APPROXIMATION AND COMPRESSION OF IMAGES USING QUADTREES*

Hanan Samet

Computer Science Department, University of Maryland
College Park, MD 20742 USA

ABSTRACT

A new pointer-less quadtree representation is presented. It is used to define a sequence of approximations to quadtree images that are progressive and lead to compression.

1. INTRODUCTION

The quadtree [5,9] is a hierarchical data structure that is finding increasing use in image processing and graphics applications. It is usually implemented as a tree (e.g., Figure 1). However, the amount of space required for pointers from a node to its sons is not trivial. Consequently, there has been interest in pointer-less quadtree representations. They can be grouped into two categories. The first represents the image in the form of a preorder traversal of the nodes of its quadtree [4] while the second treats the image as a collection of leaf nodes (e.g., [1,2]). In this paper we define a new pointer-less quadtree representation of the latter type and use it to obtain a sequence of approximations to quadtrees. These approximations are seen to be progressive and also lead to compression.

2. A POINTER-LESS QUADTREE REPRESENTATION

Instead of using a tree consisting of BLACK,WHITE, and GRAY nodes, we use a representation in the form of a collection of the leaf nodes of the quadtree. Each leaf node is represented by a locational code which corresponds to an encoding of the path from the root of the tree to the node. In fact, we only need to maintain a collection of the BLACK nodes since the WHITE nodes can be determined given the BLACK nodes. In this paper, we make use of the following definition of a locational code. Let the sequence $<x_i>$ represent the path of nodes from the root of a quadtree to x_m, the desired node, such that x_n=root of the quadtree and x_i=FATHER(x_{i-1}). The directions NW, NE, SW, SE are represented by the directional codes 1, 2, 3, and 4 respectively and are accessed by the function SONTYPE5. The encoding of the locational code for node x_m is given by z_n where z_i is defined below:

$$z_i = \begin{cases} 0 & i=m \\ 5 \cdot z_{i-1} + \text{SONTYPE5}(x_i) & m<i\leq n \end{cases}$$

For example, node 10 of Figure 1 would be encoded by the number z_4 =582. It can be decoded into the sequence of directional codes $<b_i>$ = <2,1,3,4> = <NE,NW,SW,SE> - i.e., $z_4 = 4\cdot 5^3+3\cdot 5^2+1\cdot 5^1+2\cdot 5^0$.

The above encoding method differs from those of Gargantini [2] and Abel and Smith [1] who use fixed length codes. Our method has a number of useful features. First, it lends itself easily to decoding a locational code into the sequence of directional codes by using a combination of modulo and integer division operations. We obtain the directional codes in the order in which we traverse a path from the root of the quadtree to the root of the subquadtree. Second, sorting the codes of the BLACK nodes in increasing order, results in a sequence which is a variant of a breadth-first traversal of the BLACK portion of the tree - i.e., for $i<j$, nodes at level j (representing a block of size 2^j by 2^j) will appear in the sequence before nodes at level i. This breadth-first property means that the sequence yields a progressive approximation of the image - i.e., successive nodes lead to a better approximation. Finally, increasing the resolution of the image does not require extensive recoding of the codes for the existing nodes.

3. HIERARCHICAL APPROXIMATION METHODS

By virtue of its hierarchical structure the quadtree lends itself to serve as an image approximation device. By truncating the tree (i.e., ignoring all nodes below a certain level), we get a crude approximation. Ranade, Rosenfeld, and Samet [6] define two basic variants termed an inner and outer approximation. Given an image I, the inner approximation, IB(k), is the binary image defined by the BLACK nodes at level $\geq k$ (e.g., for Figure 1, IB(2)={16, 25}). The outer approximation, OB(k), is the binary image defined by the BLACK nodes at levels $\geq k$ and the GRAY nodes at level k (e.g., for Figure 1, OB(2)={16, 25, F, G, H}). At this point, let us use $\leq$ and $\geq$ to indicate set inclusion in the sense that A $\leq$ B and B $\geq$ A imply that the space spanned by A is a subset of the space spanned by B. It can be shown that $\text{IB}(n) \leq \text{IB}(n-1) \leq \cdots \leq \text{IB}(0)=\text{I}$ and $\text{OB}(n) \geq \text{OB}(n-1) \geq \cdots \geq \text{OB}(0)=\text{I}$. Alternatively, we can approximate the image by using its complement, $\bar{\text{I}}$ - i.e., the WHITE blocks. We define IW(k) and OW(k) in an analogous manner to that of IB(k) and OB(k) respectively except in terms of WHITE blocks. It can be shown that $\text{IW}(n) \leq \text{IW}(n-1) \leq \cdots \leq \text{IW}(0) = \bar{\text{I}}$ and $\text{OW}(n) \geq \text{OW}(n-1) \geq \cdots \geq \text{OW}(0) = \bar{\text{I}}$.

