Image Processing and Computer Vision

Robert Haralick Editor

The Quadcode and Its Arithmetic

SHU-XIANG LI and MURRAY H. LOEW

ABSTRACT: *The quadcode is a hierarchical data structure for describing digital images. It has the following properties: (1) straightforward representation of dimension, size, and the relationship between an image and its subsets; (2) explicit description of geometric properties, such as location, distance, and adjacency; and (3) ease of conversion from and to raster representation. The quadcode has applications to computer graphics and image processing because of its ability to focus on selected subsets of the data and to allow utilization of multiple resolutions in different parts of the image. A related approach is the quadtree. Samet recently presented a thorough survey of the literature in that field [7]. Gargantini [2] and Abel and Smith [1] presented linear quadtrees and linear locational keys that are efficient labeling techniques for quadtrees. In those papers the geometric concepts of the image are discussed by using the tree as an interpretive medium, and the approaches and procedures are based on traversal of the nodes in the tree. In this paper we present the quadcode system, which is a direct description of the image, and discuss the geometric concepts in terms of the coded images themselves.*

1. QUADCODE

The quadcode is a quaternary (base-4) code. A quadcode of length n is of the form

$$Q = q_1 q_2, \ldots, q_n,$$

where $q_i = 0, 1, 2, 3$ for $i = 1, 2, \ldots, n$.

When the quadcode is used in describing an image, each character represents one operation of subdividing the image or its subimage into quadrants, as shown and labeled in Figure 1.

In many applications (e.g., smoothing, edge detection, shape description) an image is usually subdivided into much smaller units, so the subdividing operation can be repeated recursively many times, until there is no further subdividing needed. Figure 2, on next page, shows the subdividing process and the corresponding quadcodes. A particular quadrant is represented by one of 0, 1, 2, or 3, concatenated to the quadcode of its predecessor, and after each subdividing operation, the length of the quadcode increases by one. As shown in the figure, the quadcode length signifies how many subdividing operations have been done.

This research was supported in part by the Lady Davis Fellowship Trust.

1.1 Properties of the Quadcode

Property 1 (*qc length*). The quadcode length of individual pixels in a 2^n-by-2^n image is n.

Property 2 (*dimension*). The side length of a subimage with quadcode length m in a 2^n-by-2^n image is 2^{n-m}.

Property 3 (*area*). The area of a subimage with quadcode length m in an image of area A_0 is $4^{-m}A_0$.

Property 4 (*subdividing*). Adding one character to a given quadcode subdivides the given subimage into its quadrants (see Figure 3, on next page).

Property 5 (*merging*). If four quadcodes of length i have the first $i - 1$ positions the same, they represent the four quadrants of the same subimage, which is represented by the $(i - 1)$-position quadcode (Figure 3).

Property 6 (*locating*). A subimage can be located in an image according to each position of its quadcode, which is equivalent to subdivision of the initial image (see Figure 4, p. 623).

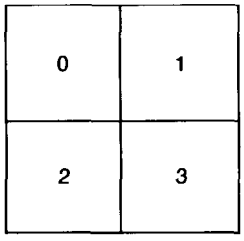

FIGURE 1. Quadcodes and Quadrants

Property 7 (*conversion*). Write the quadcode and array labels i and j of a pixel in a 2^n-by-2^n image (i and j are supposed to be from 0 to $2^n - 1$) into binary form; it can be proved [3] that

$$q_1 q_2, \ldots, q_n = i_1 j_1 i_2 j_2, \ldots, i_n j_n \quad (1)$$

or

$$\left.\begin{aligned} i &= \sum_{k=1}^{n} (q_k \text{ DIV } 2) \times 2^{n-k} \\ j &= \sum_{k=1}^{n} (q_k \text{ MOD } 2) \times 2^{n-k} \end{aligned}\right\} \quad (2)$$

$$Q = \sum_{k=1}^{n} (2i_k + j_k) \times 4^{n-k}. \quad (3)$$

In Eqs. (1)–(3), q_k, i_k, and j_k ($k = 1, \ldots, n$) represent the kth position of the quadcode and the array labels i and j, respectively. The equations show that the array labels i and j are the same as the two binary numbers that compose the odd and even bits of the quadcode when written in binary. This is represented in Table I where the two axes are array labels (or coordinates) i and j; the entries give the corresponding quadcodes in both binary and quaternary forms.

