Journal of
Electronic Imaging

SPIEDigitalLibrary.org/jei

Equivalence of chain codes

Hermilo Sánchez-Cruz
Hiram H. López-Valdez

imaging.org

Equivalence of chain codes

Hermilo Sánchez-Cruz* and Hiram H. López-Valdez
Centro de Ciencias Básicas, Departamento de Ciencias de la Computación, Universidad Autónoma de Aguascalientes, Av. Universidad 940, Col. Cd Universitaria, CP. 20131 Aguascalientes, México

Abstract. The representation of images is an active and very important area in image processing and pattern recognition. Therefore, in the literature, different contour codes for binary images have been proposed, such as $F4$, $F8$, VCC, $3OT$, and $AF8$. These codes have been used in many papers since the first chain code, $F8$, was introduced by Freeman in 1961. All the codes have been tried here as vector representations, including vertex chain code (VCC). To know their properties, this paper provides an analysis of comparisons of each code, and as an important contribution, we investigated the relationship between them and found a series of transformations that allow simple and efficient sets to go from one code to another. We found the equivalences between $F4$, VCC, $3OT$, $F8$, and $AF8$. As an important consequence of the transition matrix concept, we proposed a new code for eight connectivity by observing a missing code in the state of the art and in the inspiration of the $3OT$ code. [DOI: 10.1117/1.JEI.23.1.013031]

Keywords: chain codes; vertex chain code; *3OT*; contour; transition matrix.

Paper 13339 received Jun. 25, 2013; revised manuscript received Dec. 19, 2013; accepted for publication Jan. 23, 2014; published online Feb. 28, 2014.

1 Introduction

Using chain codes is a common way for binary object-shape representation. The Freeman code, proposed in 1961,[1] is known as Freeman chain code, and it is composed of eight directions, so we call it here $F8$ code. It is a compact way to represent binary object shapes, and it has the characteristic of going through a sequence of pixels of the contour on the basis of eight connectivity. Each movement direction is codified using a symbol α, where $\alpha = 0, 1, 2, \ldots, 7$ in clockwise direction [see Fig. 1(b)]. Analogous to $F8$, $F4$ code visits the pixels of the contour using four connectivity. A movement direction is codified using a symbol α, with $\alpha = 0, 1, 2, 3$ [see Fig. 1(a)]. $F4$ is also known as crack code because it covers the contour shape along edges of border pixels.[2–6]

Since Freeman proposed the first chain code,[1] a considerable amount of papers using chain codes for a wide variety of issues have appeared.

One motivation to study chain codes is that, despite the different chain codes that have been proposed independently, a theoretical analysis is necessary to know if the codes are related. Of course, implementing the codes takes some technical effort. For example, $3OT$ is more difficult to implement than vertex chain code (VCC) because the former takes into account relative changes regarding a reference vector, which remains the same unless another change occurs. As can be studied from the literature, different proposals have been implemented independently. However, a question arises: can a general scheme be constructed that relates the different codes and predicts the existence of more?

Another motivation to study chain codes is that they can be used particularly in compression of scanned documents. For example, recently $3OT$ code has been utilized to compress image text documents, obtaining more than twice the levels of compression regarding the international standard called Joint Bilevel Image Expert Group 2.[7]

Other applications are, for example, for map representations,[2,8] to look for dominant points,[9–13] for analysis, and, as we already mentioned, for document compression.[7,14]

Since the basic codes were appearing independently, a question to answer is: can existing codes be obtained through a general model? Therefore, the main objective of this paper is to make a theoretical analysis of the different codes existing in the literature and to find a general model that relates them, although they emerged independently. From this study, filling the gaps of missing codes is necessary.

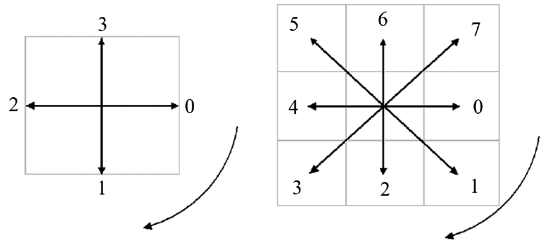
Among the advantages of this study, we can say that having equivalences of the codes, we can handle different geometric interpretations of the shapes-of-objects information, mainly depending on four or eight neighborhood in contour representations for analysis and recognition purposes.

Depending on the model of representation used, it is necessary to work with either eight or four neighborhood. For example, if a comprehensive analysis of discrete straight lines detection is required, then eight neighborhood is necessary.[15] On the other hand, if improving the level of compression is required, then $3OT$ code plays an important role. While $3OT$ code improves levels of compression regarding the VCC, the latter has some advantages in terms of representation. For example, due to its nature, VCC can give information of concavities and convexities of the contours; another advantage is its relationship with the Euler number.[16] Other proposals can search more geometric properties. The codes give particular geometric interpretations depending on whether they represent four or eight connectivity.

In 1999, Bribiesca[17] proposed VCC. Among its most important features, VCC is composed of three symbols: $\alpha = 1, 2, 3$; it is invariant under mirror and rotation transformations and invariant to the initial point, which has some

*Address all correspondence to: Hermilo Sánchez-Cruz, E-mail: hsanchez@correo.uaa.mx

0091-3286/2014/$25.00 © 2014 SPIE and IS&T



(a) (b)

Fig. 1 Freeman symbols obtained in clockwise direction: (a) $F4$ code and (b) $F8$ code.

Table 1 Basic codes and their derived codes.

Year proposed	Basic code	Derived codes
1961	$F8, F4$	
1999	VCC	E_{VCC}, C_{VCC}, V_{VCC}
2005	$3OT$	M_{3OT}
2005	$AF8$	$MDF9$

relationship between the contour and inside information. This code represents the changes made to an object contour by computing the number of affected pixels.

In 2005, Sánchez-Cruz and Rodríguez-Dagnino[18] proposed the $3OT$ code. They compared it with $F4$ and obtained a better performance, thanks to the use of the symbols 0, 1, and 2 to label the changes generated in relation to orthogonal directions. $3OT$ code has the same number of symbols as VCC; however, it is composed of three vectors to codify the contour: *reference*, *support*, and *change*. In this case, the symbol 0 represents no changes between reference and support, whereas 1 represents a change equal to reference and 2 represents a change in contrary sense of reference.

Supported on the $F8$, in 2005 Kui and Zalik[19] proposed a new chain code, which we call here $AF8$. It is based on changes obtained with every pair of $F8$ code vectors when following the contour, i.e., any vector change in the contour is compared with the previous one, and depending on the angle, a symbol is given. This code was compared with $F4$, $F8$, VCC, and $3OT$,[20] and the results showed that $AF8$ offered more advantages.

The set of codes studied is $\mathcal{B} = \{F4, VCC, 3OT, F8, AF8\}$, and we call them basic codes (see Definition 1).

From $\mathcal{B}$ other codes have been derived. The derived codes were obtained by combining the symbols that appear in the contours and making a probabilistic model to modify the number of bits required to store the coded contours. For example, E_{VCC}, V_{VCC}, and C_{VCC} were proposed in Ref. 21. E_{VCC} was obtained by considering that VCC uses two bits to represent three symbols, V_{VCC} arises by considering a variable length of VCC chain, and C_{VCC} is based on applying the Huffman algorithm on the VCC code symbols. On the other hand, M_{3OT} was proposed by considering grouping symbols of $3OT$,[22] whereas $MDF9$ was proposed by considering the $AF8$ patterns in pieces of discrete straight lines of the contour shapes.[15] Table 1 shows the basic and the derived codes.

According to different criteria to satisfy the need to represent the object shapes and looking for relationships of contours and inner regions, in the literature, it is reported that each code of the set $\mathcal{B}$ was obtained independently and in different years. However, we demonstrate that there is an equivalence of all basic codes, even more there is a way to go from one code to another. In 1997, Wilson found an equivalence between $F8$ and $F4$.[6] However, in this work, we found the equivalence between the basic codes proposed since then.

