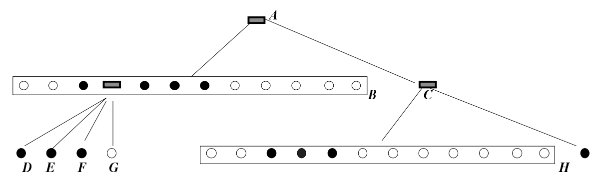


Figure 1.d. The FrSq representation for Figure 1a

4. FORMAL DESCRIPTION OF FrSq METHOD

The proposed method can be described recursively as follows: In the forward direction of the recursion the binary array is treated as 2^n by 2^n square. Let k is the step number of the image division in a square and a frame. If the square consists entirely of 1's (is BLACK) or entirely of 0's (is WHITE) the recursion stops. Otherwise (the array is GRAY) we increment k, and we subdivide the central square into 2^{n-k} by 2^{n-k} square with $X^k_{sq}=Y^k_{sq}=\Sigma_{j=1,\ \dots,\ k}\ 2^{n-1-j}$ and a frame around it with $X^k_{fr}=Y^k_{fr}=X^{k-1}_{sq}$ where $X^0_{sq}=Y^0_{sq}=0$ and the number of rays and columns in the matrix started with 0. If $k=n-1$, we apply the quadtree technique for the last 2 by 2 square. In the backward direction of the recursion, if the frame at the step k is GRAY, we investigate the twelve 2^{n-1-k} by 2^{n-1-k} squares, composing the frame (frame squares), numbered from 1 to 12 starting from the sought-west corner of the frame, moving counter clock-wise direction. The coordinates $(X^k_{i,fr},\ Y^k_{i,fr})$ of the south-west corner of the i-th frame square $(i=1,2,\ \dots\ ,12)$, at the step k are:

$$X^k_{i,fr} = X^k_{fr} + C_{1,i} * 2^{n-1-k},\quad Y^k_{i,fr} = Y^k_{fr} + C_{2,i} * 2^{n-1-k}, \ (1)$$

where C is the following matrix:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 3 & 3 & 3 & 2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 2 & 3 & 3 & 3 & 3 & 2 & 1 \end{pmatrix}$$

For all gray frame squares we apply the quadtree representation technique.

5. ALGORITHMS

We present two algorithms: (1) for obtaining FrSq representation of a 2^n by 2^n raster image, and (2) for obtaining raster representation of a given FrSq tree.

The raster to FrSq representation algorithm processes the image row by row for the square at step k $(k=1, 2, ..., n-1)$. The procedure BUILDSQ is invoked recursively for $k=0,1,...,n-1$. It constructs the right side of the FrSq tree. The recursion stops when full BLACK or WHITE square is found. For the last 2 by 2 square ($k=n-1$), the RASTER_TO_QUADTREE algorithm [7] is used. The procedure BUILDFR processes the frames at steps $n-1,n-2,...,0$. It investigates the twelve squares in the frame using (1) and if the square is GRAY the RASTER_TO_QUADTREE method is applied for it.

The FrSq representation to raster algorithm processes the raster points from left to right row-by-row. The procedure FINDPOINT is invoked to locate the node in the FrSq representation tree that describes the block containing the current point. The procedure uses the coordinates of the squares to determine the corresponding node in the tree. If the raster point is in a frame, the procedure SEARCHSQ evaluates the frame square number, containing the point. If this frame square is GREY, we apply the QUADTREE_TO_RASTER algorithm [8] for it.

6. SPACE EFFICIENCY

In the following sections we consider questions, regarding the size of the representation of 2^n by 2^n image in which a single rectilinearly oriented 2^m by 2^m region occurs. In this section we consider the best and the worst cases for the variables B, W, and G as a function of n and m for FrSq and Qt representation methods.

6.1. Best case for the FrSq method

The best case for the FrSq method occurs when 2^m by 2^m region can be represented by a single node at some level in the FrSq tree. There exist two cases:

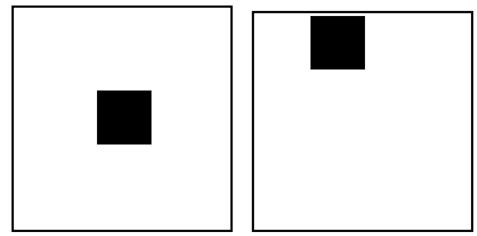
Case A: the region coincides with the central square number "m"; and

Case B: the region coincides with the frame square in frame number "m" (Figure 2.).

