PERGAMON

Pattern Recognition 32 (1999) 235–251

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

A new chain code

Ernesto Bribiesca*

Department of Computer Science, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Apdo. 20-726, México, D.F., 01000, Mexico

Received 19 November 1997; received in revised form 25 August 1998; accepted 25 August 1998

Abstract

A new chain code for shapes composed of regular cells is defined. This boundary chain code is based on the numbers of cell vertices which are in touch with the bounding contour of the shape. This boundary chain code is termed *vertex chain code* (VCC). The VCC is invariant under translation and rotation. Also, it may be starting point normalized and invariant under mirroring transformation. Using this concept of chain code it is possible to relate the chain length to the *contact perimeter*, which corresponds to the sum of the boundaries of neighboring cells of the shape (Bribiesca, E., *Comp. Math. Appl.* 33(11) (1997) 1–9); also, to relate the chain nodes to the contact vertices, which correspond to the vertices of neighboring cells. So, in this way, these relations among the chain and the characteristics of interior of the shape allow us to obtain interesting properties. This work is motivated by the idea of obtaining various shape features computed directly from the VCC without going to Cartesian-coordinate representation. Finally, in order to illustrate the capabilities of the VCC: we present some results using real shapes. © 1999 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Chain coding; Vertex chain code; Contact perimeter; Contact vertices; Shapes

1. Introduction

Shape-of-object representation has always been an important topic in computer vision. This work deals with representation of shape based on a new proposed boundary chain code. Chain-code techniques are widely used because they preserve information and allow considerable data reduction. The first approach for representing digital curves using chain code was introduced by Freeman in 1961 [2].

Many interesting applications using chain-code representation have been reported, for example, McKee and Aggarwal [3] have used chain coding in the process of recognizing objects. Various shape features may be computed directly from this representation, contour smoothing and correlation for shape comparison are also relatively simple [4]. Classical methods for processing chains are referred to [5]. Other coding schemes are available, which are related to chain code [6–11]. Freeman [5] states that "in general, a coding scheme for line structures must satisfy three objectives: (1) it must faithfully preserve the information of interest; (2) it must permit compact storage and be convenient for display; and (3) it must facilitate any required processing. The three objectives are somewhat in conflict with each other, and any code necessarily involves a compromise among them". The here proposed VCC comply with these three objectives.

The proposed chain-code notation has some important differences in relation to the others cited above, these

* Fax: (525)622 3620; E-mail: ernesto@cic1.iimas.unam.mx

PII: S0031-3203(98)00132-0

differences will be analyzed in this paper. Some important characteristics of the VCC are: (1) The VCC is invariant under translation and rotation, and optionally may be invariant under starting point and mirroring transformation. (2) Using the VCC it is possible to represent shapes composed of triangular, rectangular, and hexagonal cells. (3) The chain elements represent real values not symbols such as other chain codes, are part of the shape, indicate the number of cell vertices of the contour nodes, may be operated for extracting interesting shape properties. (4) Using the VCC it is possible to obtain relations between the bounding contour and interior of the shape.

In this work, we present a new chain code for shapes composed of a finite number of regular cells. This paper is organized as follows. In Section 2 we present the VCC, its main characteristics, and some examples. Section 3 presents some results using real shapes. Finally, in Section 4 we give some conclusions.

2. The vertex chain code

Our purpose in this section is to present the VCC and its main characteristics. An important simplification in this work is the assumption that *entities* have been isolated from the real world. These entities are termed *discrete shapes*, and are defined as a result of previous processing: shapes are composed of regular *cells*. Fig. 1 shows a discrete shape composed of different forms of cells: (a) triangular, (b) rectangular, and (c) hexagonal cells. In the content of this work, the length l of each side of cells is considered equal to one.

The boundaries or contours of any discrete shape composed of regular cells can be represented by chains. Therefore, these chains represent closed boundaries. The minimum perimeter of a closed boundary corresponds to the shape composed only of one cell. Thus, in the content of this paper all chains are closed. When we use pixels to represent shapes, we have a structural problem called the *connectivity paradox*. There are two ways of connecting pixels: *four-connectivity* and *eight-connectivity*. In the content of this paper we use pixels with four-connectivity. In order to introduce the proposed chain code, a number of definitions are defined below:

Definition 1. An element a_i of a chain indicates the number of cell vertices, which are in touch with the bounding contour of the shape in that element position.

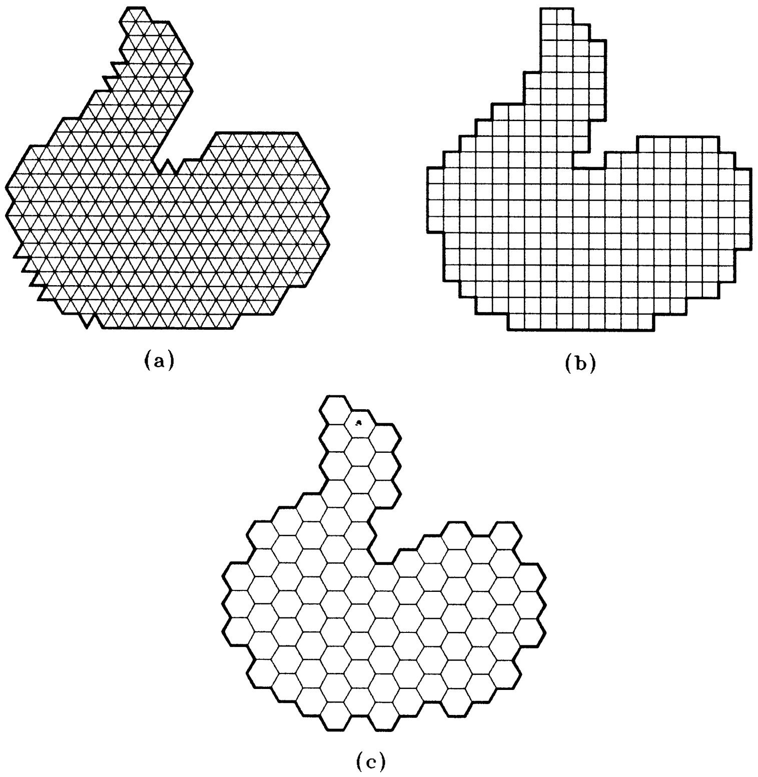
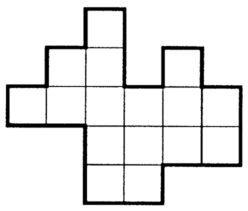
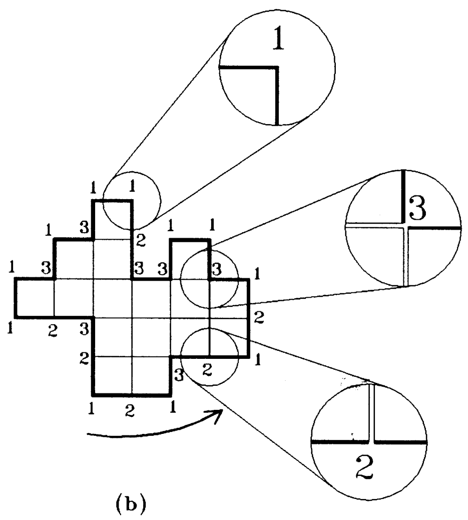


Fig. 1. Examples of discrete shapes composed of different forms of cells: (a) a shape composed of triangular cells; (b) a shape composed of rectangular cells; (c) a shape composed of hexagonal cells.



(a)



(b)

121321213113321131311232

(c)

112321213212131133211313

(d)

Fig. 2. The vertex chain code: (a) a shape composed of pixels; (b) the elements of the VCC; (c) the counterclockwise chain of the shape shown in (a); (d) the counterclockwise chain already invariant under starting point.

Fig. 2a presents a discrete shape composed of pixels. Fig. 2b shows the elements of the shape presented in Fig. 2a, note that when we are using pixels; there are only three different numbers of cell vertices for the bounding contour: 1, 2, and 3.

Definition 2. A chain A is an ordered sequence of elements, and is represented by

$$A = a_1 a_2 \ldots a_n = \{a_i : 1 \leqslant i \leqslant n\}. \tag{1}$$

Fig. 2c shows the counterclockwise chain which corresponds with the shape presented in Fig. 2a. This chain is composed of 24 elements.

Definition 3. The length L of a chain is the sum of the lengths of its elements [12]. L may be expressed as

$$L = nl. \tag{2}$$

The length L of the chain shown in Fig. 2c is 24.

2.1. Independence of starting point

Using the VCC for shapes composed of regular cells: all chains are closed. In this way, the VCC may be starting point normalized by choosing the starting point so that the resulting sequence of elements forms an integer of minimum magnitude [13]. Thus, the chain shown in Fig. 2c may be invariant under starting point by rotating the digits until the number is minimum. Fig. 2d presents the chain which is already invariant under starting point.

2.2. Independence of rotation

The VCC is invariant under rotation, this is due to the VCC is based on the number of cell vertices which are in touch with the bounding contour. Fig. 3a illustrates a discrete shape and its corresponding counterclockwise chain, this chain is invariant under starting point. Fig. 3b shows the independence of rotation of the shape presented in Fig. 3a and its corresponding counterclockwise chain.