In this paper, we demonstrate that there is a transition matrix that allows us to go from one code C_1 to another C_2, with $C_i \in \mathcal{B}$. If this happens, we say the codes C_1 and C_2 are equivalent.

The transition matrix that allows us to go from a code to another will be very important because if a coding has some properties, like rotation invariance, they have to be reflected in the transition matrix.

Of course, all the codes can be handled as vector components, VCC inclusive, as we show in this paper. The mentioned codes are useful to represent contour shapes in either four or eight connectivity. A summary of the basic codes for eight and four connectivity contours appears in Table 2. According to the number of vectors used to codify, we classify the basic codes in three types.

- Type 1 is composed of the codes that use one vector.
- Type 2 is composed of the codes that use two vectors.
- Type 3 is composed of the codes that use three vectors.

This information is shown in Table 2, in which we notice that there are two codes of types 1 and 2; however, there is a missing code for type 3 represented by a small box, which has not been proposed. In this work, after analyzing matrix transformations to find code equivalences, we discover this missing code.

In Sec. 2, we introduce the basis of the codes $F4, VCC, 3OT, F8$, and $AF8$. In Sec. 3, we study the case of four connectivity, and we demonstrate that $F4$, VCC, and $3OT$ are equivalent. In Sec. 4, we deal with the case of eight connectivity, and we demonstrate that $F8$ and $AF8$ are equivalent. Despite that some codes are for eight connectivity and others are for four connectivity, in Sec. 5, we demonstrate that all the codes of the basis are equivalent. As a result of equivalences of chain codes, in Sec. 6, we found a new code for eight connectivity. In Sec. 7, we test our new code on some sample objects and obtain some results. Finally, in Sec. 8, we present some conclusions and future work.

2 Concepts and Definitions

The codes that we use in this work are $F4, VCC, 3OT, F8$, and $AF8$; from these we give the following definitions:

Table 2 Bases of chain codes representing four and eight connectivity. The box represents a missing code.

	One vector	Two vectors	Three vectors
4 connectivity	$F4$	VCC	$3OT$
8 connectivity	$F8$	$AF8$	$\square$

Definition 1 The set of codes studied is $\mathcal{B} = \{F4, VCC, 3OT, F8, AF8\}$, and we call them basic codes.

Definition 2 If a figure is codified with some code $\mathcal{X}$, and the coding is

$$C = \{C_{\mathcal{X}}(1), \dots, C_{\mathcal{X}}(n)\},$$

then the number n is called the length of the code $\mathcal{X}$.

2.1 *F*4 *and F*8 *as a Vector Representation*

Traditionally, $F4$ and $F8$ symbols are obtained by considering their basis vector in counterclockwise direction. However, for convenience, in this paper, we use the clockwise direction (see Fig. 1).

2.2 *VCC as a Vector Representation*

The VCC was initially conceived as a code based on the number of pixel vertices that are in touch with the boundary contour of the shape, giving three cases coded by an alphabet of three symbols. However, by analyzing the nature of this code, we can give it a vector interpretation. As can be seen in Fig. 2 for VCC basis, given two connected resolution pixels, there are three adjacent edges at the vertex of the contour. The two edges of the outer contour can be used as directed segments in the same direction. On the other hand, when a pixel resolution is visited and we see that the visited vertex has two adjacent edges, we define two vectors and consider one of them as a vector change to the right. Finally, if the vertex visited is adjacent to three pixels of resolution, there is a vector change to the left.

So, when we use VCC code we need to analyze every two vectors (first and second). If the second vector has the same direction as the first, we put 0. If the second vector

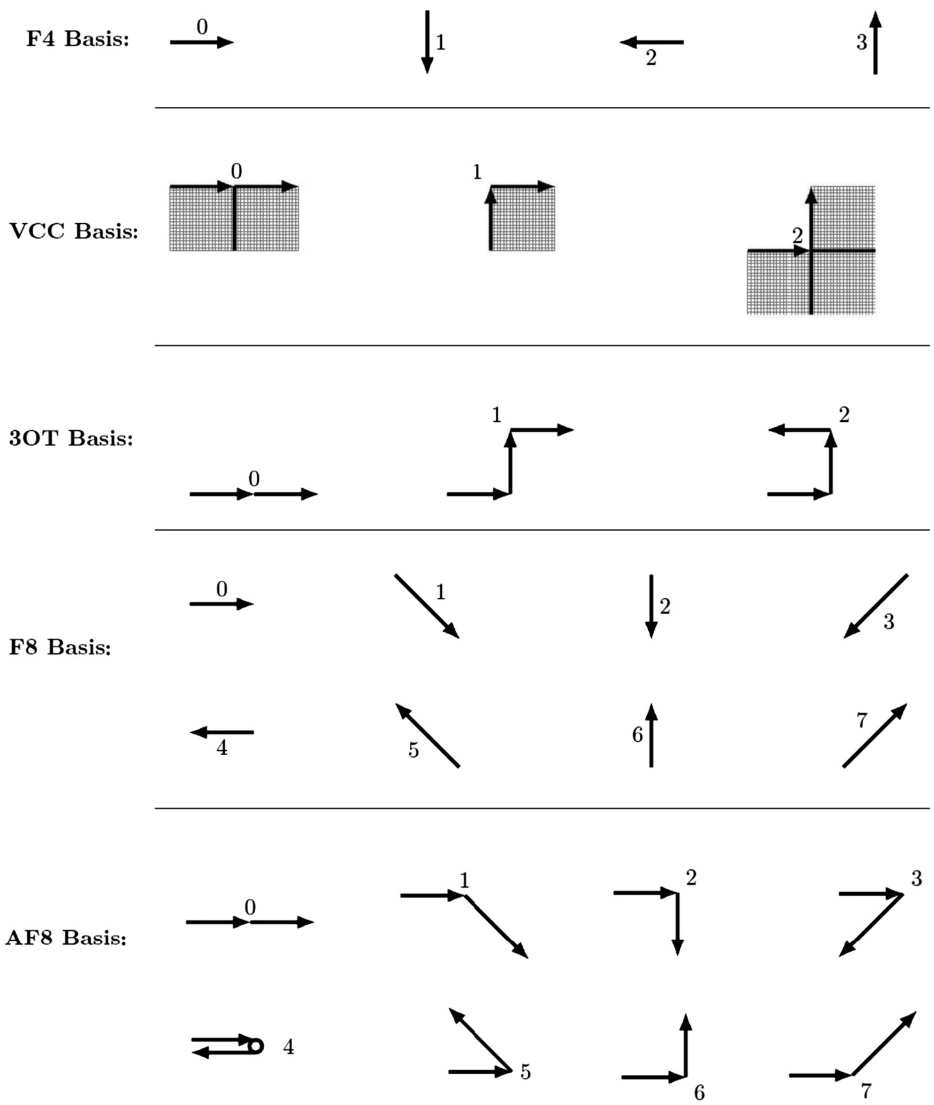


Fig. 2 Bases used for the codes, all with vector representations, including VCC code. $C_{F4} = \{0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 3, 2, 3, 3, 0, 0, 0, 3, 2, 2, 3, 3, 0, 3\}$, $C_{VCC} = \{1, 0, 0, 0, 1, 0, 0, 2, 1, 0, 1, 2, 1, 2, 1, 0, 0, 0, 1, 2, 1, 0, 1, 0, 0, 2, 2, 0, 1, 0, 1, 2\}$, $C_{3OT} = \{0, 0, 0, 2, 0, 0, 1, 1, 0, 2, 1, 1, 1, 1, 0, 0, 0, 2, 1, 1, 0, 2, 0, 0, 1, 2, 0, 1, 0, 2, 1, 1\}$.

changes 90 deg to the right regarding the first, we put 1. If the second vector changes 90 deg to the left regarding the first, we put 2. The vectorial base, or simply base, is given in Fig. 2.