The better case is the second one, because in the first case one more subdivision must be done.

In case A: $B = 1$, $W = 12*(n-m)$, $G = n-m$. If we apply the Qt representation technique for this case, we obtain the following results: $B = 4$, $W = 12*(n-m)$, $G = 4*(n-m)+1$.

In case B: $B = 1$, $W = 12*(n-m-1)$, $G = n-m-1$. The case B is the best case for the Qt method to.

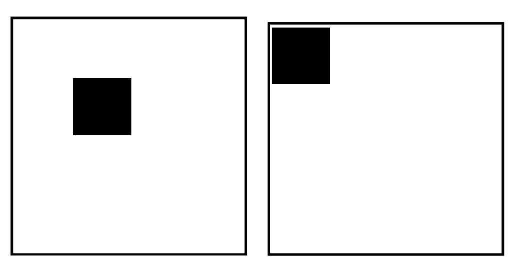


case A case B

Figure 2. Best case for the FrSq method

6.2. Best case for the Qt method

The best case for the Qt method occurs when the region can be represented by a single black node at level m (the level in the quadtree) [3]. Charles Dyer showed that in this case: $B=1$, $W = 3*(n-m)$, $G = n-m$. Applying the FrSq method we can see that there exists two cases, according to the region position in the image. If the black region has an angle that coincides with the image center (case A) (Figure 3-A), we have one black square in the center and $3*m$ black frame squares. Hence $B=3*m+1$. The white nodes are 3 in the square number "1", $9*m$ in the frames that contain the region and $12*(n-m-1)$ in the other frames. We obtain $W=3*(4n-m-3)$. In this case the number of gray nodes is n. If the black region is at any other position (case B) (Figure 3-B), we can calculate that the number of black nodes is 1, the number of white nodes is 1 (in the central white square) plus $3*(n-(k+1)-m)$ (in the square of the $n-(k+1)$-th frame) plus the 11 white squares in the same frame. $W=3*(n-m+3*k)$. Applying the same technique, we obtain that $G=2*k-m-1$.



case A case B

Figure 3. Best case for the Qt method

6.3. Worst case for the FrSq method

The worst case for the FrSq method occurs when the black region $m=n-1$ is at the left or at the right half of the image (maximal number of subdivision steps must be done), and is displaced one position to the top or to the bottom of the image (assuring maximal number of subdivision steps in the frame squares) (Figure 4.). We have 2 black blocks in the last central square, $m-2$ frames with 6 black blocks in every one of them. The number of the black blocks in the frame number "$m-2$" is $4+\Sigma_{i=2,\ldots,m-1} 2^i$ and their number in the frame "m-1" is $\Sigma_{i=0,\ldots,m} 2^i$. The total number of black nodes is $B = 2+6*(m-2)+4+\Sigma_{i=2,\ldots,m-1} 2^i + \Sigma_{i=0,\ldots,m} 2^i$ hence $B = 3*(2^m+2*m)-11$. The number of white nodes is: in central square "1": 2; in frames "0", … ,"m-3": $(m-2)*6$; in frame "m-2": $6+2^{m-1}$; in frame "m-1": $8+\Sigma_{i=2,\ldots,m-1}2^i$. Hence $W=6*2^{m-1}+m$. Grey squares exist only in frames with numbers "m-1" and "m-2". Hence, the number of gray nodes is: in frame "m-1": $\Sigma_{i=0,\ldots,m-2} 3*2^i$; in frame "m-2": $\Sigma_{i=1,\ldots,m-2} 2^i$. We have m+1 gray squares in the image center. Hence $G=m+1+3*(2^{m-1}-1)+2^{m-1}-2 = 4*(2^{m-1}-1)+m$. If we apply the quadtree method in this case we obtain: $B= \Sigma_{i=1,\ldots,m-2} 2^i+2*2^m = 3*2^{m-2}$, $W=2+ \Sigma_{i=1,\ldots,m-2} 2^i+2^{m+1} = 3*2^m$, and $G=(B+W-1)/3 = 2^{m+1}-1$.

