JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 9, No. 4, December, pp. 271–286, 1998
ARTICLE NO. VC980401

Recursive Representation and Progressive Display of Binary Objects for Efficient Network Browsing

I-Pin Chen and Zen Chen

Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China

Received December 18, 1997; accepted September 23, 1998

When binary objects are browsed in a network environment, data transmission rate, progressive display capability, and view modification under rotation, scaling, and/or translation (R/S/T) are the major factors for selection of an appropriate representation model of binary objects. A new half-plane-based representation and display method for 2D binary objects is proposed. Within this modeling framework, a binary object approximated by a shape of a polygon can be represented as a collection of half-planes defined over the edges of the polygon under operations of union and intersection. The basic shape attributes of the object model are the slope and the *y*-intercept of the boundary line of the constituent half planes. In the progressive display of the binary object our method adopts the quadtree block subdivision to divide the object image into hierarchical levels of detail (or resolution). Our method determines the color of a quadtree node based on the (angle, intercept) representation parameters. It is shown that the representation parameters at the parent node are recursively related to those at the child nodes. This recursive relation is crucial for deriving the color of the nodes for progressive object display. Lemmas for the node color determination for an object expressed in the form of half-planes, a convex polygon, or a concave polygon are derived step by step. Our method is generally better than many existing methods in terms of data transmission rate, progressive display capability, and view modification under R/S/T variations. Simulation results are provided to illustrate the performance of our method. © 1998 Academic Press

1. INTRODUCTION

Binary objects are used in shape-oriented computer applications such as geographic information systems, business trademark registration, mechanical CAD systems, and cartoon character design. With Advances in Internet/intranet technology, many computer applications are shifting from personal or local use to network or global use. When binary objects are browsed in a network environment, data transmission rate, display capability in a progressive mode, and view modification under rotation, scaling, or translation (R/S/T) are the major concerns for selection of an appropriate representation model of binary objects. In the existing facsimile system, a run-length representation of the scanned data is used. This system has a nice data compression capability, but it cannot be used for progressive display. Also, it cannot generate a new view to reflect any R/S/T change. As far as progressive display capability is concerned, two types of object representation methods are commonly used: quadtree representation [1, 2, 3] and transform coding [18]. These methods are different in their data coding and progressive display mechanisms. The quadtree method subdivides the image of the binary objects into successive levels of four blocks (or regions) until all pixels in the block have the same color (black or white) or a specified level of resolution is reached. The transform coding methods apply a set of basis images, each with a different block structure ranging from a low spatial frequency to a high spatial frequency, to model the object. The progressive display is achieved through successive inverse transformations using an increasing number of transform coefficients. Figure 1 shows the results of progressive display of a 1-D binary object by the quadtree representation method and the Walsh transform method. In the quadtree representation method, the shape of a block with a uniform color (black or white) remains the same at the successive levels of resolution while the block shape and color (i.e., gray level) in the Walsh transform method vary drastically at the first few levels of resolution and gradually settle down. Therefore, the Walsh transform coding method is not very effective for object browsing. There are some newly developed progressive transmission methods including JPEG [19], and Wavelet [20]. The JPEG progressive transmission mode and the wavelet representation both use sets of transform coefficients of variable size for the coarse-to-fine representation. Their progressive nature is similar to that of the Walsh transform coding method mentioned above.

Furthermore, when binary objects are browsed in a large database, it is often necessary to make a view change with a specified R/S/T effect. The data model of the quadtree representation method is generally sensitive to these view

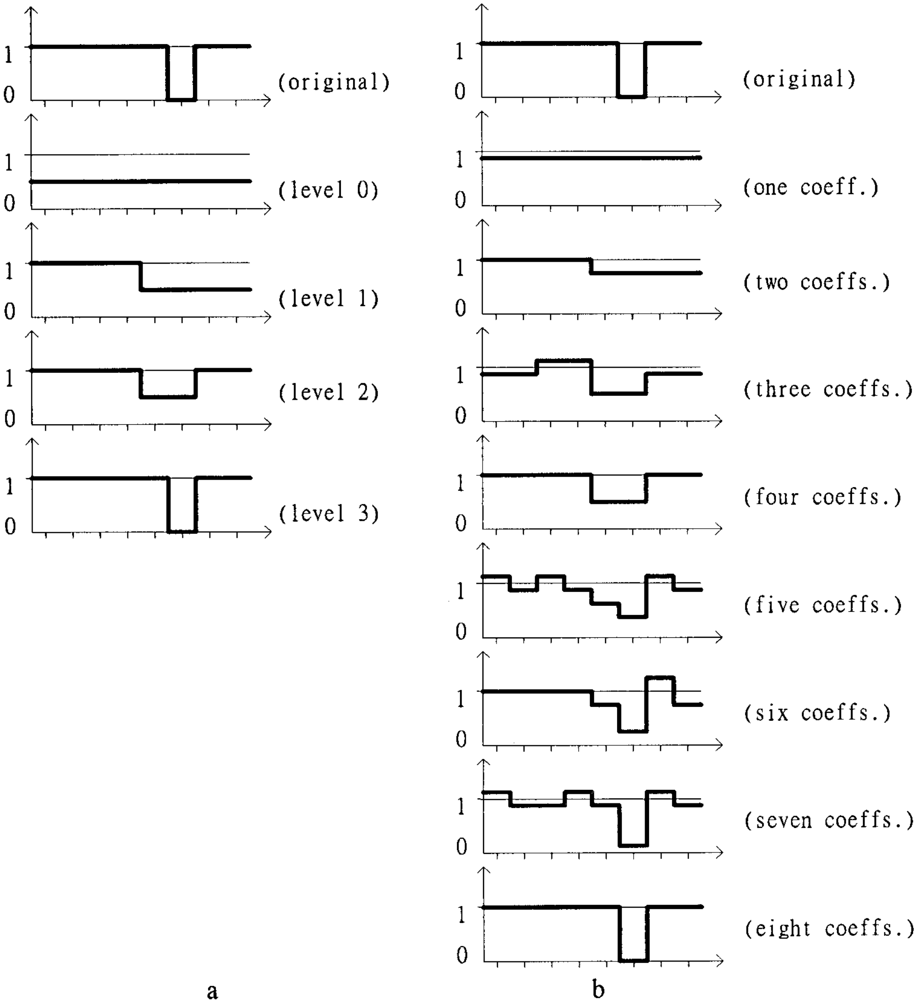


FIG. 1. A 1-D binary object is progressively displayed: (a) the quadtree representation scheme; (b) the Walsh transform coding.

changes and the method yields a larger display error, so it is not suitable for these view change operations.

We shall propose a new method for modeling and progressively displaying the binary objects. The method will meet the above requirements related to data transmission rate, progressive display, and view change under R/S/T variation. It adopts the quadtree block subdivision scheme to divide the image into coarse-to-fine levels of blocks, but it uses a half-plane-based representation model. The quadtree variants (e.g., line quadtree [4], PM quadtree [5], edge quadtree [6], and template quadtree [7]) may refer to the line or edge features of the object, but they still use the region-based representation model and, therefore, face the same problems as the standard quadtree method. Our method represents a binary object as a collection of convex or concave polygons. These polygons are further expressed in the union and/or intersection form for their constituent half planes. The angle and intercept (i.e., ordinates) of each half plane are the two attributes of our basic data elements.

As will be seen, efficient data transmission should present no problem to our method. View change under R/S/T variations can be also handled gracefully. The remaining issue is the progressive display capability of the new method. It will be shown that the (angle, intercept) parameters for the parent node are recursively related to those for the child nodes. This recursive relation is crucial for deriving the colors of the nodes for progressive display. Lemmas for the node color determination are derived for objects expressed in the form of half planes, convex polygon, and concave polygon step by step. Table 1 summarizes the comparison between the existing object modeling and displaying methods and our method.

Samet and Taminen [10] presented a bintree representation of a binary object expressed in a CSG tree and applied their technique to the interference detection problem. They gave four rules to decide the color (black, white, or gray) of each node in the bintree. Their method is found inadequate for progressive display use. Consider the binary

TABLE 1
A Comparison between the Existing Methods and Our Method

	Method			
Factor	Facsimile system	Transform coding method	Quadtree representation method	Our method
Data representation	run length	transform coefficients	multilevel-region-based representation	half-plane-based representation
Average data transmission time	medium	fast	slow	fast
Progressive display capability	no	yes, but nonsteady[a]	yes, steady	yes, steady
Data model modification under the R/S/T variations	hard	easy	sensitive[b]	easy

[a] See the text.

