

On Approaches to Polygonal Decomposition for Hierarchical Image Representation

NARENDRA AHUJA

*Coordinated Science Laboratory and Department of Electrical Engineering, University of Illinois,
Urbana, Illinois 61801*

Received June 9, 1982; revised September 13, 1982

Approaches to polygonal decomposition for hierarchical image representation are described. For planar decomposition, quad trees using square and triangular neighborhoods are found to be the only feasible methods, having the same computational complexity. For grid images the choice of the appropriate tree type is determined by the grid topology. Triangular and square quad trees are appropriate for the triangular and square grids, whereas trees of order 7 are necessary for the hexagonal grid.

1. INTRODUCTION

The major approaches to image representation [1] may be divided into two broad categories: (1) those which specify the borders of the regions, and (2) those which describe their interiors. Most of the approaches, and the more interesting ones, belong to the latter category. This may be attributed to the increased dimensionality of information (regions, rather than curves) to be represented. An important subclass of these methods, called medial axis transforms (MAT) [1], involves representation of the regions by a set of maximal blocks, say, squares. Each maximal square lies completely within a single region, and is not contained in any other such square. The squares may have any size and may be placed anywhere in the image as determined by the locations of the regions. An alternative method, called quad-tree representation, was proposed by Klinger and Dyer [2] and Tanimoto and Pavlidis [3]. Their approach also involves square blocks of various sizes, but the locations of the squares are fixed in advance and are the same for any image. A recursive decomposition of squares into quadrants is used to obtain blocks of smaller sizes (Fig. 1).

As in case of MAT, blocks other than squares may also be considered for use in decomposition of the plane. This paper attempts to enumerate possible planar decomposition schemes that use polygonal blocks. Section 2 discusses planar partitioning schemes. Section 3 describes quad trees that use triangular tessellation of the plane. It is argued that the triangular quad trees are the only feasible alternative to quad trees using the square tessellation. The performances of the two methods are compared. In Section 4 we examine decomposition methods for grids. Section 5 presents concluding remarks.

2. PARTITIONING THE PLANE

Any planar decomposition scheme for image representation must possess the following properties.

(1) The partition should be an infinitely repetitive pattern in the plane. This would allow the representation to be useful for images of any size.

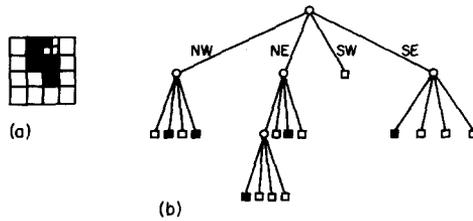


FIG. 1. (a) A binary image. (b) Quad tree for (a). Black leaves denote black regions.

(2) The partition should be infinitely (recursively) decomposable into increasingly fine patterns. This would allow the representation to be useful for images with arbitrarily fine spatial features.

We will now examine the different schemes that could be used for recursive decomposition of the plane using partitions consisting of polygonal cells. We will not consider nonpolygonal partitions such as exponential tessellations that are based on logarithmic spirals [8] rather than Cartesian coordinates.

2.1 Partitioning Schemes

Let k denote the number of sides of a face (cell) in a given partition. Let v denote the number of cells meeting at a vertex. Consider the partitions in which the value of $k(v)$ is the same for all cells (vertices), i.e., all cells are regular polygons. We call such tessellations kv regular tessellations. Now, the interior angle formed by each adjacent pair of edges in a regular k -gon is $\pi(k-2)/k$. By considering the v k -gons meeting at a point, it is apparent [9] that v and k must satisfy the equation $\pi v(k-2)/k = 2\pi$. Accordingly, there exist only three kv regular tessellations [4, 5, 9, 10]—the possible (k, v) values being $(3, 6)$, $(4, 4)$, and $(6, 3)$. They correspond to the division of the plane into regular triangular, square (rectangular), and hexagonal cells, respectively (Fig. 2). The triangular and the hexagonal tessellations form a pair of dual graphs.

If the value of k is allowed to vary from cell to cell, keeping v fixed, the resulting tessellations are called semiregular tessellations [5]. By definition, these partitions do not consist of congruent cells, but a mixture of as many different regular k -gons as there are different values of k . Using an approach similar to that for regular tessellations outlined in the previous paragraph, it can be shown that only eight semiregular tessellations are possible. A semiregular tessellation may be characterized by an ordered sequence of v integers, where the i th integer denotes the number of sides of the i th cell surrounding a vertex, starting at any of the surrounding cells arbitrarily and moving, say, clockwise. In this notation, the eight

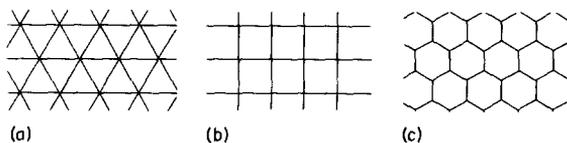


FIG. 2. Regular tessellations: (a) triangular, (b) square, (c) hexagonal.