2.3 3*OT* as a Vector Representation

In the case of 3OT, we need to use three vectors to codify the contour: *reference*, *support*, and *change*. The symbol 0 represents no changes between reference and support, 1 represents a change in the same direction to a reference, and 2 represents a change in contrary sense of the reference. Of course, support vector can be a sequence of r vectors in the same direction, with $r \geq 1$.

Figure 3 shows an example of coding using $F4$, VCC, and 3OT.

Before starting to define the bases, we mention some features that the codes of $\mathcal{B}$ have in common.

Note 1 $F4, VCC, 3OT, F8$, and $AF8$ have the following common features (Figs. 3 and 4 illustrate some examples):

1. Contour pixels are visited on clockwise direction.
2. By simplicity, the first pixel is visited at leftmost top of the figure.

The second part of Note 1 is a convention. We have always started with this pixel to compare different chain codes of the same figure. Of course, another convention is possible if starting with another pixel is required.

Note 2 $F4$, VCC, and 3OT have the following common features (Fig. 3):

1. Edges of contour pixels are visited.
2. The first vertex visited is in the leftmost part of the top of the first pixel.
3. The first two contour edges of the first pixel visited correspond to 3 and 0 symbols of $F4$ code.

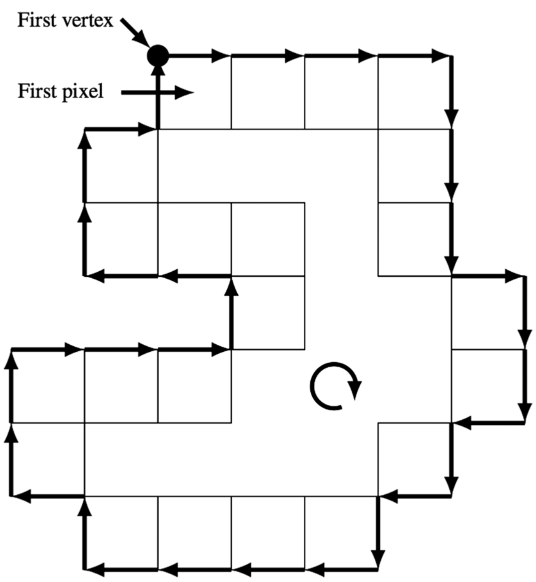


Fig. 3 The figure is covered through outer vertex.

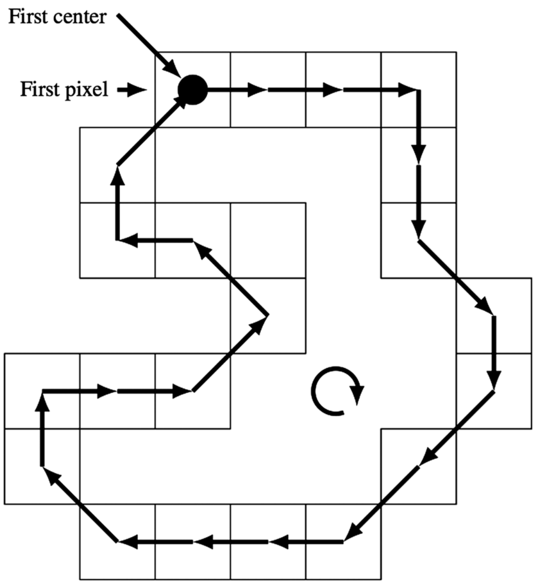


Fig. 4 The figure is covered through centers of pixels. $C_{F8} = \{0, 0, 0, 2, 2, 1, 2, 3, 3, 4, 4, 4, 5, 6, 0, 0, 7, 5, 4, 4, 6, 7\}$, $C_{AF8} = \{1, 0, 0, 2, 0, 7, 1, 1, 0, 1, 0, 0, 1, 1, 2, 0, 7, 6, 7, 2, 1\}$.

4. The first change of a discrete curve corresponds to the symbol "2" of 3OT code.
5. The chain codes have the same length.

Note 3 $F8$ and $AF8$ have the following common features (see Figs. 4 and 5):

1. Centers of contour pixels are visited.
2. The first center visited is that of the first pixel.
3. The chain codes have the same length.

The following three notes are a consequence of Table 2.

Note 4 If $F4$ or $F8$ needs to be coded/decoded, vector by vector have to be analyzed.

Note 5 If VCC or $AF8$ needs to be coded/decoded, every two consecutive vectors have to be analyzed.

Note 6 If 3OT needs to be coded/decoded, every three consecutive vectors have to be analyzed.

Note 7 As usual, $F4$ and $F8$ codes are used following clockwise direction (see Fig. 1).

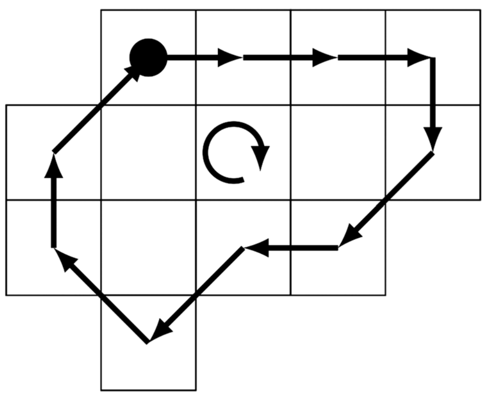


Fig. 5 The figure is covered through centers of pixels. $C_{F8} = \{0, 0, 0, 2, 3, 4, 3, 5, 6, 7\}$; $C_{AF8} = \{1, 0, 0, 2, 1, 1, 7, 2, 1, 1\}$.

Figure 4 shows an example of codification using $F8$ and $AF8$.

Definition 3 Two codes C_1 and C_2 are equivalent if it is possible to go from one code to another without visiting object shapes; in this case, we write $C_1 \equiv C_2$.

Definition 4 Let C_1 and C_2 be equivalent codes. A matrix M is called transition matrix from C_1 to C_2 if it allows to go from the code C_1 to the other code C_2. If this happens, this matrix M is denoted by ${}_{C_1}T_{C_2}$.

3 F4, VCC, and 3OT are Equivalent

In this section, we demonstrate $F4 \equiv VCC$ and $F4 \equiv 3OT$.

From linear algebra, a matrix T of r rows and s columns can be written as

$$T = (t_{pq})_{r\times s},$$

where t_{pq} represents the element of row p and column q of the matrix T.

Theorem 1. ($F4 \equiv VCC$) If $S_0 = 0$, $S_1 = 1$. $S_2 = *$, and $S_3 = 2$, then a transition matrix from $F4$ to VCC is

$${}_{F_4}T_{VCC} = (t_{pq})_{4\times 4},$$

where $t_{pq} = S_{(q-p \bmod 4)}$. Conversely, a transition matrix from VCC to $F4$ is

$${}_{VCC}T_{F4} = (t_{pq})_{4\times 3},$$

where

$$t_{pq} = \begin{cases} p+q-2 \bmod 4 & \text{if } q \in \{1,2\} \\ p+2 \bmod 4 & \text{if } q = 3 \end{cases}.$$

Proof.

- $F4$ to VCC: Assume that a figure is codified with $F4$, and the coding is

$$C_{F4} = \{C_{F4}(1), \ldots, C_{F4}(i), \ldots, C_{F4}(n)\}.$$

Let us denote by C_{VCC} the chain code we want to obtain of the same figure coded by $F4$. By Note 1 and Note 2, the first two orthogonal vectors of the first pixel compose the first code symbol, 1, of VCC, i.e., $C_{F4}(n) = 3$ and $C_{F4}(1) = 0$, and $C_{VCC}(1) = 1$. While we cover the $F4$ chain, we have to focus on two contiguous vectors. Now, taking into account every pair of vectors from $F4$ basis, we can obtain one of the symbols of VCC basis, independent of orientation. Observe that if $C_{F4}(i-1) = j$ and $C_{F4}(i) = j$, then $C_{VCC}(i) = 0$,

if $C_{F4}(i-1) = j$ and $C_{F4}(i) = j + 1 \bmod 4$, then $C_{VCC}(i) = 1$, $C_{F4}(i-1) = j$ and $C_{F4}(i) = j + 2 \bmod 4$ never happens (for this is used $S_2 = *$),

if $C_{F4}(i-1) = j$ and $C_{F4}(i) = j + 3 \bmod 4$, then $C_{VCC}(i) = 2$, where $i = 2, \cdots, n$ and $j \in \{0,1,2,3\}$. These facts can be represented with the transition matrix ${}_{F4}T_{VCC}$, which can be read like a multiplication table.