4. FOREST-BASED APPROXIMATION METHODS

Jones and Iyengar [3] introduced the concept of a forest of quadtrees which is a decomposition of a quadtree into a collection of subquadtrees, each of which corresponds to a maxi-

*The support of the U.S. Army Engineer Topographic Laboratory under contract DAAK 70-81-C-0059 is gratefully acknowledged.

mal square. The maximal squares are identified by refining the concept of a non-terminal node to indicate some information about its subtrees. An internal node is said to be of type GB if at least two of its sons are BLACK or of type GB. Otherwise, the node is said to be of type GW. For example, in Figure 1, nodes F,J, and M are of type GB and nodes A, B, C, D, E, G, H, I, K, L, and N are of type GW. Each BLACK node or an internal node with a label GB is said to be maximal square. A *BLACK forest* is the minimal set of maximal squares that are not contained in other maximal squares and that span the BLACK area of the image. Thus the BLACK forest corresponding to Figure 1 is {F, 10, 16, 25, 27, M, 38} and their corresponding subtrees. The elements of the BLACK forest are specified by locational codes (although Jones and Iyengar use a different definition than z_i). Such a representation can lead to a savings of space since large WHITE areas are ignored by it.

A forest can also be used as an approximation where we treat its elements as BLACK and all remaining nodes as WHITE. It is useful to sort the nodes of the forest according to their codes. We use the locational codes given by z_i. For example, for Figure 1, the nodes will appear in the order 25, 16, F, M, 27, 38, and 10. This order is a partial ordering $(S,\geq)$ such that $S_i \geq S_{i+1}$ means that the block subsumed by S_i is $\geq$ in size than the block subsumed by S_{i+1}. In fact, for a breadth-first traversal we only need to process the nodes in an order that satisfies the above subsumption relation. It should be clear that a sorted list is just one of many possible orderings satisfying the subsumption relation.

An approximation based on BLACK forests can be formulated as follows. Let FBB(n) be the forest as defined by Jones and Iyengar. We say that FBB(i) is the result of replacing GRAY nodes in FBB(i+1) by their BLACK forests. It should be clear that FBB(n) $\geq$ FBB(n–1) $\geq$ ··· $\geq$FBB(0) = I. For example, for Figure 1, FBB(4)={25, 16, F, M, 27, 38, 10}; FBB(3) is obtained by adding the BLACK forests of F (i.e., nodes 19, J, 22, and 23) and M (i.e., nodes 28 and 29) to yield FBB(3)={25, 16, J, 22, 23, 28, 27, 29, 38, 19, 10}. Continuing the replacement of GRAY nodes by their BLACK forests leads to FBB(2)={25, 16, 22, 23, 20, 14, 27, 29, 28, 38, 19, 21, 10}=FBB(1)=FBB(0)=I. Clearly, FBB(n) $\geq$ FBB(n–1) $\geq$ ··· $\geq$ FBB(0) =I. Note that FBB provides a closer approximation to the image than OB in the sense that OB(i) $\geq$ FBB(i). This can be seen by observing that OB(i) and FBB(i) both contain all the terminal BLACK nodes at level $k \geq i$, but all nodes of FBB(i) at levels $j<i$ are contained in the GRAY nodes at level i which are elements of OB(i). It should be clear that the comparisons between OB(i) and FBB(i) can be made despite the fact that i corresponds to a level in the former and an iteration in the latter.