2.3. Independence of mirroring transformation

The VCC is invariant under mirroring transformation. Any shape may be built directly from its chain code, when a shape is going to be built it we have to consider if our shape is to the left-hand side or to the right-hand side. Fig. 4a shows a shape which was built considering it to the right-hand side. On the contrary: the shape in Fig. 4b. Note that the shape in Fig. 4b is invariant under mirroring transformation.

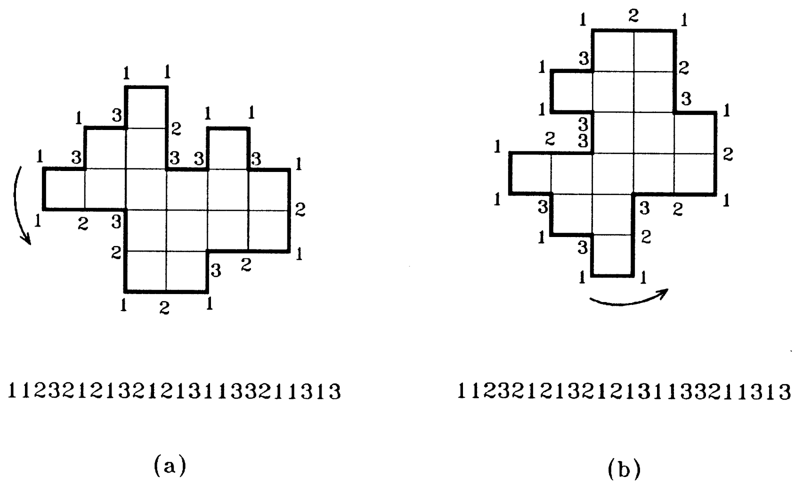


Fig. 3. Independence of rotation: (a) a shape and its corresponding counterclockwise chain; (b) the shape presented in (a) and its corresponding counterclockwise chain showing the independence of rotation.

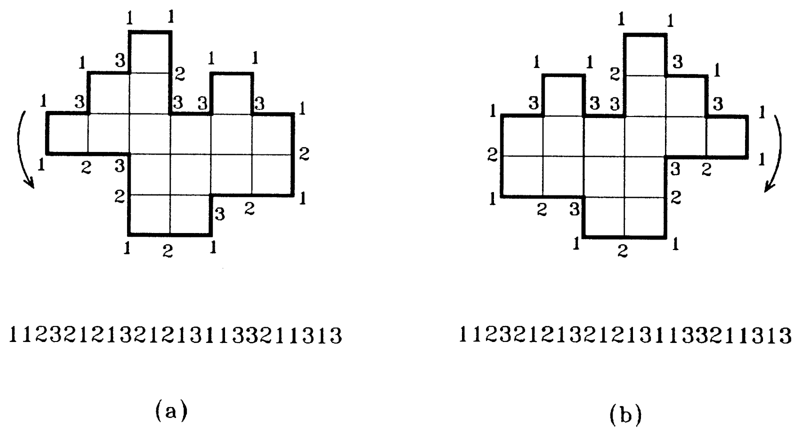


Fig. 4. Independence of mirroring transformation: (a) a discrete shape and its chain; (b) the shape and its chain showing the independence of mirroring transformation.

2.4. The proposed chain code for shapes composed of triangular and hexagonal cells

The chain-code concepts presented in this paper for shapes composed of rectangular cells (pixels), are valid for triangular and hexagonal cells too. Fig. 5a shows a shape composed of 13 triangular cells and its corresponding chain. In this case, there are five different numbers of cell vertices for the bounding contour: 1, 2, 3, 4, and 5. Fig. 5b presents a shape composed of 11 hexagonal cells and its corresponding chain. Note that when we are using this form of cell, we have only two different numbers of cell vertices for the bounding contour: 1 and 2.

2.5. The contact perimeter

Definition 4. The contact perimeter [1] of a shape composed of m regular cells corresponds to the sum of the boundaries of neighboring cells of the shape.

Theorem 1. *For any shape composed of m regular cells, the following equation is satisfied*:

$$2P_c + L = Tlm, \tag{3}$$

where P_c is the contact perimeter, L is the chain length or the sum of the chain lengths (for shapes with holes, which are described using more chains), and T is the number of the sides of the cell (in the content of this paper $l = 1$).

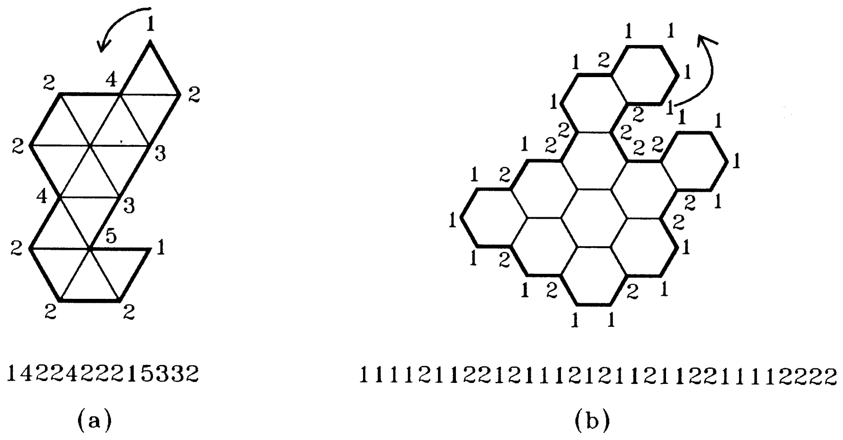


Fig. 5. Shapes composed of triangular and hexagonal cells: (a) a shape composed of 13 triangular cells and its chain; (b) a shape composed of 11 hexagonal cells and its chain.

Geometrically, this means that the sum of two times the contact perimeter plus the chain lengths is equal to the total sum of the perimeters of all the cells from the shape. The proof of this theorem is presented in Ref. [1].

Using Eq. (3), the contact perimeter is defined as follows:

$$P_c = \frac{Tlm - L}{2}. \tag{4}$$

Fig. 6a shows a shape composed of 16 pixels. Fig. 6b presents the bounding contour of the shape shown in Fig. 6a and its chain; its chain length is equal to 24. Thus, $m = 16$, $T = 4$, and $L = 24$. Substituting these values in Eq. (4): $P_c = 20$, which is shown in Fig. 6c.

$T = 3$ for shapes composed of triangular cells and $T = 6$ for shapes composed of hexagonal cells, respectively.

2.6. The contact vertices

Definition 5. The contact vertices of a shape composed of regular cells corresponds to the cell vertices with the largest number of neighbors. These vertices do not belong to the vertices of the boundary chain code, they belong to interior of the shape.

Theorem 2. *For any discrete shape composed of m regular cells, the following equation is satisfied*:

$$MV_c + V_s = Vm, \tag{5}$$

where M indicates the largest number of neighboring vertices that contact vertices may have, V_c is the number of contact vertices, V_s is the sum of the elements of the chain (*or chains for shapes with holes*), *i.e.* $V_s = \sum_{i=1}^{n} a_i$, *and V is the number of vertices of the cell which is used.*

The proof of this theorem is not included. For shapes composed of triangular cells: $M = 6$ and $V = 3$; for shapes composed of pixels: $M = 4$ and $V = 4$; and for shapes composed of hexagonal cells: $M = 3$ and $V = 6$.

Using Eq. (5) the number of contact vertices is defined as follows:

$$V_c = \frac{Vm - V_s}{M}. \tag{6}$$

Fig. 7a shows a discrete shape. Fig. 7b shows the elements of the chain of the shape presented in Fig. 7a and c illustrates the contact vertices of the shape. For this shape we have: $V = 4$, $m = 16$, $M = 4$, and $V_s = 1 + 1 + 2 + 3 + 2 + 1 + 2 + 1 + 3 + 2 + 1 + 2 + 1 + 3 + 1 + 1 + 3 + 3 + 2 + 1 + 1 + 3 + 1 + 3 = 44$. Substituting this values in Eq. (6) we obtain $V_c = 5$, which is shown in Fig. 7c. For the shape composed of triangular cells presented in Fig. 5a: $V = 3$, $m = 13$, $M = 6$, and $V_s = 33$. Therefore, the number of contact vertices $V_c = 1$, which is presented in Fig. 5a. Finally, for the shape shown in Fig. 5b, which is composed of hexagonal cells: $V = 6$, $m = 11$, $M = 3$, and $V_s = 45$. Therefore, $V_c = 7$, which is shown in the same figure.

2.7. Shapes without holes

In this subsection we will present an important theorem, which relates the sum of the chain elements to chain length for shapes (without holes) composed of pixels.

