Planar Decomposition for Quadtree Data Structure

PINAKI MAZUMDER

Computer Systems Group, Coordinated Science Laboratory, 1101 West Springfield Avenue, University of Illinois, Urbana, Illinois 61801

Received April 23, 1985; accepted June 2, 1986

The quadtree data structure is extensively used in representing 2-dimensional data in many applications like image processing, cartographic data processing. VLSI embedding, graphics, computer animation, computer-aided architecture, etc. The data structure employs the divide-and-conquer technique to recursively decompose the planar region. This paper addresses the problem of planar tessellation which yields the quadtree data structure. Arbitrary triangles and parallelograms have been used as basic cells and hyper-cellular structures corresponding to lower order k-gons have been shown to represent such data structures. Different tessellation schemes have been discussed using the notion of tessellation matrix. The performances of different tessellation schemes have been compared, introducing the concept of the neighborhood graph. © 1987 Academic Press, Inc.

1. INTRODUCTION

The recursive decomposition of a complex problem into simpler subproblems, known in the literature as the divide-and-conquer paradigm [1], is a well-practiced technique for solving many computational problems. Klinger and Dyer [2] and Tanimoto and Pavlidis [3] have successfully employed such techniques for 2-dimensional image representation. Their technique recursively decomposes the original image area of square shape into four identical squares at each step. The resulting data structure represents a 4-ary tree, more commonly called a *quadtree*, since each son of the tree represents one of the four quadrants of its father's square (Fig. 1). The root of the tree represents the complete picture and the leaf nodes of the tree represent the tessellated picture segments such that if they are recombined into groups as indicated by the internal nodes of the tree, the original image can be reconstructed. The advantages of such a hierarchical picture decomposition have been discussed in [2, 4]. Hierarchical decomposition enables addressing for rapid access to any geographical part of the image. It retains explicitly in the data structure a hierarchical description of picture patterns, elements, and their relationships. It permits recursive analysis of subpictures. Also its data structure distinguishes object from background and thereby can focus on the interesting subsets of the data. These merits associated with the quadtree representation have encouraged many researchers to design many useful algorithms for encoding, transforming, and searching quadtrees [2, 5], converting between quadtree and other data structures like chain codes [6], arrays [7, 8], and polygons [5], measuring region properties e.g., area and centroid [9], perimeter [8], directional symmetries [4], etc. An excellent literary review of these work have been presented by Samet [10] in his tutorial survey on the quadtree and related hierarchical data structure. Two limitations of quadtrees are that it needs large memory storage space and its representation of an object heavily depends on its location, orientation, and relative size. Normally the quadtree data structure needs a large memory space to maintain the pointers corresponding to the edges of the tree. Every nonterminal node needs five pointers

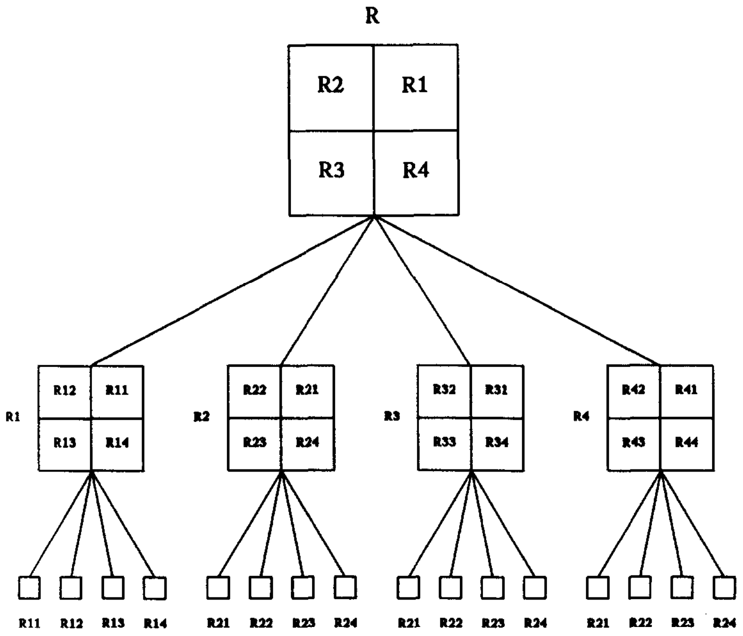

FIG. 1. Quadtree representation of planar decomposition.

to set up the links to the immediate predecessor and successors. But recently Gargantini [11], Tamminen [12], and Samet [13] have proposed methods of storing the quadtree efficiently in computer memory. Aggarwal and Chien [14] and Li [15] have proposed normalized quadtree representation which is invariant to rotation, translation, and object size change. A comprehensive bibliography of papers on quadtree can be found in the survey on picture processing published annually in this journal.

Quadtree data structures are also extensively used for many other 2-dimensional computational problems. Matsuyana [16], McKeown [17], Rosenfeld [18, 19], and others have employed the hierarchical data structures for cartographic data processing. The quadtree data structure was applied by Hunter and Woodwark for graphics and animation [5, 20, 21, 22], by Duda for scene analysis [23], and by Eastman for architectural design [24]. Leiserson and Leighton [25] used quadtree type hierarchical partitioning for constructing fault-tolerant 2-dimensional meshes on silicon wafer. Ahuja [26] has proposed an efficient planar embedding for VLSI implementation of quad and other trees. A VLSI pyramid machine for parallel image processing was designed by Dyer [27].

This wide spread application of hierarchical type data structures for 2-dimensional computational problems has prompted researchers to investigate the planar topologies which provide recursive decomposition. Ahuja [28] has done an extensive investigation on polygonal decomposition for hierarchical representation. He has proposed the equilateral triangle as an alternate geometry for quadtree decomposition and has demonstrated that the triangular quadtree has, on the average, the same computational complexity for image operations. Other researchers like Bell [29], Holroyd [30], etc., have discussed a number of tilings of the plane and presented a taxonomy of criteria to distinguish among the various tilings. The problem of planar tessellation has also been addressed by a number of mathematicians over the centuries. Toth [31], Shubnikov [32], Stevens [33], Grunbaum [34], Loeb [35], and Bourgoin [36] provide descriptions of different tilings and patterns. It is beyond the limits of this paper to make a reasonable review of all these past works.

