IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. I (Mar-Apr. 2016), PP 29-35
www.iosrjournals.org

Direction-Length Code (DLC) To Represent Binary Objects

Dr G D Jasmin, M G Justin Raj
Assistant Professor Institute of Aeronautical Engineering Telangana Dundigal, Hyderabad
Assistant Professor Annai Vailankanni College of Engineering Pottalkulam, K K Dist Tamil Nadu

Abstract: *More and more images have been generated in digital form around the world. Efficient way of description and classification of objects is a well needed application to identify the objects present in images. Shape is an important visual feature of an image. Searching for images using shape features has also attracted much attention. In this paper the Direction-Length Code which is also called a knowledge vector that gives the information about the direction and length of pixels in every direction in an object is generated to represent objects. The formal language attempts to simulate the structural and hierarchical nature of the human vision system. Here, sentences are strings of symbols and languages correspond to pattern class. The patterns over a 3 x 3 array of vertices have been generated which form the basic alphabet of the digital picture language. That is, one can visualize any digital image as a spatial distribution of these patterns. The Direction-Length code (DLC) compresses the bi-level images by preserving the information at the same time allowing a considerable amount of data reduction. Furthermore, these codes acts as a standard input format for numerous shape-analysis algorithms.*

I. Introduction

Pattern recognition could be formally defined as categorization of input data into identifiable classes via extraction of significant features or attributes of the data from the background of irrelevant detail.

The encoding efficiency to represent shapes of objects is very important for image storage and transmission [1]. It is also important for shape analysis and shape recognition of objects in pattern recognition. Various methods such as colour based representation, texture based representation, shape based representation and appearance based representation have been developed to represent objects [2-4]. More studies on these methods are being carried out and advanced techniques are being developed [5,6]. Shape based representation of objects gives more clarity about the shape of the objects that acts as a more unique and important feature of objects for the further identification process [7-9].

II. Object Recognition System

The problem of object recognition is divided into the following sub problems:

1. Image pre-processing
2. Object description
3. Image reconstruction
4. Classification

The input image to the system may be a colour image, a gray scale image or a binary image. The contours are extracted as they give the outlines of shapes of the objects present in the image. Object description involves running through the contour pixels and finding a code to represent the object. This involves finding the length of the contour in every possible direction.

Syntactic pattern recognition is inspired by the phenomenon that composition of a natural scene is an analogue to the composition of a language, that is, sentences are built up from phrases, phrases are built up from words and words are built up from alphabets, etc. [10]. The matching between shapes can use string matching by finding the minimal number of edit operations to convert one string into another. A more general method is to formulate the representation as a string grammar. Each primitive is interpreted as a alphabet of some grammar, where a grammar is a set of rules of syntax that govern the generation of sentences formed from symbols of the alphabet. The set of sentences generated by a grammar G is called its language and is denoted as L(G). Here, sentences are strings of symbols (which in turn represent patterns), and languages correspond to pattern class. After grammars have been established, the matching is straightforward. For a sentence representing an unknown shape, the task is to decide in which language the shape represents a valid sentence. Syntactic shape analysis is based on the theory of formal language [11]. It attempts to simulate the structural and hierarchical nature of the human vision system. However, it is not practical in general applications due to the fact that it is not possible to infer a pattern of grammar which can generate only the valid patterns. In addition, this method needs a priori knowledge for the database in order to define code words or alphabets.

III. Patterns Over 3x3 Array Of Pixels

Let us consider an object S present in an image f(x,y). A pixel p at coordinates (x,y) has 4 horizontal and vertical neighbours called direct neighbours and 4 diagonal neighbours whose coordinates are given by (x+1,y), (x-1,y), (x,y+1), (x,y-1), (x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1).

The symbols used to represent the 8 possible directions are R, DR, D, DL, L, UL, U and UR. Note that R, D, L and U are elementary symbols and DR, DL, UL and UR are composite symbols of the alphabet. The directions denoted by these symbols are shown in Figure 1.