We can also have a forest approximation that is made up entirely of nodes corresponding to WHITE blocks. In other words, we approximate the complement of the image, $\overline{\text{I}}$. Such a forest is defined analogously to the one presented earlier using the BLACK blocks - i.e., each quadtree is a collection of subquadtrees, each of which corresponds to a maximal square. A *WHITE forest* is the minimal set of maximal squares that are not contained in other maximal squares and that span the WHITE area of the image. Let FWW(n) be the WHITE forest as defined above. We say that FWW(i) is the result of replacing GRAY nodes in FWW(i+1) by their WHITE forests. It should be clear that FWW(n) $\geq$ FWW(n–1) $\geq$ ··· $\geq$ FWW(0) = $\overline{\text{I}}$. For example, for Figure 1, FWW(4)={A}; FWW(3) is obtained by adding the WHITE forests of A (i.e., nodes B, C, D, E) to yield FWW(3)={B, C, D, E}. Continuing the replacement of GRAY nodes by their WHITE forests leads to FWW(2)={1, 2, 11, I, 15, G, 5, 17, 24, 40, 41, H, 30, 42, 43}, etc. By analogy to the BLACK forest, we see that FWW provides a closer approximation to the inverse image than OW in the sense that OW(i) $\geq$ FWW(i) for all i.

We can also make use of FWW to approximate I by working with its complement, $\overline{\text{FWW}}$. It has the following properties. OW(i) $\geq$ FWW(i) implies that $\overline{\text{FWW}(i)} \geq \overline{\text{OW}(i)}$ = IB(i). Similarly, FWW(i) $\geq$ $\overline{\text{I}}$ implies $\overline{\text{FWW}(i)} \leq$ I. These two properties plus the fact that I $\leq$ FBB(i) $\leq$ OB(i) lead to the following theorem

Theorem 1: IB(i) $\leq$ $\overline{\text{FWW}(i)}$ $\leq$ I $\leq$ FBB(i) $\leq$ OB(i), $0 \leq i \leq n$.

This is an important result because it means that we have found better approximations to I (i.e, FBB and $\overline{\text{FWW}}$) than OB and IB. In particular, we see that using FBB results in overestimating the area spanned by the image while using $\overline{\text{FWW}}$ results in underestimating the area spanned by the image. In essence, we are approximating the image solely by use of BLACK blocks or solely by use of WHITE blocks. However, we could also approximate the image by a combination of BLACK and WHITE blocks. Let FBW be such an approximation. FBW(n) is FBB(n). For FBW(n–1), we augment FBW(n) by adding the WHITE forests (using FWW) of all the GRAY nodes in FBW(n). For FBW(n–2), we apply the same augmentation process except that now we add the BLACK forests of all the GRAY nodes added in the previous step. This alternating process of adding BLACK and WHITE forests is continued until no GRAY nodes are left. For example, for Figure 1, we have FBW(4)={25, 16, F, M, 27, 38, 10}. FBW(3) is obtained by adding the WHITE forests of node F (i.e., nodes I and 15) and M (i.e., nodes 33 and 34) to yield FBW(3)={25, 16, F, M, 27, 38, 10, I, 15, 33, 34}. FBW(2) is obtained by adding the BLACK forests of node I (i.e., node 19) to yield FBW(2)={25, 16, F, M, 27, 38, 10, I, 15, 33, 34, 19}=FBW(1)=FBW(0)=I.

The quadtree corresponding to the image represented by the approximation sequence FBW can be easily reconstructed. This does not require the indication of the colors of the individual nodes in the sequence. We start with the empty tree. We add the nodes in increasing order of their locational codes (i.e., a breadth-first tree traversal). More specifically, we add the nodes to the tree in an order so that for any two nodes P and Q such that P is an ancestor of Q, P is added to the tree before Q. When adding a node, a path (designated by the node's locational code) is traced from the root of the quadtree to the node. Any terminal node encountered during this process will be expanded. In particular, the node's type is changed to GRAY and its four sons take on its previous value. When the decoding of the path is done, the node's type is complemented. For example, when adding I, one of the WHITE forests of F in Figure 1, F becomes a GRAY node, I becomes WHITE, and all brothers of I become BLACK. The reconstruction sequence shows the use of $\overline{\text{FWW}}$ in the sense that the WHITE forests (i.e., FWW) are added to the tree as WHITE nodes thereby subtracting from the BLACK forests.