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1 & * & 2 \\ 2 & 0 & 1 & * \\ * & 2 & 0 & 1 \\ 1 & * & 2 & 0 \end{pmatrix} \end{array}$$

For example, if we have in some place of $F4$: a symbol 2 followed by 3, in VCC we have 1.

- VCC to $F4$: Assume that a figure is codified with VCC, and the coding is

$$C_{VCC} = \{C_{VCC}(1), \ldots, C_{VCC}(i), \ldots, C_{VCC}(n)\}.$$

Let us denote by C_{F4} the coding of the same figure with $F4$. For Note 1 and Note 2, $C_{F4}(n) = 3$ and $C_{F4}(1) = 0$. To find the element $C_{F4}(i)$, for $i = 2, \cdots, n$, we analyze the elements $C_{F4}(i-1)$ and $C_{VCC}(i)$. Thus, we have

if $C_{F4}(i-1) = 0$ and $C_{VCC}(i) = 0$, then $C_{F4}(i) = 0$,
if $C_{F4}(i-1) = 0$ and $C_{VCC}(i) = 1$, then $C_{F4}(i) = 1$,
if $C_{F4}(i-1) = 0$ and $C_{VCC}(i) = 2$, then $C_{F4}(i) = 3$.

Of course, this analysis can be carried out independent of the vector orientations. So, transition matrix from VCC to $F4$ ${}_{VCC}T_{F4}$ is

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{pmatrix} 0 & 1 & 3 \\ 1 & 2 & 0 \\ 2 & 3 & 1 \\ 3 & 0 & 2 \end{pmatrix} \end{array}$$

which we can also read like a multiplication table. For example, if we have $C_{F4}(i-1) = 2$ and we have $C_{VCC}(i) = 1$, then $C_{F4}(i) = 3$. ∎

Now see how to go from $F4$ to $3OT$ and vice versa.

Theorem 2. ($F4 \equiv 3OT$) A transition matrix from $F4$ to $3OT$ is

$${}_{F4}T_{3OT} = (t_{pq})_{4\times 4},$$

where

$$t_{pq} = \begin{cases} 1 & \text{if } p = q, \\ 2 & \text{if } |p-q| = 2 \\ * & \text{otherwise} \end{cases}.$$

And a transition matrix from $3OT$ to $F4$ is

$${}_{3OT}T_{F_4} = (t_{pq})_{4\times 2},$$

where $t_{pq} = p + (-1)^q \bmod 4$.

Proof.

- $F4$ to $3OT$: Assume that a figure is codified with $F4$, and the coding is

$$C_{F4} = \{C_{F4}(1), \dots, C_{F4}(i), \dots, C_{F4}(n)\}.$$

Denote by C_{3OT} the coding of the same figure with $3OT$. For Note 1 and Note 2 $C_{F4}(n) = 3$ and $C_{F4}(1) = 0$. For $i = 1, 2, \cdots, n$, we find $C_{3OT}(i)$ with the help of $C_{F4}(i+1)$, $C_{F4}(i)$, and $C_{F4}(k)$, where $k < i$ and $C_{F4}(k) \neq C_{F4}(k+1) = C_{F4}(k+2) = \cdots = C_{F4}(i)$. If $C_{F4}(1) = C_{F4}(2) = \cdots = C_{F4}(i)$, i.e., if there is no k that satisfies the condition, then $k := n$. Of course, $C_{F4}(n+1) := C_{F4}(1)$. Now see the following facts:

- If $C_{F4}(i+1) = C_{F4}(i)$, then $C_{3OT}(i) = 0$.
- On the contrary, if $C_{F4}(i+1) \neq C_{F4}(i)$, we have two options:
  - if $C_{F4}(i+1) = C_{F4}(k)$, then $C_{3OT}(i) = 1$,
  - if $C_{F4}(i+1) = C_{F4}(k) + 2 \bmod 4$, then $C_{3OT}(i) = 2$.

These facts can be represented with the transition matrix ${}_{F4}T_{3OT}$ as

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & * & 2 & * \\ * & 1 & * & 2 \\ 2 & * & 1 & * \\ * & 2 & * & 1 \end{pmatrix} \end{array}$$

which can be read like a multiplication table. For example, if we have in some place $C_{F4}(j) = 3$ and $C_{F4}(i+1) = 1$, in $3OT$, we have $C_{3OT}(i) = 2$.

- $3OT$ to $F4$: Assume that a figure is codified with $3OT$, and the coding is

$$C_{3OT} = \{C_{3OT}(1), \dots, C_{3OT}(i), \dots, C_{3OT}(n)\}.$$

Denote by C_{F4} the coding of the same figure with $F4$. For Note 1 and Note 2, $C_{F4}(n) = 3$ and $C_{F4}(1) = 0$, and to find the element $C_{F4}(i)$, for $i = 2, \cdots, n$, the element $C_{F4}(k)$ (k defined in proof of $F4$ to $3OT$) and the element $C_{3OT}(i-1)$ are analyzed.

If $C_{3OT}(i-1) = 0$, then $C_{F4}(i) = C_{F4}(i-1)$,
if $C_{3OT}(i-1) = 1$, then $C_{F4}(i) = C_{F4}(k)$,
if $C_{3OT}(i-1) = 2$, then $C_{F4}(i) = C_{F4}(k) + 2 \bmod 4$.

This study can be represented with the transition matrix ${}_{3OT}T_{F4}$ as

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} 0 & 2 \\ 1 & 3 \\ 2 & 0 \\ 3 & 1 \end{pmatrix} \end{array}$$

and taking into account that we have $C_{3OT}(i-1) = 0$, $C_{F4}(i) = C_{F4}(i-1)$. □

4 $F8$ and $AF8$ are Equivalent

In this section, we give a proof that $F8$ and $AF8$ are equivalent.

Theorem 3. ($F8 \equiv AF8$) A transition matrix from $F8$ to $AF8$ is

$${}_{F8}T_{AF8} = (t_{pq})_{8\times 8},$$

where $t_{pq} = q - p \bmod 8$ and a transition matrix from $AF8$ to $F8$ is

$${}_{AF8}T_{F8} = (t_{pq})_{8\times 8},$$

where $t_{pq} = p + q - 2 \bmod 8$.

Proof.

- $F8$ to $AF8$: Assume that a figure is codified with $F8$, and the coding is

$$C_{F8} = \{C_{F8}(1), \dots, C_{F8}(i), \dots, C_{F8}(n)\}.$$

Let us denote by C_{AF8} the coding of the same figure with $AF8$. For $i = 1, 2, \cdots, n$, we find $C_{AF8}(i)$ with the help of $C_{F8}(i)$ and $C_{F8}(i-1)$, where $C_{F8}(0) := C_{F8}(n)$. We have the following facts:

if $C_{F4}(i) = C_{F4}(i-1) + r$, with $0 \le r \le 7$,
then $C_{AF8}(i) = r$.

These facts can be represented by the transition matrix ${}_{F8}T_{AF8}$ as

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 7 & 0 & 1 & 2 & 3 & 4 & 5 \\ 5 & 6 & 7 & 0 & 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 & 0 & 1 & 2 & 3 \\ 3 & 4 & 5 & 6 & 7 & 0 & 1 & 2 \\ 2 & 3 & 4 & 5 & 6 & 7 & 0 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 \end{pmatrix} \end{array}$$

which can be read like a multiplication table. For example, if we have in some place of $F8$ 4 followed by 3, in $AF8$, we have 7.