On the other hand, the focus of this paper is primarily restricted to the planar tessellations which yield the quadtree type data structure. The reason for selecting the quadtree is that it provides the most cost-effective tree structure for hierarchical data representation. In an r-ary tree the expected cost of a query involving the leaf nodes is proportional to $(r + 1)/\log r$ which is a monotonically increasing function for $r > 3$. In this paper, it has been shown that any arbitrary parallelogram or triangle can be decomposed to yield a quadtree data structure. Also, it has been shown that a wide repertoire of planar tiles can be composed using these basic cells and any planar region of arbitrary dimensions having a shape identical to these classes of tiles can be decomposed recursively to represent a complete quadtree. A quadtree for which every nonleaf node has four successors is called (in this paper) a complete quadtree (similar to a complete binary tree [1]). The advantage of this hypercelluar decomposition is that it provides a more economical storage of quadtree in the computer memory. Since k (where k is a positive integer) cells are combined to form a hypercellular tile, $(k - 1)4^n$ memory space (where n is the height of the tree) is saved by decomposing the region into hypercellular tiles. Another justification for exploring hypercellular structures is that they are suitable for representing the grid images. Even though for normal image processing application, the square tessellation is highly desirable (because the regular polygon like a square pixel represents the point image most adequately), for applications in other areas like VLSI layout there is no binding reason to conform to a square geometry. In [37], Mazumder has identified a number of hypercellular processor geometries which can be hierarchically embedded to evolve parallel processing interconnection networks. The hierarchical embedding allows a logarithmic layout cost while the tessellation property of hypercellular structures provides an area efficient layout—two key aspects of VLSI design.

In this paper a number of tessellation schemes for different planar topologies have been shown using the concept of shape polymonials and tessellation matrices. The shape polynomial represents the geometry of the planar region while the tessellation matrix reveals the spatial adjacency of the regions. A comparison of the computational complexity for different types of tessellation schemes has been made using the notion of a neighborhood graph. The nodes in the neighborhood graph represent the tiles and the edges indicate the physical adjacency between the tiles. The edges are grouped into different cutsets which represent the costs associated with tree traversal between the nodes. Thus the analysis of a neighborhood graph provides an insight on the cost-effectiveness of the planar decomposition. The rest of the paper has been organized as follows. Section 2.1 establishes the formal framework of planar tessellation on quadrilateral lattices. Section 2.2 gives three tessellation algorithms for generating quadtree on quadrilateral lattices. Section 3 briefly presents the triangular lattices and four tessellation algorithms. Finally, Section 4 compares the computational complexity of the different quadtree topologies. The conclusion from this comparison is that the computational complexities of the different tessellation algorithms discussed in this paper are comparable.

2. QUADRILATERAL LATTICES AND QUADTREES

At first, a formal framework for the construction of tessellation algorithms has been made introducing the notion of the Euclidean planar covering by polyomino tiles [38]. The planar regions are represented by the tiles and the properties of polyomino tiles have been utilized to select the region geometry which can be recursively decomposed to yield the quadtree data structure. Tessellation algorithms have been designed by transforming tiles on the Euclidean plane by applying *linear transformations*. Linear transformations have linear as well as rotational components. The linear component is not included here and the tiles are juxtaposed to one another to align their outer edges parallel to $\{x, y\}$ basis. The rotational component has been defined as an operator and the tiles are restricted to orient along a finite direction as permitted by the constraint stated before. The tessellation algorithms have been succinctly represented as the spatial adjacency of tiled regions which can be recursively constructed employing the operators as postulated by the tessellation matrix.

2.1 Formal Framework of Planar Tessellation

Let $\mathbf{E}^2$ be a 2-dimensional Euclidean plane and $\{x, y\}$ be the basis on $\mathbf{E}^2$ such that the angle subtended by x and y at the point of intersection, called the origin, O, is equal to θ, where $0 < \theta < \pi$. Let $X = \{i | i = 0, 1, 2, \ldots\}$ be the set of points on the x axis such that the distance between any two neighboring points is equal to δx. Let $Y = \{j | j = 0, 1, 2, \ldots\}$ be the set of points on the y axis such that the distance between any two neighboring points is equal to δy. Let V denote the cartesian product of X and Y such that $V = X \times Y = \{(i, j) | i \in X \text{ and } j \in Y\}$.

The quadratic lattice graph G on $\mathbf{E}^2$ is defined by connecting all the ordered pairs of V such that the edges are parallel to either the x axis or y axis.

G is called an *orthogonal lattice* iff $\theta = \pi/2$. If $\delta x = \delta y$, cells in G are square in shape; otherwise, they are rectangular in shape.

G is called a *rhombic lattice* iff $\theta \neq \pi/2$. If $\delta x = \delta y$, cells in G are rhombic in shape; otherwise, they are rhomboid in shape.

The (i, j)th cell on $\mathbf{E}^2$ is the area bounded by the edges on G connecting the quad ordered pairs $\{(i, j), (i + 1, j), (i, j + 1), (i + 1, j + 1)\}$ and is denoted by

$$\text{cell}(i, j) = x^i y^j.$$

A cell(i, j), where $i > 0$ and $j > 0$, is called the *base* cell and the adjoining cells cell$(i - 1, j)$, cell$(i + 1, j)$, cell$(i, j - 1)$, and cell$(i, j + 1)$ are called the *4-neighbors* [39, 40] or the *von Neumann's neighbors* [41] of the *base* cell. These four cells are said to be *4-adjacent* to the *base* cell. The eight adjoining cells of the cell(i, j) (viz., cell$(i - 1, j)$, cell$(i + 1, j)$, cell$(i, j - 1)$, cell$(i, j + 1)$, cell$(i - 1, j - 1)$, cell$(i + 1, j - 1)$, cell$(i - 1, j + 1)$, and cell$(i + 1, j + 1)$) are called its *8-neighbors* [39, 40] or *Moore's neighbors* [41]. These eight cells are said to be *8-adjacent* to the *base* cell; cell$(i, j - 1)$ and cell$(i, j + 1)$ are horizontally adjacent to the *base* cell, cell$(i - 1, j)$ and cell$(i + 1, j)$ are vertically adjacent to the *base* cell, and cell$(i - 1, j - 1)$, cell$(i + 1, j - 1)$, cell$(i - 1, j + 1)$, and cell$(i + 1, j + 1)$ are diagonally adjacent to the *base* cell.

A *rookwise path* from cell(i, j) to cell(l, k) consists of a sequence of distinct contiguous cells

$$\text{cell}(i, j) = \text{cell}(i_0, j_0), \text{cell}(i_1, j_1), \dots, \text{cell}(i_n, j_n) = \text{cell}(l, k)$$

such that cell(i_k, j_k) is *4-adjacent* to cell(i_{k-1}, j_{k-1}), $1 \le k \le n$. If a *rookwise path* exists between two cells, they are said to be *rookwise connected.* It should be noted that all the pairs of *8-connected* cells do not generate a *rookwise path*. In this section, the regions which consist of cells which are rookwise connected are only considered. Rosenfeld [39, 42] and Kak [40] have discussed other criteria for adjacency and connectivity in digital pictures.

DEFINITION 1. *Tile.* A title, τ_k, on $\mathbf{E}^2$ consists of k rookwise [43] connected cells. If G is an orthogonal lattice, τ_k is called a k-omino and if G is rhombic lattice, τ_k is called k-diamond. If the 8-adjacent cells of the *base* cell belong to the tile and the *base* cell does not belong to the tile, then a *hole* occurs. Tiles with holes or with cells having only diagonally adjacent neighbors (i.e., cells which do not have a *rookwise path* between them) are not considered in this paper.

