HIERARCHICAL DATA STRUCTURES FOR IMAGE PROCESSING

Hanan Samet

Computer Science Department and
Computer Vision Laboratory
University of Maryland
College Park, MD 20742

Abstract

Representation is an important issue in the design of image processing systems. In this paper we discuss the quadtree data structure and a number of its variants and show how they can be adapted to encode image data of a geographic nature.

1. Introduction

Representation plays an important role in image processing systems. Today's systems must be capable of dealing with many different types of data and queries. A wide number of representations is currently in use. Recently, there has been a considerable amount of interest and research in hierarchical data structures primiarly in the field of information retrieval and image processing. This interest is based to a large degree on considerations of space and execution time economies. In this paper we discuss the quadtree data structure and a number of its variants and show how they can be adapted to handle image data of a geographical nature. In particular, we focus our attention on their ability to represent data corresponding to regions, points, and curves with a greater emphasis on regions. We also give examples of the types of operations which are faciliated through their use.

2. Background

The term quadtree is used to describe a class of hierarchical data structures whose common property is that they are based on the principle of recursive decomposition. They can be differentiated on the basis of the type of data that they are used to represent and on the principle guiding the decomposition process. When the decomposition is into equal-sized parts, it is said to be regular. Much of our discussion centers on region representation. In essence, there are two ways in which hierarchical data structures can be used to describe regions - those that describe their interiors and those that specify their borders. Section 3 focusses on the former while Section 4 focusses on the latter.

The quadtree is an approach to region representation which is based on the successive subdivision of an image array into quadrants. If the array does not consist entirely of 1's or 0's, then we subdivide it into quadrants, subquadrants,... until we obtain blocks (possibly single pixels) that consist of 1's or 0's; i.e., they are entirely contained in the region or entirely disjoint from it. This process is represented by a tree of out degree 4 (i.e., each non-leaf node has four sons) in which the root node represents the entire array. The four sons of the root node represent the quadrants (labeled in order NW, NE, SW, SE), and the leaf nodes correspond to those blocks of the array for which no further subdivision is necessary. Leaf nodes are said to be BLACK or WHITE depending on whether their corresponding blocks are entirely within or outside of the region respectively. All non-leaf nodes are said to be GRAY. For example, consider the region shown in Figure 1a which is represented by the 2**3 by 2**3 binary array in Figure 1b. Observe that 1's correspond to picture elements (termed pixels) which are in the region and 0's correspond to picture elements that are outside the region. Figure 1c contains the block decomposition for Figure 1b while Figure 1d is the actual quadtree.

The principle of recursive decomposition has been in existence for a long time. One of its first uses was by Warnock [1] in the implementation of a hidden surface elimination algorithm. The picture area is repeatedly subdivided into successively smaller rectangles while searching for areas sufficiently simple to be output. The formulation of the region quadtree as described above is due to Klinger [2,3] who used the term Q-tree whereas Hunter [4-6] was the first to use the term quadtree in such a context.

3. Applications of the Region Quadtree

The region quadtree is relatively compact [3] and is well suited to operations such as union and intersection [4,5,7] and detecting various region properties [3-6]. Hunter and Steiglitz [4-6], in the domain of computer graphics, developed a variety of algorithms for the manipulation of a quadtree region representation. In [8,9] variations of the quadtree (termed an octree) are applied in three dimensions to represent solid objects and in [10] to more dimensions.

There has been much work on the interchangeability between the quadtree and other traditional methods of region representation. Algorithms have been developed for converting a binary array to a quadtree [11], run lengths to a quadtree [12] and a quadtree to run lengths [13], as well as boundary codes to a quadtree [14] and a quadtree to boundary codes [15]. Work has also been done in computing geometric properties such as connected component labeling [16], perimeter [17,18], Euler number [19], areas and moments [7], as well as a distance transform [20,21].

By virtue of its tree-like nature, in the quadtree representation most operations are carried out by techniques which traverse the tree. For each node visited during the traversal, an operation is usually performed at each node which more often than not makes use of neighboring nodes (i.e., nodes representing image blocks that are adjacent to the given node's block).