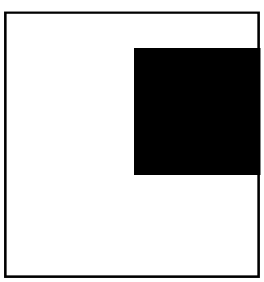


Figure 4. Worst case for the FrSq method

6.4. Worst case for the Qt method

The worst case for the Qt method occurs when the region is at position (r,c) such that r mod $2^m = C \bmod 2^m = 1$, i.e. the region is shifted to the right and down one pixel from the best case position for this method [3] (Figure 5). Charles Dyer [3] showed that using the Qt method: $B = 3*(2^{m+1}-m)-5$, $W = 3*(2^{m+1}+4*n-3*m-5)$, $G = 2^{m+2}+4*(n-m)-7$, $Total= 2^{m+4}+16*(n-m)-27$.

Using the FrSq method we can investigate the following cases:

Case A: The region coincides with the central square number "1". Then

$B=1$, $W=12*(n-m)$, $G= n-m$, $Total_{case\ A}=13*(n-m)+1$.

Case B: The region includes the central square number "1". We have one black block in the center of the image and $\Sigma_{i=1,\ldots,m-1}(3+2^i)$ black blocks in the frames. In the most external frame we have $(m-1)+ \Sigma_{i=2,\ldots,m} (m-i+1)*2^i$ black blocks. Thus $B = 1+ \Sigma_{i=1,\ldots,m-1}(3+2^i)+(m-1)+ \Sigma_{i=2,\ldots,m}(m-i+1)*2^i = 2^{m+1}+4*m+ \Sigma_{i=2,\ldots,m}(m-i+1)*2^i-7$.

The number of white nodes is: in frame "1": 7; in frame "2": $7+2^2$; in frame "3": $7+2^2+2^3$; in frame "m-1": $7+2^2+\ldots+2^{m-1}$. In the most external frame the number of white squares is $2^{m+1}+1$. Hence $W = 2^{m+1}+1+7+7*(m-1)+(m-1)*2^2+\ldots+(m-(m-1))*2^m = 2^{m+1}+7*m+1+\Sigma_{i=1,\ldots,m-1}(m-i)*2^{i+1}$. The number of gray nodes is calculated using the same technique: $G = n + \Sigma_{i=2,\ldots,m-1}(m-i)*2^{i-1} + 4*\Sigma_{i=1,\ldots,m-2}2^i + 3*\Sigma_{i=0,\ldots,m-3}2^{i+5} = m + \Sigma_{i=1,\ldots,n-3} (n-2-i)*2^i + 11*2^{m-2}-7$.

The total number is:

$Total_{case\ B}= 2^{m+2} +11*m+ \Sigma_{i=2,\ldots,m} (m-i+1)*2^i+ \Sigma_{i=1,\ldots,m-1}(m-i)*2^{i+1} +n +\Sigma_{i=2,\ldots,m-1} (m-i)*2^{i-1} + 11*2^{m-2}-13$.

Case C: The region does not include the central square number "2". To calculate the number of black nodes, we must investigate the frames covering the region. We can see that the case is the same like the described one by Dyer [7]. Thus: $B = 3*(2^{m+1}-m)-5$. The number of white nodes is not the same, because we have a white central square, and W is calculated as follows: the number of white 2^0 by 2^0 squares to the left and to the bottom of the region is $2^{m+1}+1$; the number of white squares in frame number "m" is $10+ \Sigma_{i=1,\ldots,m}2^i$; in frame number "m-1" is $3*\Sigma_{i=1,\ldots,m-1}2^i + m+8$; plus one central white square; plus $12*(n-m-2)$ white squares in the frames number "m+1", ..., "n-2". Hence $W = 7*2^m-11*m+12*n-12$. The number of gray nodes is calculated using the above technique: $G = 2^{m+2}+n-m-6$. *The total number of nodes is:* $Total_{case\ C}=11*2^m+13n-15m-16$.