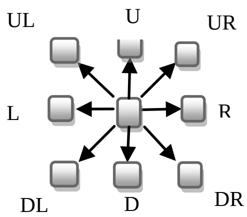


Fig 1: 3 X 3 Pixel representation and the Direction Codes

The direct neighbours of pixel *p* is the set $N_{dt}(p) = \{6,8,4,2\}$

The diagonal neighbours of pixel *p* is the set $N_{dl}(p) = \{9,7,1,3\}$

The 8-neighbours of pixel *p* is the set $N_8(p) = N_{dt}(p) \cup N_{dl}(p)$

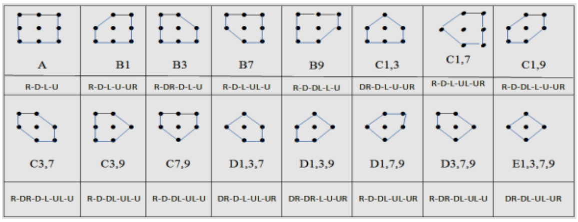
Concave patterns can be constructed by removing the pixels in the direct neighbours. Removing pixels 2,4,6 and 8 the total number of concave patterns obtained are $\sum_{i=0}^{4} 4Ci = 16$.

In the same way, the Convex patterns can be constructed by removing the pixels in the diagonal neighbours. So, removing pixels 1,3,7 and 9 the total number of concave patterns obtained are $\sum_{i=0}^{4} 4Ci = 16$.

Hybrid patterns can be constructed by removing pixels from direct and diagonal neighbours. Thus, the total number of hybrid patterns removing pixels 1,2,3,4,6,7,8 and 9 are $\sum_{i=0}^{4} 8Ci = 256$.

The convex contour patterns constructed from pattern A are shown below.

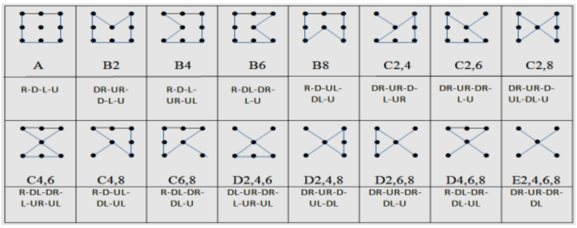
Table 1: Convex Patterns



A	B1	B3	B7	B9	C1,3	C1,7	C1,9
R-D-L-U	R-D-L-U-UR	R-DR-D-L-U	R-D-L-UL-U	R-D-DL-L-U	DR-D-L-U-UR	R-D-L-UL-UR	R-D-DL-L-U-UR
C3,7	C3,9	C7,9	D1,3,7	D1,3,9	D1,7,9	D3,7,9	E1,3,7,9
R-DR-D-L-UL-U	R-D-DL-UL-U	R-D-DL-UL-U	DR-D-L-UL-UR	DR-DR-L-U-UR	R-D-DL-UL-UR	R-DR-DL-UL-U	DR-DL-UL-UR

Note that the direction codes for all 16 convex contour patterns are given below the respective patterns. For instance, the direction code for the pattern A is R-D-L-U and the direction code for the pattern B3 is R-DR-D-L-U.

Table 2: Concave Patterns



A	B2	B4	B6	B8	C2,4	C2,6	C2,8
R-D-L-U	DR-UR-D-L-U	R-D-L-UR-UL	R-DL-DR-L-U	R-D-UL-DL-U	DR-UR-D-L-UR	DR-UR-DR-L-U	DR-UR-D-UL-DL-U
C4,6	C4,8	C6,8	D2,4,6	D2,4,8	D2,6,8	D4,6,8	E2,4,6,8
R-DL-DR-L-UR-UL	R-D-UL-DL-UL	R-DL-DR-DL-U	DL-UR-DR-L-UR-UL	DR-UR-D-UL-DL	DR-UR-DR-DL-U	R-DL-DR-DL-UL	DR-UR-DR-DL

The 15 concave patterns constructed from the pattern A are shown in table 2. One can also construct a total of 225 more patterns from the pattern A as shown below and determine their direction codes. Thus a total of 256 patterns could be generated in the 3X3 array of vertices.

