Pattern Recognition Vol. 15, No. 1, pp. 31–40, 1982.
Printed in Great Britain.

0031-3203/82/010031-10 $02.00/0
Pergamon Press Ltd.

SHAPE APPROXIMATION USING QUADTREES

SANJAY RANADE, AZRIEL ROSENFELD* and HANAN SAMET
Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742, U.S.A.

(*Received* 23 *January* 1981; *in revised form* 8 *May* 1981; *received for publication* 1 *June* 1981)

Abstract—The quadtree representation encodes a 2^n by 2^n binary image as a set of maximal blocks of 1's or 0's whose sizes and positions are powers of 2. With the aid of the quadtree, a hierarchy of approximations to the image can be defined. Several ways of doing this are described. The accuracy of these approximations is empirically evaluated by studying how fast estimates of the first few moments of the image, computed from the approximations, converge to the true values, using a database of 112 airplane silhouettes. Approaches to the problem of fast shape matching using these approximations are also discussed.

Image representations   Quadtrees   Shape approximation   Matching   Moments

1. INTRODUCTION

In recent years there has been rapidly growing interest in "quadtree" representations for binary images.[1–15] Given a 2^n by 2^n binary image I, we construct its quadtree as follows: The root node of the tree corresponds to all of I. If I consists of all 0's or all 1's, we label the root node 0 or 1, and it is all of the tree. Otherwise, the root node has four sons corresponding to the four quadrants of I, and we repeat the process for each of these quadrants. When this construction is complete, the leaf nodes of the tree correspond to blocks (= sub … subquadrants of I) consisting entirely of 0's or 1's. A node at level k (where the root is at level n) corresponds to a block of size 2^k by 2^k, in a position whose coordinates are multiples of 2^k. From now on we will call a leaf node "white" or "black" if it is labelled 0 or 1, respectively, and we will refer to nonleaf nodes as "gray". An example of a binary image of an airplane and its quadtree is shown in Fig. 1. In this example we have $n = 6$ (i.e. the binary image is 64 by 64), so that the tree has seven levels (including the root); there are no black leaf nodes at levels 6, 5, 4 or 3.

With the aid of the quadtree, we can define a hierarchy of approximations to the given image I. This can be done in various ways, as discussed in Section 2. To test the accuracy of these approximations, Section 3 presents an empirical investigation of how fast estimates of the first few moments of I, computed from the approximations, converge to their true values, for a set of 112 binary images of airplanes.

Approximations should also be useful for matching purposes, since they should make it possible to reject mismatches rapidly. For shapes that are all similar to one another, however, e.g. for airplanes, the savings inherent in this approach may not be very great; the use of quadtrees for matching binary (or arbitrary) images is discussed in Section 4.

* All correspondence with Dr. Rosenfeld

2. APPROXIMATIONS

Given the quadtree representation of a binary image I, we can define several kinds of approximations to I:

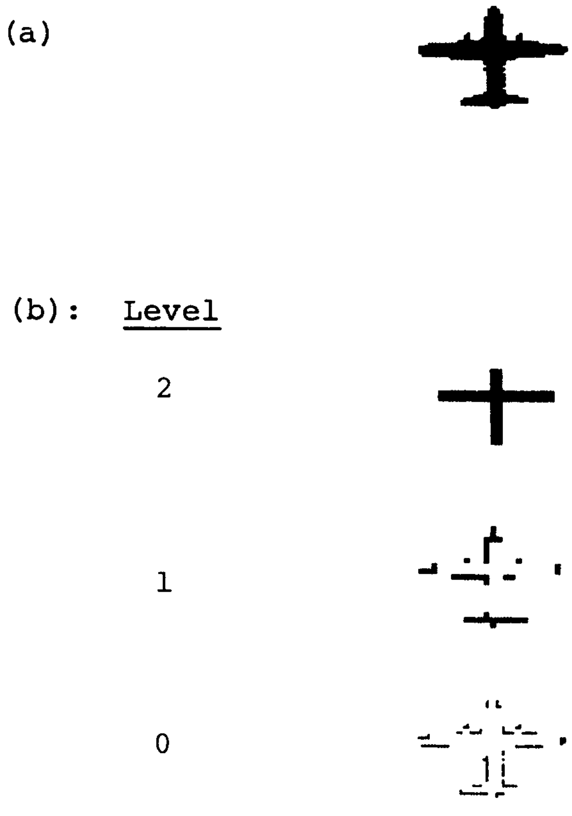

Fig. 1. (a) Binary image of an airplane (64 × 64); (b) Black nodes in the quadtree representation of (a), displayed as black blocks. There are no black nodes at levels 6, 5, 4, 3.

(a) Let $I_{(k)}$, the kth-order *inner approximation* to I, be the binary picture defined by the blocks of 1's corresponding to the black nodes at levels $\geq k$ of I's quadtree. Evidently $I_{(n)} \leq I_{(n-1)} \leq \cdots \leq I_{(0)} = I$, where "$A \leq B$" means that the set of 1's of A is contained in the set of 1's of B.

(b) Let $I^{(k)}$, the kth-order *outer approximation* to I, be defined in the same way as $I_{(k)}$, except that it also contains blocks of 1's corresponding to the *gray* nodes at level k. It is not hard to see that $I^{(n)} \geq I^{(n-1)} \geq \ldots \geq I^{(0)} = I$.

These two series of approximations, for the binary image in Fig. 1, are shown in Fig. 2. Note that unless I consists entirely of 1's, $I_{(n)}$ is empty; and unless I consists entirely of 0's, $I^{(n)}$ is all of I.