TABLE I. Quadcodes in Binary ($i_1 j_1 i_2 j_2$) and Quaternary ($q_1 q_2$) Forms

$i_1 i_2$ \ $j_1 j_2$	00	01	10	11
00	0000 00	0001 01	0100 10	0101 11
01	0010 02	0011 03	0110 12	0111 13
10	1000 20	1001 21	1100 30	1101 31
11	1010 22	1011 23	1110 32	1111 33

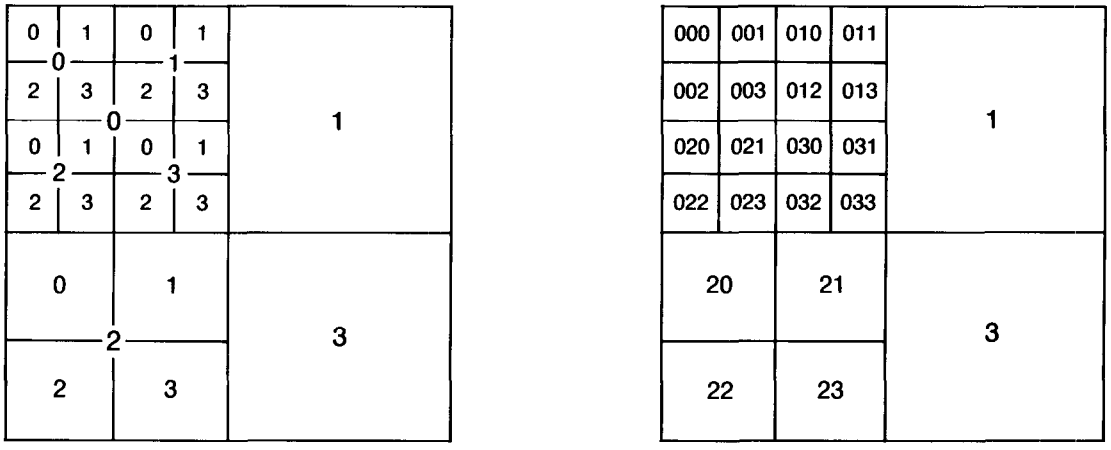

FIGURE 2. Subdividing and Quadcode Representation

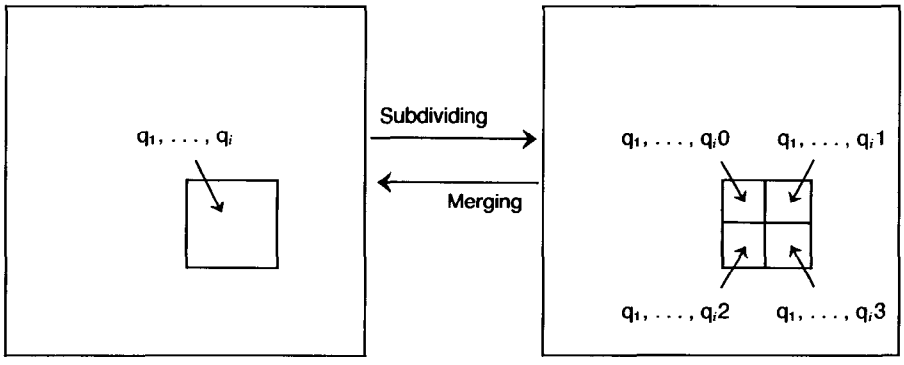

FIGURE 3. Subdividing and Merging

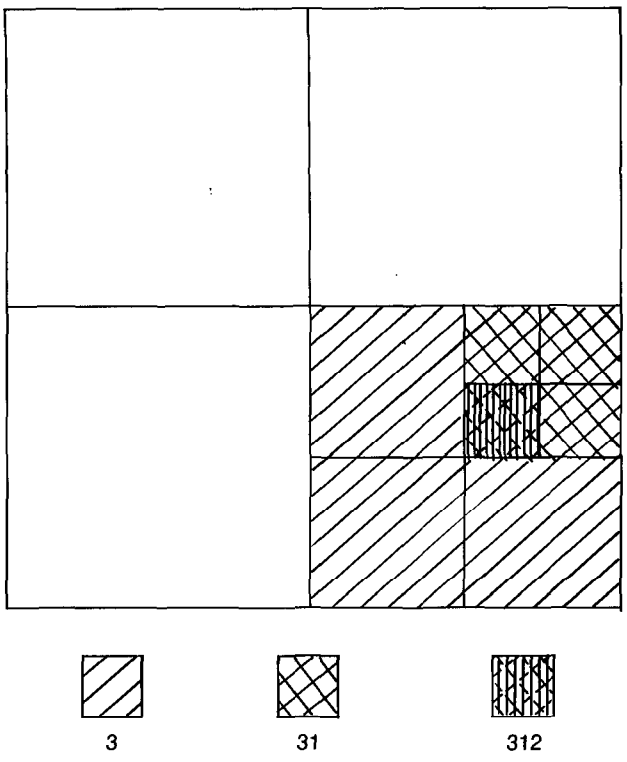

FIGURE 4. Locating a Subimage Using Quadcode

2. ANALYSIS OF ELEMENTARY SQUARES

2.1 Elementary Squares

When a quadcode can represent a subset of an image, we call the subset an elementary square. For example, in Figure 5 there are two elementary squares, and their quadcodes are A = 12 and B = 221. In general, an elementary square in a 2^n-by-2^n image can be represented by $q_1, \cdots, q_m$, where $m \leq n$.

The size of an elementary square with quadcode length m is 2^{n-m} by 2^{n-m}. For instance, in Figure 5 the size of elementary square A is 2^{3-2} by 2^{3-2} (i.e., 2 by 2), and the size of B is 2^{3-3} by 2^{3-3} (i.e., 1 by 1).

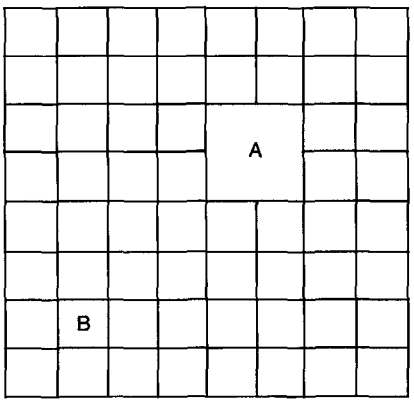

FIGURE 5. Location and Separation

2.2 Location

If we know the size of a given elementary square, we can locate it by the coordinates of one of the characteristic points in the square. We choose the upper left corner point as the characteristic point. Then, from eq. (2), we derive the following:

$$\left.\begin{aligned} x &= \sum_{k=1}^{m} (q_k \text{ MOD } 2) \times 2^{n-k} \\ y &= \sum_{k=1}^{m} (q_k \text{ DIV } 2) \times 2^{n-k}. \end{aligned}\right\} \quad (4)$$

For example, the locations of the squares A = 12 and B = 221 in Figure 5 are

$$x(\mathrm{A}) = \sum_{k=1}^{2} (q_{Ak} \text{ MOD } 2) \times 2^{3-k} = 4$$

$$y(\mathrm{A}) = \sum_{k=1}^{2} (q_{Ak} \text{ DIV } 2) \times 2^{3-k} = 2$$

and

$$x(\mathrm{B}) = \sum_{k=1}^{3} (q_{Bk} \text{ MOD } 2) \times 2^{3-k} = 1$$

$$y(\mathrm{B}) = \sum_{k=1}^{3} (q_{Bk} \text{ DIV } 2) \times 2^{3-k} = 6.$$

2.3 Separation

The separation between two elementary squares A and B is composed of the smallest distances between any points on the boundaries of the two elementary squares in both the horizontal and vertical directions, and they are calculated by

$$\left.\begin{aligned} D_x(\mathrm{A}, \mathrm{B}) &= |x(\mathrm{B}) - x(\mathrm{A})| - S_L \\ D_y(\mathrm{A}, \mathrm{B}) &= |y(\mathrm{B}) - y(\mathrm{A})| - S_U. \end{aligned}\right\} \quad (5)$$

where S_L is the horizontal side length of the leftmost of the two elementary squares A and B, and S_U is the vertical side length of the uppermost of the elementary square. Because the quadcodes of A and B give their dimensions and their horizontal and vertical locations, both S_L and S_U can be obtained from their quadcodes.