Table 3: Hybrid Patterns

A	B9,2	B9,4	B9,6	B9,8	C9,2,4	C9,2,6	C9,2,8
R-D-L-U	DR-UR-D-DL-L-U	R-D-DL-L-UR-U	R-DL-D-L-U	R-D-L-DL-U	DR-UR-D-DL-L-UR	DR-UR-D-L-U	DR-UR-D-L-DL-U
C9,4,6	C9,4,8	C9,6,8	D9,2,4,6	D9,2,4,8	D9,2,6,8	D9,4,6,8	E9,2,4,6,8
R-DL-D-L-UR-UL	R-D-L-DL-UL	R-DL-U	DR-UR-D-L-UR	DR-UR-D-R-DL	DR-UR-DL-U	R-DL-DL-UL	DR-UR-DL

A	B1,2	B1,4	B1,6	B1,8	C1,2,4	C1,2,6	C1,2,8
R-D-L-U	D-L-U-R-UR	R-D-L-UR-U	R-DL-DR-L-U-R	R-D-UL-DL-U-UR	D-L-UR	DL-DR-L-U-R	D-UL-DL-U-R-UR
C1,4,6	C1,4,8	C1,6,8	D1,2,4,6	D1,2,4,8	D1,2,6,8	D1,4,6,8	E1,2,4,6,8
R-DL-DR-L-UR-U	R-D-UL-DL-U	R-DL-DR-DL-U-R-U	DL-DR-L-UR	D-UL-DL-UR	DL-DR-DL-U-R	R-DL-DR-DL-U	DL-DR-DL

A	B3,2	B3,4	B3,6	B3,8	C3,2,4	C3,2,6	C3,2,8
R-D-L-U	DR-R-D-L-U	R-DR-D-L-UR-UL	R-D-DR-L-U	R-DR-D-UL-DL-U	DR-R-D-L-UR	DR-L-U	DR-R-D-UL-DL-U
C3,4,6	C3,4,8	C3,6,8	D3,2,4,6	D3,2,4,8	D3,2,6,8	D3,4,6,8	E3,2,4,6,8
R-D-DR-L-UR-UL	R-DR-D-UL-DL-UL	R-D-DR-DL-U	DR-DR-L-UR	DR-R-D-UL-DL	DR-DR-DL-U	R-D-DR-DL-UL	DR-DR-DL

A	B7,2	B7,4	B7,6	B7,8	C7,2,4	C7,2,6	C7,2,8
R-D-L-U	DR-UR-D-L-UL-U	R-D-L-U-UL	R-DL-DR-L-UL-U	R-D-UL-L-U	DR-UR-D-L-U	DR-UR-DR-L-U-L-U	DR-UR-D-UL-L-U
C7,4,6	C7,4,8	C7,6,8	D7,2,4,6	D7,2,4,8	D7,2,6,8	D7,4,6,8	E7,2,4,6,8
R-DL-DR-L-U-UL	R-D-UL	R-DL-DR-L-U	DR-UR-DR-L-U	DR-UR-D-UL	DR-UR-DR-L-U	R-DL-DR-UL	DR-UR-DR

A	B1,3,2	B1,3,4		B1,3,8	C1,3,2,4	C1,3,2,6	C1,3,2,8
R-D-L-U	R-D-L-U	D-R-D-L-UR	D-DR-L-U-UR	D-R-D-UL-DL-U-R	R-D-L-UR	R-DR-L-U	R-D-UL-DL-U
C1,3,4,6	C1,3,4,8	C1,3,6,8	D1,3,2,4,6	D1,3,2,4,8	D1,3,2,6,8	D1,3,4,6,8	E1,3,2,4,6,8
D-DR-L-UR	DR-D-UL-DL-U	D-UR-DL-U-UR	DR-L-UR	R-D-UL-DL	R-DR-DL-U	D-DR-DL	DR-DL

A	B1,7,2	B1,7,4	B1,7,6	B1,7,8	C1,7,2,4	C1,7,2,6	C1,7,2,8
R-D-L-U	D-L-UL-R-UR	R-D-L-U	R-DL-DR-L-UL-UR	R-D-L-U-L-UR	D-L-U-DR	DL-DR-L-UL-R	D-UL-L-UR
C1,7,4,6	C1,7,4,8	C1,7,6,8	D1,7,2,4,6	D1,7,2,4,8	D1,7,2,6,8	D1,7,4,6,8	E1,7,2,4,6,8
R-DL-DR-L-U	R-D-UL-U	R-DL-DR-L-UR	DL-DR-L-U	D-UL-UR	DL-DR-L	R-DL-DR-U	DL-DR