[b] Unless the rotation angle is a multiple of 90° or the scale factor is an integer.

object in Fig. 2a. Figures 2b–2d show the correct quadtree display results of the object for the first few resolution levels. However, the color of nodes derived from the Samet and Taminen's method for the corresponding levels of resolution are shown in Figs. 2e–2g. Here, the bintree display is converted to an equivalent quadtree display. The erroneous gray nodes produced by their method are due to the inadequacy of the four rules for node color determi-

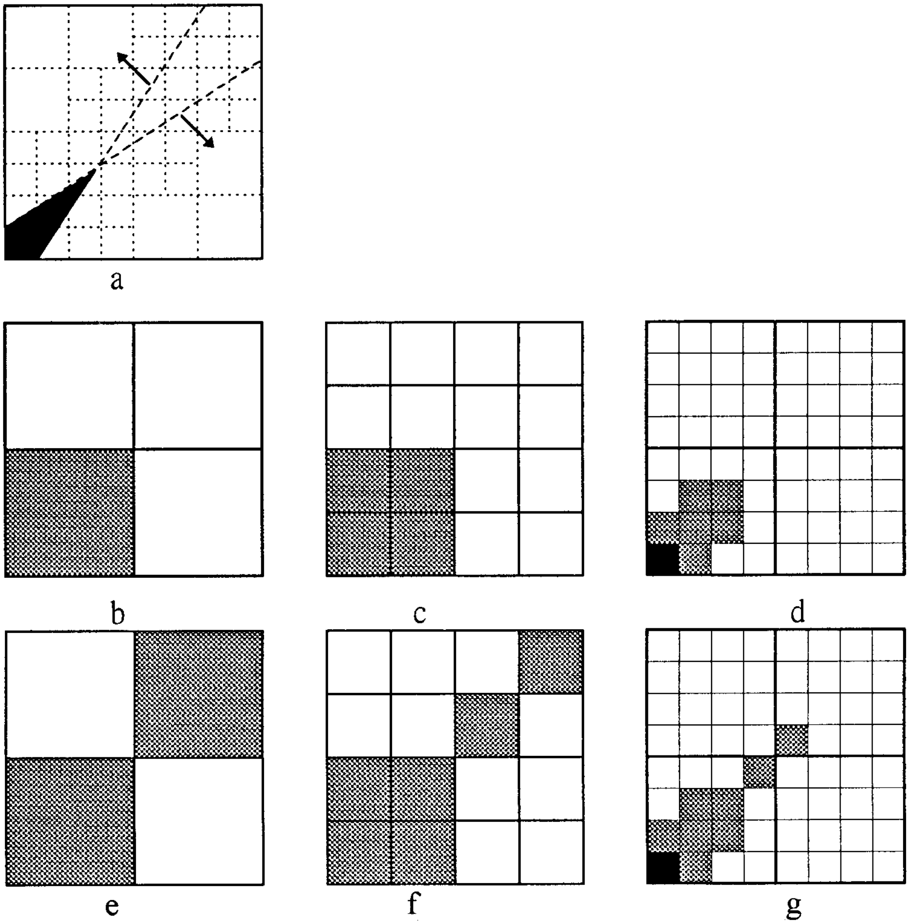


FIG. 2. (a) An object which is the intersection of two half-planes (indicated by dash lines). (b)–(d) Correct color determination. (e)–(g) Incorrect color determination.

nation. When the coloring result is output, the gray nodes at each display level will be treated as black node. It will result in a larger display error. Also, the resultant tip of the displayed object is unnatural, which becomes a dangling part.

In the following, for a succinct presentation of the method, it is assumed that there is only a single binary object in a given image. However, in the simulations provided later in the experiment section, multiple objects and objects with multiconnected regions are implemented.

The organization of the paper is as follows. Section 2 describes the binary object representation model and the color determination for progressive display of the primitive object, i.e., a half plane. Section 3 uses the intersection of half planes to represent a convex polygonal object. Rules are given to derive the final block color from the block colors individually determined for the constituent half planes. Section 4 extends the object to a concave polygonal object. Differences in implementing the convex and concave polygons are then explained. Section 5 describes the object model modification under rotation, scaling, and translation variations. Section 6 shows the simulation results of our method when applied to the binary images regarding a business trademark, a mechanical part, and a sport event sign. Furthermore, comparisons between our method and the quadtree representation method are given for an object undergoing a scaling or rotation change. A concluding remark is provided in Section 7.

2. RECURSIVE REPRESENTATION AND PROGRESSIVE DISPLAY OF A HALF PLANAR PRIMITIVE OBJECT

Before describing the representation and progressive display of a general binary object, we shall first consider a primitive object, a half-plane. A half-plane is specified by its directed boundary line, whose direction is defined so that the object lies on the right-hand side of the directed boundary line. The conventional quadtree block subdivision of the object image is used here for progressive display. Let $S[\,]$ denote the entire square image array consisting of $L \times L$ pixels, each black or white, depending on whether the pixel is an object pixel or a background pixel. Let $S[\,]$ denote the root node of the quadtree and let the successive subblocks be denoted as $S[0]$, $S[1]$, $S[2]$, $S[3]$, and $S[00]$, $S[01]$, $S[02]$, $S[03]$, etc.

Assume there is only one half-planar object HP in the image array $S[\,]$. Now consider a new method for the representation and progressive display of this primitive object HP. To display the image of HP, we shall check if each image subblock $S[q_1, \ldots, q_i]$, $i = 1, 2, 3, \ldots$ and $q_i \in \{0, 1, 2, 3\}$, is completely contained in the half-planar object HP. An efficient method for determining whether a subblock is contained in a half plane is to be devised. First, we use the two parameters, angle and y-intercept (or simply intercept), to write the equation of the directed boundary line of the half-plane, which varies with respect to the different coordinate systems defined in the parent block and the four subblocks. Based on this representation scheme, the angle attribute of the half-plane HP with respect to each subblock remains constant, but the intercept attribute with each image subblock is different. There will exist a recursive relation between the intercept attribute of a parent block and that of a child subblock. Hence, we can obtain the intercept attribute of a child subblock from its parent block throughout the progressive display. The information of angle and intercept of HP will be used in the node color determination to be described below.

To represent each individual half-plane in the image array $S[\,]$, let $\mathrm{SH}[q_1, \ldots, q_i]$ denote the subblock $S[q_1, \ldots, q_i]$ with regard to a particular half planar object HP. In other words, $\mathrm{SH}[q_1, \ldots, q_i]$ denotes the same image array as $S[q_1, \ldots, q_i]$, but is used exclusively to represent the particular half-planar object HP by ignoring the existence of other possible half planes.

Let the color of $\mathrm{SH}[q_1, \ldots, q_i]$ be denoted by color($\mathrm{SH}[q_1, \ldots, q_i]$); then obviously (refer to Fig. 3a)

(a) color($\mathrm{SH}[q_1, \ldots, q_i]$) is black iff (if and only if) $S[q_1, \ldots, q_i] \subseteq \mathrm{HP}$.

(b) color($\mathrm{SH}[q_1, \ldots, q_i]$) is white, iff $S[q_1, \ldots, q_i] \cap \mathrm{HP} = \Phi$ (i.e., empty set).

(c) color($\mathrm{SH}[q_1, \ldots, q_i]$) is gray, otherwise.

Note that if color($\mathrm{SH}[q_1, \ldots, q_i]$) = black, then color($\mathrm{SH}[q_1, \ldots, q_i, q_{i+1}]$) = black for $q_{i+1} \in \{0, 1, 2, 3\}$. Similarly, if color($\mathrm{SH}[q_1, \ldots, q_i]$) = white, then color($\mathrm{SH}[q_1, \ldots, q_i, q_{i+1}]$) = white for $q_{i+1} \in \{0, 1, 2, 3\}$.

Now consider the coordinate system within each block. In any block $S[q_1, \ldots, q_i]$, let the origin point (0, 0) be located at the upper left corner, let the x-axis be the horizontal axis whose positive direction is to the right, and let the y-axis be the vertical axis whose positive direction points downward, as depicted in Fig. 3b. The y-intercept and angle of the directed boundary line of the half-planar object HP are denoted as $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i])$ and $\mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i])$. Here the angle is defined to be the counterclock angle between the x-axis and the directed boundary line of the half-planar object.

Then the equation of the (directed) boundary line of the half planar object is represented by

$$y = \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i]) - x \tan(\mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i])),$$

where tan( ) is the tangent function.

A recursive relation between $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i])$ and $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_{i-1}])$ can be derived based on the follow-

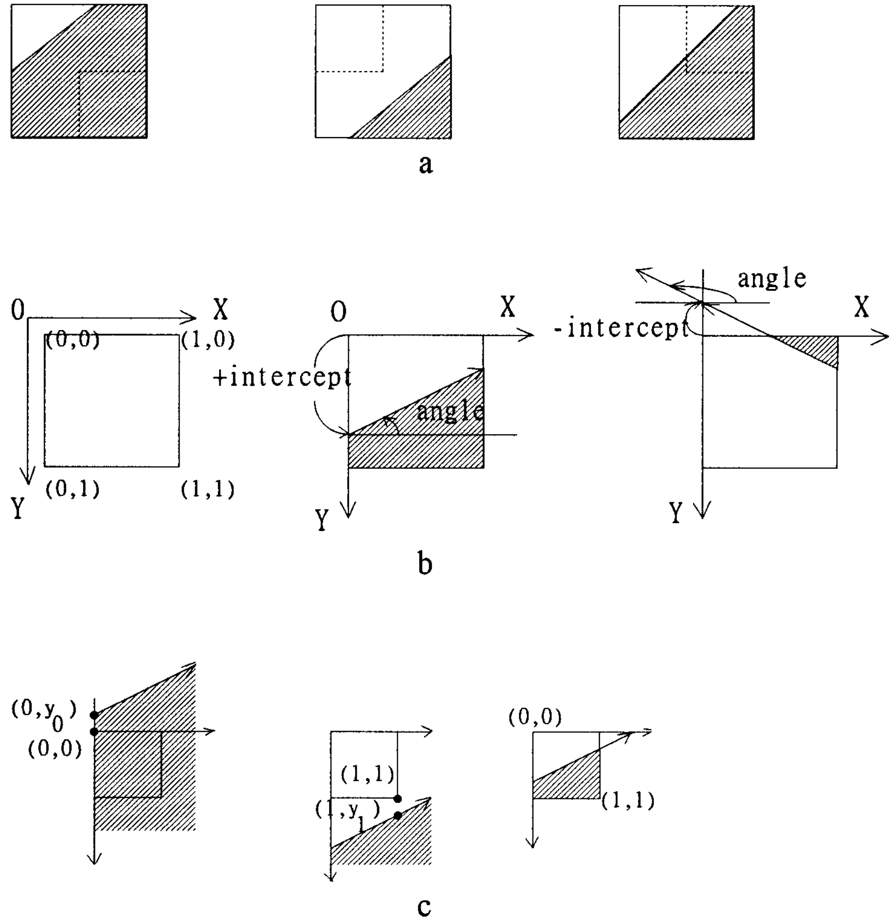


FIG. 3. (a) The color of a subblock is determined by its location related to the half-plane. (b) Definition of the coordinate system and intercept and angle of a half plane boundary line in SH[$q_1, \ldots, q_i$]. (c) The color of a subblock is determined based on the locations of point (0, 0) and point (1, 1).