DEFINITION 2. *Shape polynomial of a tile.* The shape polynomial, $\mathrm{S}(\tau_k)$, of the tile, τ_k, whose lower left corner is on the origin, O, is given by

$$\mathrm{S}(\tau_k) = \sum_{(i,j)\in\mathbf{E}^2} \gamma_{ij} x^i y^j \tag{1}$$

where

$$\gamma_{ij} = \begin{cases} 1 & \text{iff for all } (x, y) \in \text{cell}(i, j) \to (x, y) \in \tau_k, \\ 0 & \text{otherwise.} \end{cases}$$

The area of τ_k is equal to $|\mathrm{S}(\tau_k)|$.

DEFINITION 3. *Conjugate of the shape polynomial.* The conjugate of the shape polynomial, $\mathrm{S}(\tau_k)$, is defined as a shape polynomial, $\mathrm{S}^*(\tau_k)$ obtained by permuting x and y elements [44, 45] of $\mathrm{S}(\tau_k)$ such that

$$\mathrm{S}^*(\tau_k) = \sum_{(i,j)\in\mathbf{E}^2} \mu_{ji} x^j y^i$$

where $\mu_{ji} = \gamma_{ij}$.

DEFINITION 4. *Operator.* An operator, Ψ_σ, rotates a tile, τ_k, by an angle Ψ about one of its edges parallel to the σ axis and the new shape polynomial of τ_k is given by $\Psi_\sigma \cdot \mathrm{S}(\tau_k)$.

Properties of Ψ_σ. (1) If $\Psi_{\sigma 1}$ and $\Psi_{\sigma 2}$ are two operators applied on τ_k, then

$$(\Psi_{\sigma 1}\Psi_{\sigma 2}) \cdot \mathrm{S}(\tau_k) = \Psi_{\sigma 1} \cdot (\Psi_{\sigma 2} \cdot \mathrm{S}(\tau_k)) = \Psi_{\sigma 2} \cdot (\Psi_{\sigma 1} \cdot \mathrm{S}(\tau_k)).$$

(2) If $\Psi = 2m\pi$ (where $m \in \{1, 2, \dots\}$), then $(2m\pi)_\sigma \cdot \mathrm{S}(\tau_k) = \mathrm{S}(\tau_k)$ and $\iota = (2m\pi)_\sigma$ is called the *identity operator.*

(3) If $\Psi = \phi$, then $\phi \cdot \mathrm{S}(\tau_k) = \phi$ and $\Psi = \phi$ is called the *null operator.*

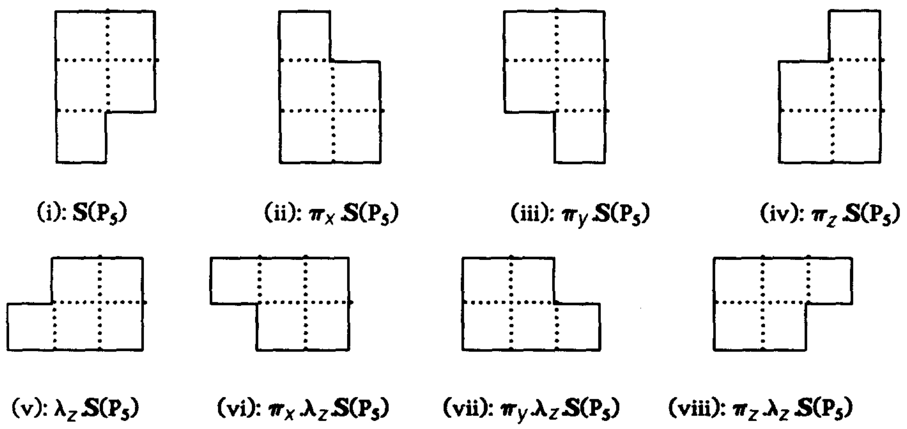

FIG. 2. Permissible orientation of P_5.

In order to decompose the regions to yield the quadtree data structure, the tiles should be oriented such that their edges are always parallel to the basis $\{x, y\}$. This requirement poses restrictions on the possible orientation of the tiles and only finite permissible orientations of the tiles are possible.

DEFINITION 5. *Permissible orientations.* The permissible orientations of a tile τ_k, on E^2 are due to the following operators applied in any arbitrary sequence: $(m\pi)_x$, $(m\pi)_y$, $(m\pi)_z$, and λ_z, where $\lambda_z \cdot S(\tau_k) = \pi_y \cdot S^*(\tau_k)$ and z is orthogonal to E^2. A cell(i, j) can be mapped to the cell$(m + i, n - j)$ by applying the operator π_x, to the cell$(m - i, n + j)$ by applying the operator π_y and to the cell$(m - i, n - j)$ by applying the operator π_z; m and n are two arbitrary integers which depend on the translation associated with the transformation. The transformations due to $(m\pi)_x$ and $(m\pi)_y$ are called *reflections* and transformations due to $(m\pi)_z$ and λ_z are called *rotations.*

THEOREM 1. *A τ_k has at most 8 distinct shape polynomials corresponding to all permissible orientations.*

Proof. Let $S(\tau_k)$ be the shape polynomial of τ_k. By applying the operators π_x, π_y, and π_z on $S(\tau_k)$, three more shape polynomials can be generated. By permuting the x and y elements, $S^*(\tau_k)$ can be derived and the above operators can be applied to get three more shape polynomials. Depending on the symmetry of τ_k about the x, y, or z axis, there will be 8, 4, 2, or 1 distinct shape polynomials. □

Figure 2 shows the shape polynomials of a tile, designated as P_5, corresponding to all its permissible orientations.

DEFINITION 6. *Tile notations.* (1) A tile is denoted by I_k iff, by any permissible orientation, its shape polynomial can be represented by

$$S(I_k) = \sum_{i=0}^{k-1} x^i.$$

Figure 3a shows I_2.

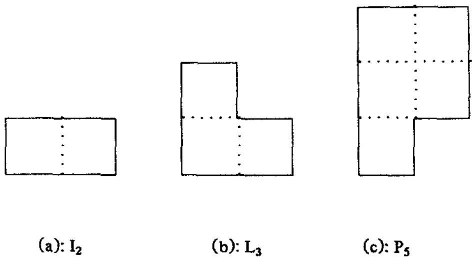

FIG. 3. Shape polynomial of some polyominoes.

(2) A tile is denoted by L_k iff, by any permissible orientation, its shape polynimial can be represented by

$$\mathbf{S}(L_k) = \sum_{i=0}^{m-1} x^i + \sum_{i=1}^{s} y^j \qquad \text{s.t. } m + s = k > 2.$$

Figure 3b shows L_3.

(3) A tile is denoted by P_k iff, by any permissible orientation, the shape polynomial can be represented by

$$\mathbf{S}(P_k) = \sum_{i=1}^{s} x^i \sum_{j=m}^{n-1} y^j + \sum_{j=0}^{m-1} y^j \qquad \text{s.t. } m + s(n-m) = k > 2.$$

Figure 3c shows P_5.