Theorem 3. *For any shape* (*without holes*) *composed of pixels, the sum of the chain elements of the shape is equal to* $2n - 4$, *i.e.*

$$\textstyle\sum_{i=1}^{n} a_i = 2n - 4. \tag{7}$$

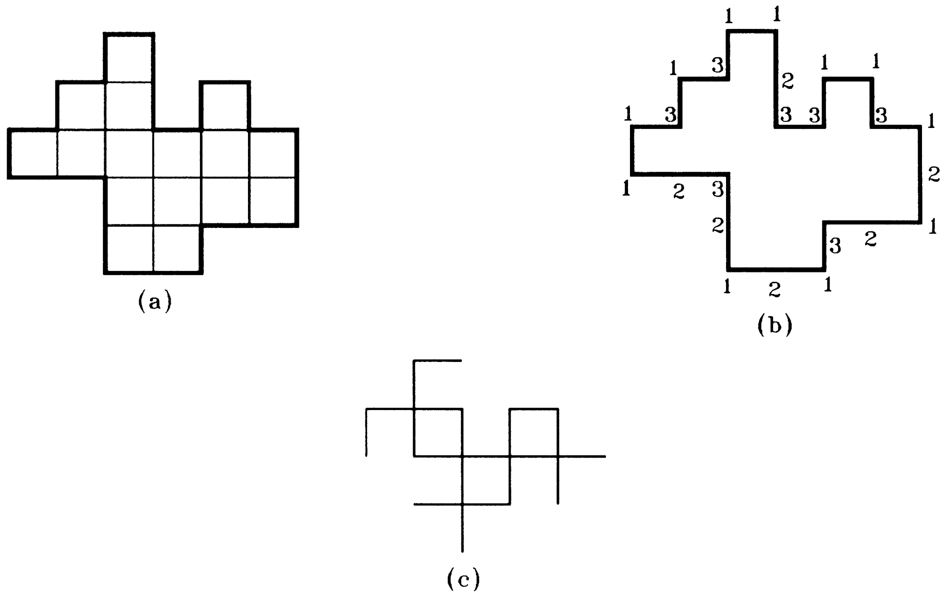


Fig. 6. The contact perimeter: (a) a shape composed of 16 pixels; (b) The bounding contour of the shape presented in (a) and its elements; (c) the contact perimeter of the shape shown in (a).

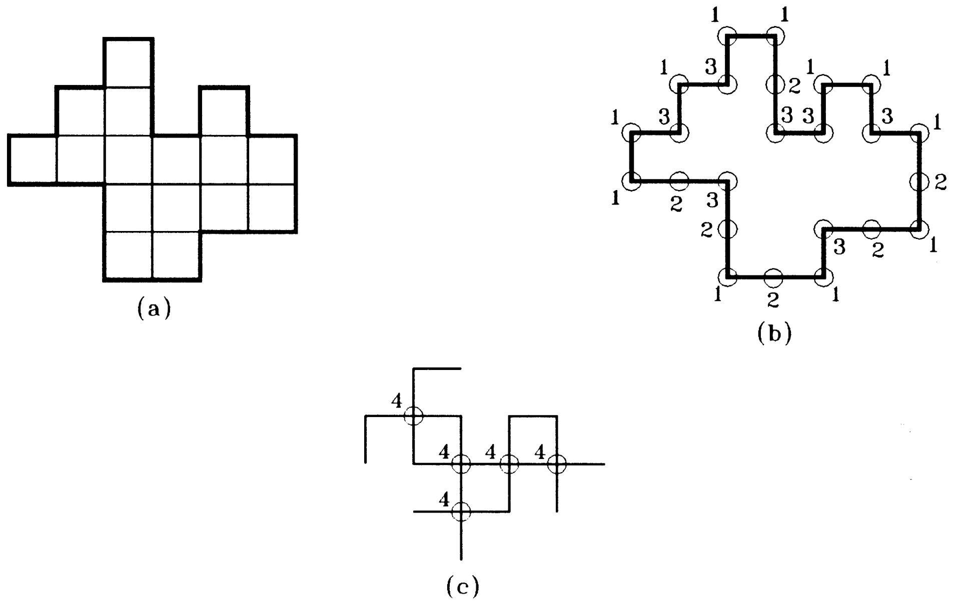


Fig. 7. The contact vertices: (a) a discrete shape; (b) the elements of the chain of the shape presented in (a); (c) the contact vertices of the shape shown in (a).

The proof of this theorem is not included. Fig. 8a shows the simplest shape only composed of one pixel, its chain is 1111, its element number is equal to 4 and the sum of the elements of the chain V_s is equal to 4, which may be obtained using Eq. (7). Fig. 8b illustrates a shape and its corresponding chain, its element number is equal to 16 and $V_s = 28$ using Eq. (7). For the shape shown in Fig. 8c: $n = 38$ and $V_s = 72$. Finally, for the shape

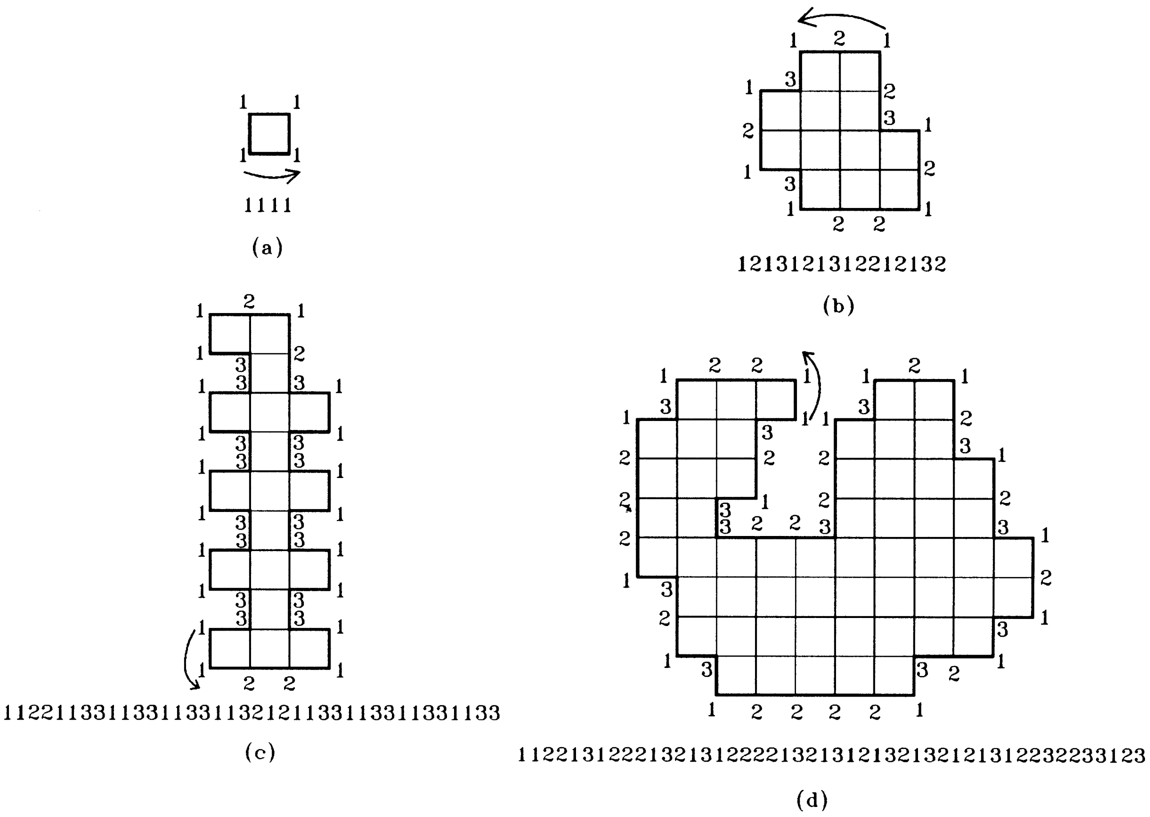


Fig. 8. Shapes without holes: (a) the simplest shape composed only of one pixel and its chain; (b) a shape and its corresponding chain which is composed of 16 elements; (c) a shape and its chain which is composed of 38 elements; (d) a discrete shape and its corresponding chain composed of 48 elements.

presented in Fig. 8d: $n = 48$ and $V_s = 92$. Notice that when we use Eq. (7): V_s is obtained directly, which permits to simplify some computations among chains.

Considering P_c as the number of contact straight-line segments and $l = 1$: $T = V$, then from Eqs. (3) and (5) we have

$$2P_c + L = MV_c + V_s, \tag{8}$$

substituting V_s for $2n - 4$, M for 4, and L for n, we obtain

$$2P_c = 4V_c + n - 4. \tag{9}$$

Eq. (9) relates the number of segments of the contact perimeter to the number of contact vertices.

Any chain must have at least four elements "1", this is due to the fact that only closed shapes are considered. Also, any chain which does not have elements "3" represents a rectangular shape.

2.8. Isoperimetric shapes

In this subsection, we introduce the VCC for isoperimetric shapes.

Definition 6. Two discrete shapes in the plane are isoperimetric if they have the same chain length, or perimeter.

Fig. 9 shows the discrete universe of all shapes which are composed of 12 elements. All these shapes are isoperimetric. Using Eq. (7), if $n = 12$ then $V_s = 20$. Thus, the sum of chain elements of each shape presented in Fig. 9 is equal to 20. In conclusion, if the sums of chain elements of two or more shapes are equal then they are isoperimetric. Note that the shapes presented in Fig. 9 are already invariant under mirroring transformation.

2.9. Complementary chains

Definition 7. The complement of a chain of a shape composed of pixels is another chain (termed complementary chain) whose elements "1" and "3" are replaced by "3" and "1", respectively.