In general, it is preferable to avoid having to use position (i.e., coordinates) and size information (e.g., [22]) when making relative transitions (i.e., locating neighboring nodes) in the quadtree since they involve computation (rather than simply chasing links) and are clumsy when adjacent blocks are of different sizes (e.g., when a neigboring block is larger). Similarly, it is preferable not to assume the existence of links from a node to its neighbors, termed ropes and nets [5], because it is not desirable to use links in excess of four links from a non-leaf node to its sons and the link from a non-root node to its father. Such techniques, described in [23], are used in [12-17;19-21,24] and result in algorithms that only make use of the existing structure of the tree. Note that an

CH1962-0/83/0000-1072/$00.75 © 1983 IEEE

alternative to computing neighbors is to precompute them in the eight possible directions and to transmit them as actual parameters to the part of the algorithm that examines the neighboring nodes [18].

It turns out that a large number of traditional region processing operations can be achieved by use of the quadtree representation. Many of the methods used on the pixel level carry over to the quadtree domain (e.g., connected component labeling, genus, etc.). Because of its compactness, the quadtree permits faster execution of these operations. Often the quadtree algorithms require time proportional to the number of blocks in the image, independent of their size.

The quadtree data structure requires storage for the various links. However, use of neighbor finding techniques rather than ropes a la Hunter [5] is one compromise. In fact, experimental results discussed in [25] show that the extra storage cost of ropes is justified by the resulting minor decrease in execution time. This is because the average number of links traversed by neighbor finding methods is 3.5 in contrast with 1.5 for ropes. Nevertheless, there is a possibility that the quadtree may not be efficient spacewise. For example, a checkerboard-like region does not lead to economy of space. Also, they are not shift-invariant (but see [24]). The space efficiency of the quadtree is analyzed in [26]. Some savings can be obtained by normalizing the quadtree [27,28] as is also possible by constructing a forest of quadtrees [29] to avoid large regions of WHITE. Storage can also be saved by using a locational code for all BLACK blocks [30,31]. Gray level quadtrees using a sequence of gray codes to economize on storage are reported in [32].

The variable resolution property of the quadtree sets it apart from a representation based on a hexagonal decomposition [33] - i.e., a square can be repeatedly decomposed into smaller squares (as can be done for triangles as well [34]); whereas once the smallest hexagon has been chosen, it can not be further decomposed into smaller hexagons.

The quadtree is especially useful for point in polygon operations as well as for query operations involving image overlays and set operations. The hierarchical nature enables one to use image approximations. In particular, a breadth first transmission of an image yields a successively finer image yet enables the user to have a partial image [35]. Thus the quadtree could be used in perusing through a large image database.

The region quadtree is often characterized as a variable resolution data structure. Somewhat related is the pyramid [36] which is a multiple resolution data structure. In essence, it is an exponentially tapering stack of arrays each at one half of the resolution of the previous one and hence one quarter of the size. It is discussed in considerable detail in [37].

4. Line Data

As mentioned in Section 2, regions may be described in two ways - those that describe their interior and those that specify their borders. In this section we concentrate on hierarchical border representations. One of the most common border representations is the chain code [38]. Another very popular representation is a description using vectors [39], e.g., for polygons, in the form of lists of pairs of x and y coordinate values corresponding to their start and end points. Both of these representations have the advantage of being compact. However, they are very local data structures. This makes them cumbersome for the performance of set operations such as union and intersection.

Recently, there has been a growing amount of interest in hierarchical representations of border information. These are primarily based on rectangular approximations to the data. In particular, they include upright rectangles [40], rectangular strips of arbitrary orientation [41], and sets of bands [42]. It is interesting to note that the method of [40], resulting in a data structure termed BSPR, can be characterized as being bottom-up while the method of [41] uses a top-down approach resulting in a data structure that is termed a strip tree. Both the BSPR and the strip tree are independent of the grid system in which they are embedded. The strip tree requires special handling for closed curves whereas the BSPR does not. However, the BSPR is not as flexible as the strip tree since the resolution of the approximation is fixed as the width of the rectangle may not be varied whereas the strip tree has no such problem.

There also exist methods of describing boundaries of regions by methods that are based on a regular decomposition. Hunter and Steiglitz [5] represent a boundary as a sequence of BLACK pixels in a region quadtree. All the remaining nodes can be treated as WHITE nodes. Warnock [1] and Shneier [43] subdivide the set of lines comprising the boundaries until obtaining square regions containing a single curve that can be approximated by a single straight line. Some other approaches include the line quadtree [44], PM quadtree [45], and the EXCELL method [46]. Regular decomposition methods have an advantage over the strip tree in that more than one curve can be represented. Also, they are unique whereas the strip tree is not unique when the curve is closed or extends past its endpoints. Line quadtrees are superior to the representation of Hunter and Steiglitz in that they do not commit a specific thickness for the boundary line. The PM quadtree and the EXCELL method have the property that they yield an exact representation rather than being an approximation in the case of polygons as may be the case when the other regular decomposition methods are used.

5. Point data

Multidimensional point data can be represented by a multidimensional generalization of a binary search tree. Actually, it is a marriage between the grid method which is used by cartographers [47] and the binary search tree. The resulting data structure is also named a quadtree [48]. However, in order to distinguish it from the region quadtree, we shall refer to it as a point quadtree. In two dimensions, each data point is a node in a tree having four sons which are roots of subtrees corresponding to quadrants labeled in order NW, NE, SW, and SE. Each data point is assumed to be unique. Data points are inserted into a point quadtree by searching for the record based on its x and y coordinates. At each node of the tree, a four-way comparison operation is performed and the approriate subtree is chosen for the next test. Reaching the bottom of the tree without finding the record means that it should be inserted at that position. Clearly, the shape of the resulting tree depends on the order in what records are inserted into it.

Point quadtrees are particuarly attractive in applications that involve search in that they enable a pruning of the search space that is involved in the query. It should be clear that the point quadtree can be generalized for an arbitrary number of dimensions. However, in such a case there is a rather large branching factor associated with each point quadtree node (2**k for k dimensions). The k-d tree [49] is an improvement on the point quadtree. In essence, it is a binary search tree with the distinction that at each level of the tree a different coordinate is tested when determining the direction in which a branch is to be made. Thus in the two-dimensional case, we compare x coordinates at the root and at odd levels of the tree and y coordinates at even levels. In [50] an improvement on the k-d tree is reported which removes the requirements of alternating the tests at the price of requiring the storage with each node of the coordinate being tested.

There also exists methods of representing multidimensional point data that are based on a regular decomposition. The point space is repeatedly decomposed into four equal-sized square regions until no region contains more than a predetermined number of points [51]. A similar decomposition method termed EXCELL is also in use [46] as well as one that is not based on a regular decomposition [52]. These methods are often characterized as bucket methods.

6. Concluding Remarks

A brief review of a number of hierarchical data structures for representing data useful in image processing application has been given. The data structures that we have discussed are not unique and in the future we will most likely see further variations on the concepts discussed here.

References

[1] J. L. Warnock, "A hidden surface algorithm for computer generated half tone pictures," Computer Science Department TR 4-15, University of Utah, Salt Lake City, June 1969.

[2] A. Klinger, "Patterns and Search Statistics," in Optimizing Methods in Statistics. J. S. Rustagi, Ed., New York: Academic Press, 1971.

[3] A. Klinger and C. R. Dyer, "Experiments in picture representation using regular decomposition," Computer Graphics and Image Processing, vol. 5, pp. 68-105, 1976.

[4] G. M. Hunter, "Efficient computation and data structures for graphics," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ, 1978.

[5] G. M. Hunter and K. Steiglitz, "Operations on images using quadtrees," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, pp. 145-153, 1979.

[6] G. M. Hunter and K. Steiglitz, "Linear transformation of pictures represented by quadtrees," Computer Graphics and Image Processing, vol. 10, pp. 289-296, 1979.

[7] M. Shneier, "Calculations of geometric properties using quadtrees," Computer Graphics and Image Processing, vol. 16, pp. 296-302, 1981.

[8] C. L. Jackins and S. L. Tanimoto, "Oct-trees and their use in representing three-dimensional objects," Computer Graphics and Image Processing, vol. 14, pp. 249-270, 1980.