The outer approximations to I are actually the complements of the inner approximations to $\bar{I}$ (the complement of I); in other words, $I^{(k)} = \overline{\bar{I}_{(k)}}$ for all k. To see this, let P be any 1 in $\overline{I^{(k)}}$; thus P is 0 in $I_{(k)}$, so that P does not belong to a black node at level $\geq k$ in the quadtree of $\bar{I}$. This is equivalent to saying that P belongs to either a white node at level $\geq k$, or a gray node at level k, in $\bar{I}$'s quadtree; or, equivalently, P belongs to either a black node at level $\geq k$, or a gray node at level k, in the quadtree of I, so that P is in $I^{(k)}$, and conversely.

These approximations are reasonable when the 1's in I define a compact shape, but they may not be so useful for shapes that contain elongated parts, e.g. a "body" and "limbs". In order for $I_{(k)}$ to adequately represent the limbs, k must be relatively small (2^k must be less than the limb width); but approximating the body does not require a small k. We can solve this problem by using approximations based on "maximal" black nodes. A black node will be called maximal if its block is not adjacent to any larger block of 1's. As we shall see in Section 3, these maximal nodes comprise about 5% of the black nodes. More generally, a black node will be called k-maximal if its block is not adjacent to any block of 1's that is at least 2^k times as large.* In terms of this concept we can define two additional types of approximation:

(c) Let $J_{(k)}$ be defined by the blocks of 1's corresponding to the k-maximal black nodes of the quadtree. Evidently $J_{(0)} \leq J_{(1)} \leq \cdots \leq J_{(n)} = I$.

(d) Analogously, let $J^{(k)} = \overline{\bar{J}_{(k)}}$, where $\bar{J}_{(k)}$ is the $J_{(k)}$ approximation to $\bar{I}$. Thus $J^{(0)} \geq J^{(1)} \geq \ldots \geq J^{(n)} = I$.

These approximations are shown, for the image of Fig. 1, in Fig. 3.

Typically, most nodes will be k-maximal for relatively small k, so that $J_{(k)}$ involves nearly all of the nodes; but $J_{(0)}$ is a rather crude approximation to I. A reasonable compromise is to combine $I_{(k)}$ with $J_{(0)}$, or $I^{(k)}$ with $J^{(0)}$ – in other words, to use nodes that are either large or maximal:

* In other words: A black node at level l is k-maximal if any black nodes whose blocks are adjacent to its block are at levels $< l+k$.

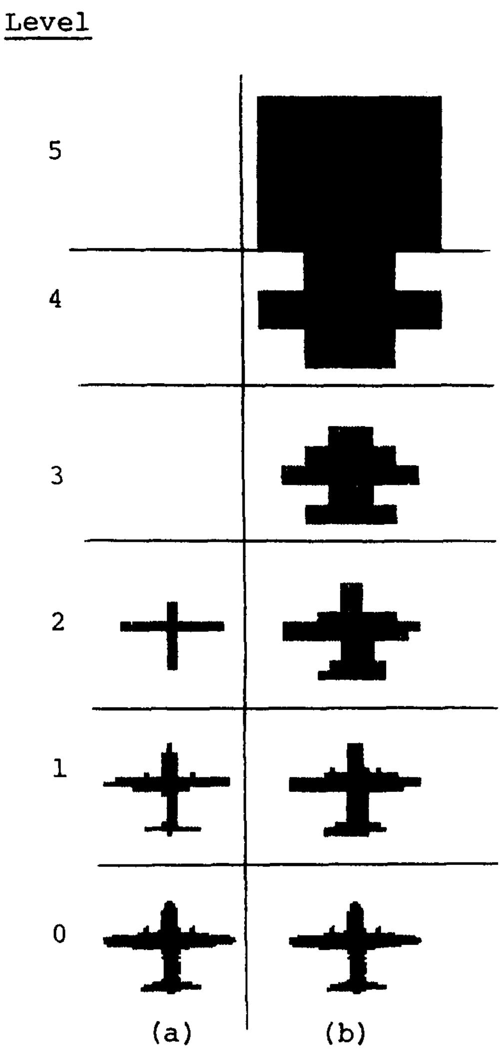

Fig. 2. Approximations to the image in Fig. 1a based on levels in the tree. (a) Black nodes at level $\geq k$, displayed as black blocks; (b) Black nodes at level $\geq k$ and gray nodes at level k, displayed as black blocks Level 6 is identical to level 5.

(e) $I^*_{(k)} = I_{(k)} \vee J_{(0)}$

(f) $I^{(k)*} = I^{(k)} \vee J^{(0)}$

Figure 4 shows these approximations for the image of Fig. 1. In the next section we present some empirical results about the accuracy and usefulness of these approximations for a set of airplane shapes.