- $AF8$ to $F8$: Assume that a figure is codified with $AF8$, and the coding is

$$C_{AF8} = \{C_{AF8}(1), \dots, C_{AF8}(i), \dots, C_{AF8}(n)\}.$$

Denote by C_{F8} the coding of the same figure with $F8$. Because $AF8$ is invariant under rotation, we can suppose that $C_{F8}(1) = 0$. To find the element $C_{F8}(i)$, for $i = 2, \cdots, n$, we analyze the elements $C_{F8}(i-1)$ and $C_{AF8}(i)$. Next conditions take place:

If $C_{AF8}(i) = r$, with $0 \le r \le 7$,
then $C_{F8}(i) = C_{F8}(i-1) + r \bmod 8$.

Because there are 8×8 different combinations of $C_{AF8}(i)$ and $C_{F8}(i-1)$, this study is represented by the transition matrix ${}_{AF8}T_{F8}$. The transition matrix can be read like a multiplication table.

$$\begin{array}{c c} & \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{array}{c} 0\\1\\2\\3\\4\\5\\6\\7 \end{array} & \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 \\ 2 & 3 & 4 & 5 & 6 & 7 & 0 & 1 \\ 3 & 4 & 5 & 6 & 7 & 0 & 1 & 2 \\ 4 & 5 & 6 & 7 & 0 & 1 & 2 & 3 \\ 5 & 6 & 7 & 0 & 1 & 2 & 3 & 4 \\ 6 & 7 & 0 & 1 & 2 & 3 & 4 & 5 \\ 7 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix} \end{array}$$

For example, if we have $C_{AF8}(i) = 3$, and $C_{F8}(i-1) = 5$, then $C_{F8}(i) = 0$. □

Example 1 In this example, we show how to use the transition matrix ${}_{F8}T_{FA8}$. The chain code C_{F8} of Fig. 5 is $C_{F8} = \{0, 0, 0, 2, 3, 4, 3, 5, 6, 7\}$, and the transition matrix ${}_{F8}T_{FA8}$ can be read like a multiplication table.

$$\begin{array}{c c} & \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{array}{c} 0\\1\\2\\3\\4\\5\\6\\7 \end{array} & \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 7 & 0 & 1 & 2 & 3 & 4 & 5 \\ 5 & 6 & 7 & 0 & 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 & 0 & 1 & 2 & 3 \\ 3 & 4 & 5 & 6 & 7 & 0 & 1 & 2 \\ 2 & 3 & 4 & 5 & 6 & 7 & 0 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 \end{pmatrix} \end{array}$$

Taking into account two consecutive symbols of Fig. 5, we have the following steps:

$C_{F8}(0) = 7$ and $C_{F8}(1) = 0$, then $C_{AF8}(1) = 1$ [position (7,0) of the table]

$C_{F8}(1) = 0$ and $C_{F8}(2) = 0$, then $C_{AF8}(2) = 0$ [position (0,0) of the table]

$C_{F8}(2) = 0$ and $C_{F8}(3) = 0$, then $C_{AF8}(3) = 0$ [position (0,0) of the table]

$C_{F8}(3) = 0$ and $C_{F8}(4) = 2$, then $C_{AF8}(4) = 2$ [position (0,2) of the table]

$C_{F8}(4) = 2$ and $C_{F8}(5) = 3$, then $C_{AF8}(5) = 1$ [position (2,3) of the table]

$C_{F8}(5) = 3$ and $C_{F8}(6) = 4$, then $C_{AF8}(6) = 1$ [position (3,4) of the table]

$C_{F8}(6) = 4$ and $C_{F8}(7) = 3$, then $C_{AF8}(7) = 7$ [position (4,3) of the table]

$C_{F8}(7) = 3$ and $C_{F8}(8) = 5$, then $C_{AF8}(8) = 2$ [position (3,5) of the table]

$C_{F8}(8) = 5$ and $C_{F8}(9) = 6$, then $C_{AF8}(9) = 1$ [position (5,6) of the table]

$C_{F8}(9) = 6$ and $C_{F8}(10) = 7$, then $C_{AF8}(10) = 1$ [position (6,7) of the table]

So

$$C_{AF8} = \{1, 0, 0, 2, 1, 1, 7, 2, 1, 1\}.$$

This result can be verified geometrically in Fig. 5.

5 All Chain Codes are Equivalent

The coded vectors of $F4$ visit outer edges of the contours, whereas those of $F8$ visit the centers of pixels, and also $F4$ is a four connected representation, whereas $F8$ is for eight connectivity. In this section, we prove one of the main results of this paper: $F4 \equiv VCC \equiv 3OT \equiv F8 \equiv AF8$.

Theorem 4. ($F4 \equiv F8$)

$${}_{F4}T_{F8} = \begin{pmatrix} 0 & \square & \ast & 7 \\ 1 & 2 & \square & \ast \\ \ast & 3 & 4 & \square \\ \square & \ast & 5 & 6 \end{pmatrix}$$

and

$${}_{F8}T_{F4} = \begin{pmatrix} 0 & 010 & 01 & 0121 & \ast & \ast & \ast & 03 \\ \square & 10 & 1 & 121 & 12 & \ast & \ast & 3 \\ \ast & 10 & 1 & 121 & 12 & 1232 & \ast & \ast \\ \ast & 0 & \square & 21 & 2 & 232 & 23 & \ast \\ \ast & \ast & \ast & 21 & 2 & 232 & 23 & 2303 \\ 30 & \ast & \ast & 1 & \square & 32 & 3 & 303 \\ 30 & 2010 & \ast & \ast & \ast & 32 & 3 & 303 \\ 0 & 010 & 01 & \ast & \ast & 2 & \square & 03 \end{pmatrix}.$$

Proof.

- $F4$ to $F8$: Let us assume that a figure is codified with $F4$, and the coding is

$$C_{F4} = \{C_{F4}(1), \ldots, C_{F4}(i), \ldots, C_{F4}(n)\}.$$

Let C_{F8} be the coding of the same figure with $F8$. For $i = 1, 2, \cdots, n$, we find $C_{F8}(i)$ with the help of $C_{F4}(i)$ and $C_{F4}(i-1)$, where $C_{F4}(0) \coloneqq C_{F4}(n)$. We have to analyze every two vectors in $F4$ and see how they become in $F8$. Next conditions take place.

If $C_{F4}(i-1) = 0$ and $C_{F4}(i) = 0$, then $C_{F8}(i) = 0$,

if $C_{F4}(i-1) = 0$ and $C_{F4}(i) = 1$, then $C_{F8}(i) = \square$(empty), since for $F8$ we are in the center of the same pixel, so we do nothing.

$C_{F4}(i-1) = 0$ and $C_{F4}(i) = 2$ never happens (for this $\ast$ is used),

if $C_{F4}(i-1) = 0$ and $C_{F4}(i) = 3$, then $C_{F8}(i) = 7$, where $i = 1, \ldots, n$. All these facts can be summarized in the next transition matrix.

$$\begin{array}{c c} & \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0\\1\\2\\3 \end{array} & \begin{pmatrix} 0 & \square & \ast & 7 \\ 1 & 2 & \square & \ast \\ \ast & 3 & 4 & \square \\ \square & \ast & 5 & 6 \end{pmatrix} \end{array}.$$

- $F8$ to $F4$: Let us assume that a figure is codified with $F8$, and the coding is

$$C_{F8} = \{C_{F8}(1), \ldots, C_{F8}(i), \ldots, C_{F8}(n)\}.$$

Denote by C_{F4} the coding of the same figure with $F4$. For Note 1 and Note 2 $C_{F4}(n) = 3$ and $C_{F4}(1) = 0$.

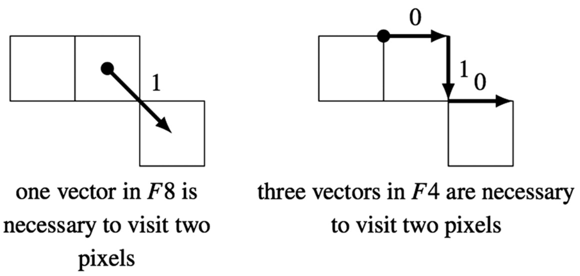


Fig. 6 More vectors are necessary in $F4$ than in $F8$ to code the same set of pixels.

For $i = 2, \cdots, n$ we find $C_{F4}(i)$ with the help of $C_{F8}(i)$ and $C_{F8}(i-1)$. We have to analyze every two vectors in $F8$ and we see how they become in $F4$. Some examples are shown below.

If $C_{F8}(i-1) = 0$ and $C_{F8}(i) = 0$, then $C_{F4}(i) = 0$,

if $C_{F8}(i-1) = 0$ and $C_{F8}(i) = 1$, then $C_{F4}(i) = 010$ because we need the vectors 0, 1, and 0 in $F4$ to visit the same pixels of vector 1 in $F8$ (Fig. 6),

$C_{F8}(i-1) = 0$ and $C_{F8}(i) = 4$ never happens,

if $C_{F8}(i-1) = 0$ and $C_{F8}(i) = 7$, then $C_{F4}(i) = 03$, where $i = 2, \cdots, n$. All these facts can be written in the following transition matrix:

	0	1	2	3	4	5	6	7
0	0	010	01	0121	*	*	*	03
1	□	10	1	121	12	*	*	3
2	*	10	1	121	12	1232	*	*
3	*	0	□	21	2	232	23	*
4	*	*	*	21	2	232	23	2303
5	30	*	*	1	□	32	3	303
6	30	~~2010~~ 3010	*	*	*	32	3	303
7	0	010	01	*	*	2	□	03

It can also be read like a multiplication table. For example, if we have $C_{F8}(i-1) = 5$ and $C_{F8}(i) = 3$, then $C_{F4}(i) = 1$. □

Theorem 12. $F4 \equiv VCC \equiv 3OT \equiv F8 \equiv AF8$

Proof. Using Theorem 1 we have $F4 \equiv VCC$, and for Theorem 2, we have $F4 \equiv 3OT$. Thanks to Theorems 3 and 4, we know that $F8 \equiv AF8$ and $F8 \equiv F4$, respectively. Thus, the assertion is true. □

6 New Code

As an important application of the transition matrix concept, we create a new code using a matrix. This code is inspired in $3OT$, but the new code is obtained by visiting the centers of pixels. Let us call the new code $AAF8$ (because it uses two angles to determine a symbol).

- $F8$ to $AAF8$: Let us assume that a figure is codified with $F8$, and the code is

$$C_{F8} = \{C_{F8}(1), \ldots, C_{F8}(i), \ldots, C_{F8}(n)\}.$$

For $i = 1, 2, \cdots, n$, we create $C_{AAF8}(i)$ with the help of $C_{F8}(i+1), C_{F8}(i)$, and $C_{F8}(k)$, where $k < i$ and $C_{F8}(k) \neq C_{F8}(k+1) = C_{F8}(k+2) = \cdots = C_{F8}(i)$. If $C_{F8}(1) = C_{F8}(2) = \cdots = C_{F8}(i)$, kcn. It is defined that $C_{F8}(n+1) := C_{F8}(1)$. Thus,

– if $C_{F8}(i+1) = C_{F8}(i)$, then $C_{AAF8}(i) = 8$.
– On the contrary, if $C_{F8}(i+1) \neq C_{F8}(i)$, then consider the next cases.

If $C_{F8}(i+1) = C_{F8}(k) + r \bmod 8$, with $0 \leq r \leq 7$, then $C_{AAF8}(i) = r$.

So, $C_{AAF8}(i)$ arises from the following transition matrix ${}_{AF8}T_{AAF8}$:

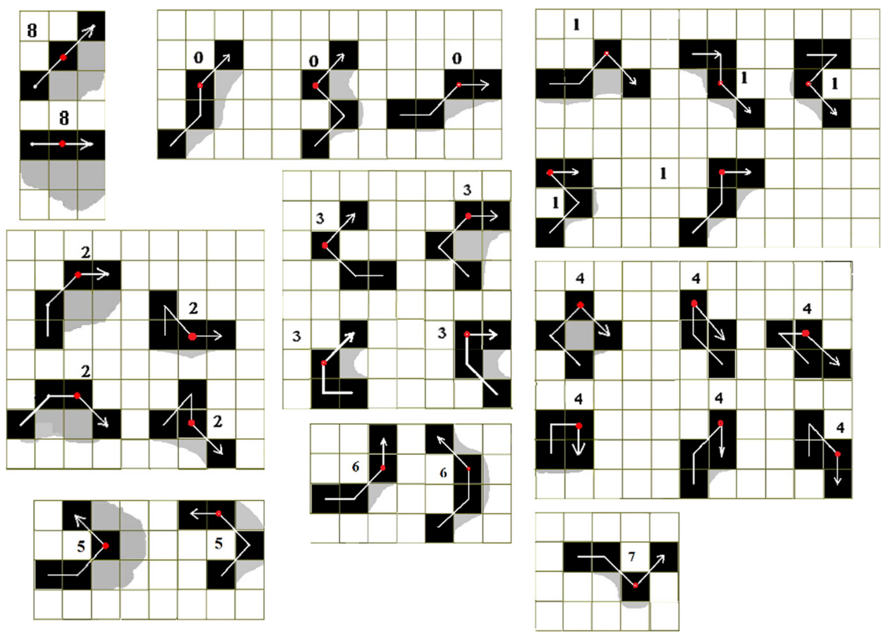


Fig. 7 The set of allowed vector arrays of our proposed $AAF8$ code.

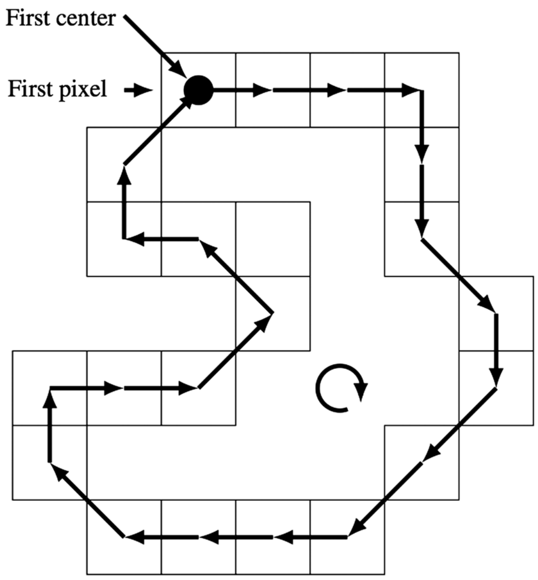


Fig. 8 The figure is covered through the centers of pixels. $C_{F8} = \{0,0,0,2,2,1,2,3,3,4,4,4,5,6,0,0,7,5,4,6,7\}$, $C_{AAF8} = \{8,8,3,8,1,0,2,8,2,8,8,2,2,3,8,1,5,5,1,3,2\}$.

$$\begin{array}{c} \\ 0\\1\\2\\3\\4\\5\\6\\7 \end{array}\begin{array}{c} \begin{array}{cccccccc} 0&1&2&3&4&5&6&7 \end{array} \\ \begin{pmatrix} 0&1&2&3&4&5&6&7\\ 7&0&1&2&3&4&5&6\\ 6&7&0&1&2&3&4&5\\ 5&6&7&0&1&2&3&4\\ 4&5&6&7&0&1&2&3\\ 3&4&5&6&7&0&1&2\\ 2&3&4&5&6&7&0&1\\ 1&2&3&4&5&6&7&0 \end{pmatrix} \end{array}$$

This matrix is written like that which transforms $F8$ to $AF8$ because both show what happens between reference and change vectors. However, in this transformation, a support vector is involved, and no matter the direction of support vector, we consider that reference and change vectors give the codification of the shape.

Since the new code is

$$C_{AAF8} = \{C_{F8}(n), C_{F8}(1), C_{AAF8}(3), \ldots, C_{AAF8}(i), \ldots, C_{AAF8}(n)\},$$

where $C_{AAF8}(1) = C_{F8}(n)$ and $C_{AAF8}(2) = C_{F8}(1)$, we should keep the information of $C_{F8}(n)$ and $C_{F8}(1)$ to find the transition matrix ${}_{AAF8}T_{F8}$. As can be observed, given the first two known $F8$ vectors, a third vector (the change

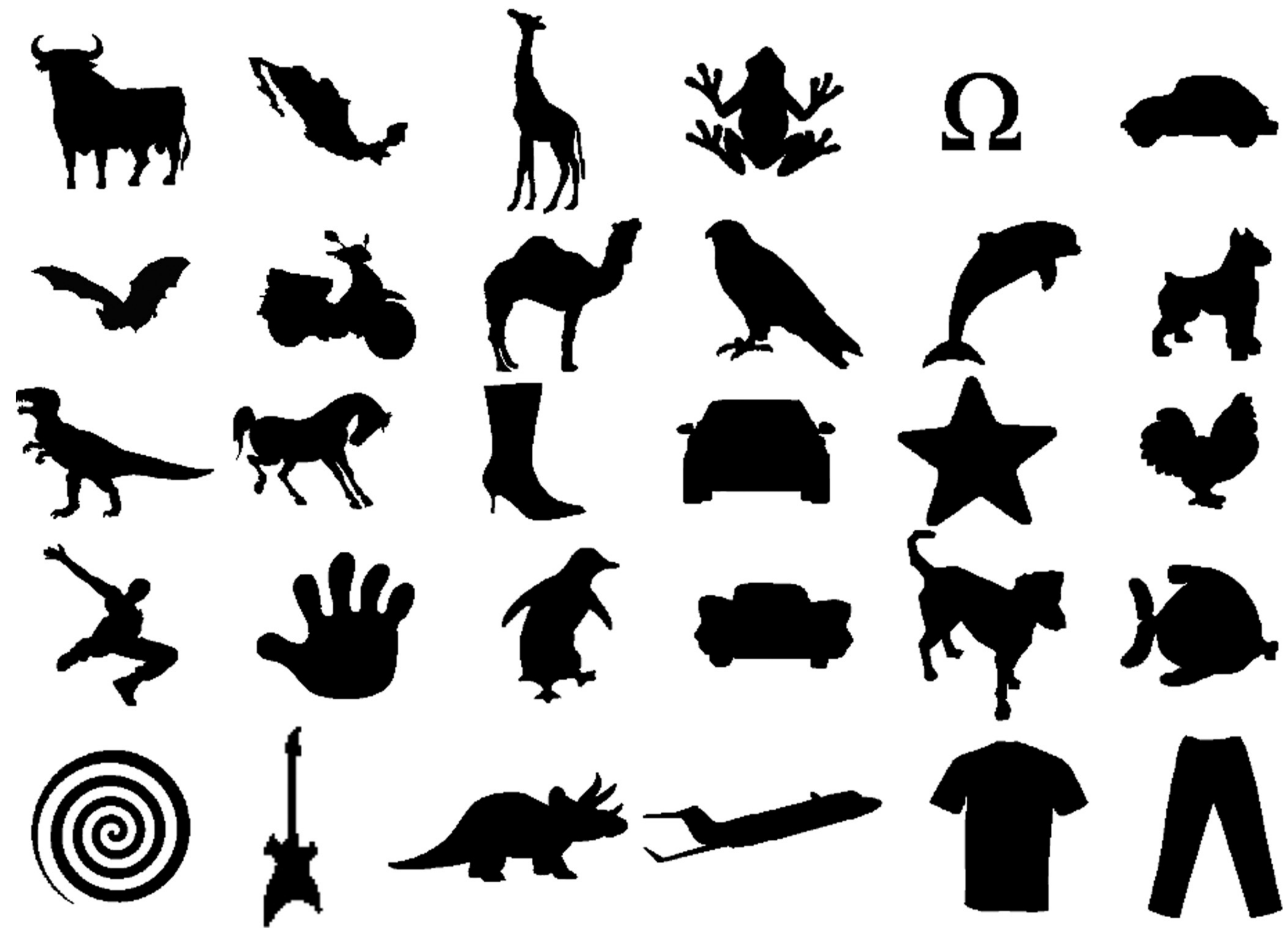


Fig. 9 Images encoded using $F4, VCC, 3OT, F8, AF8$, and $AAF8$.

vector) composes one of the symbols given in Fig. 7, which shows a graphical aspect of the different symbols of the new code.

A very important observation arises: to consider the first change of direction, it is necessary to know the first two vectors $F8(n)$ and $F8(1)$, reference and support, respectively, so that the third vector (of the change) is oriented with a certain angle with respect to the reference, and codify the change with one of the eight symbols of Fig. 7 (clearly, symbol 8 actually does not represent change of direction). In the following change to encode, the reference vector becomes in what was of support $F8(1)$ and the new vector of change has a certain angle regarding reference vector. The behavior of the support vector was already coded in the previous step, so the information of the actual path of the shape is never lost.

- $AAF8$ to $F8$: Assume that a figure is codified with $AAF8$, and the code is

$$C_{AAF8} = \{C_{F8}(n), C_{F8}(1), C_{AAF8}(3), \ldots, C_{AAF8}(i), \ldots, C_{AAF8}(n)\}.$$

Let C_{F8} be the coding of the same figure with $F8$. To find the element $C_{F8}(i)$, for $i = 2, \cdots, n-1$, consider, again, $k < i$ and the condition that $C_{F8}(n)$ and $C_{F8}(1)$ are known.

$$\text{If } C_{AAF8}(i) = r, \quad \text{with } 0 \le r \le 7,$$
$$\text{then } C_{F8}(i+1) = C_{F8}(k) + r \bmod 8.$$

Of course, there are also eight different values for $C_{F8}(k)$, so there is a complete set of 64 combinations of $C_{AAF8}(i)$ and $C_{F8}(k)$ to obtain a symbol $C_{F8}(i+1)$, which can be written in terms of the next table.

$$\begin{array}{c} \\ 0\\1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 \\ 2 & 3 & 4 & 5 & 6 & 7 & 0 & 1 \\ 3 & 4 & 5 & 6 & 7 & 0 & 1 & 2 \\ 4 & 5 & 6 & 7 & 0 & 1 & 2 & 3 \\ 5 & 6 & 7 & 0 & 1 & 2 & 3 & 4 \\ 6 & 7 & 0 & 1 & 2 & 3 & 4 & 5 \\ 7 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix} \end{array}$$

Finally, if we have $C_{AAF8}(i) = 8$, then $C_{F8}(i+1) = C_{F8}(i)$.

Each row of the matrix represents C_{AAF8} symbol, whereas the columns represent C_{F8}.

The complete set of vector arrays represented by AAF8 symbols appear in Fig. 7.

The difference of this equivalence with that of $F8$ to $AF8$ is that in this a support vector is involved, and this vector can have one of the eight different directions of $F8$ code.

Figure 8 shows an example of codification using $F8$ and $AAF8$.

To assign a symbol to each change direction in Fig. 8, we can be assisted by Fig. 7. Observe that we can choose the vector patterns that best match the contour in the position we are coding depending on the angle given mainly by

Table 3 Original sizes, number of pixels, and lengths of the different codes.

Object	Size	Number of pixels	l_4	l_8
Bull	286 × 278	38,115	2122	1651
Map	310 × 213	21,747	1560	1120
Giraffe	197 × 505	36,945	2866	2316
Frog	183 × 151	12,789	1704	1172
Omega	187 × 184	14,565	1492	1178
VW	447 × 204	57,337	1388	1058
Bat	550 × 265	53,096	2054	1444
Motorcycle	853 × 695	289,672	5044	3689
Camel	293 × 285	32,202	1796	1330
Eagle	226 × 213	19,363	1214	816
Dolphin	305 × 284	27,613	1394	1022
Pitbull	220 × 246	26,126	1328	1019
Trex	431 × 287	34,053	2208	1651
Horse	238 × 174	11,886	1836	1277
Boot	269 × 359	45,787	1458	1144
Car	309 × 204	47,973	1202	1010
Star	259 × 244	28,244	1178	860
Chicken	184 × 180	17,946	966	678
Jump	203 × 244	13,776	1378	947
Hand	309 × 287	47,132	1864	1371
Penguin	122 × 160	8500	831	564
Car 2	406 × 221	1085	1378	1179
Dog	230 × 275	23,805	1562	1152
Fish	221 × 171	22,037	1158	841
Spiral	187 × 181	14,501	3128	2213
Guitar	24 × 77	584	240	186
Trice	356 × 164	25,758	1460	1056
Plane	355 × 109	13,306	1216	932
T-shirt	191 × 206	23,890	860	682
Jeans	220 × 297	35,541	1410	1121

the change direction vector regarding the reference vector and taking into account if the represented vector pattern of the contour points to the right or to the left of the direction of travel.

7 Evaluation and Results

We tested our method on a sample of shapes. We evaluated the different codes over 30 different irregular shapes that appear in Fig. 9, obtained from a repository,[23] whose sizes and number of pixels are given in Table 3. The images shown in Fig. 9 were coded using $F4$, VCC, $3OT$, $F8$, $AF8$, and $AAF8$.

Let l_4 be the length (i.e., the number of symbols in the chains) of the $F4$, VCC, and $3OT$ chain codes, and l_8 the length of $F8$, $AF8$, and $AAF8$ chains. Table 3 also presents the lengths of the chain codes.

A transition matrix makes it possible to go from any chain code to another independently if it is to represent four or eight neighborhood.

If the basic codes are equivalent, they have different information content. However, since there is a conflict between the code representation in four or eight neighborhood when recognition tasks are carried out, the feasibility in the representation should be taken into account when looking for patterns within chains.

8 Conclusions and Further Work

We have obtained a new relative code composed of three vectors: a reference, support, and change direction vector for eight connectivity.

A method to obtain the main codes of the literature has been developed. This method allows us to go from one code to another. So we have demonstrated that all basic codes are equivalent, although each of them was proposed independently.

We have observed a missing code, of three vectors, to complete the schema. Such a code is composed of three vectors for eight connectivity, which we have called $AAF8$. This new code is relative and two angles are involved in its construction.

The main characteristics of $AAF8$ are as follows: it is invariant under rotation transformation and as the other eight connected codes, $AAF8$ has the smallest length chain.

Just as with previous codes, future works should be implemented for analysis and recognition tasks and, also, to find and compare derived codes from $AAF8$.

Acknowledgments

We would like to thank Universidad Autónoma de Aguascalientes under grant PIINF14-2.

References

1. H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.* **EC-10**(2), 260–268 (1961).
2. A. Akimov, A. Kolesnikov, and P. Franti, "Lossless compression of map contours by context tree modeling of chain codes," *Pattern Recognit.* **40**(3), 944–952 (2007).
3. S. Blackburn, "Extraction of color region boundaries," in *IAPR Workshop on Machine Vision Applications*, pp. 63–66, MVA'92, Tokyo, Japan (1992).
4. L. Cederberg, "Chain-link coding and segmentation for raster scan devices," *Comput. Graphics Image Process.* **10**(3), 224–234 (1979).
5. T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD (1982).
6. G. Wilson, "Properties of contour codes," *IEE Proc. Vis., Image Signal Process.* **144**(3), 145–149 (1997).
7. H. Sánchez-Cruz and M. A. Rodríguez-Díaz, "Binary document image compression using a three-symbol grouped code dictionary," *J. Electron. Imaging* **21**(2) 023013 (2012).
8. A. J. Pinho, "Adaptative context-based arithmetic coding of arbitrary contour maps," *IEEE Signal Process. Lett.* **8**(1), 4–6 (2001).
9. F. Arrebola and F. Sandoval, "Corner detection and curve segmentation by multi resolution chain-code linking," *Pattern Recognit.* **38**(10), 1596–1614 (2005).
10. H. Beus and S. Tiu, "An improved corner detection algorithm based on chain coded plane curves," *Pattern Recognit.* **20**(3), 291–296 (1987).
11. H. Freeman and L. S. Davis, "A corner-finding algorithm for chain-coded curves," *IEEE Trans. Comput.* **C-26**(3), 297–303 (1977).
12. H. Sánchez-Cruz, "A proposal method for corner detection with an orthogonal three-direction chain code," in *Advanced Concepts for Intelligent Vision Systems Lecture Notes in Computer Science*, J. Blanc-Talon et al., Ed., Vol. 4179, pp. 161–172, Springer Berlin Heidelberg (2006).
13. H. Sánchez-Cruz and E. Bribiesca, "Polygonal approximation of contour shapes using corner detectors," *J. Appl. Res. Technol.* **7**(3), 275–291 (2009).
14. S. Hoque, K. Sirlantzis, and M. C. Fairhust, "A new chain-code quantization approach enabling high performance handwriting recognition based on multi-classifier schemes," in *IEEE Proc. of the Seventh Int. Conf. on Document Analysis and Recognition*, pp. 834–838, IEEE Computer Society, Washington, DC (2003).
15. H. Sánchez-Cruz, "Proposing a new code by considering pieces of discrete straight lines in contour shapes," *J. Vis. Commun. Image Represent.* **21**(4), 311–324 (2010).
16. J. H. Sossa-Azuela et al., "Computing the Euler number of a binary image based on a vertex codification," *J. Appl. Res. Technol.* **11**(3), 360–370 (2013).
17. E. Bribiesca, "A new chain code," *Pattern Recognit.* **32**(2), 235–251 (1999).
18. H. Sánchez-Cruz and R. M. Rodriguez-Dagnino, "Compressing bilevel images by means of a three-bit chain code," *Opt. Eng.* **44**(9), 097004 (2005).
19. Y. Kui-Liu and B. Zalik, "An efficient chain code with Huffman coding," *Pattern Recognit.* **38**(4), 553–557 (2005).
20. H. Sánchez-Cruz, E. Bribiesca, and R. M. Rodriguez-Dagnino, "Efficiency of chain codes to represent binary objects," *Pattern Recognit.* **40**(6), 1660–1674 (2007).
21. Y. Kui-Liu et al., "Compressed vertex chain codes," *Pattern Recognit.* **40**(11), 2908–2913 (2007).
22. H. Sánchez-Cruz, M. López-Cruces, and H. Puga, "A proposal modification of the $3OT$ chain code," in *Proc. of the 10th IASTED Int. Conf. on Computer Graphics and Imaging*, pp. 6–11, ACTA Press, Anaheim, CA (2008).
23. Images, Bilevel_images.zip, https://sites.google.com/site/herssan/images.

Hermilo Sánchez-Cruz received his PhD in computer science in 2002 from the Universidad Nacional Autónoma de México (UNAM). Also, he was an assistant researcher with the Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas at UNAM in 1998. Currently, he is a full-time professor with the Universidad Autónoma de Aguascalientes in México (UAA). His areas of interest are pattern recognition, image compression, bi-dimensional and three-dimensional (3-D) image recognition.

Hiram H. López-Valdez received his MSc in mathematics in 2010 from Centro de Investigacion y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN). Currently, he is a PhD candidate, also from CINVESTAV-IPN. He is a professor at UAA, where he teaches undergraduate courses in image processing, pattern recognition, and mathematics. He is interested in solving problems of features selection of objects in images, as well as topology aspects of two-dimensional and 3-D objects.