[9] D. Meagher, "Geometric modeling using octree encoding," Computer Graphics and Image Processing, vol. 19, pp. 129-147, 1982.

[10] M. Yau and S. N. Srihari, "Recursive generation of hierarchical data structures for multidimensional digital images," in Proceedings of the IEEE Conference on Pattern Recognition and Image Processing, Dallas, 1981, 42-44.

[11] H. Samet, "Region representation: quadtrees from binary arrays," Computer Graphics and Image Processing, vol. 18, pp. 88-93, 1980.

[12] H. Samet, "An algorithm for converting rasters to quadtrees, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 3, pp. 93-95, 1981.

[13] H. Samet, "Algorithms for the conversion of quadtrees to rasters," to appear in Computer Graphics and Image Processing, 1983 (also University of Maryland Computer Science TR-979).

[14] H. Samet, "Region representation: quadtrees from boundary codes," Communications of the ACM, vol. 23, pp. 163-170, March 1980.

[15] C. R. Dyer, A. Rosenfeld, and H. Samet, "Region representation: boundary codes from quadtrees," Communications of the ACM, vol. 23, pp. 171-179, March 1980.

[16] H. Samet, "Connected component labeling using quadtrees," Journal of the ACM, vol. 28, pp. 487-501, July 1981.

[17] H. Samet, "Computing perimeters of images represented by quadtrees," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 3, pp. 683-687, 1981.

[18] C. Jackins and S. L. Tanimoto, "Quad-trees, oct-trees, and k-trees - a generalized approach to recursive decomposition of Euclidean space," Department of Computer Science Technical Report 82-02-02, University of Washington, Seattle, 1982.

[19] C. R. Dyer, "Computing the Euler number of an image from its quadtree," Computer Graphics and Image Processing, vol. 13, pp. 270-276, 1980.

[20] M. Shneier, "Path-length distances for quadtrees," Information Sciences, vol. 23, pp. 49-67, 1981.

[21] H. Samet, "Distance transform for images represented by quadtrees," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 4, pp. 298-303, 1982.

[22] A. Klinger and M. L. Rhodes, "Organization and access of image data by areas," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, pp. 50-60, 1979.

[23] H. Samet, "Neighbor finding techniques for images represented by quadtrees," Computer Graphics and Image Processing, vol. 18, pp. 37-57, 1982.

[24] H. Samet, "A quadtree medial axis transform," to appear in Communications of the ACM, 1983 (also University of Maryland Computer Science TR-803).

[25] 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.

[26] C. R. Dyer, "The space efficiency of quadtrees," Computer Graphics and Image Processing, vol. 19, pp. 335-348, 1982.

[27] M. Li, W. I. Grosky, and R. Jain, "Normalized quadtrees with respect to translations," Computer Graphics and Image Processing, vol. 20, pp. 72-81, 1982.

[28] W. I. Grosky and R. Jain, "Optimal quadtrees for image segments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5, pp. 77-83, 1983.

[29] 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.

[30] I. Gargantini, "An effective way to represent quadtrees," Communications of the ACM, vol. 25, pp. 905-910, December 1982.

[31] D. J. Abel and J. L. Smith, "A data structure and algorithm based on a linear key for a rectangle retrieval problem," to appear in Computer Graphics and Image Processing, 1983.

[32] E. Kawaguchi, T. Endo, and J. Matsunaga, Depth-first-expression viewed from digital picture processing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5, pp. 373-384, 1983.

[33] L. Gibson and D. Lucas, "Vectorization of raster images using hierarchical methods," Computer Graphics and Image Processing, vol. 20, pp. 82-89, 1982.

[34] N. Ahuja, "On approaches to polygonal decomposition for hierarchical image representation, to appear in Computer Graphics and Image Processing, 1983 (see also Proceedings of the IEEE Conference on Pattern Recognition and Image Processing, Dallas, 1981, 75-80).

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

[36] S. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing," Computer Graphics and Image Processing, vol. 4, pp. 104-119, 1975.

[37] A. Rosenfeld, Ed., Multiresolution Image Processing and Analysis. Berlin: Springer-Verlag, 1983.

[38] H. Freeman, "Computer processing of line-drawing images, ACM Computing Surveys, vol. 6, pp. 57-97, March 1974.