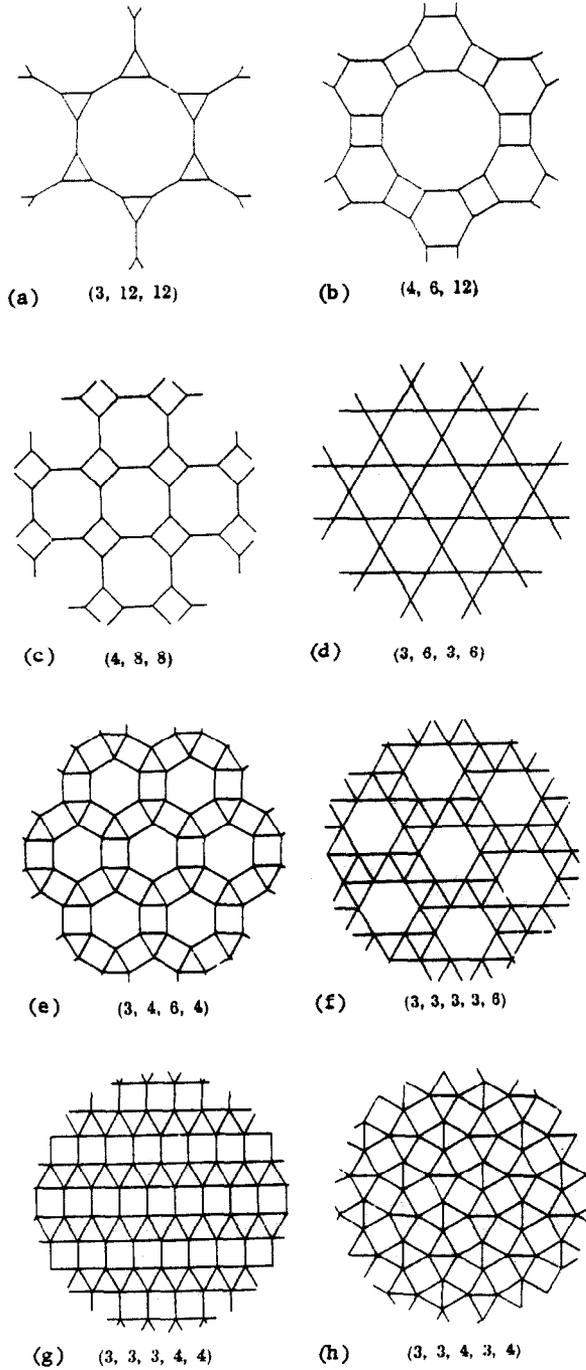


FIG. 3. Semiregular tessellations (from [5]).

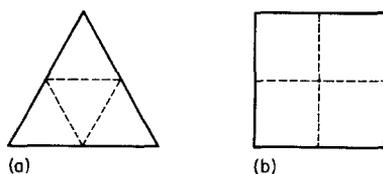


FIG. 4. Cells (dotted lines) in (a) triangular and (b) square tessellations merge into larger cells (thick lines).

semiregular tessellations are given by $(3, 12, 12)$, $(4, 6, 12)$, $(4, 8, 8)$, $(3, 6, 3, 6)$, $(3, 4, 6, 4)$, $(3, 3, 3, 3, 6)$, $(3, 3, 3, 4, 4)$, and $(3, 3, 4, 3, 4)$ (Fig. 3).

Both regular and semiregular tessellations possess the first property demanded from a planar partition for image representation given earlier. We now examine them with respect to the second requirement, namely, the recursive decomposability. Cells in a regular tessellation are all congruent. If we can partition each cell further into smaller cells such that the new tessellation still is a kv regular tessellation having the same kv values, then the infinite decomposability requirement is met. Alternatively, it should be possible to merge cells locally to obtain a kv regular tessellation having larger cells and the same values of k and v .

Clearly, the triangular and square tessellations possess this property (Fig. 4). On the other hand, cells in a hexagonal tessellation cannot be further divided into regular congruent hexagons. To prove this, imagine merging neighboring hexagons (of side d) in a regular tessellation to form a larger hexagon. By the requirement of recursive decomposability, the edges of the larger hexagon must be contained in the given tessellation. However, all the straight line segments in the tessellation are of length d . They cannot possibly define hexagons of side longer than d . A similar argument rules out all semiregular tessellations except for $(3, 6, 3, 6)$ and $(3, 3, 3, 3, 6)$ (Figs. 3d, f). In the latter two, the placement of star-shaped cells leaves holes (Figs. 5a, b). Thus adjacent cells cannot merge to form a larger cell of the same shape, making the tessellation recursively nondecomposable. Here, we allow a single type of cell and decomposition scheme. Thus, for example, we do not allow triangles and hexagons as two different types of cells for $(3, 6, 3, 6)$ employing different decomposition schemes (Fig. 5c). The regular square and triangular tessellations, therefore, are the only partitioning schemes that place no restriction on the resolution at which an image can be represented.