ing coordinate system transformations. Let the five coordinate systems within the five blocks, SH[$q_1, \ldots, q_{i-1}$], SH[$q_1, \ldots, q_{i-1}$, 0], SH[$q_1, \ldots, q_{i-1}$, 1], SH[$q_1, \ldots, q_{i-1}$, 2], and SH[$q_1, \ldots, q_{i-1}$, 3], be shown in Fig. 4. Each coordination system has its own origin, x-axis, and y-axis. Also, assume that the scaling at the upper level $i - 1$ is one-half of that at the lower level i. Let the coordinates of a point be (x, y) at level $i - 1$ and (x', y') at level i. Although the boundary line of the half-planar object remains fixed, it is described differently within the five blocks. We have the following properties:

Property 1. ANG(SH[$q_1, \ldots, q_i$]) is the same for $i = 1, 2, \ldots, n$. That is, ANG(SH[$q_1, \ldots, q_i$]) = ANG(SH[$q_1, \ldots, q_{i-1}$]) = ... = ANG(SH[q_1, q_2]) = ANG(SH[q_1]) = ANG(SH[ ]).

Property 2. ITC(SH[$q_1, \ldots, q_i$]) is a function of ITC(SH[$q_1, \ldots, q_{i-1}$]) and ANG(SH[ ]), to be derived below.

The boundary line equation at level $i - 1$ is given by

$$y = \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_{i-1}]) - x \tan(\mathrm{ANG}(\mathrm{SH}[\,])).$$

However, in the subblock S[$q_1, \ldots, q_{i-1}$, 0], the equation for the same line is given by

$$y' = \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_{i-1}, 0]) - x' \tan(\mathrm{ANG}(\mathrm{SH}[\,])).$$

Through the coordinate transformation $x' = 2x$; $y' = 2y$, the boundary line equation in S[$q_1, \ldots, q_{i-1}$, 0] is also given by

$$y' = 2\,\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_{i-1}]) - x' \tan(\mathrm{ANG}(\mathrm{SH}[\,])).$$

Therefore, ITC(SH[$q_1, \ldots, q_{i-1}$, 0]) = 2 ITC(SH[$q_1, \ldots, q_{i-1}$]).

Similarly, in the subblock SH[$q_1, \ldots, q_{i-1}, q_i$], q_i = 1, 2, 3, as shown in Fig. 4, we can derive

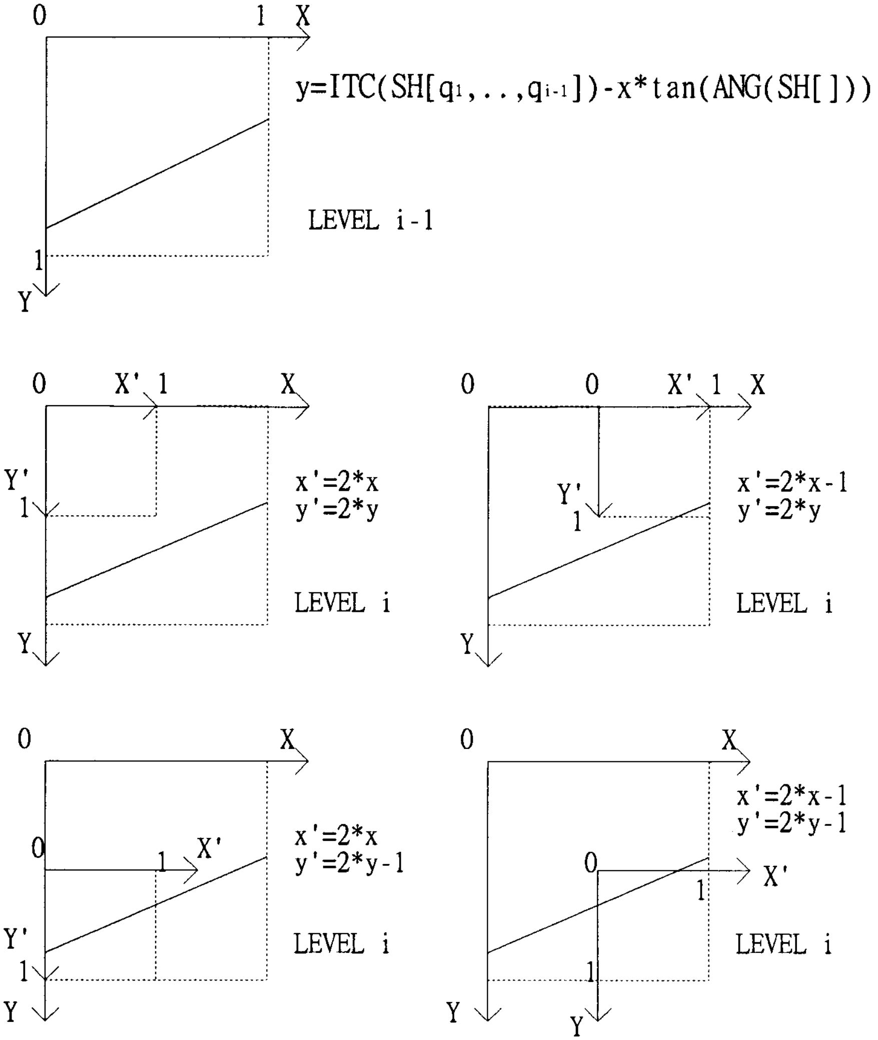


FIG. 4. The relationship between the ITC(SH[$q_1, \ldots, q_i$]) value of a parent block $S[q_1, \ldots, q_i]$ and the ITC(SH[$q_1, \ldots, q_i$]) values of the child subblocks.

$$\begin{aligned}\mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1},1]) &= 2\,\mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1}])\\ &\quad - \tan(\mathrm{ANG}(\mathrm{SH}[\,])).\end{aligned}$$

$$\mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1},2]) = 2\,\mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1}]) - 1.$$

$$\begin{aligned}\mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1},3]) &= 2\,\mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1}])\\ &\quad - \tan(\mathrm{ANG}(\mathrm{SH}[\,])) - 1.\end{aligned}$$

In general matrix notation, we have

$$\begin{bmatrix}\mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1},0])\\ \mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1},1])\\ \mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1},2])\\ \mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1},3])\end{bmatrix} = \begin{bmatrix}2 & 0 & 0\\ 2 & -1 & 0\\ 2 & 0 & -1\\ 2 & -1 & -1\end{bmatrix} \times \begin{bmatrix}\mathrm{ITC}(\mathrm{SH}[q_1,\ldots,q_{i-1}])\\ \tan(\mathrm{ANG}(\mathrm{SH}[\,]))\\ 1\end{bmatrix}.$$

With the above formula, we can refine the color of a gray block SH[$q_1, \ldots, q_{i-1}$] as the four colors of the subblocks SH[$q_1, \ldots, q_i$], $q_i \in \{0, 1, 2, 3\}$, by the block subdivision of the gray block SH[$q_1, \ldots, q_{i-1}$]. Assume the four corner points in each subblock coordinate system $S[q_1, \ldots, q_i]$ are denoted as points (0, 0), (0, 1), (1, 0), and (1, 1). Since the boundary line equation of half planar object is known, it can be determined whether a corner point is inside the half plane. Furthermore, it will be shown that at most two corner points (instead of four) are needed to determine the color of SH[$q_1, \ldots, q_i$]. For example, refer to Fig. 3c, if point (0, 0) lies inside the half planar object, then the entire $S[q_1, \ldots, q_i]$ lies inside, too; if point (1, 1) does not lie inside the half planar object, then neither does $S[q_1, \ldots, q_i]$. The choice of which corner point to check is based on the angle value of the directed boundary line of the half planar object, ANG(SH[$q_1, \ldots, q_i$]). There are four possible ranges of values for ANG(SH[$q_1, \ldots, q_i$]):

Case (1): $0 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 90$,

Case (2): $90 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 180$,

Case (3): $180 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 270$,

Case (4): $270 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 360$.

Consider Case (1) first. Refer to the half planar object in Fig. 3c.

LEMMA HP1-1. *Assume* $0 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 90$.

(a) *If point* $(0, 0) \in \mathrm{HP}$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ black.

(b) *If point* $(1, 1) \notin \mathrm{HP}$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ white.

(c) *If* (a) *and* (b) *do not hold, then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ gray.

Proof. The equation of the boundary line of the half plane is given by $y = \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i]) - x * \tan(\mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]))$.

(a) When $x = 0$, $y = y_0 = \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i])$. If $y_0 \le 0$, then point (0, 0) is in HP, and so is block $\mathrm{SH}[q_1, \ldots, q_i]$. Thus, $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ black.

(b) When $x = 1$, $y = y_1 = \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i]) - \tan(\mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]))$. If $y_1 > 1$, then point (1, 1) is not in HP, so $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ white.

(c) If $y_0 > 0$ and $y_1 \le 1$, then $\mathrm{SH}[q_1, \ldots, q_i] \cap \mathrm{HP} \neq \Phi$, so $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ gray. ■

The following lemma can be readily shown to be equivalent to the above lemma by checking if $y_0 \le 0$ and $y_1 > 1$:

LEMMA HP1-2. *Assume* $0 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 90$.