In general, if by any permissible orientation, the shape polynomial of a tile approximately represents an English letter in block capital, τ, then the tile is represented by τ_k, where k is the number of cells in the tile.

DEFINITION 7. *Planar tessellation.* The planar tessellation of a region $R = \cup_{i=1}^{n} R_i$ is its shape polynomial $\mathbf{S}$, represented as a spatial distribution of the shape polynomials $\mathbf{S}_i$'s (or the permissible orientations) of R_i's.

Assuming two of the edges of R constitute the basis $\{x, y\}$ such that the lower left corner of R is the origin, the spatial distributions of $\mathbf{S}_i$'s can be expressed as a matrix, such that for all R_i and R_j, if R_j is adjacent to R_i horizontally, vertically, or diagonally, $\mathbf{S}_i$ and $\mathbf{S}_j$ are also similarly adjacent in the matrix. The horizontal, vertical, or diagonal adjacency of the region R_j can be established with respect to R_i by computing the center of masses of all the regions. The center of mass of a region R_j having the shape polynomial $\mathbf{S}_j$ is given by the doublet $\{(1/|\mathbf{S}_j|)\Sigma_{(k,l)\in \mathbf{E}^2}\gamma_{kl}k, (1/|\mathbf{S}_j|)\Sigma_{(k,l)\in \mathbf{E}^2}\gamma_{kl}l\}$, where γ_{kl} is defined as in Eq. (1).

If $R = \cup_{i=1}^{9} R_i$ (Fig. 4) is a planar region similar to Moore's neighborhood structure, then

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_7 & \mathbf{S}_6 & \mathbf{S}_5 \\ \mathbf{S}_8 & \mathbf{S}_9 & \mathbf{S}_4 \\ \mathbf{S}_1 & \mathbf{S}_2 & \mathbf{S}_3 \end{pmatrix}. \tag{2}$$

Two planar regions, R_i and R_j, having shape polynomials $\mathbf{S}_i$ and $\mathbf{S}_j$ such that $|\mathbf{S}_i| = t|\mathbf{S}_j|$, where t or t^{-1} is a positive integer, are said to be *homothetic* [46] iff there

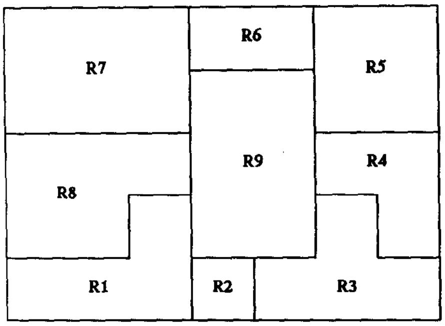

FIG. 4. Planar regions having Moore's neighborhood.

exists a sequence of permissible orientations which can be applied on R_j (or R_i) to make the two regions similar. Planar regions in Fig. 4 are not *homothetic* and can not be recursively decomposed to yield a quadtree structure. In order to generate the quadtree structure, planar regions whose shape polynomials are identical under a sequence of permissible orientations have been considered in this paper. Thus if $\mathbf{S}_i = \alpha_i \cdot \mathbf{S}_0$, where $\alpha_i \in \{(m\pi)_x, (m\pi)_y, (m\pi)_z, \lambda_z\}$, then

$$\mathbf{S} = \begin{pmatrix} \alpha_7 & \alpha_6 & \alpha_5 \\ \alpha_8 & \alpha_9 & \alpha_4 \\ \alpha_1 & \alpha_2 & \alpha_3 \end{pmatrix} \cdot \mathbf{S}_0 \tag{3}$$

and the matrix

$$\zeta = \begin{pmatrix} \alpha_7 & \alpha_6 & \alpha_5 \\ \alpha_8 & \alpha_9 & \alpha_4 \\ \alpha_1 & \alpha_2 & \alpha_3 \end{pmatrix} \tag{4}$$

is called the *tessellation matrix* of S.

Properties of ζ.

$$\zeta^n = \zeta \cdot \zeta^{(n-1)} = \begin{pmatrix} \alpha_7 \cdot \zeta^{(n-1)} & \alpha_6 \cdot \zeta^{(n-1)} & \alpha_5 \cdot \zeta^{(n-1)} \\ \alpha_8 \cdot \zeta^{(n-1)} & \alpha_9 \cdot \zeta^{(n-1)} & \alpha_4 \cdot \zeta^{(n-1)} \\ \alpha_1 \cdot \zeta^{(n-1)} & \alpha_2 \cdot \zeta^{(n-1)} & \alpha_3 \cdot \zeta^{(n-1)} \end{pmatrix}. \tag{5}$$

Operations on S. The operations on S (where S is given by Eq. 1) are:

(1) reflection about x axis

$$\pi_x \cdot \mathbf{S} = \begin{pmatrix} \pi_x \cdot \mathbf{S}_1 & \pi_x \cdot \mathbf{S}_2 & \pi_x \cdot \mathbf{S}_3 \\ \pi_x \cdot \mathbf{S}_8 & \pi_x \cdot \mathbf{S}_9 & \pi_x \cdot \mathbf{S}_4 \\ \pi_x \cdot \mathbf{S}_7 & \pi_x \cdot \mathbf{S}_6 & \pi_x \cdot \mathbf{S}_5 \end{pmatrix}; \tag{6}$$

(2) reflection about y axis

$$\pi_y \cdot \mathbf{S} = \begin{pmatrix} \pi_y \cdot \mathbf{S}_5 & \pi_y \cdot \mathbf{S}_6 & \pi_y \cdot \mathbf{S}_7 \\ \pi_y \cdot \mathbf{S}_4 & \pi_y \cdot \mathbf{S}_9 & \pi_y \cdot \mathbf{S}_8 \\ \pi_y \cdot \mathbf{S}_3 & \pi_y \cdot \mathbf{S}_2 & \pi_y \cdot \mathbf{S}_1 \end{pmatrix}; \tag{7}$$

(3) rotation about z axis

$$\pi_z \cdot \mathbf{S} = \begin{pmatrix} \pi_z \cdot \mathbf{S}_3 & \pi_s \cdot \mathbf{S}_2 & \pi_z \cdot \mathbf{S}_1 \\ \pi_z \cdot \mathbf{S}_4 & \pi_z \cdot \mathbf{S}_9 & \pi_z \cdot \mathbf{S}_8 \\ \pi_z \cdot \mathbf{S}_5 & \pi_z \cdot \mathbf{S}_6 & \pi_z \cdot \mathbf{S}_7 \end{pmatrix}; \tag{8}$$

(4) conjugate operation

$$\lambda_z \cdot \mathbf{S} = \begin{pmatrix} \lambda_z \cdot \mathbf{S}_5 & \lambda_z \cdot \mathbf{S}_4 & \lambda_z \cdot \mathbf{S}_3 \\ \lambda_z \cdot \mathbf{S}_6 & \lambda_z \cdot \mathbf{S}_9 & \lambda_z \cdot \mathbf{S}_2 \\ \lambda_z \cdot \mathbf{S}_7 & \lambda_z \cdot \mathbf{S}_8 & \lambda_z \cdot \mathbf{S}_1 \end{pmatrix}. \tag{9}$$