FBW is attractive as an approximation because it converges to the image from both directions - i.e., it alternates between overestimating and underestimating the BLACK component. It is not hard to see that FBW satisfies the following relationships. $FBW(n) \geq FBW(n-1)$, $FBW(n-1) \leq FBW(n-2)$, $FBW(n-2) \geq FBW(n-3)$ and in general, $FBW(n-2i) \geq FBW(n-2i-1)$ and $FBW(n-2i-1) \leq FBW(n-2i-2)$. Furthermore, it is easy to show that $FBW(n) \geq FBW(n-2) \geq \cdots \geq FBW(n-2i) \geq \cdots \geq FBW(0) = I \geq \cdots \geq FBW(n-2i-1) \geq \cdots \geq FBW(n-3) \geq FBW(n-1)$. In other words, the approximations FBW spiral in from both sides of I in converging to I. Note that individually, FBB and $\overline{FWW}$ may, at times, be better approximations to I than FBW - i.e., there exist images for which the amount of BLACK by which FBB overestimates I is less than the amount by which FBW underestimates I. The opposite is also true - i.e., an image can be constructed such that the amount of BLACK by which FBB overestimates I is greater than the amount by which FBW underestimates I.

The FBW approximation has a number of interesting properties. First of all, it is progressive. Second, use of the FBW approximation will often lead to compression in the sense that it reduces the amount of data that is needed to encode the image (and transmit it). Recall that we can represent a quadtree by merely specifying all of the BLACK blocks or all of the WHITE blocks. Depending on the image we would use the color with the smaller cardinality in order to save storage. The FBW approximation consists of a combination of GRAY, BLACK, and WHITE nodes thereby striking a balance between using all BLACK or all WHITE nodes. In fact, use of FBW may lead to the result that less data is being transmitted than were the quadtree transmitted using all BLACK or all WHITE nodes. For example, encoding Figure 1 with FBW requires 12 nodes whereas the image contains 13 BLACK and 30 WHITE nodes. Letting F, B, and W denote the number of nodes when encoding the quadtree using FBW, BLACK, and WHITE nodes respectively, compression is said to exist whenever $F < MIN(B,W)$. As we shall see below, variants of FBW can be constructed so that F is always $\leq MIN(B,W)$. Thus we can guarantee that our approximation methods are always at least as good or better than encoding the quadtree by listing its BLACK nodes (or its WHITE nodes).

To see the type of compression that is achievable, let $C=F/MIN(B,W)$ be a compression factor. C can be made as close to zero as desired. Figure 2a demonstrates the empty tree which has $F=0$ (i.e., a WHITE node at the root) which we exclude. Figure 2b illustrates a tree with $F=1$ but $C=1$. For a 2^n by 2^n image, a tree having depth $n=2 \cdot m$ can be constructed such that $F=3$ and $C=3/3 \cdot m=1/m$. Figure 2c is such a tree with $n=6$. In this case $F=3$ (nodes 1, D, and 19) while $B=10$, $W=9$, and $C=1/3$. In general, the achievable compression increases with the frequency of the occurrence of 3 out of 4 sons of the same color at different levels of the tree. The following is an upper bound on the number of nodes comprising FBW (i.e., F). Its proof can be found in [8].

Theorem 2: The maximum number of nodes in an FBW approximation is less than or equal to one plus the number of WHITE nodes in the quadtree (i.e. $F \leq W + 1$).

The FBW approximation relies on alternating between the FBB and FWW approximations. We can also define an approximation FWB which alternates between FWW and FBB approximations. In other words, $FWB(n)$ is $FWW(n)$. For $FWB(n-1)$, we augment $FWB(n)$ by adding the BLACK forests (using FBB) of all the GRAY nodes in $FWB(n)$. For $FWB(n-2)$ we apply the same augmentation process except that now we add the WHITE forests of all the GRAY nodes added in the previous step. This alternating process of adding WHITE and BLACK forests is continued until no GRAY nodes are left. In this case, W, the number of WHITE nodes in the quadtree, is an upper bound on the number of nodes in an FWB approximation [8].

We can also obtain upper bounds in terms of B, the number of BLACK nodes. In order to do this we redefine our approximation sequence in terms of maximal WHITE blocks. In particular, we relabel our quadtree with GB′ and GW′ such that an internal node is said to be of type GW′ if at least two of its sons are WHITE or of type GW′; otherwise, the node is said to be of type GB′ (i.e., at least three of its sons are BLACK or of type GB′). We now redefine FBB, FWW, FBW, and FWB in terms of GB′ and GW′ to yield FBB′, FWW′, FBW′, and FWB′ respectively. In other words, FWB′ now alternates between FWW′ and FBB′, and FBW′ alternates between FBB′ and FWW′. This results in upper bounds of B and B+1 for the FBW′ and FWB′ approximations respectively [8]. These upper bounds as well as those of FBW and FWB can be tightened slightly by examining the type of the root node. A summary of these results is given in Table 1. We now see that a judicious choice of an approximation method means that $F \leq MIN(B,W)$.

5. CONCLUDING REMARKS

The various encoding schemes discussed in Section 3 were applied to a 512 by 512 image (i.e., $n=9$) consisting of a floodplain used in prior experiments with quadtrees [7]. A quadtree encoding of the image contains 2208 BLACK nodes and 2485 WHITE nodes. Table 2 contains a summary of the results for the FBW and FBW′ approximations. No results are tabulated for the FWB and FWB′ approximations because their node counts will differ by one. To see this we consider the two possible cases depending on the type of the root. When the root is of type GB, $FBW(n) = \{root\}$ and $FBW(n-1) = \{root\}+FWB(n)$. An analogous statement can be made with respect to the FBW′ and FWB′ approximations. Since the approximations alternate between BLACK and WHITE nodes, our table specifies the counts for them as well as the total number of nodes.

Table 2 correlates with our theoretical results with respect to upper bounds on the number of nodes necessary. In particular, we find that FBW′ requires 1714 nodes to encode the image while FBW requires 1815 nodes. Thus, comparing these counts with the minimum of the BLACK and WHITE nodes in the quadtree (i.e., 2208 BLACK nodes), we find that FBW′ leads to 22.3% fewer nodes while FBW leads to 17.8% fewer nodes. These compression factors increase considerably as larger images are used (i.e., greater than 2^9 by 2^9 as in this example). We also observe that $FBW(n)$ and $FBW'(n)$ have a different number of nodes. This is because of the different definition of GB. Recall, that for FBW, GB corresponds to at least two sons of type GB or BLACK terminal nodes; whereas for FBW′, GB corresponds to at least three sons of type GB or BLACK terminal nodes. Thus it should be clear that the GB criterion corresponding to FBW′ is harder to satisfy than the GB criterion of FBW thereby causing the initial approximation $FBW'(n)$ to contain nodes from lower levels in the tree (and hence more of them!).

It is interesting to make some general observations on the quality of the approximations. First, the FBB approximation is less "blocky" at the edges than OB or IB. Second, IB does not preserve connectivity whereas OB and FBB do so at the possible expense of creating holes where there may not be any. This is not surprising in light of the fact that IB underestimates the BLACK regions. It should be clear that the FBW approximations have connectivity problems similar to IB. In particular, FBW alternates between GB and GW nodes. At iterations that use GW nodes, connectivity may be destroyed. It should be clear that FWB has the same problems with respect to connectivity. Similarly, for FWB' (and FBW'), we have the problem that spurious holes may result at iterations that use GB' nodes. Third, use of FBW as an approximation method is superior to the inner and outer approximations of [6]. FBW-like approximations are biased in favor of objects with so-called "panhandles" rather than "staircases." Nevertheless, it does not always result in compression (e.g., a checkerboard).

ACKNOWLEDGMENTS

I would like to thank Robert E. Webber for his valuable comments and criticisms, and for generating Table 2.

REFERENCES

1. D.J. Abel and J.L. Smith, A data structure and algorithm based on a linear key for a rectangle retrieval problem, *Computer Vision, Graphics, and Image Processing 24*, 1(October 1983), 1-13.

2. I. Gargantini, An effective way to represent quadtrees, *Communications of the ACM 25*, 12(December 1982), 905-910.

3. L. Jones and S.S. Iyengar, Representation of regions as a forest of quadtrees,, *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, Dallas, 1981, 57-59.

4. E. Kawaguchi and T. Endo, On a method of binary picture representation and its application to data compression, *IEEE Transactions on Pattern Analysis and Machine Intelligence 2*, 1(January 1980), 27-35.