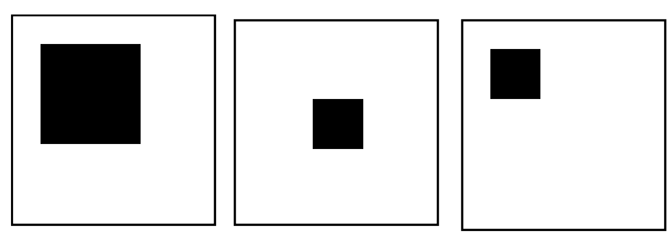


Figure 5. Worst case for the Qt method

6.5. Space efficiency result analysis

According to the obtained results we can conclude that: When used in its best case: in case A the FrSq method is better (more space efficient) than the Qt method when $m=n-1$ or $m=n-2$ (Figure 6a); in case B when 2^m by 2^m region coincides with the central square number "m", the FrSq method is always better (Figure 6a). In the best case for the Qt method, when the region has an angle that coincides with image center - case A (Figure 6b) the Qt method is more space efficient than the FrSq method. In case B (Figure 6b, for k=1 and k=8) the FrSq method is better than the Qt method when *4*n-3*m > 11*k*. In the FrSq method worst case, The Qt representation technique is allays more space efficient (Figure 6c). In the worst case for the Qt method, the FrSq method is allays better (Figure 6d).

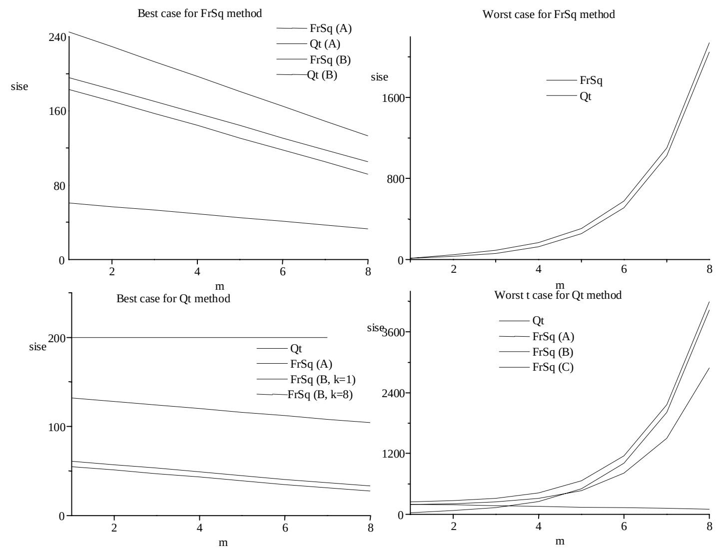


Figure 6. Memory needed for storage of a 2^{16} by 2^{16} image contained a square region in it

7. CONCLUSION

The described FrSq representation method is used in some experimental works [4]. Presently, we develop a set of algorithms, using the FrSq representation, to calculate some region characteristics, like region perimeter, area, etc. Conversion methods from and to other representations will be developed too. The FrSq method realization is done on IBM/PC using Pascal language.

REFERENCES

[1] Chen Mei-Juan, Hsieh Yuan-Pin, Wang Yu-Ping, Multi-resolution shape coding algorithm for MPEG-4, International Conference on Consumer Electronics, 126-127, 2000.

[2] Clarke, R.J., Image and video compression: a survey, International Journal of Imaging Systems and Technology: vol.10, no.1, 20-32, 1999.

[3] Dyer, C., The space efficiency of quadtrees, Computer Graphics and Image Processing, Vol 19 335-348, 1988.

[4] Grosky W., Stanchev P., An Image Data Model, in Advances in Visual Information Systems, R. Laurini (edt.), Lecture Notes in Computer Science 1929, 14-25, 2000.

[5] Ma Chil-Cheang, Chen Mei-Juan, Efficient shape coding algorithm by quadtree decomposition for MPE-4, Proceedings of the SPIE, vol.3974, 709-719, 2000.

[6] Samet, H., Algorithms for the convertion of quadtrees to rasters, Comput. Vision Gr. Image Process. Vol 26, 1-16, April 1984.

[7] Samet, H., An algorithm for converting rasters to quadtrees, IEEE Trans. Pattern Anal. Mach. Intell. Vol 3, 93-95, Jan. 1981.

[8] Samet, H., The quadtree and related hierarchical data structures, ACM Comput. Surv. Vol 16, 184-260 , June 1984.

[9] Stanchev P., Vutov V., Instrumentos Computadoorizados para la Codification, Almacenamiento, Reconocimiento y Recuperacion de Imagenes", 45th Conference and Congress Information, a Resource for Development, 54-56, 1990.