A	B1,9,2	B1,9,4	B1,9,6	B1,9,8	C1,9,2,4	C1,9,2,6	C1,9,2,8
R-D-L-U	DR-UR-D-L-U	R-D-DL-L-UR-U	R-DL-D-L-U-UR	R-D-L-DL-U-DR	D-DL-L-UR-UR	DL-D-L-U-R	D-L-DL-U-R-UR
C1,9,4,6	C1,9,4,8	C1,9,6,8	D1,9,2,4,6	D1,9,2,4,8	D1,9,2,6,8	D1,9,4,6,8	E1,9,2,4,6,8
R-DL-D-L-UR-U	R-D-L-DL-U	R-DL-U-DR	DL-D-L-UR	D-L-DL-UR	DL-DL-U-R	R-DL-DL-U	DL

A	B3,7,2	B3,7,4	B3,7,6	B3,7,8	C3,7,2,4	C3,7,2,6	C3,7,2,8
R-D-L-U	R-D-L-UL-U	R-DR-D-L-U-UL	R-D-DR-L-UL-U	R-DR-D-UL-L-U	DR-R-D-L-U	DR-L-UL-U	DR-R-D-UL-L-U
C3,7,4,6	C3,7,4,8	C3,7,6,8	D3,7,2,4,6	D3,7,2,4,8	D3,7,2,6,8	D3,7,4,6,8	E3,7,2,4,6,8
R-D-DR-L-U-UL	R-DR-D-UL	R-D-DR-L-U	DR-DR-L-U	DR-R-D-UL	DR-DR-L-U	R-D-DR-UL	DR

A	B3,9,2	B3,9,4	B3,9,6	B3,9,8	C3,9,2,4	C3,9,2,6	C3,9,2,8
R-D-L-U	DR-R-DL	R-DR-DL-L-UR-UL	R-D-L-U	R-DR-L-DL-U	DL-R-DL-L-UR	DR-D-L-U	DR-R-DL-U
C3,9,4,6	C3,9,4,8	C3,9,6,8	3,9,2,4,6	D3,9,2,4,8	D3,9,2,6,8	D3,9,4,6,8	E3,9,2,4,6,8
R-D-L-UR-UL	R-DR-L-DL-UL	R-D-DL-U	DR-D-L-UR	DR-R-DL	DR-DL-U	R-D-DL-UL	DR-DL

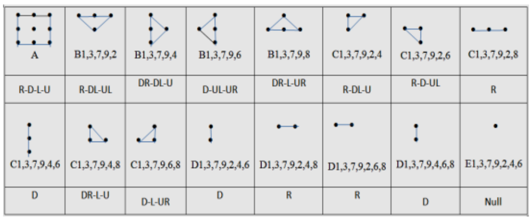
A	B7,9,2	B7,9,4	B7,9,6	B7,9,8	C7,9,2,4	C7,9,2,6	C7,9,2,8
R-D-L-U	DR-DR-D-DL-UL-U	R-D-DL-U-UL	R-DL-D-UL-U	R-D-L-U	DR-UR-D-DL-U	DR-UR-D-UL-U	DR-UR-D-L-U
C7,9,4,6	C7,9,4,8	C7,9,6,8	D7,9,2,4,6	D7,9,2,4,8	D7,9,2,6,8	D7,9,4,6,8	E7,9,2,4,6,8
R-DL-D-UL	R-D-L-UL	R-DL-L-U	DR-UR-D	DR-UR-D-L	DR-UR-L-U	R-DL-UL	DR-UR

A	B1,3,7,2	B1,3,7,4	C1,3,7,6	C1,3,7,8	C1,3,7,2,4	C1,3,7,2,6	C1,3,7,2,8
R-D-L-U	R-D-L-UL	DR-D-L-U	D-DR-L-UL-UR	DR-D-UL-L-UR	R-D-L-U	R-DR-L-UL	R-D-UL-L
C1,3,7,4,6	C1,3,7,4,8	C1,3,7,6,8	D1,3,7,2,4,6	D1,3,7,2,4,8	D1,3,7,2,6,8	D1,3,7,4,6,8	E1,3,7,2,4,6,8
D-DR-L-U	D-R-D-UL	D-DR-L-UR	DR-L-U	R-D-UL	R-DR	D	DR