5. A. Klinger, Patterns and Search Statistics, in *Optimizing Methods in Statistics*, J.S. Rustagi, Ed., Academic Press, New York, 1971, 303-337.

6. S. Ranade, A. Rosenfeld, and H. Samet, Shape approximation using quadtrees, *Pattern Recognition 15*, 1(1982), 31-40.

7. A. Rosenfeld, H. Samet, C. Shaffer, and R.E. Webber, Application of hierarchical data structures to geographical information systems, Computer Science TR-1197, University of Maryland, College Park, MD, June 1982.

8. H. Samet, Data structures for quadtree approximation and compression, Computer Science TR-1209, University of Maryland, College Park, MD, August 1982.

9. H. Samet, The quadtree and related hierarchical data structures, to appear in *ACM Computing Surveys*, also Computer Science TR-1329, University of Maryland, College Park, MD, November 1983.

Table 1. Upper bounds for the various approximations.

Approximation	Root Type	
	GB (GB')	GW (GW')
FBW	W+1	W-1
FWB	W	W
FBW'	B	B
FWB'	B-1	B+1

Table 2. FBW and FBW' approximations for floodplain image.

i	FBW(i)			FBW'(i)		
	BLACK	WHITE	TOTAL	BLACK	WHITE	TOTAL
9	38	0	38	1038	0	1038
8	38	901	939	1038	226	1264
7	268	901	1169	1410	226	1636
6	268	1424	1692	1410	272	1682
5	344	1424	1768	1437	272	1709
4	344	1470	1814	1437	275	1712
3	345	1470	1815	1439	275	1714

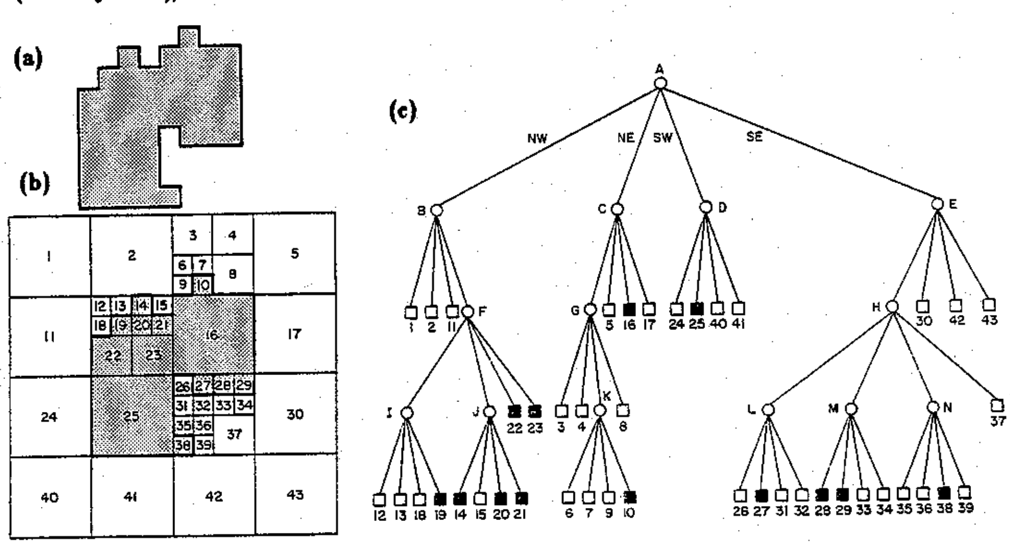

Figure 1. An image (a), its maximal blocks (b), and the corresponding quadtree (c). Blocks in the image are shaded; background blocks are blank.

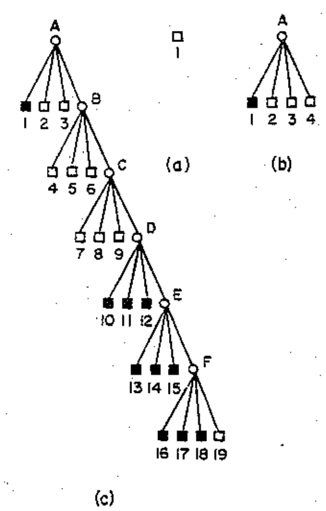

Figure 2. Examples illustrating the compression factors available through the use of FBW.