If an upper limit on the coarsest allowable image resolution is acceptable, then some of the semiregular tessellations can also be used by starting with a tessellation having the desired cell dimensions and recursively dividing each cell independently. Thus, for example, with two different types of cells and decomposition schemes, the semiregular tessellation $(3, 6, 3, 6)$ can be used as illustrated in Fig. 5c. In addition, we may also consider the partitions generated by completely regular polygonal graphs, which are graphs contained within a polygon such that the $k(v)$ value is the same for all cells (vertices). Completely regular polygonal graphs differ from the regular and semiregular tessellations partly in their inability to cover the entire plane by repetition. It can be shown that there exist only five different completely regular polygonal graphs [4]. The corresponding (k, v) values are $(3, 3)$, $(4, 3)$, $(5, 3)$, $(3, 4)$, and $(3, 5)$. All cells are not regular polygons (Fig. 6). However, the parent graph can

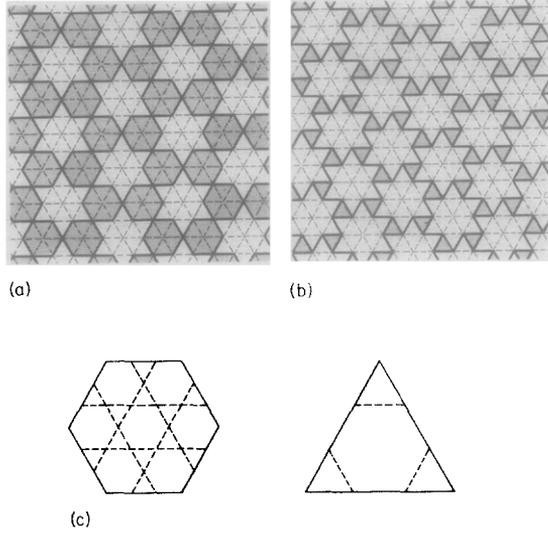


FIG. 5. The placement of the star-shaped tile in the semiregular tessellations (a) $(3,6,3,6)$ and (b) $(3,3,3,3,6)$ leaves holes (hatched) between adjacent tiles. (c) An example of a pair of cells and their decompositions for $(3,6,3,6)$. Such multicell recursive decomposition schemes are not acceptable.

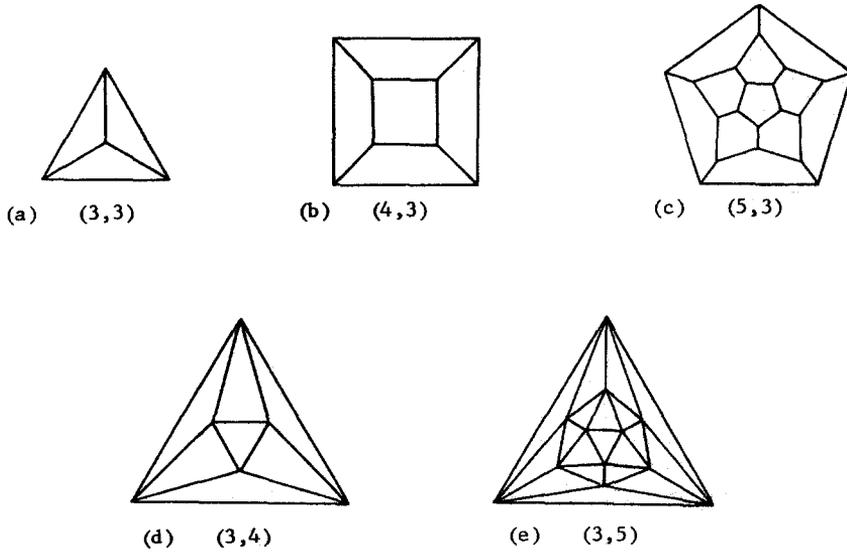


FIG. 6. Completely regular polygonal graphs (from [4]).

be redrawn within each cell to define a recursive partition (Fig. 7). This increases the value of v at all vertices at every step of decomposition, in contrast with the regular or semiregular tessellations, where only new vertices are generated, keeping the degree of the vertices fixed. As a result, cells in completely regular graphs become increasingly oblong as finer partitions are generated. Cells in graphs consisting of only quadrangles and triangles (Figs. 6a, b, d, e) may be partitioned in a way similar to the decomposition of regular square and triangular tessellations, thus keeping the degrees of the new vertices fixed.

2.2 Feasibility and Appropriateness

Since cells in a semiregular tessellation cannot be merged (Sect. 2.1), the semiregular tessellations cannot be used to partition the cells. However, those semiregular tessellations consisting of square, regular triangular, and/or hexagonal cells (Fig. 3) can be recursively partitioned by performing square decomposition of square cells and regular triangular decomposition of triangular and hexagonal cells. Such a scheme differs from the regular tessellations only in the initial selection of image windows where the two decomposition methods are applied. However, the use of different window shapes in different parts of the image increases the sensitivity of the size of the representation to translation and rotation.