EXAMPLE. Let $\alpha_7 = \pi_x$, $\alpha_6 = \phi$, $\alpha_5 = \phi$, $\alpha_8 = \phi$, $\alpha_9 = \iota$, $\alpha_4 = \phi$, $\alpha_1 = \iota$, $\alpha_2 = \phi$, and $\alpha_3 = \pi_y$ such that the *tessellation matrix* can be represented by

$$\zeta = \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix}$$

Then from Eqs. (5), (6), (7), and (8), it can be shown that

$$\zeta^2 = \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} \cdot \zeta$$

$$\zeta^2 = \begin{bmatrix} \pi_x \cdot \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} & \phi \cdot \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} & \phi \cdot \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} \\ \phi \cdot \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} & \iota \cdot \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} & \phi \cdot \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} \\ \iota \cdot \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} & \phi \cdot \begin{pmatrix} \pi_z & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} & \pi_y \cdot \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} \end{bmatrix}$$

$$\zeta^2 = \begin{bmatrix} \begin{pmatrix} \pi_x & \phi & \pi_z \\ \phi & \pi_x & \phi \\ \iota & \phi & \phi \end{pmatrix} & \begin{pmatrix} \phi & \phi & \phi \\ \phi & \phi & \phi \\ \phi & \phi & \phi \end{pmatrix} & \begin{pmatrix} \phi & \phi & \phi \\ \phi & \phi & \phi \\ \phi & \phi & \phi \end{pmatrix} \\ \begin{pmatrix} \phi & \phi & \phi \\ \phi & \phi & \phi \\ \phi & \phi & \phi \end{pmatrix} & \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} & \begin{pmatrix} \phi & \phi & \phi \\ \phi & \phi & \phi \\ \phi & \phi & \phi \end{pmatrix} \\ \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix} & \begin{pmatrix} \phi & \phi & \phi \\ \phi & \phi & \phi \\ \phi & \phi & \phi \end{pmatrix} & \begin{pmatrix} \phi & \phi & \pi_z \\ \phi & \pi_y & \phi \\ \iota & \phi & \pi_y \end{pmatrix} \end{bmatrix}$$

If $\mathbf{S}_0 = \mathbf{S}(L_3)$, then $\zeta^2 \cdot \mathbf{S}_0$ results into the tessellation corresponds to Fig. 5.

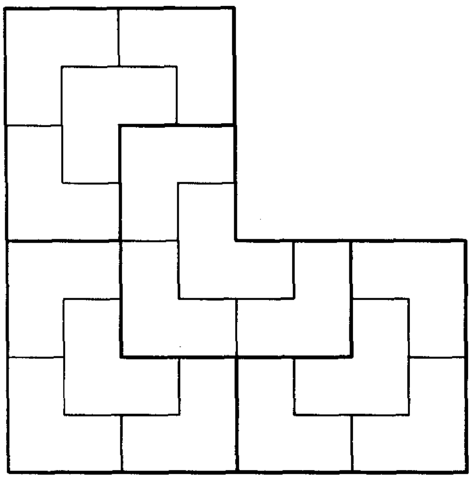

FIG. 5. Planar region of shape L_3.

2.2 *Tessellation Algorithms*

THEOREM 2. *The planar region* $R1$, *having shape polynomial* $S1$, *can be represented by a complete quadtree of height n if* $S1 = \zeta_1^n S(I_k)$, *such that* $|S1| = 4^n|S(I_k)|$, *where*

$$\zeta_1 = \begin{pmatrix} \iota & \iota \\ \iota & \iota \end{pmatrix}$$

and $k = 1, 2, 3, \ldots$.

Proof. Refer to Fig. 1. Since the *tessellation matrix* is symmetrical and the tile shape is 1-dimensional, by using induction the result follows. □

THEOREM 3. *The planar region* $R2$, *haivng shape polynomial* $S2$, *can be represented by a complete quadtree of height n if* $S2 = \zeta_2^n S(P_{3k}) = S^n(P_{3k})$, *such that* $|S2| = 4^n|S(P_{3k})|$, *where*

$$\zeta_2 = \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \iota & \phi \\ \iota & \phi & \pi_y \end{pmatrix}$$

and $k = 1, 2, 3, \ldots$.

Alternately, Theorem 3 can be written as

THEOREM 3′.

$$S2 = S^n(P_{3k}) = \begin{pmatrix} \pi_x \cdot S^{-1}(P_{3k}) & \phi & \phi \\ \phi & S^{n-1}(P_{3k}) & \phi \\ S^{n-1}(P_{3k}) & \phi & \pi_y \cdot S^{n-1}(P_{3k}) \end{pmatrix}$$

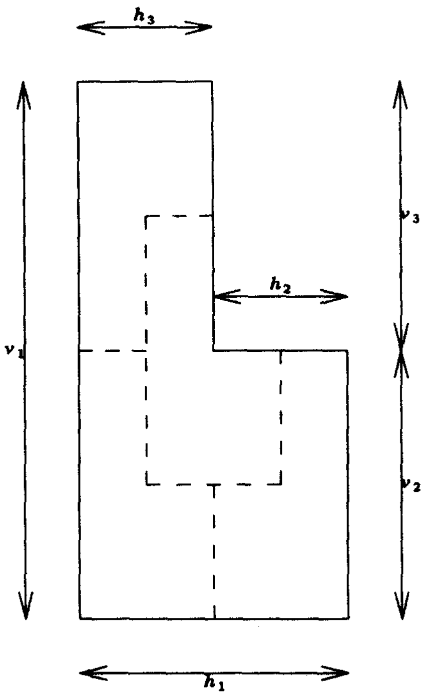

FIG. 6. Planar region of shape P_{3k}.

where, $n > 1$ is called the level of tessellation *and* $S^1(P_{3k}) = S(P_{3k})$ *is the shape of polynomial of* P_{3k}.

Proof. Let $R2$ be an arbitrary region of shape P and its dimensions are as shown in Fig. 6. By using ζ_2, $R2$ can be tessellated into exactly 4 regions having an identical shape to $R2$ if $v2 = v3 = (1/2)v1$ and $h2 = h3 = (1/2)h1$. The area of $R2$ is equal to $(3/4)h1v1$. Thus $R2$ can be represented as a complete quadtree whose leaves have shape polynomials $S(P_{3k})$ iff $v1/h1 = m$ and $h1 = 2p$, where m and p are positive integers. □

THEOREM 4. *The planar region $R3$, haivng shape polynomial* S3, *can be represented by a complete quadtree of height n if* $S3 = \zeta_3^n S(P_{5k}) = S^n(P_{5k})$, *such that* $|S3| = 4^n|S(P_{5k})|$, *where*

$$\zeta_3 = \begin{pmatrix} \pi_x & \phi & \phi \\ \phi & \pi_z \cdot \lambda_z & \phi \\ \iota & \phi & \pi_y \end{pmatrix}$$

and $k = 1, 2, 3, \ldots$.

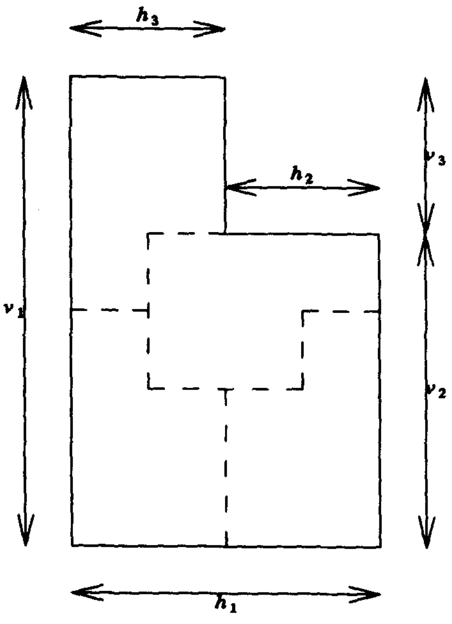

FIG. 7. Planar region of shape P_{5k}.

Alternately, Theorem 4 can be written as

THEOREM 4′.

$$S3 = S^n(P_{5k}) = \begin{pmatrix} \pi_x \cdot S^{n-1}(P_{5k}) & \phi & \phi \\ \phi & \pi_z \cdot \lambda_z \cdot S^{n-1}(P_{5k}) & \phi \\ S^{n-1}(P_{5k}) & \phi & \pi_y \cdot S^{n-1}(P_{5k}) \end{pmatrix}$$

where, $n > 1$ is called the level of tessellation *and* $S^1(P_{5k}) = S(P_{5k})$ *is the shape polynomial of* P_{5k}.

Proof. Refer to Fig. 7. By using ζ_3, $R3$ can be tessellated into exactly 4 regions having an identical shape to $R3$ if $v2 = h1$ and $v3 = h2 = (1/2)h1$.

The area of $R3$ is equal to $(5/4)h1^2$. Thus $R3$ can be represented as a complete quadtree whose leaves have the shape polynomial $S(P_{5k})$ iff $v1/h1 = 3m/2$ and $h1 = 2p$ where m and p are positive integers. □

3. TRIANGULAR LATTICES AND QUADTREES

Triangular lattice graph, G', can be obtained by joining the ordered pairs $\{(i, j), (j, i)\}$ of G on E^2. Let θ_x and θ_y be the angles subtended by $\langle (i, j), (j, i) \rangle$ edge on lines parallel to the x and y axes respectively such that $\theta + \theta_x + \theta_y = \pi$. Let ξ denote the set of all such edges in G'. Thus each cell of G is divided into two triangles about an edge, $e \in \xi$. Each triangle on E^2 is called a cell of G'. Let

TABLE 1
Tessellation Scheme over Triangular Lattices

Region name	Tile name	Shape polynomial	Tessellation matrix
$R'1$	1-trigon	1^+	$\zeta_1' = \begin{pmatrix} \phi & \iota & \phi \\ \phi & \pi_x & \phi \\ \iota & \phi & \iota \end{pmatrix}$
$R'2$	2-trigon	1	$\zeta_2' = \begin{pmatrix} \iota & \iota \\ \iota & \iota \end{pmatrix}$
$R'3$	3-trigon	$1 + x^+$	$\zeta_3' = \pi_x \cdot \begin{pmatrix} \phi & \iota & \phi \\ \pi_\zeta & \phi & \pi_y \\ \phi & \pi_x & \phi \end{pmatrix}$
$R'4$	6-trigon	$1 + x + x^{2+} + y^+$	$\zeta_4' = \pi_x \cdot \begin{pmatrix} \pi_z & \phi & \pi_z \\ \phi & \iota & \phi \\ \pi_\zeta & \phi & \phi \end{pmatrix}$

cell($i+, j+$) and cell($i-, j-$) denote the lower and upper triangles on G', corresponding to cell(i, j) on G. Obviously, cell($i-, j-$) is the *reflection* of cell($i+, j+$) about $e \in \xi$. Cells of G' are represented as cell($i+, j+$) $= x^{i+}y^{j+}$ and cell($i-, j-$) $= x^{i-}y^{j-}$ such that the *base* cell is $x^iy^j = x^{i+}y^{j+} + x^{i-}y^{j-}$. A tile, τ_k', on G', called a k-trigon, consists of k connected cells such that there is no *hole* within the tile. It should be noted that a *hole* occurs if at least one of the trigons of the *base* cell is absent. The quadtree representability of the regions whose shape polynomial is identical in shape to these tiles and the corresponding tessellation matrices are shown in Table 1.

4. COMPARISON OF QUADTREE TOPOLOGIES

The quadtree shown in Fig. 1 represents only the hierarchical organization of the leaf nodes and does not explicitly indicate the spatial neighborhood of the leaf nodes. An alternate representation of the quadtree for 2-dimensional data structures can be made by the neighborhood graph, $G^*(V, E)$. The nodes, $v \in V$, of the graph represent the tessellated regions corresponding to the leaf nodes of the quadtree and the edges, $e \in E$, of the graph represent the spatial adjacency (corresponding to 4-*adjacency*) of the nodes. The neighborhood graphs for regions $R1$ (as in Theorem 2), $R2$ (as in Theorem 3) and $R'1$ (as in first row of Table 1) corresponding to the tessellation matrices ζ_1, ζ_2, and ζ_1', respectively, are shown in Fig. 8. Figure 8a represents the neighborhood graph corresponding to the well known $\{4,4\}$ tessellation. Figure 8c represents the neighborhood graph for triangular quadtree having $\{3,6\}$ tessellation. The pair $\{k, v\}$ is called the Schlafli symbol [47] where k denotes the number of sides in a polygonal cell and v denotes the number of cells meeting at a vertex. It is interesting to note that the neighborhood graph of $\{3,6\}$ tessellation represents the $\{6,3\}$ tessellation of regular hexagon.

In the neighborhood graph, the dotted lines (called the *equi-level lines*) are the cuts corresponding to the different level of tessellations and indicate the hierarchical distribution of the nodes. All the edges cut by the ith level tessellations are called ith cut-set of G^*. Let QUERY (v_i, v_j) be a diadic operation on G^* involving $v_i, v_j \in V$. The cost of the operation, $c(\text{QUERY}(v_i, v_j))$, is equal to the total path

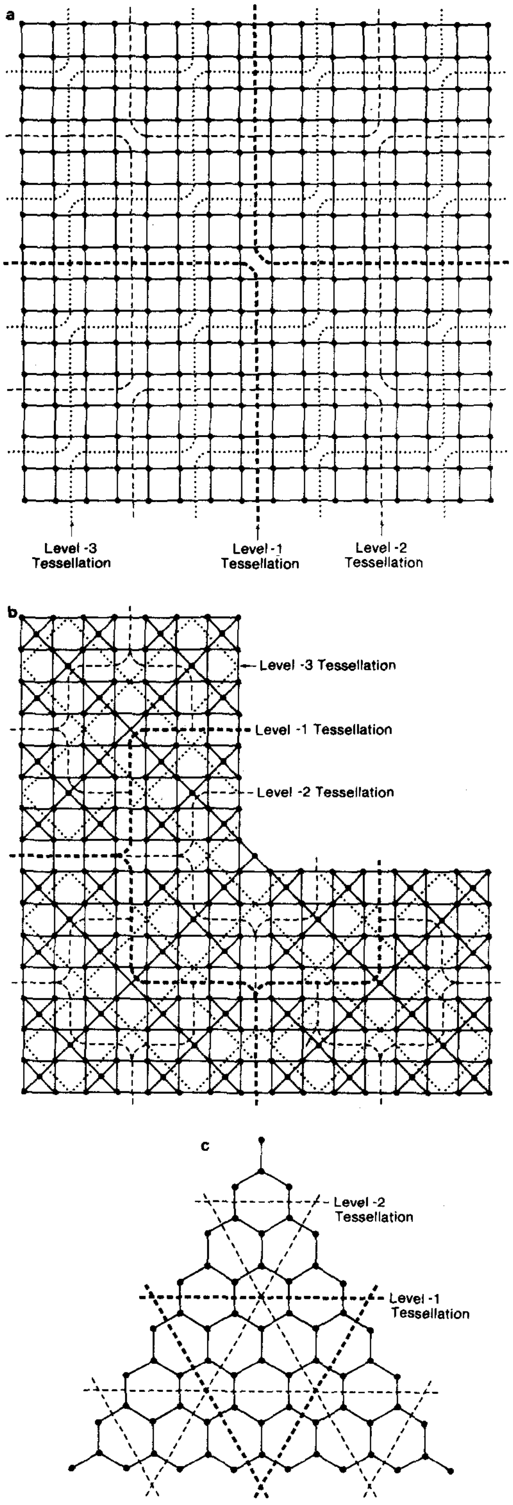

FIG. 8. (a) Graph for $R1$. (b) Graph for $R2$. (c) Graph for $R'1$.

length between the nodes, v_i and v_j in the quadtree of height n. Obviously, an ith level equi-level line describes an *equi-cost line* of cost $2(n+1-i)$ and $c(\text{QUERY}(v_i, v_j))$ is equal to $2(n+1-m)$, where $1 \leq m \leq n$ is the lowest value of the equi-level line intersected by the line joining v_i and v_j. Let f_{2k} denote the total number of nodes corresponding to all the edges in the $(n+1-k)$-th cut-set of G^* and $p_{2k} = f_{2k}/4^n$ denote the probability that a randomly chosen node will be connected by such an edge. Let $v_j \in V$ be any randomly chosen neighbor of $v_i \in V$. The probability of the event $c(\text{QUERY}(v_i, v_j)) = 2k$, denoted by $\mathbf{P}_{2k}$, is a performance metric of the quadtree corresponding to its tessellation matrix, ζ. If p_2 is the probability that v_i and v_j are siblings (i.e., they have a common father) and q_{2k} is the conditional probability that the edge $\langle v_i, v_j \rangle$ is in $(n+1-k)$th cut-set of G^* when for some $v \in V$, $\langle v, v_i \rangle$ is in $(n+1-k)$th cut-set, then

$$\mathbf{P}_{2k} = P_2\delta_{1k} + P_{2k}q_{2k}(1-\delta_{1k})$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

δ_{ij} is called the *Kronecker delta function.*

The values of $\mathbf{P}_{2k}$ for different neighborhood graphs of Fig. 8 are:

(1) Graph for $R1$ (Fig. 8a) corresponding to tessellation matrix ζ_1: Since each node has two siblings, $p_1 = \frac{1}{2}$ for non-peripheral nodes in the graph. Also, $p_{2k} = (4^{n+1-k}2^k)/4^n = 1/2^{k-2}$, where $k > 2$, and $q_{2k} = ((\frac{1}{4})*(2^k - 1) + (\frac{1}{2}))/2^k = \frac{1}{4} + 1/2^{k+2}$ so that

$$\mathbf{P}_{2k} = \left(\tfrac{1}{2}\right)\delta_{1k} + \left((1+2^k)/2^{2k}\right)(1-\delta_{1k}).$$

(2) Graph for $R2$ (Fig. 8b) corresponding to tessellation matrix ζ_2: The non-peripheral nodes have degrees 4, 5, or 6 depending on their locations in the graph. The average value of p_2 is approximately equal to $\frac{1}{2}$ and $p_{2k} = \sqrt{3}/2^k$. The exact value of q_{2k} is difficult to compute but the average value of q_{2k} lies between $\frac{1}{5}$ and $\frac{1}{2}$. Thus

$$\mathbf{P}_{2k} \approx \left(\tfrac{1}{2}\right)\delta_{1k} + \left(\sqrt{3}\,q_{2k}/2^k\right)(1-\delta_{1k}).$$

(3) Graph for $R'1$ (Fig. 8c) corresponding to tessellation matrix ζ_1: Each non-peripheral node has either one or three siblings and $p_2 = \frac{1}{4}(1*1 + 3*1/3) = \frac{1}{2}$. Also, $p_{2k} = (4^{n-k}(4.5*2^k))/4^n = 4.5/2^k$ and $q_{2k} = ((\frac{2}{3})*3 + (\frac{1}{3})(4.5*2^k - 3))/(4.5*2^k) = (1 + 1.5*2^k)/(4.5*2^k)$ so that

$$\mathbf{P}_{2k} = \left(\tfrac{1}{2}\right)\delta_{1k} + \left((1+1.5*2^k)/2^{2k}\right)(1-\delta_{1k}).$$

From the above results, it is evident that $\mathbf{P}_2$, corresponding to $k = 1$, is approximately the same on the average for different planar topologies.

5. CONCLUSIONS

From the discussion in this paper, it can be concluded that a wide repertoire of planar geometries can be represented as quadtrees. The regions can be tessellated recursively using simple algorithms postulated by a tessellation matrix. The cost for diadic operation, which is equal to the path traversal necessary in the tree, is comparable for different planar geometries.

This paper is restricted to only hyper-cellular structures corresponding to k-gon, where $k \leq 6$. But the same concept can be easily employed for higher order k-gons.

The notion of planar tessellation by polyomino tiles can be extended to 3-dimensional space to identify the 3-dimensional hypercellular topologies that can be utilized to generate the oct-tree [48]. The oct-tree data structure is highly suitable for representing the 3-dimensional objects.

The concept of recursive tessellation of planar regions as discussed in this paper has been utilized to design cost-effective embedding algorithms for cellular arrays in VLSI design [37]. Regular geometric structures represented by polyominoes can be utilized to select the basic cell geometry corresponding to the requisite communication structure and simple embedding algorithms can be designed utilizing the notions described in the paper. Embedding algorithm similar to theorem 3 has been shown to construct fault-tolerant 2-dimensional meshes.