(a) *If* $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i]) \le 0$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ black.

(b) *If* $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i]) > 1 + \tan(\mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]))$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ white.

(c) *If* (a) *and* (b) *do not hold, then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ gray.

Similarly, the color determination rules which are stated below for other cases can be also derived:

LEMMA HP2. *Assume* $90 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 180$.

(a) *If* $1 \le \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i])$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ black.

(b) *If* $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i]) < \tan(\mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]))$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ white.

(c) *If* (a) *and* (b) *do not hold, then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ gray.

LEMMA HP3. *Assume* $180 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 270$.

(a) *If* $1 + \tan(\mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i])) \le \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i])$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ black.

(b) *If* $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i]) < 0$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ white.

(c) *If* (a) *and* (b) *do not hold, then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ gray.

LEMMA HP4. *Assume* $270 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 360$.

(a) *If* $1 < \mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i])$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ white.

(b) *If* $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i]) \le \tan(\mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]))$, *then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ black.

(c) *If* (a) *and* (b) *do not hold, then* $\mathrm{color}(\mathrm{SH}[q_1, \ldots, q_i]) =$ gray.

The progressive display of a half plane is to color the blocks from level to level. If a block has a white or black color, then its successive subblocks all have the same color, so it is not necessary to subdivide the block any further. Only those blocks with color of gray are to be further subdivided into smaller subblocks to see if the color of the smaller subblock becomes white or black.

Now take the half-planar object in Fig. 5 as an example. Its ITC(SH[ ]) is 0.75, and its ANG(SH[ ]) is 26.5. Since $0 < 26.5 < 90$, the half planar object corresponds to case (1), in which $0 \le \mathrm{ANG}(\mathrm{SH}[q_1, \ldots, q_i]) < 90$. Then the new $\mathrm{ITC}(\mathrm{SH}[q_1])$ values are

$$\begin{bmatrix} \mathrm{ITC}(\mathrm{SH}[0]) \\ \mathrm{ITC}(\mathrm{SH}[1]) \\ \mathrm{ITC}(\mathrm{SH}[2]) \\ \mathrm{ITC}(\mathrm{SH}[3]) \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 2 & -1 & 0 \\ 2 & 0 & -1 \\ 2 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0.75 \\ 0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1 \\ 0.5 \\ 0 \end{bmatrix}.$$

Since $1 + \tan(\mathrm{ANG}(\mathrm{SH}[\ ])) = 1 + \tan(26.5) = 1 + 0.5 = 1.5$, the colors can be determined based on Lemma HP1-2:

$$\begin{aligned} &\mathrm{color}(\mathrm{SH}[0]) = \text{gray, for } \mathrm{ITC}(\mathrm{SH}[0]) = 1.5 \\ &\mathrm{color}(\mathrm{SH}[1]) = \text{gray, for } \mathrm{ITC}(\mathrm{SH}[1]) = 1 \\ &\mathrm{color}(\mathrm{SH}[2]) = \text{gray, for } \mathrm{ITC}(\mathrm{SH}[2]) = 0.5 \\ &\mathrm{color}(\mathrm{SH}[3]) = \text{black, for } \mathrm{ITC}(\mathrm{SH}[3]) = 0. \end{aligned}$$

Gray blocks SH[0], SH[1], and SH[2] need to be further subdivided into subblocks. The computation of $\mathrm{ITC}(\mathrm{SH}[q_1, \ldots, q_i])$ for all subblocks can be done in parallel by some parallel hardware such as the pyramid machine [16, 17].

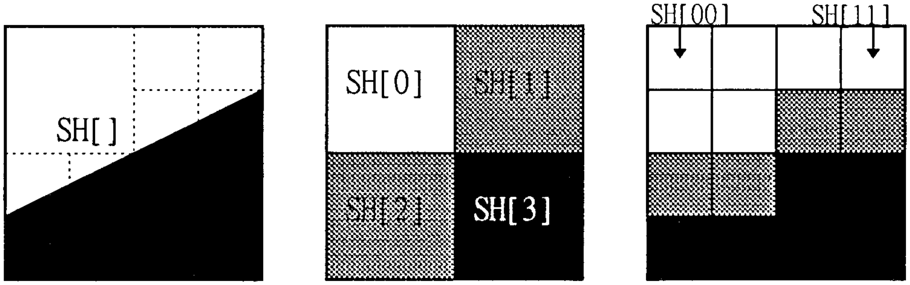


FIG. 5. The subdivision of a half-planar object. ITC(SH[ ]) = 0.75, ANG(SH[ ]) = 26.5.

3. REPRESENTATION AND DISPLAY OF A CONVEX POLYGONAL OBJECT EXPRESSED AS AN INTERSECTION OF HALF-PLANES

Any polygonal object can be expressed as an intersection–union form of half-planes. The interior of a convex polygonal object always lies on the right-hand side of the directed boundary line of constituent half-planes (refer to Fig. 6). However, this is not the case for a concave polygonal object (refer to Fig. 8). Therefore, we need to classify the polygon into two cases: convex and concave. We shall consider the convex polygon case first. A convex polygon (denoted as CV) can be expressed as the intersection of half-planar objects CV = $\mathrm{HP}_1 \cap \mathrm{HP}_2 \cap .. \cap \mathrm{HP}_\mathrm{h}$, is where HP_h is the hth half-planar object. Each half-plane boundary line coincides with an edge of the convex polygon. In Fig. 6a a convex polygon CV is equal to the intersection of HP_1, HP_2, and HP_3 shown in Figs. 6b–6d. Here the boundary line of HP_h contains two parts: the real edge of CV (the heavy line in Fig. 6) and the extended line of the real edge (the dashed line in Fig. 6).

Let $\mathrm{SV}[q_1, \ldots, q_i]$ denote the image subblock at location $q_1, \ldots, q_i$ with regard to the image of the particular convex polygonal object CV. For the convex object shown in Fig. 6, Figs. 7a–7c are the contents of color($\mathrm{SH}_1[q_1 q_2]$), color($\mathrm{SH}_2[q_1 q_2]$), and color($\mathrm{SH}_3[q_1 q_2]$), where **B**, **W**, and **G** indicate the respective subblock colors black, white, and gray, "**real edge**" indicates real-edge-of-HP, and "**ext-line**" indicate extended-line-of-HP. Figure 7d shows the content of color($\mathrm{SV}[q_1 q_2]$).

Now we will describe the color determination rules for a convex polygonal object based on the block coloring results determined with respect to the individual constituent half-planes. As shall be seen, not all constituent half-planar objects need to be examined in order to determine the final color of the image subblocks.

The relationship between the subblock and the half-planar object can be only one of the following three possibilities:

(1) $S[q_1, \ldots, q_i] \cap \mathrm{HP}_h = \Phi$ (that is, color($\mathrm{SH}_h[q_1, \ldots, q_i]$) = white).

(2) $S[q_1, \ldots, q_i] \subseteq \mathrm{HP}_h$ (that is, color($\mathrm{SH}_h[q_1, \ldots, q_i]$) = black).

(3) $S[q_1, \ldots, q_i]$ intersects with HP_h. Namely,

(3.a) $S[q_1, \ldots, q_i] \cap$ real-edge-of-$\mathrm{HP}_h \neq \Phi$, or

(3.b) $S[q_1, \ldots, q_i] \cap$ extended-line-of-$\mathrm{HP}_h \neq \Phi$.

Based on the above spatial relationships, the connection between color($\mathrm{SH}_h[q_1, \ldots, q_i]$) and color($\mathrm{SV}[q_1, \ldots, q_i]$) can be described by one of the following four lemmas. Assume CV = $\mathrm{HP}_1 \cap \mathrm{HP}_2 \cap .. \cap \mathrm{HP}_n$.

LEMMA CV1. *If for some h,* $S[q_1, \ldots, q_i] \cap \mathrm{HP}_h = \Phi$, color($\mathrm{SV}[q_1, \ldots, q_i]$) = white.

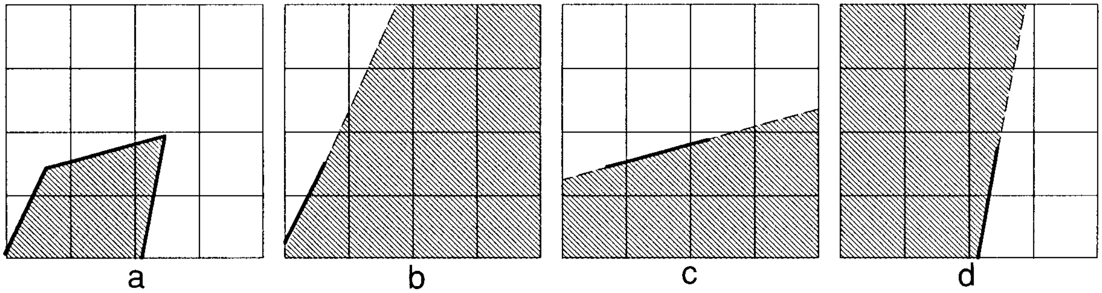


FIG. 6. (a) The convex polygonal object, CV. (b), (c), (d) Three half planar objects HP_1, HP_2, and HP_3. CV = $\mathrm{HP}_1 \cap \mathrm{HP}_2 \cap \mathrm{HP}_3$. The heavy lines are the real edges of the polygon CV, the dashed lines are the extended lines.