Fig. 10a shows a shape and its chain. Fig. 10b illustrates the complementary chain of the chain presented in Fig. 10a. Fig. 10c presents one more example of complementary chains; notice that the chain shown to the

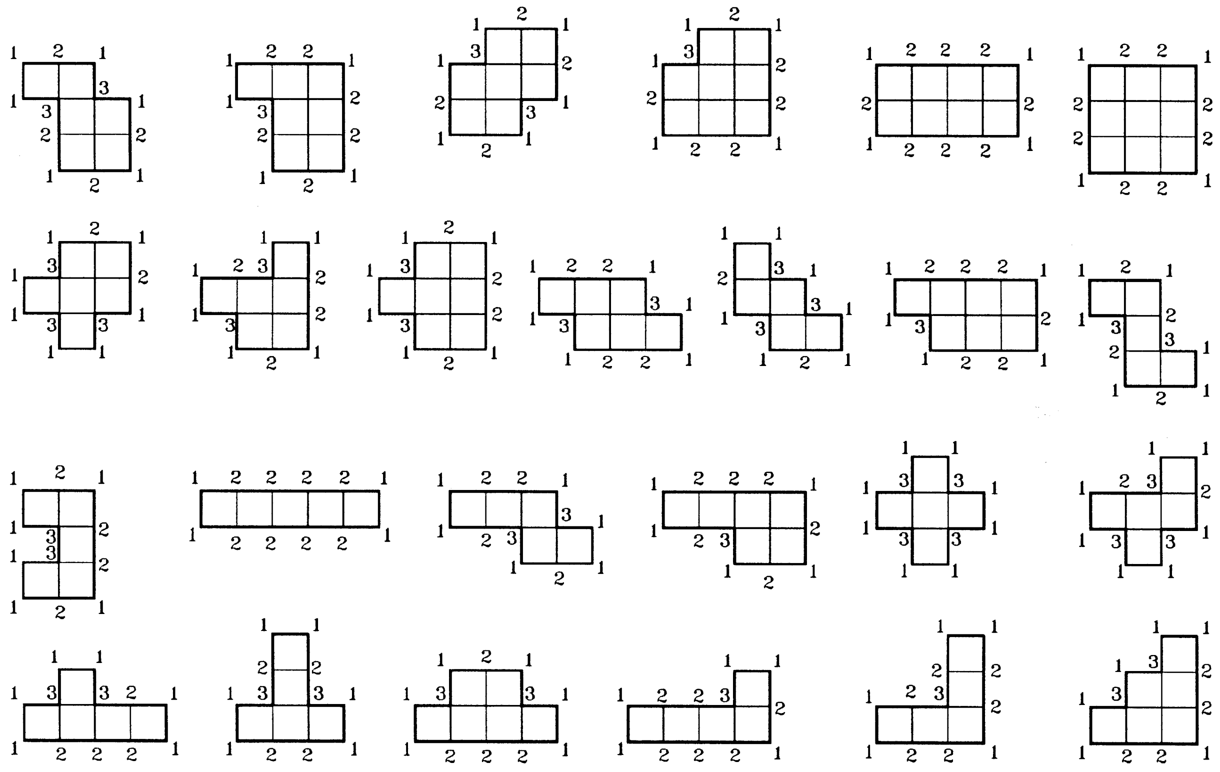


Fig. 9. The discrete universe of all shapes, which have a chain length equal to 12. All these shapes are isoperimetric.

right-hand side of the figure is the complementary chain and vice versa.

2.10. *How to determine the x and y components*

In this subsection, we present a method for obtaining x and y components from the VCC for shapes composed of pixels. Generally speaking, this work is motivated by the idea of obtaining various shape features computed directly from the proposed chain code without going to Cartesian-coordinate representation. Nevertheless, it is important to establish the relation between the VCC and the Cartesian-coordinate representation.

Based on the slope change notation proposed in Ref. [11], we transform the VCC into slope changes (these slope changes vary continuously from -1 to 1). Thus, the element "1" of the VCC produces the slope change "0.5", the element "2" produces the "0.0", and the element "3" the "-0.5", respectively. The integral of the chain of slope changes is the addition of the chain of slope changes, element by element. Therefore, the increments c_{x_i} and c_{y_i} in the axes "x" and "y" are defined as follows:

$$c_{x_i} = l\cos(\pi c_i), \qquad (10)$$

$$c_{y_i} = l\sin(\pi c_i). \qquad (11)$$

For example: Given the chain $A = 1131212122$, obtain its rectangular Cartesian coordinates.

Solution. First, we obtain the slope change chain B, which results from replacing "1" by "0.5", "2" by "0.0", and "3" by "-0.5", respectively. Thus, B is defined as follows:

$$B = 0.5\ 0.5\ \ -0.5\ 0.5\ 0.0\ 0.5\ 0.0\ 0.5\ 0.0\ 0.0.$$

Second, we get the integral of the chain B (termed C) by the addition of B, element by element, i.e.

$$C = 0.5\ 1.0\ 0.5\ 1.0\ 1.0\ 1.5\ 1.5\ 2.0\ 2.0\ 2.0.$$

Finally, substituting these values in Eqs. (10) and (11), we obtain the x and y increments:

$$(0,1)(-1,0)(0,1)(-1,0)(-1,0)(0,-1)(0,-1)(1,0)(1,0)(1,0).$$

Considering $(0,0)$ as the origin, the rectangular Cartesian coordinates of the chain A are as follows:

$$(0,0)(0,1)(-1,1)(-1,2)(-2,2)(-3,2)(-3,1)$$
$$(-3,0)(-2,0)(-1,0)(0,0).$$

Thus, using the above-mentioned method we may obtain the Cartesian-coordinate representation of any shape codified by the proposed chain code.

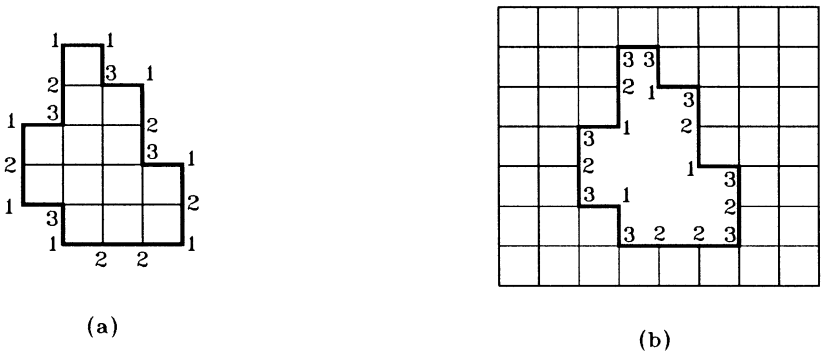
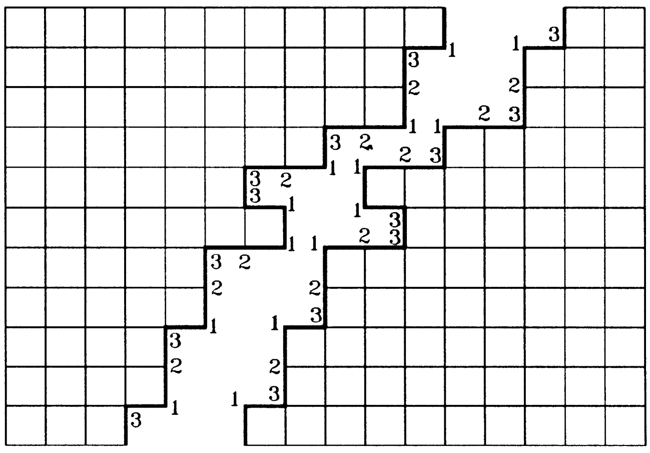



Fig. 10. Complementary chains: (a) a discrete shape and its chain; (b) The complementary chain of the chain of the shape presented in (a); (c) a second example of complementary chains.

2.11. How is this chain code related to the Euler's theorem?

Euler's theorem states that, for any simple polyhedron

$$V_p - E_p + F_p = 2, \tag{12}$$

where V_p, E_p, and F_p are the number of vertices, edges, and faces, respectively [14].

There are only five regular polyhedra; they are the regular *tetrahedron* (4 faces), the regular *hexahedron* or cube (6 faces), the regular *octahedron* (8 faces), the regular *dodecahedron* (12 faces), and the regular *icosahedron* (20 faces). Using the concepts defined previously: $V_p = (MV_c + V_s)/N_v$, $E_p = (2P_c + L)/2$, and $F_p = m$, where P_c may be considered as the number of contact straight-line segments of the contact perimeter, and L as the number of straight-line segments of the chain since $l = 1$; and N_v as the number of vertices of neighboring polygons for the polyhedron which is considered. Substituting V_p, E_p, and F_p in Eq. (12), we have

$$\frac{MV_c + V_s}{N_v} - \frac{2P_c + L}{2} + m = 2. \tag{13}$$

Fig. 11a shows the regular tetrahedron in the plane and its chain. For this polyhedron: $M = 6$, $V_c = 0$, $V_s = 12$, $N_v = 3$, $P_c = 3$, $L = 6$, and $m = 4$. Substituting these values in Eq. (13) the equality is satisfied. Fig. 11b shows the regular tetrahedron.

For the regular hexahedron or cube: $M = 4$, $V_c = 0$, $V_s = 24$, $N_v = 3$, $P_c = 5$, $L = 14$, and $m = 6$. Substituting these values in Eq. (13) the equality is satisfied. Fig. 11c illustrates the regular hexahedron in the plane and its chain. Finally, Fig. 11d shows the cube.

For the regular octahedron: $M = 6$, $V_c = 0$, $V_s = 24$, $N_v = 4$, $P_c = 7$, $L = 10$, and $m = 8$. Also, in this case using these values, Eq. (13) is satisfied. Fig. 11e shows the regular octahedron in the plane and its chain. Fig. 11f shows the regular octahedron. The regular dodecahedron is composed of 12 pentagons, which do not generated a regular tesselation of the plane.

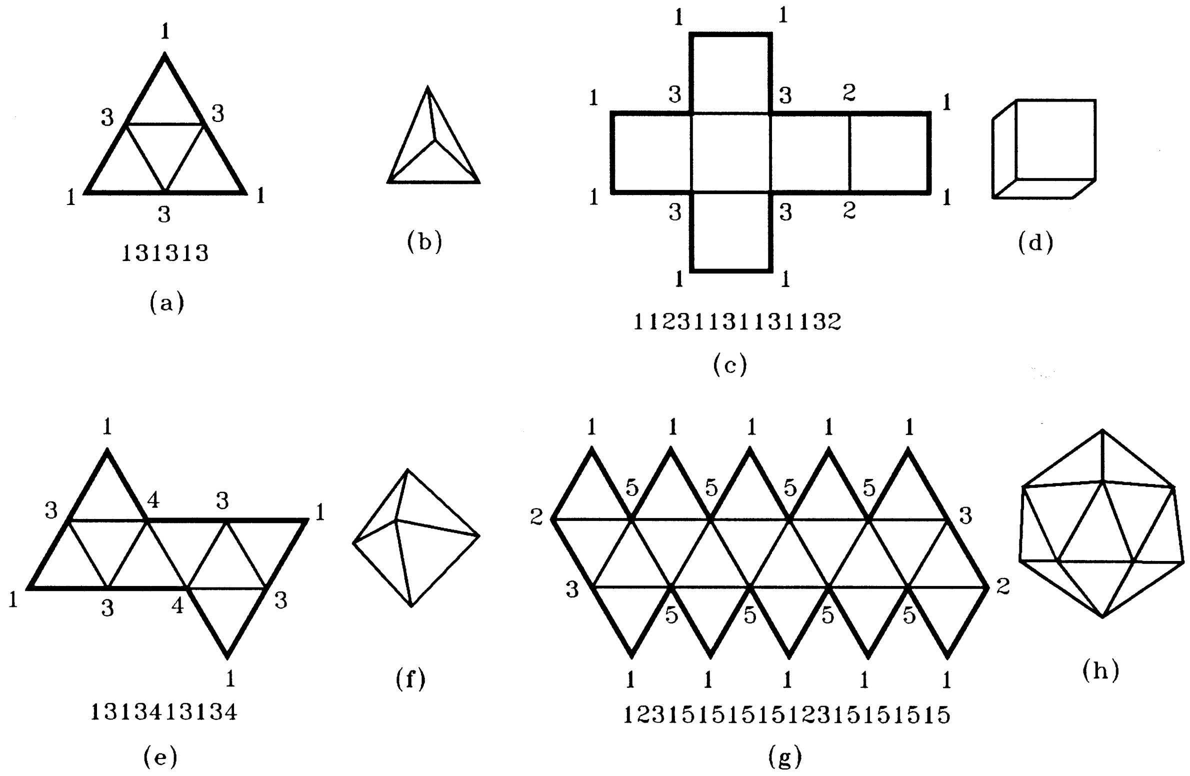


Fig. 11. (a) the regular tetrahedron in the plane and its chain; (b) the regular tetrahedron; (c) the regular hexahedron in the plane and its chain; (d) the regular hexahedron or cube; (e) the regular octahedron in the plane and its chain; (f) the regular octahedron; (g) the regular icosahedron in the plane and its chain; (h) the regular icosahedron.

Finally, for the regular icosahedron: $M = 6$, $V_c = 0$, $V_s = 60$, $N_v = 5$, $P_c = 19$, $L = 22$, and $m = 20$. Substituting these values in Eq. (13) the equality is satisfied. Fig. 11g illustrates the regular icosahedron in the plane and its chain. Fig. 11h shows the regular icosahedron.

3. Results

In order to illustrate the capabilities of the proposed chain code, we present some results of shape comparison, occluded shapes, and invariance under rotation and mirroring transformation using real shapes. Fig. 12 shows a digitalized image, which displays shapes of the real world. These shapes correspond to plant and tree leaves. Note that these shapes differ in size and orientation and some of them are presented by the mirroring transformation.

Fig. 13a presents "the fig leaf" composed of pixels. Fig. 13b shows the discrete contour of "the fig leaf" and its counterclockwise chain. Note that the starting point normalized is represented by a point over the contour. Fig. 13c shows the chain of "the fig leaf" already invariant under starting point. Fig. 14 shows the discrete contours of the plant and tree leaves and their corresponding chains, which were presented in Fig. 12, respectively. The chains of these shapes were obtained using the concepts of the VCC. The curves shown in Fig. 14 are simple closed curves, which divide the plane into two regions, an *interior* and an *exterior* (Jordan curve theorem).

3.1. Shape comparison

Shape comparison has always been an important topic in computer vision. Since the VCC is invariant under translation, rotation, mirroring transformation, and starting point normalized; to determine if two curves have the same shape, it is only necessary to see if their chain elements are equal. For instance, the chains of the shapes in the Fig. 14b, f, j, r, and v (which represent "the fern leaf") are equal. Note that the shape in Fig. 14f is the mirroring shape in Fig. 14j. The other leaves shown in Fig. 14 present more examples of shape comparison. The examples of the leaves shown in Fig. 14, which are invariant under rotation are based on discrete rotations of 90°. Therefore, the chain elements are preserved.

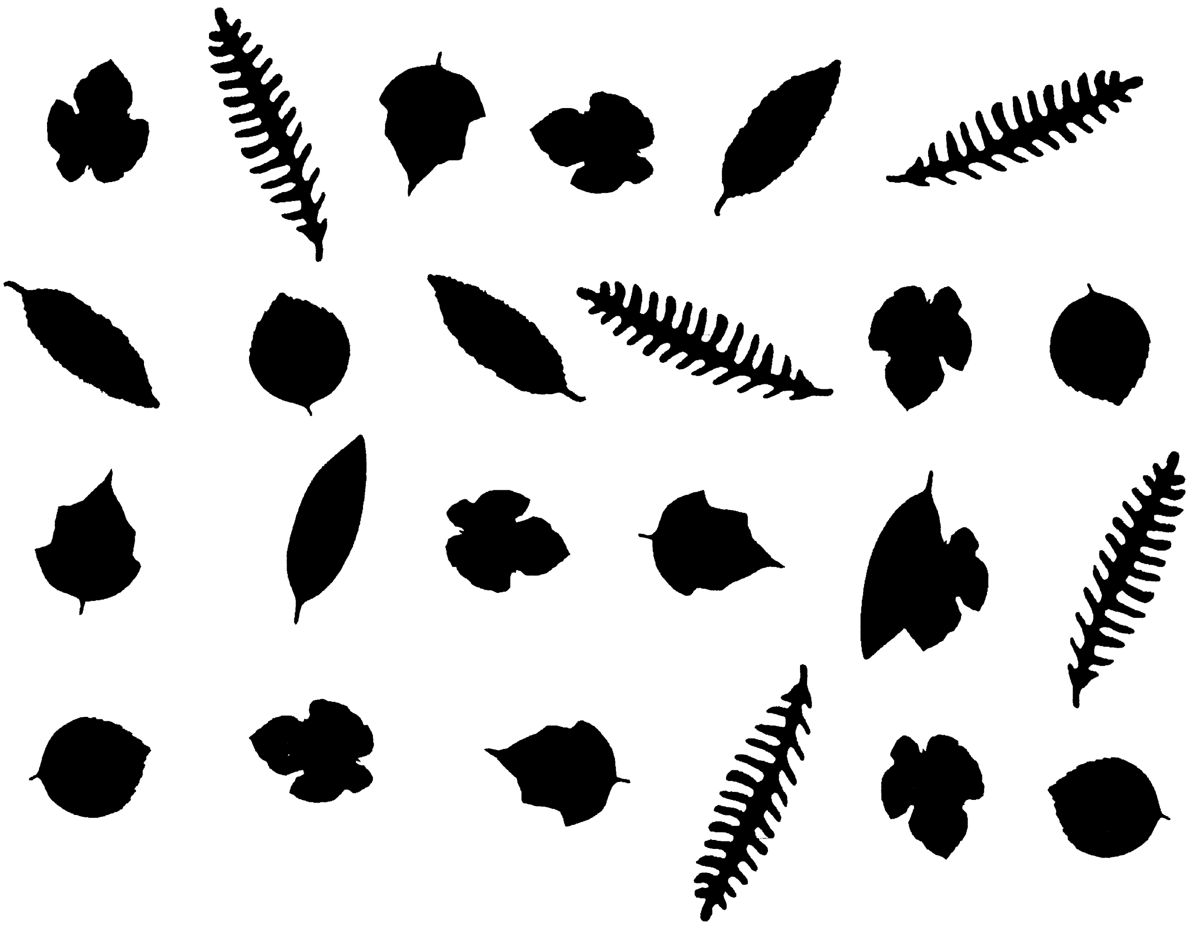


Fig. 12. A digitalized image of plant and tree leaves.

3.2. Occluded shapes

The identification of partially occluded shapes is an important part of the area of shape recognition. The occlusion takes place when a shape is either touched or overlapped by another shape [15]. Local features of shape play an important role when we are identifying occluded shapes. Thus, the chain elements (or part of them) of the proposed VCC are then used to extract local features of shape. Therefore, it is possible to decide whether or not a certain given local shape occurs within another shape to compare their chains or part of them. For instance, Fig. 14q shows two occluded shapes, which correspond to "the lemon and fig leaves". So, part of the chain of "the lemon leaf" occurs within the chain of the shape shown in Fig. 14q, which allows us to identify the required shape.

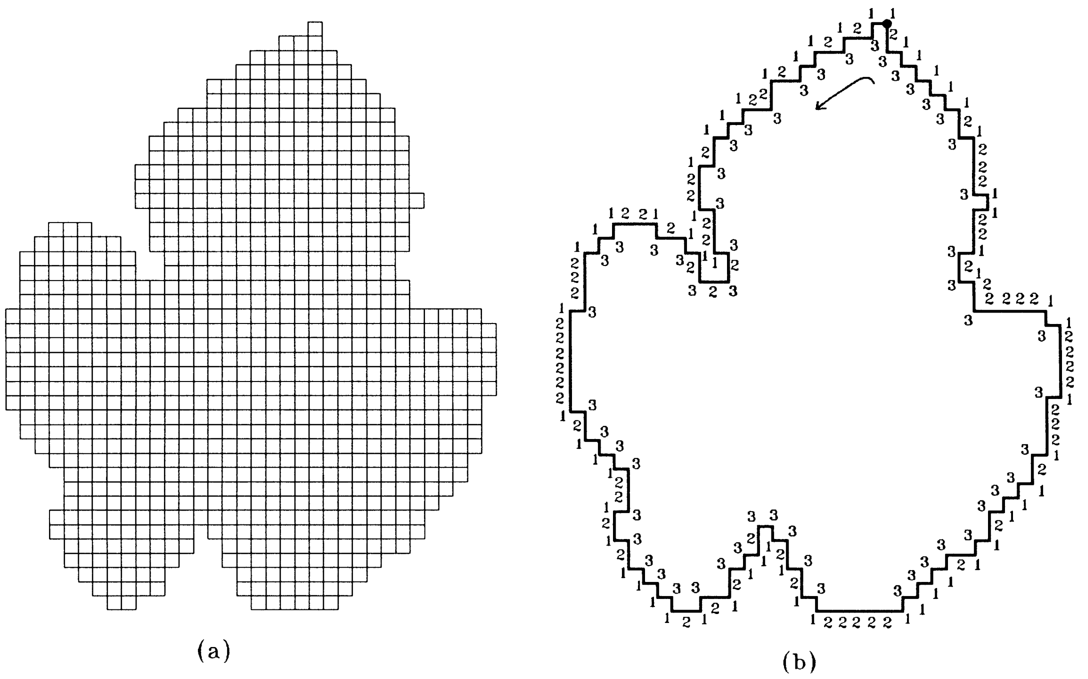
The above-mentioned example is very simple. Generally speaking, The occlusion of shapes is complex and may produce several chains. Therefore, additional criteria must be considered.

3.3. Noisy shapes

The invariance under rotation of any chain code (such as *shape numbers* [9]) is based on the derivative of its elements, i.e. the difference between contiguous elements. Thus, each element of the chain invariant under rotation depends on the direction of the previous element. The invariance under rotation of the VCC does not depend on previous elements, rather it only depends on the number of cell vertices in that element position. This allows us to have a notation which is more robust to noise, when a chain element is lost or changed.

Finally, we consider the problem of eliminating slight differences of shape (due perhaps to noise in the digitalization of real-world images). For instance, we apply a transformation to chain elements such as the following:

$$3\ 1\ 1\ 3 \rightarrow 2\ 2.$$



1132132131321232131313231221322132323213123122131312223122222213213131322312132 13
1313121321231312331321321312222213131313213123131312312223122221312222322132312 21
13222132131313131 32

(c)

Fig. 13. "The fig leaf" and its chain: (a) "the fig leaf" composed of pixels; (b) "the fig leaf" and its corresponding counterclockwise chain; (c) the counterclockwise chain already invariant under starting point.

When we apply this transformation of chain elements to a shape we eliminate protruding pixels, which are placed on straight parts of the shape contour. Due to the fact that we use pixels with four-connectivity, the use of this transformation does not change the topological properties of shapes.

Fig. 15a shows the contour of "the lemon leaf", which corresponds to the shape in Fig. 14n. Finally, we apply the above-mentioned transformation to the chain elements of "the lemon leaf" and we obtain the resultant contour shown in Fig. 15b. Note that three pixels were eliminated from the shape.

3.4. A better representation for shapes

The simplest *contour following* algorithms were presented by Papert [16] (Papert's turtle) and Duda et al. [17]. Thus, using these algorithms it is possible to represent shape contours by only two states: left turn (represented by "1") and right turn (represented by '0"). The above-mentioned process produces a chain composed of only binary elements. Fig. 16a shows "the poinsettia leaf" composed of pixels, this shape corresponds to the shape shown in Fig. 14u. Fig. 16c illustrates the contour following on "the poinsettia leaf". This contour was obtained using the algorithm proposed by Duda et al. [17] who state it as follows:

Scan the picture until a figure cell is encountered.
Then:
If you are in a figure cell turn left and take a step.
If you are in a ground cell turn right and take a step.
Terminate when you are within one cell of the starting point.

Fig. 16e shows the contour following on "the poinsettia leaf" and its corresponding chain composed of binary elements.

Is it possible to have a better contour representation than the above mentioned using only binary elements?. Yes, by means of the use of the proposed VCC, we can obtain a better definition of the shape contour. Firstly, our shapes must be composed of hexagonal cells, which have only two different number of cell vertices for the bounding contour: "1" and "2". After, the chain elements "1" and "2" are replaced by "0" and "1", respectively.

Thus, we obtain chains composed of only binary elements ("0" or "1") for shapes composed of hexagonal cells, which produces a better definition of shape contours. In this case, we use hexagonal cells with *six-connectivity*.

The Fig. 16b, d, and f illustrates the above mentioned. Fig. 16b displays "the poinsettia leaf" now composed of hexagonal cells. Fig. 16d shows the shape contour of "the poinsettia leaf" and its corresponding chain. Finally,

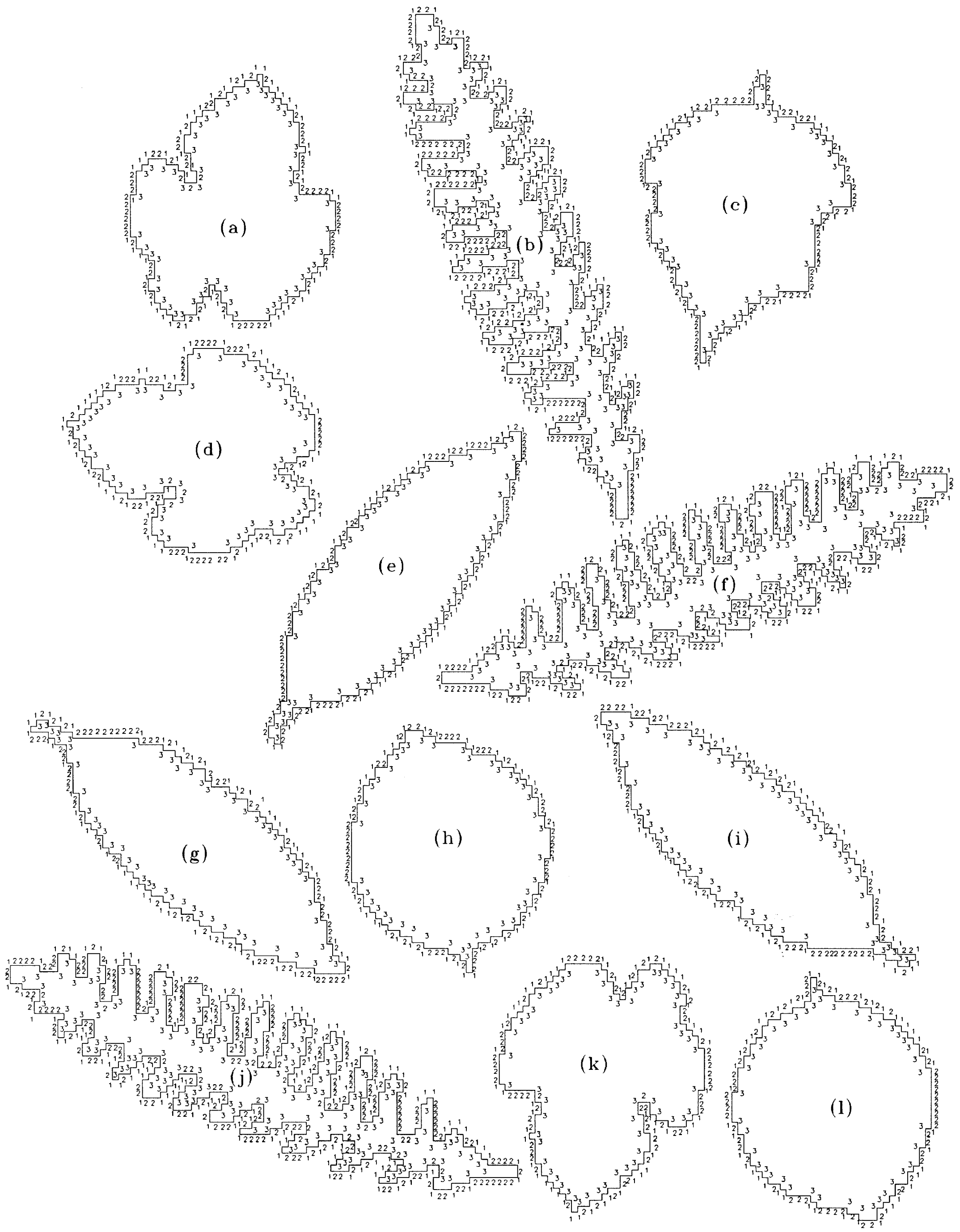


Fig. 14. The discrete contours of the plant and tree leaves and their corresponding chains, which were shown in Fig. 12, respectively. The chains of these shapes were obtained using the concepts of the VCC.

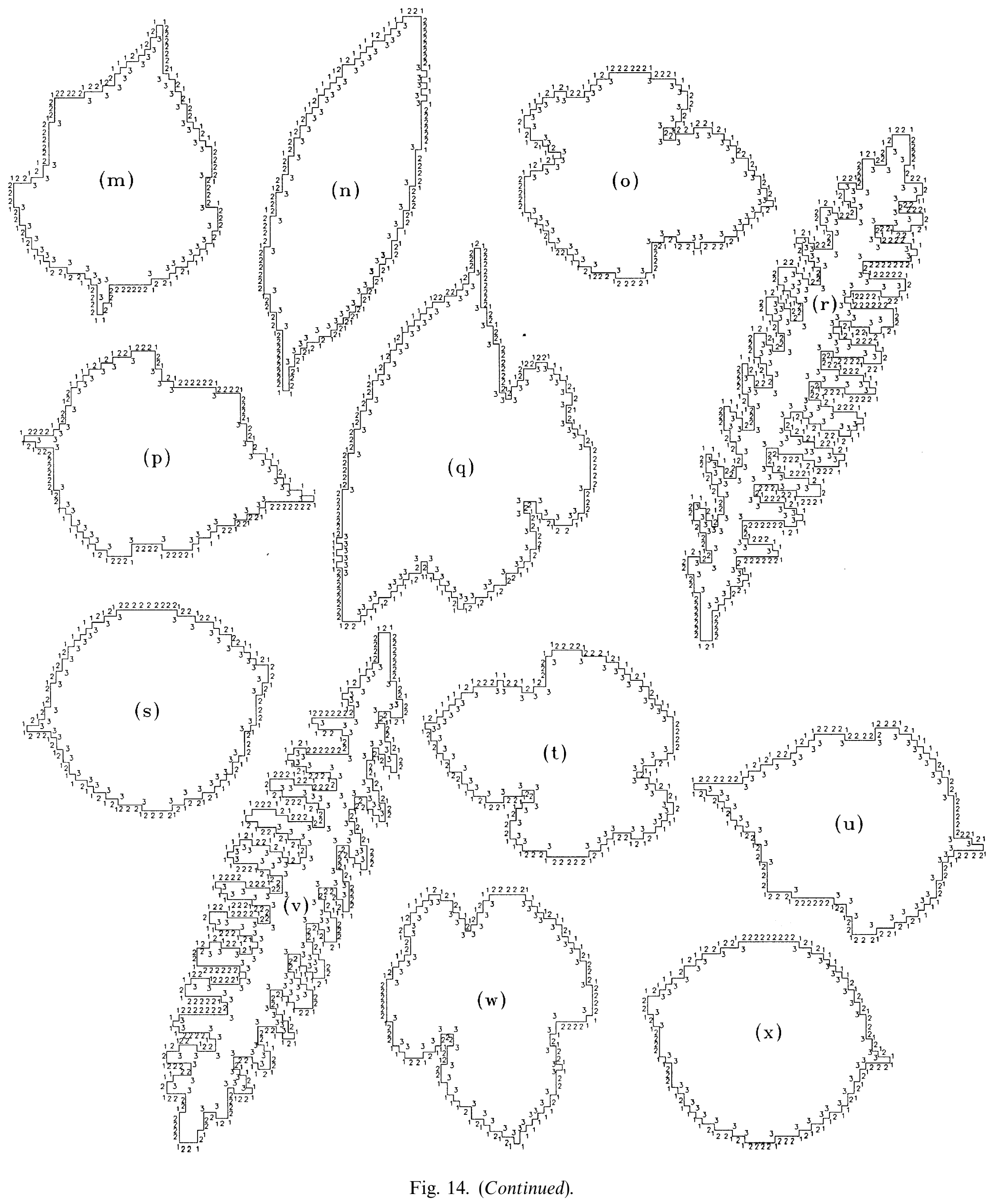


Fig. 14. (*Continued*).

Fig. 16f illustrates the shape contour of "the poinsettia leaf" composed of hexagonal cells and its corresponding chain, which is composed of only binary elements. Note that the contour presented in Fig. 16f has a better definition than the contour shown in Fig. 16e, and its topologial properties are preserved.

4. Conclusions

In this work, a new chain code for shapes composed of finite number of cells is defined. This definition of chain code (termed *vertex chain code*, VCC) is valid for shapes composed of triangular, rectangular, and hexagonal cells.

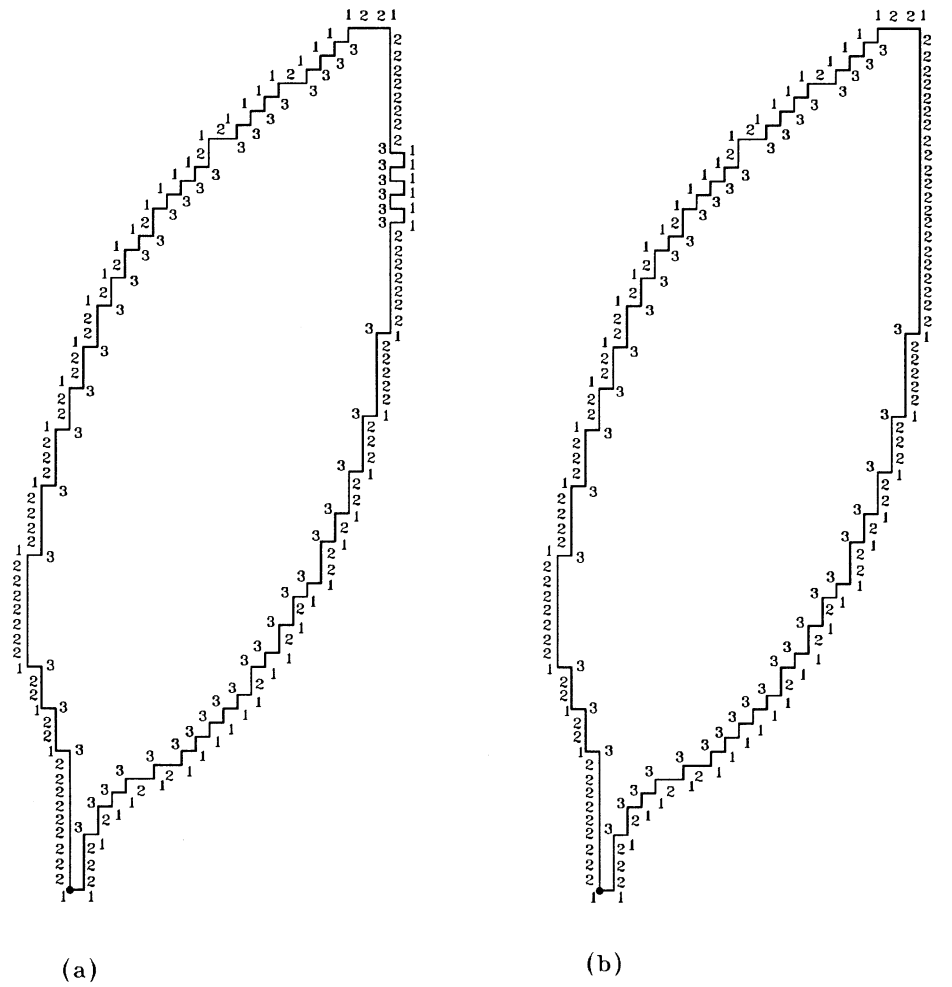


Fig. 15. Noisy shapes: (a) the discrete contour of "the lemon leaf"; (b) the resultant contour of "the lemon leaf".

The VCC preserves information and allows considerable data reduction. This chain code is invariant under translation and rotation, and optionally, under starting point and mirroring transformation. The chain elements represent real values not symbols such as other above-mentioned chain codes, they are part of the shape and indicate the number of cell vertices of the shape nodes.

A number of definitions and theorems are presented, which allow us to find out interesting properties, such as: the relation between the chain length and the *contact perimeter* [1]; the relation between the chain nodes and the contact vertices. Also, we study isoperimetric shapes composed of pixels, simple closed shapes, complementary chains, and others. Some results are presented which illustrate the capabilities of the proposed chain code. We present some examples of shape comparison, occluded shapes, and noisy shapes. In some cases, using these techniques it is possible to perform shape classification.

Suggestion for further work: extend the concepts of the proposed chain code for representing 3D shapes.

5. Summary

The study of shape representation is an important topic in computer vision. This work deals with representation of shape based on a new proposed boundary chain code, termed *vertex chain code* (VCC). This work is motivated by the idea of obtaining various shape features computed directly from the VCC without going to Cartesian-coordinate representation. The main characteristics of the VCC are: (1) the VCC preserves information and allows considerable data reduction; (2) this chain code is invariant under translation and rotation, and optionally, under starting point and mirroring transformation; (3) the VCC is valid for shapes composed of

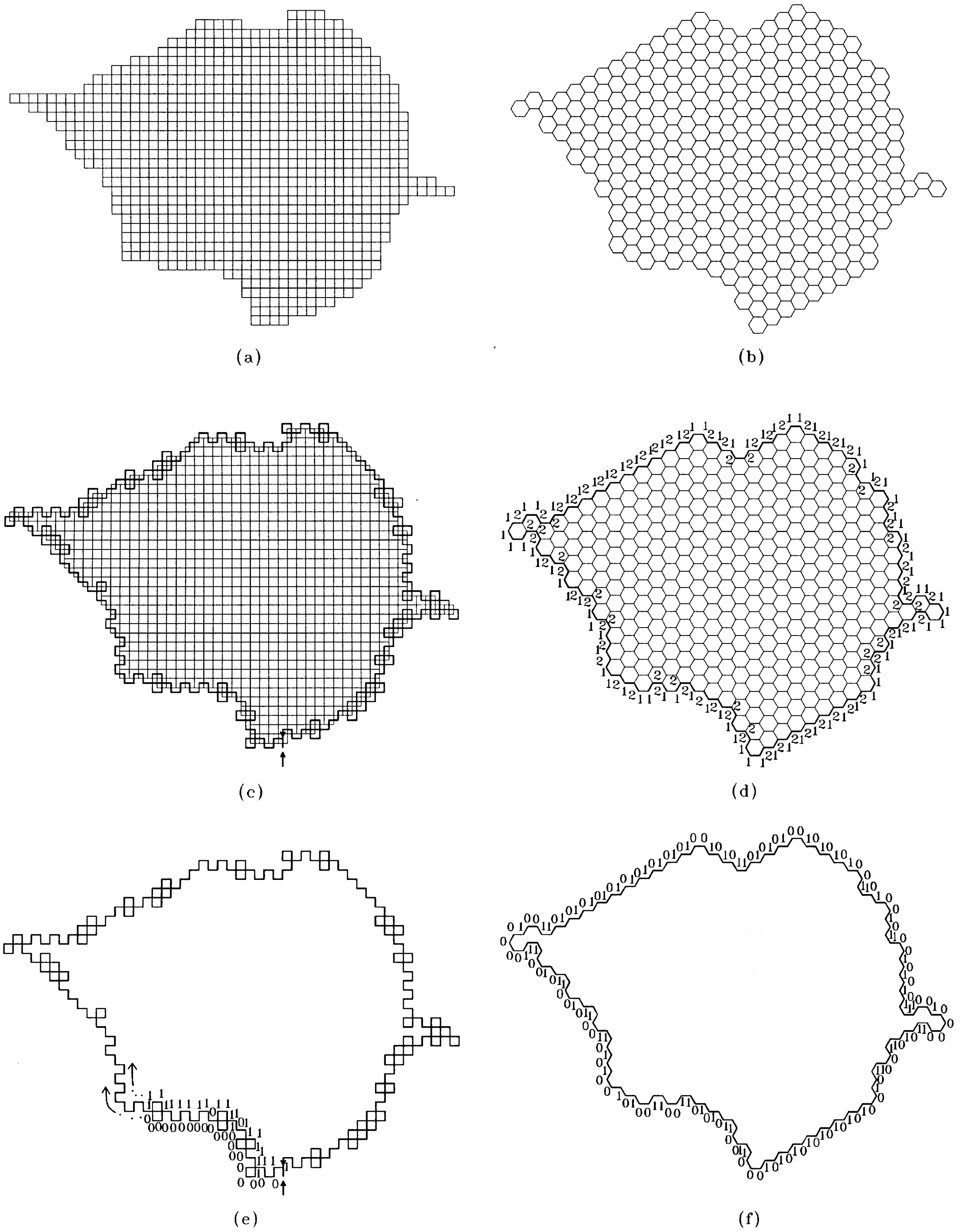


Fig. 16. A better representation for shapes: (a) "the poinsettia leaf" composed of pixels; (b) "the poinsettia leaf" composed of hexagonal cells"; (c) the contour following on "the poinsettia leaf"; (d) the shape contour of "the poinsettia leaf" and its corresponding chain; (e) the contour following on "the poinsetia leaf" and its corresponding chain composed of only binary elements; (f) the shape contour of "the poinsetia leaf" and its corresponding chain, which is composed of only binary elements.

triangular, rectangular, and hexagonal cells; (4) the chain elements represent real values not symbols such as other above-mentioned chain codes, they are part of the shape and indicate the number of cell vertices of the shape nodes; (5) Using the VCC it is possible to relate the chain length to the *contact perimeter* [1], also the chain nodes to the contact vertices, respectively. In this work, some definitions, theorems, and examples are presented, which describe the proposed chain code in a clear way.

Finally, in order to illustrate the capabilities of the proposed chain code: we present some results of shape comparison, occluded shapes, invariance under rotation and mirroring transformation. All the above mentioned using real shapes.

Acknowledgements

We are grateful to the referees for their constructive criticisms. We also wish to express our gratitude to Dr Richard G. Wilson for his help in reviewing this work.

References

[1] E. Bribiesca, Measuring 2D shape compactness using the contact perimeter, Comp. Math. Appl. 33(11) (1997) 1–9.

[2] H. Freeman, On the encoding of arbitrary geometric configurations, IRE Trans. Electron. Comput. EC-10 (1961) 260–268.

[3] J.W. McKee, J.K. Aggarwal, Computer recognition of partial views of curved objects, IEEE Trans. Comput. C-26 (1977) 790–800.

[4] M.D. Levine, Vision in Man and Machine, McGraw-Hill Publishing Company, New York, USA, 1985.

[5] H. Freeman, Computer processing of line drawing images, ACM Comput. Surveys 6 (1974) 57–97.

[6] F. Kuhl, Classification and recognition of hand-printed characters, IEEE Int. Conv. Record Pt. 4 (1963) 75–93.

[7] R.D. Merrill, Representation of contours and regions for efficient computer search, Comm. ACM 16 (1969) 534–549.

[8] G.S. Sidhu, R.T. Boute, Property encoding: applications in binary picture encoding and boundary following, IEEE Trans. Comput. C-21 (1972) 1206–1216.

[9] E. Bribiesca, A. Guzmán, How to describe pure form and how to measure differences in shapes using shape numbers, Pattern Recognition 12 (1980) 101–112.

[10] A. Blumenkrans, Two-dimensional object recognition using a two-dimensional polar transform, Pattern Recognition 24 (1991) 879–890.

[11] E. Bribiesca, A geometric structure for two-dimensional shapes and three-dimensional surfaces, Pattern Recognition 25 (1992) 483–496.

[12] H. Freeman, Techniques for the digital computer analysis of chain-encoded arbitrary plane curves, Proc. Natn. Electron. Conf. 18 (1961) 312–324.

[13] D.H. Ballard, C.M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[14] James, James, Mathematics Dictionary, 4th ed., Van Nostrand Reinhold Company, New York, 1976.

[15] M.H. Han, D. Jang, the use of maximum curvature points for the recognition of partially occluded objects, Pattern Recognition 23 (1990) 21–33.

[16] S. Papert, Uses of technology to enhance education, Technical Report 298, AI Lab, MIT, 1973.

[17] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

About the Author—ERNESTO BRIBIESCA received the B.Sc. degree in electronics engineering from the Instituto Politécnico Nacional in 1976. He received the Ph.D. degree in mathematics from the Universidad Autónoma Metropolitana in 1996, he was researcher at the IBM Latin American Scientific Center, and at the Dirección General de Estudios del Territorio Nacional (DETENAL). He is associate editor of the Pattern Recognition Journal. He has twice been chosen Honorable Mention winner of the Annual Pattern Recognition Society Award. Currently, he is Professor at the Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) at the Universidad Nacional Autónoma de México (UNAM), where he teaches graduate courses in Pattern Recognition.