[39] G. Nagy and S. Wagle, "Geographic data processing," ACM Computing Surveys, vol. 11, pp. 139-181, June 1979.

[40] W. Burton, "Representation of many-sided polygons and polygonal lines for rapid processing," Communications of the ACM, vol. 20, pp. 166-171, March 1977.

[41] D. H. Ballard, "Strip trees: A hierarchical representation for curves," Communications of the ACM, vol. 24, pp. 310-321, May 1981 (see also corrigendum, Communications of the ACM, vol. 25, p. 218, March 1982).

[42] T. Peucker, "A theory of the cartographic line," International Yearbook of Cartography, 1976.

[43] M. Shneier, "Two hierarchical linear feature representations: edge pyramids and edge quadtrees," Computer Graphics and Image Processing, vol. 17, pp. 211-224, 1981.

[44] H. Samet and R. E. Webber, "Line quadtrees: a hierarchical data structure for encoding boundaries, in Proceedings of the IEEE Conference on Pattern Recognition and Image Processing, Las Vegas, 1982, 90-92 (also University of Maryland Computer Science TR-1162).

[45] H. Samet and R. E. Webber, "Using quadtrees to represent polygonal maps to appear in the Proceedings of Computer Vision and Pattern Recognition, Washington, DC, 1983, vol. 83, 127-132.

[46] M. Tamminen, "The EXCELL method for efficient geometric access to data," Acta Polytechnica Scandinavica, Mathematics and Computer Science Series No. 34, Helsinki, 1981.

[47] J. L. Bentley and J. H. Friedman, "Data structures for range searching," ACM Computing Surveys, vol. 11, pp. 397-409, December 1979.

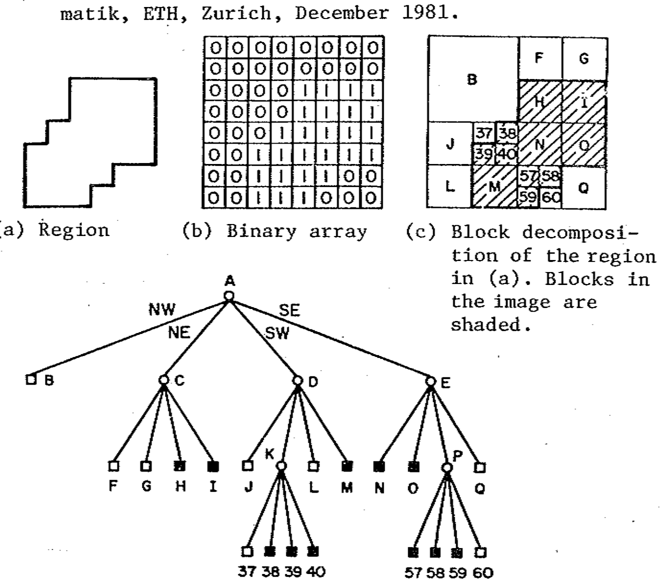
[48] R. A. Finkel and J. L. Bentley, "Quad trees: a data structure for retrieval on composite keys," Acta Informatica, vol. 4, pp. 1-9, 1974.

[49] J. L. Bentley, "Multidimensional binary search trees used for associative searching," Communications of the ACM, vol. 18, pp. 509-517, September 1975.

[50] J. H. Friedman, J. L. Bentley, and R. A. Finkel, An algorithm for finding best matches in logarithmic expected time," ACM Transactions on Mathematical Software, vol. 3, pp. 209-226, September 1977.

[51] T. Matsuyama, L. V. Hao, and M. Nagao, "A file organization for geographic information systems based on spatial proximity," to appear in Computer Graphics and Image Processing, 1983.

[52] J. Nievergelt, H. Hinterberger, and K. C. Sevcik, "THE GRID FILE: an adaptable, symmetric multi-key file structure," Report 46, Institut fur Informatik, ETH, Zurich, December 1981.



(a) Region (b) Binary array (c) Block decomposition of the region in (a). Blocks in the image are shaded.

(d) Quadtree representation of the blocks in (c).

Figure 1. A region, its binary array, its maximal blocks, and the corresponding quadtree.