W	ext-line	B	B
ext-line	ext-line	B	B
real edge	B	B	B
real edge	B	B	B

a

W	W	W	W
W	W	ext-line	ext-line
real edge	real edge	real edge	B
B	B	B	B

b

B	B	ext-line	W
B	B	ext-line	W
B	B	real edge	W
B	B	real edge	W

c

W	W	W	W
W	W	W	W
G	G	G	W
G	B	G	W

d

FIG. 7. Color determination results of the object in Fig. 6: (a) color(SH$_1[q_1q_2]$), (b) color(SH$_2[q_1q_2]$), (c) color(SH$_3[q_1q_2]$), (d) color(SQCV$[q_1q_2]$).

Proof. If $S[q_1, \ldots, q_i] \cap \mathrm{HP}_h = \Phi$, then $S[q_1, \ldots, q_i] \cap \mathrm{CV} = \Phi$. Therefore, color(SV$[q_1, \ldots, q_i]$) = white. ■

LEMMA CV2. *If and only if for all h,* $S[q_1, \ldots, q_i] \subseteq \mathrm{HP}_h$, color(SV$[q_1, \ldots, q_i]$) = black.

Proof. (1) The "if" part:

If for all h, $S[q_1, \ldots, q_i] \subseteq \mathrm{HP}_h$, then $S[q_1, \ldots, q_i] \subseteq \mathrm{CV}$. Therefore, color(SV$[q_1, \ldots, q_i]$) = black.

(2) The "only if" part:

Assume color(SV$[q_1, \ldots, q_i]$) = black. If for some h, $S[q_1, \ldots, q_i] \not\subset \mathrm{HP}_h$, then $S[q_1, \ldots, q_i] \not\subset \mathrm{CV}$, then color(SV$[q_1, \ldots, q_i]$) $\neq$ black. It is a contradiction, so the only if part is valid. ■

LEMMA CV3. *If and only if for some h,* $S[q_1, \ldots, q_i] \cap$ *real-edge-of-*HP$_h \neq \Phi$, color(SV$[q_1, \ldots, q_i]$) = gray.

Proof. (1) The "if" part:

If $S[q_1, \ldots, q_i] \cap$ real-edge-of-HP$_h \neq \Phi$, then $S[q_1, \ldots, q_i] \not\subset \mathrm{HP}_h$ and then $S[q_1, \ldots, q_i] \not\subset \mathrm{CV}$. Also since real-edge-of-HP$_h \subset$ CV and $S[q_1, \ldots, q_i] \cap$ real-edge-of-HP$_h \neq \Phi$, then $S[q_1, \ldots, q_i] \cap \mathrm{CV} \neq \Phi$. Therefore, color(SV$[q_1, \ldots, q_i]$) = gray.

(2) The "only if" part:

If color(SV$[q_1, \ldots, q_i]$) = gray, then $S[q_1, \ldots, q_i] \cap$ boundary-of-CV $\neq \Phi$. It implies that for some h, $S[q_1, \ldots, q_i] \cap$ real-edge-of-HP$_h \neq \Phi$, since boundary-of-CV is the union of all real-edge-of-HP$_h$. ■

LEMMA CV4. *If the following conditions are satisfied,*

(a) *for all h,* $S[q_1, \ldots, q_i] \cap$ *real-edge-of-*HP$_h = \Phi$

(b) *for some h,* $S[q_1, \ldots, q_i] \cap$ *extended-line-of-*HP$_h \neq \Phi$, *then* color(SV$[q_1, \ldots, q_i]$) = Wt.

Proof. If for some h, $S[q_1, \ldots, q_i] \cap$ extended-line-of-HP$_h \neq \Phi$, then $S[q_1, \ldots, q_i] \not\subset \mathrm{HP}_h$. It implies $S[q_1, \ldots, q_i] \not\subset \mathrm{CV}$ and color(SV$[q_1, \ldots, q_i]$) $\neq$ black. Therefore, color(SV$[q_1, \ldots, q_i]$) = gray or white.

Suppose color(SV$[q_1, \ldots, q_i]$) = gray, then for some h, $S[q_1, \ldots, q_i] \cap$ real-edge-of-HP$_h \neq \Phi$. By definition, this is not true, therefore, color(SV$[q_1, \ldots, q_i]$) cannot be gray. There is only one possibility left, that is, color(SV$[q_1, \ldots, q_i]$) = white. ■

4. REPRESENTATION AND DISPLAY OF A CONCAVE POLYGONAL OBJECT EXPRESSED AS A UNION OF CONVEX POLYGONS

A concave polygonal object (denoted as CC) can be expressed as the union of convex subpolygons [9] which lie within the boundary of the concave polygon. Let CC = CV$_1$ ∪ CV$_2$ ∪ $\cdots$ ∪ CV$_k$, where CV$_c$ is the cth convex subpolygon of CC. For example, the concave object CC in Fig. 8a is the union of two convex polygonal objects CV$_1$, CV$_2$ shown in Figs. 8b and 8c. CC = CV$_1$ ∪ CV$_2$. A boundary of CV$_c$ may be either a virtual boundary, denoted as "virtual-boundary-of-CV$_c$," or a real boundary of CC, denoted as "real-boundary-of-CV$_c$." For example, the heavy lines in Fig. 8 are the real boundary lines. The dashed lines are the virtual boundary lines.

Let SC$[q_1, \ldots, q_i]$ denote the image subblock at location $q_1, \ldots, q_i$ with regard to the image of a particular concave polygonal object CC. Figures 9a and 9b show the contents of color(SV$_1[q_1q_2]$) and color(SV$_2[q_1q_2]$), where **V-B** indicates virtual-boundary-of-CV, which means there is no real boundary but only a virtual boundary intersecting with this subblock. Figure 9c shows the content of color(SC$[q_1q_2]$).

The main difference between a concave polygonal object and a convex polygonal object is the interior of the concave polygonal object does not always lie on the right-hand side of all half-planes related to all sides of the concave polygon. Instead, we need to decompose the concave polygon into a set of disjoint convex polygons. Also, the convex polygons may contain virtual edges which do not exist physically.

Assume CC = CV$_1$ ∪ CV$_2$ ∪ . . . CV$_k$. The relationship between subblock $S[q_1, \ldots, q_i]$ and a convex polygonal object CV$_c$ can be only one of the following three possibilities:

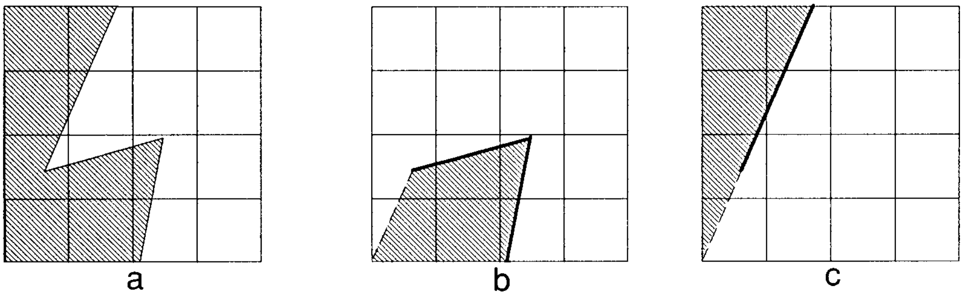


FIG. 8. (a) A concave polygonal object CC. (b), (c) Two convex polygonal objects CV_1 and CV_2. $\mathrm{CC} = \mathrm{CV}_1 \cup \mathrm{CV}_2$. The heavy lines are the real boundary lines. The dashed lines are the virtual boundary lines.

(1) If $S[q_1, \ldots, q_i] \cap \mathrm{CV}_c = \Phi$, then color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = white.

(2) If $S[q_1, \ldots, q_i] \subseteq \mathrm{CV}_c$, then color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = black.

(3) If $S[q_1, \ldots, q_i]$ intersect with the boundary of CV_c. Namely, if

$$S[q_1, \ldots, q_i] \cap \text{real-boundary-of-}\mathrm{CV}_c \neq \Phi,$$

then color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = gray.

The relationship between color($\mathrm{SV}_c[q_1, \ldots, q_i]$), $c = 1, 2, \ldots, k$ and color($\mathrm{SC}[q_1, \ldots, q_i]$) can be described by one of the following four lemmas:

Lemma CC1. *If for some c,* color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = black, color($\mathrm{SC}[q_1, \ldots, q_i]$) = black.

Proof. If color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = black, then $S[q_1, \ldots, q_i] \subseteq \mathrm{CV}_c$, and $S[q_1, \ldots, q_i] \subseteq \mathrm{CC}$. Therefore, color($\mathrm{SC}[q_1, \ldots, q_i]$) = black. ■

Lemma CC2. *If and only if for all c,* $c = 1, 2, \ldots, k$, color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = white, color($\mathrm{SC}[q_1, \ldots, q_i]$) = white.

Proof. (1) The "if" part:

If for all c, color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = white, then for all c, $S[q_1, \ldots, q_i] \cap \mathrm{CV}_c = \Phi$, and $S[q_1, \ldots, q_i] \cap \mathrm{CC} = \Phi$. Therefore, color($\mathrm{SC}[q_1, \ldots, q_i]$) = white.

(2) The "only if" part can be easily proved by contradiction. ■

Lemma CC3. *If and only if for some c,* color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = gray, color($\mathrm{SC}[q_1, \ldots, q_i]$) = gray.

Proof. (1) The "if" part:

If color($\mathrm{SV}_c[q_1, \ldots, q_i]$) = gray (that is, $S[q_1, \ldots, q_i] \cap$ real-boundary-of-$\mathrm{CV}_c \neq \Phi$), then $S[q_1, \ldots, q_i] \not\subseteq \mathrm{CV}_c$, and $S[q_1, \ldots, q_i] \not\subseteq \mathrm{CC}$. Also since real-boundary-of-$\mathrm{CV}_c \subseteq \mathrm{CC}$ and $S[q_1, \ldots, q_i] \cap$ real-boundary-of-$\mathrm{CV}_c \neq \Phi$, then $S[q_1, \ldots, q_i] \cap \mathrm{CC} \neq \Phi$. Therefore, color($\mathrm{SC}[q_1, \ldots, q_i]$) = gray.

(2) The "only if" part:

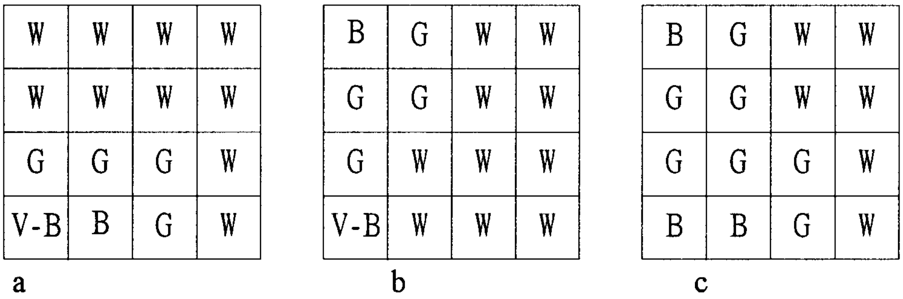
If color($\mathrm{SC}[q_1, \ldots, q_i]$) = gray, then $S[q_1, \ldots, q_i] \cap$ boundary-of-CC $\neq \Phi$. It implies for some c, $S[q_1, \ldots, q_i] \cap$ real-boundary-of-$\mathrm{CV}_c \neq \Phi$, since boundary-of-CC is the union of all real-boundary-of CV_c. ■

Lemma CC4. *If the following conditions are all satisfied:*

(a) *for all c,* $S[q_1, \ldots, q_i] \cap$ real-boundary-of-$\mathrm{CV}_c = \Phi$

(b) *for some c,* $S[q_1, \ldots, q_i] \cap$ virtual-boundary-of-$\mathrm{CV}_c \neq \Phi$,

then color($\mathrm{SC}[q_1, \ldots, q_i]$) = black.



W	W	W	W
W	W	W	W
G	G	G	W
V-B	B	G	W

a

B	G	W	W
G	G	W	W
G	W	W	W
V-B	W	W	W

b

B	G	W	W
G	G	W	W
G	G	G	W
B	B	G	W

c

FIG. 9. Color determination results of the object in Fig. 8. (a) Color($\mathrm{SV}_1[q_1q_2]$), (b) Color($\mathrm{SV}_2[q_1q_2]$), (c) Color($\mathrm{SC}[q_1q_2]$).

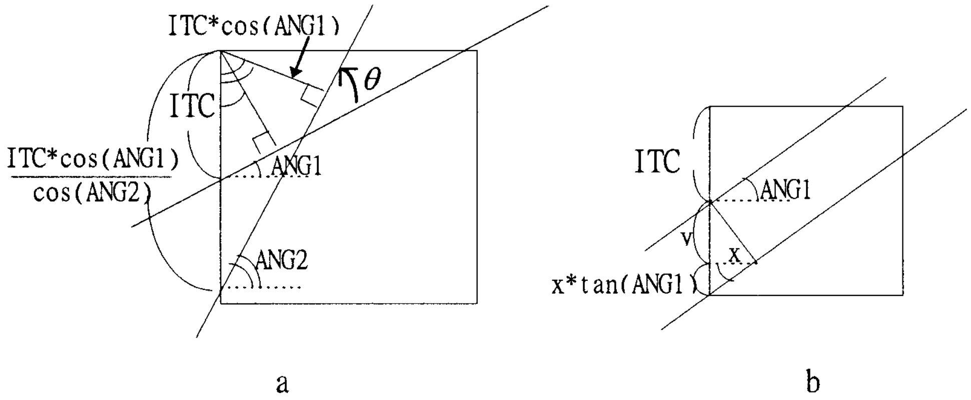


FIG. 10. (a) The rotation of a half plane from angle ANG1 to ANG2. (b) The translation of a half plane by a vector (x, y).

Proof. If for some c, $S[q_1, \dots, q_i] \cap$ virtual-boundary-of-$\mathrm{CV}_c \neq \Phi$, then $S[q_1, \dots, q_i] \cap \mathrm{CV}_c \neq \Phi$, and $S[q_1, \dots, q_i] \cap \mathrm{CC} \neq \Phi$, so color(SC[$q_1, \dots, q_i$]) $\neq$ white. Therefore, color(SC[$q_1, \dots, q_i$]) can only be gray or black.

Next, suppose color(SC[$q_1, \dots, q_i$]) = gray, then $S[q_1, \dots, q_i] \cap$ boundary-of-CC $\neq \Phi$, and for some c, $S[q_1, \dots, q_i] \cap$ real-boundary-of-$\mathrm{CV}_c \neq \Phi$. But this is not true, since $S[q_1, \dots, q_i] \cap$ real-boundary-of-$\mathrm{CV}_c = \Phi$, for all c.

Thus, color(SC[$q_1, \dots, q_i$]) cannot be gray. The only possibility left is that color(SC[$q_1, \dots, q_i$]) = black. ■

5. DATA MODEL MODIFICATION UNDER ROTATION/SCALING/TRANSLATION VARIATIONS

During browsing a binary object, it is often desirable to modify the object view with regard to a change in its orientation, scale, or location. With our data representation, these view modifications can be readily made in the following manners:

(a) Rotation

Assume the object is rotated by a counterclock angle θ around the origin of the coordinate system. Then the two parameters of the boundary line each constituent half-plane of the object are modified as (refer to Fig. 10a)

(i) ANG2 = ANG1 + θ.
(ii) ITC2 = ITC1 cos(ANG1)/cos(ANG2).

If the rotation center is not at the origin of the coordinate system, then one can translate the coordinate system such that the center of rotation is at the new origin. After making the rotation, one translates the new coordinate system back to its old coordinate system. The formula for ITC2 is slightly complicated in this case.

(b) Scaling

If the object is undergoing a scaling change by a factor of k, then the two parameters associated with each constituent half-plane are modified as

(i) ANG2 = ANG1
(ii) ITC2 = ITC1 × k.

(c) Translation

If the object is translated by a vector (x, y), then the two parameters associated with each constituent half-plane are modified as (refer to Fig. 10b)

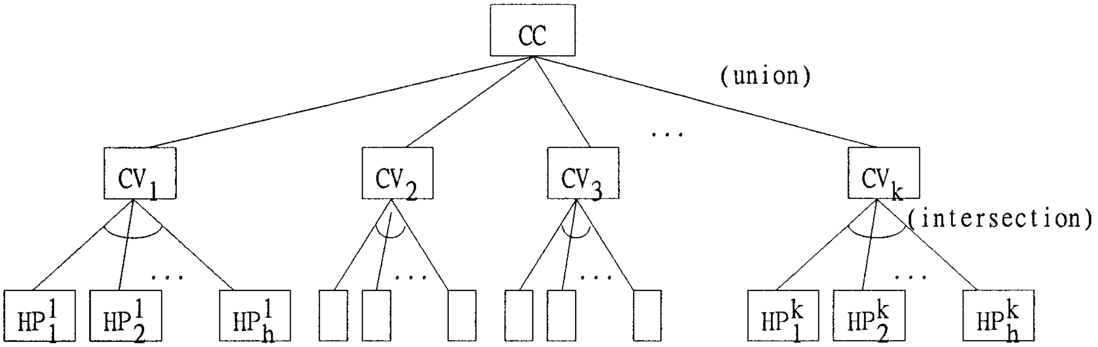


FIG. 11. A hierarchical representation of a concave polygonal object.

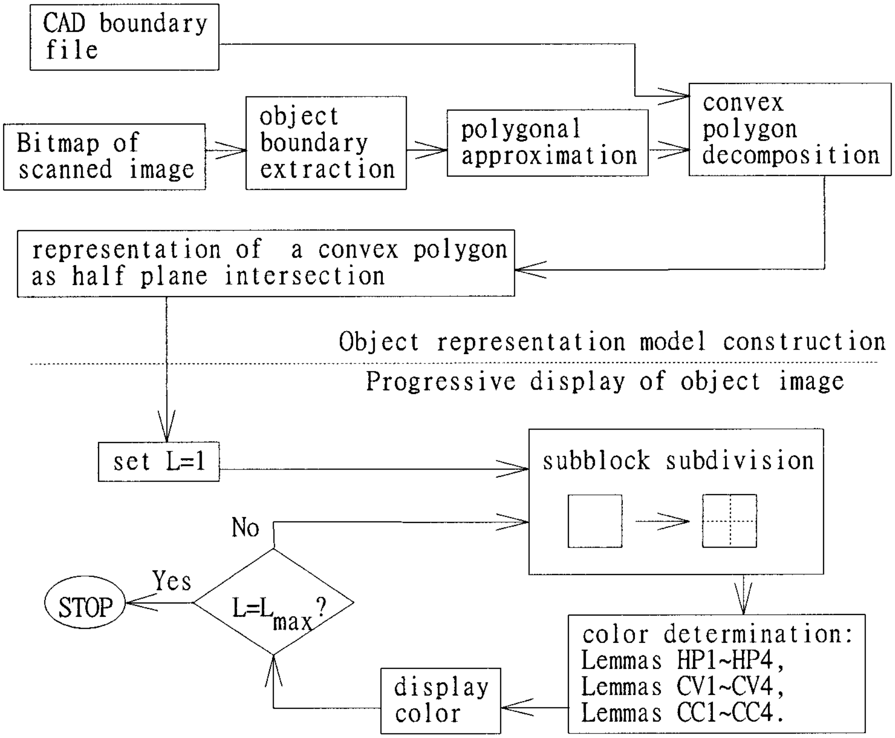

CAD boundary file
Bitmap of scanned image
object boundary extraction
polygonal approximation
convex polygon decomposition
representation of a convex polygon as half plane intersection
Object representation model construction
Progressive display of object image
set L=1
subblock subdivision
No
Yes
STOP
L=L_max?
color determination: Lemmas HP1~HP4, Lemmas CV1~CV4, Lemmas CC1~CC4.
display color

FIG. 12. The overall system block diagram.

(i) ANG2 = ANG1

(ii) ITC2 = ITC1 + y + x × tan(ANG1).

Any combination of the above rotation, scaling, and translation can be implemented, too. Recall that the object model always retains the real (not the truncated) values of the (angle, intercept) parameters of the half-planes at all levels of display resolution. Therefore, the display error will not be accumulated as the level of resolution goes from top to bottom.

6. SIMULATION RESULTS

In this section we shall first show the half-plane-based representation of the binary objects and apply the color determination rules to obtain the progressive object display results. Then we compare the performance between our method and the quadtree representation method under the scaling and rotation variations. There are mainly two ways of inputting a binary object. One is to use the CAD tool to create the boundary representation (BREP) file, a set of polygon edges, of the object, and the other is to use the bit map of the object generated by a scanner. For the bit map input, the conventional techniques for object boundary extraction and polygonal approximation must be applied first to obtain the boundary edges of the object. After the boundary edges of the object are obtained, a convex polygon decomposition method [9] is used to partition the object in a form of a general concave polygon into a union of convex subpolygons. Each subpolygon can be,

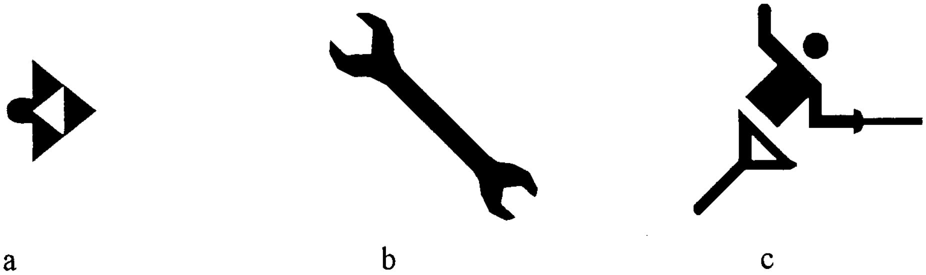


FIG. 13. CAD generated objects: (a) a trademark; (b) a wrench; (c) a sport event sign.

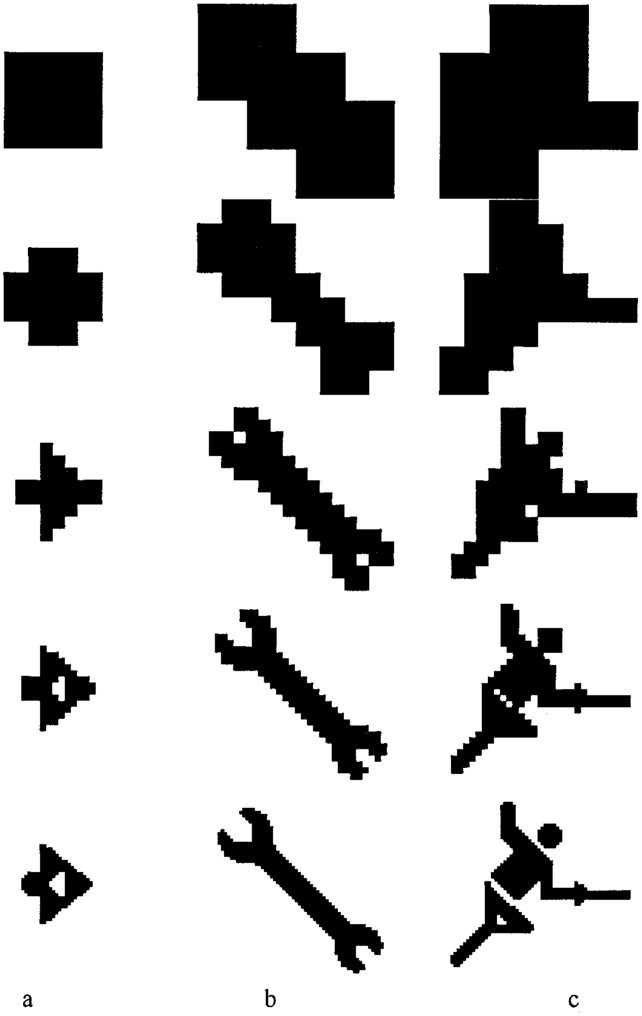


FIG. 14. The progressive displays of binary objects obtained by our method for the first five levels of resolution.

TABLE 2
The Memory Storage and Computation Time Required

	Resource	
Object	Memory storage (bytes)	Computation time (s)
Trademark	80	0.05
Wrench	238	0.38
Sport event sign	280	0.61

in turn, represented as an intersection of half-planes, as shown in Fig. 11. The overall block diagram of our system is given in Fig. 12.

Three test objects, one business trademark, one mechanical part (wrench), and one sport event sign, are chosen to illustrate our method. They are shown in Fig. 13. The trademark is a union of four convex polygons which are represented by 18 half-planes. The object model is given in Fig. 16a. The representation model of the wrench contains 11 convex polygons and 54 half-planes, and the sport event sign contains 12 convex polygons and 64 half-planes. The progressive displays of three binary objects for the first five levels of resolution are given in Figs. 14a–14c. The quadtree representation model is given in Fig. 15. As the resolution level goes down, the display becomes finer and finer. This display mode with a changing level of detail is very useful for object browsing on the internet/intranet. Here, the color determination rules applied to each image block are based on Lemmas HP1–HP4 for the half-planes, Lemmas CV1–CV4 for the convex objects, and CC1–CC4 for the concave objects. The program is run on a Pentium-90 PC under DOS 6.22. The language used is C. The memory space and computation time required for modeling and displaying these three objects are given in Table 2.

Next, we examine the effects of scaling and rotation on the display error and object model when using both our method and the quadtree representation method. Figure

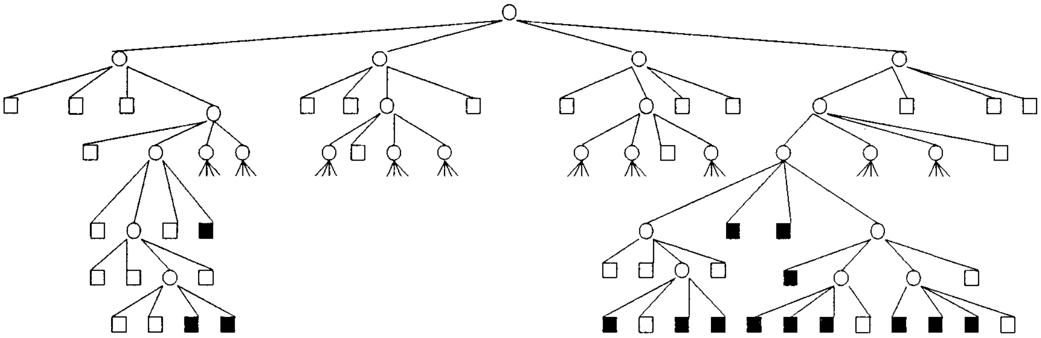


FIG. 15. Part of the quadtree representation for the trade mark shown in Fig. 14a.

CV_j	constituent half planes
1	1~9
2	10~12
3	13~15
4	16~18

HP_i	intercept value	angle value
1	0.2580	164.3577
2	0.1680	135.0000
3	-0.1071	105.6423
4	0.6071	74.3577
5	0.3320	45.0000
6	0.2420	15.6423
7	0.2000	0.0000
8	114591.56	269.9999
9	0.3000	180.0000
10	-0.1100	321.3402
11	0.4100	218.6598
12	114591.56	89.9999
13	-0.1100	321.3402
14	0.6100	218.6598
15	186211.28	89.9999
16	0.6100	218.6598
17	114591.56	89.9999
18	0.0900	321.3402

a

HP_i	intercept value	angle value
1	0.5160	164.3577
2	0.3360	135.0000
3	-0.2143	105.6423
4	1.2143	74.3577
5	0.6640	45.0000
6	0.4840	15.6423
7	0.4000	0.0000
8	229183.12	269.9999
9	0.6000	180.0000
10	-0.2200	321.3402
11	0.8200	218.6598
12	229183.12	89.9999
13	-0.2200	321.3402
14	1.2200	218.6598
15	372422.57	89.9999
16	1.2200	218.6598
17	229183.12	89.9999
18	0.1800	321.3402

b

HP_i	intercept value	angle value
1	0.5129	194.3577
2	0.2460	165.0000
3	-0.0808	135.6423
4	-1.3203	104.3577
5	1.8140	75.0000
6	0.6666	45.6423
7	0.4618	30.0000
8	-0.8000	299.9999
9	0.6928	210.0000
10	-0.1737	351.3402
11	1.7595	248.6598
12	-0.8000	119.9999
13	-0.1737	351.3402
14	2.6178	248.6598
15	-1.3000	119.9999
16	2.6178	248.6598
17	-0.8000	119.9999
18	0.1422	351.3402

c

FIG. 16. (a) The half-plane-based representation of the trademark. (b) The model modification of the trademark under a scaling change shown in Fig. 17a. (c) The model modification of the trademark under a rotation shown in Fig. 18a.

17a shows the trade mark enlarged by a scale factor of 2. In our method the angle of all constituent half planes remain unchanged, while the intercept values of all constituent half planes are multiplied by a factor of 2. Figure 17b shows the enlarged display result obtained by our method. Figure 17c shows the display error of our method. Figure 17d shows the enlarged display result by the quadtree representation method, which is implemented by backward computation; namely, the block color of the enlarged display at level L is equal to the block color of the original quadtree display at level L-1, for $L = 1, 2, \ldots$. Figure 17e is the display error of the quadtree representation method. Figure 17f indicates that our method has a smaller display error. This is due to the fact that our edge-based representation method has a higher accuracy in the object boundary line information, compared to the quadtree representation method using a region-based representation.

On the other hand, when the binary object is rotated counterclockwise by an angle of 30°, the angle values of all constituent half planes used in our method are simply increased by 30° and the intercept values are adjusted by a constant factor equal to {cos(ANG(SH[ ]))/cos(ANG(SH[ ]) + 30)}.

The display results and display errors are shown in Figs. 18a–18f. Again, the display by the quadtree representation method is larger. Therefore, our method outperforms the quadtree representation method in terms of display error. Besides, the data model modification under R/S/T variations is much easier in our method, since the data structure of our method remains the same, only the values of (angle, intercept) attributes are modified. In contrast, in the quadtree representation method the quadtree data content generally changes dramatically when the object is subject to R/S/T variations.

7. CONCLUSION

In this paper we have presented a half-plane-based data modeling method for the binary objects such as business trademarks, mechanical parts, and sport event signs. The two basic representation parameters, intercept and angle, used at the parent node are recursively related to those at the child nodes. This recursive relation is crucial for deriving the colors of the nodes for progressive object display. Lemmas for the node color determination are derived first for individual half-planes, then for convex polygons (as intersections of constituent half-planes), and finally for concave polygons (as unions of constituent convex subpolygons). Our modeling method is better than the quadtree representation method in terms of the memory space and data transmission time on the Internet/intranet. Also, when the object is subject to R/S/T variations, our method outperforms the quadtree representation method in terms of the display error and ease in model modification. Simulation results are conducted to show the performance of our method. In the future we shall apply the results obtained in this paper to search for similar binary objects in a database. The objects in the database will be represented by our scheme. In this system the user may use a scanner to input his binary object and the system will convert the binary bitmap into the recursive representation form we proposed. A scheme for computing the similarity measure between the searched object and each reference object at various resolutions is under development.

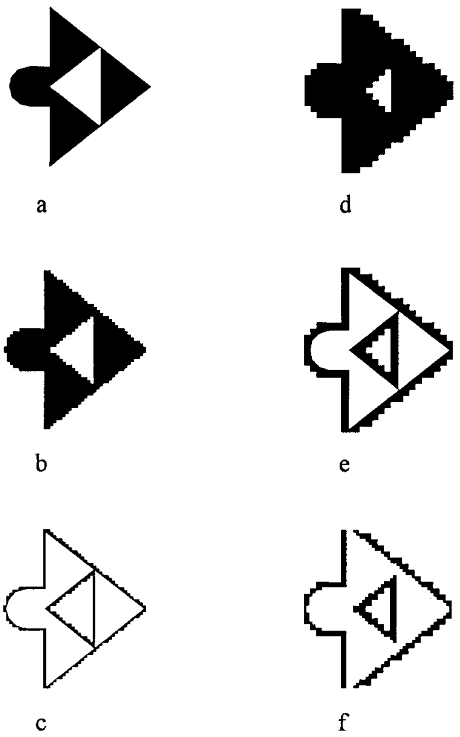


FIG. 17. (a) The enlarged CAD model of the trademark (scale factor: 2). (b) The enlarged display obtained by our method corresponding to level = 6. (c) The display error between our display in (b) and the true one in (a). (d) The enlarged display obtained by the quadtree representation method. (e) The display error between (a) and (d). (f) The display error between (b) and (d).

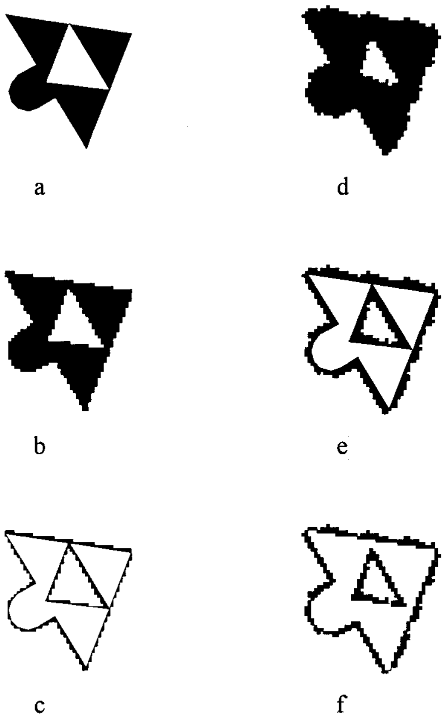


FIG. 18. (a) The true enlarged CAD model of the trademark (a scale change of 2 followed by a counterclock rotation of 30°). (b) The display result obtained by our method. (c) The display error between our display in (b) and the true one in (a). (d) The display result obtained by the quadtree representation method. (e) The display error between (a) and (d). (f) The display error between (b) and (d).

REFERENCES

1. H. Samet, *Applications of Spatial Data Structures.* Addison–Wesley, New York, 1989.
2. H. Samet, *The Design and Analysis of Spatial Data Structures.* Addison–Wesley, New York, 1989.
3. H. Samet, The quadtrees and related hierarchical data structures, *ACM Comput. Surv.* **16,** 1984, 187–260.
4. H. Samet and R. E. Webber, Storing collection of polygons using quadtrees, *ACM Trans. Graphics* **4,** 1985, 182–222.
5. H. Samet and R. E. Webber, On encoding boundaries with quadtrees, *IEEE Pattern Anal. Mach. Intell.* **6,** 1984, 365–369.

6. M. Shneier, Two hierarchical linear feature representations: Edge pyramids and edge quadtrees, *Comput. Graphics Image Process.* **17,** 1981, 211–224.
7. M. Manohar, P. S. Rao, and S. S. Iyengar, Template quadtrees for representing region and line data presented in binary images, *Comput. Vision Graphics Image Process.* **51,** 1990, 338–354.
8. J. O. Omolayole and A. Klinger, A hierarchical data structure scheme for storing pictures, in *Pictorial Information System,* (S. K. Chang and K. S. Fu, Eds.), Springer-Verlag, New York/Berlin, 1980.
9. T. Pavlidis, *Structural Pattern Recognition,* Springer-Verlag, New York, 1977.
10. H. Samet and M. Tamminen, Approximating CSG trees of moving objects, *The Visual Computer,* 1990, 182–209.
11. H. Samet and M. Tamminen, Bintrees, CSG trees, and time, in *Proceedings of the SIGGRAPH '85 Conference, San Francisco, July 1985,* pp. 121–130.
12. B. K. Ray and K. S. Ray, A non-parametric sequential method for polygonal approximation of digital curves, *Pattern Recognition Lett.* **15,** 1994, 161–167.
13. K. Wall and P.-E. Danielsson, A fast sequential method for polygonal approximation of digitized curves, *Comput. Vision Graphics Image Process.* **28,** 1984, 220–227.
14. C. Williams, An efficient algorithm for the piecewise linear approximation of planar curves, *Comput. Graphics Image Process.* **8,** 1978, 286–293.
15. S.-X. Li and M. H. Loew, The quadcode and its arithmetic, *Comm. ACM* **30,** 1987, 621–626.
16. R. V. Shankar and S. Ranka, Hypercube algorithms for operations on quadtrees, *Pattern Recognition* **25,** 1992, 741–747.
17. F. Dehne and R.-C. Andrew, Hypercube algorithms for parallel processing of pointer-based quadtrees, *Comput. Vision Image Understanding* **62,** 1995, 1–10.
18. R. C. Gonzalez and R. E. Woods, *Digital Image Processing,* Addison–Wesley, Reading, MA, 1992.
19. G. K. Wallace, Overview of the JPEG Still Image Compression Standard, in *Visual Communications and Image Processing '89,* SPIE, Philadelphia, 1989.
20. S. G. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Pattern Anal. Mach. Intell.* **11,** 1989, 674–693.

I-PIN CHEN received the B.S. degree from Feng Chia University in 1978 and the M.S. degree from National Chiao Tung University in 1986. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering, National Chiao Tung University. He has been affiliated with the Mechanical Industry Research Laboratories, Industrial Technology and Research Institute, Taiwan since 1980. His research areas include computer graphics, image representation, and automatic inspection.

ZEN CHEN received the B.Sc. degree from National Taiwan University in 1967, the M.Sc. degree from Duke University in 1970, and the Ph.D. degree from Purdue University in 1973, all in electrical engineering. After graduating from Purdue University, he worked for Burroughs Corporation, Detroit, Michigan, where he was engaged in the development of document recognition systems. He joined National Chiao Tung University, Taiwan in 1974 and served as the director of the Institute of Computer Engineering from 1975 to 1981. He spent the academic year 1981–1982 at Lawrence Berkeley Laboratory, University of California, Berkeley, California, as a visiting scientist. He was a visiting professor at the Center for Automation Research, University of Maryland, College Park, Maryland, from August 1989 to January 1990. His current research interests include computer vision, pattern recognition, virtual reality, and parallel algorithms and architectures.

Dr. Chen is a member of Sigma Xi and Phi Kappa Phi. He was the founding president of the Chinese Society of Image Processing and Pattern Recognition in Taiwan. He received the outstanding engineering professor award from the Chinese Institute of Engineers in 1998.