The cells in completely regular graphs are irregular and variable in shape. Decomposition of the cells using the parent graph yields increasingly irregular cells (Fig. 7). As decomposition progresses, the cells become increasingly oblong. Thus, the representation becomes more sensitive to, say, linelike features. This effect is less pronounced whenever a decomposition similar to regular tessellations is possible for the cells in completely regular graphs, e.g., for those in Figs. 6a, b, d, e.

The complexity of computer data structures to represent recursive partitions defines another criterion for comparing different approaches. Clearly, the semiregular tessellations and completely regular polygonal graphs require more complex data structures than the regular tessellations, resulting in slower image computations. This is because the former must store information about more than one partitioning scheme, and/or about the regions where different schemes apply.

Thus, we see that the square and triangular regular tessellations are the most well behaved among all the planar, polygonal partitioning schemes in the sense we have

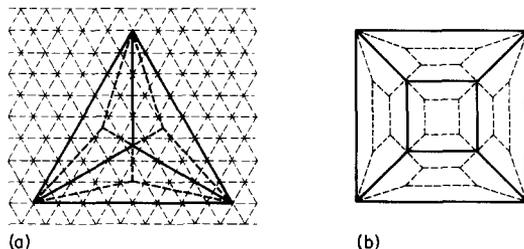


FIG. 7. Examples of decomposition of cells in completely regular polygonal graphs using completely regular polygonal graphs: (a) (3,3) and (b) (4,3). Cells consisting of old (thick) edges become more oblong.

discussed. Since in a regular square decomposition each square is partitioned into four smaller squares, the nodes in a tree representation of the partition have four children each. Such (square quad) trees have received extensive attention recently [6]. In the next section we examine the relative performance of the triangular tessellation.

3. TRIANGULAR QUAD TREES

To decompose a triangular tessellation recursively, each triangle is divided into four others (Fig. 8). Each node corresponding to a triangle thus has four children, making it a quad tree. Each node also has three neighboring triangles. A triangle has one of two orientations that differ by 60° . Among the four children, the central triangle differs in orientation from its parent (Fig. 8). Let us label it as child node 1. The remaining three children all assume the orientation of the parent. Let us label these 2, 3, and 4, clockwise, starting from the top (Fig. 8a) or top left (8b). All three neighbors of any node labeled 1 are its siblings. A node labeled 2, 3, or 4, on the other hand, has exactly one sibling among its neighbors.

By comparison, a square quad tree has all squares at the same orientation. Each square has four neighbors; exactly two of these are its siblings. In this sense, the triangular quad tree exhibits less homogeneity from node to node.

The basic structure of the algorithms using triangular or square tessellations may be the same since both involve quad trees. Thus, the various algorithms proposed for computing image properties using square quad trees (see [6] and "Representation"—Chap. 11 of [1] and references therein) may easily be modified to work in case of triangular quad trees. Operations involving a single node, e.g., computation of its area, centroid, etc., have the same order of complexity, since the only difference is in the geometry of the nodes. The other kind of operations involve access to a node's neighbors. Complexity of such computations depends upon the complexity of the neighbor-accessing mechanism. The relative performances of the two quad trees may, therefore, be determined in terms of the complexities of finding neighbors of a node, in their respective tree structures.

Suppose we want to access a randomly chosen neighbor N of a leaf node L in a complete quad tree. Let p be the probability that N is a sibling of L . Then, with probability p , N is accessed by traversing two links in the tree: uplink to the parent and downlink to sibling (two operations). With probability $(1 - p)$, N will be reached via the closest common ancestor of L and N at a higher level. This may take K operations, $K = 4, 6, \dots$, depending upon the number of links between L or N , and their closest common ancestor. Let f_{2k} denote the probability that $K = 2k$, given that $k > 1$. If P_{2k} denotes the probability that $2k$ operations are required to reach N

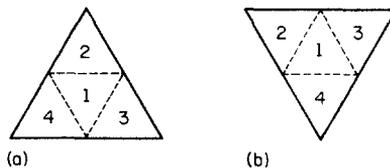


FIG. 8. Decomposition of a triangle into four triangles.

from L , we have

$$P_{2k} = \begin{cases} p & k = 1 \\ (1 - p)f_{2k} & k = 2, 3, \dots \end{cases}$$

Given a pair of nodes L and N , the function f depends only upon the tree structure and the locations of L and N in the tree. Thus, f is the same for both triangular and square quad trees. The only other parameter that affects P_{2k} is p .