A	B1,3,9,2	B1,3,9,4	B1,3,9,6	B1,3,9,8	C1,3,9,2,4	C1,3,9,2,6	C1,3,9,2,8
R-D-L-U	R-DL-L-U	DR-DL-L-UR-U	D-L-U-UR	DR-L-DL-U-R-U	R-DL-L-UR	R-D-L-U	R-R-DL-U
C1,3,9,4,6	C1,3,9,4,8	C1,3,9,6,8	D1,3,9,2,4,6	D1,3,9,2,4,8	D1,3,9,2,6,8	D1,3,9,4,6,8	E1,3,9,2,4,6,8
D-D-L-UR	DR-L-DL-U	D-DL-U-UR	D-L-UR	L-DL	R-DL-U	D-DL	DL

A	B1,7,9,2	B1,7,9,4	B1,7,9,6	B1,7,9,8	C1,7,9,2,4	C1,7,9,2,6	C1,7,9,2,8
R-D-L-U	D-DL-UL-R-UR	R-D-DL-U	R-DL-DR-UL-UR	R-D-L-UR	D-DL-U-UR	DL-D-UL-R	D-L-UR
C1,7,9,4,6	C1,7,9,4,8	C1,7,9,6,8	D1,7,9,2,4,6	D1,7,9,2,4,8	D1,7,9,2,6,8	D1,7,9,4,6,8	E1,7,9,2,4,6,8
R-DL-D-U	R-D-L-U	R-DL-L-UR	DL-D	D-L-UR	DL-L	R-DL-U	DL

A	B3,7,9,2	B3,7,9,4	B3,7,9,6	B3,7,9,8	C3,7,9,2,4	C3,7,9,2,6	C3,7,9,2,8
R-D-L-U	DR-R-DL-UL-U	R-DR-DL-U-UL	R-D-UL-U	R-DR-L-U	DR-R-DL-U	DR-D-L-U	DR-R-L-U
C3,7,9,4,6	C3,7,9,4,8	C3,7,9,6,8	D3,7,9,2,4,6	D3,7,9,2,4,8	D3,7,9,2,6,8	D3,7,9,4,6,8	E3,7,9,2,4,6,8
R-D-UL	R-DR-L-UL	R-D-L-U	DR-D	DR-R	DR-L-U	R-D-UL	DR



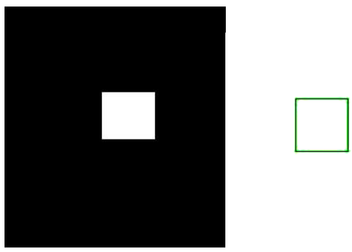
All these 256 patterns form the basic alphabet of what we call as a digital picture language. That is, one can visualize any digital image as a spatial distribution of these patterns. Once such patterns are generated for a given image, it can then be used for the successful classification of objects.

IV. Direction Length Code (DlC) Of Objects

The connectivity between the pixels is an important concept used in the field of object recognition. Two pixels are said to be connected, if they are adjacent in some sense and their intensity levels satisfy a specified criterion of similarity. If pixel p with coordinates (x,y) and pixel q with coordinates (s,t) are pixels of object (image subset) S, a path from p to q is a sequence of distinct pixels with coordinates

$(x_0, y_0), (x_1, y_1), \ldots\ldots\ldots.., (x_n, y_n)$,

where $(x_0, y_0) = (x,y)$ and $(x_n, y_n) = (s,t)$, (x_i, y_i) is adjacent to (x_{i-1}, y_{i-1}), $1 \leq i \leq n$, and n is the length of the path. If there a path exists from p to q consisting of pixels in S, then p is said to be connected to q. The gap of one or two pixels marks the presence of the next component in the same object or the beginning of the next object which can be then found by analysing the relationship among the components. Figure 2(c) shows the vector code generated for the square shown in figure 2(a).



(a) Image of a square (b) contour map

<104,109>/R81*D81*L81*U80*/<105,109>#

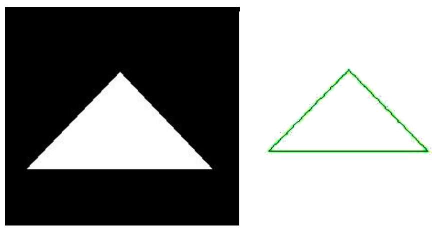
(c) Vector code

Figure 2: The DLC of a square

V. Generation Of Normalized Vector Code

It is important to note that, when the diagonal lines appear in the image, they do not appear as the combination of only diagonal sides. The direction codes D and R appear with DR lines, D and L appear with DL, U and L appear with UL and U and R direction codes appear with UR.

Consier the image of a triangle and the corresponding vector code shown below.