ACKNOWLEDGMENTS

This research was partly supported by the SRC under the Contract 84-06-049. The author wishes to thank Professors Janak Patel, Narendra Ahuja, and finally, the reviewers for their criticism and suggestions which improved the readability and content of this paper.

REFERENCES

1. A. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison–Wesley, Reading, Mass., 1974.
2. A Klinger and C. R. Dyer, Experiments in picture representation using regular decomposition, *Comput. Graphics Image Process.* **5**, 1976, 68–105.
3. S. L. Tanimoto and T. Pavlidis, A hierarchical data structure for picture processing, *Comput. Graphics Image Process.* **4**, 1975, 104–119.
4. N. Alexandridis and A. Klinger, Picture decomposition, tree data-structures, and identifying directional symmetries and node combinations, *Comput. Graphics Image Process.* **8**, 1978, 43–47.
5. G. M. Hunter and K. Steiglitz, Linear transformation of pictures represented by quadtrees, *Comput. Graphics Image Process.* **5**, 1976, 68–105.
6. C. R. Dyer, A. Rosenfeld, and H. Samet, Region representation: Boundary codes from quadtrees, *Commun. ACM* **23**, 1980, 171–179.
7. H. Samet, Region representation: quadtrees from binary arrays, *Comput. Graphics Image Process.* **13**, 1980, 88–93.
8. H. Samet, An algorithm for converting rasters to quadtrees, *IEEE Trans. Pattern Anal. Mach. Intell.* **3**, 1981, 93–95.
9. M. Shneier, Calculations fo geometric properties using quadtrees, *Comput. Graphics Image Process.* **16**, 3, 1981, 296–302.
10. H. Samet, The quadtree and related hierarchical data structures, *ACM Comput. Surv.* **16**, 1984, 187–260.
11. I. Gargantini, An effective way to represent quadtrees, *Commun. ACM* **25**, 1982, 905–910.
12. M. Tamminen, Comment on quad-and octtrees, *Commun. ACM* **27**, 1984, 248–249.
13. H. Samet, Data structures for quadtree approximation and compression, *Commun. ACM* **28**, 1985, 973–993.

14. C. H. Chien and J. K. Aggarwal, A normalized quadtree representation, *Comput. Vision Graphics Image Process.* **26**, 1984, 331–346.
15. M. Li, W. I. Gorsky, and R. Jain, Normalized quadtrees with respect to translations, in *Proceedings*, PRIP-81, Dallas, Texas, 1981, pp. 60–62.
16. T. Matsuyama, L. V. Hao, and M. Nagao, A file organization for geographic information systems based on spatial proximity, *Comput. Vision Graphics Image Process.* **26**, 1984, 303–318.
17. D. M. McKeown and J. L. Denlinger, Map-guided feature extraction from aerial imagery, in *Proceedings, Workshop on Computer Vision*, New York, April 1984, pp. 205–213.
18. A. Rosenfeld, H. Samet, C. Shaffer and R. E. Webber, *Application of Hierarchical Data Structures for Geographical Information Systems*, TR-1197, Computer Science Department, University of Maryland, June, 1982.
19. A. Rosenfeld, H. Samet, C. Shaffer, and R. E. Webber, *Application of Hierarchical Data Structures for Geographical Information Systems Phase II*, TR-1327, Computer Science Department, University of Maryland, September 1983.
20. G. M. Hunter, Computer animation survey, *Comput. Graphics* **2**, 1977, 225–229.
21. G. M. Hunter, *Efficient Computation and Data Structures for Graphics*, Ph.D. thesis, 1979.
22. J. R. Woodwark, The explicit quadtree as a structure for computer graphics, *Comput. J.* **25**, No. 2, 1982, 235–238.
23. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
24. C. M. Eastman, Representations for space planning, *Commun. ACM* **13**, 1970, 243–250.
25. F. T. Leighton and C. E. Leiserson, Wafer-scale integration for systolic arrays, in *Proceedings, 23rd Ann. IEEE Sympos. Found. of Comput. Sci.*, 1982, pp. 297–311.
26. N. Ahuja, Efficient planar embedding of trees for VLSI layout, private communication, October 1985.
27. C. R. Dyer, A VLSI pyramid machine for parallel image processing, in *Proceedings*, PRIP-81, Dallas, Texas, 1981, 381–386.
28. N. Ahuja, On approaches to polygonal decomposition for hierarchical image representation, *Comput. Vision Graphics Image Process.* **24**, 1983, 200–214.
29. S. B. M. Bell, B. M. Diaz, F. Holroyd, and M. J. Jackson, Spatially referenced methods of processing raster and vector data, *Image Vision Comput.* **4**, 1983, 211–220.
30. F. C. Holroyd, The geometry of tiling heirarchies, *Ars Combin.* **16-B**, 1983, 211–240.
31. F. Toth, *Regular Figures*, Macmillan, New York, 1964.
32. A. V. Shubnikov and N. V. Belov, *Colored Symmetry*, Pergamon, Oxford, 1964.
33. P. S. Stevens, *Handbook of Regular Patterns*, MIT Press, Cambridge, Mass., 1980.
34. B. Grunbaum and G. C. Shepard, *Tilings and Patterns*, Freeman, San Francisco, 1981.
35. A. L. Loeb, *Color and Symmetry* Krieger, Huntington, N.Y., 1978.
36. J. Bourgoin, *Arabic Geometrical Patterns and Design*, Dover, New York, 1973.
37. P. Mazumder, *Networks and Embedding Aspects of Cellular Structures for on-chip Parallel Processing*. M.Sc. thesis, Department of Computer Science, University of Alberta, October 1984.
38. S. W. Golomb, Tiling with polyominoes, *J. Combin. Theory.* **2**, 1966, 280–296.
39. A. Rosenfeld, Adjacency in digital pictures, *Inform. Control* **26**, 1974, 24–33.
40. A. Rosenfeld and A. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.
41. B. Hayes, Computer recreations, *Sci. Amer.* **250**, 1984, 12–21.
42. A. Rosenfeld, Connectivity in digital pictures, *J. Assoc. Comput. Mach.* **17**, 1970, 146–160.
43. S. W. Golomb, *Polyominoes*, Scribner, New York, 1965.
44. F. W. Barnes, Algebraic theory of brick packing 1, *Discrete Math.* **42**, 1982, 7–26.
45. F. W. Barnes, Algebraic theory of brick packing 2, *Discrete Math.* **42**, 1982, 129–144.
46. H. S. M. Coxeter, *Introduction to Geometry*, Wiley, New York, 1980.
47. H. S. M. Coxeter, *Regular Polytopes*, Dover, New York, 1973.
48. C. L. Jackins and S. L. Tanimoto, Oct-trees and their use in representing three-dimensional objects, *Comput. Graphics Image Process.* **14**, 1980, 249–270.