For example, in Figure 5, A = 12 and B = 221; the leftmost elementary square is B, and the uppermost is A. Then,

$$x(\mathrm{A}) = 4 \qquad y(\mathrm{A}) = 2$$

$$x(\mathrm{B}) = 1 \qquad y(\mathrm{B}) = 6$$

$$S_L = S_B = 2^{3-3} = 1 \qquad S_U = S_A = 2^{3-2} = 2.$$

Hence, the distance between A and B is

$$D_x(\mathrm{A}, \mathrm{B}) = |1 - 4| - 1 = 2$$

$$D_x(\mathrm{A}, \mathrm{B}) = |6 - 2| - 2 = 2$$

as shown in Figure 5.

2.4 Adjacency

The detection of connectivity is a basic and important subject in image analysis.

THEOREM 1. *The general condition of adjacency for elementary squares is the following:*

$$D_x(A, B) \times D_y(A, B) = 0$$
$$D_x(A, B) + D_y(A, B) \leq 0 \qquad (6)$$

PROOF. If $D_x(A, B) = 0$, then $D_y(A, B) \leq 0$.

According to the first part of eq. (5), $D_x(A, B) = 0$ means that the right side of the far left square is collinear with the left side of the far right square. And, according to the second part of (5), $D_y(A, B) \leq 0$ means that either the upper left corner of the far right square is on the right side of the far left square (see "a" of Figure 6), or the upper right corner of the far left square is on the left side of the far right square (see "b" of Figure 6). If $D_y(A, B) = 0$, then $D_x(A, B) \leq 0$, and the proof is the same. □

In Figure 7 there are three elementary squares A = 1, B = 221, and C = 310. According to eq. (5), the separations among them are

$$D_x(A, B) = 2 \quad D_x(A, C) = -2 \quad D_x(B, C) = 4$$
$$D_y(A, B) = 2 \quad D_y(A, C) = 0 \quad D_y(B, C) = 1,$$

so

$$D_x(A, B) \times D_y(A, B) = 4, \quad D_x(B, C) \times D_y(B, C) = 4$$

and

$$D_x(A, C) \times D_y(A, C) = 0, \quad D_x(A, C) + D_y(A, C) = -2.$$

In view of eq. (6), we observe that elementary squares A and C are adjacent, but A and B and B and C are not, as shown in Figure 7.

Equation (6) is the general adjacency condition for elementary squares of any size. For the connectivity detection of pixels, $S_L = S_U = 1$, the separation equation (eq. (5)) becomes

$$D_x(p_1, p_2) = |x(p_2) - x(p_1)| - 1$$
$$D_y(p_1, p_2) = |y(p_2) - y(p_1)| - 1 \qquad (7)$$

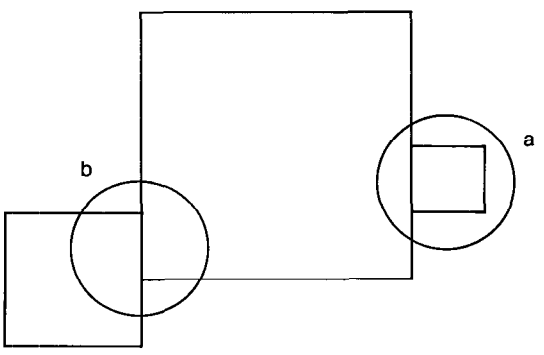

FIGURE 6. Examples for the Definition of Separation as Expressed in Equation (5)

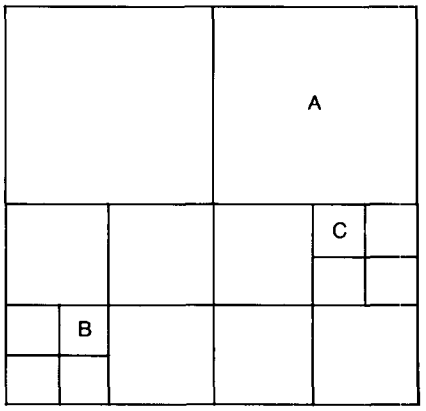

FIGURE 7. Examples for the Definition of Adjacency as Expressed in Equation (6)

and adjacency condition (eq. (6)) becomes

$$\max(|x(p_2) - x(p_1)|, |y(p_2) - y(p_1)|) = 1$$
or
$$|x(p_2) - x(p_1)| + |y(p_2) - y(p_1)| = 1. \qquad (8)$$

Equations (5) and (6) can be extended to detect the connectivity of elongated regions, if we know their locations (still represented by the coordinates of their upper left corner points) and the lengths of their sides.