(a): Binary Image (b): Contour

Figure 3: Image of a triangle

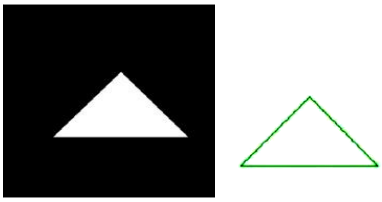
The DLC generated is
<75,121>/D1*DR2*D1*DR3*R1*DR1*D1*DR3*D1*DR3*D1*R7*D1*DR5*D1*DR2*D1*DR3*R1*DR2*D1*DR1*D1*DR2*DR4*D2*DR2*R1*DR4*D2*DR7*D2*DR1*R1*DR5*D1*DR1*D1*DR8*D1*DR4*D1*DR3*D1*DR3*R1*DR1*D1*DR3*D1*DR2*L175*UR3*U1*UR2*U1*UR2*R1*UR3*U1*UR2*U1*UR7*U

1*UR7*U2*UR5*R1*UR1*U1*UR2*U1*UR4*U1*UR7*U1*UR5*U1*UR1*U1*UR4*R1*UR3*U2*UR7*U2*UR1*R1*UR5*U1*UR2*U1*UR4*U1*UR1*R1*UR2*U1*UR2*/<77,120>#
This is then reduced to single occurrence of eight directions as
<75,121>/R10*DR82*D23*L175*U21*UR82*/<77,120>#

An approximate measure is taken such that when the amount of pixels in a particular direction is less than 10% of its total amount of pixels it need to be added to their corresponding major directions. This reduces the memory space needed as well as makes the recognition process easier.
Thus the vector code obtained is
<75,121>/DR110*L175*UR108*/<77,120>#
and then normalized to 100 pixels as

DR28*L45*UR27

So, the final vector is a normalized vector called knowledge vector as it consists of more information of the object in its simpler form. The knowledge vector can be more informative by separating the basic sides and the diagonal sides. Initially, it can be viewed as a string with every small change reflected and then limited to 8 directions. From the initial string obtained, the continuous pixels that make the straight lines (chosen only if at least 2% pixels present) can be extracted first and the remaining pixels can be approximated to diagonal directions. Let us consider an image of a triangle



(a) Image (b) Contour

Figure 4: Image of a Triangle and its contour

The knowledge vector obtained is
<75,121>/D1*DR2*D1*DR3*R1*DR1*D1*DR3*D1*DR3*D1*DR7*D1*DR5*D1*DR2*D1*DR3*R1*DR2*D1*DR1*D1*DR2*DR4*D2*DR2*R1*DR4*D2*DR7*D2*DR1*R1*DR5*D1*DR1*D1*DR8*D1*DR4*D1*DR3*D1*DR3*R1*DR1*D1*DR3*D1*DR2*L175*UR3*U1*UR2*U1*UR2*R1*UR3*U1*UR2*U1*UR7*D1*U2*UR7*U2*UR5*R1*UR1*U1*UR2*U1*UR4*U1*UR7*U1*UR5*U1*UR1*U1*UR4*R1*UR3*U2*UR7*U2*UR1*R1*UR5*U1*UR2*U1*UR4*U1*UR1*R1*UR2*U1*UR2*/<77,120>#
Vector limited to single occurrence of 8 directions as
<75,121>/R10*DR82*D24*L175*U22*UR82*/<77,120>#
And then approximated to

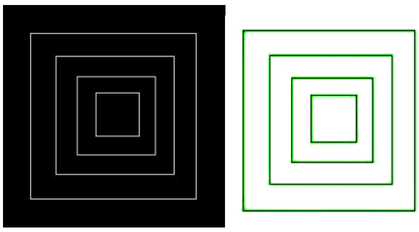
DR116*L175*UR104

Vector code normalized to 100 pixels DR29* L44* UR26

If we observe the knowledge vector obtained, L175 (L44 in the normalized vector) appears as a vector for a straight line with no intermediate changes in directions. Such vectors can be separated as basic directions. The other directions present are DR82, D24, U22, R10 and UR82. In this case, D appears with DR and U appears with UR. So, these lengths can be added to get the directions DR and UR. Thus we obtain the vector code: DR116*L175*UR104. This code is then normalized as DR29* L44* UR26.

VI. Component Wise Image Description And Regeneration

Let us consider the contour of the digital image Figure 5 consisting of four squares (not ideal squares).

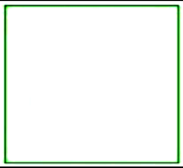
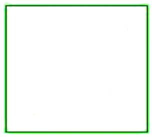
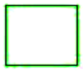


a) Original image b) Contour map

Figure 5: Sample image consisting of four squares

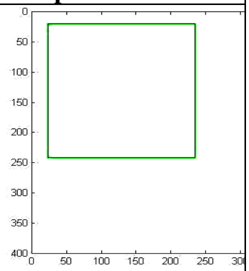
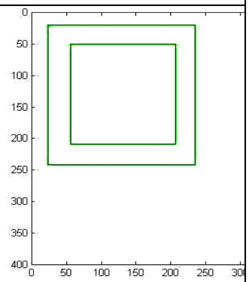
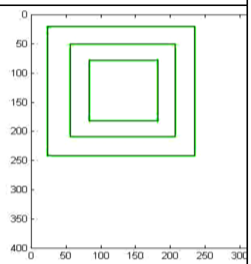
The DLC generating algorithm was applied to this image and the knowledge string was obtained. Table 4 shows the component wise retrieval of the knowledge string for each component in the image.

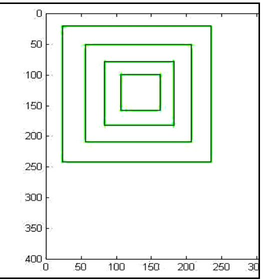
Table 4 : Component wise extraction of the Knowledge string.

Step	**Components**	**Knowledge string**
1.		<19,23>/R212*D223* L212*U222*/<20,23>#
2.		<50,56>/R151*D159* L151*U158*/<51,56>#
3.		<78,83>/R100*D104* L100*U103*/<79,83>#
4.		<99,107>/R56*D58* L56*U57*/<100,107>#

The bottom-up processing is also done to recognize the object by redrawing the same using the knowledge string to verify the accuracy of the DLC generating algorithm.

Table 5: Redrawn images of squares from the extracted knowledge.

Step	**Knowledge string**	**Component**
1.	<19,23>/R212*D223* L212*U222*/<20,23>#	
2.	<50,56>/R151*D159* L151*U158*/<51,56>#	
3.	<78,83>/R100*D104* L100*U103*/<79,83>#	

4.	<99,107>/R56*D58* L56*U57*/<100,107>#	

The knowledge base consists of the various components of the image. Large knowledge bases could be then processed and classified using knowledge manipulation and classification algorithms.

VII. Conclusion

The normalized Direction-Length Code which is also called a knowledge vector that gives the information of direction and length of pixels in every direction in an object is generated to represent objects. The reverse process of regeneration of images from the vector code generated is also done. The algorithm shown good results by generating the same image as its output even when there are more number of components present.

The Direction-Length code (DLC) that compresses the bi-level images by preserving the information at the same time allowing a considerable amount of data reduction can act as a standard input format for numerous shape-analysis and classification algorithms.

References

[1]. H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," *in Proceedings of IRE Translation Elec-tron Computer,* New York, pp-260-268, 1961.

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

[3]. R. Chellappa, R. Bagdazian, Fourier coding of image boundaries, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1) 102–105 (1984).

[4]. C.W. Richard, H. Hemami, "Identi1cation of three dimensional objects using Fourier descriptors of the boundary curve", IEEE Trans. System Man Cybernet, SMC-4 (4) 371–378 (1974).

[5]. Torralba, A., Oliva, A., Freeman, W. T., Object recognition by scene alignment [Abstract]. Journal of Vision, 3(9): 196, 196a (2003)

[6]. Leordeanu et al., "Beyond local appearance: Category recognition from pair wise interactions of simple features", IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007).

[7]. D. Zhang, G. Lu , "Review of shape representation and description techniques", Pattern Recognition 37 : 1–19 (2004)

[8]. [8] Biederman, I., "Recognition-by-components: A theory of human image understanding", Psychological Review, 94: 115-147 (1987)

[9]. J. Iivarinen, A. Visa, "Shape recognition of irregular objects", Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, Proc. SPIE 2904 25–32 (1996).

[10]. K.S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, New York, 1974.

[11]. N. Chomsky, Syntactic Structures, Mouton, The Hague, Berlin, 1957.