3. MOMENT COMPUTATION

Moments are frequently used for pattern description and recognition[16-18]; they provide information about the balance and spread of the gray levels in the pattern relative to given coordinate axes. The (i, j) moment of the picture $f(x, y)$ is defined as

$$m_{ij} \equiv \Sigma\Sigma f(x, y)x^i y^j$$

where the sum is taken over the entire picture. Thus

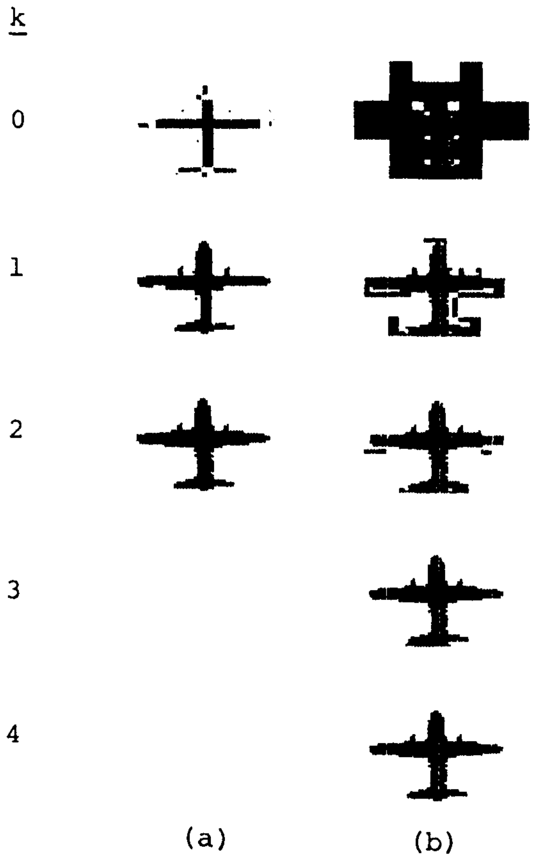

Fig. 3. Approximations based on maximal nodes. (a) k-maximal black nodes, displayed as black blocks, for $k = 0, 1, 2$. Note that for $k = 2$ every node is k-maximal. (b) Complement of k-maximal white nodes, displayed analogously, for $k = 0, 1, 2, 3, 4$.

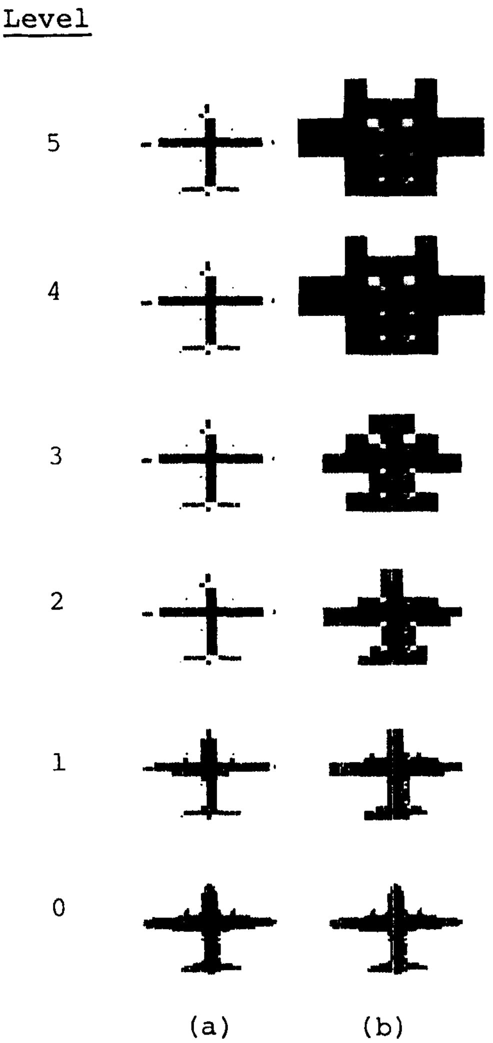

Fig. 4. Approximations based on level, together with 0-maximal nodes. Note that in (a) the results are identical for levels 5, 4, 3, 2, since there are no black nodes at levels 5, 4, 3, and the black nodes at level 2 are all maximal.

m_{00} is simply the sum of the gray levels of f. The *centroid* of f is the point whose coordinates are $(m_{10}/m_{00}, m_{01}/m_{00})$. If we compute moments taking the centroid as origin, they are called *central moments*, and are denoted by $\bar{m}_{ij}$.

When $f \equiv I$ is binary-valued, m_{ij} becomes the sum of $x^i y^j$ for those points (x, y) at which I has value 1. In particular, m_{00} is just the number of 1's in I. Given the quadtree representation of I, we can compute its moments blockwise, since the moments of I are the sums of the moments of its blocks. On moment computation from quadtree representations see Shneier.[15]

We will now test the accuracy of our approximations to I by using them to estimate some of the moments of I. In particular, we investigate how accurately we can estimate the area of $I(m_{00})$, the coordinates of its centroid (m_{10}/m_{00} and m_{01}/m_{00}), and its central moments ($\bar{m}_{20}$ and $\bar{m}_{02}$).

Table 1 shows approximations *a, b, e* and *f* to these moments for the airplane shape of Fig: 1. (Approximations *c* and *d* are not shown, since (*c*) converges so fast, as we saw in Fig. 3.) For each pair of approximations, (*a*–*b*) and (*e*–*f*), we also show the estimates obtained by averaging the "inner" and "outer" approximations of each order. Note that the order-6 "approximations" are the true values. We see that the approximations to the coordinates of the centroid are quite good even at the level where black leaf nodes first appear; in most cases the errors are only fractions of a pixel. It seems reasonable to predict that similar results would hold for larger images; when we use the quadtree levels at which blocks are, say, 4 by 4 pixels or larger, the errors should be only fractions of a pixel.

Similar approximations were computed for a set of 112 airplane shapes shown in Fig. 5. (Figure 1 is the shape in the sixth row, first column.) Table 2 shows the mean error and standard deviation of the errors for each approximation. We see that the average errors in the centroid coordinates are consistently low even at the levels where black leaves first appear. Approximation (*e*) is especially good. Note that the shapes used are binary silhouettes, not grayscale images; quadtree approximations to grayscale images will not be considered in this paper.

Table 1. Approximations to the moments of the airplane in Fig. 1(a)

				Centroid		Second moments	
Approximation	Order	No. of nodes	Area (m_{00})	m_{10}/m_{00}	m_{01}/m_{00}	m_{20}	m_{02}
a							
	5	—	—	—	—	—	—
	4	—	—	—	—	—	—
	3	—	—	—	—	—	—
	2	15	240	35.63	33.50	65.2	35.1
	1	53	392	34.64	32.38	73.8	70.5
	0	155	494	34.17	32.36	80.7	76.9
b							
	5	4	4096	31.50	31.50	341.2	341.2
	4	8	2048	39.50	31.50	213.2	149.2
	3	18	1152	34.61	31.50	106.6	111.1
	2	53	848	33.65	31.76	98.2	90.6
	1	106	604	33.67	32.35	85.7	84.4
	0	155	494	34.17	32.36	80.7	76.9
$\frac{a+b}{2}$							
	5		2048	—	—	—	—
	4		1024	—	—	—	—
	3		576	—	—	—	—
	2		544	34.64	32.63	81.7	62.8
	1		498	34.15	32.36	79.7	77.4
	0		494	34.17	32.36	80.7	76.9
e							
	5	35	302	34.16	32.93	72.8	72.2
	4	35	302	34.16	32.93	72.8	72.2
	3	35	302	34.16	32.93	72.8	72.2
	2	35	302	34.16	32.93	72.8	72.2
	1	59	398	34.64	32.46	75.8	70.6
	0	155	494	34.17	32.36	80.7	76.9
f							
	5	26	1863	38.15	31.48	231.9	131.9
	4	30	1863	38.15	31.48	231.9	131.9
	3	40	1095	34.39	31.46	110.4	111.7
	2	75	823	33.76	31.72	100.4	91.3
	1	128	599	33.70	32.32	86.3	84.5
	0	177	494	34.17	32.36	80.7	76.9
$\frac{e+f}{2}$							
	5		1082.5	36.15	32.20	152.4	102.0
	4		1082.5	36.15	32.20	152.4	102.0
	3		698.5	34.27	32.19	91.6	92.0
	2		562.5	33.95	32.32	86.6	81.8
	1		498.5	34.17	32.38	81.0	77.5
	0		494	34.17	32.36	80.7	76.9

4. COARSE-FINE MATCHING

In order to reduce the computational cost of image matching, a number of "coarse-fine" matching schemes have been proposed, in which some type of low-resolution matching is used to rapidly eliminate definite mismatches, so that full resolution matching need only be performed in the remaining cases.[19-21] In this section we discuss the applicability of quadtree approximations to coarse-fine matching.

We will consider two types of matching problems: (a) finding a known pattern in an unknown position; (b) identifying a pattern, in a given position, as being one of a given set of patterns. We will refer to these as the "location" and "identification" problems, respectively.

4.1. *Location*

The quadtree representation is not especially appropriate for the location problem, since the quadtree changes as the input pattern is shifted. For example, Fig. 6 shows the quadtrees for the airplane in Fig. 1 when it is shifted by (1,0), (0,1) and (1,1). It should be pointed out that shifts by odd amounts cause the greatest changes in the tree; a shift whose components are high powers of 2 may cause very little change. Thus the quadtree is quite sensitive to small shifts, as Fig. 6 illustrates; note in particular level 1.

Shifts can cause changes even at high levels of the tree; if we shift an isolated 2^k by 2^k block of 1's by (1, 1), it breaks up into a large number of smaller blocks. Note, however, that one of these is 2^{k-1} by 2^{k-1}; in general,

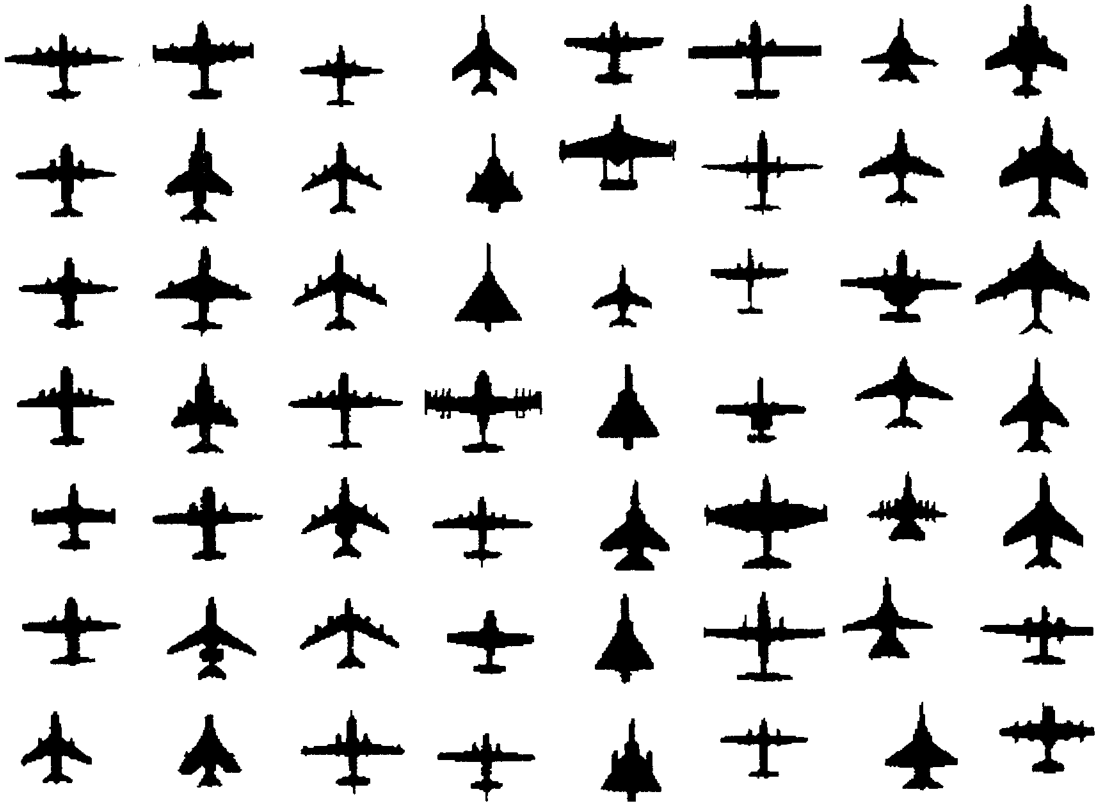

Fig. 5. 112 airplane shapes.

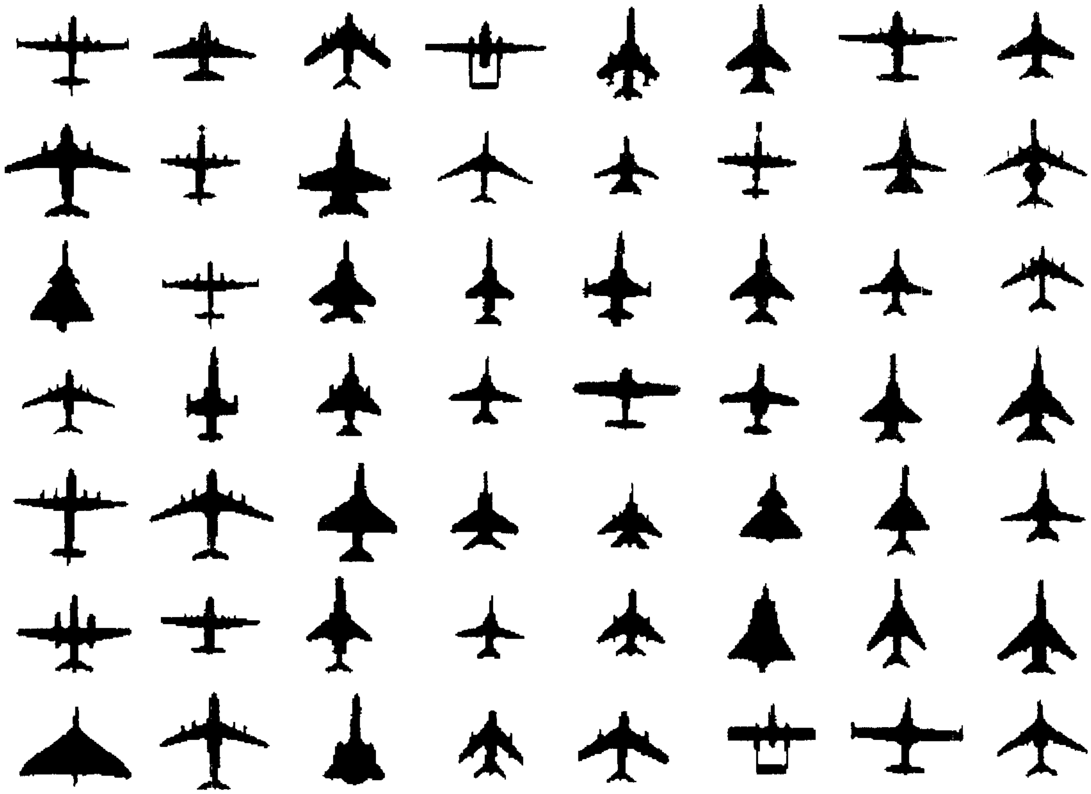

Fig. 5., continued

Table 2a. Means of the errors in approximating the moments of the 112 airplanes in Fig. 5

Approximation	Order	Area (m_{00})	Centroid m_{10}/m_{00}	Centroid m_{01}/m_{00}	Second moments m_{20}	Second moments m_{02}
a						
	5	504.1	—	—	—	—
	4	504.1	—	—	—	—
	3	457.4	2.22	2.01	42.5	91.4
	2	284.6	1.70	0.78	27.8	48.2
	1	104.1	0.37	0.20	6.8	12.8
	0	0	0	0	0	0
b						
	5	3563.2	3.38	1.66	272.7	253.9
	4	1806.1	2.25	1.25	118.9	156.1
	3	752.4	1.22	1.70	45.3	61.1
	2	319.8	0.66	0.31	18.2	27.3
	1	106.0	0.27	0.14	6.1	10.6
	0	0	0	0	0	0
$\frac{a+b}{2}$						
	5	1529.1	—	—	—	—
	4	650.6	—	—	—	—
	3	143.6	10.35	11.14	15.4	17.0
	2	27.3	0.95	0.53	8.5	13.5
	1	3.9	0.15	0.11	1.5	2.8
	0	0	0	0	0	0
e						
	5	223.6	0.59	0.34	6.5	11.5
	4	223.6	0.59	0.34	6.5	11.5
	3	223.6	0.59	0.34	6.5	11.5
	2	182.6	0.43	0.29	6.7	6.7
	1	92.5	0.34	0.18	5.5	8.6
	0	0	0	0	0	0
f						
	5	1474.5	2.39	1.0	123.4	136.0
	4	1467.5	2.40	0.97	121.0	136.9
	3	679.1	1.27	0.67	46.4	61.0
	2	301.5	0.64	0.31	19.2	28.0
	1	101.8	0.27	0.14	6.4	10.6
	0	0	0	0	0	0
$\frac{e+f}{2}$						
	5	625.4	1.23	0.53	59.9	70.9
	4	621.9	1.24	0.53	58.7	71.4
	3	227.8	0.75	0.35	21.4	33.4
	2	59.9	0.37	0.20	7.3	12.1
	1	5.8	0.12	0.07	1.0	1.8
	0	0	0	0	0	0

when we shift the pattern, a node corresponding to a 2^k by 2^k block always gives rise to at least one node corresponding to a 2^{k-1} by 2^{k-1} (or larger) block. If a node corresponds to a non-isolated block, after shifting it may contribute to a block of much larger size; but if the given block is maximal, it is not hard to see that it cannot contribute (after shifting) to a block more than one size larger. Thus shifting does preserve some sort of crude correspondence, particularly between maximal nodes. Note, however, that when we shift a maximal node, the "corresponding" node may no longer be maximal.

The foregoing remarks suggest the following quadtree-based approach to the location problem: Given the quadtrees Q_1 and Q_2 of the shifted and unshifted patterns, consider all pairs composed of a maximal node of Q_1, say at level k, and a node of Q_2 at level $k-1$, k or $k+1$. Each of these pairs defines a possible shift, or rather a range of possible shifts. For each such shift, we can compute a match score in terms of the numbers and sizes of node pairs that support it. In the resulting "correlogram", we may hope to detect a peak representing the actual shift. Fine matching in the vicinity of this estimated shift could then be used to locate the pattern exactly.

In practice, this approach seems to be reasonably effective. Fig. 7 shows the "correlogram" for the airplane in Fig. 1, unshifted and shifted by (1, 1). There is a peak corresponding to the correct shift, though many other shifts are also given high scores.

A more robust approach to the location problem is to use the quadtree of the shifted pattern to compute an

Table 2b. Standard deviations of the errors in approximating the moments of the 112 airplanes in Fig. 5

Approximation	Order	Area (m_{00})	Centroid		Second moments	
			m_{10}/m_{00}	m_{01}/m_{00}	m_{20}	m_{02}
a						
	5	110.9	—	—	—	—
	4	110.9	—	—	—	—
	3	78.8	1.53	2.41	28.2	29.8
	2	56.8	1.21	1.14	21.4	23.8
	1	18.8	0.30	0.20	4.3	6.8
	0	0	0	0	0	0
b						
	5	205.9	2.56	1.46	32.4	26.9
	4	391.3	1.77	0.99	40.6	46.3
	3	151.3	0.88	0.51	24.9	29.4
	2	54.3	0.47	0.28	8.8	13.5
	1	18.3	0.18	0.14	3.1	5.4
	0	0	0	0	0	0
$\frac{a+b}{2}$						
	5	142.6	—	—	—	—
	4	198.3	—	—	—	—
	3	67.4	6.4	7.0	14.7	12.4
	2	18.4	1.8	1.3	8.1	9.7
	1	3.2	0.1	0.1	1.5	3.1
	0	0	0	0	0	0
e						
	5	77.3	0.53	0.28	6.2	13.5
	4	77.3	0.53	0.28	6.2	13.5
	3	77.3	0.53	0.28	6.2	13.5
	2	42.6	0.33	0.23	5.0	5.6
	1	16.6	0.22	0.16	3.2	5.8
	0	0	0	0	0	0
f						
	5	228.8	1.97	0.80	38.1	35.7
	4	235.2	1.98	0.75	37.4	35.5
	3	122.8	0.97	0.48	21.9	24.7
	2	49.4	0.47	0.27	8.8	13.2
	1	16.9	0.19	0.14	3.2	5.4
	0	0	0	0	0	0
$\frac{e+f}{2}$						
	5	129.1	1.03	0.39	20.2	20.9
	4	133.4	1.03	0.38	19.9	21.3
	3	70.6	0.53	0.28	11.1	17.1
	2	25.5	0.28	0.17	5.0	7.2
	1	4.6	0.10	0.05	1.0	2.0
	0	0	0	0	0	0

approximation to the centroid, as in Section 3; the position of this approximation relative to the centroid of the unshifted pattern then approximately defines the shift. Based on the results of the previous section, even at the early stages the centroids are all correctly located to within a fraction of a pixel, so that the shift can be determined to within a fraction of a pixel by examining the quadtree levels corresponding to blocks of pixels that are, say, 4 by 4 or larger.

4.2. *Identification*

We now consider the problem of identifying an unknown pattern as being one of a given set of patterns. The following quadtree-based approach suggests itself: Let I' and I'' be two of the reference patterns, and let I be the unknown pattern. At any level of approximation, we determine bounds on the discrepancy between I and I' (or I''). If the lower bound on one of these discrepancies, say of I with I', becomes larger than the upper bound on the (I, I'') discrepancy, we can reject I', since it cannot be as good a match to I as I'', and so cannot be the correct match.

The discrepancy between two binary images is the number of points at which their values differ. We can compute bounds on this discrepancy, based on the inner and outer approximations at a given quadtree level k, as follows: The points in $I_{(k)} \wedge \overline{I'^{(k)}}$ are 1 in I and 0 in I', and the reverse is true for the points in $I'_{(k)} \wedge \overline{I^{(k)}}$; thus the number of 1's in the OR of these is a lower bound on discrepancy. On the other hand, we do not know whether the points in $I^{(k)} - I_{(k)}$ are 1 or 0 in I, and similarly for $I'^{(k)} - I'_{(k)}$ in I', so that (in the worst

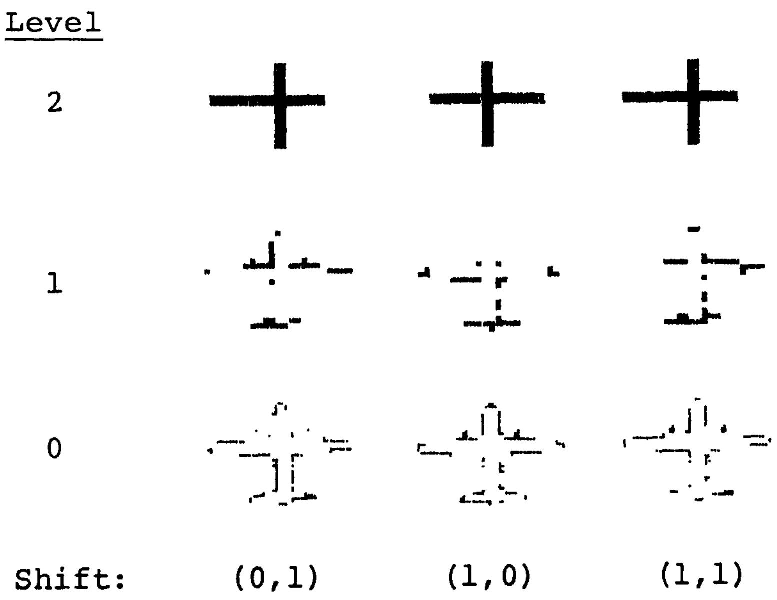

Fig. 6. Analogous to Fig. 1(b) for three shifted versions of the airplane image in Fig. 1(a).

case) all of these points may contribute to the discrepancy (or nearly all; when a 2^m by 2^m gray node of I corresponds to a black leaf in I', for example, the discrepancy cannot be more than $4^m - 1$, since the gray node cannot be all white). Thus we get an upper bound on the discrepancy by adding the number of 1's in $(I^{(k)} - I_{(k)}) \vee (I'^{(k)} - I'_{(k)})$ to the lower bound.

This method does provide some capability for eliminating mismatches without going all the way down to the pixel level. As an example, Fig. 8 shows the quadtrees for two of the airplane shapes and the successive bounds on the discrepancy when they are matched with themselves and with one another, level by level. At level 5, the lower bound for the mismatch of

48	16	85	0	54	0	18	0	37	0	192	0	130	0	98	0	74	74	38	40	0	0
16	0	32	0	16	0	34	0	0	0	68	0	66	0	48	42	36	74	38	40	0	0
51	0	85	0	68	0	54	0	82	0	504	0	136	0	144	0	74	39	82	43	0	0
18	0	19	0	19	0	0	0	0	0	0	0	0	0	0	42	38	82	86	45	47	0
36	0	76	0	38	0	19	0	90	0	410	0	73	0	147	0	78,	42	45	47	0	0
14	15	15	0	0	0	0	0	44	0	0	0	77	0	0	42	74	80	0	0	0	0
50	11	66	0	40	0	50	0	258	0	696	0	237	0	141	0	66	72	78	0	42	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	0	152	0	102	0	112	0	450	75	700	0	450	0	200	0	108	0	138	0	144	0
24	30	35	0	35	0	34	52	138	285	246	236	222	180	70	62	56	120	,64	66	0	0
108	45	208	0	216	0	324	0	728	119	2072	142	675	113	651	85	316	82	504	86	174	0
0	0	0	0	0	0	0	0	94	120	146	0	143	0	0	0	0	0	0	0	0	0
39	0	92	0	0	0	0	0	184	116	280	0	0	0	0	0	0	0	0	0	0	0
66	72	72	36	35	40	46	61	76	98	240	125	130	0	0	0	82	86	0	0	0	0
75	52	78	26	66	28	108	50	192	84	624	110	116	0	88	0	66	69	72	0	67	0
20	21	23	26	24	28	33	45	57	78	99	0	103	0	0	0	0	0	0	0	0	0
40	22	69	26	24	26	30	40	100	72	188	0	90	0	53	0	43	0	44	0	42	0
21	24	0	0	0	0	0	0	0	0	90	0	84	67	0	0	44	46	0	0	0	0
57	22	69	0	72	0	78	0	150	0	430	0	78	63	49	0	43	46	48	0	47	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	44	0	23	0	25	0	98	0	160	0	68	0	45	0	41	0	47	0	46	0
18	20	23	0	24	0	25	0	49	0	0	0	64	0	42	42	41	42	0	0	0	0

Fig. 7. "Correlogram" for the airplane in Fig. 1, based on maximal nodes at shift (0, 0) and at shift (1, 1).

(a) (b)

8a to 8a

L	LB	UB
1	0	4096
2	0	2048
3	0	1152
4	0	608
5	0	212
6	0	0

8b to 8b

L	LB	UB
1	0	4096
2	0	2304
3	0	1216
4	0	592
5	0	200
6	0	0

8a to 8b

L	LB	UB
1	0	4096
2	0	2816
3	0	1600
4	352	1392
5	504	900
6	601	601

(c)

Fig. 8. Result of matching the airplane in Fig. 1(a) with itself and another airplane (row 7, column 1 in Fig. 5). [L = level, LB = lower bound, UB = upper bound.] Note that at level 5, the lower bound on the mismatch (8a, 8b) exceeds the upper bounds on (8a, 8a) and (8b, 8b), so that matching 8a with 8b can be ruled out.

the two shapes with one another exceeds the upper bound for their mismatches with themselves, so that the unequal pair can be rejected.

CONCLUDING REMARKS

Quadtrees can be used to define various types of approximation to a binary image. From these approximations it should be possible to estimate properties of the image, such as its moments, with reasonable accuracy using only a fraction of the quadtree nodes; at least this proved to be true for the airplane shapes studied here. We should also be able to estimate the position of a shifted image, either directly or via its centroid coordinates. On the other hand, the approximations do not seem to be very useful for quickly identifying one out of a set of images unless the images differ greatly from one another. Of course, these tentative conclusions are based on studies with a specialized class of shapes (airplane silhouettes); they should be checked using shapes of other types. However, they are at least suggestive that quadtree approximations to binary images should be of some use in shape property computation and matching.

Acknowledgement—The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Eleanor Waters in preparing this paper.

REFERENCES

1. A. Klinger, Data structures and pattern recognition, Proc. 1IJCPR, 497–498 (1973).
2. A. Klinger and C. R. Dyer, Experiments in picture representation using regular decomposition, *Computer Graphics and Image Processing* **5**, 68–105 (1976).
3. N. Alexandridis and A. Klinger, Picture decomposition, tree data-structures and identifying directional symmetries as node combinations, *Computer Graphics and Image Processing* **8**, 43–77 (1978).

4. A. Klinger and M. L. Rhodes, Organization and access of image data by areas, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1**, 50–60 (1979).
5. G. M. Hunter, Efficient computation and data structures for graphics, Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Princeton University, NJ (1978).
6. G. M. Hunter and K. Steiglitz, Operations on images using quadtrees, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1**, 145–153 (1979).
7. G. M. Hunter and K. Steiglitz, Linear transformation of pictures represented by quadtrees, *Computer Graphics and Image Processing* **10**, 289–296 (1979).
8. C. R. Dyer, A. Rosenfeld and H. Samet, Region representation: boundary codes from quadtrees, *Communications of the ACM* **23**, 171–179 (1980).
9. H. Samet, Region representation: quadtrees from boundary codes, *Communications of the ACM* **23**, 163–170 (1980).
10. H. Samet, Computing perimeters of images represented by quadtrees, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.
11. H. Samet, Connected component labeling using quadtrees, *J. ACM*, to appear.
12. H. Samet, Region representation: raster-to-quadtree conversion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.
13. H. Samet, Region representation: quadtrees from binary arrays, *Computer Graphics and Image Processing* **13**, 88–93 (1980).
14. C. R. Dyer, Computing the Euler number of an image of its quadtree, *Computer Graphics and Image Processing* **13**, 270–276 (1980).
15. M. Shneier, Linear-time calculations of geometric properties using quadtrees, *Computer Graphics and Image Processing*, to appear.
16. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, NY, Sections 10.1.3, 10.2.2 (1976).
17. R. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, MA, Section 7.2.2 (1977).
18. W. K. Pratt, *Digital Image Processing*, Wiley, NY, Section 18.4.3 (1978).
19. A. Rosenfeld and G. J. VanderBrug, Coarse-fine template matching, *IEEE Transactions on Systems, Man, and Cybernetics* **7**, 104–107 (1977).
20. R. Y. Wong and E. L. Hall, Sequential hierarchical scene matching, *IEEE Transactions on Computers* **27**, 359–366 (1978).
21. R. Y. Wong and E. L. Hall, Scene matching with invariant moments, *Computer Graphics and Image Processing* **8**, 16–24 (1978).

About the Author—SANJAY RANADE received the BSc degree in Aeronautics, and the MSc and PhD degrees in Computer Science from the University of Southampton, England, in 1972, 1974 and 1977 respectively. In 1976 he was awarded a Latin American Teaching Fellowship from Tufts University at the Instituto Technologico de Estudios Superiores de Monterrey, Mexico. Since 1978 he has been engaged in research work in image processing at the University of Maryland and at the National Institutes of Health, Bethesda, Maryland.

About the Author—AZRIEL ROSENFELD received the Ph.D. in mathematics from Columbia University in 1957. After 10 years in the defense electronics industry, in 1964 he joined the University of Maryland, where he is Research Professor of Computer Science. He edits the journal *Computer Graphics and Image Processing*, and is president of the consulting firm ImTech, Inc. He has published 13 books and over 250 papers, most of them dealing with the computer analysis of pictorial information. He is currently President of the International Association for Pattern Recognition.

About the Author—HANAN SAMET received the B.S. degree in engineering from the University of California, Los Angeles, and the M.S. degree in operations research and the M.S. and Ph.D. degrees in computer science from Stanford University, Stanford, CA.

In 1975 he joined the University of Maryland as an Assistant Professor of Computer Science. In 1980 he became an Associate Professor. His research interests are data structures, image processing, programming languages, artificial intelligence, and data base management systems.