3. REGION REPRESENTATION AND ANALYSIS

3.1 Region Representation

In quadcode representation, a region is represented as a set of quadcodes

$$R = \{Q_n\} \quad n = 1, \ldots, N, \qquad (9)$$

where each element represents an elementary square in region R, and N is the total number of those elementary squares in region R.

For example, the region in Figure 8 can be represented as

R = {03, 120, 121, 21, 230, 231, 300, 301, 320, 321}.

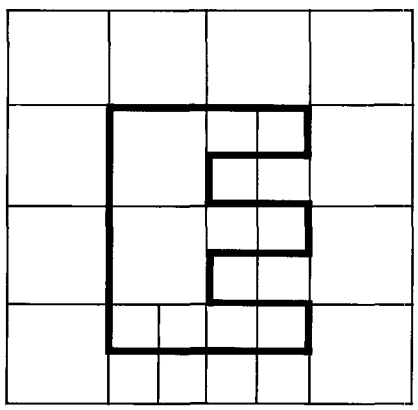

FIGURE 8. Quadcode Set of a Region

3.2 Set Arithmetic for Quadcodes

Suppose A and B are the quadcodes of two elementary squares

$$A = a_1, \ldots, a_k$$

$$B = b_1, \ldots, b_m$$

and $k \leq m$.

THEOREM 2. *The intersection of two such quadcodes is either empty or the longer quadcode:*

$$a_1, \ldots, a_k \bigcap_{k \leq m} b_1, \ldots, b_m = \begin{cases} b_1, \ldots, b_m & \text{if } a_1, \ldots, a_k = b_1, \ldots, b_k \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

PROOF.

(1) $a_1, \ldots, a_k = b_1, \ldots, b_k$
$B = b_1, \ldots, b_m = a_1, \ldots, a_k b_{k+1}, \ldots, b_m \subset a_1, \ldots, a_k = A$

(2) $a_1, \ldots, a_k \neq b_1, \ldots, b_k$

According to the definition of the quadcode,

$$a_1, \ldots, a_k \cap b_1, \ldots, b_k = 0$$
$$\text{and} \quad b_1, \ldots, b_m \subset b_1, \ldots, b_k \qquad \square$$

THEOREM 3. *The union of two such quadcodes is either their concatenation or the shorter quadcode:*

$$a_1, \ldots, a_k \bigcup_{k \leq m} b_1, \ldots, b_m = \begin{cases} a_1, \ldots, a_k & \text{if } a_1, \ldots, a_k = b_1, \ldots, b_k \\ \{a_1, \ldots, a_k, b_1, \ldots, b_m\} & \text{otherwise.} \end{cases}$$

PROOF. The proof is the same as for Theorem 2. □

The laws of set arithmetic are true for quadcode sets too, and they include the commutative, associative, and distributive laws, as well as de Morgan's law.

3.3 Union and Intersection

Definition. The *union* of two regions is represented by the union of the sets of the two regions.

For example, in Figure 9 there are two regions:

$$A = \{12, 211, 3\} \quad \text{and} \quad B = \{03, 1, 21, 310\}.$$

The union of A and B, according to eq. (10), is

$$A \cup B = \{12, 211, 3\} \cup \{03, 1, 21, 310\}$$
$$= \{03, 1 \cup 12, 21 \cup 211, 3 \cup 310\}$$
$$= \{03, 1, 21, 3\}.$$

Definition. The *intersection* of two regions is represented by the intersection of the sets of the two regions.

According to eq. (9),

$$A \cap B = \{12, 211, 3\} \cap \{03, 1, 21, 310\}$$
$$= \{1 \cap 12, 21 \cap 211, 3 \cap 310\}$$
$$= \{12, 211, 310\},$$

as shown in Figure 9.