Every node in a square quad tree has exactly two siblings as its neighbors. Therefore, $p = \frac{1}{2}$. In the case of triangular quad trees, the probability that a randomly chosen neighbor of a node is its sibling is 1 for nodes labeled 1 and $1/3$ for each of the remaining three types of nodes. Since L is equally likely to have any of the four labels, we have

$$\begin{aligned} p &= 1/4 \cdot 1 + 3/4(1/3) \\ &= 1/2 \end{aligned}$$

which is the same as for square quad trees. Thus, the probability P_{2k} that $2k$ operations are required to access N from L is the same for both square and triangular quad trees and is given by

$$\begin{aligned} P_{2k} &= \begin{cases} \frac{1}{2} & k = 1 \\ \frac{1}{2}f_{2k} & k > 1 \end{cases} \\ &= \frac{1}{2}\delta_{k1} + \frac{1}{2}(1 - \delta_{k1})f_{2k} \end{aligned}$$

where δ_{ij} is the Kronecker delta function. If N is allowed to occur on a level different from that of L , then P_n is the same as f_n , which, as already pointed out, is the same for both representations.

Therefore, the complexity of image operations in the square and triangular quad tree representations appears to be the same.

4. GRID IMAGES

In the previous sections, we have described approaches to recursive decomposition of the Euclidean plane. Quad trees using squares or triangles are found to have the same computational complexity and either one may be used for a given pattern. However, for patterns on grid, the grid topology dictates the choice of the tree type. For example, on a triangular grid triangular blocks are easy to define, but there is no natural definition of square blocks. Conversely, square quad trees are appropriate for square grid. This may suggest that the method of image decomposition should be chosen in accordance with the choice of sampling neighborhood used to obtain the grid version of the Euclidean image to begin with. In this section, we will consider various possible tessellations of the three basic grids and identify the most desirable decompositions in each case. We will examine different decomposition patterns and the grid tessellations generated by each. The appropriateness of a given pattern-tessellation pair will be judged in terms of the following properties [7]:

- (1) the pattern should tessellate the grid;
- (2) the centroids of the patterns in the tessellation should themselves form the parent grid;

(3) the pattern should be compact.

Different tessellations of the grid may be obtained by considering (a) patterns of different shapes and sizes, and (b) different arrangements of patterns of a fixed size and shape. The number and complexity of possible shapes of decomposition patterns grow with size. Each pattern may be arranged in more than one way to give rise to different tessellations. To enumerate all possible tessellations for an arbitrary n appears to be a difficult problem. However, the fraction of noncompact decomposition patterns increases with the pattern size. These noncompact patterns may not be acceptable. Compact large patterns are often similar in shape to compact smaller patterns. The use of larger patterns only results in a shallower tree for a given image. In the following discussion, we will consider patterns of sizes 1 through 7.

4.1 Triangular Grid

Figure 9 shows topologically distinct patterns on a triangular grid for $2 \leq n \leq 6$ and the corresponding Euclidean plane regions (tiles). Figure 10 shows examples of tessellations for some of the patterns in Fig. 9. For $n = 2$ and $n = 3$, the only possible tessellating tiles of Figs. 9a, b are convex and have four sides. Therefore, a

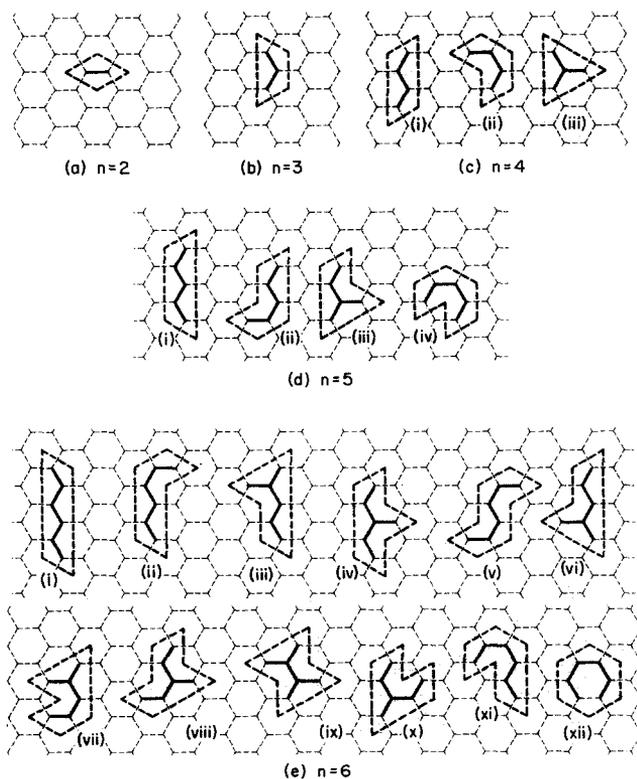


FIG. 9. Topologically distinct triangular grid decomposition patterns of sizes 2 to 6 (thick lines). Heavy dashed lines mark the borders of the corresponding tiles in the Euclidean plane.

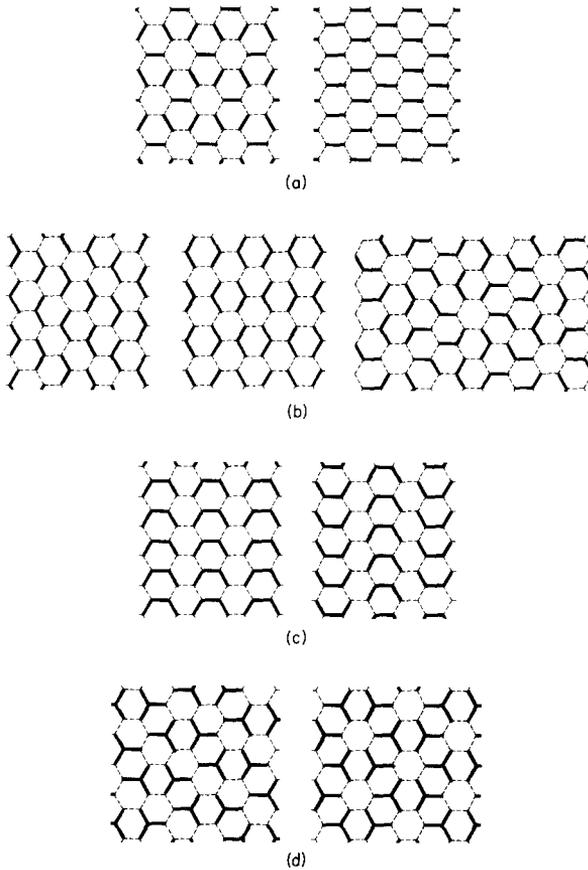


FIG. 10. Examples of triangular grid tessellations using patterns of (a) Fig. 9a(ii), (b) Fig. 9b, (c) Fig. 9c(ii) and c(iii).

tile must share edges with at least four others in any conceivable tessellation. The centroids of the tiles, therefore, cannot possibly define a triangular grid. The same comments hold for patterns (i) in Figs. 9c, d, and e. The remaining patterns in Figs. 9c, d, and e, except for c (iii) and d (xii), do not meet the compactness requirement mentioned earlier. In addition, the nonconvexity, the large number of sides, and the large variance of side lengths limit the number of tessellations possible using these tiles. Among any tessellations that are possible, the large ratio of the numbers of convex and concave corners, and the large number (> 3) of sides per tile results in more than three neighbors for each tile. This in turn makes the grid formed by the centroids of the tiles of order greater than three, i.e., nontriangular. The hexagonal tile of Fig. 9e (xii), although compact, provides a hexagonal grid. The patterns for $n = 7$ are obtained by adding one more point to the patterns in Fig. 9e. By inspection, it is clear that none yields an acceptable tessellating pattern.

As the value of n increases, the simple compact shapes (quadrangle, triangle, and hexagon) recur with increased sizes. For example, triangular patterns of side 3, 5, 7, ... points are possible. We will consider only the smallest tile of a given shape. In addition to the recurrent shapes, an increasingly large variety of more

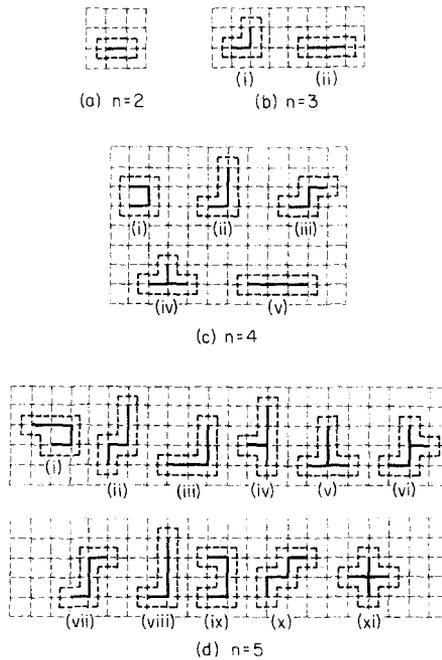


FIG. 11. Topologically distinct square grid decomposition patterns of size 2 to 5 (thick lines). Heavy dashed lines mark the borders of the corresponding tiles in the Euclidean plane.

complex shapes also emerges that violates the compactness and/or grid criterion. For $2 \leq n \leq 7$, only the tessellating pattern in Fig. 9c (iii) corresponding to a triangular tile satisfies all the requirements described earlier, resulting in a quad tree representation ($n = 4$) of the grid image (Figs. 13a, 14a).

4.2 Square Grid

Figure 11 shows topologically distinct patterns for $2 \leq n \leq 5$ and the corresponding Euclidean plane tiles. All but the patterns in Figs. 11a, b(ii), c(i, v) have irregular shapes and may be ruled out for reasons similar to those discussed in Section 4.1. The patterns in Fig. 11a, b(ii), and c(i, v) are all quadrangles and satisfy our first two criteria. The square pattern of Fig. 11c(i) is the most compact. The patterns for $n = 6, 7$ are large in number. It is easily seen by appending one ($n = 6$) or two ($n = 7$) additional points to the patterns in Fig. 11d that all of the resulting patterns either continue to remain irregular in shape or are rectangles, both of which are not acceptable for the reasons discussed in Section 4.1. The smallest square pattern (2×2), thus, is the only acceptable tessellating pattern ($n = 4$). This suggests the use of a quad tree (Figs. 13b, 14b) representation, which has already been pursued for some time [2, 3, 6].

4.3 Hexagonal Grid

While the regular hexagonal tessellation of the Euclidean plane cannot be recursively decomposed, this does not rule out the existence of recursive decomposition methods for the hexagonal grid. Figure 12 shows distinct patterns for $2 \leq n \leq 5$.

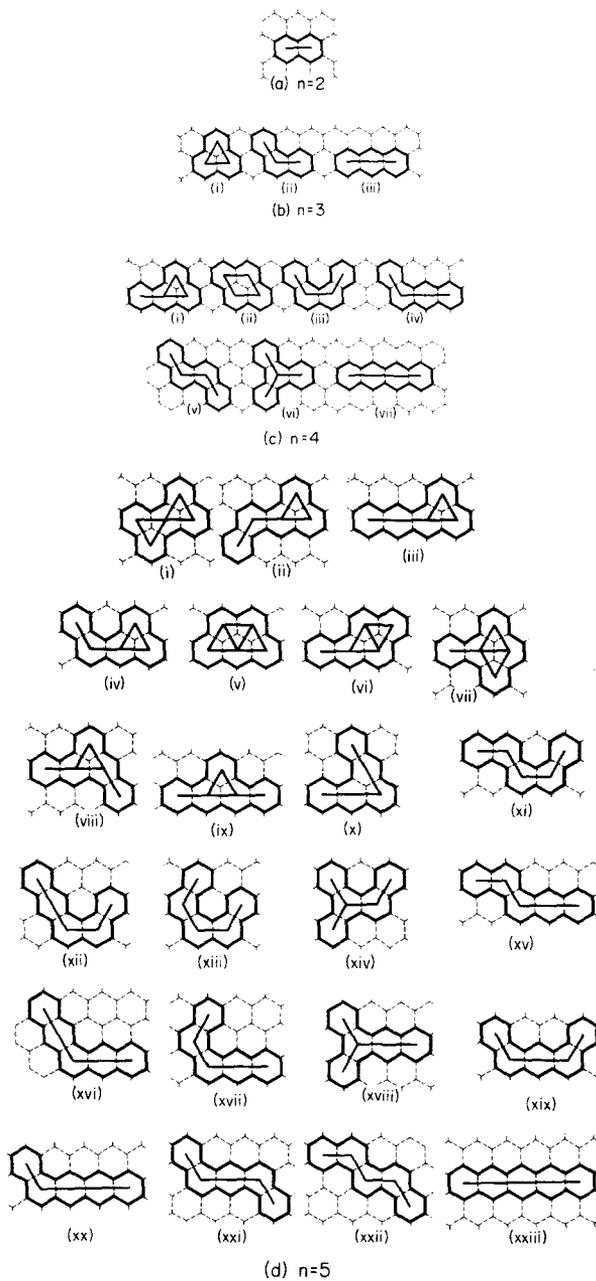


FIG. 12. Topologically distinct hexagonal grid decomposition patterns of sizes 2 to 5 (thin lines). Thick lines mark the borders of the corresponding tiles in the Euclidean plane.

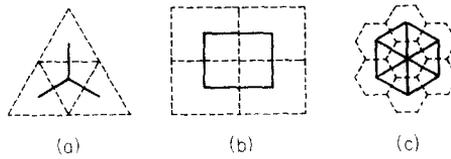


FIG. 13. Acceptable decomposition patterns and their corresponding Euclidean plane tiles for the (a) triangular grid, (b) square grid, and (c) hexagonal grid.

Most of these are irregular and can be eliminated from consideration. The relatively symmetric or compact patterns of Figs. 12b(i), c(ii, vi), d(i, v) cannot tessellate the grid to give a coarser hexagonal grid. Patterns for $n = 6$ can be obtained by adding a point each to those in Fig. 12d. All but one such pattern (Fig. 12e), obtained from Fig. 12d(x), are noncompact. This compact pattern, however, fails to provide a hexagonal grid after tessellation. For $n = 7$ also, all but one pattern (Fig. 12f), obtained by adding two points each to the patterns in Fig. 12d (i, v, vi, xii), are unsatisfactory. Thus, for the hexagonal grid, the simplest decomposition results in a sept tree representation of the image (Figs. 13c, 14c). Such sept trees have already been proposed [7] for multiresolution pyramid representation of images, although the acceptability criteria there do not permit tessellations with multiple orientations of a grid pattern. However, as it turns out, the more general case treated here that allows multiple orientations of a pattern does not result in any new decomposition methods in addition to Burt's [7] sept trees.

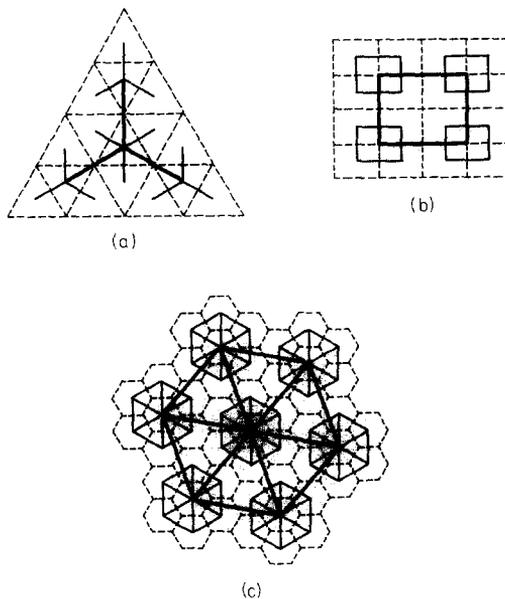


FIG. 14. Grid tessellations at different resolutions (levels) generated by the decomposition patterns shown in Fig. 13.

5. CONCLUDING REMARKS

We have considered various possible approaches to recursive decomposition of images to obtain hierarchical (tree) representations. We have discussed only those approaches involving polygonal tiles. We have not considered decomposition of an image using nonpolygonal partitions such as exponential tessellations that are based on logarithmic spirals [8], rather than Cartesian coordinates. For the case of the Euclidean plane images, we concluded that the square and triangular tessellations define the only viable approaches. Squares in the former have a fixed orientation, whereas the latter involves triangles having two orientations differing by 60° . Both methods appear to have the same computational complexity. However, such a choice between square and triangular quad trees is not available for images on a grid. The nature of the grid dictates the type of tree to be used. For example, on a triangular grid there is no natural definition of square blocks; triangular quad trees are natural. Conversely, square quad trees are appropriate for the square grid. This may be interpreted as saying that the choice of the method for image decomposition should not be in conflict with the choice of sampling neighborhoods used to obtain the square or triangular grid version of the Euclidean image to begin with. Nontriangular (nonsquare) tessellating patterns of the triangular (square) grid are less compact and their centroids form a different grid, thus making the resulting decomposition schemes unacceptable. Decomposition of the hexagonal grid gives a sept tree representation. Although adjacent tiles do not merge into a precisely scaled up version of the individual tiles, the approximate (if border segments of the tiles are smoothed) shapes remain hexagonal. Thus, the above statement about the relationship between the decomposition method and the choice of sampling neighborhoods is generally true.

Given a grid and the corresponding tree representation, the most compact representation will be achieved for a certain given image shape. Triangular (square) images will be most compactly described by triangular (square) quad trees. Sept trees will be most compact for images with generally hexagonal shape having jagged borders. Images having other shapes will result in larger trees as additional nodes (depth) will be necessary to represent the fragmented, near-border regions. Thus, the overall efficiency (size) of a tree representation is determined by the grid topology as well as the image shape. Since square (or quadrangular) images on square grids are most common, the square quad trees would be used most often. For other combinations of grid type, image size, and image shape, the most efficient decomposition scheme would be different in each case.

ACKNOWLEDGMENTS

The support of the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract N00014-79-C-0424 is gratefully acknowledged, as is the help of Mary Foster in preparing this paper.

REFERENCES

1. A. Rosenfeld and A. C. Kak, *Digital Picture Processing* 2nd ed., Academic Press, New York, 1982.
2. A. Klinger and C. R. Dyer, Experiments in picture representation using regular decomposition, *Computer Graphics Image Processing* 5, 1976, 68-105.
3. S. L. Tanimoto and T. Pavlidis, A hierarchical data structure for picture processing, *Computer Graphics Image Processing* 4, 1975, 104-119.
4. O. Ore, *Graphs and Their Uses*, Random House, New York, 1963.

5. L. Fejes Toth, *Regular Figures*, Macmillan, New York, 1964.
6. H. Samet and A. Rosenfeld, Quad tree representations of binary images, *Proceedings 5th International Conference on Pattern Recognition*, pp. 815-818, Miami Beach, Fla., December 1-4, 1980.
7. P. J. Burt, Tree and pyramid structures for coding hexagonally sampled binary images, Computer Science TR 814, University of Maryland, October 1979.
8. C. F. R. Weiman and G. M. Chaikin, Logarithmic spiral grids for image processing, *IEEE Pattern Recognition and Image Processing Conference*, pp. 25-31, Chicago, Ill., August 6-8, 1979.
9. B. Grunbaum and G. C. Shepard, Tilings by regular polygons, *Math. Mag.* **50**, November 1977.
10. N. Ahuja and B. J. Schachter, *Pattern Models*, Wiley, New York, 1983.