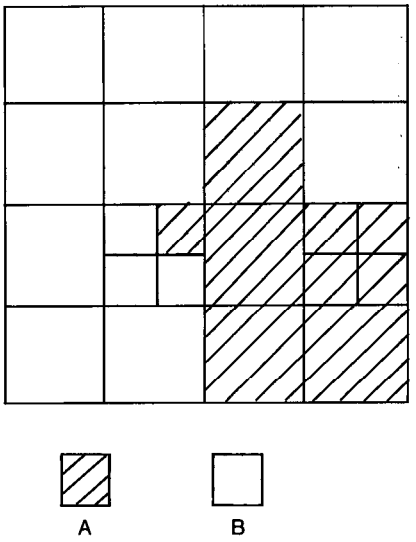

FIGURE 9. Union and Intersection

4. CONCLUSIONS

In this paper we introduced the quadcode system, discussed its geometric properties, analyzed the geometric concepts of elementary squares, and presented the quadcode set representation of regions. From our discussion we see that the information-compact and intrinsic hierarchical characteristics of the quadcode benefit the representations and analysis by making them more intuitive and computable. The quadcode can also be used in a tree-structured representation. In earlier work [4, 5] we discussed the quadcode-labeled tree (the qc-tree), storage efficiency [5], and applications of the qc-tree, such as boundary search [4]. Instead of using a tree as the interpretive medium—transferring an image into the tree, performing the processing by traverse and search of the tree, and then transferring back to the image—this paper discussed opportunities and methods for analysis *directly on the coded image*. In a companion paper [6], we discuss adjacency detection and perform all the computations in terms of quadcodes.

The properties and the results presented here for two-dimensional images can be extended to three-dimensional images, where the octcode combines the coordinates of three dimensions into one code. In binary form, each position of the octcode is composed of, respectively, z, y, and x coordinate

$$0_i = z_i y_i x_i.$$

The so-constructed octcode is also an intrinsically hierarchical coding system.

REFERENCES

1. Abel, D.J., and Smith, J.L. A data structure and algorithm based on a linear key for a rectangle retrieval problem. *Comput. Graph. Image Process. 24*, 1 (Oct. 1983), 1–13.
2. Gargantini, I. An effective way to represent quadtrees. *Commun. ACM 25*, 12 (Dec. 1982), 905–910.

3. Li, S.X., and Loew, M.H. Quadcodes and their application in image processing. GWU/IIST Rep. 83-15, Dept. of Electrical Engineering and Computer Science, George Washington Univ., Washington, D.C., Oct. 1983.
4. Li, S.X., and Loew, M.H. Boundary chain codes from quadcode trees. In *Proceedings of the IEEE 1984 Computer Vision Conference* (Annapolis, Md., Mar.). IEEE Press, New York, 1984, pp. 178–182.
5. Li, S.X., and Loew, M.H. The quadcode and its application. In *Proceedings of the 7th International Conference on Pattern Recognition* (Montreal, Canada, July 30–Aug. 2). IEEE Press, New York, 1984, pp. 227–229.
6. Li, S.X., and Loew, M.H. Adjacency detection using quadcodes. GWU/IIST Rep. 84-52, Dept. of Electrical Engineering and Computer Science, George Washington Univ., Washington, D.C., Dec. 1984 (also in this issue of *Communications*).
7. Samet, H. The quadtree and related hierarchical data structures. *ACM Comput. Surv. 16*, 2 (June 1984), 187–260.

CR Categories and Subject Descriptors: E.1 [**Data**]: Data Structures—*trees*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*hierarchy and geometric transformations*; I.4.2 [**Image Processing**]: Compression (coding)—*exact coding*; I.4.6 [**Image Processing**]: Segmentation—*pixel classification; partitioning*

General Terms: Algorithms, Performance, Theory
Additional Key Words and Phrases: Operations on encoded images, parallelism, pyramids

Received 3/84; revised 10/85, 11/86; accepted 2/87

Authors' Present Addresses: Shu-Xiang Li, Dept. of Automatic Control, Changsha Institute of Technology, Changsha City, Hunan Province, China; Murray H. Loew, Dept. of Electrical Engineering and Computer Science, George Washington University, Washington, DC 20052 (currently on leave at the Dept. of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel).