

SOME STUDIES IN ANALYSIS AND
RECOGNITION OF 2-DIMENSIONAL SHAPES

SWAPAN KUMAR PARUI

Thesis submitted to the
Indian Statistical Institute in partial fulfilment
of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY

CALCUTTA

1984

ACKNOWLEDGEMENTS

The work reported in this thesis was carried out under the supervision of Prof. D Dutta Majumder. I am grateful to him for his valuable guidance and continuous encouragement.

I acknowledge with pleasure the help, academic and extra-academic, that I received from Dr. B.B. Chaudhuri throughout the period of preparation of this thesis.

I am especially grateful to Professor D.Dutta Majumder, Dr. B.B. Chaudhuri and Mr. S.E. Sarma for their kind permission to include in this thesis some of our joint work.

I take this opportunity to express my sincere thanks to Mr. C.A. Murthy, Mr. Bhabatosh Chanda and Dr. V.V. Srivatsa for the many valuable discussions I had with them.

I wish to thank the authorities of the Indian Statistical Institute for the facilities extended to carry out my research.

Thanks are also due to Mrs. S.De Bhowmick and Mr. J.Gupta for their typing, to Mr. Dilip Chatterjee for cyclostyling and to Mr. S. Chakraborty for drawing the diagrams.

Finally, I am grateful to my parents for their patience.

CONTENTS

0.	INTRODUCTION	
0.1	Background	1
0.2	Scope of the Thesis	8
1.	SHAPE AND DIGITIZATION OF BILEVEL PICTURES	
1.1	Introduction	20
1.2	Analogue Case	22
1.3	Digital Case	25
1.4	Some Results Regarding Preservation of Topological Properties of Bilevel Pictures After Digitization	31
2.	SOME OPERATIONS ON BILEVEL/BINARY PICTURES	
2.1	Introduction	38
2.2	Some Algorithms for Propagation and Shrinking of Binary Pictures	42
2.3	Some Algorithms for Rotation of Binary Pictures and Their Properties	72
2.4	Conclusion	80
3.	THINNING ALGORITHMS FOR ELONGATED OBJECTS	
3.1	Introduction	107
3.2	Two Thinning Algorithms	112
3.3	Conclusion	128

4. SHAPE DISTANCE AND SHAPE SIMILARITY OF
TWO-DIMENSIONAL CURVES

4.1	Introduction	147
4.2	Analytic Definitions of a Curve and Its Shape	150
4.3	Two Measures of Shape Similarity for Open Curves	162
4.4	Description of Open Curves	180
4.5	Techniques to Compute Shape Distance and Shape Similarity of Open Curves	183
4.6	Conclusion	190

5. SHAPE DISTANCE AND SHAPE SIMILARITY OF
TWO-DIMENSIONAL REGIONS

5.1	Introduction	197
5.2	A Modified Least Squares Method	202
5.3	Definition and Shape of a Region	206
5.4	Shape Vector	208
5.5	Major Axis and Reference Points of a Region	214
5.6	Shape Distance on the Basis of Unmatched Area	220
5.7	Two Shape Distances on the Basis of Region Boundaries	229
5.8	Conclusion	236

6. SYMMETRY OF CURVES AND REGIONS

6.1	Introduction	253
6.2	Symmetry and Antisymmetry of Open Curves	255

6.3	Symmetry and Antisymmetry of Regions	267
6.4	Hierarchical Search for the Best Axis of Symmetry	273
6.5	Results and Discussion	279
7.	CONCLUSION	292
	APPENDIX A	296
	APPENDIX B	302
	APPENDIX C	308
	APPENDIX D	309
	REFERENCES	312

CHAPTER 0

Introduction

0.1 Background

Analysis and recognition of 2-dimensional shapes constitute an important problem in the fields of Pattern Recognition, Image Analysis, Computer Vision, Robotics and other related areas of research. If we look at the problem from the point of view of image or scene analysis, in a sense, the entire subject of automatic scene analysis might be defined as the problem of describing and recognizing the shape of the objects in an image. Shape is the primal intrinsic property for the vision system because we associate the definitions of objects with shape, rather than with colour or reflectivity, for example.

According to Webster, the shape of an object is that quality of the object which depends on the relative position of all points composing its outline. This definition emphasizes the fact that we are aware of shapes through outlines which may be visually perceived.

In this thesis, the shape of an object is defined as that property of the object which is invariant under translation, dilation and rotation.

Shape recognition is undoubtedly one of the most important facilities of our visual system. We know how important shape information can be extracted from images in early processing and segmentation. One of the major challenges to computer vision is to represent shapes, or the important aspects of shapes, so that they may be learned, matched against, recollected and used.

This thesis deals with description and classification of the shapes of parts of a two-dimensional picture. For this purpose, we assume that parts of the picture have been isolated as meaningful entities. In other words, our assumption is that some subsets of the picture plane, called the objects, have already been defined as a result of previous processing. These objects are nothing but bilevel pictures in the plane. In this thesis, two types of bilevel pictures are considered (line structures or curves and regions).

Now any scheme to describe the shape of a bilevel picture (that is, of an object) should ideally satisfy the three following objectives: (1) It should preserve the information of interest, that is, it should not lose any relevant information about the

object while deleting the irrelevant ones (whether ^{some} information is relevant or not depends on the particular context), (2) It should permit compact storage and be convenient for display, (3) It should facilitate any required future processing. It is clear that the above three conditions are conflicting with one another in the sense that compact storage may lead to loss of some relevant information, or preservation of all relevant information may not be always good for subsequent processing. So, a compromise has to be made to get an optimal strategy depending on specific problems.

It has been mentioned above that one desirable property of a description of an object is preservation of relevant information. Since in practice it is not possible in general to preserve all the relevant information, attempts are made to preserve as much of it as possible. We will now formalize the notion of what a more informative description means. The more informative the description, the fewer number of objects there are that satisfy that description. More specifically, a description D_1 is more informative than another description D_2 if the set of objects described by D_1 is properly contained in the set of objects described by D_2 . If one of the above two sets neither is contained in nor contains the other, we say that the information contents of

the two descriptions D_1 and D_2 are not comparable. Under this definition, it is clear, no description of an object can be more informative than the object itself.

We will now give a formal definition of what is known as an information preserving description. Let $D^{(n)}$ denote a description D with size n (i.e. n varying, D fixed). (The size n is the number of dimensions if it is a vector description, n is the number of elements in the string if it is a syntactic description etc). In general, the information content of $D^{(2n)}$ increases as n increases. In other words, $D^{(2n)}$ is more informative than $D^{(n)}$, that is, any object described by $D^{(2n)}$ is described by $D^{(n)}$ while not all objects described by $D^{(n)}$ are described by $D^{(2n)}$. But though, in general, the information content of $D^{(2n)}$ is increasing as n increases, asymptotically $D^{(2n)}$ may not describe an object uniquely. A description D is called information preserving if $D^{(2n)}$ asymptotically describes an object uniquely, that is, if the set of objects described by $D^{(2n)}$ for all finite n is a singleton. Otherwise, we call a description information non-preserving.

While very informative descriptions have the advantage of precision, they also have several significant disadvantages. Perhaps the most important among them revolves around the fact that it is often convenient to think of certain families of objects as

being equivalent. For example, it may be necessary to think of all translations of a given object as being indistinguishable from the object itself. If we restrict ourselves to very informative descriptions, unfortunately the descriptions of a given object and one of its translations will be different. Other often-used equivalence classes of an object are the sets of its dilations and rotations. In general, we speak of descriptions that are invariant under^a certain class of transformations. The challenge is to find descriptions that are insensitive to transformations which change objects only in 'unimportant' ways, yet sensitive to transformations which change objects in any 'important' way. The 'unimportant' transformations we will deal with in this thesis are translation, dilation and rotation. By dilation we mean both propagation and shrinking.

Now we will discuss the basic ways of describing the shape of objects. Fig.0.1 presents a summary of the relevant methodologies. To describe the shape of a region, one approach is to extract a number of scalar measurements directly from the region and thus obtain a feature vector (one such feature vector, called the shape vector, has been used in this thesis). Quantities like area, width, elongation, ratio of perimeter square over area etc. have been used as shape descriptors though they are not information

preserving. Among the information preserving descriptions are two-dimensional Fourier transform and moments (Hu, 1962; Dudani et al, 1977). But the computational requirements of these two-dimensional transforms are quite substantial. Moreover, the scalar measurements in general do not contain the structural information of complicated shapes. The techniques that reduce a region to a feature vector are called scalar transform techniques (Pavlidis, 1978).

A second approach to describe the shape of a region consists of techniques called space domain techniques which retain the structural shape properties of a region. Such a technique reduces the region to a graph, that is, a thin line drawing which reflects the original shape. This can be performed in two ways. The first is thinning where the region is reduced to a skeleton which can be treated as a graph. The second is decomposition where the region is expressed as a union of simpler shapes and is represented by an adjacency or intersection graph of its components. One example is the decomposition into primary convex regions (Pavlidis, 1968; 1972) and one of the earliest and most widely studied techniques for thinning is the medial axis transformation (Blum, 1964; 1973; Mott-Smith, 1970). The data structures produced by thinning or decomposition are particularly appropriate for syntactic or structural pattern recognition while the scalar measurements produced by the

first approach are in general appropriate for classical statistical pattern recognition.

So far we have discussed the shape analysis techniques that involve not only the boundary points of the region but also its interior points. Now we will discuss the techniques that trace only the boundary. Among them some extract scalar measurements, that is, a feature vector from the boundary while others treat the points on the boundary directly and in a strictly local fashion. As before, the former are called scalar transform techniques and the latter space domain techniques. The scalar transform techniques involve mostly the Fourier transform or moments of the boundary (Zahn and Roskies, 1972; Granlund, 1972; Persoon and Fu, 1977; Dudani et.al, 1977). The major advantage of these algorithms is that they are rather easy to program while their main disadvantage lies in the difficulty of describing the local information of the boundary.

The earliest among the space domain techniques are the syntactic descriptions of Ledley (1964) and those based on the Freeman chain code (Freeman, 1961; 1970). Most of these techniques directly or indirectly involve polygonal approximations of the boundary (Pavlidis, 1974; 1977; Davis, 1977; Freeman and Davis, 1977; Freeman and Saghri, 1978; Sklansky and Kibler, 1976). Others include Shape numbers (Bribiesca and Gazman, 1980),

B-splines, Strip trees, Conic sections etc. (Ballard and Brown). Extensive studies of the subject have been made by Fu(1974; 1977) and Pavlidis (1977;1978;1980). The basic advantage of the space domain techniques is that they seem quite promising in reproducing the descriptions of human observers. In this thesis also a boundary is described as a chain of line segments and this description is information preserving.

So far we have discussed the shape of objects that are regions. Now the shape of objects that are curves can be treated in the same way as that of region boundaries.

Recently there has been a new approach to shape approximation, namely, the 'quadtree' representation (Ranade, Rosenfeld and Samet, 1982). But how faithful this representation will be for shape matching in general is still open to question, since the quadtree representation is quite sensitive to translation and rotation of the input picture.

0.2 Scope of the Thesis

One of the important aspects of the study of shape is shape matching which is the major topic in this thesis. There are three ways of shape matching — (i) String matching, (ii) Area matching and (iii) Graph matching. In this thesis, we use only

the first two methods. Through shape matching we quantify the shape distance or the shape similarity between 2-dimensional objects (curves and regions). As mentioned earlier, by the shape of an object we mean a subclass of objects within which any object can be obtained from any other through translation, dilation and rotation. Also two objects of different shapes can not be made to coincide with each other through those transformations. Thus, the measures of shape similarity of objects do not depend on the position, size and orientation of objects. The definition of shape can be extended so that it is invariant under reflection or mirror image also.

Shape similarity measurement can be used for shape recognition. If the value of shape similarity measurement between a known shape and an unknown shape is greater than some threshold value, then the unknown shape can be recognized as the known shape.

In the case of string matching, in general, two strings to be matched may or may not have equal lengths. In this thesis, string lengths are always equal and most of the shape similarity measurements here are based on string matching (Parui and Dutta Majumder, 1982B;1983C;1983D). When shape similarity measurement is based on the area of mismatch, shapes to be matched are usually first converted into polygons. The polygons are then matched against each other. The polygon matching can be either edge based or vertex based (Kashyap and Oommen, 1982). But in this thesis,

shape similarity measurement based on the area of mismatch does not involve conversion of a shape into a polygon (Parui and Dutta Majumder, 1983E;1984).

It is clear by now that in this thesis, we consider pictures with only two levels — black and white. Such pictures are called analogue bilevel pictures (or just bilevel pictures) in the analogue case and digital binary pictures (or just binary pictures) in the digital case. In Chapters 4,5 and 6, emphasis is on shapes of pictures in the analogue case while Chapters 2 and 3 put emphasis on digital pictures. In Chapter 1, both analogue and digital cases are equally emphasized.

The chapterwise break up of the present thesis is as follows:

Chapter 1 In this chapter we present some basic definitions and terminology which are relevant for bilevel and binary pictures. Shape properties in general and topological properties in particular, of bilevel pictures before and after digitization are studied. In general, shape properties are not preserved after digitization. Pavlidis (1980;1982) proposes a condition on digitization of bilevel pictures so that the shape of the bilevel pictures is preserved after their digitization. A similar condition on digitization is considered in this chapter. This

condition imposes an upper limit to the side size of pixels depending on^a certain 'width' of a bilevel picture. Pavlidis's digitization is on the basis of the position of the centre of a pixel while in this thesis it is on the basis of the area of a pixel. With this new definition of digitization, we prove some results on shape preservation after digitization, which are similar to Pavlidis's (1980; 1982).

Chapter 2 This chapter deals with some operations on bilevel/binary pictures with the aim of matching of 2-dimensional shapes. The operations considered here are translation, dilation (i.e. scaling by arbitrary factors more than or less than 1) and rotation (by arbitrary angles, not necessarily by multiples of 45 degrees). The main aim of this chapter is to propose some algorithms for propagation (i.e. scaling by factors more than 1), shrinking (i.e. scaling by factors less than 1) and rotation of digital binary pictures. In the existing algorithms for propagation and shrinking of digital binary pictures, the scaling factors along the x-axis and y-axis are not usually the same (i.e., the x and y co-ordinates do not increase or decrease by the same factor). Hence these algorithms do not, in general, preserve the shape of the input picture.

But the algorithms proposed for propagation and shrinking in this chapter have the property of uniform change of scale and

hence preserve the shape of the input picture. (This shape preservation is not always exact, since some loss of information through these operations is inevitable due to the digital nature of the input and output pictures). The shape preservation property of the propagation and shrinking algorithms proposed in this chapter is required because these algorithms are used for shape matching of 2-dimensional regions in Chapter 5.

We propose algorithms for propagation and shrinking of digital binary pictures first by integer factors and then by arbitrary (real) factors. We take two approaches to the propagation shrinking and rotation algorithms. The first approach involves all the pixels of the input and output pictures while the second approach involves only the boundary pixels. Clearly, the second approach takes less time than the first one. But the second approach has some limitations in the sense that it is more economical in time only for a specific class of input pictures.

Results regarding shape preservation of the above algorithms are finally proved (Parui, 1983E).

Chapter 3 This chapter deals with some thinning algorithms with the aim of shape matching of elongated 2-dimensional regions. Such elongated objects are transformed into 2-dimensional curves through thinning and some methods of shape matching of

2-dimensional curves are given in Chapter 4.

In this chapter, two thinning algorithms are proposed for elongated digital regions. The algorithms are sequential in nature and hence are appropriate for usual general purpose sequential computers. The first algorithm is very simple but has many drawbacks. These drawbacks are removed to a great extent by the second algorithm which is an improved version of the first algorithm. The thinned output of the second algorithm is very close to the well-known medial axis. Both the algorithms preserve the connectivity property of the input picture.

Chapter 4 This chapter deals with shape matching of 2-dimensional open curves (i.e. curves that are not self-intersecting and not closed) and proposes two definitions of shape distance and shape similarity. Curves are usually approximated by polylines as well as by splines and other curves. In this chapter we consider only polyline description which is a concatenation of some line segments and which can be represented as a finite sequence of points in the plane. In general, the points are so chosen as to minimize the difference between the original curve and the polyline described by those points. (The measure of the difference is usually either/maximum error or the integral squared error). But such a description may not be always appropriate for shape matching. In this chapter, with the aim of shape matching,

the points are chosen in such a way that any two consecutive points have the same arc distance between them, though this description does not in general minimize the difference between the original curve and the resulting polyline. But it can be seen that this description also is information preserving.

Two measures of shape distance and shape similarity are proposed here. For this purpose a one-to-one correspondence is established between the points on any two open curves. One measure is based on the Euclidean distances between corresponding points on the two curves and the other on the differences between the slopes of the two curves at corresponding points. The first measure requires equal lengths of the two curves while the second one does not. Both these measures are invariant under translation, dilation and rotation of the curves and satisfy all the metric properties (among shapes). Two shape similarity measures are defined on the basis of these two distance measures. These two distance or similarity measures can be used to discriminate between shapes of 2-dimensional open curves.

Computational methods for the above two distance and similarity measures are given along with some results which are consistent with the human visual judgement. How the shape distances are to be modified if the shape is made invariant under reflection

also, is finally discussed. The above two shape distances for 2-dimensional curves are used in Chapter 5 to define two corresponding shape distances for 2-dimensional regions on the basis of their boundaries (Parui and Dutta Majumder, 1982B; 1983C).

Chapter 5 This chapter deals with shape matching of 2-dimensional regions. By regions we mean compact and path-connected subsets in the plane. Four measures of shape distance and shape similarity of 2-dimensional regions are proposed here. For shape matching, a region has to be normalized in terms of its position, size and orientation. Normalization of a region in terms of its position and size can easily be achieved if its centre of gravity and area are considered. The main problem in shape matching of regions involves the normalization of orientation of a region and also the normalization of the starting point on the boundary of a region if the boundary only is studied. These are solved on the basis of ^acertain major axis of a region in this chapter (Parui and Dutta Majumder, 1982B; 1983E; 1984).

Of the four shape distances for regions proposed here, only the first one involves scalar transform technique while the others are essentially based on space domain techniques. Though the first technique is information non-preserving, the other three are information preserving. The first shape distance is based on ^acertain shape vector of a region which does not depend

on the position, size and orientation of the region. But this shape distance does not satisfy all the metric properties. The second shape distance is based on area matching and the last two on string matching through region boundaries. These three shape distances satisfy all the metric properties. Rotational invariance in these cases is achieved by the major axis of a region which has some desirable invariance properties. In the last two cases, shape distances are defined on the basis of region boundaries and the problem of finding the starting point is solved with the help of the points of intersection of ^{the} boundary with the major axis. These two shape distances for regions are, in fact, based on the two shape distances defined for curves in Chapter 4.

Four shape similarity measures for regions are defined on the basis of the above four shape distances and can be used to discriminate between shapes of 2-dimensional regions. Methods for computing these shape distance and similarity measures in the digital case are given along with some results which are consistent with the human visual system. Sometimes it may be necessary to make shape invariant under reflection also. The way in which the above shape distances are then to be suitably modified is also discussed. Finally, methods of extending some of these shape distances to 3 dimensions are proposed.

Chapter 6 This chapter deals with symmetry of 2-dimensional curves and regions. Symmetry can play many roles in understanding shapes. A symmetric shape can be represented economically. One need store a description of only one half of a symmetric shape along with the axis of symmetry. Again, if a part of a symmetric shape is missing or noisy, then with the help of symmetry one can complete the shape or rid the shape of noise. The axis of symmetry is also useful in defining the orientation of a shape.

Two types of symmetry are discussed in this chapter. The first type is symmetry around an axis and is usually termed as axial or lateral symmetry. Here, this is called just 'symmetry' while the other type is called 'antisymmetry' which is symmetry around a point and is usually known as rotational or radial symmetry.

This chapter proposes some measures of degree of symmetry and antisymmetry for curves on the basis of the two shape distances for curves suggested in Chapter 4. As far as symmetry of a region is concerned, it can be noted that for a symmetric region, its axis of symmetry either coincides with or is perpendicular to its major axis and passes through its centre of gravity. Thus the major axis is used here to define some

measures of degree of symmetry and antisymmetry for regions.

It can also be seen that a region is symmetric if and only if its boundary is symmetric.

So, symmetry and antisymmetry of a region are here dealt with on the basis of its boundary. Also, a hierarchical approach is suggested to locate the best axis of symmetry of a region and to define another measure of degree of symmetry. The axis of symmetry of a region found hierarchically is better than the axis of symmetry found on the basis of the major axis (Parui, '83A). Finally, some computational results regarding the position of the axis of symmetry and degrees of symmetry and antisymmetry are given.

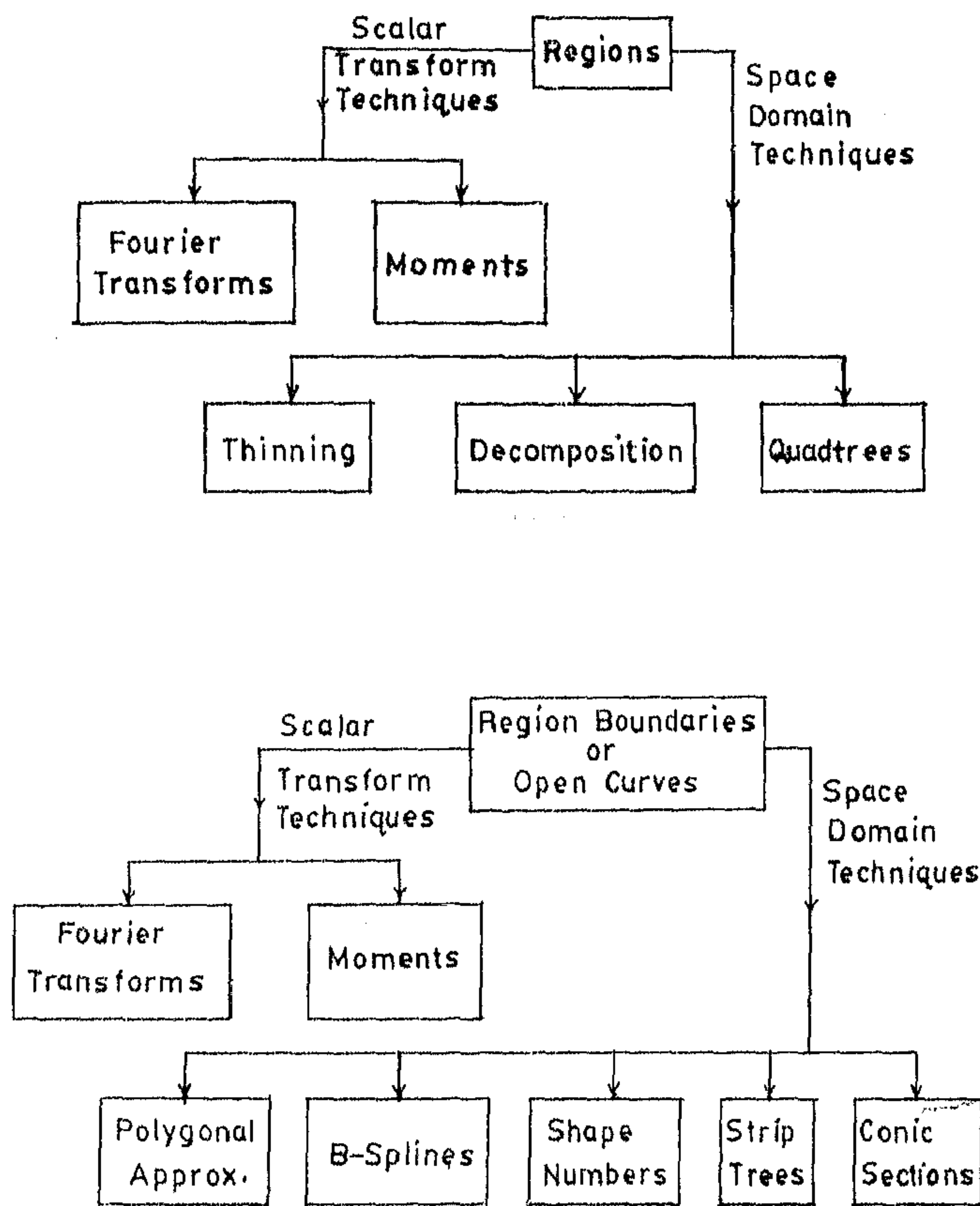


Fig. 0.1 A summary of various methodologies for describing the shape of 2-dimensional objects (regions and curves).

CHAPTER 1

Shape and Digitization of Bilevel Pictures

1.1 Introduction

In this chapter we will present some basic definitions and terminologies which are relevant for black and white pictures. (By black and white pictures, we mean bilevel pictures). We will also discuss sampling or digitization of an analogue bilevel picture which preserves its shape. Pavlidis (1982) proposes a condition on digitization so that the shape is preserved. We will prove in this chapter some results on shape preservation under a similar condition. Topological properties (like connectedness, surroundedness etc.) are a subset of shape properties of black and white pictures. In this chapter, we will discuss preservation of topological properties. But it will be seen in the sequel that the shape properties that are preserved under the proposed condition are much more than just topological properties.

Two sets A and B in the plane are said to have the same topological properties or to be topologically equivalent if there is a one-to-one mapping between A and B and if both the mapping from A to B , and its inverse from B to A , are continuous. Since, in image processing, one deals with the topology induced by the Euclidean metric, topological equivalence means that there is a

one-to-one mapping such that points which are close to each other are mapped to points which are close. In other words, one can transform one set onto another by stretching or squeezing without tearing or cutting. Thus, if a region has a hole, it can not be topologically equivalent to a region without a hole. It can be proved that the two pictures or sets (dark areas) in Fig.1.1(a) and (b) are topologically equivalent since, informally speaking, each set is connected and has a hole. But it is clear that these two pictures have widely different shapes. We will show in what follows that the condition on digitization for shape preservation does not allow such output (Fig.1.1(b)) for the analogue input in Fig.1.1(a). In general, the condition implies preservation not only of topological properties but of other shape properties as well. Under this condition, Fig.1.1(c) is a possible digitized output of Fig.1.1(a).

In the discrete grid Pavlidis (1982) makes a pixel a black pixel if its centre falls in the input analogue bilevel picture. But here a pixel is made black if at least half of the area of the pixel falls in the analogue picture. With this new digitization scheme we will prove results similar to Pavlidis's (1982;1980).

In the next section, we will present some basic definitions and terminologies about the topology of analogue bilevel pictures

while in Section 1.3 we will present the same for digital binary pictures. In Section 1.4 we will prove some results about shape preservation of analogue pictures after digitization.

1.2 Analogue Case

Suppose, A is a subset in the plane $\mathbb{R}^2$ and d is a positive real number. Let $F_d(A)$ be the class of all closed discs D of diameter d contained in A . Let $C_d(A)$ be the set of centres of the discs belonging to $F_d(A)$.

In this section, we will mainly deal with those subsets A of the plane such that there exists a positive real number d satisfying

- (a) $F_d(A)$ is non-empty
- (b) A is the union of all discs in $F_d(A)$
- (c) $C_d(A)$ is path-connected
- (d) A is compact



(1.1)

and (e) A^o can be expressed as a finite union of non-empty open sets $\bigcup_{j=1}^m B_j$ such that B_j satisfies conditions (a), (b) and (c) above for all j and $B_j \cup B_k$ is disconnected for $j \neq k$.

From above it is clear that A is path-connected and m is said to be the order of connectivity of A .

Let $Z(A) = \{d: d \text{ is a positive real number satisfying conditions (a), (b) and (c) above}\}$. So, for a subset A satisfying conditions (a), (b) and (c) above, $Z(A)$ is non-empty.

Definition 1.2.1 Let A be a subset satisfying conditions (a), (b) and (c) in (1.1). Then $\lambda(A) = \sup_{d \in Z(A)} d$ is said to be the width of A , provided $Z(A)$ is bounded.

Definition 1.2.2 A subset A in the plane is said to be an analogue bilevel picture if A can be expressed as a finite union $\bigcup_{i=1}^n A_i$ such that $A_i \cup A_r$ is disconnected for $i \neq r$ and A_i satisfies the five conditions in (1.1) for all i . Each A_i is called a component of the analogue bilevel picture A . In Fig.1.2, there are two components A_1 and A_2 .

Let m_i be the order of connectivity of A_i . From condition (e) in (1.1), $A_i^c = \bigcup_{j=1}^{m_i} B_{ij}$ where B_{ij} is non-empty and satisfies conditions (a), (b) and (c) in (1.1) for all j and $B_{ij} \cup B_{ik}$ is disconnected for $j \neq k$.

Proposition 1.2.1 For each i , B_{ij} is unbounded for exactly one value of j . For all other values of j , if any, B_{ij} is bounded.

Proof. See Appendix A.1.

Without loss of generality, let B_{i1} be unbounded. So, $B_{i2}, \dots, B_{im_i}$, if any, are bounded.

Proposition 1.2.2 If A_i and A_r ($i \neq r$) are two components of an analogue bilevel picture A , then A_r is a subset of B_{ij} for exactly one j ($m_i \geq j \geq 1$).

Proof This is because A_r is a path-connected subset of A_i^C and $A_i^C = \bigcup_{j=1}^{m_i} B_{ij}$ where $B_{ij} \cup B_{ik}$ is disconnected for $j \neq k$.

Definition 1.2.3 If a component A_r is a subset of B_{ij} for $j > 1$, then A_i is said to surround A_r (or A_r is said to be surrounded by A_i). In Fig.1.2, A_2 is surrounded by A_1 .

Proposition 1.2.3 If $m_i > 1$, then $\bigcup_{j=2}^{m_i} B_{ij} - \bigcup_{i=1}^n A_i$ can be expressed as a finite union $\bigcup_{r=1}^{w_i} H_{ir}$ where H_{ir} satisfies conditions (a), (b) and (c) in (1.1) for all r and $H_{ir} \cup H_{is}$ is disconnected for $r \neq s$.

Proof See Appendix A.2.

Definition 1.2.4 H_{ir} , $r=1, \dots, w_i$ are called the holes of the component A_i .

Note that according to this definition, a hole may be common to two or more components (Fig.1.2).

Definition 1.2.5 If a component A_i has no holes, A_i is said to be a simply connected component.

Let $B = \bigcap_{i=1}^n B_{i1}$. It is easy to see that B is unbounded.

Definition 1.2.6 B is said to be the background of the analogue bilevel picture A . A set H is said to be a hole of A if H is a hole of some component of A .

Proposition 1.2.4 B satisfies conditions (a), (b) and (c) in (1.1).

Proof. Above result can be proved in the same way as it has been proved in Appendix A.2 that D_{i1} satisfies conditions (a), (b) and (c) in (1.1).

Note that the widths of the components A_i and the holes H_{ir} are well-defined and positive for all i and r .

Definition 1.2.7 The width of the analogue bilevel picture A is defined as

$$\delta(A) = \text{Min} \left\{ \text{Min}_i \lambda(A_i), \text{Min}_{i,r} \lambda(H_{ir}), \lambda(B) \right\} \quad (1.2)$$

If $\lambda(B)$ does not exist, i.e., if $Z(B)$ is unbounded, $\lambda(B)$ is ignored in the above definition. Clearly, $\delta(A)$ is finite and positive.

1.3 Digital Case

First we will briefly discuss how an analogue bilevel picture is digitized. A grid is a finite rectangular array of closed squares of equal size. The grid is superimposed on the input analogue bilevel picture A so that it contains the whole picture. Suppose, the grid has M squares in a column and N squares in a row. The aim of digitization is to assign the value 0 or 1 to each of the MN sampling squares or pixels, where 0 will mean white and 1 black. We will call them 0-pixels and 1-pixels correspondingly.

Let h be the length of the side of each pixel. Let a_{ij} be the area of the part of the (i,j) -th pixel which falls in the input analogue bilevel picture A . So, $h^2 \geq a_{ij} \geq 0$. Let $P = ((p_{ij}))$ be a matrix of the order $M \times N$ such that

$$p_{ij} = 1 \quad \text{if } a_{ij} \geq ph^2 \\ = 0 \quad \text{otherwise}$$

(p is always taken to be $1/2$ here, though p can take any value in $(0,1]$). We will denote the (i,j) -th pixel by P_{ij} or simply by (i,j) .

Definition 1.3.1 The matrix P is a digitization matrix of the analogue bilevel picture A .

Definition 1.3.2 The set of all 1-pixels, $S = \{P_{ij} : p_{ij} = 1\}$, is a digitized version of the analogue bilevel picture A and is called a digital binary picture. Without loss of generality, we assume that after digitization the first and last rows and the first and last columns of P contain only 0-pixels. These rows and columns are called the frame of the digital binary picture.

We write $S = \text{Dig}_{h,p}(A)$ which is not unique but depends on the position of the grid in relation to the analogue picture A . However, S tends to A as h decreases to 0. S^c , the complement of S , is the set of all 0-pixels. Note that S^c contains the frame of the digital picture.

Definition 1.3.3 A pixel (i_1, j_1) is an 8-neighbour of another pixel (i_2, j_2) if (i_1, j_1) is any of the eight pixels $(i_2, j_2 \pm 1)$, $(i_2 \pm 1, j_2)$, $(i_2 \pm 1, j_2 \pm 1)$. (i_1, j_1) is a 4-neighbour of (i_2, j_2) if (i_1, j_1) is any of the four pixels $(i_2, j_2 \pm 1)$, $(i_2 \pm 1, j_2)$.

Definition 1.3.4 Two pixels (i_1, j_1) and (i_n, j_n) belonging to S are said to be 8-connected or connected if any one of the following conditions holds



- (i) $i_1 = i_n$ and $j_1 = j_n$
- (ii) (i_1, j_1) is an 8-neighbour of (i_n, j_n)
- (iii) there exist pixels (i_k, j_k) , $k=2, \dots, n-1$ ($n > 2$) belonging to S such that (i_k, j_k) is an 8-neighbour of (i_{k+1}, j_{k+1}) for $k=1, \dots, n-1$

A set of pixels is said to be 8-connected or connected if every pair of pixels belonging to the set is 8-connected.

Definition 1.3.5 Let S^* be any subset of the whole set of pixels, $S \cup S^c$. Two pixels (i_1, j_1) and (i_n, j_n) belonging to S^* are said to be 4-connected or background-connected in S^* if any of the following conditions holds

- (i) $i_1 = i_n$ and $j_1 = j_n$
- (ii) (i_1, j_1) is a 4-neighbour of (i_n, j_n)
- (iii) there exist pixels (i_k, j_k) , $k=2, \dots, n-1$ ($n > 2$) belonging to S^* such that (i_k, j_k) is a 4-neighbour of (i_{k+1}, j_{k+1}) for $k=1, \dots, n-1$

A set of pixels is said to be 4-connected or background - connected if every pair of pixels belonging to the set is 4-connected.

Proposition 1.3.1 'Connectedness' is an equivalence relation in S . That is, for all $(i_1, j_1), (i_2, j_2)$ and (i_3, j_3) in S ,

- (i) Reflexivity: (i_1, j_1) and (i_1, j_1) are connected
- (ii) Symmetry: if (i_1, j_1) and (i_2, j_2) are connected, then (i_2, j_2) and (i_1, j_1) are connected
- (iii) Transitivity: if (i_1, j_1) and (i_2, j_2) are connected and (i_2, j_2) and (i_3, j_3) are connected, then (i_1, j_1) and (i_3, j_3) are connected (Rosenfeld and Kak, 1976)

The proof of the above proposition is trivial.

Since S is finite, the number of equivalence classes generated by the equivalence relation 'connectedness' is finite. Let $S_i, i=1, \dots, n$ be the equivalence classes.

Definition 1.3.6 Each S_i is called a component of the digital binary picture S . S_i 's are the maximal 8-connected subsets of S and are analogous to A_i 's of the analogue case.

The complement of S_i is S_i^c .

Proposition 1.3.2 'Background connectedness' is an equivalence relation in S_i^c .

The proof of the above proposition is trivial.

Since S_i^C is finite, the number of equivalence classes generated by the equivalence relation 'background connectedness' in S_i^C is finite. Let T_{ij} , $j=1, \dots, m_i$ be the equivalence classes. T_{ij} 's are the maximal 4-connected subsets of S_i^C and are analogous to B_{ij} 's of the analogue case. It is clear that the frame of the digital picture S is contained in exactly one equivalence class. Without loss of generality, let us assume that T_{i1} contains the frame.

Definition 1.3.7 The order of connectivity of the component S_i is m_i .

Proposition 1.3.3 If S_i and S_r ($i \neq r$) are two components of a digital binary picture, then S_r is a subset of T_{ij} for exactly one j ($m_i > j \geq 1$).

Proof Suppose not. That is, no single T_{ij} contains S_r . Without loss of generality, suppose S_r is a subset of $T_{i1} \cup T_{i2}$ such that both $S_r \cap T_{i1}$ and $S_r \cap T_{i2}$ are non-empty. Now, consider the pixels which are 4-connected to some pixel in S_r . Let V be the set of all such pixels. Let $S_r^* = V \cup S_r$. Note the S_r^* is 4-connected. Now, no pixel in V belongs to S_i , since $S_i \cup S_r$ is not 4-connected. Similarly, no pixel in V belongs to T_{ij} for $j > 2$, since $T_{i1} \cup T_{ij}$ as well as $T_{i2} \cup T_{ij}$ is not 4-connected. Thus S_r^* is a subset of $T_{i1} \cup T_{i2}$. Since S_r^* is 4-connected and $S_r^* \cap T_{i1}$ and $S_r^* \cap T_{i2}$ are non-empty, $T_{i1} \cup T_{i2}$ is 4-connected. Contradiction.

The above proposition is analogous to Proposition 1.2.2 of the analogue case.

Definition 1.3.8 If a component S_r is a subset of T_{ij} for $j > 1$, then S_i is said to surround S_r (or S_r is said to be surrounded by S_i).

Let $V_i = \bigcup_{j=2}^{m_i} T_{ij} - \bigcup_{i=1}^n S_i$ which is non-empty.

Let W_{ir} , $r=1, \dots, w_i$ be the equivalence classes generated by the equivalence relation 'background connectedness' in V_i .

Definition 1.3.9 W_{ir} , $r=1, \dots, w_i$ are called the holes of the component S_i . If the component S_i has no holes, S_i is said to be a simply connected component.

W_{ir} 's are analogous to H_{ir} 's of the analogue case.

Let $T = \bigcap_{i=1}^n T_{i1}$ which is non-empty since T contains the frame of the picture. T is analogous to B defined in the analogue case.

Definition 1.3.10 T is said to be the background of the digital binary picture S . A subset W of S^c is said to be a hole of S if W is a hole of some component of S .

1.4 Some Results Regarding Preservation of Topological Properties of Bilevel Pictures After Digitization

In the last two sections some topological properties (like connectedness, surroundedness, number of connected components, number of holes) of analogue as well as digital pictures have been discussed. In what follows we will put some condition on the side size h of pixels of the grid so that topological properties of an analogue picture are preserved after its digitization. The condition on h for shape preservation put forward by Pavlidis(1982) is the following.

The pixel side size h should be such that (i) there exists a number $d > \sqrt{2}h$ so that for each boundary point of an analogue picture A , it is possible to find a disc D with diameter d that is tangent to the boundary at that boundary point and is contained in A and (ii) the same is also true for the complement of A .

The condition put forward later in this section is essentially the same as above. According to Pavlidis a pixel is made a 1-pixel if its centre belongs to A while here a pixel is made a 1-pixel if at least half of the area of the pixel falls in A . Now we will prove a set of results regarding shape preservation.

Proposition 1.4.1 Suppose A_r is a component of an analogue bilevel picture with width $\lambda_r = \lambda(A_r)$. Then the digitized version of A_r is non-empty and 4-connected with arbitrary position and

orientation of the grid, provided $\lambda_r > 2\sqrt{2}h$.

Proof. Let X be the digitized version of A_r . That is, X is the set of all pixels P_{ij} such that $a_{ij} \geq h^2/2$. Since $\lambda_r = \lambda(A_r)$, there is at least one disc D of diameter λ_r such that D is contained in A_r . Again, D contains at least one pixel since $\lambda_r > 2\sqrt{2}h$ (Fig.1.3). Hence X is non-empty. To show that X is 4-connected. Let $E = C_{2\sqrt{2}h}(A_r)$. If a pixel P_{ij} has some intersection with E , then there is a disc D of diameter $2\sqrt{2}h$ with centre in P_{ij} such that P_{ij} is contained in D and D is contained in A_r . So, $a_{ij} = 1$. Hence P_{ij} is a 1-pixel and belongs to X . Let Y be the set of all pixels P_{ij} such that P_{ij} has some intersection with E . Clearly, Y is a subset of X . First we will show that Y is 4-connected and then we will show that if a pixel P_{ij} belongs to $X-Y$, P_{ij} is 4-connected to some pixel in Y .

Let $P_{i_1j_1}$ and $P_{i_2j_2}$ be two pixels in Y . So, there are two points x in $P_{i_1j_1}$ and y in $P_{i_2j_2}$ such that x and y belong to E . From condition (c) in (1.1), E is path-connected. So, there is a path f in E from x to y . Note that the set of pixels having some intersection with f is 4-connected, since pixels are closed squares. Hence $P_{i_1j_1}$ and $P_{i_2j_2}$ are 4-connected. So, Y is 4-connected.

Let P_{ij} belong to $X-Y$ (Fig.1.4). So, P_{ij} does not have any intersection with E but $a_{ij} \geq h^2/2$. Now, from condition (b) in

(1.1), E has to have some intersection with at least one 8-neighbour of P_{ij} , because otherwise a_{ij} will be less than $h^2/2$. If E has some intersection with any of the 4-neighbours of P_{ij} , then the result becomes trivial since the intersecting 4-neighbour belongs to Y . Suppose, E does not have any intersection with any of the 4-neighbours of P_{ij} . So, E has intersection with at least one of the corner neighbours (8-neighbours but not 4-neighbours) of P_{ij} . Without loss of generality, suppose E has intersection with $P_{i+1,j+1}$. So, E does not have any intersection with $P_{i,j+1}$ and $P_{i+1,j}$. It can be seen that any disc with centre in $P_{i+1,j+1}$ will have more intersection (in terms of area) with $P_{i,j+1}$ or $P_{i+1,j}$ than with P_{ij} . In general, the union of any number of discs with centres in $P_{i+1,j+1}$ will have more intersection with $P_{i,j+1}$ or $P_{i+1,j}$ than with P_{ij} . This result is true for pixels $P_{i,j+2}$ and $P_{i+2,j}$ also. So, either $a_{i,j+1}$ or $a_{i+1,j}$ is greater than or equal to a_{ij} . Since $a_{ij} \geq h^2/2$, either $a_{i,j+1}$ or $a_{i+1,j}$ is greater than ^{or} equal to $h^2/2$. Hence either $P_{i,j+1}$ or $P_{i+1,j}$ belongs to X . Thus P_{ij} is 4-connected to $P_{i+1,j+1}$ which belongs to Y .

Proposition 1.4.2 Suppose H_{ir} is a hole of an analogue bilevel picture with width $\lambda_{ir} = \lambda(H_{ir})$. Then after digitization with arbitrary position and orientation of the grid, H_{ir} gives

rise to at least one 0-pixel and the resulting 0-pixels, if more than one in number, are 4-connected, provided $\lambda_{ir} > 2\sqrt{2}h$.

Proof. Same as that of Proposition 1.4.1.

Proposition 1.4.3 Suppose A_i and $A_r (i \neq r)$ are two components of an analogue bilevel picture A with width $\delta = \delta(A)$. Suppose S_i and S_r are the digitized versions of A_i and A_r respectively after digitization with arbitrary position and orientation of the grid and $\delta > 2\sqrt{2}h$. Then S_i and S_r are disconnected, that is, $S_i \cup S_r$ is not 8-connected.

Proof. From the proof of Proposition 1.4.1 it is clear that the set of 0-pixels given rise to by the background or any hole of A is 4-connected since $\delta = \min \left\{ \min_j \lambda(A_j), \min_{j,k} \lambda(H_{jk}), \lambda(B) \right\}$. So, $S_i \cup S_r$ can not be 8-connected. But note that S_i as well as S_r is 4-connected from Proposition 1.4.1.

Proposition 1.4.4 Suppose E is a hole and F is either a hole or the background of an analogue bilevel picture A with width $\delta = \delta(A)$. Suppose, T_1 and T_2 are the sets of 0-pixels given rise to by E and F respectively. Then T_1 and T_2 are disconnected, that is, $T_1 \cup T_2$ is not 8-connected, provided $\delta > 2\sqrt{2}h$.

Proof. Same as that of Proposition 1.4.3.

From the above four propositions we can state the following proposition.

Proposition 1.4.5 Suppose, A_i is a component of an analogue bilevel picture A such that A_i has v_i holes $H_{i1}, \dots, H_{iv_i}$ and A_i surrounds w_i components $A_{i1}, \dots, A_{iw_i}$. Suppose, the width of A is $\delta = \delta(A)$ and the digital version of A is S . Then after digitization with arbitrary position and orientation of the grid and $\delta > 2\sqrt{2}h$,

(i) each of $A_i, A_{i1}, \dots, A_{iw_i}$ gives rise to one single component of S such that the digitized version of A_i surrounds the digitized versions of $A_{i1}, \dots, A_{iw_i}$.

(ii) each of $H_{i1}, \dots, H_{iv_i}$ gives rise to one single hole of S which is a hole of the digitized version of A_i .

From above it is clear that the condition $\delta > 2\sqrt{2}h$ preserves not only the topological properties but all shape properties of the input analogue picture. The only distortion after digitization in this case will be lack of smoothness along the boundary. For example, under the condition $\delta > 2\sqrt{2}h$, the digitized version of Fig.1.1(a) may be Fig.1.1(c), but never Fig.1.1(b).

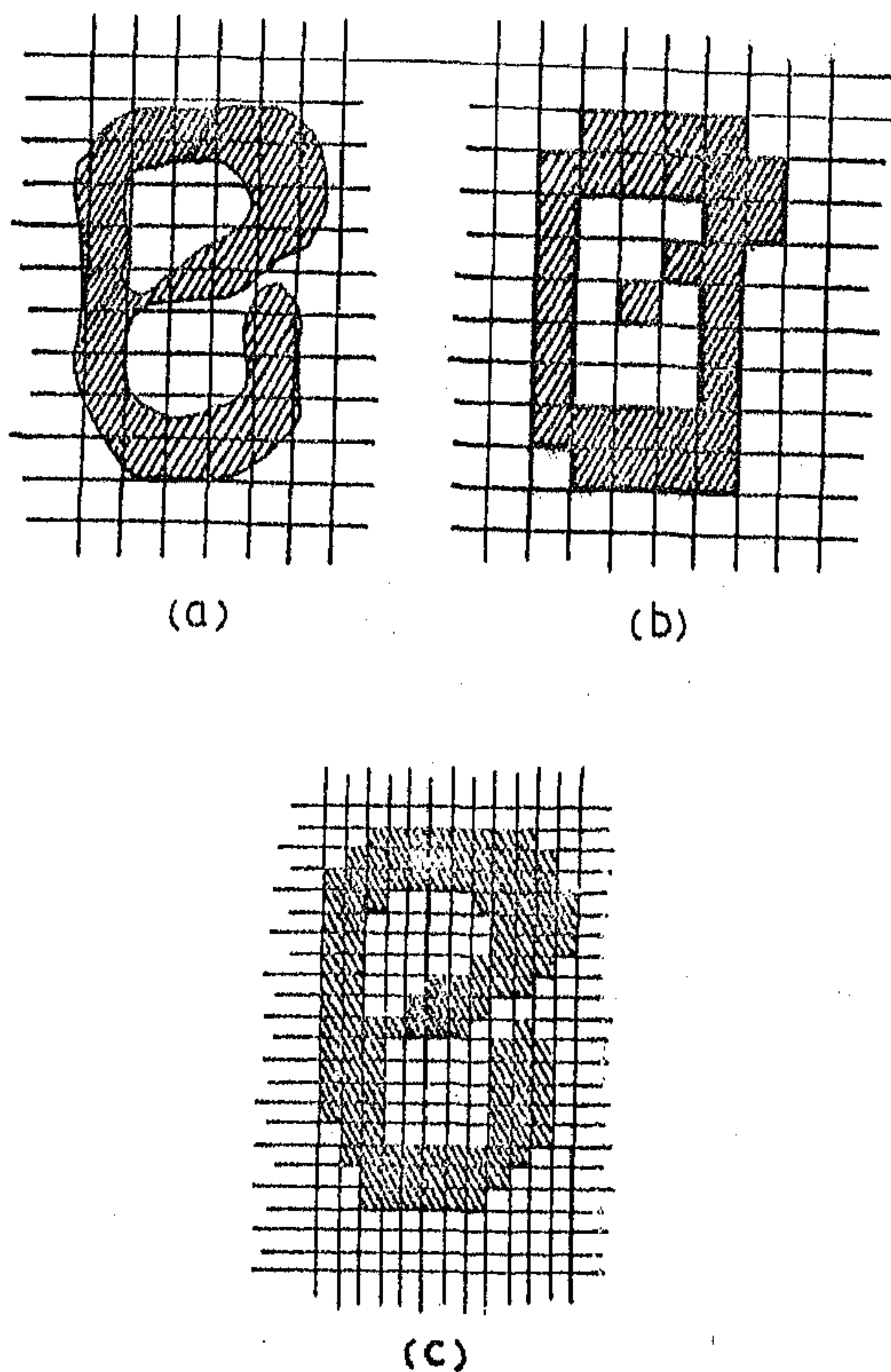


Fig.1.1 Though figures (a) and (b) are topologically equivalent, their shapes vary widely. The shapes of figures (a) and (c) are very close. Under the condition $\delta > 2\sqrt{2}h$, the digitized version of figure (a) may be figure (c), but never figure (b).

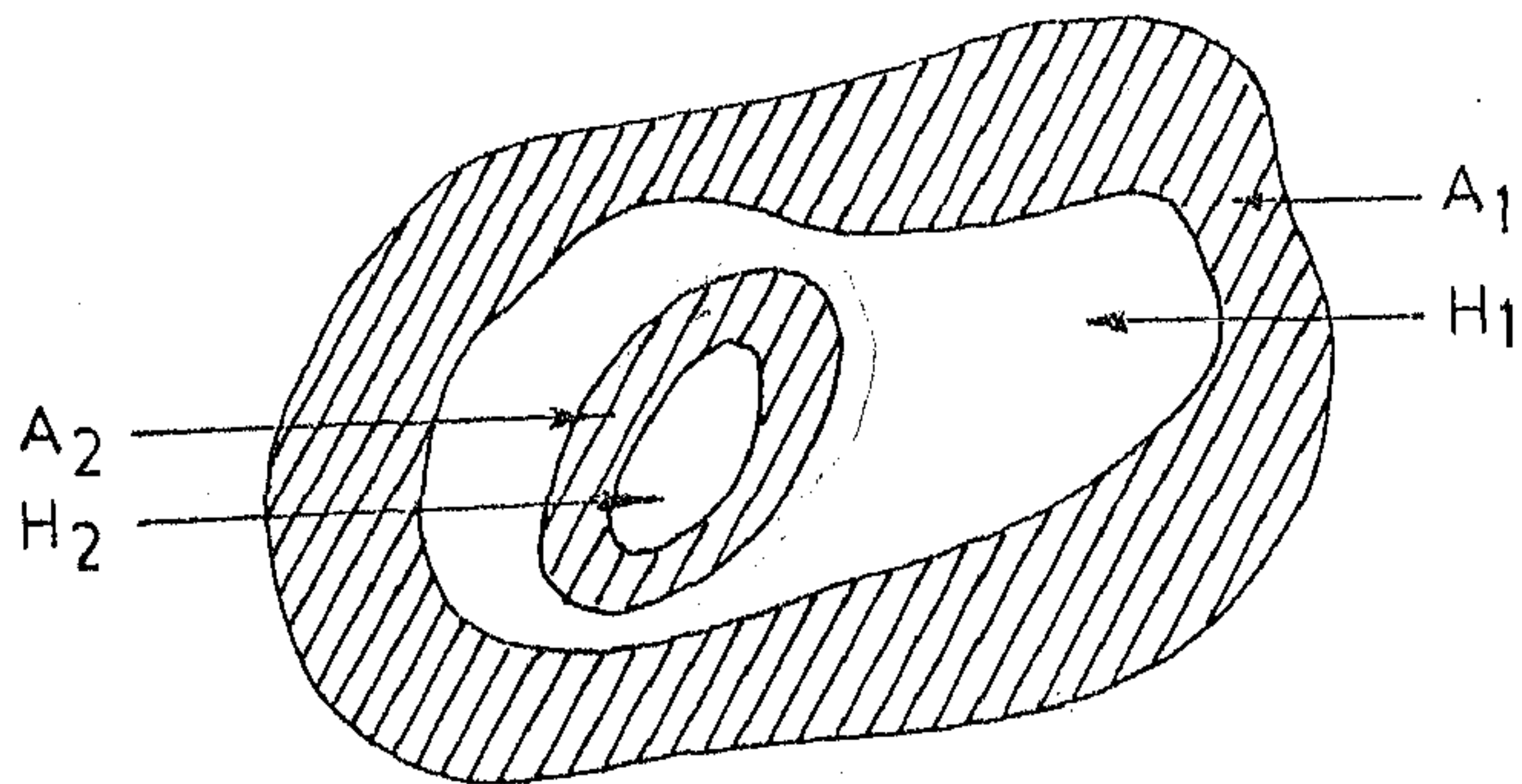


Fig.1.2 A_1 and A_2 are two components and H_1 and H_2 are two holes. A_2 is surrounded by A_1 . H_2 is a hole of both A_1 and A_2 .

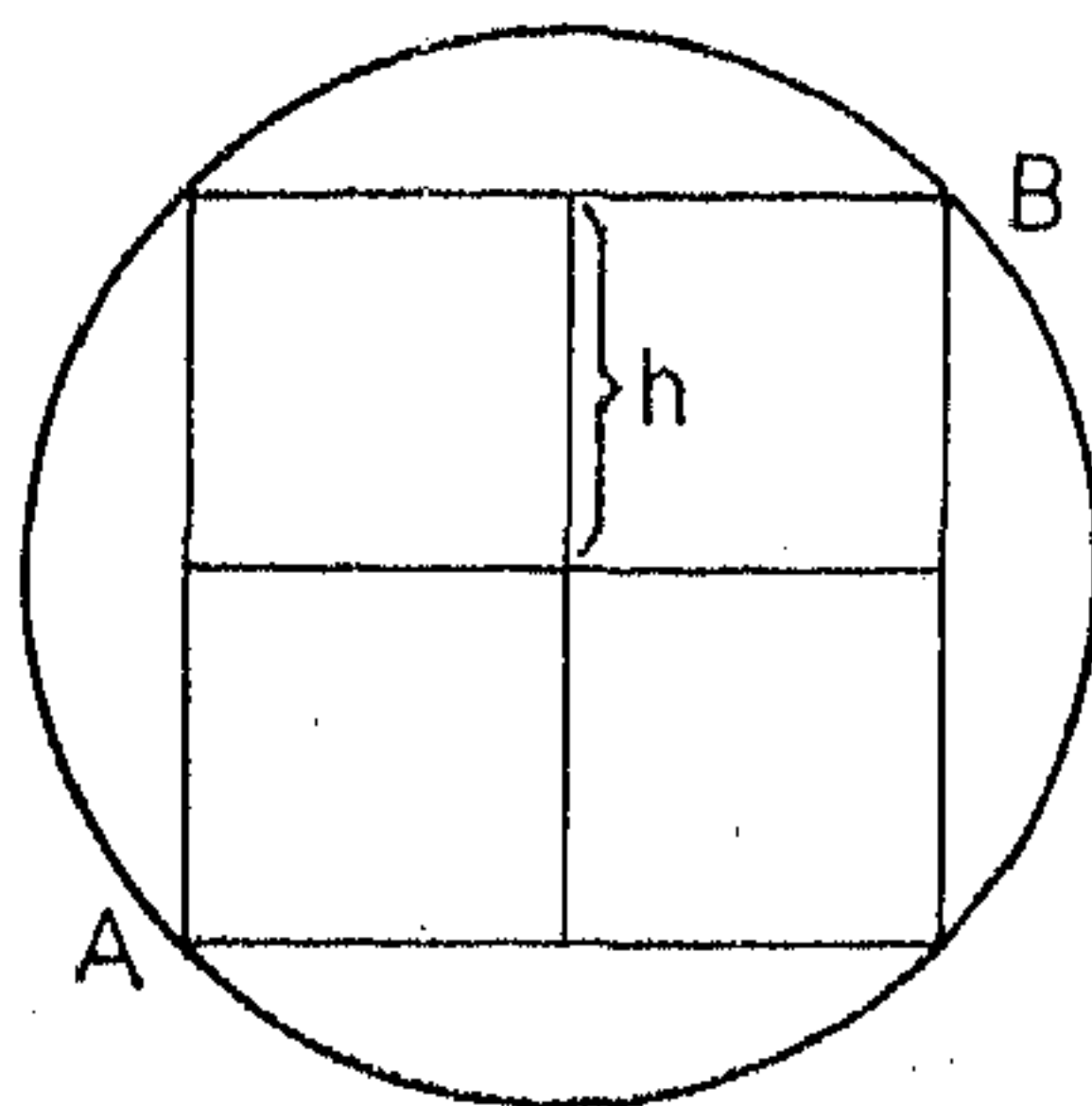


Fig.1.3

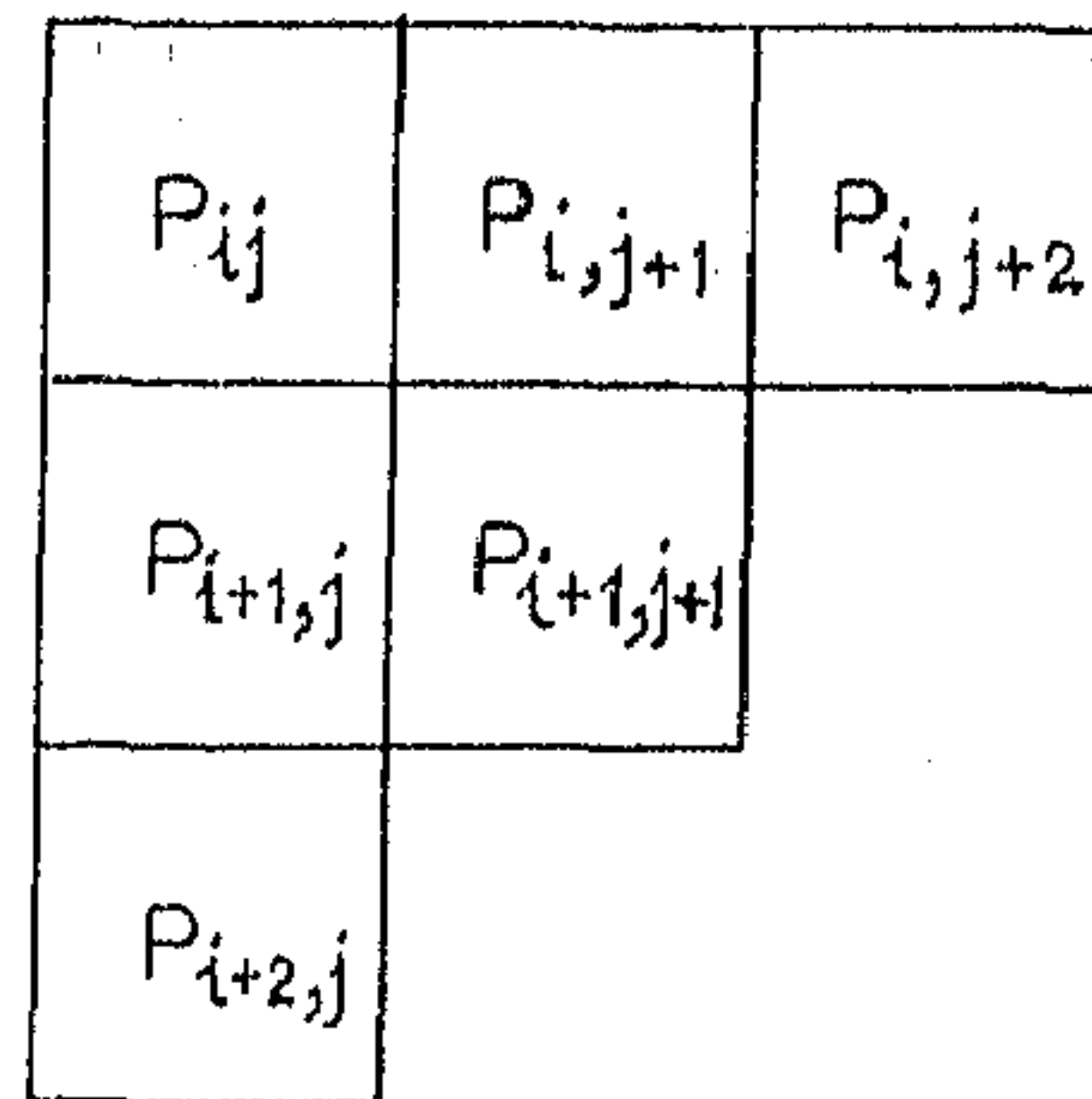


Fig. 1.4

CHAPTER 2

Some Operations on Bilevel/Binary Pictures

2.1 Introduction

An operation on a picture is defined as a mapping from one picture space into another. That is, an operation transforms a picture into another picture. The operations we will deal with in this chapter are translation, dilation (i.e. scaling by arbitrary factors more than or less than 1) and rotation (by arbitrary angles) on bilevel/binary pictures. The main aim of this chapter is to propose some algorithms for propagation (i.e. scaling by factors more than 1), shrinking (i.e. scaling by factors less than 1) and rotation of digital binary pictures. In the existing algorithms for propagation and shrinking of digital binary pictures (Rosenfeld and Kak, 1976; Young, Peverini, Verbeek and Otterloo, 1981), the scaling factors along the x and y axes are not usually the same. In fact, shrinking in such cases may be viewed as the peeling away of the boundary pixels of a binary picture much as one would peel layers of onion skin while propagation is a kind of inverse operation of shrinking. That is, propagation may be thought of as the inflation of the picture boundary as one would inflate a balloon. Since these existing algorithms for propagation and shrinking do not have the property of uniform change of scale

(i.e. the x and y co-ordinates do not increase/decrease by the same factor), they do not in general preserve the shape of the input picture. (The term 'shape' in this thesis will mean what remains of a picture after disregarding its position, size and orientation. The formal definitions of 2-dimensional curves and regions will be given in Chapters 4 and 5).

But the algorithms proposed for propagation and shrinking of binary pictures in this chapter have the property of uniform change of scale and hence preserve the shape of the input picture. (Now, this shape preservation is not entirely complete, since loss of information through these operations is inevitable due to the digital, in particular binary, nature of the input and output pictures). The shape preservation property of the propagation and shrinking algorithms proposed in this chapter is required because these algorithms will be used for shape matching of 2-dimensional regions in Chapter 5. Once again we point out that the terms 'propagation' and 'shrinking' as used in this chapter in particular and in this thesis in general, will mean shape-invariant propagation and shrinking.

Before discussing the aforesaid operations on digital binary pictures, we will define translation, dilation and rotation for continuous, i.e., analogue bilevel pictures in

order to clarify the basic natures of these operations.

Let A be an analogue bilevel picture as defined in Chapter 1.

We can write A as $\{(x_t, y_t) : t \in I\}$ where I is an uncountable index set.

Definition 2.1.1 The set $B = \{(x_t + a, y_t + b) : t \in I\}$ is said to be a translation of A where a and b are any real numbers.

Note that B also is an analogue bilevel picture according to the definition in Chapter 1. We write $B = \text{Tran}_{a,b}(A)$. Clearly, $B = \text{Tran}_{a,b}(A)$ if and only if $A = \text{Tran}_{-a,-b}(B)$.

Definition 2.1.2 The set $B = \{(cx_t, cy_t) : t \in I\}$ is said to be a dilation of A where c is any positive real number. c is the factor of dilation.

Note that B also is an analogue bilevel picture. We write $B = \text{Dil}_c(A)$. Clearly, $B = \text{Dil}_c(A)$ if and only if $A = \text{Dil}_{1/c}(B)$.

Definition 2.1.3 If $c > 1$, dilation is called propagation (with factor c) and if $c < 1$, dilation is called shrinking (with factor $1/c$). Thus, A is a propagation of B if and only if B is a shrinking of A (by the same factor).

Definition 2.1.4 The set $B = \{(x_t \cos \alpha - y_t \sin \alpha, x_t \sin \alpha + y_t \cos \alpha) : t \in I\}$ is said to be a rotation of A where α belongs to $[0, 2\pi)$.

Note that rotation of A is in the anticlockwise direction by angle α . B also is an analogue bilovel picture. We write $B = \text{Rot}_\alpha(A)$. Clearly, $B = \text{Rot}_\alpha(A)$ if and only if $A = \text{Rot}_{2\pi-\alpha}(B)$. Also, $B = \text{Rot}_\alpha(A), C = \text{Rot}_\gamma(B) \Rightarrow C = \text{Rot}_{\alpha+\gamma}(A)$.

It is clear from the definitions above that the original analogue picture can be reconstructed from the transformed (translated or dilated or rotated) one if the parameters (a, b or c or α) are known. But it is not true in the case of digital pictures as we will see later in this chapter.

In the analogue case, ^{the} operations are denoted by Tran, Dil and Rot while in the digital case, they will be denoted by Tra, Pro, Shr and Ro.

In the next section we will propose some algorithms for propagation and shrinking of digital binary pictures first by integer factors (Dutta Majumder, Parui and Chaudhuri, 1980) and then by arbitrary (real) factors. The algorithms with arbitrary factors first involve all the pixels of a picture and then involve only the boundary pixels of a picture.

In Section 2.3, some algorithms are proposed for rotating a digital binary picture by an arbitrary angle.

Here also the algorithms first involve all the pixels of a picture and then involve only the boundary pixels of a picture. Conditions under which these ^{propagation, shrinking and} rotation algorithms preserve the digital topology of the input pictures are also discussed.

The algorithms for propagation, shrinking and rotation of binary pictures proposed in Sections 2.2 and 2.3 will be used for shape matching of 2-dimensional regions in Chapter 5.

In the concluding section, we point out one drawback of the propagation algorithms proposed in Section 2.2 regarding preservation of the shape of the input picture and try to give a solution to remove it. There we propagate a picture through its boundary chain codes (not through its boundary pixels). Then we indicate that operations through boundary chain codes can be extended to shrinking and rotation of a picture also.

2.2 Some Algorithms for Propagation and Shrinking of Binary Pictures

A digital binary picture S is represented as $\{(i_k, j_k) : k = 1, 2, \dots, K\}$ where (i_k, j_k) denotes the co-ordinates of the k -th 1-pixel of the picture matrix (that is, the 1-pixels of the picture matrix are arranged in a 1-dimensional

array (row-wise)) and K is the total number of 1-pixels of S .

Definition 2.2.1 The total number of 1-pixels in $S = \{(i_k, j_k) : k=1, 2, \dots, K\}$, that is, K is the area of the digital binary picture S . It will be denoted by $\text{Area}(S)$.

Definition 2.2.2 The thinness of a connected digital binary picture is the minimum number of 1-pixels whose removal from the picture reduces the order of connectivity of the picture. The thinness of a hole of a connected digital binary picture is the minimum number of 1-pixels whose incorporation into the hole changes (either reduces or increases) the order of connectivity of the picture. (Note that with this definition of thinness of a hole, a hole with two 0-pixels has thinness 2. Thinness of only such a hole contradicts our intuition. Since this thinness should have been 1 intuitively). The thinness of the background of a digital binary picture is the minimum number of 1-pixels ^{the} whose incorporation into / background increases the order of connectivity of the picture (Fig.2.1). (We use the term 'thinness' here because in Chapter 3 we will use a term 'thickness' (of a binary picture) which will be different from this thinness).

Before presenting the algorithms for propagation and shrinking, we provide an algorithm for boundary detection of

a connected binary picture, which will be used later in this chapter for propagation and shrinking of a picture through only the boundary pixels of the picture.

Definition 2.2.3 A 1-pixel of a digital binary picture is called a boundary pixel if at least one of its 4-neighbours is a 0-pixel. The set of all boundary pixels of a connected binary picture is the boundary of the picture. A 1-pixel of a digital binary picture is called an interior pixel if it is not a boundary pixel.

Now we will see that the boundary of a connected binary picture A may have several disjoint parts. Suppose A has n holes H_i , $i=1,2,\dots,n$ (defined in Chapter 1).

Definition 2.2.4 A 1-pixel of a connected binary picture A is called (a) an outer boundary pixel if at least one of its 4-neighbours is a 0-pixel of the background and (b) an i-inner boundary pixel if at least one of its 4-neighbours is a 0-pixel of the i -th hole H_i . The set of all outer boundary pixels is called the outer boundary of A and the set of all i -inner boundary pixels is called the inner boundary of H_i or the i -th inner boundary of A for $i = 1,2,\dots,n$.

The outer boundary and an inner boundary or two inner boundaries may have some 1-pixels in common (Fig.2.1). It is

clear that the boundary of a ^{connected} picture is the union of its outer boundary and all its inner boundaries.

Now we will give an algorithm for finding the outer boundary and the inner boundaries, if any, of a connected binary picture A with the binary picture matrix M .

Algorithm 2.0 To find the boundary of a connected (8-connected) binary picture A (similar to an algorithm proposed by Ledley, 1964).

Step 1 Scan the pixels of M row-wise (i.e. top down and left to right) and find the first 1-pixel (i, j) . That is, the i -th row is the first row of M to have a 1-pixel and (i, j) is the first 1-pixel in the i -th row from the left. In fact, (i, j) will be a pixel on the outer boundary of A . Set $DIR = 2$.

Step 2 Let (i_1, j_1) be the neighbour of (i, j) in the direction DIR . If (i_1, j_1) is a 1-pixel, then this is the starting pixel of the boundary (Fig.2.1) and set $M(i_1, j_1) = 2$, $DIR_0 = DIR$, $DIR = DIR + 6 \pmod{8}$, $i_0 = i_1$, $j_0 = j_1$, $i = i_1$, $j = j_1$ and go to Step 3. Otherwise, set $DIR = DIR + 1 \pmod{8}$ and repeat Step 2. (Note that at least one neighbour (i_1, j_1) of (i, j) will be a 1-pixel since A is a connected picture).

Step 3 Let (i_1, j_1) be the neighbour of (i, j) in the direction DIR . If (i_1, j_1) is not a 0-pixel, then go to Step 4. If (i_1, j_1) is a 0-pixel, then set $DIR = DIR + 1 \pmod{8}$ and repeat

Step 3. (Note again that at least one neighbour of (i, j) will be a 1-pixel).

Step 4 If $M(i_1, j_1)=1$, then set $M(i_1, j_1)=2$, $i=i_1$, $j=j_1$, $DIR=DIR+6(\text{mod}.8)$ and go to Step 3. If $M(i_1, j_1)=2$ and $DIR=DIR_0$, $i_1=i_0$ and $j_1=j_0$, then stop. Otherwise, set $i=i_1$, $j=j_1$, $DIR=DIR+6(\text{mod}.8)$ and go to Step 3.

It is easy to see that the algorithm above traces the outer boundary of A in the clockwise direction. Now we will modify this algorithm in order to trace the inner boundaries, if any, of a binary picture. In fact, only Step 1 in the Algorithm 2.0 has to be modified for this purpose. The other three steps will remain the same. The inner boundaries are traced only after the outer boundary has been traced, that is, after all the outer boundary pixels have been given level 2. The new first step will be as follows.

Step 1* Scan the rows of M (top down and left to right) and locate the first 1-0 transition in the matrix M . (Note that for each pixel (i, j) on the outer boundary, $M(i, j)$ is already set to 2). Let the first transition involve the pixels (i, j) and $(i, j+1)$. That is, $M(i, j)=1$ and $M(i, j+1)=0$. (If there is no such transition, then the picture has no holes and hence no inner boundaries). Note that (i, j) will be a pixel on the inner boundary. Set $DIR=3$ and go to Step 2. (Next steps are the same as the ones in Algorithm 2.0).

It can be seen that the inner boundaries, unlike the outer boundary, are traced in the anticlockwise direction. In Fig.2.1, the outer boundary and an inner boundary of a digital picture are illustrated.

If it is required that the inner boundaries also be traced in the clockwise direction, then the following changes have to be made

- (i) 'Set DIR=1' instead of 'Set DIR=3' in Step 1*.
- (ii) 'Set DIR=DIR-1(mod.8)' instead of 'Set DIR=DIR+1(mod.8)' in Steps 2 and 3.
- (iii) 'Set DIR=DIR+2(mod.8)' instead of 'Set DIR=DIR+6(mod.8)' in Steps 2 and 4.

Again if it is required that the outer boundary be traced in the anticlockwise direction, then (ii) and (iii) of the above three changes alongwith 'Set DIR=5' instead of 'set DIR=2' in Step 1 have to be incorporated.

Now we will define the directional codes in terms of which a boundary of a binary picture can be expressed as a 1-dimensional string. Each directional code corresponds to one of the 8 neighbours of a pixel (Fig.2.2). The set of all directional codes $\{1,2,\dots,8\}$ will be denoted as I_8 . The outer boundary in Fig.2.1 can be expressed as 22233344555556677888888 112 (clockwise) while the inner boundary can be expressed as 4444322111177766654 (anticlockwise).

Now we will define translation, propagation and shrinking for digital binary pictures. Clearly, the output remains a digital binary picture after such operations. Suppose, $S = \{(i_k, j_k) : k=1, 2, \dots, K\}$ is a digital binary picture.

Definition 2.2.5 The set of 1-pixels $S_1 = \{(i_k + m, j_k + n) : k=1, \dots, K\}$ is said to be a translation of S where m and n are any integers. We write $S_1 = \text{Tra}_{m,n}(S)$. (It is assumed that the new elements $(i_k + m, j_k + n)$ do not fall outside the picture matrix).

Clearly, $S_1 = \text{Tra}_{m,n}(S)$ if and only if $S = \text{Tra}_{-m,-n}(S_1)$. Note that if A and B are two analogue pictures such that B is a translation of A , then $\text{Dig}_{\delta,p}(B)$ is not necessarily a translation of $\text{Dig}_{\delta,p}(A)$ where δ and p are the digitization parameters defined in Chapter 1. If $B = \text{Tran}_{a,b}(A)$ and $a = m\delta$, $b = n\delta$ where m, n are integers, then $\text{Dig}_{\delta,p}(B)$ is a translation of $\text{Dig}_{\delta,p}(A)$. In fact, $\text{Dig}_{\delta,p}(B) = \text{Tra}_{m,n}(\text{Dig}_{\delta,p}(A))$. Note that Tran and Tra indicate translation in analogue and digital cases respectively.

Definition 2.2.6 Let $S = \{(i_k, j_k) : k=1, \dots, K\}$ be a binary picture. Let m be a positive integer. Suppose, S_1 is a set of 1-pixels (i, j) such that $(i, j) \in S_1$ if and only if $mi_k \geq i > m(i_k - 1)$ and $mj_k \geq j > m(j_k - 1)$ for some (i_k, j_k) .

belonging to S . S_1 is said to be ^apropagation of S (Dutta Majumder, Parui and Chaudhuri, 1980). We write $S_1 = \text{Pro}_m(S)$. The factor of propagation is m .

This is analogous to propagation of an analogue bilevel picture (Definition 2.1.3). $\text{Area}(S_1) = m^2 \text{Area}(S)$. Clearly, after propagation a connected binary picture remains connected and a disconnected binary picture remains disconnected. In other words, the above propagation preserves the digital topological properties.

Definition 2.2.7 Let $S = \{(i_k, j_k) : k=1, \dots, K\}$ be a binary picture. Let m be a positive integer. Recall $M_{ij} = 1$ if and only if $(i, j) \in S$ and $M_{ij} = 0$ otherwise (Chapter 1). Let S_1 be a set of pixels (i_0, j_0) such that $(i_0, j_0) \in S_1$ if and only if $\sum_{i=m_{i_0}}^{m_{i_0}+m-1} \sum_{j=m_{j_0}}^{m_{j_0}+m-1} M_{ij} \geq pm^2$, where $m_{i_0} = m(i_0-1)+1$, $m_{j_0} = m(j_0-1)+1$ and p is the threshold value ($1 \geq p > 0$) which usually takes the value 0.5. S_1 is said to be a shrinking of S (Dutta Majumder, Parui and Chaudhuri, 1980). We write $S_1 = \text{Shr}_m(S)$. The factor of shrinking is m .

This is analogous to shrinking of an analogue picture (Definition 2.1.3). But $\text{Area}(S)$ is not necessarily m^2 times $\text{Area}(S_1)$. If $p=1$, then $\text{Area}(S) \geq m^2 \text{Area}(S_1)$. Though a connected analogue picture remains connected and a disconnected

analogue picture remains disconnected after shrinking, the same is not true for digital pictures. That is, a connected binary picture may become disconnected (Fig.2.3) and a disconnected binary picture may become connected (Fig.2.3) after shrinking. In other words, the above shrinking does not always preserve the digital topological properties of the input picture.

Though for an analogue picture A , shrinking of propagation of $A =$ propagation of shrinking of $A = A$ (with the same factor for propagation and shrinking), it is not always true for digital pictures. That is, propagation and shrinking are inverse operations of each other in the analogue case while they are so in the digital case only approximately. This is because some loss of information is inevitable in the digital case.

For segmentation of a digital binary picture, the method of first shrinking and then propagating the picture can sometimes be useful. For example,

1 1 1 1 . . 1 1 1 1		1 1 1 1 . . 1 1 1 1
1 1 1 1 . 1 1 1 1 1	1 1 . 1 1	1 1 1 1 . . 1 1 1 1
1 1 1 1 1 1 1 1 1	1 1 . 1 1	1 1 1 1 . . 1 1 1 1
1 1 1 1 . . 1 1 1 1		1 1 1 1 . . 1 1 1 1

where $m=2$ for both shrinking and propagation and $p = 0.5$ and the dots indicate 0-pixels. The same method can be useful for removing noise from a digital binary picture.

Now when two digital binary pictures with different areas are to be matched by some kind of template matching, either the larger picture is to be reduced through shrinking or the smaller picture is to be enlarged through propagation in order to make the two areas same (at least approximately). Note that the ratio of the two areas is arbitrary (that is, not necessarily a perfect square or even an integer). The propagation or shrinking defined above involves only integer-valued factors, and hence is inadequate for our matching purpose. We will now propose some algorithms for propagation and shrinking of digital binary pictures by arbitrary (real) factors (Parui and Dutta Majumder, 1983E).

Algorithm 2.1 To propagate a digital binary picture by an arbitrary (real) factor.

From Fig.2.4 it is clear that each row of the propagated (output) picture matrix, say N , partially contains one or two rows of the input matrix, say M . The same is true for columns also. Let the I -th row of N contain the $IROW(I)$ -th and $IROW(I)+1$ st rows of M with weights $R1(I)$ and $S1(I)$ respectively, where $R1(I)+S1(I)$ is the size of the pixels of the output picture assuming the size of the pixels of the input picture to be 1. The size of the output pixel is, say, $FACT$ which is less than 1. That is, the factor of propagation here is $1/FACT$.

Suppose for the J -th column of N , the corresponding variables are $JCOL(J)$, $R2(J)$, $S2(J)$, where again $R2(J)+S2(J)=FACT$.

Step 1 $P1 = 0$
 $P2 = FACT$
 $I = 1$
 $IROW(1) = 1$
 $R1(1) = FACT$
 $S1(1) = 0$

Step 2 $I = I+1$
 $P1 = P2$
 $P2 = P2 + FACT$
 If there is an integer INT such that $P2 > INT > P1$, set $IROW(I) = INT$
 $R1(I) = INT - P1$
 $S1(I) = P2 - INT$
 Repeat Step 2.
 If there is an integer INT such that $INT \geq P1$, $P2 \geq INT + 1$, set $IROW(I) = INT$
 $R1(I) = FACT$
 $S1(I) = 0$
 Repeat Step 2.

Stop when all the rows of the output matrix N are exhausted. (Note that since $P2 - P1 = FACT < 1$ above, either $P1$ and $P2$ lie between two consecutive integers or exactly one integer lies between $P1$ and $P2$).

In the same way as above, $JCOL(J)$, $R2(J)$, $S2(J)$ are computed for the output columns. Now, the construction of the output matrix N follows.

For each row I and each column J of the output matrix N ,
 let $X(I, J) = R1(I)R2(J)M(IROW(I), JCOL(J))$
 $+ R1(I)S2(J)M(IROW(I), JCOL(J)+1)$
 $+ S1(I)R2(J)M(IROW(I)+1, JCOL(J))$
 $+ S1(I)S2(J)M(IROW(I)+1, JCOL(J)+1)$

where M is the binary input matrix. (This X value of the output pixels will be used later in the propositions in this chapter).

Make $N(I, J) = 1$ if $X(I, J) \geq pFACT^2$
 $= 0$ otherwise

where $1 \geq p > 0$. In our algorithm, p is taken to be 0.5 always.

Note that the above algorithm is flexible in the sense that the factors of propagation along the x and y axes need not be the same. In other words, we can use $FACT_1 < 1$ and $FACT_2 < 1$ for the x and y axes respectively, where $FACT_1$ and $FACT_2$ are not necessarily the same. In that case, $N(I, J)$ should be redefined as

$$N(I, J) = 1 \text{ if } X(I, J) \geq pFACT_1 FACT_2$$

$$= 0 \text{ otherwise.}$$

But in this thesis, we are concerned with propagation with uniform change of scale only. In other words, $FACT_1$ and $FACT_2$ are always the same in this thesis.

It is clear that propagation by the Algorithm 2.1 does always not/entirely preserve the shape of the input picture. Distortions may be there near the boundary of the picture. But when the perimeter is small compared to the area of the picture, shape distortions will be low.

Now, we will prove some properties of the propagation algorithm above (i.e. Algorithm 2.1), regarding preservation of the shape of the input binary picture.

The parameter FACT in the Algorithm 2.1 will be denoted by α in the subsequent propositions and their proofs.

Proposition 2.2.1 If a hole of a binary picture has only two 0-pixels, then after propagation with arbitrary factors ($\alpha=0.5$), the hole does not disappear, that is, gives rise to at least one output 0-pixel and the resulting output 0-pixels (if more than one in number) are 4-connected.

Proof. From the definition of a hole (Chapter 1) the two 0-pixels of this hole are 4-connected. In Fig.2.5, the unbroken lines and broken lines indicate the input and output respectively pixels/. Without loss of generality, we assume that the two input 0-pixels (i.e. P and Q in Fig.2.5) are horizontally connected and the size of the output pixels, i.e., α is quite close to and less than 1. (That is, the factor of propagation is only slightly more than 1). In Fig.2.5, vertically 2 or 3 output pixels and horizontally 3 or 4 output pixels can have intersection with the input hole. In the present proof, we will consider 2 such vertical output pixels and 3 such horizontal pixels. The other cases can be proved in a similar way. Let A, B, C, D, E and F be the output pixels having intersection with

the input hole (Fig.2.5). In Algorithm 2.1, X value of a pixel is defined to be the area of the black part of the pixel. Now, we define Y value of a pixel to be the area of the white part of the pixel. For example, for the output pixel A in Fig.2.5, $X(A) + Y(A) = a^2$.

In Fig.2.5, ϵ can be positive or negative, but it is close to zero. Now $\delta > 0$ and $a \geq 1 - \delta$, since the size of the output pixels is a . For the same reason $a \geq 1 - a + \delta$. Now, we will find Y values of the output pixels A, B, C, D, E and F .

$$\begin{aligned} Y(A) &= \frac{1}{2} (1 + \epsilon)(1 - \delta) & Y(B) &= \frac{1}{2} (1 + \epsilon)a \\ Y(D) &= \frac{1}{2} (1 - \epsilon)(1 - \delta) & Y(E) &= \frac{1}{2} (1 - \epsilon)a \\ \text{and } Y(C) &= \frac{1}{2} (1 + \epsilon)(1 - a + \delta) \} & \text{provided } a &\geq 1 - a + \delta \\ Y(F) &= \frac{1}{2} (1 - \epsilon)(1 - a + \delta) \} & \text{since } a &\text{ is the size of the} \\ & & & \text{output pixels} \end{aligned}$$

Without loss of generality, let $\epsilon \geq 0$. Then $Y(B) > a^2/2$ since $1 > a$. So B is a 0-pixel. Now, we will show that

(i) F is a 0-pixel implies E is a 0-pixel

and (ii) D is a 0-pixel implies E is a 0-pixel.

(i) is true because $Y(E) \geq Y(F)$ since $a \geq 1 - a + \delta$ from above and

(ii) is true because $Y(E) \geq Y(D)$ since $a \geq 1 - \delta$ from above.

Thus in the output we have at least one 0-pixel, that is, the pixel B and 4-connectedness of the output is trivial from (i) and (ii) above, since the levels of the pixels A and C do

not matter as far as 4-connectedness is concerned.

In the beginning we assumed a is close to 1. Now if the value of a is still less, that is, if the factor of propagation is still larger, then also the above result is obviously true.

Hence the Proposition 2.2.1.

But if a hole of a binary picture has only one 0-pixel, then after propagation ($p=0.5$) the hole may disappear provided a (the size of the output pixels) is sufficiently large. An example is given in Fig.2.6.

Proposition 2.2.2 If a hole of a binary picture has only one 0-pixel, then after propagation ($p=0.5$), the hole does not disappear, that is, gives rise to at least one output 0-pixel and the resulting output 0-pixels (if more than one in number) are 4-connected, provided a (the size of the output pixels) is strictly less than $1/\sqrt{2}$ (i.e. the factor of propagation is more than $\sqrt{2}$).

Proof. In Fig.2.7, the unbroken lines indicate the input 0-pixel P while the broken lines indicate the output pixels. Let us now assume that only four output pixels (A, B, C and D) have intersection with P . Other cases can be proved in a similar way. From Fig.2.7 ($a \geq x, y \geq 0$),

$$Y(A) = (1-x)(1-y) \quad Y(B) = x(1-y)$$

$$Y(C) = (1-x)y \quad Y(D) = xy$$

From Fig. 2.7, it is clear that either $x \geq 1/2$ or $(1-x) \geq 1/2$ and either $y \geq 1/2$ or $1-y \geq 1/2$.

Case 1 $x \geq 1/2$ and $y \geq 1/2$.

Then $Y(D) = xy \geq \frac{1}{4} > \frac{a^2}{2}$ since $1/\sqrt{2} > a$

So D is a 0-pixel. Now we will show that A is a 0-pixel implies both B and C are 0-pixels. This is true because $Y(B) \geq Y(A)$ and $Y(C) \geq Y(A)$ since $x \geq 1-x$ and $y \geq 1-y$ because $x \geq 1/2$ and $y \geq 1/2$.

Thus in the output there is at least one 0-pixel and the output 0-pixels are 4-connected, if their number is more than 1.

The other three cases can be dealt with in a similar way. Hence the Proposition 2.2.2.

Proposition 2.2.3 If a hole of a binary picture has three 0-pixels which fall neither in a row nor in a column (Fig. 2.8), then after propagation with arbitrary factors ($p=0.5$), the hole does not disappear and the resulting output 0-pixels (if more than one in number) are 4-connected.

Proof. In Fig. 2.8, the unbroken lines and broken lines respectively indicate the input and output pixels/. We assume that a (the size of the output pixels) is quite close to and less than 1. That is, the factor of propagation is only slightly more than 1. If the value of this factor is still more, then the result in this proposition is still more obvious. The input 0-pixels are

P, Q and R (Fig.2.8). We assume that Q does not entirely contain an output pixel, because in that case the proof of the present proposition is trivial. So, the output pixels are taken as they are in Fig.2.8. Let us assume that eight output pixels A, B, C, D, E, F, G and H entirely contain the three input 0-pixels P, Q and R (Fig.2.8). Let $a > x > 0$ and $a > y > 0$.

We will first show that at least one of the output pixels D and H is a 0-pixel. Suppose, H is a 1-pixel. So, $Y(H) = xy \leq a^2/2$. Now, $X(D) = (a-1+x)(a-1+y) < xy$ (since $1 > a$) $\leq a^2/2$. So, D is a 0-pixel.

Again suppose D is a 1-pixel. So, $X(D) = (a-1+x)(a-1+y) > a^2/2$. Now, $Y(H) = xy > (a-1+x)(a-1+y) > a^2/2$. So, H is a 0-pixel.

Case 1 H is a 0-pixel. That is, $xy > a^2/2$. So, $Y(G) = ay > xy > a^2/2$ and $Y(E) = ax > xy > a^2/2$. Hence G and E are output 0-pixels. Now, from the Proposition 2.2.1, 4-connectivity of the output 0-pixels is clear.

Case 2 D is a 0-pixel. In that case, 4-connectivity of the output 0-pixels is clear from the proof of Proposition 2.2.1.

Hence the Proposition 2.2.3.

Proposition 2.2.4. Disconnected components of a binary picture can not get connected after propagation ($p=0.5$) with arbitrary factors.

Proof. It is clear from the Propositions 2.2.1 and 2.2.3.

Proposition 2.2.5 A hole of a binary picture does not disappear and remains 4-connected after propagation ($p=0.5$) with arbitrary factors, provided the hole has more than one 0-pixel.

Proof. It is clear from the Propositions 2.2.1 and 2.2.3.

Proposition 2.2.6 A connected binary picture remains connected after propagation ($p=0.5$) if a (the size of the output pixels) is less than or equal to 0.5. That is, the factor of propagation here is greater than or equal to 2.

Proof. In Fig.2.9, P and Q are ^{the} only two 1-pixels. Other pixels, in particular the pixels P_1 and Q_1 , are 0-pixels. We consider this because it is the weakest connectivity that a connected component (8-connected) can have. We will prove the above proposition for this input ^{picture} (Fig.2.9). So, for any other connected component (8-connected), the proposition will be more obvious.

Let the output pixel A contain the centre point O . Let A_1, A_2, A_3 and A_4 be other output pixels as shown in Fig.2.9. The five output pixels A, A_1, A_2, A_3 and A_4 are contained in the input pixels P, Q, P_1 and Q_1 since $0.5 \geq a$. Let x and y be as shown in Fig.2.9 such that $a \geq x \geq 0$ and $a \geq y \geq 0$. Now,

$$X(A) = xy + (a-x)(a-y)$$

$$X(A_1) = ay, X(A_2) = a(a-x), X(A_3) = a(a-y) \text{ and } X(A_4) = ax$$

Now since $a \geq x \geq 0$ and $a \geq y \geq 0$, either $x \geq a/2$ or $a/2 > x$ and either $y \geq a/2$ or $a/2 > y$.

Case 1 $x \geq a/2$ and $y \geq a/2$.

Without loss of generality, let us assume $y = x + \epsilon$ for some $\epsilon \geq 0$.

$$\begin{aligned} \text{So, } X(A) &= xy + (a-x)(a-y) \\ &= x^2 + \epsilon x + (a-x)^2 - \epsilon(a-x) \\ &= x^2 + (a-x)^2 + \epsilon(2x-a) \\ &\geq x^2 + (a-x)^2 \text{ since } x \geq a/2 \end{aligned}$$

suppose, $f(x) = x^2 + (a-x)^2$

$$\text{So, } f'(x) = 2x - 2(a-x) = 4x - 2a$$

$$\text{and } f''(x) = 4$$

Thus the minimum value of $f(x)$ is attained when $x = a/2$. That is, the minimum value of $f(x)$ is $a^2/2$. Thus, $X(A) \geq a^2/2$. Also, $X(A_1) \geq a^2/2$ and $X(A_4) \geq a^2/2$. So, A , A_1 and A_4 are output 1-pixels. Hence the above proposition is true.

Case 2 $a/2 > x$ and $a/2 > y$.

The proof for this is similar to the one in Case 1. Here A , A_2 and A_3 are output 1-pixels.

Case 3 $x \geq a/2$ and $a/2 > y$.

So, $X(A_4) = ax \geq a^2/2$ and $X(A_3) = a(a-y) > a^2/2$. So A_3 and A_4 are output 1-pixels. Since A_3 and A_4 are 8-connected (Fig.2.9), the above proposition is true.

Case 4 $a/2 > x$ and $y \geq a/2$.

In the same way as in Case 3, it can be shown that A_1 and A_2 are output 1-pixels.

Hence the Proposition 2.2.6.

But if the value of a (the size of the output pixels) is more than 0.5, a connected binary picture may get disconnected after propagation ($p=0.5$). One such example is given Fig.2.10.

Proposition 2.2.7 A connected binary picture with thinness more than 1 remains connected after propagation ($p=0.5$) with arbitrary factors.

Proof. The weakest connectivity that a connected binary picture with thinness 2 can have is shown in Fig.2.11. Now, that the picture shown in Fig.2.11 remains connected (in fact, 4-connected) after propagation with arbitrary factors, is clear from the Proposition 2.2.3.

Hence the Proposition 2.2.7.

Proposition 2.2.8 The topological properties of a binary picture remain unchanged after propagation ($p=0.5$) if a is less than or equal to 0.5, that is, if the factor of propagation is more than or equal to 2.

Proof. It is clear from the Propositions 2.2.2 and 2.2.6.

Proposition 2.2.9 If a binary picture is such that each of its components has thinness more than 1 and none of its components has a hole with only one 0-pixel, then the topological

properties of the ^{input} picture do not change after propagation ($p=0.5$) with arbitrary factors.

Proof. It is clear from Propositions 2.2.4, 2.2.5 and 2.2.7.

But no change in topological properties does not say much about the change in the general shape of a picture as pointed out earlier in Chapter 1. For example, in Fig.2.12, though the pictures have the same topological properties, the picture in Fig.2.12(b) can never be a propagation of the picture in Fig.2.12(a). In other words, the propagation proposed in Algorithm 2.1 retains not only the topological properties but all the shape properties of the input picture if the distortions due to the digital nature of a picture are ignored, provided the thinness and the size of the holes (if any) of the input picture are more than 1, as is clear from the preceding propositions. (As mentioned earlier, the shape of a picture means, in this thesis, what remains of the picture after disregarding its position, size and orientation).

So far we have discussed about propagating a binary picture by arbitrary factors. Now we will propose an algorithm for shrinking a binary picture by arbitrary factors, which is a kind of inverse of Algorithm 2.1. Also, as we will see later, the shrinking algorithm proposed here may drastically change the shape of the input picture without disturbing its topological

properties (Fig.2.12).

Algorithm 2.2. To shrink a digital binary picture by an arbitrary (real) factor.

Let the size of the pixels of the input picture be 1. Let $FACT > 1$ be the factor of shrinking. That is, the size of the pixels of the output picture is $FACT$. Suppose, $INTGR+1 \geq FACT > INTGR$ for some integer $INTGR$. Let $INT=INTGR+2$. It is clear that each row of the output picture matrix N partially contains at least $INTGR+1$ and at most INT rows of the input matrix M . The same is true for columns also. Let the I -th row of N contain the $IROW(I)$ -th, $IROW(I)+1$ st, ... $IROW(I)+INTGR+1$ st rows of M with weights $R(I, IROW(I))$, $R(I, IROW(I)+1)$, ... $R(I, IROW(I)+INT-1)$ respectively. The corresponding variables for columns are $JCOL(J)$, $S(J, JCOL(J))$, $S(J, JCOL(J)+1)$, ... $S(J, JCOL(J)+INT-1)$.

```

Step 1      P1 = 0
            P2 = FACT
            I = 1
            IROW(1) = 1
            R(1,1) = 1
            .
            .
            R(1,INTGR) = 1
            R(1,INTGR+1) = FACT - INTGR

```

```

Step 2      I = I+1
            P1 = P2

```


$P2 = P2 + FACT$

Now there will be integers $INT_1, \dots, INT_k$ such that $P2 \geq INT_1 > \dots > INT_k \geq P1$ where k is either $INTGR$ or $INTGR+1$.

Set $IROW(I) = INT_k$

$R(I, INT_i) = 1$ for $i=1, \dots, k-1$

$R(I, INT_k) = INT_k - P1$

$R(I, INT_1+1) = P2 - INT_1$

Repeat Step 2 until all the rows of the output matrix N are exhausted.

In the same way as above, $JCOL(J)$, $S(J, JCOL(J))$, $\dots$ $S(J, JCOL(J)+INT-1)$ are computed for the output columns. The construction of the output matrix N follows.

For each row I and each column J , the element $N(I, J)$ of the output matrix is computed as —

Let $U = IROW(I) - 1$

$V = JCOL(J) - 1$

$T(I, J) = 0$

DO $K = 1, INT$

DO $L = 1, INT$

$T(I, J) = R(I, U+K)S(J, V+L)M(U+K, V+L)$

where M is the input matrix.

Set $N(I, J) = 1$ if $T(I, J) \geq pFACT^2$
 $= 0$ otherwise

where $1 \geq p > 0$. In our algorithm, p is always taken to be 0.5.

Like Algorithm 2.1, the above algorithm also is flexible in the sense that the factors of shrinking along the x and y

axes need not be the same. In other words, we can use $FACT_1, FACT_2$ for the x and y axes respectively, where $FACT_1$ and $FACT_2$ are not necessarily the same. In that case, $N(I, J)$ should be redefined as

$$N(I, J) = 1 \text{ if } T(I, J) \geq p \cdot FACT_1 \cdot FACT_2$$

$$= 0 \text{ otherwise.}$$

But in this thesis, we are concerned with shrinking with uniform change of scale only. In other words, $FACT_1$ and $FACT_2$ are always the same in this thesis.

Unlike propagation, shrinking of a binary picture does not in general preserve the topological properties of a binary picture. For example, a connected picture may become disconnected and a disconnected picture may become connected through shrinking (Fig.2.3)

But if we put some heavy restrictions on the input picture, then the above shrinking algorithm also preserves the shape properties of the input picture. From the preceding propositions about the propagation (Algorithm 2.1), we can state the following proposition (analogous to Proposition 2.2.9 for propagation) —

Proposition 2.2.10 If a binary picture is such that the thickness of its background, the thickness of each of its components and the thickness of each of its holes (if any) are sufficiently large compared to the factor of shrinking, then the topological properties of the input picture remain unchanged after shrinking.

Under the above conditions, all the shape properties of the input picture are retained (if the distortions due to the digital nature of a picture are ignored) as was the case for propagation. But it is clear that with increasing factor of shrinking, the above conditions become more stringent. That is, the shape properties of the input picture are almost always preserved through propagation (Algorithm 2.1) while it is not so through shrinking (Algorithm 2.2). Hence, for matching purpose (in Chapter 5), for two pictures of unequal areas, it is always better to propagate the smaller picture than to shrink the larger one.

The Algorithms 2.1 and 2.2 above have been suggested for dilation of binary pictures by arbitrary (real) factors with an aim to match shapes of 2-dimensional regions in Chapter 5. But these two algorithms involve all the pixels of both input and output pictures. We will next modify the Algorithms 2.1 and 2.2 so that almost none of the interior input pixels will be involved in the modified algorithms. And for the output picture, only the boundary pixels will be found (Fig.2.13). For filling in the interior of the output picture, there are already standard algorithms (Pavlidis, 1982). For our purpose, we use the contour filling algorithm by connectivity (nonrecursive) given in Chapter 8 of the book 'Algorithms for Graphics and Image Processing' (Pavlidis, 1982).

We will first provide an algorithm which gives only the boundary pixels of a propagated version of a digital binary picture. This algorithm is based on the Algorithms 2.0 and 2.1.

Algorithm 2.3 To propagate the boundary of a binary picture by an arbitrary (real) factor ($p=0.5$).

Suppose for the time being that the input picture is with thickness more than 1. simply connected/ The outer boundary pixels of the output picture will be traced in the clockwise direction as in Algorithm 2.0. The pixels of the output matrix N are 0 initially.

Step 1 Scan the pixels of the output picture matrix N row-wise (i.e. top down and left to right) and find the first pixel (I, J) such that $X(I, J)$ (defined in the Algorithm 2.1) is greater than or equal to $pFACT^2$. Set $N(I, J)=1$. In fact, (I, J) will be an outer boundary pixel of the output picture, more specifically, the left uppermost boundary pixel. Set $DIR=2$. (The factor of propagation here is $1/FACT$, as in the Algorithm 2.1).

Step 2 Let (I_1, J_1) be the neighbour of (I, J) in the direction DIR . If (I_1, J_1) is such that $X(I_1, J_1) \geq pFACT^2$, then set $I_0=I_1$, $J_0=J_1$, $N(I_1, J_1)=1$, $DIR_0=DIR$, $DIR=DIR+6 \pmod{8}$, $I=I_1$, $J=J_1$ and go to Step 3. Otherwise, set $DIR=DIR+1 \pmod{8}$ and repeat Step 2. (Note that at least one neighbour (I_1, J_1) of (I, J) will be a 1-pixel of N , i.e., will satisfy $X(I_1, J_1) \geq pFACT^2$, because (I, J) can not be an isolated 1-pixel of the

output picture since the input picture is connected and propagation preserves connectivity property).

Step 3 Let (I_1, J_1) be the neighbour of (I, J) in the direction DIR. If (I_1, J_1) is such that $X(I_1, J_1) \geq pFACT^2$, then go to Step 4. Otherwise, set $DIR = DIR+1 \pmod{8}$ and repeat Step 3. (Note again that at least one neighbour of (I, J) will be a 1-pixel of N).

Step 4 If $I_1 = I_0$, $J_1 = J_0$ and $DIR = DIR_0$ then stop. Other-wise, set $N(I_1, J_1) = 1$, $I = I_1$, $J = J_1$, $DIR = DIR+6 \pmod{8}$ and go to Step 3.

The Algorithm 2.3 is sure to terminate since I_0, J_0 and DIR_0 have to repeat sometime in Step 4. It is clear that the output picture $\{(I, J) : N(I, J) = 1\}$ is the propagated boundary of the input picture (Fig.2.14).

So far we have assumed that the input picture is simply connected, i.e., without any holes. Now if there are holes then the above algorithm has to be repeated as many times as the number of holes. But in that case the search for the starting pixel (one for each of the holes) on the output boundary (Step 1) becomes pretty costly in terms of the computing time. Particularly if the number of holes in the input picture is unknown, then the Algorithm 2.3 is highly uneconomical and perhaps no better than the Algorithm 2.1. But if the input picture is known to be simply connected then the Algorithm 2.3 is more

economical than the Algorithm 2.1 because the interior pixels of the output picture do not have to be dealt with and there is only one search for the starting pixel (i.e. there is only the outer boundary and no inner boundaries). In order to obtain the interior pixels of the output picture, we use, as mentioned earlier, the contour filling algorithm by connectivity (Pavlidis, 1982).

Now we will propose an algorithm which gives only the boundary pixels of a shrunk (reduced) version of a digital binary picture. This algorithm is based on the Algorithms 2.0 and 2.2. Here, the pixels of the output matrix N are 0 initially.

Algorithm 2.4 To shrink the boundary of a binary picture by an arbitrary (real) factor ($p=0.5$).

Recall that a connected digital binary picture may become disconnected or a disconnected digital binary picture may become connected through shrinking. We will assume for the time being that the output picture is simply connected.

That is, the output picture does not have any holes (although the input picture may be disconnected).

Step 1 Scan the pixels of the output picture matrix N row-wise (i.e. top down and left to right) and find the first pixel (I, J) such that $T(I, J)$ (defined in the Algorithm 2.2) is greater than or equal to $pFACT^2$. Set $N(I, J) = 1$. In fact, (I, J) will be an outer boundary pixel of the output picture, more specifically, the left uppermost boundary pixel. Set $DIR = 2$.

(The factor of shrinking here is FACT, as in the Algorithm 2.2).

Step 2 : Let (I_1, J_1) be the neighbour of (I, J) in the direction DIR. If (I_1, J_1) satisfies $T(I_1, J_1) \geq pFACT^2$, then set $N(I_1, J_1) = 1$, $I_0 = I_1$, $J_0 = J_1$, $DIR_0 = DIR$, $DIR = DIR + 6 \pmod{8}$, $I = I_1$, $J = J_1$ and go to Step 3. Otherwise, set $DIR = DIR + 1 \pmod{8}$ and repeat Step 2. (Note that at least one neighbour (I_1, J_1) of (I, J) will be a 1-pixel of N, i.e., will satisfy $T(I_1, J_1) \geq pFACT^2$, because (I, J) can not be an isolated 1-pixel of the output picture since the output picture is assumed to be connected).

Step 3 : Let (I_1, J_1) be the neighbour of (I, J) in the direction DIR. If (I_1, J_1) satisfies $T(I_1, J_1) \geq pFACT^2$, then go to Step 4. Otherwise, set $DIR = DIR + 1 \pmod{8}$ and repeat Step 3. (Note again that at least one neighbour of (I, J) will be a 1-pixel of N).

Step 4 : If $I_1 = I_0$, $J_1 = J_0$ and $DIR = DIR_0$ then stop. Otherwise, set $N(I_1, J_1) = 1$, $I = I_1$, $J = J_1$, $DIR = DIR + 6 \pmod{8}$ and go to Step 3.

The Algorithm 2.4 is sure to terminate since I_0 , J_0 and DIR_0 have to repeat sometime in Step 4. It is clear that the output picture $\{(I, J) : N(I, J) = 1\}$ is the shrunk (reduced) boundary of the input picture (Fig.2.15).

So far we have assumed that the output picture is simply connected i.e., it has only one outer boundary and no

inner boundaries. But in general the output picture may have many holes and many connected components (depending on the input picture and the factor of shrinking) so that there will be many boundaries (both inner and outer). In that case, shrinking through boundary is perhaps much less economical than the Algorithm 2.2. But if it is known that the input picture is simply connected and that the thinness of the input picture and the thinness of its background are sufficiently large compared to the factor of shrinking, then the output picture has only one connected component without any holes (from Proposition 2.2.10). That is, there will be only one outer boundary in the output. In that case, the Algorithm 2.4 will be more economical than the Algorithm 2.2. The interior pixels of the output picture are obtained by an algorithm mentioned earlier.

From the preceding algorithms, it is clear that the Algorithm 2.3 alongwith a contour filling algorithm and the Algorithm 2.1 give rise to the same propagated output picture for any input picture. Similarly, the Algorithm 2.4 alongwith a contour filling algorithm and the Algorithm 2.2 give the same shrunk (reduced) output picture for any input picture.

2.3 Some Algorithms for Rotation of Binary Pictures and Their Properties

If a pixel (i, j) of a picture is rotated by an angle α in the anticlockwise direction, we get the 2-dimensional point $(i \cos \alpha - j \sin \alpha, i \sin \alpha + j \cos \alpha)$, where the co-ordinates are not necessarily integers. We take the nearest integers I and J of $(i \cos \alpha - j \sin \alpha)$ and $(i \sin \alpha + j \cos \alpha)$ respectively where $I + 0.5 \geq i \cos \alpha - j \sin \alpha > I - 0.5$ and $J + 0.5 \geq i \sin \alpha + j \cos \alpha > J - 0.5$. We call (I, J) the rotated pixel of the pixel (i, j) (Fig.2.16) and the point $(i \cos \alpha - j \sin \alpha, i \sin \alpha + j \cos \alpha)$ the rotated version of the pixel (i, j) .

Definition 2.3.1 Let $S = \{(i_k, j_k) : k=1, \dots, K\}$ be a binary picture. Let α be an angle belonging to $[0, 2\pi)$. Let S_1 be the set of pixels $\{(I, J) : (I, J) \text{ is the rotated pixel of } (i_k, j_k) \text{ belonging to } S\}$. The rotation is by angle α in the anticlockwise direction. We call the binary output picture S_1 a rotation of S (by angle α in the anticlockwise direction (Fig.2.17)). This is analogous to rotation of an analogue bilevel picture (Definition 2.1.4).

If some I and J become negative, then by adding two suitable constants to the two co-ordinates of all (I, J) , all of them can be made positive. In other words, a proper translation of S_1 will serve the purpose. So, from now on, we will ignore the signs of I and J .

But the rotation algorithm based on the above definition of rotation has a drawback. That is, a connected binary picture without any holes may contain some holes after rotation (Fig.2.18). We will now modify the above definition of rotation of a binary picture so that the above drawback is removed.

Definition 2.3.2 Let $S = \{(i_k, j_k) : k=1, \dots, K\}$ be a binary picture. Let α be an angle in $[0, 2\pi)$. Suppose, S_1 is the set of pixels $\{(I, J) : (i_k, j_k) \text{ is the rotated pixel of } (I, J) \text{ by angle } (2\pi - \alpha) \text{ in the anticlockwise direction for some } (i_k, j_k) \text{ in } S\}$. S_1 is called a rotation of S. Here also the rotation is by angle α in the anticlockwise direction. We write $S_1 = \text{Ro}_\alpha(S)$.

Although for analogue bilevel pictures A, B and C, (i) $B = \text{Rot}_\alpha(A)$ if and only if $A = \text{Rot}_{2\pi - \alpha}(B)$ and (ii) $B = \text{Rot}_\alpha(A)$ and $C = \text{Rot}_\beta(B)$ implies $C = \text{Rot}_{\alpha + \beta}(A)$ for all angles α and β , they are not always true in the case of rotation of digital binary pictures. Note that in the digital case, (i) and (ii) above are true if α and β are multiples of $\pi/2$ (Parui and Dutta Majumder, 1983E).

The difference between the above two definitions is that in the former each input 1-pixel is rotated by angle α in the anticlockwise direction and the nearest pixel (of its rotated version) in the output picture is made a 1-pixel, while in the latter each output pixel is rotated by angle α in the clockwise direction and it is made a 1-pixel, if the nearest pixel (of its rotated version) in the input picture is a 1-pixel and a 0-pixel, otherwise.

Note that the algorithm for rotation based on Definition 2.3.1 involves all the pixels of the input picture while that based on Definition 2.3.2 involves all the pixels of the output picture. Now, if the size of the input picture

matrix M is $n \times n$, then the minimum size of the output picture matrix N which contains all possible rotations of M around its centre $(n/2, n/2)$, is $\sqrt{2}n \times \sqrt{2}n$. (The input matrix M needs the maximum size of the output matrix N , i.e. $\sqrt{2}n \times \sqrt{2}n$, when M is rotated by the angle $\pi/4$). In order to make the rotation of M around its centre, we translate the input picture so that its centre $(n/2, n/2)$ coincides with $(0,0)$. Thus the algorithms for rotation based on Definitions 2.3.1 and 2.3.2 involve n^2 and $2n^2$ pixels respectively. Yet we will use the latter in this chapter because it has an advantage over the former as mentioned earlier.

It is clear that rotation by Definition 2.3.2 does not always entirely preserve the shape of the input picture. Distortions may be present near the boundary of the picture. But when the perimeter is small compared to the area of the picture, shape distortions, if any, will be low.

Now, we will prove some properties of rotation (Definitions 2.3.2) regarding preservation of the shape of the input picture.

Proposition 2.3.1 If a hole of a binary picture has thickness more than 1, then after rotation by arbitrary angles, the hole gives rise to at least two output 0-pixels and the resulting 0-pixels are 4-connected.

Proof. The smallest hole with thickness more than 1 is shown in Fig.2.19. Let α be the angle of rotation. Now, there will be an output pixel (I, J) such that the rotated version of (I, J) by angle α will fall inside one of the four input 0-pixels in Fig.2.19. (Falling of a 2-dimensional point inside a pixel has been defined in Fig.2.16). Without loss of generality, suppose the rotated version of (I, J) falls inside the input 0-pixel A. In Fig.2.20, if O is the rotated version of (I, J) , then the rotated versions of the eight neighbours of (I, J) are $P_1, Q_1, P_2, Q_2, P_3, Q_3, P_4$ and Q_4 with some rotation (in fact, by angle α) of the structure. Since the input and output pixels have the same size, say 1, the lengths of the line segments OP_j and OQ_j are 1 and $\sqrt{2}$ respectively, $j=1, \dots, 4$. We have assumed that O falls inside the input pixel A. So, from Fig.2.19 and Fig.2.20, it is clear that (i) at least one of P_j 's will fall inside the union of the four pixels A, B, C and D and (ii) if any Q_j falls inside the union of those four input pixels, then either P_j or P_{j+1} also falls inside the same union. Now, since P_j 's are the 4-neighbours of (I, J) in the output picture, the input 0-pixels A, B, C and D give rise to output 0-pixels that are 4-connected. From (i) above, it is clear that A, B, C, D give rise to at least two 0-pixels.

Hence the Proposition 2.3.1.

But if a hole of binary picture has thickness 1, then

after rotation the hole may get disconnected. One such example is given in Fig.2.21.

Proposition 2.3.2 Disconnected components of a binary picture can not get connected after rotation by arbitrary angles, provided the thinness of the background is more than 1.

Proof. It is clear from the Proposition 2.3.1.

But if the thinness of the background is 1, then disconnected components may get connected after rotation. One such example is given in Fig.2.22.

Proposition 2.3.3 A connected binary picture with thinness more than 1 remains connected after rotation with arbitrary angles.

Proof. The weakest connectivity that a connected binary picture with thinness 2 can have is shown in Fig.2.11. For the worst case, suppose there is an output pixel (I, J) whose rotated version falls inside the pixel A in Fig.2.11. Now the present proof will be similar to that of the Proposition 2.3.1. Consider Fig.2.20. We have assumed that the point O is inside A. Now, with any rotation of the structure in Fig.2.20 (with the point O in A, of course), at least one of the eight points $P_i, Q_i (i=1, \dots, 4)$ will fall inside the union of the input 1-pixels A, B and C. In other words, at least one of the eight neighbours of the output pixel (I, J) will be a 1-pixel. Recall that (I, J) also is an output 1-pixel.

Hence the Proposition 2.3.3.

Proposition 2.3.4 If a binary picture is such that (i) its background has thinness more than 1, (ii) each of its components has thinness more than 1 and (iii) none of its components has a hole with thinness less than 2, then the topological properties of the input picture do not change after rotation by arbitrary angles.

Proof. It is clear from the Propositions 2.3.1, 2.3.2 and 2.3.3.

But preservation of topological properties does not say much about the preservation of the general shape of the input picture as mentioned earlier in Chapter 1. In fact, rotation by the Definition 2.3.2 retains not only the topological properties but all the shape properties of the input picture if the distortions due to the digital nature of a picture are ignored, provided the input picture satisfies the conditions in Proposition 2.3.4. (As mentioned earlier, the shape of a picture means, in this thesis, what remains of the picture after disregarding its position, size and orientation).

With the method of rotation of a picture discussed so far, the number of output pixels to be dealt with is $2n^2$ when the size of the input picture is $n \times n$. In order to reduce the computing time, we can rotate a picture where only the boundary pixels will be involved as we have done for propagation and

shrinking of a picture in the previous section (Section 2.2). We give such an algorithm below.

Algorithm 2.5. To rotate the boundary of a binary picture by an arbitrary angle α in the anticlockwise direction.

Suppose for the time being that the input picture is simply connected with thickness more than 1. The outer boundary pixels of the output picture will be traced in the clockwise direction as has been done in the Algorithm 2.0.

Step 1 Translate the input picture matrix $M(n \times n)$ so that its centre $(n/2, n/2)$ is at $(0, 0)$. Take the output picture matrix N of size $\sqrt{2}n \times \sqrt{2}n$ with its centre at $(0, 0)$. Scan the pixels of the matrix N row-wise (i.e. top down and left to right) and find the first output pixel (I, J) such that the rotated pixel (i, j) (defined at the start of this section) of (I, J) by angle α in the clockwise direction is a 1-pixel of the input matrix M . Set $N(I, J) = 1$. In fact, (I, J) will be an outer boundary pixel of the output picture, more specifically, the left uppermost boundary pixel. Set $DIR = 2$.

Step 2 Let (I_1, J_1) be the neighbour of (I, J) in the direction DIR . If the rotated pixel of (I_1, J_1) by angle α in the clockwise direction is a 1-pixel of the input matrix M , then set $I_0 = I_1$, $J_0 = J_1$, $N(I_1, J_1) = 1$, $DIR_0 = DIR$, $DIR = DIR + 6 \pmod{8}$, $I = I_1$, $J = J_1$ and go to Step 3. Otherwise, set $DIR = DIR + 1 \pmod{8}$ and repeat Step 2. (Note that at least one neighbour (I_1, J_1) of (I, J)

will be a 1-pixel of N , because (I, J) can not be an isolated 1-pixel of the output picture since the input picture is connected with thickness more than 1 and rotation in that case preserves the connectivity property from the Proposition 2.3.3).

Step 3 Let (I_1, J_1) be the neighbour of (I, J) in the direction DIR . If the rotated pixel of (I_1, J_1) by angle α in the clockwise direction is a 1-pixel of the input matrix M , then go to Step 4. Otherwise, set $DIR = DIR + 1 \pmod{8}$ and repeat Step 3. (Note again that at least one neighbour of (I, J) will be a 1-pixel of N).

Step 4 If $I_1 = I_0$, $J_1 = J_0$ and $DIR = DIR_0$, then stop. Otherwise, set $N(I_1, J_1) = 1$, $I = I_1$, $J = J_1$, $DIR = DIR + 6 \pmod{8}$ and go to Step 3.

The above algorithm is sure to terminate since I_0, J_0 and DIR_0 have to repeat sometime in Step 4. It is clear that the output picture $\{(I, J) : N(I, J) = 1\}$ is the rotated boundary of the input picture (Fig. 2.18).

So far we have assumed that the input picture is simply connected, i.e., without any holes. Now if there are holes in the input picture, then the above algorithm becomes costly in terms of computing time as was the case with the Algorithm 2.3 where propagation of the boundary of a binary picture was done. That is why we use the above algorithm (Algorithm 2.5) for rotation when the input picture is known to ^{be} simply connected with

thinness more than 1. The interior pixels of the output picture are obtained by the already mentioned contour filling algorithm by connectivity.

The above algorithm for rotation of a binary picture will be used for shape matching of 2-dimensional regions in Chapter 5.

2.4 Conclusion

We have proposed in this chapter algorithms for propagation, shrinking and rotation and have proved some desirable properties of these algorithms regarding shape preservation of binary pictures. The basic ideas of some of the above algorithms were given elsewhere (Rosenfeld and Johnston, 1970), though their shape preservation properties were not proved in detail. The above algorithms for propagation, shrinking and rotation are used for 2-dimensional shape matching in Chapter 5. Some results of propagation and shrinking algorithms are given in Figs. 2.23 and 2.24.

It can be seen that the extension of the above algorithms for propagation, shrinking and rotation of 2-dimensional regions to 3-dimensional solids is straightforward. For propagation and shrinking in 3 dimensions, voxels (instead of pixels) of different sizes are considered and for rotation in 3 dimensions, two (instead of one) angles are involved.


```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 . . . 1 1 1 . . . .
1 . 1 . 1 1 1 . . . .
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1

```

Fig.2.1(a)

- (i) The order of connectivity of the above binary picture is 2. That is, the picture has only one hole.
- (ii) The thinness of the above connected binary picture is 1.
- (iii) The thinness of the hole above is 1.
- (iv) The thinness of the background of the above picture is 2.


```

1 1 1 1 1 1 1 1 1 1 1 1 1
1                                     1
1                                     1 1 1 1 1
    1                               1
    1                               1
1                                     1 1 1 1 1
1                                     1
1 1 1 1 1 1 1 1 1 1 1 1 1

```

Fig.2.1(b) The outer boundary of Fig.2.1(a)

```

. 1 1 1 .
1       1
1     1   1
. 1   1 .

```

Fig.2.1(c) The inner boundary of Fig.2.1(a)

1	2	3
6	X	2
5	4	3

Fig.2.2 8 directional codes.
X is the reference pixel.


```

1 1 1      1 1 1
1 1 1      1 1 1
1 1 1 1 1 1 1 1 1      1 1
1 1 1 1 1 1 1 1 1      1 1
1 1 1      1 1 1
1 1 1      1 1 1

```

(a) A connected binary picture is
disconnected after shrinking

```

1 1 1      1 1 1 1 1
1 1 1      1 1 1 1 1
1 1 1      1 1 1 1 1      1 1 1
1 1 1      1 1 1      1 1
1 1 1      1 1 1
1 1 1      1 1 1

```

(b) A disconnected binary picture
becomes connected after shrinking

Fig.2.3 In both the cases above the factor
of shrinking $m=3$ and $p=0.5$.

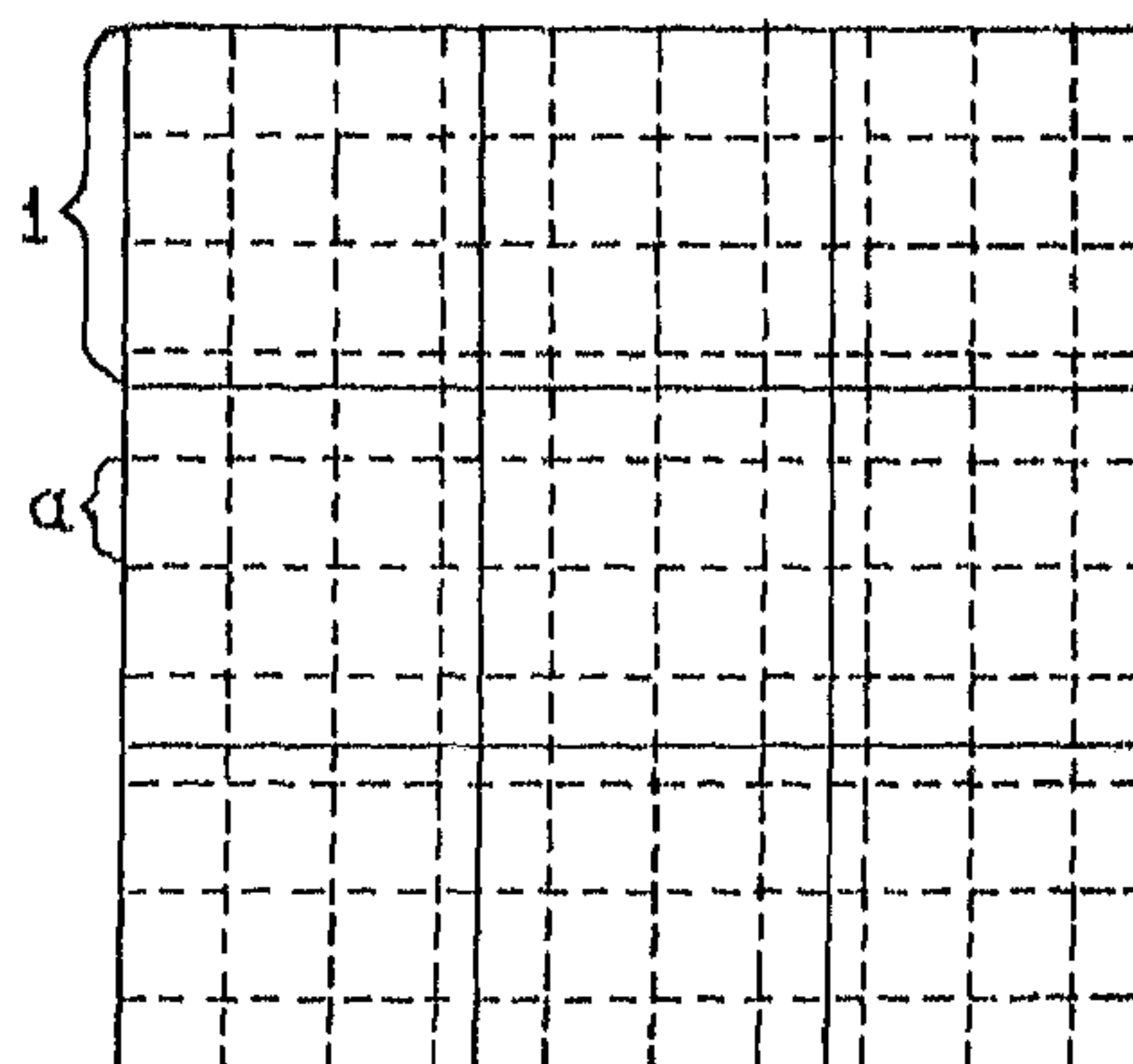


Fig.2.4 Pixels with unbroken lines and pixels with broken lines indicate the input and output pixels respectively. Factor of propagation here is $10/3$.

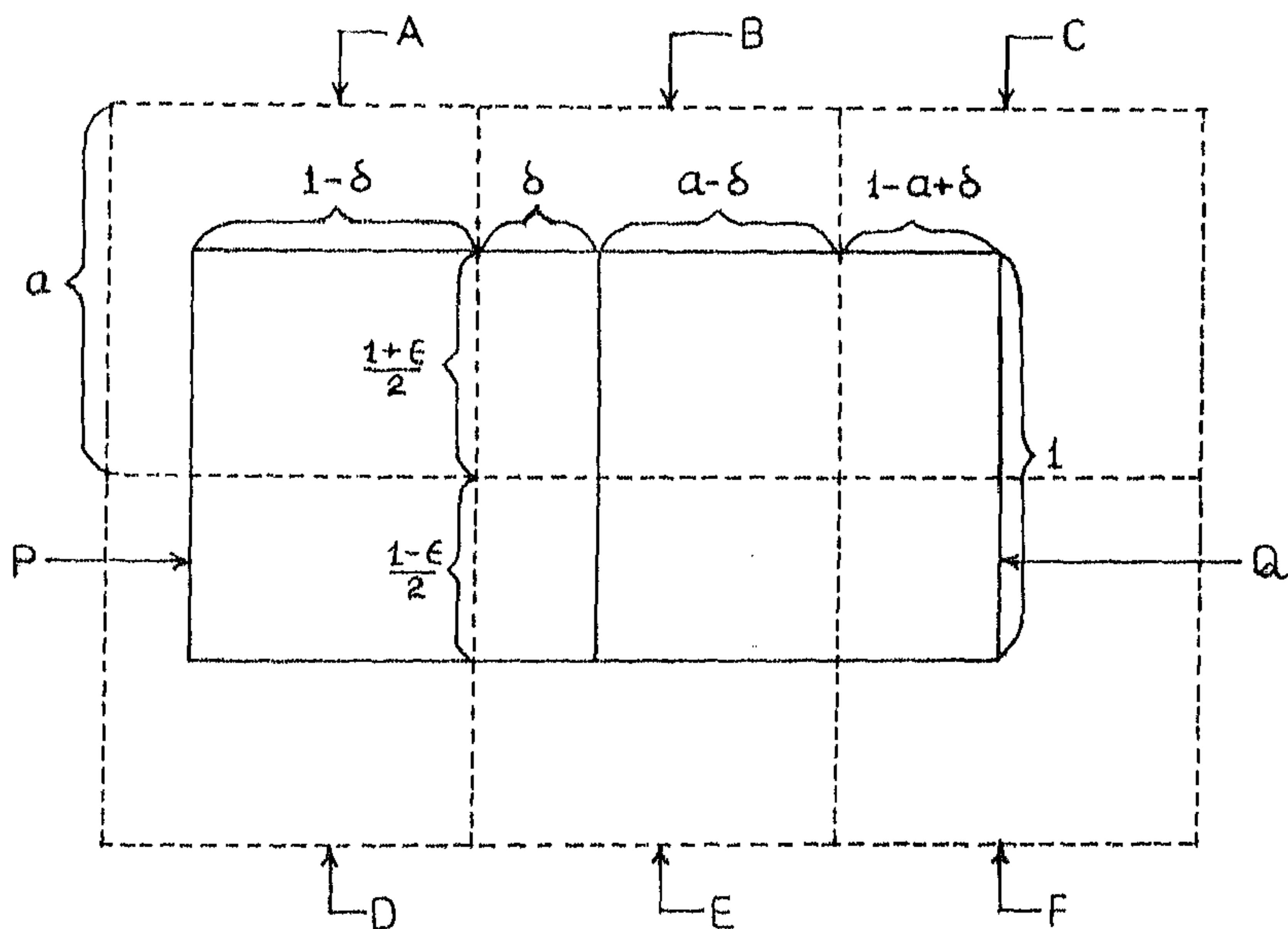


Fig.2.5 P and Q are input 0-pixels while A, B, C, D, E and F are output pixels.

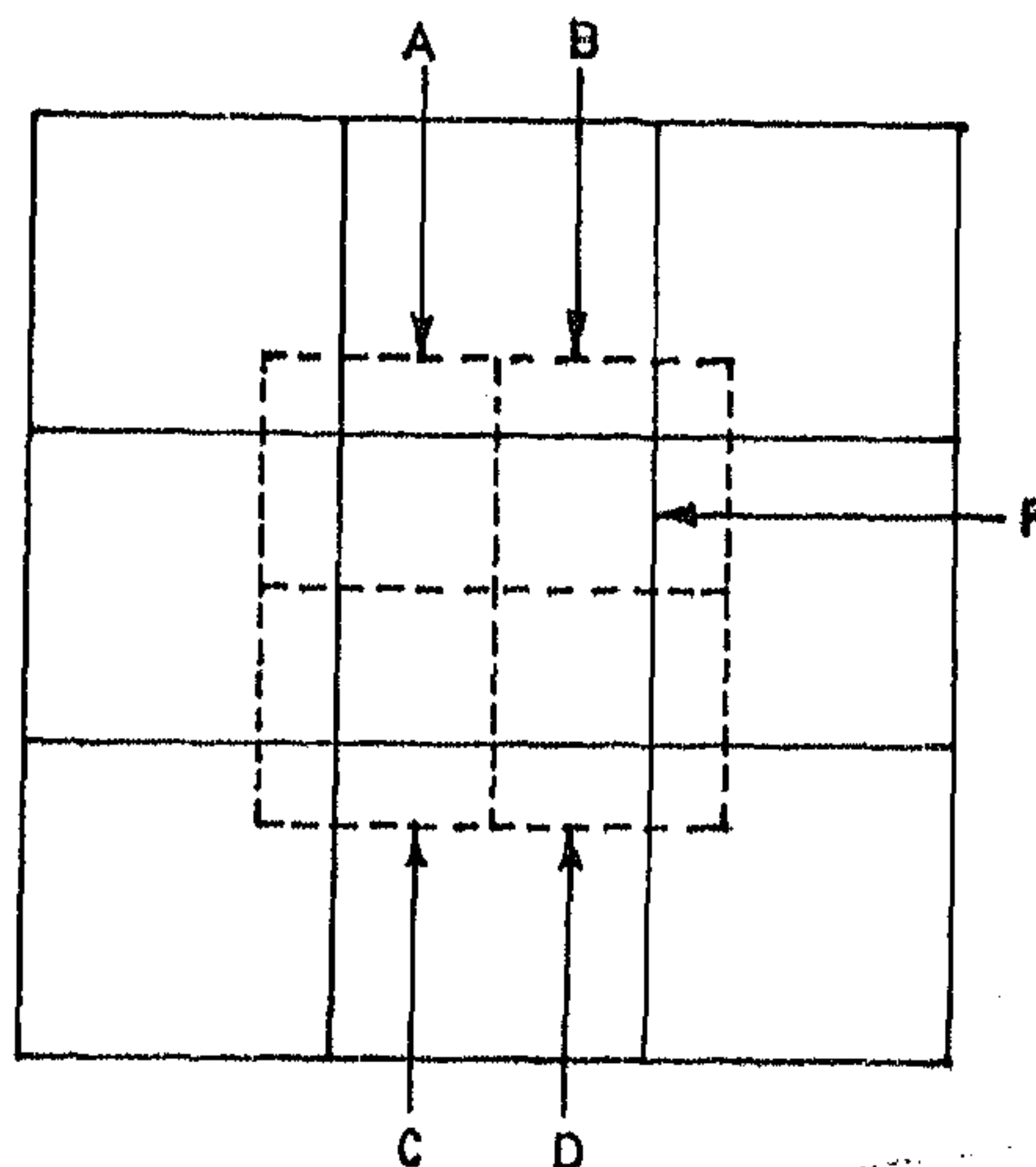


Fig.2.6 There are 9 input pixels (indicated by unbroken lines) among which P is the only 0-pixel. 8 others are 1-pixels. A, B, C and D are output pixels (indicated by broken lines). The factor of propagation is $4/3$ which is not very high. Note that the output pixels A, B, C and D are 1-pixels. So, in the output there is no hole, though the input has a hole containing only one 0-pixel P.

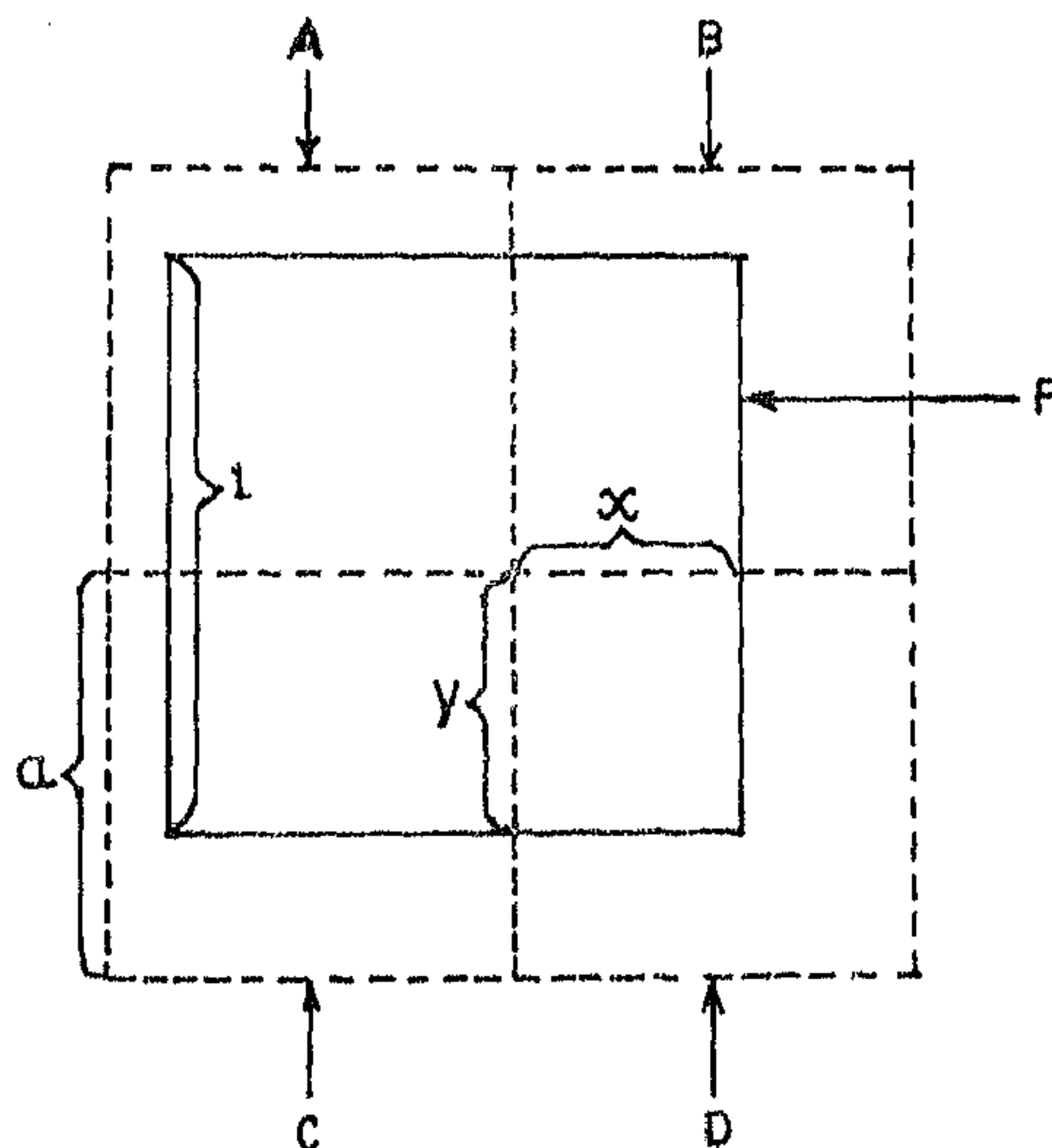


Fig. 2.7 P is the only input 0-pixel (indicated by unbroken lines) while A, B, C and D are output pixels (indicated by broken lines). a is strictly less than $1/\sqrt{2}$, i.e., the factor of propagation is strictly more than $\sqrt{2}$.

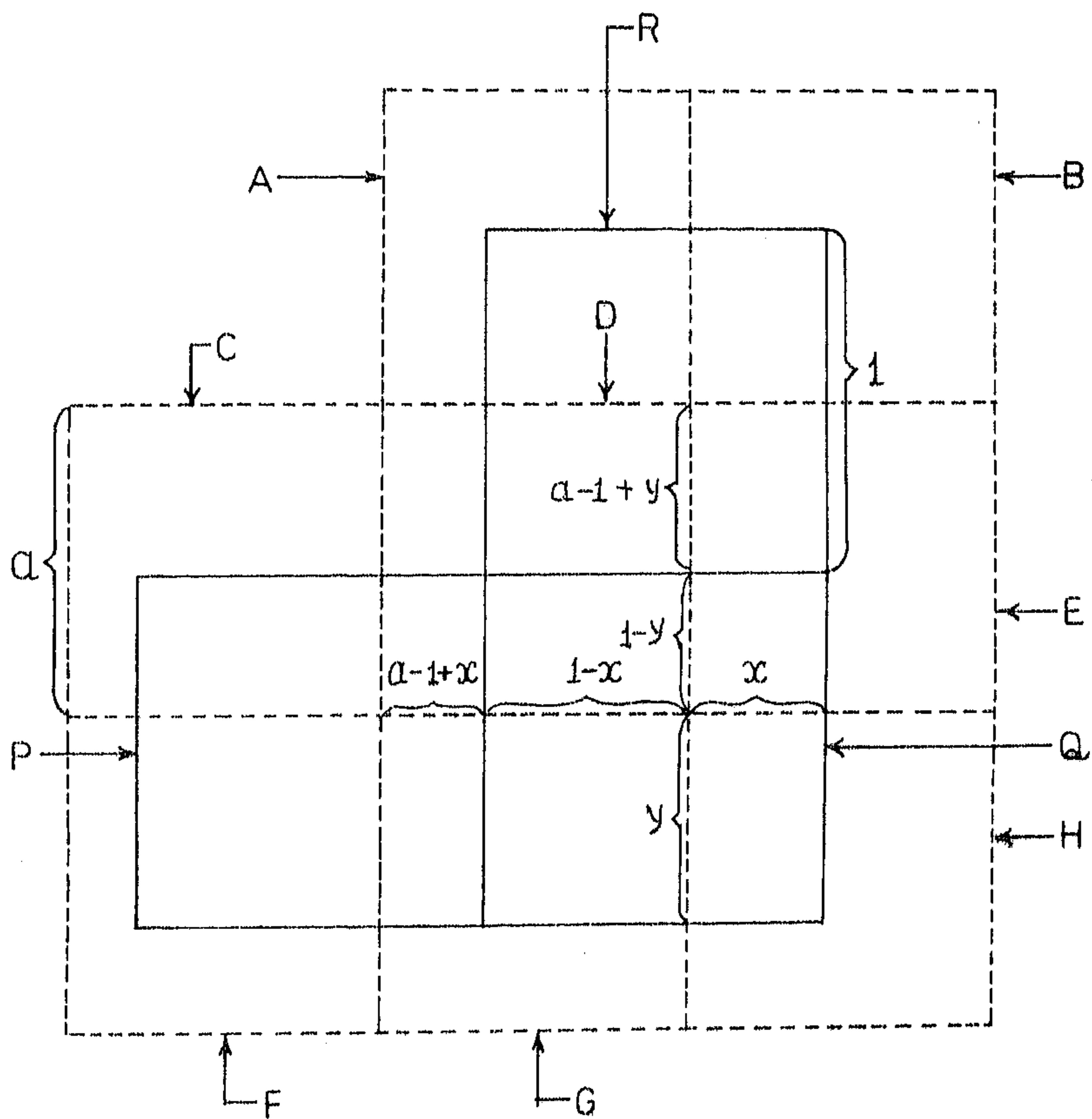


Fig.2.8 P, Q and R are input 0-pixels while
 A, B, C, D, E, F, G and H are output pixels.

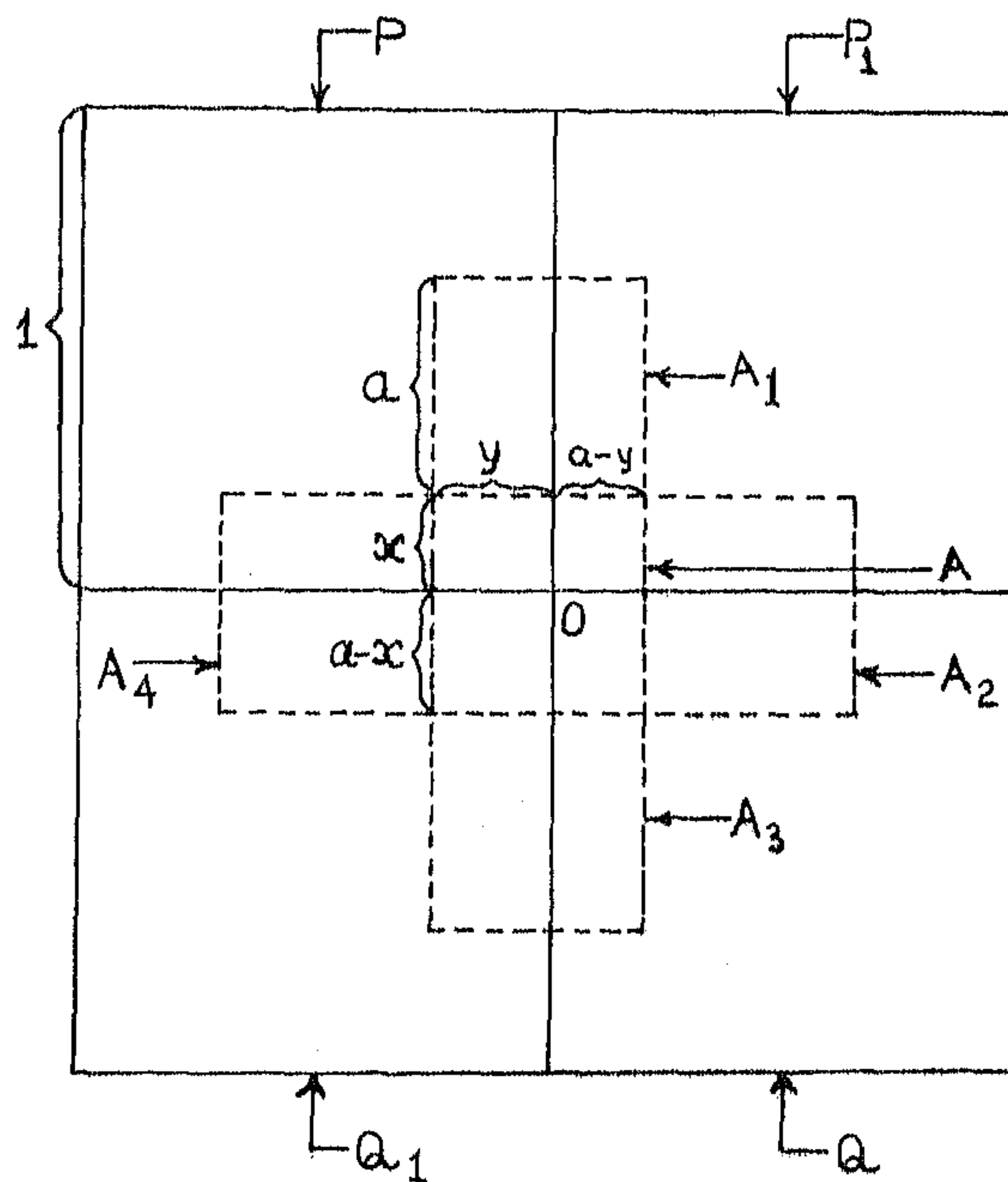


Fig.2.9 P and Q are input 1-pixels, P_1 and Q_1 are input 0-pixels and A, A_1, A_2, A_3 and A_4 are output pixels.

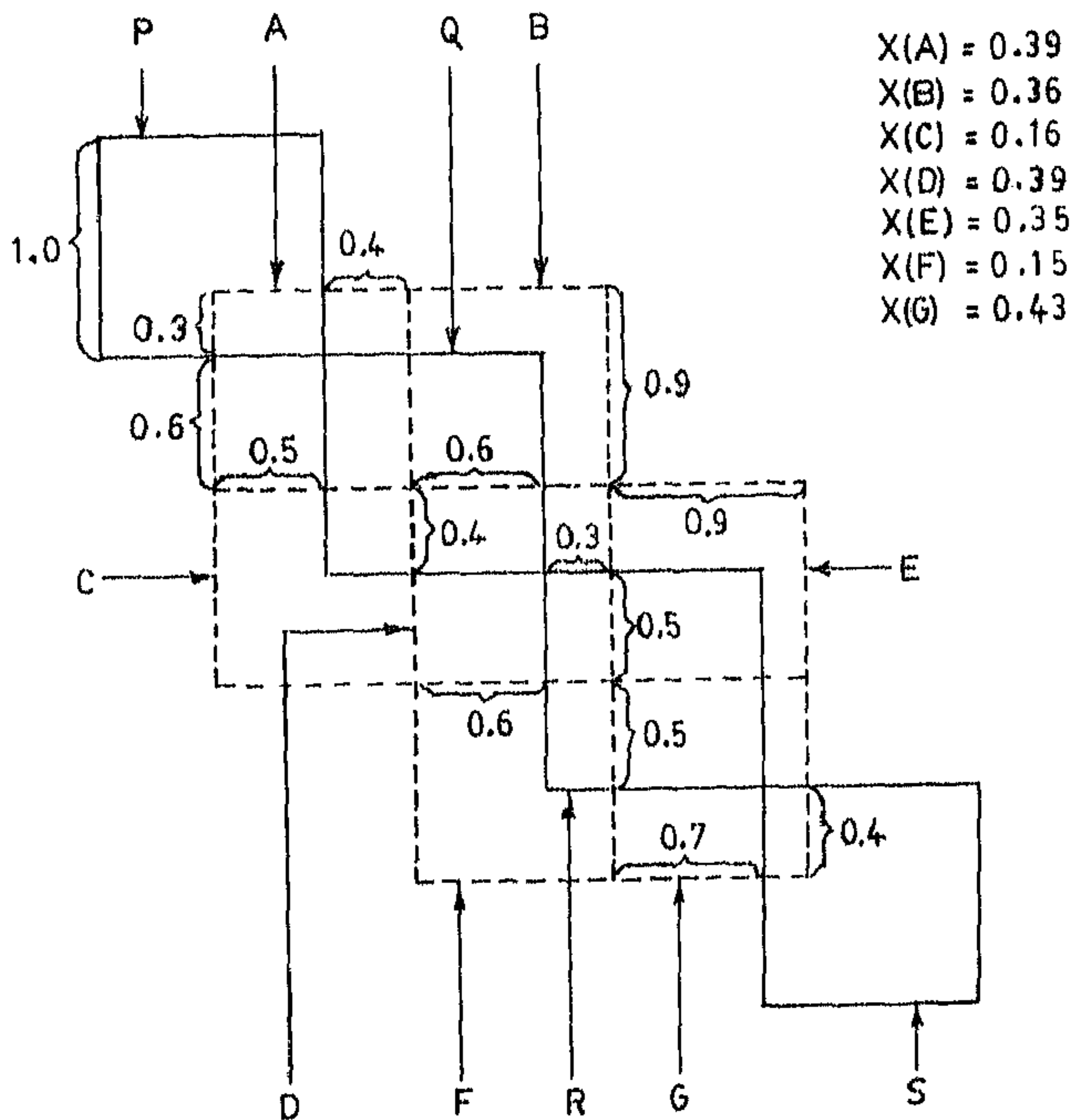


Fig. 2.10 P, Q, R and S are input 1-pixels (indicated by unbroken lines) while A, B, C, D, E, F and G are output pixels (indicated by broken lines). Here $a=0.9$. So, the factor of propagation is $10/9$ and $a^2/2 = 0.405$. From the X values above, it is clear that A, B, C, D, E, and F are 0-pixels. Thus, a connected picture with thinness 1 (corner connected) can be disconnected after propagation.

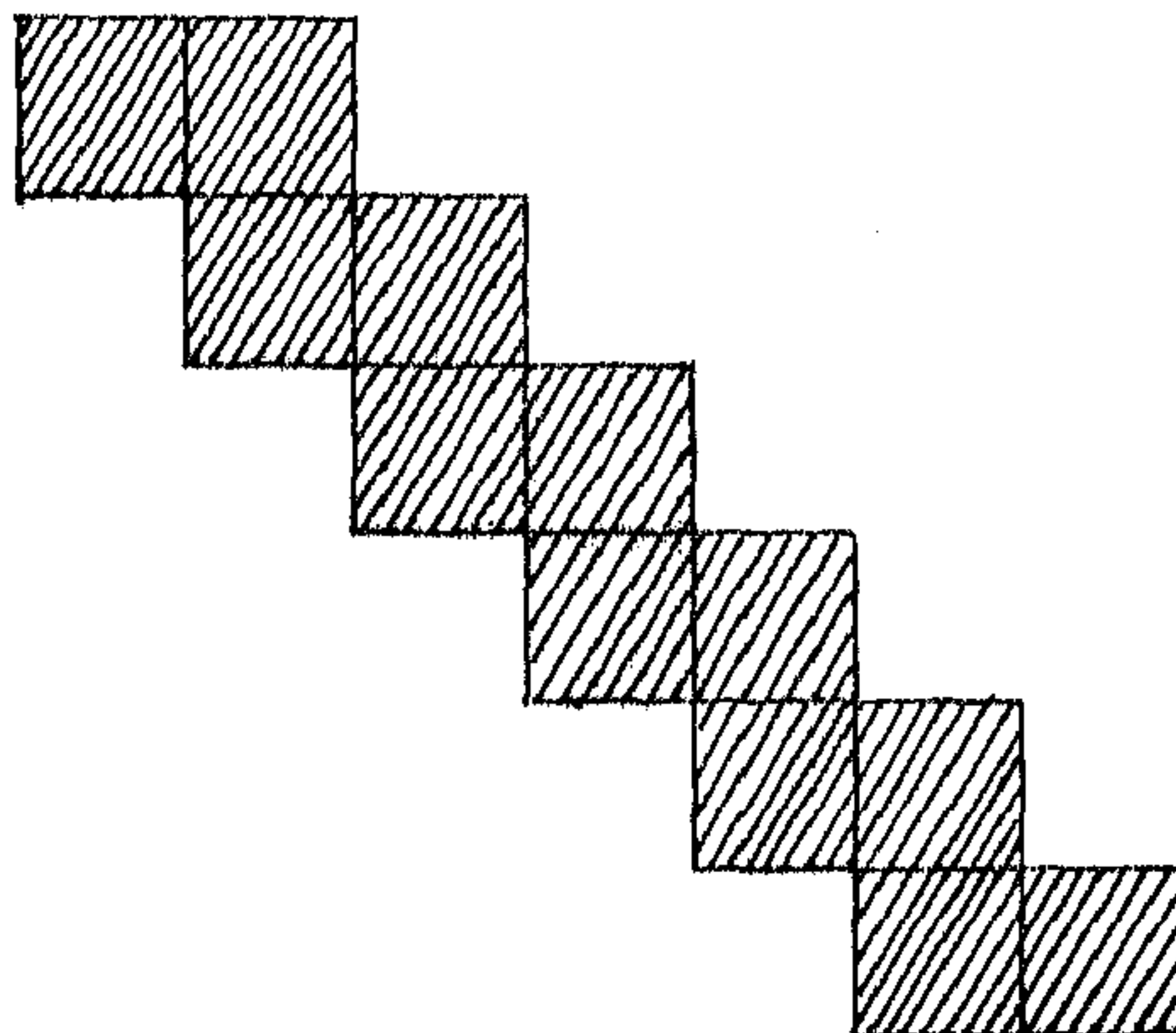


Fig. 2.11 The weakest connectivity that a connected binary picture with thickness 2 can have.

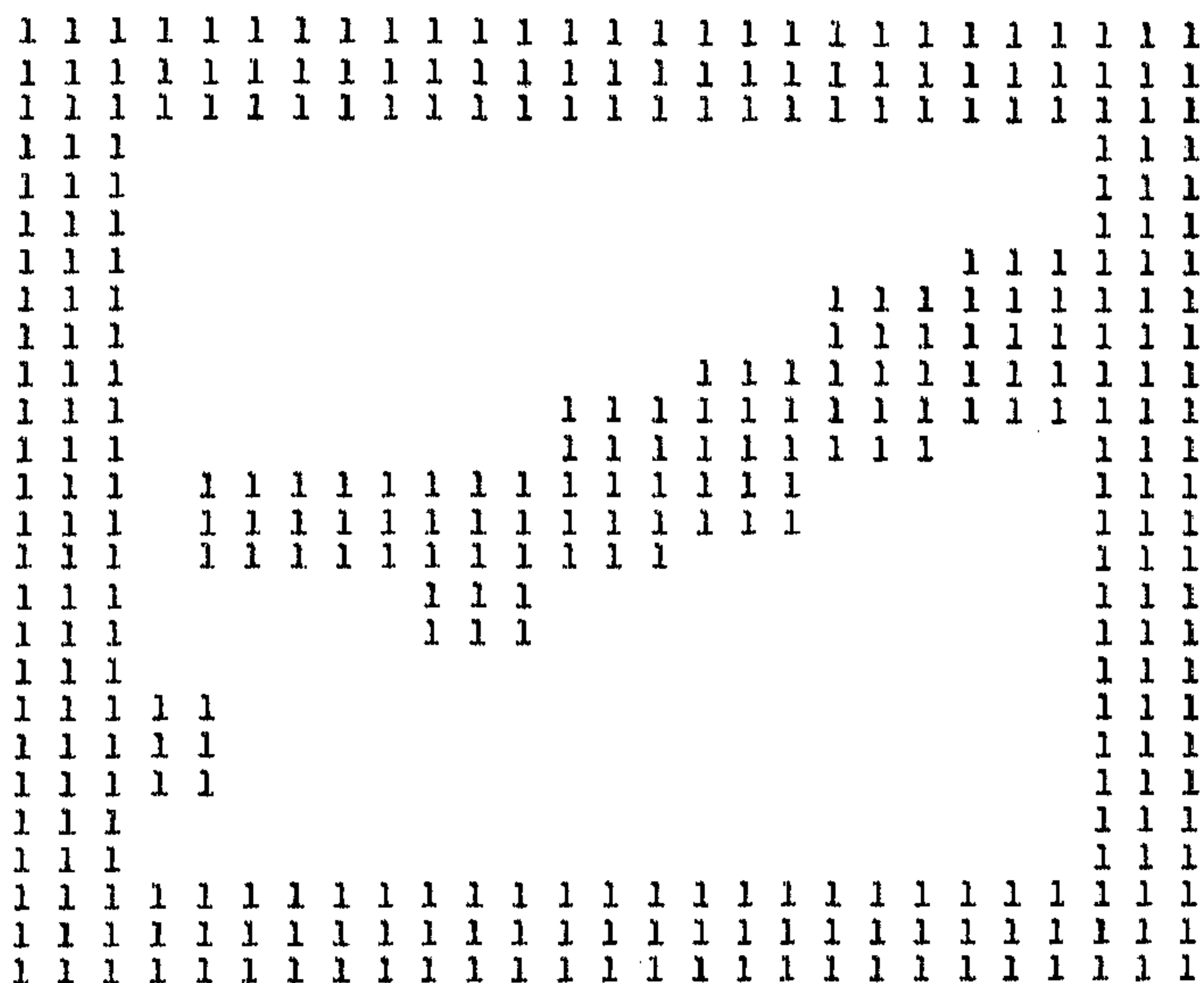
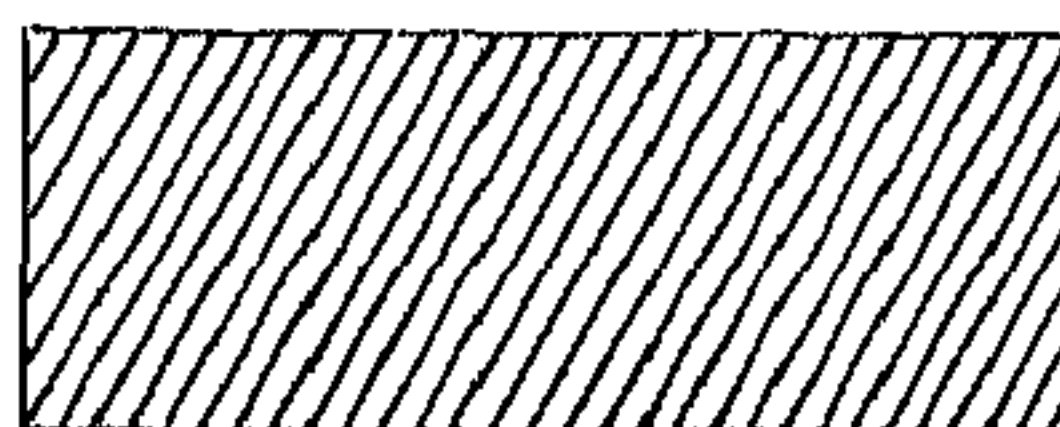
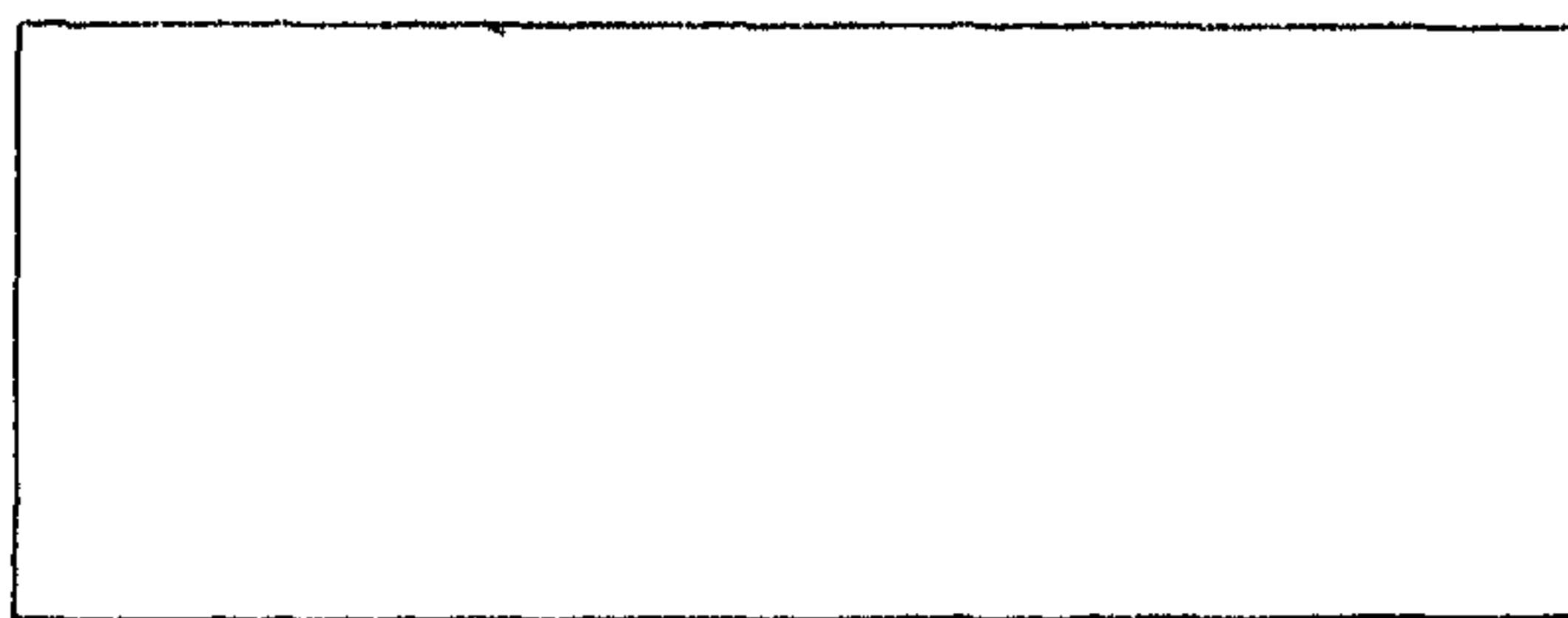


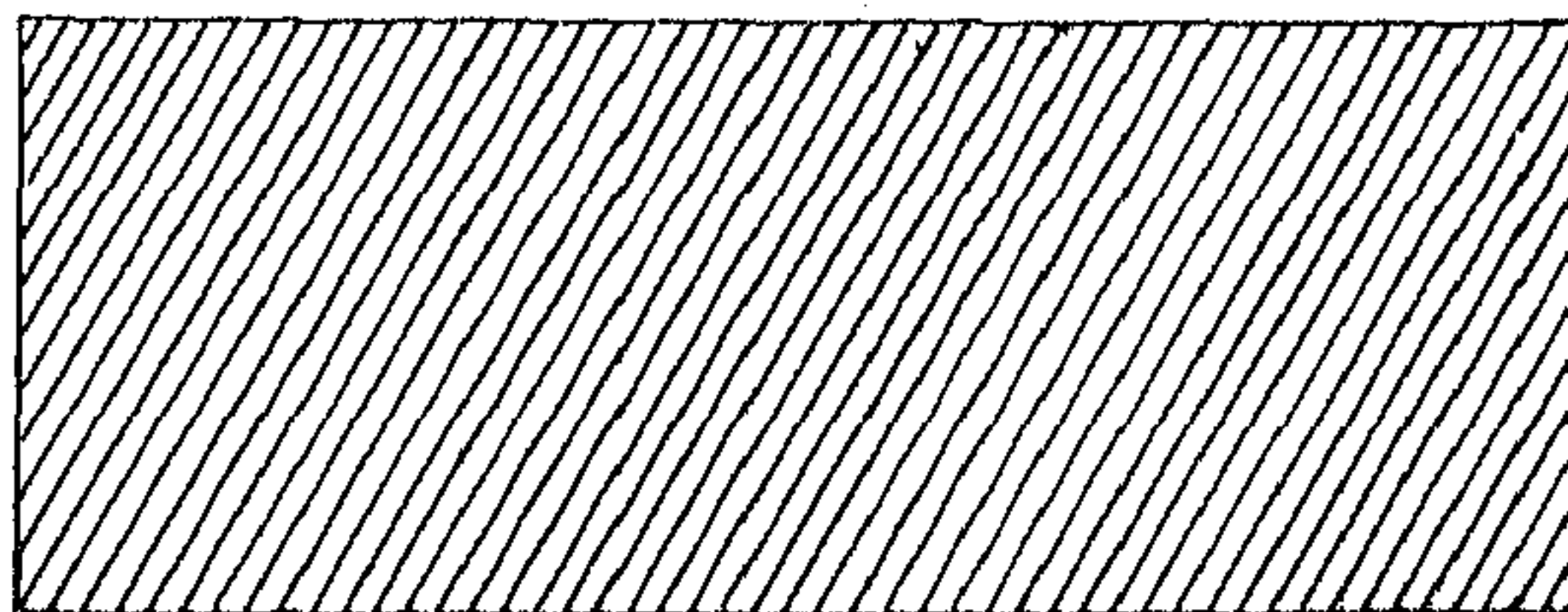
Fig. 2.12(b)



(a)



(b)



(c)

Fig. 2.13 The picture in (b) is the propagated boundary (the factor of propagation is 2) of the picture in (a). The picture in (c) can be obtained from the picture in (b) by contour filling algorithms.

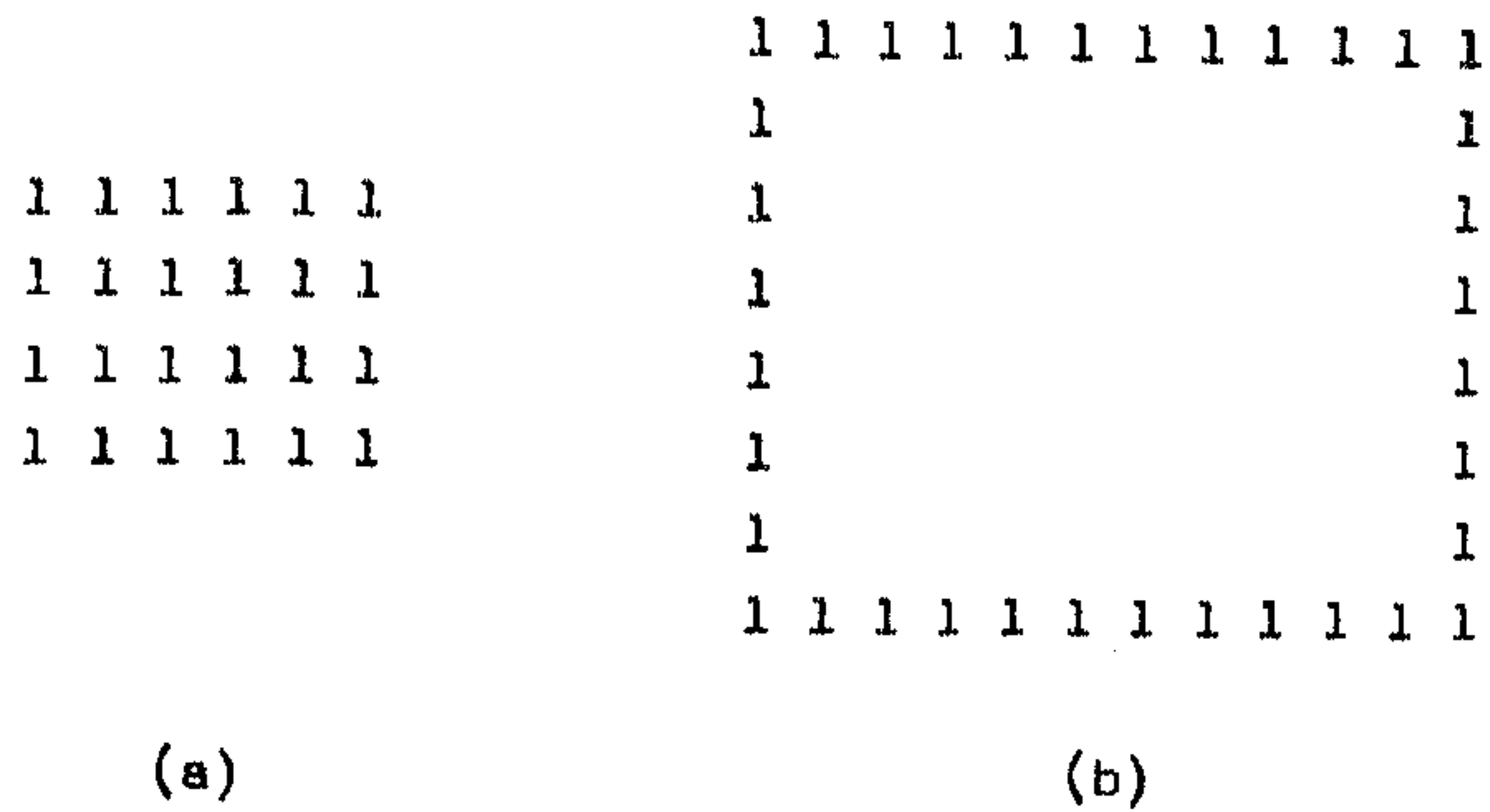


Fig.2.14 The picture in (b) is the propagated boundary of the input picture in (a). The factor of propagation here is 2.

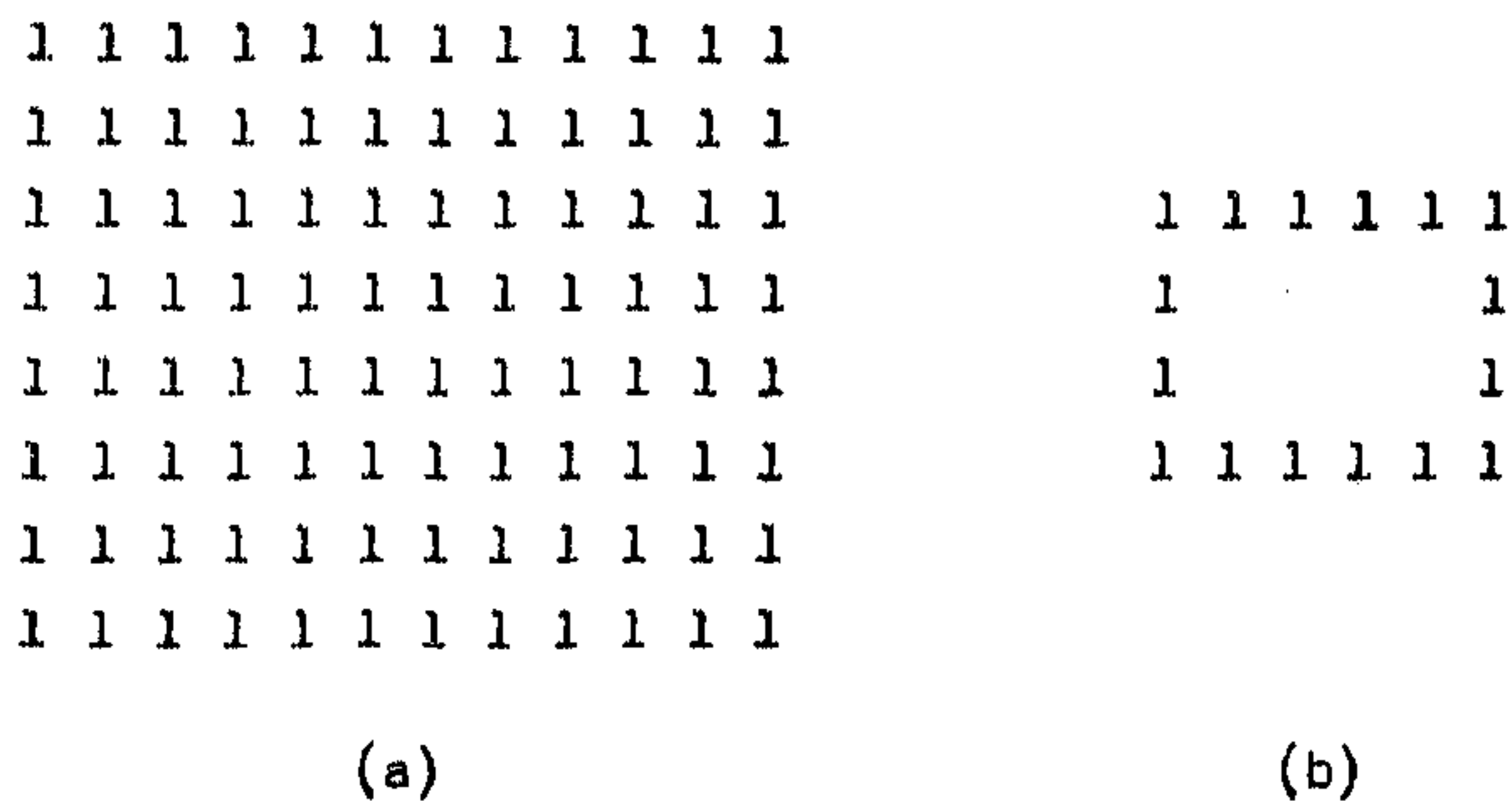


Fig.2.15 The picture in (b) is the shrunk (reduced) boundary of the input picture in (a). The factor of shrinking here is 2.

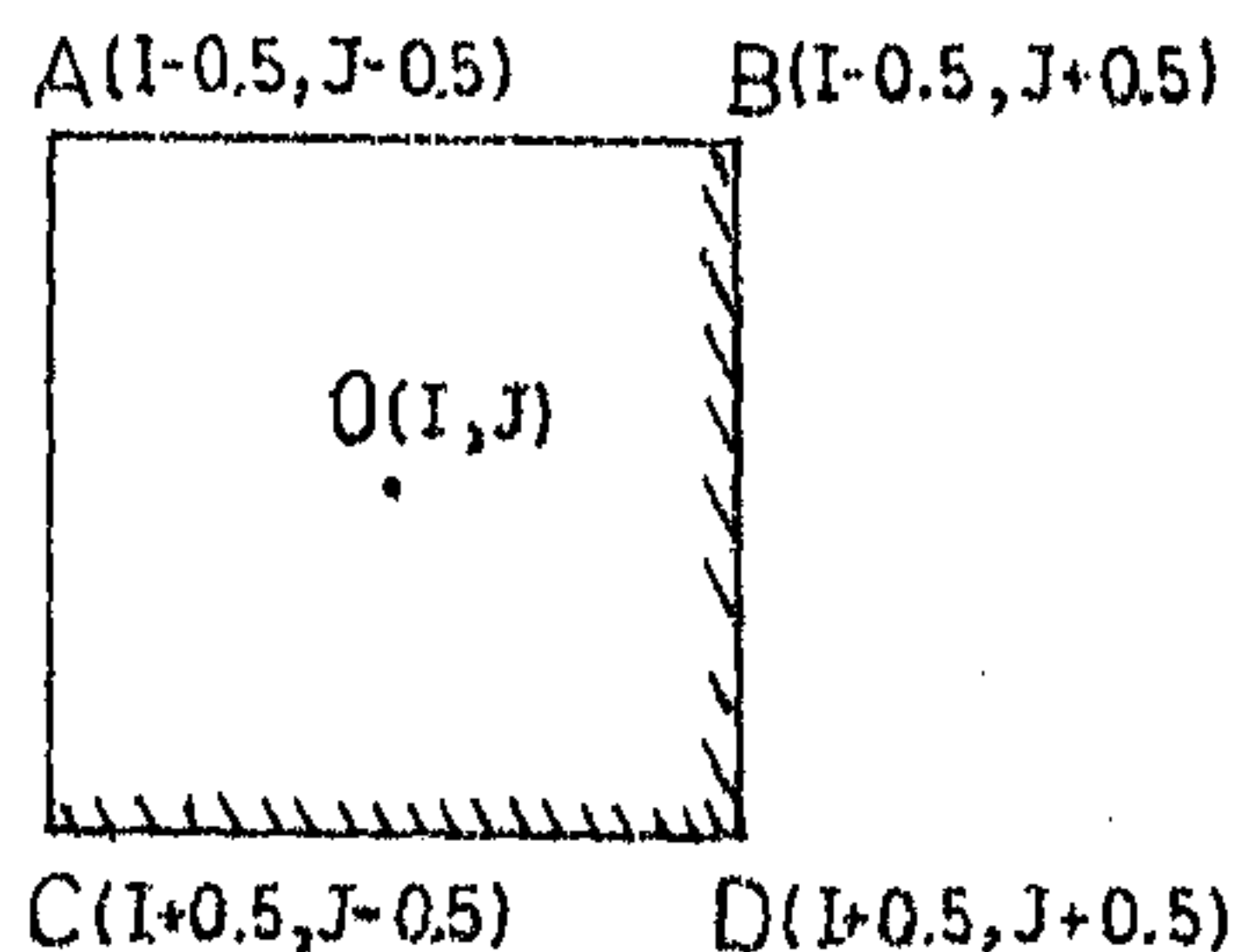


Fig. 2.16 A 2-dimensional point (x, y) is approximated as the pixel (I, J) if either (x, y) falls in the interior of the square $ABCD$ (i.e., the pixel (I, J)) or (x, y) falls on the open line segment BD or CD or (x, y) coincides with the point D .

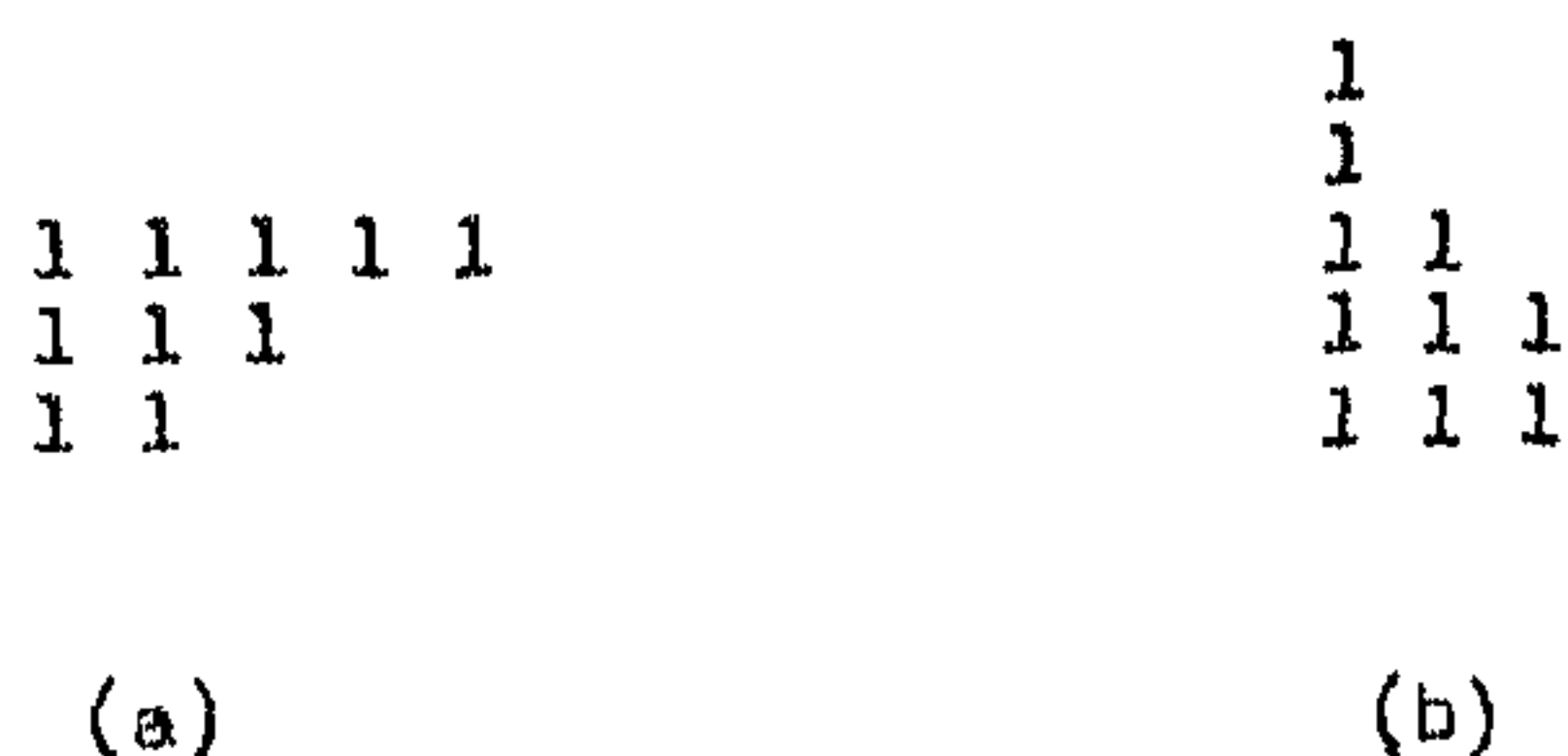


Fig. 2.17 The picture in (b) is the rotation of the picture in (a) by 45° angle in the anti-clockwise direction.

(5,1) (5,2) (5,3) (5,4) (5,5)
 (6,1) (6,2) (6,3) (6,4) (6,5)
 (7,1) (7,2) (7,3) (7,4) (7,5)
 (8,1) (8,2) (8,3) (8,4) (8,5)
 (9,1) (9,2) (9,3) (9,4) (9,5)

(a)

(0,7)
 (1,6) (1,7) (1,8)
 (2,5) (2,6) (2,8) (2,9)
 (3,4) (3,6) (3,7) (3,8) (3,10)
 (4,5) (4,6) (4,7) (4,8) (4,9)
 (5,6) (5,8)
 (6,7)

(b)

Fig.2.18 The pixels with matrix co-ordinates indicate 1-pixels while the other pixels are 0-pixels. The picture in (a) is a digital square without any holes. The picture in (b) is the rotated version (by Definition 2.3.1) of the picture in (a) by 45° angle in the anticlockwise direction. Note that the picture in (b) has some holes though the input picture in (a) does not have any holes.


```

          (0,7)
        (1,6) (1,7) (1,8)
      (2,5) (2,6) (2,7) (2,8) (2,9)
    (3,4) (3,5) (3,6) (3,7) (3,8) (3,9) (3,10)
      (4,5) (4,6) (4,7) (4,8) (4,9)
        (5,6) (5,7) (5,8)
          (6,7)

```

Fig. 2.18(c) The picture above is the rotated version (by Definition 2.3.2) of the picture in Fig.2.18(a) by 45° angle in the anti-clockwise direction. Unlike the picture in Fig.2.18(b), the above picture does not have any holes. In fact, the above picture is a digital square without any holes as the input picture in Fig.2.18(a).

```

          (0,7)
        (1,6)      (1,8)
      (2,5)          (2,9)
    (3,4)            (3,10)
      (4,5)          (4,9)
        (5,6)      (5,8)
          (6,7)

```

Fig. 2.18(d) The above picture is the rotated boundary (by Algorithm 2.5) of the picture in Fig.2.18(a) by 45° angle in the anticlockwise direction. The picture in Fig.2.18(c) can be obtained from the above boundary by any contour filling algorithm.

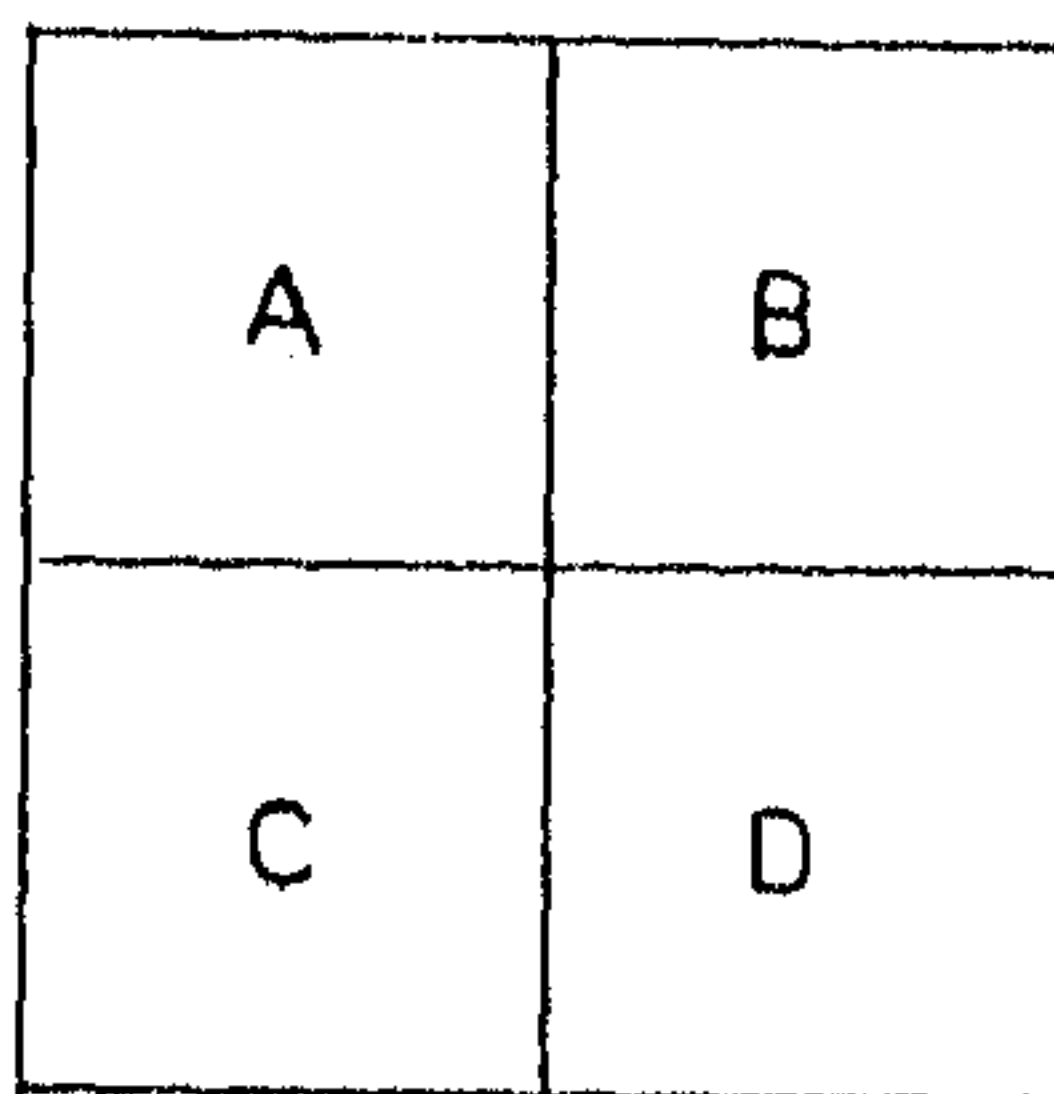


Fig. 2.19 A, B, C and D are 0-pixels and form the smallest hole with thinness more than 1.

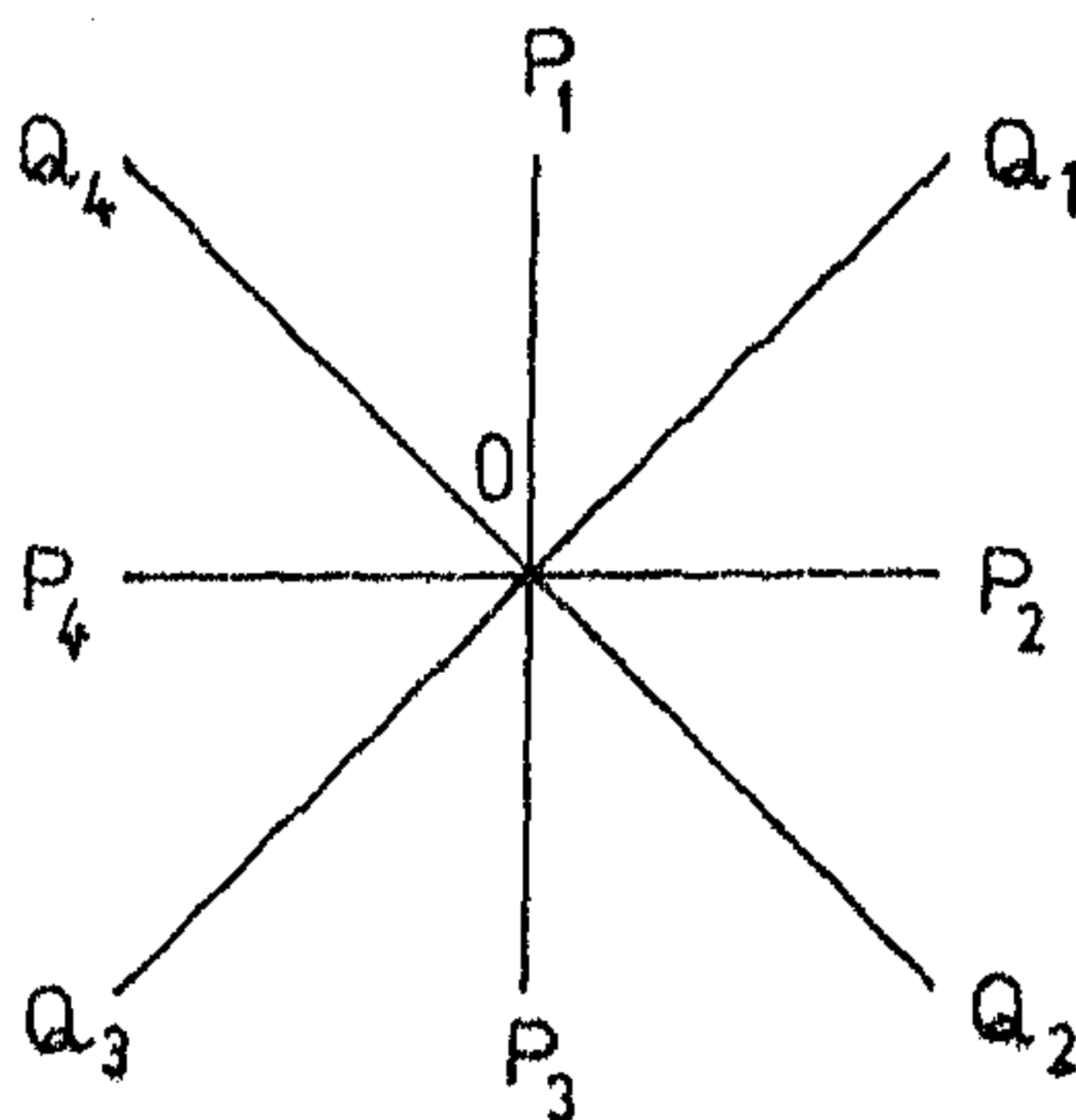


Fig. 2.20 If 0 is the rotated version of some pixel (I, J) , then P_j and Q_j ($j=1, \dots, 4$) are the rotated versions of the eight neighbours of the pixel (I, J) by the same angle. Note that 0, P_j , Q_j do not necessarily have integer co-ordinates and the lines P_1P_3 and P_2P_4 are not necessarily horizontal or vertical.


```

(5,1) (5,2) (5,3) (5,4) (5,5)
(6,1) (6,2)      (6,4) (6,5)
(7,1)              (7,5)
(8,1) (8,2)      (8,4) (8,5)
(9,1) (9,2) (9,3) (9,4) (9,5)

```

(a)

```

              (0,7)
            (1,6) (1,7) (1,8)
          (2,5)      (2,7)      (2,9)
(3,4) (3,5) (3,6)      (3,8) (3,9) (3,10)
          (4,5) (4,6) (4,7)      (4,9)
              (5,6) (5,7) (5,8)
                (6,7)

```

(b)

Fig. 2.21 The picture in (a) has only one hole (with thickness 1). The picture in (b) is the rotated version (by Definition 2.3.2) of the picture in (a) by 45° angle in the anticlockwise direction. Note that the rotated picture in (b) has four holes though the input picture in (a) has only one. Thus, a hole with thickness 1 may get disconnected.

(5,1) (5,2) (5,3) (5,4) (5,5)
 (6,1) (6,2) (6,3) (6,4) (6,5)

(8,1) (8,2) (8,3) (8,4) (8,5)
 (9,1) (9,2) (9,3) (9,4) (9,5)

(a)

(0,7)
 (1,6) (1,7) (1,8)
 (2,5) (2,6) (2,7) (2,9)
 (3,4) (3,5) (3,6) (3,8) (3,9) (3,10)
 (4,5) (4,7) (4,8) (4,9)
 (5,6) (5,7) (5,8)
 (6,7)

(b)

Fig. 2.22 The picture in (a) has two disconnected components and the background has thickness 1. The picture in (b) is the rotated version (by Definition 2.3.2) of the picture in (a) by 45° angle in the anticlockwise direction. Note that the picture in (b) has only one component. That is, the two components in (a) has become connected after rotation.


```

      1 1
      1 1
      1 2 1
     1 1 1 1 1
     1 1 1 1 1 1 1
     1 1 1 1 1 1 1 1 1 1
     1 1 1 1 1 1 1 1 1 1
       1 1 1 1 1 1 1 1 1
         1 1           1 1 1
           1 1
          1 1 1           1 1 1
          1 1 1 1 1 1 1 1 1 1
          1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1

```

Fig.2.23(a) Two propagated versions (by Algorithm 2.1) of the above picture are shown in Figs.2.23(b) and (c) where the factors of propagation are 1.6 and 2.3 respectively. The area of the above picture is 120. So, the areas of the pictures in Figs.2.23(b) and (c) are theoretically supposed to be 307 and 635 respectively. But the areas of these pictures are in fact 302 and 611 respectively. The errors (about 2 and 4 per cent respectively) are due to the digital nature of the pictures and to the fact that the perimeter is large compared to the area of the input picture above.

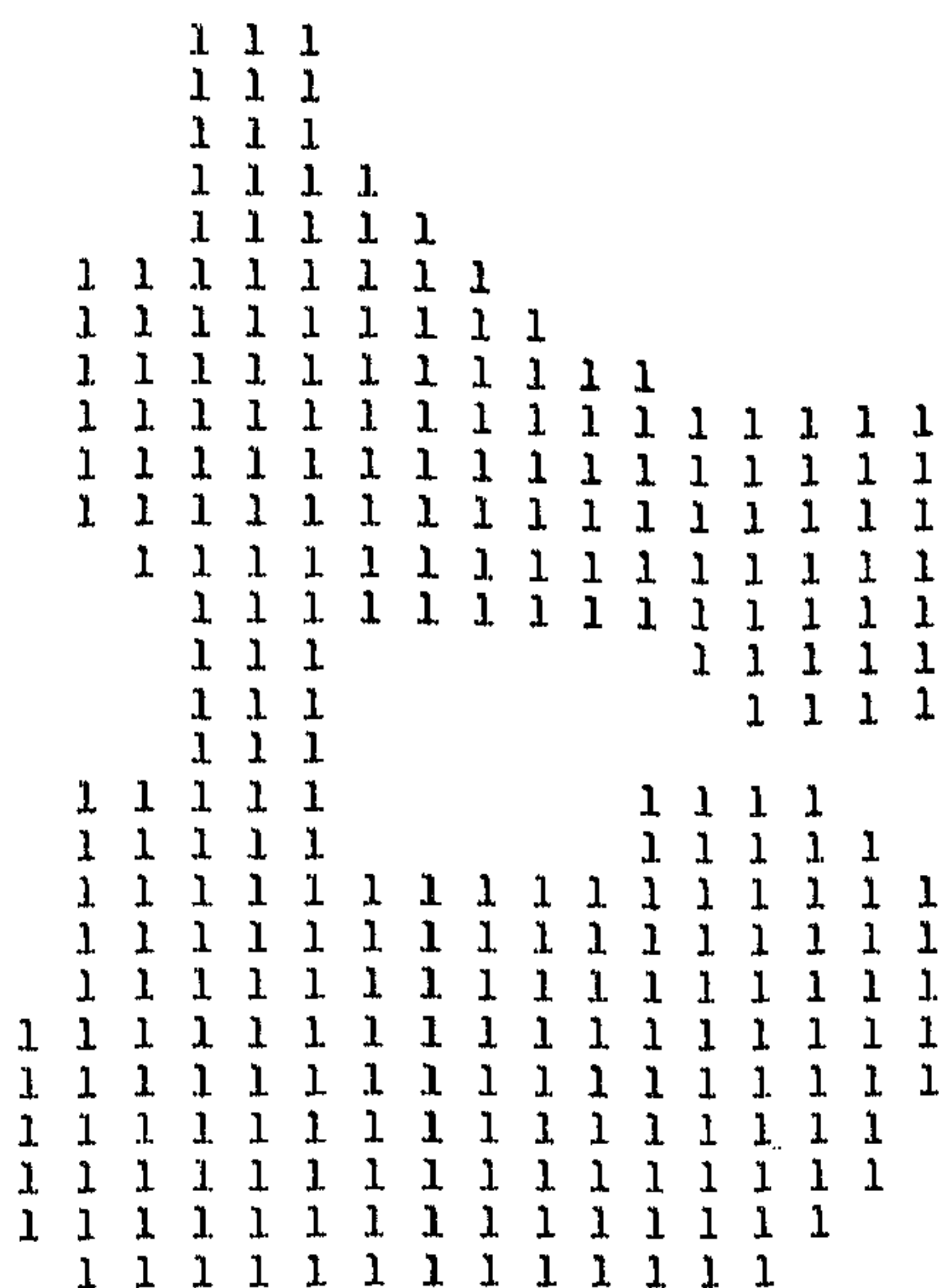


Fig.2.23(b) Propagated version (by Algorithm 2.1)
of the input picture in Fig.2.23(a)
with 1.6 as the factor of propagation.

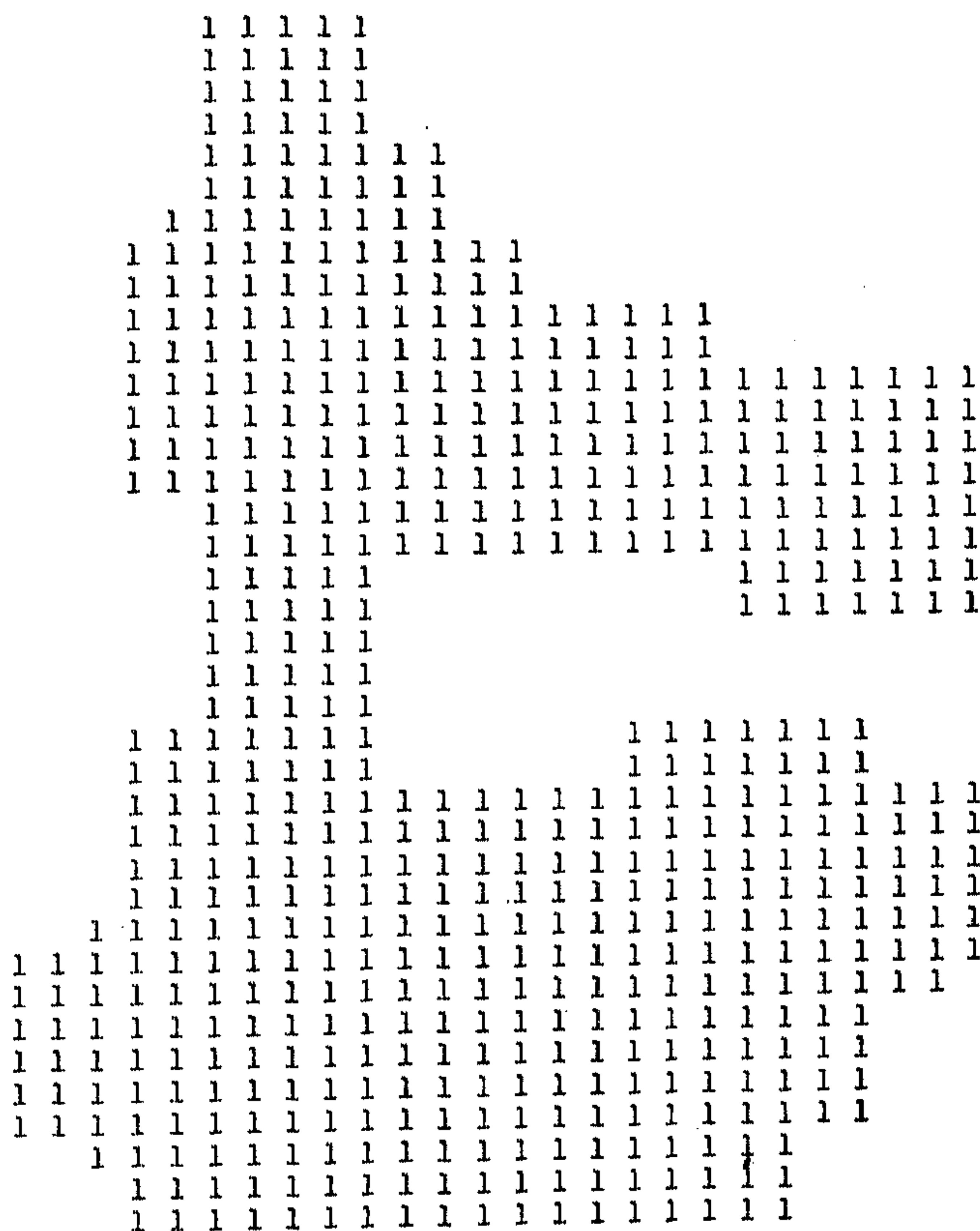


Fig.2.23(c) Propagated version (by Algorithm 2.1) of the input picture in Fig.2.23(a) with 2.3 as the factor of propagation.


```

      1 1
      1 1 1
      1 1 1
    1 1 1 1 1
    1 1 1 1 1 1 1
      1 1 1 1 1 1
      1 1      1 1 1
      1 1
    1 1 1      1 1 1
    1 1 1 1 1 1 1
    1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1
  1 1 1 1 1 1 1
    1 1 1 1 1 1

```

Fig. 2.24(a) The above picture is a shrunk (reduced) version (by Algorithm 2.2) of the picture in Fig.2.23(c) which is a propagated version of the picture in Fig.2.23(a). Both the factors of propagation and shrinking are 2.3. Note that the above picture is very close to the original picture in Fig.2.23(a), though they are not exactly the same. The error is due to the digital nature of the pictures and the algorithms. The area of the above picture is 112, though it is theoretically supposed to be 116.


```

      1
    1 1
  1 1 1
1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
  1 1      1 1
    1 1      1 1
      1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
  1 1 1 1 1

```

Fig. 2.24(b) The above picture is a shrunk (reduced) version (by Algorithm 2.2) of the picture in Fig.2.23(c). The factor of shrinking is $46/15$. Note that the above picture is topologically different from the input picture in Fig.2.23(c). The latter is a simply connected picture without any holes, but the above picture, though connected, has a hole. Thus, the shrinking algorithm does not have shape preservation properties while the propagation algorithm has these properties (discussed in Section 2.2). The loss of topological properties in the above picture is due to the fact that the factor of shrinking is too large compared to the thinness of the background of the picture in Fig.2.23(c). The area of the above picture is 67, though it is theoretically supposed to be 65.

CHAPTER 3

Thinning Algorithms for Elongated Objects

3.1 Introduction

The transformation of a bilevel picture into a line drawing is important in computer aided design, automated cartography, facsimile transmission and pattern recognition (Seuffert, 1977; Pford and Ramachandran, 1978). Such pictures were usually meant in the first place to be line drawings like printed or drawn lines, curves, letters and other graphical symbols. The resolution used for the digitization of such bilevel pictures is determined by the need to capture all detail and is high enough to make the lines of the original picture have significant width in the digitized image. It is necessary to 'thin' such thick digital lines. The thinned digital line in the output is called skeleton. The purpose of thinning is two-fold. One is the reduction in the volume of picture data because only the skeleton instead of the whole picture may be stored. The other is the suitability of the skeleton as input data to a classification system.

Blum (1964) defined a simple thinning procedure that is well defined on the Euclidean plane and is called the medial axis transformation (MAT). The definition of the medial axis is given below.

Definition 3.1.1 Let A be a bilevel picture with boundary B . Suppose P is a point inside A . A nearest neighbour of P on B is a point M on B such that there is no other point on B whose distance from P is less than the distance PM . If P has more than one nearest neighbour on B , then P is said to be a medial point of A . The set of all medial points of A is called the medial axis of A .

In our terminology, the medial axis is a skeleton, but not all skeletons are medial axes. The thinning algorithms proposed in this chapter produce skeletons that are not medial axes.

Fig.3.1 shows some examples of medial axes with some of their major features. It can be seen that they are very sensitive to noise, since a small disturbance of the boundary not only causes a disturbance in one branch but also causes the creation of new branches (Fig.3.1(c)). Another observation is that for objects that are thin to start with, the medial axes provide substantial information about the shape. This is not true with thick objects, as shown in Fig.3.1(b).

The translation of the concept of the medial axes in the discrete plane is by no means straightforward or even feasible because of the difficulty of defining equality of distance between pixels over a discrete grid. Therefore, much is left to the intuition of the individual implementer. One approach is to

translate into discrete terms the Definition 3.1.1, that is, the medial axis is the locus of the centres of discs that are tangent to the boundary in more than one point. First the discrete version of the disc is defined and then a search is made over the area of the input picture to find places where such discs fit (Yokoi, Toriwaki and Fukumura, 1980). This is a time-consuming process and it has the disadvantage that it may produce skeletons that are not connected, even if the original picture is connected. On the other hand, the approach has the advantage that it allows reconstruction of the original picture by labeling each point with the radius of the disc centred at it.

Another approach abandons the original definition of Blum and centres on connectivity. Rosenfeld suggested that the examination of the local neighbourhood of a pixel shows whether that pixel may be deleted from the set without affecting the connectivity of the original picture or its complement (Rosenfeld and Pfaltz, 1966; 1967). Clearly, if such an approach is applied sequentially it will reduce any simply connected set to a single point. Therefore, the method should be implemented in a parallel way. It turns out that this is not enough and it is required that the order of examination of pixels be severely restricted. The major advantage of the method is that it uses only local operations and it is guaranteed to preserve connectivity. There are many variations of this method, using slightly

different types of masks (Tamura, 1978).

Because of the disadvantages discussed above, most of the literature on thinning deals with algorithms defined directly on a discrete grid. Hilditch (1969) defined skeletal pixels in the discrete plane on the basis of four criteria: thinness, position, connectivity and stability for skeletal pixels. There are other direct definitions which can be connected to the original definition (Defn. 3.1.1) of Blum.

One interpretation of the MAT given by Blum describes a bilevel picture as a patch of flammable material with a fire starting along its boundary. As the fire front propagates, parts of it collide and extinguish themselves. The medial axis is the locus of all such points. If we use sequential tracing of the boundary and preserve pixels where the boundary seems to fold on itself, then we have an implementation of Blum's illustration. Pixels that are traversed more than once during contour tracing, or pixels that have no neighbours in the interior of the picture, or pixels that have a third neighbour on the boundary, besides the previous and the next elements during tracing, are places where such collisions may occur and therefore are candidates for the skeleton. Such methods have been described by Pavlidis (1980;1982) and Arcelli (1981).

There is another class of thinning algorithms that use only the boundary of a picture (Chaudhuri, 1978; Shapiro, Pisa and Sklansky, 1981). In fact, the original definition

(Definition 3.1.1) is in terms of the boundary only. Montanari (1969) suggested deriving polygonal approximations of discrete binary pictures and then finding the skeletons of the polygons analytically.

Of the two thinning algorithms proposed in this chapter, the latter is an improved version of the former. None of the algorithms produces medial axes. But the second algorithm produces skeletons that are quite close to medial axes and provide substantial information about the shape of the input. The algorithms find the skeletal pixels sequentially and preserve the connectivity property of the input. Initially, one skeletal pixel is found. Then, in order to find the next skeletal pixel we search the neighbourhood of the current skeletal pixel. Such algorithms have the advantage that they are faster to implement on the usual general purpose sequential computers than parallel algorithms. As far as computational complexities are concerned, parallel thinning requires computations of order AW while sequential thinning usually requires computations of order BW where A is the image area, B is the area of the black parts of the image and W is the maximum half-width of the black parts. But the second algorithm proposed here requires computations of order B .

The thinning algorithms proposed in this chapter demand that the input picture be elongated preferably with somewhat

uniform width or thickness. It may be noted that the resulting skeleton is not unique. For a given input picture, the output skeleton will depend on the starting pixel or starting run at which we start tracing the skeleton. Nevertheless, the output skeleton (though not unique) preserves the global shape of the input picture (Fig.3.2). The second algorithm (Algorithm 3.2) proposed here overcomes many a drawback that is inherent in the first one (Algorithm 3.1).

In case the original picture was meant to be a thick curve in the plane, the output skeleton will be a digital curve which may be used as an input to Chapter 4 in this thesis.

3.2 Two Thinning Algorithms

Definition 3.2.1 A column run is a maximal connected sequence of nonzero-pixels in a column of a picture matrix while a row run is a maximal connected sequence of nonzero-pixels in a row. A diagonal run is a maximal connected sequence of nonzero-pixels in a diagonal (making either 45° or 135° angle with the x-axis) of a picture matrix. A run is either a column run or a row run or a diagonal run (Fig.3.3). The length of a run is the number of pixels in it.

Note that for any nonzero-pixel in a binary picture, there are four runs containing that pixel, that is, one column run, one row run and two diagonal runs.

Definition 3.2.2 Consider all possible row runs and column runs of an input picture and the frequency distribution of their run lengths. The run length with the maximum frequency is called the thickness of the input picture. In case of more than one run length with the maximum frequency, we take the lowest among such run lengths as the thickness. Note that this thickness is different from the thinness defined in Chapter 2 (Fig.3.2).

Algorithm 3.1 To thin an elongated binary picture A (Parui, Chaudhuri and Dutta Majumder, 1980).

Step 1 Take any run of A. (In Fig.3.2 it is the row run containing P). Take the 1-pixel in the middle of the run. Call it P. (If the run length is even, take any of the two pixels in the middle). Now consider the four types of runs containing P. From them choose the run with the maximum length. (If there are more than one run with the maximum length, choose any one of them. In Fig.3.2 there are two such runs. One is the column run and the other is a diagonal run. The column run is chosen). This run with the maximum length is called the starting run and will be a part of the output skeleton. Let Q_1 and Q_2 be the two end 1-pixels of this run. Q_1 and Q_2 are either end pixels or corner pixels or branching pixels of the output skeleton. ^{(An} end pixel is where the skeleton ends. A corner pixel is where the skeleton takes a turn of either 45° or 90° either on the right or on the left. A branching pixel is where the skeleton takes more than one turn). Go to Step 2.

(By the directional code at a skeletal pixel, we mean the direction made by the movement from the previous skeletal pixel to that pixel).

Step 2 Let DIR be the directional code at the pixel Q_1 . (DIR=4 in Fig.3.2). The directional code at Q_2 will be DIR+4 (mod.8). We will find in this step the skeletal pixels next to Q_1 on the left, if any. For that consider the direction $DIR_{-1} = \text{DIR} - 1 \pmod{8}$. If the neighbour of Q_1 in the direction DIR_{-1} is a 1-pixel, then the run from Q_1 in the direction DIR_{-1} will be the next part of the skeleton. Go to Step 3. If the neighbour of Q_1 in the direction DIR_{-1} is a 0-pixel, then try the same thing above with the direction $DIR_{-2} = \text{DIR} - 2 \pmod{8}$. If both the neighbours of Q_1 in the directions DIR_{-1} and DIR_{-2} are 0-pixels, then Q_1 is not a branching pixel (Fig.3.2). It is either an end pixel or a corner pixel with a turn on the right. Go to Step 3.

Step 3 In this step we will find the skeletal pixels next to Q_1 on the right, if any. For this purpose we consider the directions $DIR_1 = \text{DIR} + 1 \pmod{8}$ and $DIR_2 = \text{DIR} + 2 \pmod{8}$ and repeat what is done with DIR_{-1} and DIR_{-2} in Step 2. If all the neighbours of Q_1 in the directions DIR_{-1} , DIR_{-2} , DIR_1 and DIR_2 are 0-pixels, Q_1 is an end pixel of the skeleton and in that case we do not look for skeletal pixels next to Q_1 . Go to Step 4. If the neighbours of Q_1 in the directions DIR_{-1} and DIR_{-2} are

0-pixels and at least one neighbour of Q_1 in the direction DIR_1 and DIR_2 is a 1-pixel or vice-versa (i.e., there are skeletal pixels either on the left or on the right), then Q_1 is a corner pixel of the skeleton. In that case we take the end pixel of the new run and repeat Steps 2 and 3. Otherwise, Q_1 is a branching pixel of the skeleton (i.e., there are skeletal pixels both on the left and on the right of Q_1). In that case, we take up the left turn first and store the pixel Q_1 alongwith its direction on the right (i.e. DIR_1 or DIR_2) for future processing. In Fig.3.2 Q_1 has skeletal pixels only on the right and hence is a corner pixel of the skeleton.

Step 4 Repeat with Q_2 what is done with Q_1 in Steps 2 and 3.

Step 5 The end pixels of the new runs obtained during the process are the new Q_i 's. Repeat with these new Q_i 's what is done in Steps 2 and 3. The Q_i 's are given level 3 in Figs. 3.2, 3.4 - 3.7.

Step 6 In this step we will discuss how to terminate the algorithm which essentially involves the termination of a branch of the skeleton. One possibility is to terminate at an end pixel Q , that is, when all the neighbours of Q in the directions DIR_{-1} , DIR_{-2} , DIR_1 and DIR_2 are 0-pixels (mentioned in Step 3).

Another way to terminate a branch is to check whether the

branch takes a 180° turn within a short space. This check requires the knowledge about the thickness t of the input picture which is assumed to have somewhat uniform thickness. While tracing the skeletal pixels in Steps 1 to 5, we store for each current skeletal pixel the directional codes of the immediately preceding t pixels in the sequence of skeletal pixels (t is thickness). Suppose they are $DIR(-t), DIR(-t+1), \dots, DIR(-1)$. If the angle between the current DIR and some $DIR(-i)$ ($i=1, \dots, t$) is more than or equal to 180° , then remove from the skeleton the current pixel and the $(i-1)$ pixels immediately preceding it. That is, the pixel with the directional code $DIR(-i)$ will be an end pixel of a branch. Removed pixels are given level 4 in Fig.3.2, 3.4-3.7.

A third way to terminate a branch is to check whether the current branch touches or crosses a pixel that is already in the skeleton. For this purpose, we check for each current pixel P with the directional code DIR the two neighbours Q_1 and Q_2 of P in the directions DIR_{-1} and DIR_1 . If any one of the three pixels P, Q_1 and Q_2 is already a pixel in the skeleton, then terminate the current branch at P (P is included in the skeleton).

End of Algorithm 3.1.

The above algorithm gives higher values to the levels of the skeletal pixels. The skeletons in Figs.3.2, 3.4-3.7 consist of pixels with levels 2 and 3. The pixels with level 4

are removed from those skeletons because sharp turns with 180° angle are prohibited. When the skeletal pixels are traced sequentially their levels are changed to 2 from 1. The other levels (3,4 etc.) are used only to clarify some aspects of the algorithm. We will now give an example to show the non-uniqueness of the output skeleton. In Fig.3.2 the starting pixel is P and the starting run is the column run containing P. In Fig.3.4, for the same input picture, we take a diagonal run containing P as the starting run. (Since both the column run and the diagonal run have the same length). It can be seen that the skeleton in Fig.3.4 is different from the skeleton in Fig.3.2. Thus the output skeleton depends on the starting pixel and the starting run.

We will pose another problem where certain parts of the input picture may remain untraced and hence unthinned by Algorithm 3.1. In Fig.3.5 the starting pixel is P and the starting run is the column run containing P. The output skeleton is evidently incomplete. What we do to avoid this, that is, to make sure that every part of the picture is thinned, is to have a check after each scan by Algorithm 3.1. The check involves the two following conditions.

Condition (i) After a scan by Algorithm 3.1, there is a 1-pixel P in the input picture such that no run containing P has a pixel which is already in the skeleton obtained by the

previous scans. (Q is such a pixel in Fig.3.5).

Now we will construct Condition (ii). Suppose P is a 1-pixel in the input picture after a scan by Algorithm 3.1 such that P does not satisfy Condition (i). That is, there is a run containing P such that it contains some pixels which are already in the skeleton. If such pixels lie only on one side of P in the run, then let α = the distance between P and the nearest skeletal pixel in the run (which is defined as the number of 1-pixels in the run between P and the nearest skeletal pixel + 1). If such skeletal pixels lie on both the sides of P in the run, then suppose Q_1 and Q_2 are the nearest skeletal pixels to P on the two sides in the run (Fig.3.8). In this case let α = the distance between Q_1 and Q_2 . Let β = the minimum value of α over all the runs containing P as well as some skeletal pixels (either on one side or on both the sides of P).

Condition (ii) The value of β is greater than or equal to t where t means thickness. (The pixel R in Fig.3.5 satisfies this condition).

After a scan by Algorithm 3.1, if there is a 1-pixel P in the picture satisfying either Condition (i) or Condition (ii), then we repeat Algorithm 3.1 with P as the starting pixel. If there is no 1-pixel satisfying either Condition (i) or Condition (ii), then thinning is complete and no more scan by Algorithm 3.1 is needed.

For Fig.3.5 we repeat Algorithm 3.1 (i.e. the second scan)

with R as the starting pixel. The output skeleton after the second scan is shown in Fig.3.6. It can be seen that no 1-pixel in Fig.3.6 satisfies any of the two conditions above. Hence thinning is complete after the second scan by Algorithm 3.1 (Fig.3.6).

Now we will show how the presence of noise in the input picture disturbs the output skeleton. Some noise (addition of three 1-pixels on the top and removal of one 1-pixel from the bottom) is incorporated into the input picture in Fig.3.2 and the new picture is shown in Fig.3.7. The Algorithm 3.1 is applied to the new picture with P as the starting pixel (the same starting pixel as in Fig.3.2) and it is clear that the output skeleton in Fig.3.7 is quite different from the one in Fig.3.2.

The shortcomings of Algorithm 3.1 are summarized below.

(i) The output skeleton differs very much from the medial axis. In fact, a great part of it may coincide with the boundary of the input picture.

(ii) The output skeleton is not unique for a given picture. It depends on the starting pixel and the starting run.

(iii) The presence of noise in the input picture may disturb the output skeleton greatly (Fig.3.7).

(iv) When the input picture is a thick straight line segment, the output skeleton is not necessarily a thin straight

line segment (Fig.3.9(c)). It may be a zigzag structure (Fig. 3.9(a)). This happens when the input line segment makes with the x-axis an angle which is not a multiple of 45° (Fig.3.9).

(v) One single scan by Algorithm 3.1 may not cover (i.e., may not thin) all the parts of the input picture (Fig.3.5). In that case, more scans by Algorithm 3.1 are needed (Fig.3.6) and thinning becomes more time consuming.

We will now present a modified version of Algorithm 3.1 which will reduce the above shortcomings to a great extent. In Algorithm 3.1 a branching or a turning of the skeleton takes place only at the end pixels of a run (i.e., at pixels with level 3) which are always boundary pixels. In the new algorithm (Algorithm 3.2) the following two modifications are made.

(i) A branching or a turning in the skeleton takes place well before (in fact, approximately $t/2$ pixels before) an end pixel of a run. As a consequence, the output skeleton will tend to keep to the middle of the input picture.

(ii) For each current entry (i.e. current pixel) into the skeleton we check whether there is any branching from that pixel in the directions DIR_{-1} and DIR_1 , that is, whether there is any branching with angle $+45^\circ$ or -45° . For that purpose we find the length of the runs from the current pixel in the directions DIR_{-1} and DIR_1 . As a consequence of this, only one scan by the new

algorithm will be sufficient to cover (i.e. to thin) all the parts of the input picture.

The new algorithm is as follows:

Algorithm 3.2 (Modified version of Algorithm 3.1)

Step 1 We put one condition on the way the starting pixel and the starting run are found in Step 1 of Algorithm 3.1. Instead of taking any run of the input picture to start with, we here start with any row run or column run whose length is at least approximately equal to the thickness t of the input picture. Then the starting pixel P and the starting run are found in the same way as in Step 1 of Algorithm 3.1. We also make sure that all the pixels in the starting run are not boundary pixels so that the output skeletal pixels keep to the middle of the input picture. Unlike in Algorithm 3.1, here the whole starting run is not a part of the output skeleton as will be seen later. We trace the skeleton from the starting pixel P along the starting run in both directions. Let P_1 be the next pixel to P in the starting run in the direction DIR . Let t_0 be $t/2$ if t is even and $(t-1)/2$ if t is odd, where t is the thickness of the input picture.

Step 2 Check the three runs from the pixel P_1 in the directions DIR , DIR_{-1} and DIR_1 (DIR_{-1} and DIR_1 are defined in Algorithm 3.1). If the run lengths (these are, in fact, partial

run lengths, i.e., the number of nonzero-pixels excluding P_1 in the run and on the one side of P_1) in all these three directions are less than or equal to t_0 , then P_1 is the end pixel of a branch of the output skeleton. The end pixels are given level 3 in Figs. 3.10 and 3.11 and are included in the output skeleton.

If the run length in the direction DIR only is more than t_0 and the other run lengths are less than or equal to t_0 , then there is no branching or turning of the output skeleton at P_1 and the output skeleton proceeds in the current direction DIR. Include the current P_1 in the skeleton and set new P_1 = the pixel next to the current P_1 in the direction DIR and repeat Step 2.

If the length of the run from P_1 (excluding P_1) in the direction DIR_{-1} or DIR_1 is more than t_0 , then find all the successive pixels immediately next to P_1 in the original run (i.e., in the direction DIR) with the above property. Call this set of pixels a branching set because from each of these pixels a branch in the direction DIR_{-1} or DIR_1 (i.e., on the left or right) is possible. In general, with each branching set is associated only one direction (either DIR_{-1} or DIR_1), although in some cases both may be associated (when there are branches both on the left and on the right from the same set of pixels). The branching set pixels are given levels 5 to 9 in Figs. 3.10

and 3.11 (different levels are used only for illustrations so that one branching set is distinguishable from the others in the figures). The number of pixels in a branching set is approximately the thickness of the part of the input picture which the corresponding branch represents. For example in Fig.3.11, the size of the branching set containing pixels with level 5 is 5 which is the thickness of the lower horizontal part of the input picture. Similarly, all the three branching sets containing pixels with level 6 have size 3 which is the thickness of the upper horizontal part. The middle pixel of each branching set is called a branching pixel and is stored separately because there will be a branch/turn in the skeleton from this pixel in the direction DIR_{-1} or DIR_1 or both. (If there are even number of pixels in a branching set, then the second one of the two middle pixels is the branching pixel). The branching pixels are marked $\odot$ in Figs.3.10 and 3.11. Now, all the pixels of a branching set are not necessarily skeletal pixels. The pixels upto (and including) the branching pixel are always skeletal pixels. But a pixel of a branching set beyond the branching pixel is not a skeletal pixel if its distance from the end pixel of the run (in the direction DIR) of which the branching set is a part, is less than or equal to t_0 . Otherwise, a pixel of the branching set is a skeletal pixel. The pixels that are marked $\square$ in Figs. 3.10 and 3.11

belong to branching sets but are not skeletal pixels. On the other hand, the whole branching sets containing pixels with level 5 in Figs. 3.10 and 3.11 entirely belong to the skeleton.

Before tracing the branches from the branching pixels, the algorithm completes the skeleton in the original direction DIR. That is, in Fig. 3.10 three more skeletal pixels are obtained beyond the pixels with level 5 in the direction DIR=8 before branching out from the branching pixel Q. In the process all the branching pixels are stored sequentially along with the corresponding directions (DIR_{-1} or DIR_1 or both). After the skeleton is completed in the original direction DIR, the branching pixels are taken up sequentially and Step 2 is repeated with a new direction DIR (i.e. old DIR_{-1} or DIR_1) and the corresponding new values of DIR_{-1} and DIR_1 . For example, in Fig. 3.10, originally DIR=8, $DIR_{-1}=7$ and $DIR_1=1$ and when the branch from the branching pixel Q is traced, the new values are DIR=7, $DIR_{-1}=6$ and $DIR_1=8$.

Step 3 (Termination of a branch): There are two ways to terminate a branch in the skeleton.

(i) Suppose there is a skeletal pixel P such that the lengths of the runs from P in the directions DIR, DIR_{-1} and DIR_1 are all less than or equal to t_0 . Then P is an end pixel and we terminate a branch at P. The pixels with level 3 in Figs. 3.10 and 3.11 are all end pixels (end pixels are included

in the skeleton).

(ii) Here we check whether the current branch crosses or touches an already obtained branch of the skeleton. Suppose a new entry (i.e., a new pixel) P in the skeleton with the direction DIR is such that either P or at least one of the two neighbouring pixels of P in the directions DIR_{-1} and DIR_1 is already a skeletal pixel. Then P is a crossing pixel and we terminate the current branch at P . P is included into the skeleton. The pixel with level 7 at the top left in Fig.3.11 is a crossing pixel. There are no crossing pixels in Fig.3.10.

Step 4 (Blocking of a false branch): It is possible that a branching pixel found in Step 2 is not a genuine branching pixel in the sense that from that pixel there should not be a branch in the corresponding direction. For example, in Fig.3.11, the pixels Q and R are false branching pixels. In order to prohibit branching from such a pixel, we do the following:

Suppose S is a branching set. Consider all the runs from the pixels in S in the corresponding branching direction (i.e. DIR_{-1} or DIR_1). If none of those runs contains a pixel that is already in the skeleton traced so far, then the branching pixel of S is genuine. (All the branching pixels marked  except Q and R in Fig.3.11 are genuine). Otherwise, there will be some pixels in S from which the runs in the branching direction will contain some skeletal pixels. For each such pixel in S ,

let α = the number of 1-pixels between that pixel and the nearest skeletal pixel in the run in the branching direction. For such branching set S , let β = the maximum value of α over all those pixels in S . If the value of β is more than the thickness t of the input picture, then also the branching pixel of S is genuine (Q represents such a genuine branching pixel in Fig.3.12). Otherwise, the branching pixel of S is not genuine. The pixels Q and R in Fig.3.11 are such branching pixels. We remove these ~~false~~ branching pixels from the storage and take up the next branching pixel in the storage and repeat Step 4.

Step 5 (Removal of a false protrusion): When there is a branch/turn in the skeleton, there may be a false skeletal protrusion beyond the branching pixel in the original direction DIR. If the number of skeletal pixels beyond the branching pixel in the direction DIR is less than or equal to t_0 , then remove such pixels from the skeleton. (Such pixels are marked Δ in Fig.3.11). Otherwise, do not remove. (For example, the five skeletal pixels above the branching pixel Q in Fig.3.10 are not removed).

Algorithm 3.2 terminates when there are no more branching pixels left in the storage.

End of Algorithm 3.2.

The advantages of Algorithm 3.2 over Algorithm 3.1 are summarized below.

(i) Algorithm 3.2 also does not give a unique skeleton for a given picture (Fig.3.13). But the variation among the possible output skeletons is much lower than that in the case of Algorithm 3.1. This is because an output skeleton traced by Algorithm 3.2 nearly keeps to the middle of the input picture.

(ii) Figs.3.7 and 3.14(b) show output skeletons traced by Algorithms 3.1 and 3.2 respectively for the same noisy input picture with the same starting pixel and starting run. It is clear that the distortion in Fig.3.14(b) is much less than that in Fig.3.7. The upward protrusion (i.e. the pixels marked $\square$) in Fig.3.14(b) only reflects the noise on the top of the input picture. The other parts of the output skeleton are nearly undisturbed by the noise in Fig.3.14(b), which is not true for Fig.3.7.

(iii) In the case of Algorithm 3.2 also, the output skeleton (Fig.3.9(b)) of a thick straight line segment is not necessarily a thin straight line segment. But the output skeleton in this case is much closer to the straight line skeleton (Fig.3.9(c)) than the output skeleton (Fig.3.9(a)) in the earlier case.

(iv) Algorithm 3.2 unlike Algorithm 3.1 covers all the parts of the input picture in one scan.

3.3 Conclusion

It is clear that Algorithm 3.2 takes more time than Algorithm 3.1 which is suitable for quite a restricted class of input patterns while Algorithm 3.2 is suitable for a relatively broader class of patterns. Output skeletons by Algorithm 3.1 preserve the global shape of input patterns (like fingerprints) that are smooth in nature, noise free and do not have branches or sharp turns (Fig.3.15). In that case, only one scan by Algorithm 3.1 covers the whole input picture and the output which is a digital curve may be used as an input to algorithms proposed in Chapter 4 in this thesis.

It can be seen that Algorithm 3.2 is very sensitive to the thickness of the input picture. The algorithm can be improved by modifying the value of t . If the thickness of the input picture varies from one part to another, then the minimum thickness is considered and is treated as the value of t in Algorithm 3.2. In that case, in picture parts with greater thickness there may be false branching sets or false pixels in a branching set. To solve this, first the pixels with relatively smaller run lengths (in the direction DIR_{-1} or DIR_1) are removed from the branching set, and then the branching sets giving rise to branches that are only small protrusions are ignored.

In the thinning algorithms discussed in Section 3.2, a diagonal run and a row/column run having the same length

(discrete) do not have the same analogue length. In fact, the analogue length of the diagonal run is $\sqrt{2}$ times that of the row/column run. So, better results may be achieved by those algorithms if we multiply the existing diagonal run lengths by $\sqrt{2}$.

Algorithms 3.1 and 3.2 are sequential algorithms and hence are appropriate for usual general purpose sequential computers but not for parallel processors. Now, if all the skeletal pixels are found in somewhat the same way the starting pixel is found in Algorithm 3.2, then implementation on a parallel computer is possible. The definition of the new skeleton in the analogue case is the following: Let A be a subset in the plane. A point P in A is a skeletal point of A if P is the mid-point of the shortest chord in A containing P . The skeleton of A is the set of all skeletal points of A . This definition has the disadvantage that the skeleton of a connected set may be disconnected (Fig.3.16). But its translation in the digital domain is straight-forward. For each pixel, the four runs (row run, column run and two diagonal runs) are considered, and the pixel is skeletal if it is the mid-pixel of the run with the minimum length. If the minimum run length is even, then of the two middle pixels, the one on the right or on the top is chosen. It can be seen that the output skeleton of this parallel algorithm is disconnected at the joints where there is a branching

or crossing (Fig.3.17). That is, each connected part of the skeleton represents a simple part or arc of the input pattern. Thus, such a disconnected skeleton can be used for decomposition of the input pattern. However, efforts should be made to join the disconnected parts of the skeleton so that we can simultaneously retain parallelism of the algorithm and connectivity of the output skeleton.

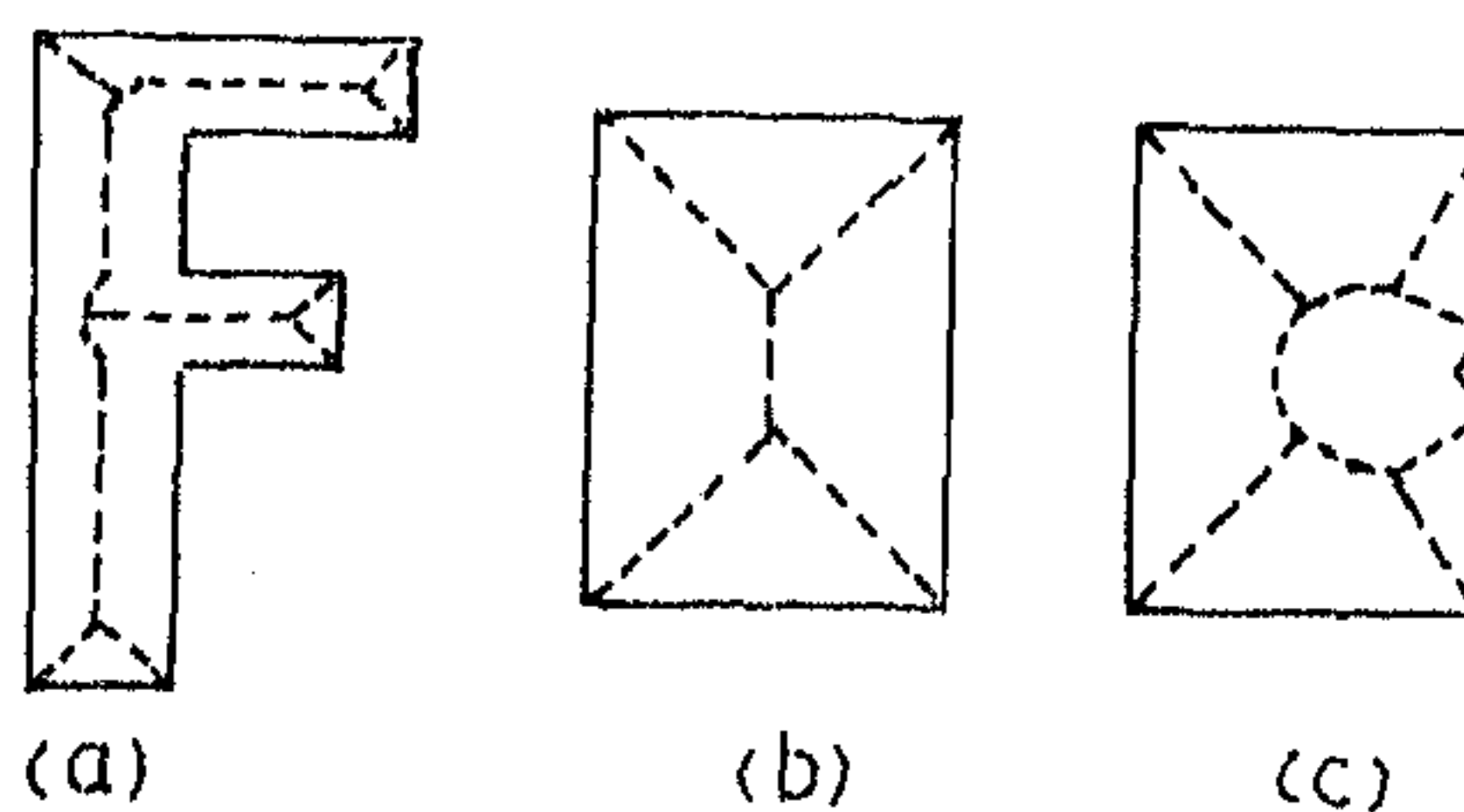


Fig.3.1 Examples of medial axes. The broken lines indicate the medial axes. (a) The lineal structure of the silhouette corresponds closely to that of the medial axis. (b) No simple correspondence exists between the medial axis and object structure. (c) Small amounts of noise disturb the medial axis drastically (Pavlidis, 1982).

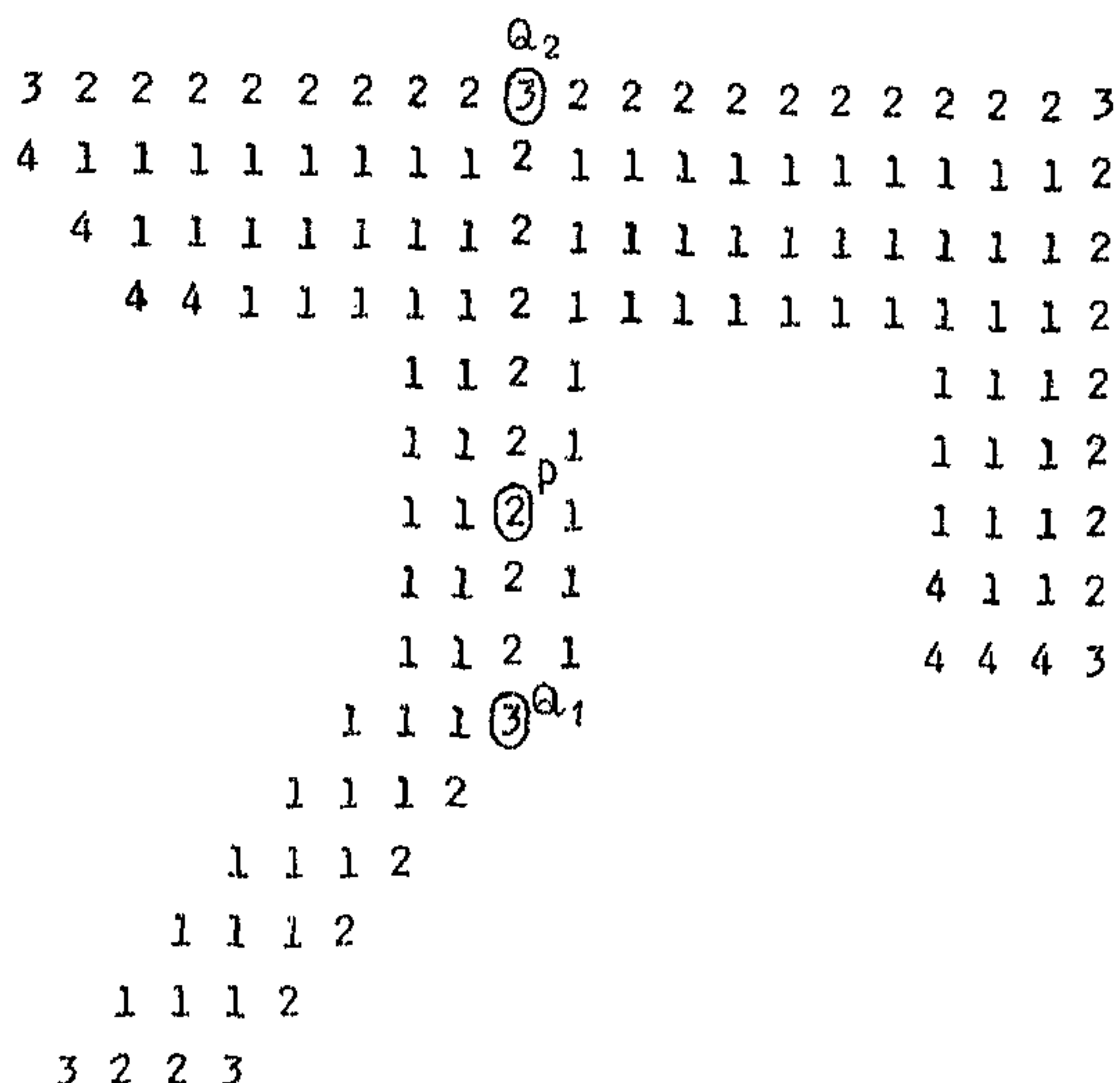


Fig.3.2 The pixels with levels 2 or 3 constitute the output skeleton. The pixels with level 4 are removed from the skeleton because sharp turns with 180° angle are prohibited. P is the starting pixel and the column run containing P is the starting run. $t=4$, $t_0=2$. The skeleton above is found by Algorithm 3.1.

Frequency distribution of run lengths for Fig.3.2

Runlength	1	2	3	4	9	10	11	12	19	20	21
Frequency	1	2	2	30	5	1	1	1	1	1	2

So the thickness of the picture in Fig.3.2 is 4.


```

1 1 2 1
 1 2 1 1
   2 1 1 1
    3 3 3 3
     1 1 1 1
      5 1 1 4
       1 5 4 1
        1 4 5 1

```

2's indicate a column run
 3's indicate a row run
 4's and 5's indicate diagonal runs

Fig. 3.3

```

3 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 3
4 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2
  4 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2
    4 4 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2
      1 1 1 2
        1 1 1 3
          1 1 ②p 1
            1 2 1 1
              2 1 1 1
                2 1 1 1
                  2 1 1 1
                    2 1 1 1
                      2 1 1 1
                        2 1 1 1
                          2 1 1 1

```

Fig.3.4 The input picture is the same as in Fig.3.2, but the output skeleton is different from the one in Fig.3.2 because of a different starting run. The starting pixel P is the same as in Fig.3.2. The output skeleton is by Algorithm 3.1. $t=4$, $t_0=2$.


```

3 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 3 4
4 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 4
  4 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1
    4 4 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1
      R(2) 1 1 1                1 1 2 1
        2 1 1 1                1 1 2 1
        2 1 1 1                1 1 2 1
        2 1 1 1                4 1 2 4
        2 1 1 1                4 4 3 4
          1 2 1 1
            1 1 2 1
              1 1 1 3
                1 1 1 2
                  1 1 1 2
                    3 2 2 3

```

Fig.3.6 The input picture of Fig.3.5 after the second scan by Algorithm 3.1. The starting pixel of the second scan is R. $t=4$, $t_0=2$.


```

          3 3 3
4 4 1 1 1 1 1 2 1 2 1 2 1 1 1 1 1 1 1 4 4
4 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1 4
  4 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 4
    3 2 3 1 1 1 1 2 1 1 1 1 3 2 2 2 2 2 2 3
      1 1 2 1                      1 1 1 2
      1 1 2 1                      1 1 1 2
      1 1 ②p1                      1 1 1 2
      1 1 2 1                      4 1 1 2
      1 1 2 1                      4 4 4 3
        1 1 1 3
          1 1 1 2
            3 2 2 3
              2 1 1
                2 1 1 1
                  3 1 1 1

```

Fig.3.7 The input picture here is obtained by adding some noise to the picture in Fig.3.2. The starting pixel P and the starting run are the same as in Fig.3.2. The pixels with levels 2 and 3 form the skeleton. If this skeleton is compared with the one in Fig.3.2, it can be seen that noise has greatly disturbed the output skeleton which is found by Algorithm 3.1.


```

1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 ② 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 1 1 2
      1 1 1          1 1 2
      1 1 ①P        1 1 2
      1 1 1          1 1 2
1 1 1 1 1 1 1 1 1 1 1 1 2
1 1 1 1 1 1 1 1 1 1 1 1 2
2 2 2 2 2 2 ② 2 2 2 2 2 2
              Q2

```

Fig. 3.8 The pixels with level 2 indicate the skeleton obtained after the first scan by Algorithm 3.1. The column run containing P has skeletal pixels Q_1 and Q_2 on the two sides of P. Here P does not satisfy Condition (i) but satisfies Condition (ii). So the next scan starts with P. $t=3$, $t_0=2$.

```

          1 1 1 1
          1 1 1 1 1
          1 1 1 3 1 1
          1 1 1 1 2 1 1
          1 1 1 1 1 2 1 1
          ⑦⑦⑦ 7 ⑥ 1 5 1 1
          1 1 2 1 1 1 6 5 1 1
          1 1 2 1 1 1 ⑤ 1 1
          1 1 2 1 1 1 5 1 1
          1 1 3 1 1 1 5 1 1
          1 1 1 1 1 1 2 1 1
          1 1 1 1 1 1 ②P 1 1
          1 1 2 1 1
          1 1 3 1 1
          1 1 1 1 1
          1 1 1 1 1

```

Fig. 3.10 The pixels with levels more than 1 (except two pixels marked $\boxed{\cdot}$) form the skeleton traced by Algorithm 3.2. The starting pixel is P. The other three pixels marked $\odot$ are branching pixels (defined in Step 2 of Algorithm 3.2). $t=5$, $t_0=3$.

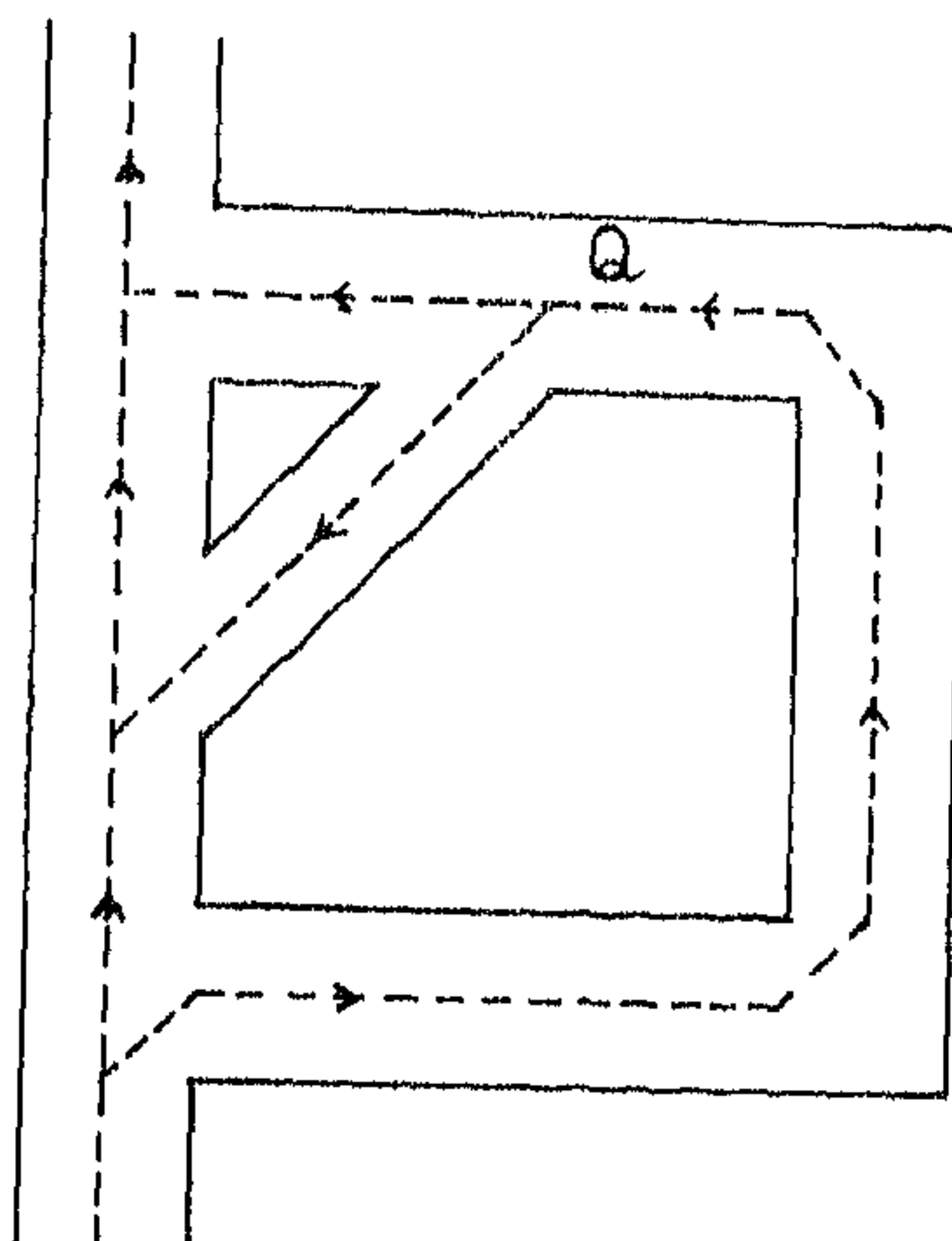


Fig.3.12 Q indicates a genuine branching pixel.

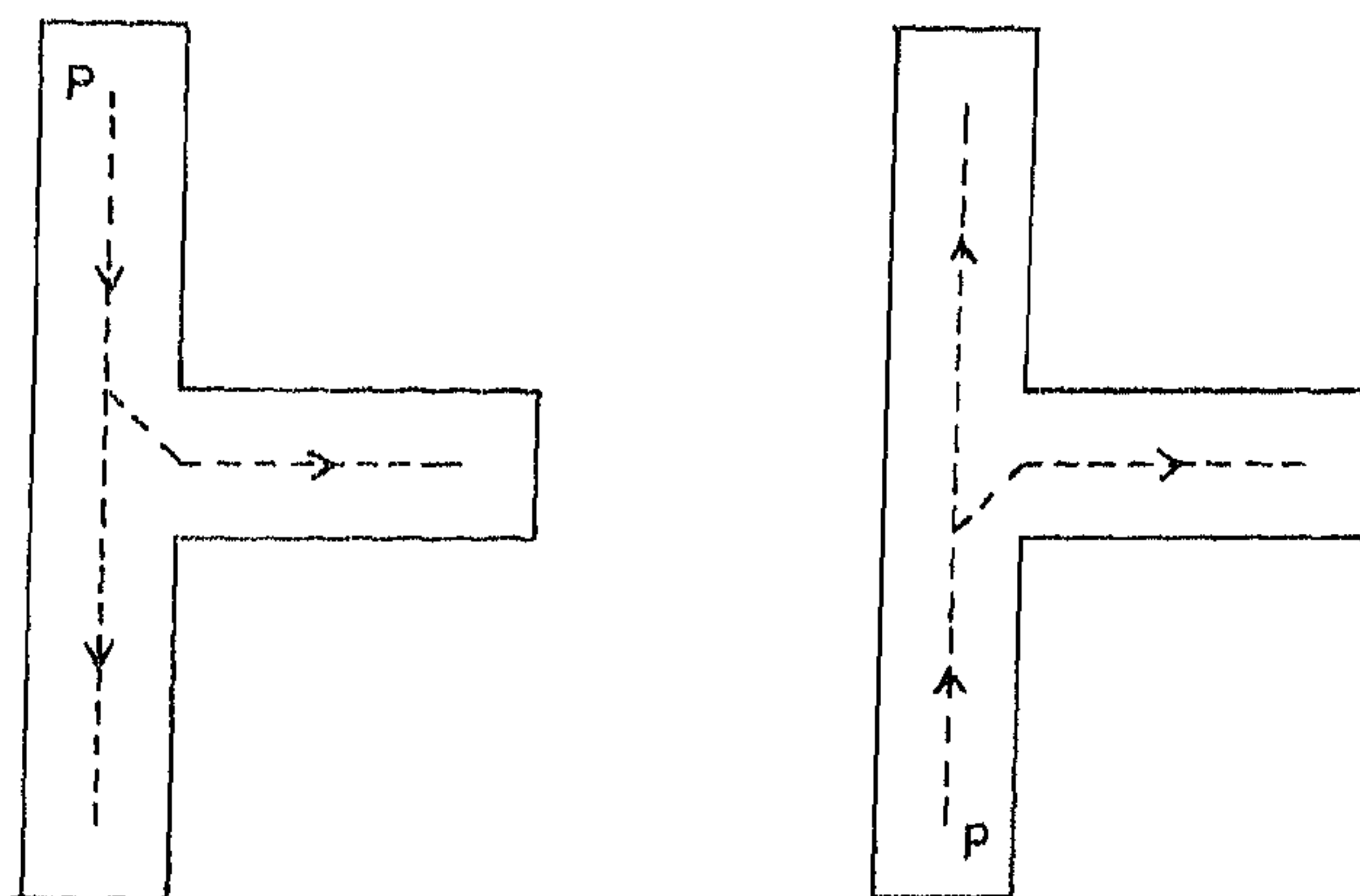


Fig.3.13 For the same input pattern, Algorithm 3.2 gives slightly different output skeletons with different starting pixels P.


```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 2 2 2 2 1 1 1 1 1 2 2 2 2 2 1 1 1
  1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 1
    1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1
      1 1 2 1                1 1 2 1
        1 1 2 1                1 1 2 1
          1 1 ②p 1                1 1 2 1
            1 1 2 1                1 1 1 1
              1 1 2 1                1 1 1 1
                1 1 2 1
                  1 1 2 1
                    1 1 2 1
                      1 1 2 1
                        1 1 1 1
                          1 1 1 1

```

Fig.3.14(a) The input picture here is the same as in Fig.3.2. The pixels with level 2 form the output skeleton which has been found by Algorithm 3.2. The starting pixel is P and the starting run is the column run containing P. In Fig.3.14(b) we will see how noise disturbs the skeleton.


```

      1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 2 2 2 2 1 1 [2] 1 1 2 2 2 2 2 1 1 1 1
  1 1 1 1 1 1 2 1 [2] 1 2 1 1 1 1 1 2 1 1 1
    1 1 1 1 1 1 2 [2] 2 1 1 1 1 1 1 1 2 1 1
      1 1 2 1
        1 1 2 1
        1 1 2p 1
        1 1 (2) 1
        1 1 2 1
        1 2 1 1
          1 2 1 1
            1 2 1 1
              1 2 1 1
                1 2 1
                  1 1 1 1
                    1 1 1 1

```

Fig.3.14(b) The input picture here is obtained by adding some noise to the picture in Fig.3.14(a). The starting pixel P and the starting run are the same as in Fig.3.14(a). The pixels with level 2 form the skeleton found by Algorithm 3.2. This skeleton is not much different from the one in Fig.3.14(a). But in the case of Algorithm 3.1, the disturbance in the skeleton due to noise was much more (Fig.3.7). Above three pixels marked [2] are added to the skeleton due to noise.

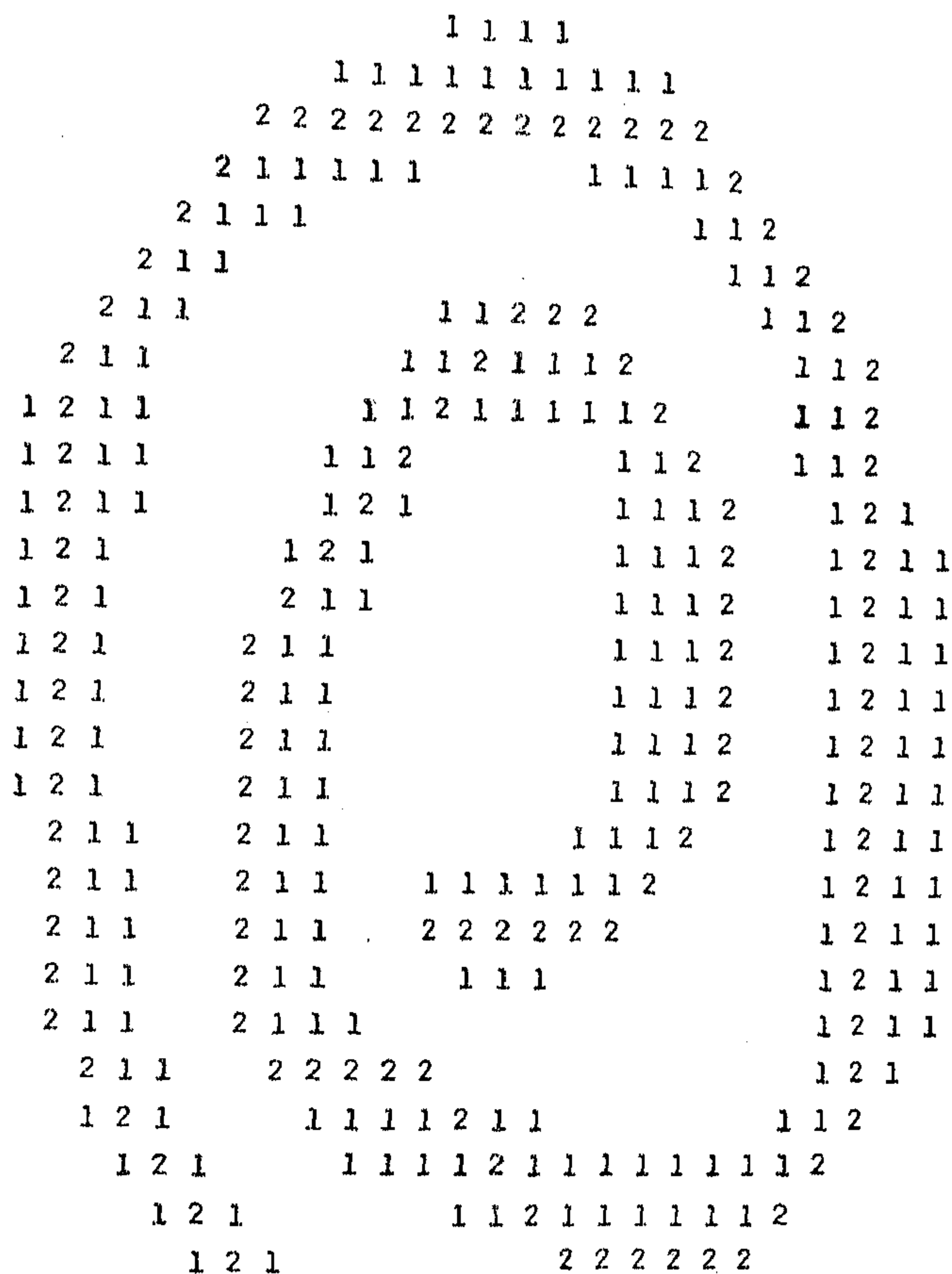


Fig.3.15(a) Algorithm 3.1 gives satisfactory results for smooth input patterns that have no branches or sharp turns. Above, pixels with level 2 form the output skeleton found by Algorithm 3.1.

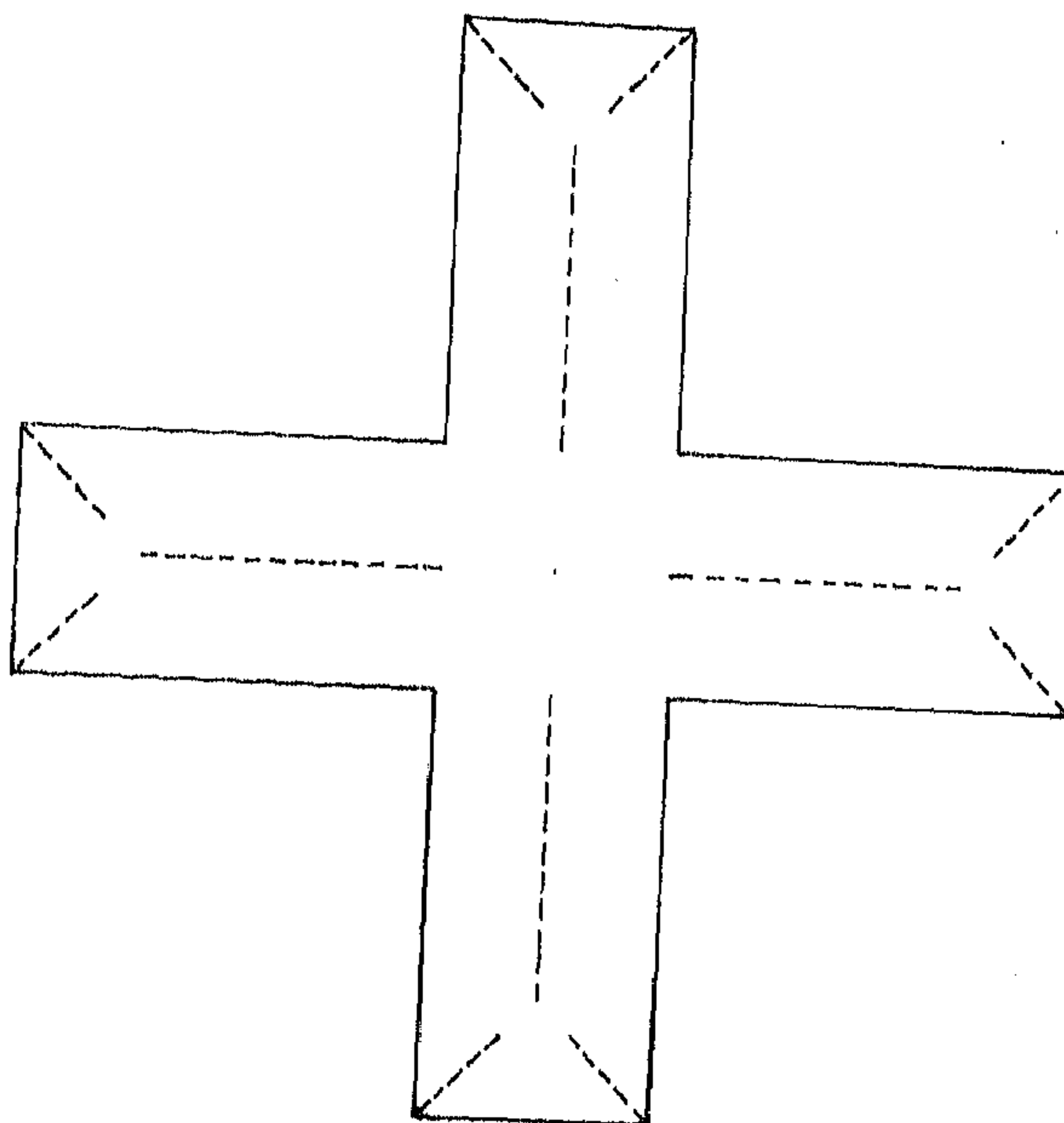


Fig.3.16 The input pattern above is connected. Its skeleton found by the parallel algorithm proposed in Section 3.3 is indicated by broken lines. It can be seen that the skeleton is disconnected, though the input pattern is connected.


```

      1 2 1 2 1
      1 2 2 2 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1
2 2 2 2 2 2 1 1 2 1 1 2 2 2 2 2 2 2 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      1 1 2 1 1
      2 2 1 2 2
      1 1 1 1 1

```

g.3.17 Skeleton (indicated by 2's) by the parallel algorithm proposed in Section 3.3. Connected parts of the skeleton indicate simple parts or areas of the input pattern.

CHAPTER 4

Shape Distance and Shape Similarity of Two-Dimensional Curves

4.1. Introduction

This chapter deals with the shape of two-dimensional curves and proposes two methods for quantifying shape distance and shape similarity. This quantified shape distance or shape similarity can be useful in classifying a new shape as one of several known shapes. A curve will mean an analogue curve and a digital curve will mean its digitized form (Freeman, 1974). Now strictly speaking a curve is not a bilevel analogue picture as defined in Chapter 1. But curves are studied in order to study bilevel pictures or plane regions which are often described as curves through transformations like thinning and border tracing

Curves are usually approximated by polylines as well as by splines and other curves (Pavlidis and Horowitz, 1974; Pavlidis, 1977; de Boor, 1978). In this thesis we will consider only polyline description which is a concatenation of some line segments and which can be represented as a finite sequence of points in the plane (Ballard and Brown, 1982). In general the points are so chosen as to minimize the difference between the original curve and the polyline

described by those points. (The measure of difference is usually either the maximum error or the integral square error (Pavlidis, 1982)). In that case the arc distance between two consecutive points is not necessarily the same. But in this thesis, a curve is described as a finite sequence of points on the curve such that any two consecutive points will have the same arc distance between them. This kind of description will not in general minimize the difference between the original curve and the resulting polyline. But this description is necessary for matching the shapes of two curves as will be clear later in this chapter. Moreover, this description also is information preserving as the former one.

Details of various theories and algorithms for approximating curves are available elsewhere (Ballard and Brown, 1982; Pavlidis, 1982).

In the next section, we give a formal definition of an open curve and then define the shape of an open curve as an equivalence class of open curves within which any curve can be transformed to any other curve through translation, dilation and rotation. In Section 4.3, we propose two measures of shape distance (also of shape similarity). For that purpose a one-to-one correspondence is established between the points on any two open curves of the same length. One measure is based on the Euclidean distances between corresponding points

on the two curves, the other on the differences between the slopes of the two curves at corresponding points. In Section 4.4, we describe an open curve as a polyline such that the two end points of the open curve are the first and last vertices of the polyline and the arc distance between any two consecutive vertices is the same. It should be noted that any three consecutive vertices are not necessarily non-collinear. The information preserving property of this description is also proved. Section 4.5 deals with digital open curves and techniques for computing their shape distance and shape similarity. Results are given in Section 4.6.

Before going on to the next section let us make an extension of the discrete directional codes in I_8 to continuous directional codes. Each pair of adjacent codes makes an angle of 45° between them. Let d be any real number in the half-open interval $(0, 8] = S_8$. Note that $i > d > i-1$ for some $i \in I_8$. Then the extended directional code d defines a direction that makes an angle of $(i-d)45^\circ$ with the direction i on the anticlockwise side (Fig.4.1). Thus there is a one-to-one correspondence between S_8 and the set of all directions in the plane. In fact, S_8 linearly represents all possible angles in $(0, 2\pi]$ in the plane (S.K. Parui and D. Dutta Majumder, 1982B; 1983A; 1983C; 1983D).

4.2 Analytic Definitions of a Curve and Its Shape

We will first give the analytic definition of a path. A curve is defined as a class of paths that give rise to the same image in the plane.

Definition 4.2.1 A path is a function f from the closed interval $[0,1]$ to $\mathbb{R}^2$ such that

- (i) f is continuous,
- (ii) f is piece-wise smooth,
- (iii) $t_1, t_2 \in [0,1]$ and $t_1 \neq t_2$ implies $f(t_1) \neq f(t_2)$ and $t_1, t_2 \in (0,1]$ and $t_1 \neq t_2$ implies $f(t_1) \neq f(t_2)$.

The image of f is $I(f) = \{f(t) : t \in [0,1]\}$. It can be seen that two different paths f_1 and f_2 can have the same image. For example, let $f_1(t) = (0, t)$ for $t \in [0,1]$ and $f_2(t) = (0, 2t)$ for $t \in [0, 1/4]$, $f_2(t) = (0, \frac{2t+1}{3})$ for $t \in [1/4, 1]$. Note that f_1 and f_2 are paths and $I(f_1) = I(f_2) =$ the line segment joining the two points $(0,0)$ and $(0,1)$.

Definition 4.2.2 A path f is said to be closed if $f(0) = f(1)$ and open if $f(0) \neq f(1)$. Note that if f is an open path and $t_1, t_2 \in [0,1]$ such that $t_1 \neq t_2$ then $f(t_1) \neq f(t_2)$.

Definition 4.2.3 Two open paths f_1 and f_2 are equivalent if $I(f_1) = I(f_2)$ and $f_1(0) = f_2(0)$.

Definition 4.2.4 Let F be the class of all open paths. E is a relation defined on F such that $(f_1, f_2) \in E$ if f_1 and f_2 are equivalent.

Proposition 4.2.1 E is an equivalence relation in F .

Proof. Trivial.

Definition 4.2.5 Each equivalence class generated by E in F is defined to be an open curve.

Definition 4.2.6 Let f be an open path. For any partition of $[0, 1]$ given by $P = \{t_0, t_1, \dots, t_m\}$ where $0 = t_0 < t_1 < \dots < t_{m-1} < t_m = 1$, define $\Delta_f(P) = \sum_{k=1}^m \|f(t_k) - f(t_{k-1})\|$ where $\|\cdot\|$ is the usual Euclidean norm. The length of f is defined to be $L(f) = \sup_P \Delta_f(P)$ where supremum is taken over all partitions. Since f is piecewise smooth by definition, $L(f)$ is always finite (Apostol, 1979).

Proposition 4.2.2 For any open path f , $L(f) = \int_0^1 \|f'(t)\| dt$ (Apostol, 1979).

Proposition 4.2.3 If two paths are equivalent then $L(f_1) = L(f_2)$.

Proof. For any partition $P_1 = \{t_0, t_1, \dots, t_m\}$ there is a partition $P_2 = \{s_0, s_1, \dots, s_m\}$ such that $f_1(t_i) = f_2(s_i)$ for $i = 0, 1, \dots, m$. So $L(f_2) \geq L(f_1)$. Similarly, $L(f_1) \geq L(f_2)$. Hence the result.

Now, we will construct a representation of an open curve in an analytic form. Let f be an open path such that $f = (g, h)$ where g and h are real-valued continuous functions

on $[0,1]$ and are piecewise smooth. Let $s = z(t) = \int_0^t \|f'(t_1)\| dt_1$ where $t \in [0,1]$. Now from (iii) in Definition 4.2.1, z is a strictly increasing continuous function on $[0,1]$ (Apostol, 1979). Thus there is a one-to-one correspondence between t in $[0,1]$ and s in $[0,L]$ where $L=L(f)$. Thus $f(t) = (g(t), h(t)) = (g(z^{-1}(s)), h(z^{-1}(s)))$ can be expressed as $x(s) = (u(s), v(s))$ where u and v are real-valued continuous functions on $[0,L]$ and are piecewise smooth.

Definition 4.2.7 $(u(s), v(s))$ is called the arc length representation of the open path $f(t)$.

Proposition 4.2.4 Two equivalent open paths give rise to the same arc length representation.

Proof. Let f_1 and f_2 be two equivalent open paths. Let t_1 be any element in $[0,1]$. Then there exists a unique $t_2 \in [0,1]$ such that $f_1(t_1) = f_2(t_2)$. Now since $f_1(0) = f_2(0)$ and f_1 and f_2 are continuous, $\{f_1(t) : t \in [0, t_1]\} = \{f_2(t) : t \in [0, t_2]\}$. So, the length of f_1 upto t_1 is the same as that of f_2 upto t_2 . The proof will be similar to the one for Proposition 4.2.3. That is, $z_1(t_1) = \int_0^{t_1} \|f_1'(t)\| dt = z_2(t_2) = \int_0^{t_2} \|f_2'(t)\| dt = s$, say. Now, if x_1 and x_2 are the arc length representations of f_1 and f_2 respectively, $x_1(s) = f_1(t_1)$ and $x_2(s) = f_2(t_2)$. Since $f_1(t_1) = f_2(t_2)$, $x_1(s) = x_2(s)$. This is true for any s in $[0,L]$. Hence the proposition.

Thus, an open curve, i.e., a class of equivalent open paths can be represented uniquely by the arc length representation. From now on we will write an open curve with length L as $\{x(s) : s \in [0, L]\}$ or $\{(u(s), v(s)) : s \in [0, L]\}$ or $\{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$.

We already defined the operations translation, dilation and rotation on a subset in the plane. Now, we define them for an open path and an open curve.

Definition 4.2.8 An open path f_2 is said to be a translation of an open path f_1 if $f_1(t) - f_2(t)$ is a constant for all t in $[0, 1]$. f_2 is said to be a dilation of f_1 if $f_2(t) = cf_1(t)$ for all t in $[0, 1]$ where c is a positive real number. f_2 is said to be a rotation of f_1 if $f_2 = f_1 \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}$ where α belongs to $[0, 2\pi)$. Note that the rotation is by an angle α in the anticlockwise direction.

Proposition 4.2.5 If two open paths f_1 and f_2 are equivalent then a translation/dilation/rotation of f_1 and the same translation/dilation/rotation of f_2 are equivalent.

Proof It is clear from the definition.

Proposition 4.2.6 (i) If f_2 is a translation/rotation of f_1 then $L(f_2) = L(f_1)$.

(ii) If f_2 is a dilation of f_1 such that $f_2 = cf_1$ then $L(f_2) = cL(f_1)$ where $c > 0$.

Proof. (i) is true for translation because $L(f) = \int_0^1 \|f'(t)\| dt$ for any open curve f (Proposition 4.2.2). Let $f_1 = (g_1, h_1)$ and $f_2 = (g_2, h_2)$ and let $f_2 = f_1 \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}$. That is, $g_2(t) = g_1(t) \cos \alpha - h_1(t) \sin \alpha$ and $h_2(t) = g_1(t) \sin \alpha + h_1(t) \cos \alpha$ for all t in $[0, 1]$. Again from Proposition 4.2.2, $L(f_1) = L(f_2)$, since

$$\left(\frac{dg_2}{dt}\right)^2 + \left(\frac{dh_2}{dt}\right)^2 = \left(\frac{dg_1}{dt}\right)^2 + \left(\frac{dh_1}{dt}\right)^2 \text{ for all (except possibly finitely many) values of } t.$$

(ii) Suppose, $f_2 = cf_1$. So, $g_2(t) = cg_1(t)$ and $h_2(t) = ch_1(t)$ for all t in $[0, 1]$. Again from Proposition 4.2.2, $L(f_2) = cL(f_1)$.

Let $f_1 = (g_1, h_1)$ and $f_2 = (g_2, h_2)$ be two open paths with lengths L_1 and L_2 respectively. Suppose, $x_1(s) = (u_1(s), v_1(s))$ and $x_2(s) = (u_2(s), v_2(s))$ are the two open curves obtained from f_1 and f_2 respectively.

Proposition 4.2.7. If f_2 is a translation of f_1 such that $g_2(t) = g_1(t) + a$ and $h_2(t) = h_1(t) + b$ for all t for some real numbers a and b , then $u_2(s) = u_1(s) + a$ and $v_2(s) = v_1(s) + b$ for all s .

Proof. Note that $L_1 = L_2$. Also, $\int_0^t \|f_1'(t_1)\| dt_1 = \int_0^t \|f_2'(t_1)\| dt_1 = s$, say, for all t . So, from the construction

of u_1, v_1, u_2, v_2 , $(u_2(s), v_2(s)) = (g_2(t), h_2(t)) = (g_1(t), h_1(t)) + (a, b) = (u_1(s), v_1(s)) + (a, b)$ for all s in $[0, L_1]$.

Definition 4.2.9 In this case, we say that the open curve x_2 is a translation of the open curve x_1 .

Proposition 4.2.8 If f_2 is a dilation of f_1 such that $g_2(t) = cg_1(t)$ and $h_2(t) = ch_1(t)$ for all t for some positive real c , then $u_2(s) = cu_1(sL_1/L_2)$ and $v_2(s) = cv_1(sL_1/L_2)$ for all s in $[0, L_2]$.

Proof. Note that $L_2 = cL_1$. Also,

$$\int_0^t \|f_2'(t_1)\| dt_1 = c \int_0^t \|f_1'(t_1)\| dt_1$$

That is, $s_2 = cs_1$ for s_1 in $[0, L_1]$ and s_2 in $[0, L_2]$.

Thus, $(u_2(s_2), v_2(s_2)) = (g_2(t), h_2(t)) = c(g_1(t), h_1(t)) = c(u_1(s_1), v_1(s_1)) = c(u_1(s_2/c), v_1(s_2/c))$ for all $s_2 \in [0, L_2]$.

Definition 4.2.10 Here we say that the open curve x_2 is a dilation of the open curve x_1 .

Proposition 4.2.8.1 If f_2 is a rotation of f_1 such that $g_2(t) = g_1(t)\cos\alpha - h_1(t)\sin\alpha$ and $h_2(t) = g_1(t)\sin\alpha + h_1(t)\cos\alpha$ for all t for some angle α , then $u_2(s) = u_1(s)\cos\alpha - v_1(s)\sin\alpha$ and $v_2(s) = u_1(s)\sin\alpha + v_1(s)\cos\alpha$ for all s .

Proof. Note that $L_1 = L_2$.

Also, $\int_0^t \|f_1'(t_1)\| dt_1 = \int_0^t \|f_2'(t_1)\| dt_1 = s$, say, for all t .

So, $(u_2(s), v_2(s)) = (g_2(t), h_2(t))$
 $= (g_1(t)\cos\alpha - h_1(t)\sin\alpha, g_1(t)\sin\alpha + h_1(t)\cos\alpha)$
 $= (u_1(s)\cos\alpha - v_1(s)\sin\alpha, u_1(s)\sin\alpha + v_1(s)\cos\alpha)$
 for all s in $[0, L_1]$.

Definition 4.2.11 Here we say that the open curve x_2 is a rotation of the open curve x_1 .

Definition 4.2.12 Suppose $\{x_1(s) : s \in [0, L]\}$ is an open curve. Then $x_1(0)$ and $x_1(L)$ are called its extreme points. The former is called the starting point and the latter the end point.

From now on we will denote an open curve A with length L as $\{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$. It can be seen that $\{(u(L-\lambda L), v(L-\lambda L)) : \lambda \in [0, 1]\}$ is also an open curve. This is because if an open path $f(t)$ gives rise to $\{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$ then the open path $f_1(t) = f(1-t)$ gives rise to $\{(u_1(\lambda L), v_1(\lambda L)) : \lambda \in [0, 1]\}$ where $u_1(\lambda L) = u(L-\lambda L)$ and $v_1(\lambda L) = v(L-\lambda L)$ for λ in $[0, 1]$.

Definition 4.2.13 The inverse of the open curve $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$ is defined as $\{(u_1(\lambda L), v_1(\lambda L)) : \lambda \in [0, 1]\}$ and is denoted by A^{-1} .

The starting point of A^{-1} is the end point of A and vice versa. Note that $(A^{-1})^{-1} = A$ and A and A^{-1} have the same length.

Notation $\mathcal{L}$ is the set of all open curves where A and A^{-1} are treated as different elements of $\mathcal{L}$.

Definition 4.2.14 An open curve $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$ is said to be antisymmetric if $u(\lambda L) + u(L - \lambda L) = 2u(L/2)$ and $v(\lambda L) + v(L - \lambda L) = 2v(L/2)$ for all $\lambda \in [0, 1]$.

Definition 4.2.15 S_1 is a relation in $\mathcal{L}$ such that for A and B belonging to $\mathcal{L}$, $(A, B) \in S_1$ if B is a translation of A .

Definition 4.2.16 S_2 is a relation in $\mathcal{L}$ such that for A and B belonging to $\mathcal{L}$, $(A, B) \in S_2$ if B is a dilation of A .

Definition 4.2.17 S_3 is a relation in $\mathcal{L}$ such that for A and B belonging to $\mathcal{L}$, $(A, B) \in S_3$ if B is a rotation of A .

Proposition 4.2.9 S_1, S_2 and S_3 are equivalence relations in $\mathcal{L}$.

Proof. Reflexivity: It holds for S_1 if $a=b=0$, holds for S_2 if $c=1$, holds for S_3 if $\alpha=0$.

Symmetry: Let $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$ and $B = \{(u_1(\lambda L_1), v_1(\lambda L_1)) : \lambda \in [0, 1]\}$.

B is a translation of $A \Rightarrow L = L_1$ and $u_1(\lambda L) = u(\lambda L) + a$ and $v_1(\lambda L) = v(\lambda L) + b$ for all $\lambda \in [0, 1] \Rightarrow u(\lambda L) = u_1(\lambda L) - a$ and $v(\lambda L) = v_1(\lambda L) - b$ for all $\lambda \in [0, 1] \Rightarrow A$ is a translation of B .
So, S_1 is symmetric.

B is a dilation of $A \Rightarrow u_1(\lambda L_1) = u(\lambda L) L_1 / L$, $v_1(\lambda L_1) = v(\lambda L) L_1 / L$ for all $\lambda \in [0, 1]$ from Proposition 4.2.8 $\Rightarrow u(\lambda L) =$

$u_1(\lambda L_1)L/L_1$ and $v(\lambda L)=v_1(\lambda L_1)L/L_1$ for all $\lambda \in [0,1] \Rightarrow A$ is a dilation of B . So, S_2 is symmetric.

B is a rotation of $A \Rightarrow L=L_1$ and $u_1(\lambda L)=u(\lambda L)\cos\alpha-v(\lambda L)\sin\alpha$ and $v_1(\lambda L)=u(\lambda L)\sin\alpha+v(\lambda L)\cos\alpha$ for all $\lambda \in [0,1] \Rightarrow u(\lambda L)=u_1(\lambda L)\cos(2\pi-\alpha)-v_1(\lambda L)\sin(2\pi-\alpha)$ and $v(\lambda L)=u_1(\lambda L)\sin(2\pi-\alpha)+v_1(\lambda L)\cos(2\pi-\alpha) \Rightarrow A$ is a rotation of B . So, S_3 is symmetric.

Transitivity: Let $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0,1]\}$, $B = \{(u_1(\lambda L_1), v_1(\lambda L_1)) : \lambda \in [0,1]\}$ and $C = \{(u_2(\lambda L_2), v_2(\lambda L_2)) : \lambda \in [0,1]\}$

$L=L_1=L_2$ and $u_1(\lambda L)=u(\lambda L)+a_1$, $v_1(\lambda L)=v(\lambda L)+b_1$, $u_2(\lambda L)=u_1(\lambda L)+a_2$ and $v_2(\lambda L)=v_1(\lambda L)+b_2$ for all $\lambda \in [0,1] \Rightarrow u_2(\lambda L)=u(\lambda L)+(a_1+a_2)$ and $v_2(\lambda L)=v(\lambda L)+(b_1+b_2)$ for all $\lambda \in [0,1]$. So, S_1 is transitive.

$u_1(\lambda L_1)=u(\lambda L)L_1/L$, $v_1(\lambda L_1)=v(\lambda L)L_1/L$, $u_2(\lambda L_2)=u_1(\lambda L_1)L_2/L_1$ and $v_2(\lambda L_2)=v_1(\lambda L_1)L_2/L_1$ all $\lambda \in [0,1] \Rightarrow u_2(\lambda L_2)=u(\lambda L)L_2/L$ and $v_2(\lambda L_2)=v(\lambda L)L_2/L$ for all $\lambda \in [0,1]$. So S_2 is transitive.

$L=L_1=L_2$ and $u_1(\lambda L)=u(\lambda L)\cos\alpha - v(\lambda L)\sin\alpha$, $v_1(\lambda L)=u(\lambda L)\sin\alpha+v(\lambda L)\cos\alpha$, $u_2(\lambda L)=u_1(\lambda L)\cos\beta-v_1(\lambda L)\sin\beta$ and $v_2(\lambda L)=u_1(\lambda L)\sin\beta+v_1(\lambda L)\cos\beta$ for all $\lambda \in [0,1] \Rightarrow$
 $u_2(\lambda L)=u(\lambda L)\cos(\alpha+\beta)-v(\lambda L)\sin(\alpha+\beta)$ and $v_2(\lambda L)=u(\lambda L)\sin(\alpha+\beta)+v(\lambda L)\cos(\alpha+\beta)$ for all $\lambda \in [0,1]$. So, S_3 is transitive.

Proposition 4.2.10 If A, B, C belonging to $\mathcal{L}$ are such that $(A, B) \in S_i$ and $(B, C) \in S_j$, then there exists a D in $\mathcal{L}$ such that $(A, D) \in S_j$ and $(D, C) \in S_i$ for $3 \geq j \neq i \geq 1$.

Proof. We shall prove the proposition for $i=1, j=2$.
 the proofs for the other cases will be similar. Let $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$, $B = \{(u_1(\lambda L_1), v_1(\lambda L_1)) : \lambda \in [0, 1]\}$ and $C = \{(u_2(\lambda L_2), v_2(\lambda L_2)) : \lambda \in [0, 1]\}$. Since $(A, B) \in S_1$, $L = L_1$ and $u_1(\lambda L) = u(\lambda L) + a$ and $v_1(\lambda L) = v(\lambda L) + b$ for all $\lambda \in [0, 1]$ for some a and b . Again since $(B, C) \in S_2$, $u_2(\lambda L_2) = u_1(\lambda L) L_2 / L$ and $v_2(\lambda L_2) = v_1(\lambda L) L_2 / L$ for all $\lambda \in [0, 1]$. So, $u_2(\lambda L_2) = u(\lambda L) L_2 / L + a L_2 / L$ and $v_2(\lambda L_2) = v(\lambda L) L_2 / L + b L_2 / L$. Let $D = \{(u_3(\lambda L_2), v_3(\lambda L_2)) : \lambda \in [0, 1]\} = \{(u(\lambda L) L_2 / L, v(\lambda L) L_2 / L) : \lambda \in [0, 1]\}$. Then it is clear that $(A, D) \in S_2$ and $(D, C) \in S_1$.

Definition 4.2.18 S is a relation in $\mathcal{L}$ such that for A, B belonging to $\mathcal{L}$, $(A, B) \in S$ if there exist C and D in $\mathcal{L}$ satisfying $(A, C) \in S_1$, $(C, D) \in S_2$ and $(D, B) \in S_3$.

Proposition 4.2.11 S is an equivalence relation in $\mathcal{L}$.

Proof. Reflexivity: $(A, A) \in S$ since $(A, A) \in S_i$ for $i=1, 2, 3$.

Symmetry: This will be proved by repeated applications of Proposition 4.2.10.

$(A, B) \in S$
 $\Rightarrow (A, C) \in S_1, (C, D) \in S_2$ and $(D, B) \in S_3$ for some C, D in $\mathcal{L}$
 $\Rightarrow (A, C_1) \in S_2, (C_1, D) \in S_1$ and $(D, B) \in S_3$ for some C_1 in $\mathcal{L}$
 $\Rightarrow (A, C_1) \in S_2, (C_1, D_1) \in S_3$ and $(D_1, B) \in S_1$ for some D_1 in $\mathcal{L}$
 $\Rightarrow (A, C_2) \in S_3, (C_2, D_1) \in S_2$ and $(D_1, B) \in S_1$ for some C_2 in $\mathcal{L}$
 $\Rightarrow (B, D_1) \in S_1, (D_1, C_2) \in S_2$ and $(C_2, A) \in S_3$
 $\Rightarrow (B, A) \in S$.

Transitivity: This also will be proved by repeated applications of Proposition 4.2.10.

$$\begin{aligned}
 & (A,B) \in S \text{ and } (B,C) \in S \\
 \Rightarrow & (A,D) \in S_1, (D,E) \in S_2, (E,B) \in S_3 \text{ and } (B,D_1) \in S_1, (D_1,E_1) \in S_2 \\
 & (E_1,C) \in S_3 \text{ for some } D,E,D_1 \text{ and } E_1 \text{ in } \mathcal{L} \\
 \Rightarrow & (A,D) \in S_1, (D,E) \in S_2, (E,B_1) \in S_1, (B_1,D_1) \in S_3, (D_1,E_1) \in S_2, \\
 & (E_1,C) \in S_3 \text{ for some } B_1 \text{ in } \mathcal{L} \\
 \Rightarrow & (A,D) \in S_1, (D,E_1) \in S_1, (E_1,B_1) \in S_2, (B_1,D_2) \in S_2, (D_2,E_1) \in S_3, \\
 & (E_1,C) \in S_3 \text{ for some } E_1,D_2 \text{ in } \mathcal{L} \\
 \Rightarrow & (A,E_1) \in S_1, (E_1,D_2) \in S_2 \text{ and } (D_2,C) \in S_3 \\
 \Rightarrow & (A,C) \in S.
 \end{aligned}$$

Definition 4.2.19 The shape of an open curve is defined to be an equivalence class generated by S in $\mathcal{L}$.

Notation $\mathcal{G}^1$ is the collection of all such equivalence classes, that is, of all shapes.

Within each equivalence class any two open curves can be made ^{to} coincide with each other by translation, dilation and rotation. On the other hand no two open curves from two different equivalence classes can be made ^{to} coincide by translation, dilation and rotation. In other words, the shape of an open curve is what remains of the curve after disregarding its position, size and orientation.

Note that the shapes of A and A^{-1} are not, in general, the same.

Proposition 4.2.12 The shapes of A and A^{-1} are the same if and only if A is antisymmetric.

Proof. Suppose, A and A^{-1} have the same shape. That is, A and A^{-1} belong to the same equivalence class generated by S . Since A and A^{-1} have the same length, A^{-1} is a translation/rotation of A . If $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$ and $A^{-1} = \{(u_1(\lambda L), v_1(\lambda L)) : \lambda \in [0, 1]\}$, $u_1(\lambda L) = u(\lambda L)\cos\alpha - v(\lambda L)\sin\alpha + a$ and $v_1(\lambda L) = u(\lambda L)\sin\alpha + v(\lambda L)\cos\alpha + b$ for all $\lambda \in [0, 1]$ for some α, a and b . Putting $\lambda=0$, $a = u_1(0) - u(0)\cos\alpha + v(0)\sin\alpha$ and $b = v_1(0) - u(0)\sin\alpha - v(0)\cos\alpha$. Then putting $\lambda=1$ and substituting a and b ,

$$u_1(L) - u_1(0) = (u(L) - u(0))\cos\alpha - (v(L) - v(0))\sin\alpha$$

$$v_1(L) - v_1(0) = (u(L) - u(0))\sin\alpha + (v(L) - v(0))\cos\alpha$$

Since $u_1(L) = u(0)$, $v_1(L) = v(0)$, $u_1(0) = u(L)$ and $v_1(0) = v(L)$,

$\alpha = \pi$ from Appendix B.1. So, $a = u(L) + u(0)$ and

$b = v(L) + v(0)$. Thus, $u_1(\lambda L) = -u(\lambda L) + u(L) + u(0)$ and

$v_1(\lambda L) = -v(\lambda L) + v(L) + v(0)$. Again, putting $\lambda=1/2$, $u(L) + u(0) =$

$$u_1(L/2) + u(L/2) = u(L - L/2) + u(L/2) = 2u(L/2).$$

Similarly, $v(L) + v(0) = 2v(L/2)$. So, $u(\lambda L) + u(L - \lambda L) = 2u(L/2)$

and $v(\lambda L) + v(L - \lambda L) = 2v(L/2)$ for all $\lambda \in [0, 1]$. Hence A is

antisymmetric.

Conversely, suppose A is antisymmetric. So, $u(\lambda L) + u(L-\lambda L) = 2u(L/2)$ and $v(\lambda L) + v(L-\lambda L) = 2v(L/2)$ for all $\lambda \in [0,1]$. That is, $-u(\lambda L) = u(L-\lambda L) - 2u(L/2)$ and $-v(\lambda L) = v(L-\lambda L) - 2v(L/2)$. So, 180° rotation of $A = A_\pi = \{(-u(\lambda L), -v(\lambda L)) : \lambda \in [0,1]\}$ is a translation of $A^{-1} = \{(u(L-\lambda L), v(L-\lambda L)) : \lambda \in [0,1]\}$. Thus A and A^{-1} can be made to coincide with each other through rotation and translation. Hence, A and A^{-1} have the same shape.

4.3 Two Measures of Shape Similarity for Open Curves

We shall now propose two methods for quantifying shape distance and shape similarity for open curves.

Method 1

In the first method we find the tangent at every point on an open curve and then find its corresponding directional code d in S_8 (Parui and Dutta Majumder, 1983C).

Let $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0,1]\}$ be an open curve.

For each λ belonging to $(0,1)$ we define the following :

$$\begin{aligned} e(\lambda, \varepsilon) &= 1 \text{ if } u(\lambda L) - u(\lambda L - \varepsilon) > 0 \\ &= 0 \text{ if } u(\lambda L) - u(\lambda L - \varepsilon) = 0 & \varepsilon > 0 \\ &= -1 \text{ if } u(\lambda L) - u(\lambda L - \varepsilon) < 0 \end{aligned}$$

$$\begin{aligned} \delta(\lambda, \varepsilon) &= 1 \text{ if } u(\lambda L) - u(\lambda L + \varepsilon) > 0 \\ &= 0 \text{ if } u(\lambda L) - u(\lambda L + \varepsilon) = 0 & \varepsilon > 0 \\ &= -1 \text{ if } u(\lambda L) - u(\lambda L + \varepsilon) < 0 \end{aligned}$$

$$e(\lambda) = \lim_{\varepsilon \downarrow 0} e(\lambda, \varepsilon) \quad \text{and} \quad \delta(\lambda) = \lim_{\varepsilon \downarrow 0} \delta(\lambda, \varepsilon)$$

We assume here that $e(\lambda)$ exists for all λ in $(0,1)$.

Definition 4.3.1 The λ -th point $(u(\lambda L), v(\lambda L))$ on the open curve A is a corner point if

$$(i) \quad \lim_{\varepsilon \downarrow 0} \frac{v(\lambda L) - v(\lambda L - \varepsilon)}{u(\lambda L) - u(\lambda L - \varepsilon)} \neq \lim_{\varepsilon \downarrow 0} \frac{v(\lambda L) - v(\lambda L + \varepsilon)}{u(\lambda L) - u(\lambda L + \varepsilon)}$$

and (ii) either the left hand side or the right hand side in (i) is finite.

Proposition 4.3.1 An open curve A has at most finitely many corner points.

Proof. See Appendix B.2.

For the slope at the λ -th point $(u(\lambda L), v(\lambda L))$, only the left limit will now be taken because the left and the right limits are not always the same.

$$\text{Let } f(\lambda) = \lim_{\varepsilon \downarrow 0} \frac{v(\lambda L) - v(\lambda L - \varepsilon)}{u(\lambda L) - u(\lambda L - \varepsilon)} \quad \text{for } \lambda \in (0, 1)$$

$$\text{Let } g(\lambda) = \frac{\pi/2 - \tan^{-1} f(\lambda)}{\pi/4} \quad \text{when } |f(\lambda)| < \infty$$

$$\text{Thus, } 4 > g(\lambda) > 0 \quad \text{since } -\pi/2 < \tan^{-1} f(\lambda) < \pi/2$$

$$\begin{aligned} \text{Define } w(\lambda) &= g(\lambda) & \text{if } e(\lambda) = 1 & \text{ and } |f(\lambda)| < \infty \\ &= g(\lambda) + 4 & \text{if } e(\lambda) = -1 & \text{ and } |f(\lambda)| < \infty \\ &= 4 & \text{if } f(\lambda) = -\infty \\ &= 8 & \text{if } f(\lambda) = +\infty \end{aligned}$$

Note that the value of $w(\lambda)$ belongs to S_8 and $w(\lambda)$ gives the directional code of the (left) tangent at $(u(\lambda L), v(\lambda L))$.

Definition 4.3.2 $\{w(\lambda) : \lambda \in (0, 1)\}$ is called the slope of the open curve $\{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$.

Proposition 4.3.2 (i) If an open curve B is a translation and dilation of an open curve A and the slopes of A and B are $\{w(\lambda) : \lambda \in (0,1)\}$ and $\{w_1(\lambda) : \lambda \in (0,1)\}$, then $w(\lambda) = w_1(\lambda)$ for all $\lambda \in (0,1)$.

(ii) If $w(\lambda) = w_1(\lambda)$ for all $\lambda \in (0,1)$, then B is a translation and dilation of A.

Proof. (i) Let $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0,1]\}$ and $B = \{(u_1(\lambda L_1), v_1(\lambda L_1)) : \lambda \in [0,1]\}$. If B is a translation of A, then $L = L_1$ and $\frac{v_1(\lambda L) - v_1(\lambda L - h)}{u_1(\lambda L) - u_1(\lambda L - h)} = \frac{v(\lambda L) - v(\lambda L - h)}{u(\lambda L) - u(\lambda L - h)}$ for all $\lambda \in (0,1)$ and arbitrarily small $h > 0$, from Proposition 4.2.6 and Proposition 4.2.7. Hence the value of $f(\lambda)$ will be the same in both the cases. Also, A and B have the same $e(\lambda)$. So, A and B have the same slope.

If B is a dilation of A,

$$\frac{v_1(\lambda L_1) - v_1(\lambda L_1 - h)}{u_1(\lambda L_1) - u_1(\lambda L_1 - h)} = \frac{v(\lambda L) - v(\lambda L - Lh/L_1)}{u(\lambda L) - u(\lambda L - Lh/L_1)}$$

for all $\lambda \in (0,1)$ and arbitrarily small $h > 0$, from Proposition 4.2.8. Hence here also $f(\lambda)$ will have the same value for A and B. Again $e(\lambda)$ is the same in both the cases. So, A and B have the same slope.

(ii) The proof for (ii) is clear from Appendix B.3.

Notation A_α denotes the open curve obtained after rotating an open curve A by an angle α in the anticlockwise direction. Clearly, $(A_\alpha)_\beta = A_{\alpha+\beta}$ and $(A^{-1})_\alpha = (A_\alpha)^{-1}$.

Proposition 4.3.3 If an open curve B is a rotation of an open curve A and the slopes of A and B are $\{w(\lambda) : \lambda \in (0,1)\}$ and $\{w_1(\lambda) : \lambda \in (0,1)\}$, then $w_1(\lambda) = w(\lambda) + k \pmod{8}$ for all $\lambda \in (0,1)$ and for some constant k. In fact, if $B = A_\alpha$, then $k = -4\alpha/\pi$.

Proof. Suppose $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0,1]\}$ and
 $B = \{(u_1(\lambda L), v_1(\lambda L)) : \lambda \in [0,1]\}$

Since $B = A_\alpha$, $u_1(\lambda L) = u(\lambda L)\cos\alpha - v(\lambda L)\sin\alpha$

and $v_1(\lambda L) = u(\lambda L)\sin\alpha + v(\lambda L)\cos\alpha$ for all λ .

$$\text{So, } \frac{v_1(\lambda L) - v_1(\lambda L - h)}{u_1(\lambda L) - u_1(\lambda L - h)} = \frac{\sin\alpha \{u(\lambda L) - u(\lambda L - h)\} + \cos\alpha \{v(\lambda L) - v(\lambda L - h)\}}{\cos\alpha \{u(\lambda L) - u(\lambda L - h)\} - \sin\alpha \{v(\lambda L) - v(\lambda L - h)\}}$$

Case I $e(\lambda) \neq 0$ and $\alpha \neq \pi/2$ i.e. $u(\lambda L) \neq u(\lambda L - h)$ for arbitrarily small $h > 0$. By dividing the above expression by $\cos\alpha \{u(\lambda L) - u(\lambda L - h)\}$ and by taking limit we get

$$\begin{aligned} f_1(\lambda) &= \lim_{h \downarrow 0} \left\{ \frac{\tan\alpha + \frac{v(\lambda L) - v(\lambda L - h)}{u(\lambda L) - u(\lambda L - h)}}{1 - \tan\alpha \frac{v(\lambda L) - v(\lambda L - h)}{u(\lambda L) - u(\lambda L - h)}} \right\} \\ &= \frac{\lim_{h \downarrow 0} \left\{ \tan\alpha + \frac{v(\lambda L) - v(\lambda L - h)}{u(\lambda L) - u(\lambda L - h)} \right\}}{\lim_{h \downarrow 0} \left\{ 1 - \tan\alpha \frac{v(\lambda L) - v(\lambda L - h)}{u(\lambda L) - u(\lambda L - h)} \right\}} \\ &= \frac{\tan\alpha + f(\lambda)}{1 - \tan\alpha f(\lambda)} \end{aligned}$$

So, $\tan^{-1} f_1(\lambda) = \tan^{-1} f(\lambda) + \alpha \pmod{\pi}$ belongs to $[-\pi/2, \pi/2]$.

$$\text{That is, } g_1(\lambda) = \frac{\pi/2 - \tan^{-1} f(\lambda) - \alpha}{\pi/4} = g(\lambda) - 4\alpha/\pi$$

So, from the definition, $w_1(\lambda) = w(\lambda) - 4\alpha/\pi \pmod{8}$.

Case II $e(\lambda)=0$. So, for all arbitrarily small $h>0$,

$$u(\lambda L) = u(\lambda L-h).$$

$$\text{Thus, } \frac{v_1(\lambda L) - v_1(\lambda L-h)}{u_1(\lambda L) - u_1(\lambda L-h)} = -\cot\alpha = \tan(-\pi/2+\alpha)$$

So, $f_1(\lambda) = \tan(-\pi/2+\alpha)$. That is, $g_1(\lambda) = 4-4\alpha/\pi$ if $\alpha \in (0, \pi)$ and $8-4\alpha/\pi$ if $\alpha \in [\pi, 2\pi)$. $u_1(\lambda L) - u_1(\lambda L-h) = -\sin\alpha (v(\lambda L) - v(\lambda L-h))$

Subcase 1 $f(\lambda) = -\infty$ i.e. $w(\lambda) = 4$

Note $v(\lambda L) - v(\lambda L-h) < 0$ for all arbitrarily small $h>0$. If $\alpha \in (0, \pi)$, $u_1(\lambda L) - u_1(\lambda L-h) = -\sin\alpha (v(\lambda L) - v(\lambda L-h)) > 0$ for all arbitrarily small $h>0$. So, $e_1(\lambda) = 1$. Thus $w_1(\lambda) = g_1(\lambda) = 4-4\alpha/\pi$. Hence, $w_1(\lambda) = w(\lambda) - 4\alpha/\pi$.

If $\alpha = \pi$, $u_1(\lambda L) - u_1(\lambda L-h) = 0$ i.e. $e_1(\lambda) = 0$.

Now, $v_1(\lambda L) - v_1(\lambda L-h) = \cos\alpha (v(\lambda L) - v(\lambda L-h)) > 0$ for all arbitrarily small $h>0$. So, $f_1(\lambda) = +\infty$.

That is, $w_1(\lambda) = 8 = w(\lambda) - 4\alpha/\pi \pmod{8}$.

If $\alpha \in (\pi, 2\pi)$, $u_1(\lambda L) - u_1(\lambda L-h) < 0$ for all arbitrarily small $h>0$.

So, $e_1(\lambda) = -1$. Thus $w_1(\lambda) = g_1(\lambda) + 4 = 8 - 4\alpha/\pi + 4$
 $= w(\lambda) - 4\alpha/\pi \pmod{8}$.

Subcase 2 $f(\lambda) = +\infty$ i.e. $w(\lambda) = 8$. The proof here will be similar to the one in Subcase 1.

Case III $e(\lambda) \neq 0$ and $\alpha = \pi/2$. The proof will be similar to the one in Case II.

Proposition 4.3.4 The slope of an open curve depends on the orientation but not on the position or size of the curve.

Proof. It is clear from Proposition 4.3.2 and Proposition 4.3.3.

Proposition 4.3.5 If the slopes of an open curve A and its inverse A^{-1} are $\{w(\lambda) : \lambda \in (0,1)\}$ and $\{w_1(\lambda) : \lambda \in (0,1)\}$ respectively, then $w_1(1-\lambda) = w(\lambda) + 4 \pmod{8}$ for all $\lambda \in (0,1)$ unless the λ -th point of A is a corner point.

Proof. In Appendix B.4.

Notation $\oplus$ will denote addition modulo 8.

Definition 4.3.3 A distance function on a set S is a non-negative real-valued function defined on $S \times S$.

Definition 4.3.4 D_1 is a distance function on S_8 such that for d_1 and d_2 belonging to S_8

$$D_1(d_1, d_2) = \text{Min} \{ |d_1 - d_2|, 8 - |d_1 - d_2| \} \quad (4.1)$$

In fact, $D_1(d_1, d_2)\pi/4$ gives the angle between the directions d_1 and d_2 .

Proposition 4.3.6 D_1 is a metric on S_8 .

Proof. It can be easily seen that D_1 satisfies all the four metric properties, i.e. (i) $D_1(d_1, d_2) \geq 0$

$$(ii) \quad D_1(d_1, d_2) = 0 \text{ if and only if } d_1 = d_2$$

$$(iii) \quad D_1(d_1, d_2) = D_1(d_2, d_1)$$

$$(iv) \quad D_1(d_1, d_2) + D_1(d_2, d_3) \geq D_1(d_1, d_3)$$

Definition 4.3.5 D_2 is a distance function on $\mathcal{L}$ such that for two open curves A and B in $\mathcal{L}$,

$$D_2(A, B) = \int_0^1 D_1(w_1(\lambda), w_2(\lambda)) d\lambda \quad (4.2)$$

where $\{w_1(\lambda) : \lambda \in (0, 1)\}$ and $\{w_2(\lambda) : \lambda \in (0, 1)\}$ are the slopes of A and B respectively.

$$\text{Proposition 4.3.7} \quad D_2(A, B) = D_2(A^{-1}, B^{-1})$$

Proof. It is clear from Proposition 4.3.1 and Proposition 4.3.5.

Proposition 4.3.8 For A, B and C belonging to $\mathcal{L}$,

$$(i) \quad D_2(A, B) \geq 0$$

$$(ii) \quad D_2(A, B) = 0 \text{ if and only if } B \text{ is a translation and dilation of } A.$$

$$(iii) \quad D_2(A, B) = D_2(B, A)$$

$$(iv) \quad D_2(A, B) + D_2(B, C) \geq D_2(A, C)$$

Proof. (i) Trivial.

(ii) Suppose B is a translation and dilation of A .

So, from Proposition 4.3.2, $w_1(\lambda) = w_2(\lambda)$ for all $\lambda \in (0, 1)$.

Thus $D_2(A, B) = 0$. The proof of the converse is in Appendix B.5.

(iii) Trivial.

(iv) This is true since $D_1(d_1, d_2) + D_1(d_2, d_3) \geq D_1(d_1, d_3)$ for any d_1, d_2 and d_3 belonging to S_8 .

Proposition 4.3.9 $D_2(A, B_\alpha) = D_2(A, C_\alpha)$ if C is a translation and dilation of B for $\alpha \in [0, 2\pi)$.

Proof. It is clear from Proposition 4.3.2 and Proposition 4.3.3 and also from the fact that $D_1(d_1, d_2) = D_1(d_3, d_4)$ implies $D_1(d_1, d_2 \oplus k) = D_1(d_3, d_4 \oplus k)$ for any k .

Proposition 4.3.10 $D_2(A, B) = D_2(A_\alpha, B_\alpha)$ for any $\alpha \in [0, 2\pi)$.

Proof. This is true because of (i) $D_1(d_1, d_2) = D_1(d_1 \div k, d_2 \div k)$ for any k and (ii) Proposition 4.3.3.

Definition 4.3.6 D_3 is a distance function on $\mathcal{L}$ such that for A, B belonging to $\mathcal{L}$

$$D_3(A, B) = \inf_{\alpha \in [0, 2\pi)} D_2(A, B_\alpha) \quad (4.3)$$

Proposition 4.3.11 For A, B belonging to $\mathcal{L}$, $D_3(A, B) = D_2(A, B_\beta)$ for some $\beta \in [0, 2\pi)$.

Proof. In Appendix B.6.

Proposition 4.3.12 For A, B and C belonging to $\mathcal{L}$

- (i) $D_3(A, B) \geq 0$
- (ii) $D_3(A, B) = 0$ if and only if B is a translation, dilation and rotation of A .
- (iii) $D_3(A, B) = D_3(B, A)$
- (iv) $D_3(A, B) + D_3(B, C) \geq D_3(A, C)$

Proof. (i) Trivial.

(ii) Suppose $D_3(A, B) = 0$. Now from

Proposition 4.3.11, there is a $\beta \in [0, 2\pi)$ such that $D_3(A, B) = D_2(A, B_\beta)$. Again from Proposition 4.3.8, $D_2(A, B_\beta) = 0$ implies B_β is a translation and dilation of A . Thus B is a translation, dilation and rotation of A .

Conversely, suppose B is a translation, dilation and rotation of A . That is, a rotation of B , say, B_β is a translation and dilation of A . So, $D_2(A, B_\beta) = 0$.

$$\text{Hence } D_3(A, B) = \inf_{\alpha} D_2(A, B_\alpha) = 0.$$

$$\begin{aligned} \text{(iii) } D_3(A, B) &= \inf_{\alpha} D_2(A, B_\alpha) \\ &= \inf_{\alpha} D_2(A_{2\pi-\alpha}, B_{2\pi}) \text{ from Proposition 4.3.10} \\ &= \inf_{\beta} D_2(A_\beta, B) \\ &= \inf_{\beta} D_2(B, A_\beta) \text{ since } D_2 \text{ is symmetric} \\ &= D_3(B, A). \end{aligned}$$

$$\text{(iv) } D_3(A, B) = D_2(A, B_\beta) \text{ for some } \beta \text{ and } D_3(B, C) = D_2(B, C_\gamma) \text{ for some } \gamma \text{ from Proposition 4.3.11.}$$

$$\begin{aligned} \text{So, } D_3(A, B) + D_3(B, C) &= D_2(A, B_\beta) + D_2(B, C_\gamma) \\ &= D_2(A, B_\beta) + D_2(B_\beta, C_{\gamma+\beta}) \text{ from Proposition 4.3.10} \\ &\geq D_2(A, C_{\gamma+\beta}) \text{ since } D_2 \text{ satisfies the triangular inequality} \\ &\geq \inf_{\alpha} D_2(A, C_\alpha) \\ &= D_3(A, C). \end{aligned}$$

Proposition 4.3.13 $D_3(A, B_\gamma) = D_3(A, B)$ for any γ .

$$\begin{aligned} \text{Proof. } D_3(A, B_\gamma) &= \inf_{\beta} D_2(A, B_{\gamma+\beta}) = \inf_{\alpha} D_2(A, B_\alpha) \\ &= D_3(A, B) \end{aligned}$$

Proposition 4.3.14 $D_3(A, B) = D_3(A, C)$ if C is a translation and dilation of B .

Proof. It is clear from Proposition 4.3.9.

Proposition 4.3.15 $D_3(A^{-1}, B^{-1}) = D_3(A, B)$.

Proof. $D_3(A^{-1}, B^{-1}) = \inf_{\alpha} D_2(A^{-1}, (B^{-1})_{\alpha})$
 $= \inf_{\alpha} D_2(A^{-1}, (B_{\alpha})^{-1})$
 $= \inf_{\alpha} D_2(A, B_{\alpha})$ from Proposition 4.3.7
 $= \alpha \text{ R.H.S.}$

Proposition 4.3.16 D_3 defines a metric on $\mathcal{S}'$, that is, on shapes.

Proof. First we will prove unambiguity. That is, if B and C have the same shape, we have to show $D_3(A, B) = D_3(A, C)$.

Since B and C have the same shape, C is a translation, dilation and rotation of B . That is, a rotation of B , say B_{γ} , is a translation and dilation of C . From Proposition 4.3.14, $D_3(A, B_{\gamma}) = D_3(A, C)$. Now, $D_3(A, B_{\gamma}) = D_3(A, B)$ from Proposition 4.3.13. Thus, $D_3(A, B) = D_3(A, C)$.

Also, $D_3(A, B) = 0$ if and only if A and B have the same shape (from (ii) in Proposition 4.3.12).

Proposition 4.3.17 $\sup_{A, B \in \mathcal{S}'} D_3(A, B) = 2$.

Proof. Let A_1 and C_n be two open curves with length 1 such that A_1 is a straight line segment and C_n is an incomplete (though nearly complete, Fig.4.2) circle. It can

be seen that the slope of A_1 , $w_1(s)$, is a fixed constant, say, B and the slope of C_n is $w_2(s) = (1-1/n)8s$ for $s \in (0,1)$ for a fixed integer n . C_n is asymptotically a circle (i.e. C_n tends to a circle as n goes to infinity) which strictly speaking is not in $\mathcal{L}$.

$$\begin{aligned} \text{So, } D_2(A_1, C_n) &= \int_0^1 D_1(w_1(\lambda), w_2(\lambda)) d\lambda \\ &= \int_0^1 D_1(8, (1-1/n)8\lambda) d\lambda \\ &= \int_0^{n/2(n-1)} (1-1/n)8\lambda d\lambda + \int_{n/2(n-1)}^1 (8 - (1-1/n)8\lambda) d\lambda \\ &= \frac{2n}{n-1} + \frac{4(n-2)}{n-1} - \frac{4(n-1)}{n} \end{aligned}$$

It is easy to see that (i) $\lim_{n \rightarrow \infty} D_3(A_1, C_n) = \lim_{n \rightarrow \infty} D_2(A_1, C_n)$

and (ii) $\lim_{n \rightarrow \infty} D_2(A_1, C_n) = 2$. So, $\sup_{A, B} D_3(A, B) \geq$

$$\lim_{n \rightarrow \infty} D_3(A_1, C_n) = 2.$$

Now, we shall show that $\sup_{A, B} D_3(A, B) \leq 2$. First note that

$D_1(d_1, d_2) + D_1(d_1, d_2 + 4) = 4$ for any d_1 and d_2 . Suppose the slopes of A and B are $\{w_1(\lambda) : \lambda \in (0,1)\}$ and $\{w_2(\lambda) : \lambda \in (0,1)\}$ respectively. So, the slope of B_π is $\{w_2(\lambda) + 4 : \lambda \in (0,1)\}$ from Proposition 4.3.3.

Thus $D_2(A, B) + D_2(A, B_\pi)$

$$\begin{aligned} &= \int_0^1 D_1(w_1(\lambda), w_2(\lambda)) d\lambda + \int_0^1 D_1(w_1(\lambda), w_2(\lambda) + 4) d\lambda \\ &= \int_0^1 4 d\lambda = 4. \end{aligned}$$

So, $D_3(A, B) \leq \min \{D_2(A, B), D_2(A, B^{-1})\} \leq 2$ for all A and B .

That is, $\sup_{A, B} D_3(A, B) \leq 2$. Hence $\sup_{A, B} D_3(A, B) = 2$.

Proposition 4.3.18 $D_3(A, A^{-1}) = 0$ if and only if A is antisymmetric.

Proof. It is clear from Proposition 4.2.12 and Proposition 4.3.12.

So far we have distinguished between the shapes of A and A^{-1} . That is, their shapes have not been the same in general. But in many cases it may be required to disregard which of the two extreme points we scan an open curve from, when studying its shape. In other words, it may be required to consider the shapes of A and A^{-1} to be the same. For that purpose we now modify D_3 .

Definition 4.3.7 D_4 is a distance function, on $\mathcal{L}$ such that for A, B belonging to $\mathcal{L}$,

$$D_4(A, B) = \min \{D_3(A, B), D_3(A, B^{-1})\} \quad (4.4)$$

The slope of B^{-1} is taken to be $\{w(1-\lambda) \oplus 4 : \lambda \in (0, 1)\}$ where $\{w(\lambda) : \lambda \in (0, 1)\}$ is the slope of B , because the slope of B^{-1} differs from $\{w(1-\lambda) \oplus 4 : \lambda \in (0, 1)\}$ at most at finitely many points (i.e. at the corner points).

The definition of D_4 is unambiguous in the sense that $D_4(A, B) = D_4(B, A)$. This can be seen from the Proposition 4.3.15.

Proposition 4.3.19 $D_4(A, A^{-1}) = 0$

Proof. Trivial.

Proposition 4.3.20

$$\sup_{A, B \in \mathcal{L}} D_4(A, B) = 2$$

Proof. It is clear from Proposition 4.3.17.

Definition 4.3.8

μ_1 is a similarity measure

(based on D_4) between two shapes such that

$$\mu_1(A, B) = 1 - D_4(A, B)/2$$

(Similarity measure can be defined on the basis of D_3 also).

Evidently, $1 \geq \mu_1 \geq 0$.

Note that $\mu_1(A, B) = 1$ if and only if B is a translation, dilation, rotation and/or inverse of A , that is, if and only if A and B have the same shape or A^{-1} and B have the same shape.

Method 2 (Parui and Dutta Majumder, 1983C)

In this method we find the Euclidean distance between every two corresponding points on two open curves. Since we are going to define a distance between shapes and shapes do not depend on the position and size of an open curve, we will consider only the open curves with the centroid* at the origin $(0, 0)$ and with a fixed length, say, L . Let $\mathcal{L}_1$ be the set of all such open curves. Let A and B be two open curves belonging to $\mathcal{L}_1$ such that $A = \{(u_1(\lambda L), v_1(\lambda L)) : \lambda \in [0, 1]\}$ and $B = \{(u_2(\lambda L), v_2(\lambda L)) : \lambda \in [0, 1]\}$.

Definition 4.3.9

D_5 is a distance function on $\mathcal{L}_1$ such that for A, B belonging to $\mathcal{L}_1$

* The centroid of an open curve $\{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$ is $(\int_0^1 u(\lambda L) d\lambda, \int_0^1 v(\lambda L) d\lambda)$.

$$D_5(A, B) = \frac{1}{L} \int_0^L \left\{ (u_1(\lambda L) - u_2(\lambda L))^2 + (v_1(\lambda L) - v_2(\lambda L))^2 \right\} d\lambda \quad (4.5)$$

Proposition 4.3.21 For A, B belonging to $\mathcal{L}_1$,

$$D_5(A, B) = D_5(A_\alpha, B_\alpha) \text{ for any } \alpha \in [0, 2\pi).$$

Proof. Let $A = \{(u_1(\lambda L), v_1(\lambda L)) : \lambda \in [0, 1]\}$
and $B = \{(u_2(\lambda L), v_2(\lambda L)) : \lambda \in [0, 1]\}$

$$\text{So, } A_\alpha = \{(u_1(\lambda L) \cos \alpha - v_1(\lambda L) \sin \alpha, u_1(\lambda L) \sin \alpha + v_1(\lambda L) \cos \alpha) : \lambda \in [0, 1]\}$$

$$B_\alpha = \{(u_2(\lambda L) \cos \alpha - v_2(\lambda L) \sin \alpha, u_2(\lambda L) \sin \alpha + v_2(\lambda L) \cos \alpha) : \lambda \in [0, 1]\}$$

$$\text{Thus, } D_5^2(A_\alpha, B_\alpha)$$

$$= \frac{1}{L^2} \int_0^L \left\{ (u_1(\lambda L) - u_2(\lambda L))^2 \cos^2 \alpha + (v_1(\lambda L) - v_2(\lambda L))^2 \sin^2 \alpha + \right. \\ \left. (u_1(\lambda L) - u_2(\lambda L))^2 \sin^2 \alpha + (v_1(\lambda L) - v_2(\lambda L))^2 \cos^2 \alpha \right\} d\lambda$$

$$= \frac{1}{L^2} \int_0^L \left\{ (u_1(\lambda L) - u_2(\lambda L))^2 + (v_1(\lambda L) - v_2(\lambda L))^2 \right\} d\lambda$$

$$= D_5^2(A, B)$$

Proposition 4.3.22 $D_5(A, B) = D_5(A^{-1}, B^{-1})$.

$$\begin{aligned} \text{Proof. } D_5^2(A^{-1}, B^{-1}) &= \frac{1}{L^2} \int_0^L \left\{ (u_1(L - \lambda L) - u_2(L - \lambda L))^2 \right. \\ &\quad \left. + (v_1(L - \lambda L) - v_2(L - \lambda L))^2 \right\} d\lambda \\ &= \frac{1}{L^2} \int_0^L \left\{ (u_1(\lambda L) - u_2(\lambda L))^2 + \right. \\ &\quad \left. (v_1(\lambda L) - v_2(\lambda L))^2 \right\} d\lambda \\ &= D_5^2(A, B) \end{aligned}$$

Proposition 4.3.23 D_5 is a metric on $\mathcal{L}_1$, that is,

for A, B, C belonging to $\mathcal{L}_1$,

$$(i) \quad D_5(A, B) \geq 0$$

$$(ii) \quad D_5(A, B) = 0 \text{ if and only if } A = B$$

$$(iii) \quad D_5(A, B) = D_5(B, A)$$

$$(iv) \quad D_5(A, B) + D_5(B, C) \geq D_5(A, C)$$

Proof. (i) Trivial.

(ii) Let $A = \{(u_1(\lambda L), v_1(\lambda L)) : \lambda \in [0, 1]\}$ and $B = \{(u_2(\lambda L), v_2(\lambda L)) : \lambda \in [0, 1]\}$. If $A=B$, that is, $u_1(\lambda L) = u_2(\lambda L)$ and $v_1(\lambda L) = v_2(\lambda L)$ for all $\lambda \in [0, 1]$, then $D_5(A, B) = 0$.

Conversely, suppose $D_5(A, B) = 0$.

That is, $\int_0^1 \{(u_1(\lambda L) - u_2(\lambda L))^2 + (v_1(\lambda L) - v_2(\lambda L))^2\} d\lambda = 0$.

But since u_1, v_1, u_2 and v_2 are continuous functions,

$(u_1(\lambda L) - u_2(\lambda L))^2 + (v_1(\lambda L) - v_2(\lambda L))^2 = 0$ for all $\lambda \in [0, 1]$. Thus, $A=B$.

(iii) Trivial.

(iv) This is true because D_5 is nothing but the L_2 norm (Apostol, 1974).

Definition 4.3.10 D_6 is a distance function on $\mathcal{L}_1$ such that for A, B belonging to $\mathcal{L}_1$,

$$D_6(A, B) = \inf_{\alpha \in [0, 2\pi)} D_5(A, B_\alpha) \quad (4.6)$$

Proposition 4.3.24 For A, B belonging to $\mathcal{L}_1$, $D_6(A, B) = D_5(A, B_\beta)$ for some $\beta \in [0, 2\pi)$.

Proof. It is clear from Appendix B.6.

Proposition 4.3.25 $D_6(A^{-1}, B^{-1}) = D_6(A, B)$

$$\begin{aligned} \text{Proof. } D_6(A^{-1}, B^{-1}) &= \inf_{\alpha} D_5(A^{-1}, (B^{-1})_\alpha) \\ &= \inf_{\alpha} D_5(A^{-1}, (B_\alpha)^{-1}) \\ &= \inf_{\alpha} D_5(A, B_\alpha) \text{ from Proposition 4.3.22} \\ &= D_6(A, B) \end{aligned}$$

Proposition 4.3.26 For A, B, C belonging to $\mathcal{L}_1$,

- (i) $D_6(A, B) \geq 0$
- (ii) $D_6(A, B) = 0$ if and only if B is a rotation of A
- (iii) $D_6(A, B) = D_6(B, A)$
- (iv) $D_6(A, B) + D_6(B, C) \geq D_6(A, C)$

Proof. (i) Trivial.

(ii) Suppose $D_6(A, B) = 0$. So, there is a $\beta \in [0, 2\pi)$ such that $D_5(A, B_\beta) = 0$. Then from (ii) of Proposition 4.3.23, $A = B_\beta$. Hence B is a rotation of A .

Conversely, suppose B is a rotation of A . So, there is a $\beta \in [0, 2\pi)$ such that $A = B_\beta$. Thus $D_5(A, B_\beta) = 0$. Hence $D_6(A, B) = 0$.

(iii) It is clear from (iii) in Proposition 4.3.23.

$$\begin{aligned}
 \text{(iv)} \quad D_6(A, B) + D_6(B, C) &= D_5(A, B_\beta) + D_5(B, C_\gamma) \text{ for some } \beta \text{ and } \gamma \\
 &= D_5(A, B_\beta) + D_5(B_\beta, C_{\gamma+\beta}) \text{ from Proposition 4.3.21} \\
 &\geq D_5(A, C_{\gamma+\beta}) \text{ since } D_5 \text{ satisfies the triangular} \\
 &\quad \text{inequality} \\
 &\geq D_6(A, C).
 \end{aligned}$$

It can be seen that two open curves A and B belonging to $\mathcal{L}_1$ have the same shape if and only if B is a rotation of A . Hence

Proposition 4.3.27 D_6 defines a metric on $\mathcal{S}^1$, that is, on shapes of open curves.

To prove unambiguity, we have to show, for A, B, C

belonging to $\mathcal{L}_1$, $D_6(A, B) = D_6(A, C)$ if C is a rotation of B (that is, if B and C have the same shape). Suppose, $C = B_\beta$.

$$D_6(A, B) = \inf_{\alpha} D_5(A, B_\alpha) = \inf_{\alpha} D_5(A, B_{\beta+\alpha}) = \inf_{\alpha} D_5(A, C_\alpha) = D_6(A, C)$$

From (ii) in Proposition 4.3.26, it is clear that $D_6(A, B) = 0$ if and only if A and B have the same shape.

Proposition 4.3.28 $\sup_{A, B \in \mathcal{L}_1} D_6(A, B) \leq 1/\sqrt{6}$.

Proof. Let $A = \{(u_1(\lambda L), v_1(\lambda L)) : \lambda \in [0, 1]\}$
and $B = \{(u_2(\lambda L), v_2(\lambda L)) : \lambda \in [0, 1]\}$

$$\text{So, } B_\pi = \{(-u_2(\lambda L), v_2(\lambda L)) : \lambda \in [0, 1]\}$$

$$\begin{aligned} \text{Now, } & \{D_5(A, B) + D_5(A, B_\pi)\}^2 \\ &= \frac{1}{L^2} \int_0^1 \{(u_1(\lambda L) - u_2(\lambda L))^2 + (v_1(\lambda L) - v_2(\lambda L))^2\} d\lambda \\ &\quad + \frac{1}{L^2} \int_0^1 \{(u_1(\lambda L) + u_2(\lambda L))^2 + (v_1(\lambda L) + v_2(\lambda L))^2\} d\lambda \\ &+ \frac{2}{L^2} \sqrt{\int_0^1 \{(u_1(\lambda L) - u_2(\lambda L))^2 + (v_1(\lambda L) - v_2(\lambda L))^2\} d\lambda} \\ &\quad \times \sqrt{\int_0^1 \{(u_1(\lambda L) + u_2(\lambda L))^2 + (v_1(\lambda L) + v_2(\lambda L))^2\} d\lambda} \\ &= \frac{2}{L^2} \int_0^1 \{u_1^2(\lambda L) + v_1^2(\lambda L) + u_2^2(\lambda L) + v_2^2(\lambda L)\} d\lambda \\ &+ \frac{2}{L^2} \sqrt{\int_0^1 \{u_1^2(\lambda L) + v_1^2(\lambda L) + u_2^2(\lambda L) + v_2^2(\lambda L)\} d\lambda} \\ &\quad - 2 \int_0^1 \{u_1(\lambda L) u_2(\lambda L) + v_1(\lambda L) v_2(\lambda L)\} d\lambda \\ &\times \sqrt{\int_0^1 \{u_1^2(\lambda L) + v_1^2(\lambda L) + u_2^2(\lambda L) + v_2^2(\lambda L)\} d\lambda} + 2 \int_0^1 \{u_1(\lambda L) u_2(\lambda L) + v_1(\lambda L) v_2(\lambda L)\} d\lambda \\ &\leq \frac{2}{L^2} \int_0^1 \{u_1^2(\lambda L) + v_1^2(\lambda L) + u_2^2(\lambda L) + v_2^2(\lambda L)\} d\lambda \\ &\quad + \frac{2}{L^2} \int_0^1 \{u_1^2(\lambda L) + v_1^2(\lambda L) + u_2^2(\lambda L) + v_2^2(\lambda L)\} d\lambda \end{aligned}$$

$$\begin{aligned}
&= \frac{4}{L^2} \left[\int_0^L \{u_1^2(\lambda L) + v_1^2(\lambda L)\} d\lambda + \int_0^L \{u_2^2(\lambda L) + v_2^2(\lambda L)\} d\lambda \right] \\
&\leq \frac{4}{L^2} \left[\frac{L^2}{12} + \frac{L^2}{12} \right] \text{ from Appendix B.7} \\
&= 2/3
\end{aligned}$$

$$\text{So, } D_5(A, B) + D_5(A, B_\pi) \leq \sqrt{2}/\sqrt{3}$$

$$\text{Thus, } D_6(A, B) \leq \text{Min}\{D_5(A, B), D_5(A, B_\pi)\} \leq 1/\sqrt{6}$$

Proposition 4.3.29 $D_6(A, A^{-1}) = 0$ if and only if A is antisymmetric.

Proof. It is clear from Proposition 4.2.12, since $D_6(A, B) = 0$ if and only if A and B have the same shape.

Now, if it is required that the shapes of A and A^{-1} be the same, D_6 can be modified as before.

Definition 4.3.11 D_7 is a distance function on $\mathcal{L}_1$ such that for A, B belonging to $\mathcal{L}_1$,

$$D_7(A, B) = \text{Min}\{D_6(A, B), D_6(A, B^{-1})\} \quad (4.7)$$

The definition of D_7 is unambiguous in the sense that $D_7(A, B) = D_7(B, A)$. This can be seen from the Proposition 4.3.25.

Proposition 4.3.30 $D_7(A, A^{-1}) = 0$.

Proof. Trivial.

Proposition 4.3.31 $\sup_{A, B \in \mathcal{L}_1} D_7(A, B) \leq 1/\sqrt{6}$.

Proof. It is clear from Proposition 4.3.28.

Definition 4.3.12 μ_2 is a similarity measure

(based on D_7) between two shapes such that

$$\mu_2(A, B) = 1 - \sqrt{6}D_7(A, B)$$

(μ_2 can be defined on the basis of D_6 also).

Evidently, $1 \geq \mu_2 \geq 0$. Note that $\mu_2(A, B) = 1$ if and only if B is a translation, dilation, rotation and/or inverse of A , that is, if and only if A and B have the same shape or A and B^{-1} have the same shape.

4.4 Description of Open Curves

An open curve is described here as a polyline, that is, as a finite sequence of distinct points on the curve. Without loss of generality, we consider for description the open curves belonging to $\mathcal{L}_1$ only (that is, open curves having centroid at the origin and a fixed length L). Let $P_0, P_1, \dots, P_n$ be $(n+1)$ points on an open curve A in $\mathcal{L}_1$ such that the arc distance between P_{j-1} and P_j is L/n and P_0 and P_n are the starting and end points respectively. In other words, the co-ordinates of P_j are $(u(jL/n), v(jL/n))$ for $j=0, 1, \dots, n$ where $A = \{(u(s), v(s)) : s \in [0, L]\}$.

Definition 4.4.1 The polyline $\underline{P} = \{P_j\}_0^n$ is the description of the open curve A .

Proposition 4.4.1 The description $\underline{P}$ is information

preserving in the sense that $\underline{P}$ tends to the original open curve A as n increases.

Proof. See in Appendix B.8.

Notation Let T_n be the set of all possible descriptions $\underline{P}$ of open curves in $\mathcal{L}_1$.

Note that the description $\underline{P}$ is the output of a transform or a function, say g_n , from $\mathcal{L}_1$ to T_n . That is $g_n(A) = \underline{P}$.

Let $\underline{P} = \{P_j\}_0^n$ and $\underline{Q} = \{Q_j\}_0^n$ be two elements of T_n . Suppose the co-ordinates of P_j are (x_j, y_j) for $j=0, 1, \dots, n$.

Definition 4.4.2 $\underline{Q}$ is a rotation of $\underline{P}$ if for some α in $[0, 2\pi)$, the co-ordinates of Q_j are $(x_j \cos \alpha - y_j \sin \alpha, x_j \sin \alpha + y_j \cos \alpha)$ for all $j=0, 1, \dots, n$.

Definition 4.4.3 Let V_n be a relation in T_n such that for $\underline{P}$ and $\underline{Q}$ in T_n , $(\underline{P}, \underline{Q}) \in V_n$ if $\underline{Q}$ is a rotation of $\underline{P}$.

Proposition 4.4.2 V_n is an equivalence relation in T_n .

Proof. Same as that of Proposition 4.2.9.

V_n induces a relation in $\mathcal{L}_1$ in the following way: Let V_n^* be a relation in $\mathcal{L}_1$ such that for A, B belonging to $\mathcal{L}_1$, $(A, B) \in V_n^*$ if $(g_n(A), g_n(B)) \in V_n$.

Proposition 4.4.3 V_n^* is an equivalence relation in $\mathcal{L}_1$.

Proof. Trivial.

Definition 4.4.4 An equivalence class generated by V_n^* in $\mathcal{L}_1$ is a shape at the level n .

Let $\mathcal{S}_n$ be the collection of all such equivalence classes, that is, of all shapes at the level n . In general, a shape at the level n means more than one single shape in $\mathcal{S}^1$ (unless it is the shape of a straight line segment). This is because for A, B in $\mathcal{L}_1$, $(A, B) \in S$ implies $(A, B) \in V_n^*$ but not vice versa. In fact, $(A, B) \in S$ if and only if $(A, B) \in V_n^*$ for all n . So, each equivalence class in $\mathcal{S}_n$ contains open curves whose shapes (in $\mathcal{S}^1$) are not necessarily the same but resemble one another 'closely'. The degree of this closeness increases as the value of n increases (for a fixed L). That is, each equivalence class in $\mathcal{S}_n$ decreases to a single equivalence class in $\mathcal{S}^1$ as n goes to infinity (Appendix B.8). In the next section we will see that the computed values of $D_3(A, B)$ and $D_6(A, B)$ (i.e. D_3^* and D_6^* respectively) are zero if and only if $(A, B) \in V_n^*$. That is, the computed shape distance values $D_3^*(A, B)$ or $D_6^*(A, B)$ can be zero even when A and B have different shapes (in $\mathcal{S}^1$). But this is a price one has to pay for finitization of an analogue or continuous phenomenon.

4.5 Techniques to Compute Shape Distance and Shape Similarity of Open Curves

There are two ways to obtain a digital curve. One is through thinning a thick digital picture (Chapter 3) while the other is through digitization of an analogue curve. The digitization principles discussed in Chapter 1 do not consider thin binary pictures like curves. That is why a different scheme (Freeman, 1974) is applied for digitizing analogue curves. Now, from the definition of an analogue open curve, it is clear that the removal of any point (except the starting and end points) from the curve makes it disconnected. Equivalently, a digital open curve can be defined as a set S of 1-pixels such that (i) S is connected and (ii) every 1-pixel (except two 1-pixels) in S has exactly two 8-neighbours in S . (Those two 1-pixels are the extreme 1-pixels of the digital open curve). But unfortunately this definition does not work in the sense that a digitized version of an analogue open curve may not satisfy the above conditions. For example, in Fig.4.3, the second 1-pixel from the bottom has three 8-neighbours. Rosenfeld and Kak (1976) defined a digital open curve as a set S of 1-pixels such that (i) S is connected and (ii) most of the 1-pixels of S have exactly two 8-neighbours in S . We shall from now on characterize a digital open curve by the position of its starting 1-pixel and the string of directional codes (of 1_8) from that point to the end point.

Without loss of generality, the starting point can be taken as the origin $(0,0)$. Thus, a digital open curve is only a string of directional codes d_i belonging to I_8 . For example, the string for Fig.4.3 ^{is} 8822211 (assuming the bottom 1-pixel to be the starting point).

We will now present a method for getting any fixed number of equidistant points on a digital open curve. Note that all the directional codes in I_8 do not have the same length. In fact, there are two different lengths x and $\sqrt{2}x$ where x is the size of a pixel. The odd directional codes 1, 3, 5 and 7 have length $\sqrt{2}x$ and the even directional codes have length x . To get rid of this ^{inequality} in length we modify the string in the following manner: replace d_i by 7 consecutive d_i 's if d_i is odd, other-wise replace d_i by 5 consecutive d_i 's. The lengths of the new d_i 's can be considered the same (approximately $x/5$) since $7/5$ is very close to $\sqrt{2}$. But the length of the new string (i.e. the total number of d_i 's in it), say N , will be 5 to 7 times that of the old one. Suppose now that we have to get $n+1$ equidistant points (distance is taken along the curve) on the digital open curve. Without loss of generality, the common length of the new d_i 's (i.e. $x/5$) can be assumed to be 1. Let $N=nr$. P_0 is the origin, i.e. the starting point of the curve. Define P_j to be the point on the digital curve obtained after rj d_i 's from the starting point ($j=1, \dots, n$). Note that it is not

essential to find the new d_i 's explicitly in order to compute $P_j (j=1, \dots, n)$. The value of N alongwith the old d_i 's is sufficient for the purpose. The new d_i 's are mentioned only for clarification of the definition of P_j . For Fig.4.3, the new d_i 's are 8888888888222/2222222222221/1111111111111. The slashes indicate the partitioning of the string if $n=3$. The co-ordinates of P_1, P_2 and P_3 are $(3, 10)$, $(15+1/\sqrt{2}, 10+1\sqrt{2})$ and $(15+7\sqrt{2}, 10+7\sqrt{2})$ respectively.

Computations of D_2^* , D_3^* , D_4^* and μ_1^*

From the equidistant points $P_0, P_1, \dots, P_n$, d_j 's are computed as the directional codes of $\overrightarrow{P_{j-1}P_j}$. (computation of d_j is given in Appendix B.9). $\underline{d} = \{d_j\}_1^n$ is the discrete slope of a digital curve. Note that this $\underline{d}$ tends to the slope of an open curve (Definition 4.3.2) as n increases. Suppose $\underline{d}_1 = \{d_{1j}\}_1^n$ and $\underline{d}_2 = \{d_{2j}\}_1^n$ are the discrete slopes of two digital open curves for which shape distance and shape similarity are to be computed. If k is a real number and $\underline{d}$ a discrete slope, then by $\underline{d}+k$ we will mean the new discrete slope $\{d_j + k\}_1^n$.

Now, the computation of D_3^* on the basis of the present definition of D_1 in (4.1) is difficult since $f(k) = D_2^*(\underline{d}_1, \underline{d}_2+k) = \frac{1}{n} \sum_{j=1}^n D_1(d_{1j}, d_{2j} + k)$ is not differentiable everywhere with respect to k . Here we propose an alternative definition of D_1 which will retain all the desirable properties of the earlier

D_1 and which will make $f(k)$ differentiable everywhere with respect to k . Changing D_1 in (4.1), we write

$$D_1(d_1, d_2) = 4 \sin^2 \frac{\pi(d_1 - d_2)}{8} \quad (4.8)$$

$$\begin{aligned} \text{Now, } f(k) &= D_2^*(d_1, d_2 + k) \\ &= \frac{1}{n} \sum_{j=1}^n D_1(d_{1j}, d_{2j} + k) \\ &= \frac{4}{n} \sum_{j=1}^n \sin^2 \frac{\pi(d_{1j} - d_{2j} - k)}{8} \end{aligned}$$

D_3^* will be the infimum of $f(k)$ over all k . (Note that here D_2^*, D_3^*, D_4^* and μ_1^* represent the discrete forms of D_2, D_3, D_4 and μ_1 defined in section 4.3. That is, D_2^*, D_3^*, D_4^* and μ_1^* tend to D_2, D_3, D_4 and μ_1 as n goes to infinity).

$$\begin{aligned} f'(k) &= \frac{df(k)}{dk} = 0 \\ \Leftrightarrow \sin \frac{\pi k}{4} \sum_{j=1}^n \cos \theta_j &= \cos \frac{\pi k}{4} \sum_{j=1}^n \sin \theta_j \end{aligned} \quad (4.9)$$

$$\Leftrightarrow k = \frac{4}{\pi} \tan^{-1} \frac{\sum_{j=1}^n \sin \theta_j}{\sum_{j=1}^n \cos \theta_j} \quad (4.10)$$

$$\text{where } \theta_j = \frac{\pi(d_{1j} - d_{2j})}{4}$$

$$\text{If } \sum_{j=1}^n \sin \theta_j = \sum_{j=1}^n \cos \theta_j = 0,$$

then $\frac{df(k)}{dk} = 0$, that is, $f(k)$ is a constant function.

In that case, the value of D_3^* can be taken as $f(k)$ for any k .

Otherwise, k in (4.10) is well-defined. In that case there exists a unique k_0 in $[0, 4)$ that satisfies (4.10) and $k_0 + 4$ also satisfies (4.10). One of k_0 and $k_0 + 4$ maximizes $f(k)$ and the other minimizes it.

$$\frac{d^2 f(k)}{dk^2} = \text{const} \left\{ \sin \frac{\pi k}{4} \sum_{j=1}^n \sin \theta_j + \cos \frac{\pi k}{4} \sum_{j=1}^n \cos \theta_j \right\} \quad (4.11)$$

So it is clear that

$$\left. \frac{d^2 f(k)}{dk^2} \right|_{k=k_0} = - \left. \frac{d^2 f(k)}{dk^2} \right|_{k=k_0+4}$$

Thus the value of the second derivative is positive at exactly one of k_0 and $k_0 + 4$ (it can not be zero because of (4.9) and (4.11)). That value of k , say $\hat{k}$, minimizes $f(k)$. So, we compute $D_3^*(\underline{d}_1, \underline{d}_2)$ as $f(\hat{k})$. In a similar way we compute $D_3^*(\underline{d}_1, \underline{d}_2^{-1})$ where $\underline{d}_2^{-1} = \{d_{3j}\}_1^n$ is such that $d_{3j} = d_{2, n-j+1} + 4$ for $j=1, \dots, n$. Thus D_4^* is computed as $\text{Min} \{D_3^*(\underline{d}_1, \underline{d}_2), D_3^*(\underline{d}_1, \underline{d}_2^{-1})\}$ and the shape similarity measure μ_1^* is computed as $1 - D_4^*/2$.

Computations of D_5^* , D_6^* , D_7^* and μ_2^*

Here D_5^*, D_6^*, D_7^* and μ_2^* are the discrete forms of D_5, D_6, D_7 and μ_2 defined in Section 4.3. Suppose, $P_{10}, P_{11}, \dots, P_{1n}$ are the $n+1$ equidistant points on a digital open curve A and $P_{20}, P_{21}, \dots, P_{2n}$ are the $n+1$ equidistant points on another digital

open curve B. P_{10} and P_{20} are the starting points and P_{1n} and P_{2n} are the end points. Suppose that the co-ordinates of the point P_{rj} are (x_{rj}, y_{rj}) , $r=1,2$ and $j=0,1,\dots,n$.

Suppose the lengths of A and B are L_1 and L_2 respectively (L_1, L_2 were computed at the time of computations of P_{rj}).

Now, the co-ordinates (x_{rj}, y_{rj}) will be normalized in terms of the size and position.

$$\text{Let } \bar{x}_r = \frac{1}{n+1} \sum_{j=0}^n x_{rj} \text{ and } \bar{y}_r = \frac{1}{n+1} \sum_{j=0}^n y_{rj}$$

$$\text{Let } u_{rj} = (x_{rj} - \bar{x}_r)/L_r \text{ and } v_{rj} = (y_{rj} - \bar{y}_r)/L_r$$

for $r=1,2$ and $j=0,1,\dots,n$. Note that the $n+1$ points (u_{rj}, v_{rj}) have the centroid at the origin $(0,0)$ and have the length, in a sense, 1 for $r=1,2$. We compute $D_5^*(A, B)$ as

$$\sqrt{\frac{1}{n} \sum_{j=1}^n \left\{ (u_{1j} - u_{2j})^2 + (v_{1j} - v_{2j})^2 \right\}}$$

Note that $D_5^*(A, B)$ tends to $D_5(A, B)$ defined in (4.5) as n goes to infinity. Now we shall compute D_6^* as an approximation for D_6 . If the points (u_{2j}, v_{2j}) are rotated by an angle α in the anticlockwise direction, the new points obtained are

$$(u_{2j} \cos \alpha - v_{2j} \sin \alpha, u_{2j} \sin \alpha + v_{2j} \cos \alpha), \quad j=1, \dots, n.$$

$$\text{Let } g(\alpha) = \frac{1}{n} \sum_{j=1}^n \left\{ (u_{1j} - u_{2j} \cos \alpha + v_{2j} \sin \alpha)^2 + (v_{1j} - u_{2j} \sin \alpha - v_{2j} \cos \alpha)^2 \right\}$$

D_6^* will be the infimum of $\sqrt{g(\alpha)}$ over all α .

$$g'(\alpha) = \frac{dg(\alpha)}{d\alpha} = 0$$

$$\Leftrightarrow \sin\alpha \sum_{j=1}^n (u_{1j}u_{2j} + v_{1j}v_{2j}) + \cos\alpha \sum_{j=1}^n (u_{1j}v_{2j} - v_{1j}u_{2j}) = 0 \quad (4.12)$$

$$\Leftrightarrow \alpha = \tan^{-1} \frac{\sum_{j=1}^n (v_{1j}u_{2j} - u_{1j}v_{2j})}{\sum_{j=1}^n (u_{1j}u_{2j} + v_{1j}v_{2j})} \quad (4.13)$$

If both $\sum_{j=1}^n (v_{1j}u_{2j} - u_{1j}v_{2j})$ and $\sum_{j=1}^n (u_{1j}u_{2j} + v_{1j}v_{2j})$ are zero,

then $g'(\alpha) = 0$, that is, $g(\alpha)$ is a constant function. In that case, the value of D_6^* can be taken as $\sqrt{g(\alpha)}$ for any α .

Otherwise, α in (4.13) is well-defined. In that case, there exists a unique α_0 in $[0, \pi)$ that satisfies (4.13) and $\alpha_0 + \pi$ also satisfies (4.13). One of α_0 and $\alpha_0 + \pi$ maximizes $g(\alpha)$ and the other minimizes it.

$$\frac{d^2g(\alpha)}{d\alpha^2} = \text{const} \left\{ \cos\alpha \sum_{j=1}^n (u_{1j}u_{2j} + v_{1j}v_{2j}) - \sin\alpha \sum_{j=1}^n (u_{1j}v_{2j} - v_{1j}u_{2j}) \right\} \quad (4.14)$$

$$\text{So, } \left. \frac{d^2g(\alpha)}{d\alpha^2} \right|_{\alpha=\alpha_0} = - \left. \frac{d^2g(\alpha)}{d\alpha^2} \right|_{\alpha=\alpha_0+\pi}$$

Thus the value of the second derivative is positive at exactly one of α_0 and $\alpha_0 + \pi$ (it can not be zero because of (4.12) and (4.14)). That value of α , say $\hat{\alpha}$, minimizes $g(\alpha)$. So, we compute $D_6^*(A, B)$ as $\sqrt{g(\hat{\alpha})}$. Now, we define $u_{3j} = u_{2, n-j}$ and $v_{3j} = v_{2, n-j}$ so that (u_{3j}, v_{3j}) can be treated as equidistant

points on B^{-1} . So, in a similar way $D_6^*(A, B^{-1})$ can be computed. Then $D_7^*(A, B)$ is computed as $\text{Min} \{D_6^*(A, B), D_6^*(A, B^{-1})\}$ and the shape similarity measure μ_2^* is computed as $1 - \sqrt{6D_7^*}$.

4.6 Conclusion

Scalar invariant shape descriptors for curves have been reported in the literature, which are based on Fourier coefficients (Zahn and Roskies, 1972; Persoon and Fu, 1977) and moments (Dudani, Breeding and McGhee, 1977). Shape distance defined on the basis of these scalar descriptors can be used for shape discrimination. Shape distances D_4 and D_7 defined in this chapter are based on structural informations and are more intuitively appealing. That is, for look-alike shapes the values of D_4 and D_7 will be small while for dissimilar shapes they will be large.

We will present here the computed values of μ_1 and μ_2 for certain shapes. In Fig.4.4, the shape 'S' (A_1) and the straight line shape (A_4) are shown alongwith two intermediate shapes (A_2 and A_3). The values of μ_1^* and μ_2^* (their computational techniques are given in Section 4.5) for each pair of shapes in Fig.4.4 are shown in Table 4.1. In Fig.4.5, some noisy straight line segments are shown with varying degrees of distortion.

The values of μ_1^* and μ_2^* between the perfect straight line segment and each of these distorted line segments are given in Table 4.2. It can be seen from the two tables that both μ_1^* and μ_2^* are monotonic in the sense that less distortion in the shape means higher values of μ_1^* and μ_2^* and more distortion means lower values of μ_1^* and μ_2^* . In other words, the results agree with the human visual judgement.

We will use shape distances D_3 and D_6 in defining two corresponding shape distances for 2-dimensional regions on the basis of their boundaries in the next chapter. In order to measure shape distance between two thick open curves, we first thin the thick curves (Chapter 3) and then find the shape distance between the two thinned curves. The value of the shape distance between two such thick curves may not be unique since a thick curve does not always give rise to a unique thinned output.

So far we have considered shape to be invariant under translation, dilation and rotation. The definition of shape can be broadened by making shape invariant under reflection also.

Reflection or mirror image of an open curve $A = \{(u(s), v(s)) :$

$s \in [0, L]\}$ is $\text{Ref1}(A) = \{(u(s), -v(s)) : s \in [0, L]\}$. It can be seen

that if the slope of A is $\{w(\lambda) : \lambda \in (0, 1)\}$, the slope of $\text{Ref1}(A)$

is $\{4-w(\lambda) \pmod{8} : \lambda \in (0, 1)\}$. In order to accomodate this extension

of definition of shape in the shape distances defined in this

chapter, we can modify $D_4(A, B)$ and $D_7(A, B)$ as

$\text{Min} \{D_4(A, B), D_4(A, \text{Ref1}(B))\}$ and $\text{Min} \{D_7(A, B), D_7(A, \text{Ref1}(B))\}$
respectively.

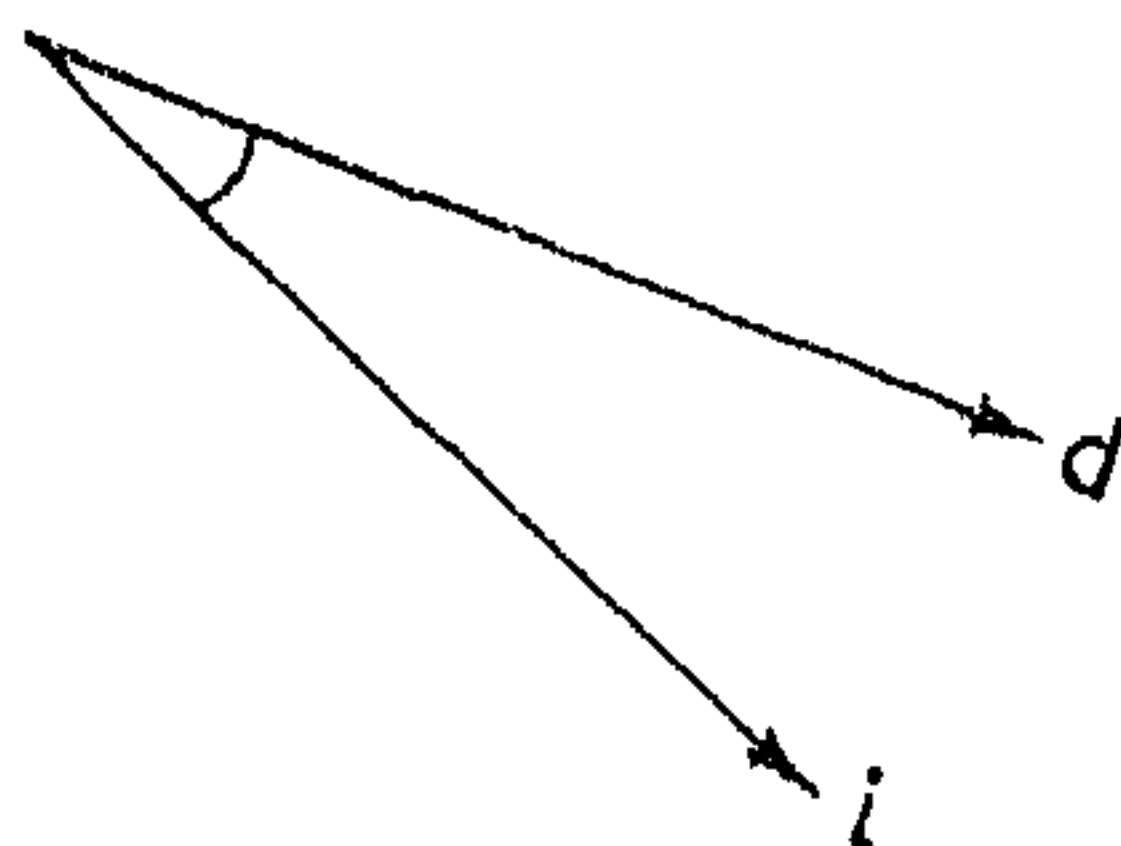


Fig.4.1 The angle between d and i is $(i-d)45^\circ$

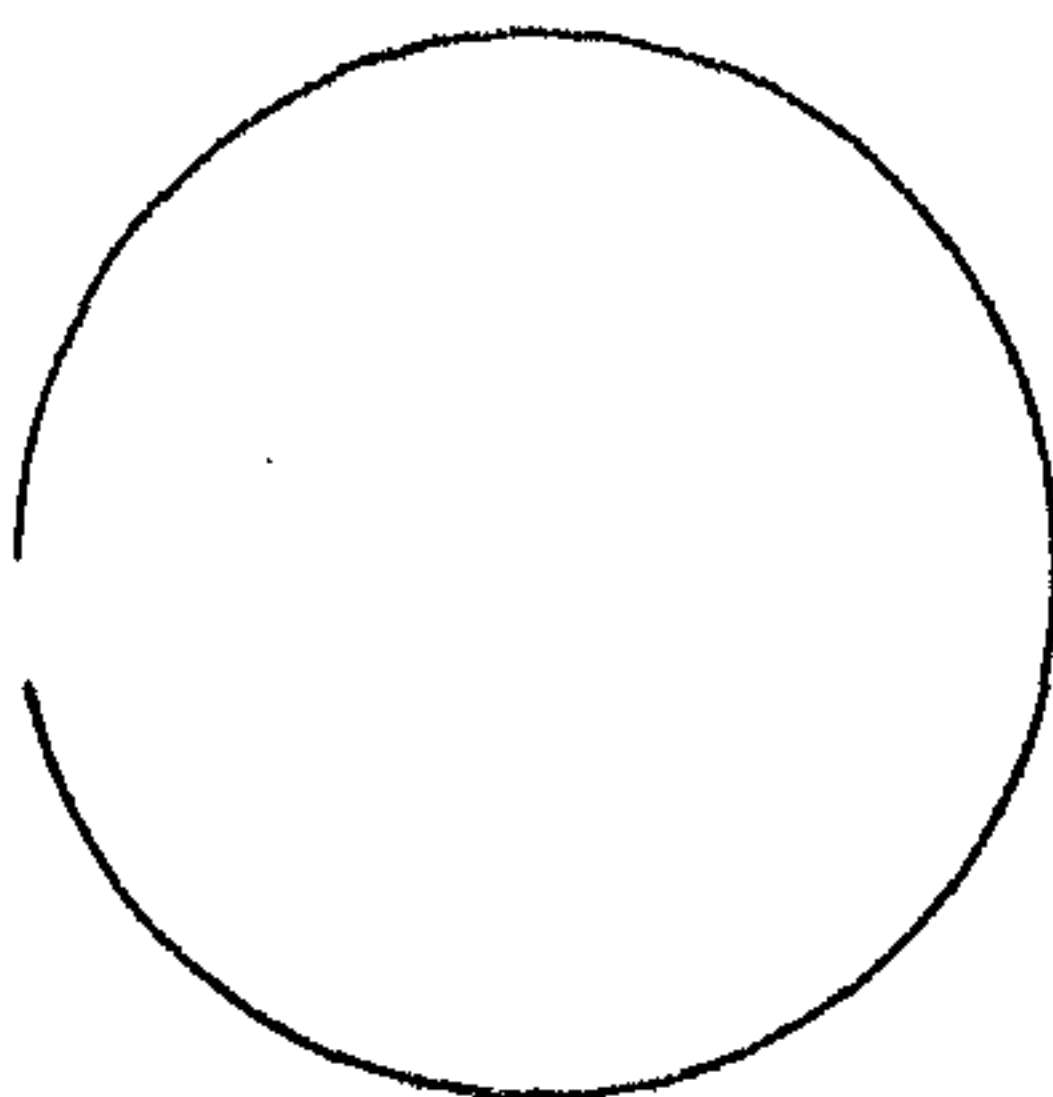


Fig.4.2

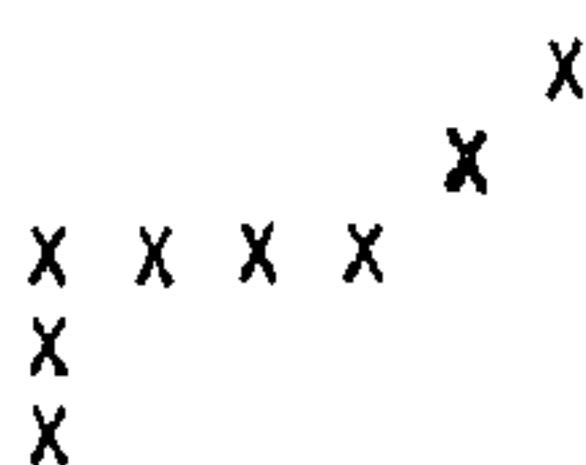


Fig.4.3 A digital open curve

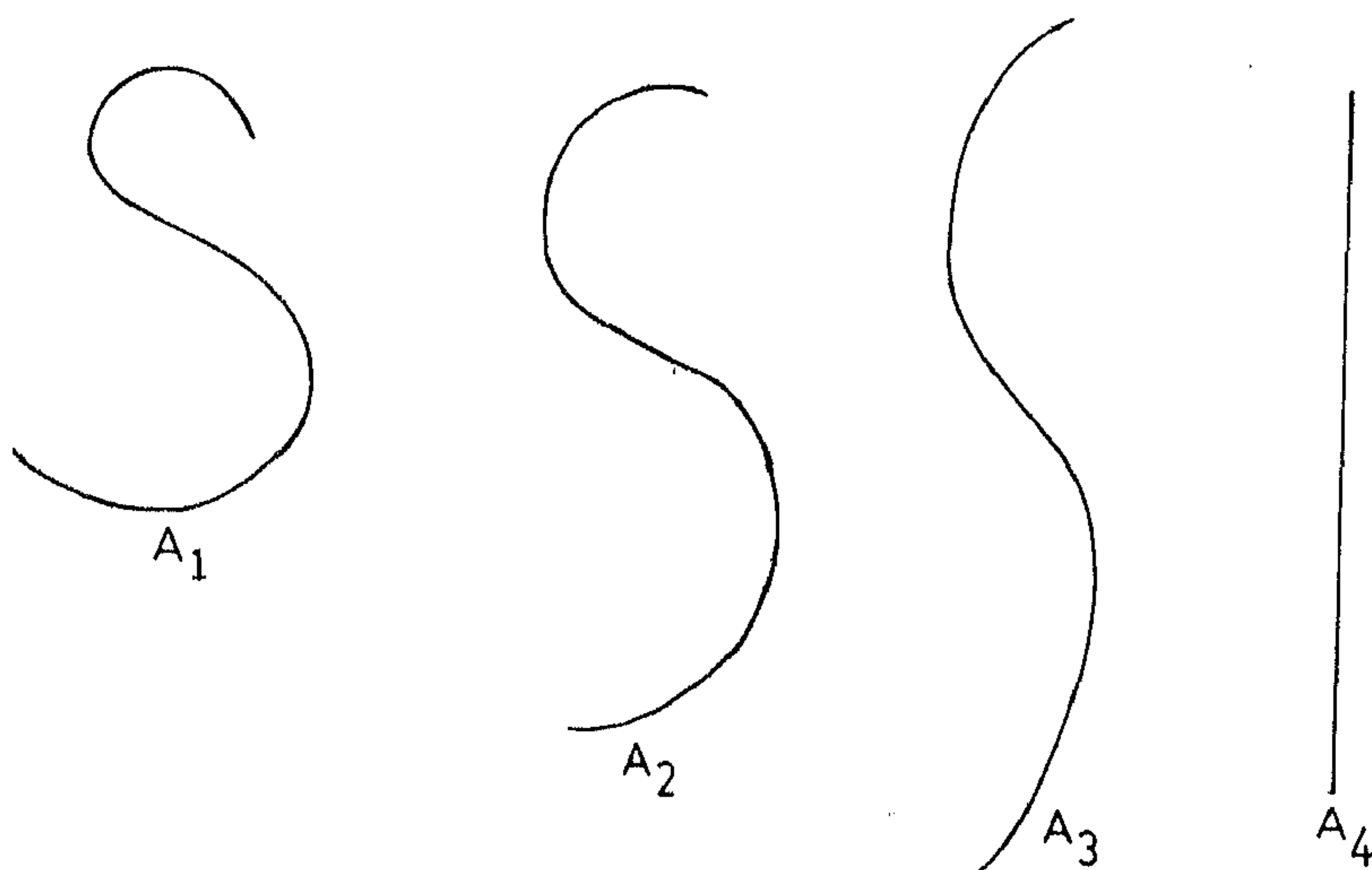


Fig. 4.4

A_1 is the shape 'S' and A_4 is the straight line shape. A_2 and A_3 are two intermediate shapes. The computed values of μ_1 and μ_2 for each pair of the above four shapes are given in Table 4.1 and agree with the human visual judgement.

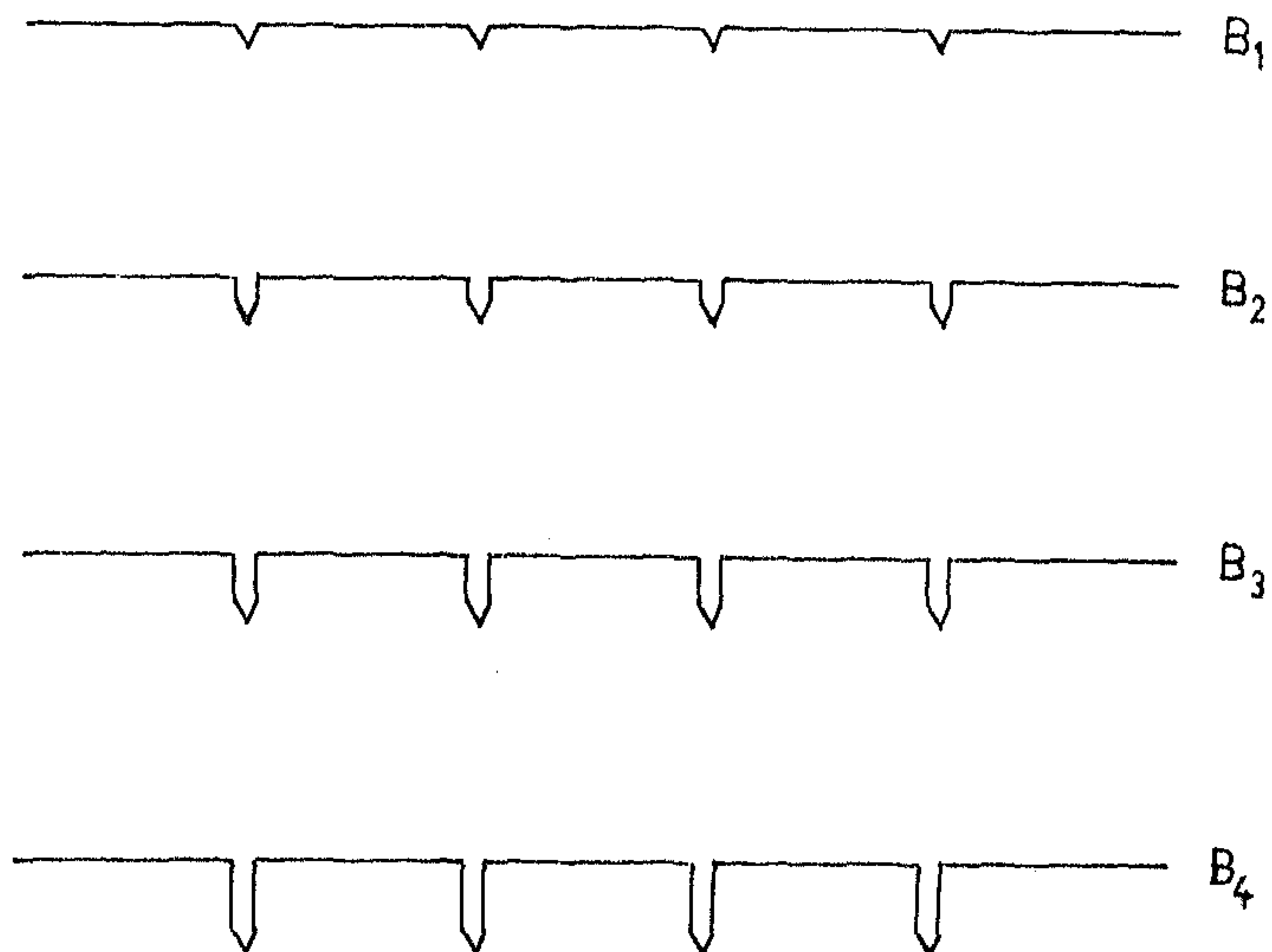


Fig. 4.5 Above are some noisy straight line segments with varying degrees of distortion. The computed values of μ_1 and μ_2 between the perfect straight line segment and each of the above shapes are given in Table 4.2 and agree with human intuition.

Table 4.1

Shape similarity values μ_1^* and μ_2^* between shapes
shown in Fig.4.4 (n=32)

μ_1^*/μ_2^*	A_2	A_3	A_4
A_1	0.9240/0.8490	0.7246/0.7277	0.3997/0.6138
A_2		0.9134/0.8737	0.6533/0.7463
A_3			0.8744/0.8627

Table 4.2

Shape similarity values μ_1^* and μ_2^* between
a straight line segment and the noisy line
segments shown in Fig.4.5 (n=32)

	B_1	B_2	B_3	B_4
μ_1^*	0.9900	0.9729	0.9021	0.8423
μ_2^*	0.9744	0.9196	0.8656	0.8166

CHAPTER 5

Shape Distance and Shape Similarity of Two-Dimensional Regions

5.1 Introduction

This chapter deals with the shape of two-dimensional regions and proposes several methods for quantifying shape distance and shape similarity. This quantified shape distance or shape similarity may be used in classifying a new shape as one of several known shapes. A region is defined as a specific kind of subset in the plane. Two regions have the same shape if and only if one is a translation, dilation and rotation of the other. Thus for shape matching, a region has to be normalized in terms of its position, size and orientation. Normalization of a region in terms of its position and size can easily be achieved if its centre of gravity and area are considered. The main problem for shape matching of regions involves the normalization of orientation of a region and also the normalization of the starting point on the boundary of a region if the boundary is studied.

There are already several descriptions which are invariant under translation, dilation and rotation of a region. Examples are methods involving moments (Hu, 1962; Smith and

Wright, 1971; Dudani, Breeding and McGhee, 1977) and Fourier coefficients (Zahn and Roskies, 1972; Granlund, 1972; Persoon and Fu, 1977). Although the method of moments is an information preserving technique, it is difficult to relate the higher order moments to shape. Lower order moments can be valuable as supplements to other recognition techniques. For example, they can be used to normalize data before other measurements are taken (Parui and Dutta Majumder, 1982B; 1983B; 1983E). The only recent publication using moments deals with identification of aircrafts (Dudani et al, 1977) where two sets of moment invariants are used as features. One set is calculated from the full silhouette and the other from the contour only.

Methods involving Fourier transform of the boundary seem to be more promising. The boundary of a region can be expressed in terms of tangent angle versus arc length, or as the complex function $b_x(t) + jb_y(t)$ where b_x and b_y are the co-ordinates of the boundary points and t is a parameter, usually the length. The first approach has been used by Zahn and Roskies (1972), Bennet and MacDonald (1975) and others. The second has been used by Granlund (1972), Persoon and Fu (1977) and others. The Zahn and Roskies formulation leads to Fourier descriptors whose values are identical for two curves which differ only in position, size and orientation but have analogous starting points. Different starting points give rise to

difference only in phase angles that can be normalized. After that normalization, it has been seen, shapes which 'look similar' to people are near each other in the space of low-order Fourier descriptors. In the formalism proposed by Granlund and developed by Persoon and Fu, the Fourier descriptors depend on the position, size, orientation and starting point of the curve. But a distance function between two curves can be defined on the basis of these Fourier descriptors in such a way that its value is independent of the position, size, orientation and starting point of the curves.

Both moments and Fourier coefficients are scalar transforms. Space domain methods also are used to define shape similarity for regions (Bribiesca and Guzman, 1980). Bribiesca et al define 'shape number' of a region on the basis of its boundary using differential chain code at various resolutions. This shape descriptor 'shape number' does not depend on the position, size and orientation of the region but contains information about shape. The largest order for which the shape numbers of two regions coincide provides a measure of shape similarity.

In this chapter, we propose four definitions of shape distance out of which only the first one involves scalar transform technique while the others involve space domain techniques. But the first is an information nonpreserving

technique and the others are information preserving techniques. Among the space domain techniques, the first involves all the points (both interior and boundary) of a region while the other two consider only the boundary. Invariance with regard to orientation and starting point on the boundary of a region is taken care of by a certain major axis of the region. The two shape distances constructed for curves in Chapter 4 will be used in this chapter when only the boundary of a region is considered. The scalar transform description is invariant under translation, dilation and rotation but the space domain descriptions do not have those invariance properties. Nevertheless, the shape distances defined on the basis of those descriptions do not depend on the position, size and orientation of regions and hence may be used as shape discriminators. We find that the shape distances defined on the basis of space domain descriptions are more intuitively compelling than those defined on the basis of scalar features.

In the next section we discuss certain drawbacks of the statistical linear fit with minimum squared error for a finite set of points in the plane and modify this linear fit in order to serve the purpose of shape matching. In Section 5.5 we extend the modified definition of the best linear fit to certain uncountable subsets in the plane and call it the major axis.

In Section 5.3, we give a formal definition of a region

and then define the shape of a region as an equivalence class of regions within which any region can be transformed to any other region through translation, dilation and rotation. In Section 5.4, a region is described as an n -dimensional vector which is invariant under translation, dilation and rotation. This vector is called the shape vector of the region. But the shape description in terms of this shape vector is not information preserving. Nevertheless, for a limited class of shapes, it can be used for shape discrimination. An algorithm for computation of the shape vector is provided. The best linear fit defined for finite sets of points in Section 5.2 is extended to uncountable sets, i.e., regions in Section 5.5 and is called the major axis which is almost always unique. The reference points of a region are defined to be the two farthest points where its major axis intersects with its boundary. Some invariant properties of the major axis and the reference points are established. The major axis is made use of in Section 5.6 to construct a shape distance which is the area of the unmatched parts of two regions in certain orientation provided by their major axes. Methods to compute this shape distance are also given. In Section 5.7, two shape distances are defined on the basis of the boundary of a region where the problem of the starting point on the boundary is solved by the reference points. These two distances involve the two shape distances for curves defined in Chapter 4. Finally, Section 5.8 is devoted to results and discussion.

5.2 A Modified Least Squares Method

The classical least squares method is as follows:

Let (x_i, y_i) , $i=1, \dots, n$ be n points in the two-dimensional plane. Suppose, y is to be predicted on the basis of x . Then the squared error of any straight line $y = a + bx$ with respect to the n points is $\sum_{i=1}^n (y_i - a - bx_i)^2$. The straight line with the minimum squared error is $y = \hat{a} + \hat{b}x$ where

$$\hat{b} = \frac{\frac{1}{n} \sum_{i=1}^n x_i y_i - \bar{x} \bar{y}}{\frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2}, \quad \hat{a} = \bar{y} - \hat{b} \bar{x},$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

Now if this least squares method is applied to a set of points (x_i, y_i) , it can be seen that the slope of the best linear fit $y = \hat{a} + \hat{b}x$ remains unchanged if the set is translated or dilated. But the relative slope of the line $y = \hat{a} + \hat{b}x$ may change if the set is rotated. (The definitions of translation, dilation and rotation are given in Chapter 2). An example is given in Fig.5.1. The three points in Fig.5.1(b) are obtained by rotating the three points in Fig.5.1(a) by an angle of 45° in the clockwise direction. The best fit in the first case is $y = 0$. But in the second case it is $y = -x/2$, not $y = -x$.

This drawback can be overcome by changing the definition of the squared error, that is, by taking the perpendicular distance

from (x_i, y_i) to a straight line. Suppose, (r_i, θ_i) is the polar representation of (x_i, y_i) , that is, $x_i = r_i \cos \theta_i$, $y_i = r_i \sin \theta_i$ where r_i is a non-negative real number and θ_i belongs to $[0, 2\pi)$. Now, the perpendicular (i.e. the shortest) distance from (x_i, y_i) to a straight line $y = x \tan \theta + c$ is the absolute value of $[c \cos \theta + r_i \sin(\theta - \theta_i)]$, where c is a real number and θ belongs to $[0, \pi)$. So the new squared error (in terms of this perpendicular distance) of the straight line $y = x \tan \theta + c$ with respect to the n points is

$$f(\theta, c) = \sum_{i=1}^n [c \cos \theta + r_i \sin(\theta - \theta_i)]^2 \quad (5.1)$$

We will now find the values of θ and c that minimize $f(\theta, c)$.

From $\delta f(\theta, c) / \delta c = 0$ we get

$$c = \bar{y} - \bar{x} \tan \theta \quad (5.2)$$

Substituting c in (5.1) we obtain

$$\begin{aligned} f(\theta, c) &= \sum_{i=1}^n [(\bar{y} - \bar{x} \tan \theta) \cos \theta + r_i \sin(\theta - \theta_i)]^2 \\ &= \sum_{i=1}^n [(x_i - \bar{x}) \sin \theta - (y_i - \bar{y}) \cos \theta]^2 \end{aligned}$$

Now, $\frac{\delta f}{\delta \theta}$

$$\begin{aligned} &= 2 \sum_{i=1}^n [(x_i - \bar{x}) \sin \theta - (y_i - \bar{y}) \cos \theta] [(x_i - \bar{x}) \cos \theta + (y_i - \bar{y}) \sin \theta] \\ &= \sin 2\theta \left[\sum_{i=1}^n (x_i - \bar{x})^2 - \sum_{i=1}^n (y_i - \bar{y})^2 \right] \\ &\quad - \cos 2\theta \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \end{aligned}$$

For optimum θ , $\frac{\delta f}{\delta \theta} = 0$

$$\text{That is, } \tan 2\theta = \frac{2 \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2 - \sum_{i=1}^n (y_i - \bar{y})^2} \quad (5.3)$$

In (5.3) we assume that at least one of the numerator and the denominator is non-zero. (5.4)

Now the equation (5.3) can be satisfied by two values of θ belonging to $[0, \pi)$, i.e., α and $\alpha + \pi/2$ for some α belonging to $[0, \pi/2)$. One of these angles minimizes f while the other maximizes it.

In order to find out which angle minimizes f , the second derivative is found.

$$\begin{aligned} \frac{\delta^2 f}{\delta \theta^2} &= 2 \cos 2\theta \left[\sum_{i=1}^n (x_i - \bar{x})^2 - \sum_{i=1}^n (y_i - \bar{y})^2 \right] \\ &\quad + 4 \sin 2\theta \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \\ &= g(\theta), \text{ say.} \end{aligned}$$

It can be seen that

$$(i) \quad g(\alpha + \pi/2) = -g(\alpha)$$

$$(ii) \quad g(\alpha) = 0 \text{ if and only if both } \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

$$\text{and } \sum_{i=1}^n (x_i - \bar{x})^2 - \sum_{i=1}^n (y_i - \bar{y})^2 \text{ are zero.}$$

Thus from the assumption (5.4) it is clear that exactly one of

$g(\alpha)$ and $g(\alpha + \pi/2)$ is positive. The corresponding angle (α or $\alpha + \pi/2$), say $\hat{\theta}$, is chosen as the optimum value of θ that minimizes f . Thus, under the assumption (5.4), the best linear fit is well-defined and unique.

Note that if the assumption (5.4) is not true, that is,

if $\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n (x_i - \bar{x})^2 - \sum_{i=1}^n (y_i - \bar{y})^2 = 0$ then

$$\begin{aligned} f(\theta, c) &= \sum_{i=1}^n [(x_i - \bar{x}) \sin \theta - (y_i - \bar{y}) \cos \theta]^2 \\ &= \frac{1}{2} \left[\sum_{i=1}^n (x_i - \bar{x})^2 + \sum_{i=1}^n (y_i - \bar{y})^2 \right] \\ &\quad + \frac{1}{2} \cos 2\theta \left[\sum_{i=1}^n (y_i - \bar{y})^2 - \sum_{i=1}^n (x_i - \bar{x})^2 \right] \\ &\quad - \sin 2\theta \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \\ &= \frac{1}{2} \left[\sum_{i=1}^n (x_i - \bar{x})^2 + \sum_{i=1}^n (y_i - \bar{y})^2 \right] \end{aligned}$$

which is independent of θ . In other words, all straight lines passing through $(\bar{x}, \bar{y})$ are equally good.

So far we have considered only finite sets of points. But clearly the above results are true for uncountable sets of points also provided the first and second order moments exist. In Section 5.5 we will deal with the best linear fit for certain uncountable sets of points in the plane.

5.3 Definition and Shape of a Region

In Chapter 4 we defined open and closed paths and defined open curves on the basis of open paths. Now, in a similar way closed curves can be defined on the basis of closed paths.

Definition 5.3.1 A subset A of the Euclidean plane is called a region if

- (i) A is closed and bounded,
 - (ii) $\text{Int}(A)$ is non-empty and connected,
 - (iii) A is simply connected, that is, A^c is connected,
 - (iv) $\text{Cl}(\text{Int}(A)) = A$
- and (v) $A - \text{Int}(A)$, the boundary of A , can be expressed as a closed curve with finite length.

That is, no holes are permitted inside a region and its outer boundary describes the region uniquely. A region is in fact an analogue bilevel picture with a single component.

Notation $\mathcal{R}$ is the set of all regions.

Definition 5.3.2 R_1 is a relation in $\mathcal{R}$ such that for A and B belonging to $\mathcal{R}$, $(A, B) \in R_1$ if B is a translation of A .

Definition 5.3.3 R_2 is a relation in $\mathcal{R}$ such that for A and B belonging to $\mathcal{R}$, $(A, B) \in R_2$ if B is a dilation of A .

Definition 5.3.4 R_3 is a relation in $\mathcal{R}$ such that for A and B belonging to $\mathcal{R}$, $(A, B) \in R_3$ if B is a rotation of A .

The definitions of translation, dilation and rotation have been given in Chapter 2.

Proposition 5.3.1 R_1, R_2 and R_3 are equivalence relations in $\mathcal{R}$.

Proof It is similar to the one in Chapter 4 (Proposition 4.2.9).

Definition 5.3.5 R is a relation in $\mathcal{R}$ such that for A, B belonging to $\mathcal{R}$, $(A, B) \in R$ if there exist C and D in $\mathcal{R}$ satisfying $(A, C) \in R_1$, $(C, D) \in R_2$ and $(D, B) \in R_3$.

Proposition 5.3.2 R is an equivalence relation in $\mathcal{R}$.

Proof. Same as the one in Chapter 4 (Propositions 4.2.10 and 4.2.11).

Definition 5.3.6 The shape of a region is defined to be an equivalence class generated by R in $\mathcal{R}$.

Notation $\mathcal{S}^2$ is the set of all such equivalence classes, that is, of all shapes.

As in the case of open curves, here also within each equivalence class any two regions can be made to coincide with each other by translation, dilation and rotation. Also, no two regions from two different equivalence classes can be made to coincide by translation, dilation and rotation. In other words,

the shape of a region is what remains of the region after disregarding its position, size and orientation. That is, all squares (also all circles) have the same shape, but the shape of a rectangle (or an ellipse) depends on the ratio of the lengths of its two axes.

5.4 Shape Vector of a Region

In this section we will extract a set of scalar features from a region such that any two regions having the same shape will produce the same values for those features. This set of features can be used for recognition of shapes.

Definition 5.4.1 The centre of gravity of a region A is defined to be the point $c = (c_1, c_2)$ such that

$$c_1 = \int_A x da / \text{area}(A) \text{ and } c_2 = \int_A y da / \text{area}(A)$$

where $\int$ means Riemann integration over A , x and y indicate the co-ordinates of points in A and $\text{area}(A)$ is $\int_A da$.

The centre of gravity may fall inside or outside the region.

Definition 5.4.2 The radius r of a region A is the distance of the farthest point of A from its centre of gravity.

In other words,
$$r = \sup_{(x,y) \in A} \sqrt{(c_1 - x)^2 + (c_2 - y)^2}$$

Note that the radius of a region A is invariant under translation and rotation of A , but not under its dilation.

Now, we will define a mapping from the set of all regions R into the n -dimensional Euclidean space $\mathbb{R}^n$ such that regions with the same shape have the same description in $\mathbb{R}^n$. In other words, the description or mapping does not depend on the translation, dilation and rotation of a region.

We define n concentric rings of equal width around the centre of gravity c in the following way: The k -th ring T_k is the set

$$\{(x, y) \in \mathbb{R}^2 : (k-1)r/n \leq \sqrt{(c_1-x)^2 + (c_2-y)^2} \leq kr/n\}$$

for $k=1, 2, \dots, n$. It is clear that the region R is a subset of $\bigcup_{k=1}^n T_k$ and in fact, a proper subset unless R is a circle. The area of T_k is $2\pi(2k-1)r^2/n^2$. Now, for each ring T_k we define the proportion of the black portion (i.e. of the region R) present in it as the area of $R \cap T_k$ divided by the area of T_k . Let this proportion for the k -th ring be v_k where $1 \geq v_k \geq 0$.

Definition 5.4.3 The n -dimensional vector $(v_1, v_2, \dots, v_n)$ is called the shape vector of the region R and is denoted by $V_n(R)$.

Thus, V_n describes a mapping that transforms a region in the plane into a point in the n -dimensional Euclidean space and hence brings about a great deal of reduction in the dimensionality of the description of a shape. From the construction it is clear that the mapping V_n is invariant under translation, dilation and

rotation of a region. That is, regions having the same shape give rise to the same shape vector V_n . But on the other hand, the mapping V_n is not unique in general. That is, two regions with different shapes may give rise to the same shape vector (Fig.5.2). However, for a limited class of shapes the mapping is often unique and can be used for pattern discrimination (Parui, Sarma and Dutta Majumder, 1983B).

The shape vector of a region is $(1,1,\dots,1)$ if and only if the region is a circular disc. The shape vectors of some standard shapes are given in Table 5.1. Note that the shape vector can be defined for pictures with holes also.

Now, we will define a distance between two shapes on the basis of their shape vectors.

Definition 5.4.4 D_θ is a distance function defined on $\mathcal{G}^2$ such that for two shapes s_1 and s_2 in $\mathcal{G}^2$,

$$D_\theta(s_1, s_2) = \frac{1}{n} \sum_{k=1}^n |v_k^1 - v_k^2|$$

where $(v_1^1, v_2^1, \dots, v_n^1)$ and $(v_1^2, v_2^2, \dots, v_n^2)$ are the shape vectors of s_1 and s_2 respectively.

Note that the value of D_θ depends on the value of n .

Proposition 5.4.1 For any fixed value of n and any shapes s_1, s_2 and s_3 in $\mathcal{G}^2$,

$$(i) \quad D_\theta(s_1, s_2) \geq 0$$

- (ii) $D_8(s_1, s_2) = 0$ if $s_1 = s_2$
- (iii) $D_8(s_1, s_2) = D_8(s_2, s_1)$
- (iv) $D_8(s_1, s_2) + D_8(s_2, s_3) \geq D_8(s_1, s_3)$

Proof. It is trivial .

D_8 does not define a metric on $\mathcal{G}^2$ since the converse of (ii) is not true. D_8 defines a pseudometric on $\mathcal{G}^2$.

A different and more general pseudometric among shapes on the basis of the shape vector is the square root of $(v^1 - v^2)^t A (v^1 - v^2)$ where A is a symmetric positive definite matrix and v^1, v^2 are shape vectors of two shapes.

Definition 5.4.5 μ_3 is a similarity measure between two shapes s_1 and s_2 in $\mathcal{G}^2$ such that

$$\mu_3(s_1, s_2) = 1 - D_8(s_1, s_2)$$

Since D_8 lies between 0 and 1, μ_3 also lies between 0 and 1. If two shapes are the same then the value of μ_3 is 1. In general, if two shapes resemble each other closely, the value of μ_3 is high. The shape vector of a circular disc is $(1, 1, \dots, 1)$. So, the degree of circularity of an arbitrary shape with shape vector $(v_1, v_2, \dots, v_n)$ is $1 - \frac{1}{n} \sum_{k=1}^n (1 - v_k) = \frac{1}{n} \sum_{k=1}^n v_k$.

The values of the similarity measure μ_3 between some standard shapes are given in Table 5.2. The computations are done using Algorithm 5.1 that follows.

Definition 5.4.6 A digital region is a digital binary picture which is simply connected, that is, connected but without holes.

Algorithm 5.1 To calculate the shape vector for a digital region.

Let the picture be represented as a binary matrix $M(i, j)$. That is, $M(i, j) = 0$ or 1 .

Step 1 The total number of 1-pixels in the picture is computed as $NUM = \sum_i \sum_j M(i, j)$. The co-ordinates of the centre of gravity of the picture are computed as $c_1 = \frac{\sum_i i \cdot M(i, j)}{NUM}$ and $c_2 = \frac{\sum_j j \cdot M(i, j)}{NUM}$. We do not truncate or round off c_1 and c_2 to integers.

Step 2 The boundary of the binary picture is scanned (Algorithm 2.0 in Chapter 2). Let B be the set of boundary pixels. Then the radius r of the binary picture is obtained

$$\text{from } r^2 = \max_{(i, j) \in B} [(i - c_1)^2 + (j - c_2)^2]$$

Step 3 The matrix M is extended such that the set $\{(i, j) : (i - c_1)^2 + (j - c_2)^2 \leq r^2\}$ is contained in the extended matrix. Obviously, for the additional pixels (i, j) , $M(i, j) = 0$. Now, each of the 0-pixels is classified into one of the n concentric rings or outside in the following way: a 0-pixel (i, j) belongs to the k -th ring if $(k-1)r^2/n < (i - c_1)^2 + (j - c_2)^2 \leq kr^2/n$ and outside if $(i - c_1)^2 + (j - c_2)^2 > r^2$. Each of the 1-pixels

is classified into one of the n concentric rings in a similar way. Note that no 1-pixel falls outside. Thus for each of the n concentric rings, we know the numbers of 0-pixels and 1-pixels in it.

Step 4 Finally, the shape vector $(v_1, v_2, \dots, v_n)$ is obtained as $v_k = (\text{number of 1-pixels in the } k\text{-th ring}) / (\text{total number of 0-pixels and 1-pixels in the } k\text{-th ring})$.

Now since the outer boundary B determines the picture uniquely, the centre of gravity can be defined on the basis of B only to save computation time. That is, $c_1 = \frac{\sum_{(i,j) \in B} i}{\text{BNUM}}$

and $c_2 = \frac{\sum_{(i,j) \in B} j}{\text{BNUM}}$ where BNUM is the number of pixels on the boundary.

It is to be noted that the description of a shape in terms of its shape vector has a basic inadequacy, i.e., it is not information preserving. As mentioned earlier, different shapes may give rise to the same shape vector. It is clear that the set of different shapes giving rise to the same shape vector reduces as n increases, but it does not tend to be a singleton, in general as n goes to infinity. For example, in Fig.5.2, two different shapes give rise to the same shape vector however large the value of n (even) is. Thus, the shape vector is not an information preserving description. However, for a class of limited number of shapes, the shape vector may be used for shape

discrimination. In the following sections, all the shape descriptions will be information preserving.

5.5 Major Axis and Reference Points of a Region

Let A be a region with the centre of gravity (c_1, c_2) . Let x and y denote the co-ordinates of the points in A . The (Riemann) integrals $\int_A (x-c_1)(y-c_2) da$, $\int_A (x-c_1)^2 da$ and $\int_A (y-c_2)^2 da$ are well-defined and finite. We extend in this section the best linear fit defined for a finite set of points in Section 5.2 to the best linear fit for a region.

Notation Let $\mathcal{R}_1$ be the set of all regions A such that $\int_A (x-c_1)(y-c_2) da = \int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da = 0$. Let $\mathcal{R}^* = \mathcal{R} - \mathcal{R}_1$. Let $\mathcal{G}^3$ be the set of all equivalence classes generated by R in $\mathcal{R}^*$. Note that $\mathcal{G}^3$ is a proper subset of $\mathcal{G}^2$, the difference being the equivalence classes containing circular discs, equilateral triangles, squares etc. Such shapes are called regular shapes. A shape encountered in practice is unlikely to be regular.

Definition 5.5.1 Let $A \in \mathcal{R}^*$. The straight line passing through the centre of gravity (c_1, c_2) of A is called the major axis of A if it makes an angle θ with the x -axis where $\theta \in [0, \pi)$ and

$$\tan 2\theta = \frac{2 \int_A (x-c_1)(y-c_2) da}{\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da} \quad (5.5)$$

(The above equation (5.5) is analogous to (5.3) in Section 5.2).

$$\begin{aligned} \text{and } 2 \cos 2\theta \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right] \\ + 4 \sin 2\theta \int_A (x-c_1)(y-c_2) da = g(\theta) > 0 \end{aligned} \quad (5.6)$$

For a region A in $\mathcal{R}^*$, at least one of the numerator and the denominator in (5.5) is non-zero. In that case, $\tan 2\theta$ is well-defined. From Section 5.2, it is clear that the above major axis is in a sense the best linear fit of the points in the region. Also, a region belonging to $\mathcal{R}^*$ will have a unique major axis. Q_1Q_2 is the major axis of the region shown in Fig. 5.3. Note that if the region is translated or dilated, the slope of the new major axis does not change. Also, the new major axis passes through the centre of gravity of the translated or dilated or rotated region. We will now show that if a region is rotated, the relative position of its major axis does not change.

Notation A_β denotes the region obtained after rotating a region A by an angle β in the anticlockwise direction. Clearly,

$$(A_\beta)_\gamma = A_{\beta+\gamma} \text{ for any } \beta \text{ and } \gamma.$$

Proposition 5.5.1 If a region in $\mathcal{R}^*$ is rotated by an angle β in the anticlockwise direction, its major axis also is rotated by the same angle in the same direction.

Proof. Let A be a region in $\mathcal{R}^*$ with points (x, y) . Let $B = A_\beta$. Let (u, v) be the co-ordinates of the point (x, y) after rotation. Let the polar co-ordinates of the point (x, y) be (r, γ) . So, the polar co-ordinates of the point (u, v) are $(r, \beta + \gamma)$. B is the set of the points (u, v) and

$$u = r \cos(\beta + \gamma) = x \cos \beta - y \sin \beta$$

$$v = r \sin(\beta + \gamma) = x \sin \beta + y \cos \beta$$

It is easy to see that the centre of gravity of B is $(c'_1, c'_2) = (c_1 \cos \beta - c_2 \sin \beta, c_1 \sin \beta + c_2 \cos \beta)$. So, we have to show that

$$(i) \quad \tan 2(\theta + \beta) = \frac{2 \int_B (u - c'_1)(v - c'_2) db}{\int_B (u - c'_1)^2 db - \int_B (v - c'_2)^2 db} \quad (5.7)$$

and (ii) $g(\theta + \beta)$ is positive (after rotation)

$$\text{where } \tan 2\theta = \frac{2 \int_A (x - c_1)(y - c_2) da}{\int_A (x - c_1)^2 da - \int_A (y - c_2)^2 da}$$

and $g(\theta) > 0$ (before rotation).

That is, if the major axis of A makes angle θ with the x -axis, then the major axis of B makes the angle $\theta + \beta$ with the x -axis.

$$\begin{aligned}
(i) \quad & \int_B (u-c'_1)(v-c'_2) db \\
&= \int_A [(x-c_1)\cos\beta - (y-c_2)\sin\beta] [(x-c_1)\sin\beta + (y-c_2)\cos\beta] da \\
&= \cos^2\beta \int_A (x-c_1)(y-c_2) da - \sin^2\beta \int_A (x-c_1)(y-c_2) da \\
&\quad + \sin\beta\cos\beta \int_A (x-c_1)^2 da + \sin\beta\cos\beta \int_A (y-c_2)^2 da \\
&= \cos 2\beta \int_A (x-c_1)(y-c_2) da \\
&\quad + \frac{\sin 2\beta}{2} \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right] \quad (5.8)
\end{aligned}$$

$$= \frac{\cos 2\beta}{2} \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right] [\tan 2\theta + \tan 2\beta] \text{ from (5.5)}$$

$$\begin{aligned}
& \int_B (u-c'_1)^2 db \\
&= \int_A [(x-c_1)\cos\beta - (y-c_2)\sin\beta]^2 da \\
&= \cos^2\beta \int_A (x-c_1)^2 da + \sin^2\beta \int_A (y-c_2)^2 da \\
&\quad - \sin 2\beta \int_A (x-c_1)(y-c_2) da
\end{aligned}$$

$$\text{Similarly, } \int_B (v-c'_2)^2 db$$

$$\begin{aligned}
&= \cos^2\beta \int_A (y-c_2)^2 da + \sin^2\beta \int_A (x-c_1)^2 da \\
&\quad + \sin 2\beta \int_A (x-c_1)(y-c_2) da
\end{aligned}$$

$$\begin{aligned}
\text{So, } \int_B (u-c'_1)^2 db &= \int_B (v-c'_2)^2 db \\
&= \cos 2\beta \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right] \\
&\quad - 2\sin 2\beta \int_A (x-c_1)(y-c_2) da \quad (5.9) \\
&= \cos 2\beta \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right] [1 - \tan 2\beta \tan 2\theta] \\
&\quad \text{from (5.5)}
\end{aligned}$$

Thus, R.H.S. of (5.7)

$$\begin{aligned}
&\frac{\cos 2\beta \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right] [\tan 2\theta + \tan 2\beta]}{\cos 2\beta \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right] [1 - \tan 2\beta \tan 2\theta]} \\
&= \tan 2(\theta + \beta)
\end{aligned}$$

(ii) After rotation

$$\begin{aligned}
g(\theta + \beta) &= 2\cos 2(\theta + \beta) \left[\int_B (u-c'_1)^2 db - \int_B (v-c'_2)^2 db \right] \\
&\quad + 4\sin 2(\theta + \beta) \int_B (u-c'_1)(v-c'_2) db \\
&= \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right] \\
&\quad [2\cos 2(\theta + \beta)\cos 2\beta + 2\sin 2(\theta + \beta)\sin 2\beta]
\end{aligned}$$

$$+ \int_A (x-c_1)(y-c_2) da [4\sin 2(\theta+\beta)\cos 2\beta - 4\cos 2(\theta+\beta)\sin 2\beta]$$

from (5.8) and (5.9)

$$= 2\cos 2\theta \left[\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da \right]$$

$$+ 4\sin 2\theta \int_A (x-c_1)(y-c_2) da$$

$$= g(\theta) \quad (\text{before rotation})$$

$$> 0.$$

Hence Proposition 5.5.1.

Thus for a region in $\mathcal{R}^*$, the major axis is unique and the relative position of the major axis is invariant under translation, dilation and rotation of the region. Consequently, the relative positions of the points of intersection of the major axis with the boundary of the region are also invariant under these operations.

Definition 5.5.2 The two farthest points among these points of intersection are called the reference points of a region (Q_1 and Q_2 are the reference points of the region in Fig.5.3).

It is clear that the relative positions (relative to the region) of the reference points are unique for a region in $\mathcal{R}^*$ and do not depend on the position, size and orientation of the region.

For regular shapes, the major axis and hence the reference points are not unique. Since perfectly regular shapes are

unlikely to occur in practice, they will not be considered in Sections 5.6 and 5.7.

5.6 Shape Distance on the Basis of Unmatched Area

In this section, we will propose a distance function among shapes of regions which defines a metric on $\mathcal{S}^3$. Since the shape of a region does not depend on its position and size, we will now consider only the regions (in $\mathcal{R}^*$) which have a fixed area a (positive) and whose centre of gravity is the origin $(0,0)$. Let $\mathcal{R}_2$, a subset of $\mathcal{R}^*$, be the set of all such regions.

Definition 5.6.1 D_g is a distance function on $\mathcal{R}_2$ such that for A, B in $\mathcal{R}_2$

$$D_g(A, B) = \text{Area} \{ (A-B) \cup (B-A) \} \quad (5.10)$$

(Area of a subset in the plane is, as indicated earlier, the Riemann integral of the indicator function over A).

Proposition 5.6.1 D_g defines a metric on $\mathcal{R}_2$. That is for A, B and C in $\mathcal{R}_2$.

- (i) $D_g(A, B) \geq 0$
- (ii) $D_g(A, B) = 0$ if and only if $A=B$
- (iii) $D_g(A, B) = D_g(B, A)$
- (iv) $D_g(A, B) + D_g(B, C) \geq D_g(A, C)$

Proof. (i) Trivial.

(ii) $A = B$ implies $D_9(A, B) = 0$. The proof for the converse is in Appendix C11.

(iii) Trivial.

$$\begin{aligned}
 \text{(iv)} \quad & \text{Area} \{ (A-B) \cup (B-A) \} + \text{Area} \{ (B-C) \cup (C-B) \} \\
 &= \text{Area}(A-B) + \text{Area}(B-A) + \text{Area}(B-C) + \text{Area}(C-B) \\
 &\geq \text{Area} \{ (A-B) \cup (B-C) \} + \text{Area} \{ (B-A) \cup (C-B) \} \\
 &\geq \text{Area}(A-C) + \text{Area}(C-A), \text{ since } (A-C) \text{ and } (C-A) \text{ are subsets of} \\
 &\quad (A-B) \cup (B-C) \text{ and } (B-A) \cup (C-B) \text{ respectively} \\
 &\geq \text{Area} \{ (A-C) \cup (C-A) \} \\
 &= D_9(A, C)
 \end{aligned}$$

Proposition 5.6.2 $D_9(A, B) = D_9(A_\beta, B_\beta)$ for any β .

Proof It can be seen that $\text{Area}(C) = \text{Area}(C_\beta)$ for any subset C in the plane. Now, $D_9(A_\beta, B_\beta) = \text{Area} \{ (A_\beta - B_\beta) \cup (B_\beta - A_\beta) \}$
 $= \text{Area} \{ (A-B)_\beta \cup (B-A)_\beta \} = \text{Area} \{ (A-B) \cup (B-A) \} = D_9(A, B)$.

$D_9(A, B)$ is, in fact, the unmatched area between two regions A and B . Now, since the regions in $\mathcal{R}_2$ have a fixed position and size, D_9 is already position and size invariant. But D_9 is not rotation invariant. We will modify D_9 to D_{10} in order to achieve rotational invariance.

Definition 5.6.2 D_{10} is a distance function on $\mathcal{R}_2$ such that for A, B in $\mathcal{R}_2$

$$D_{10}(A, B) = \inf_{\beta \in [0, 2\pi)} D_9(A, B_\beta) \quad (5.11)$$

Proposition 5.6.3 $D_{10}(A, B) = D_9(A, B_\gamma)$ for some γ in $[0, 2\pi)$.

Proof It is clear from Appendix B.6.

Proposition 5.6.4 For A, B and C belonging to $\mathcal{R}_2$

- (i) $D_{10}(A, B) \geq 0$
- (ii) $D_{10}(A, B) = 0$ if and only if A is a rotation of B
- (iii) $D_{10}(A, B) = D_{10}(B, A)$
- (iv) $D_{10}(A, B) + D_{10}(B, C) \geq D_{10}(A, C)$

Proof (i) Trivial.

(ii) Suppose $D_{10}(A, B) = 0$. So, there is a β in $[0, 2\pi)$ such that $D_9(A, B_\beta) = 0$. Then from (ii) of Proposition 5.6.1 $A = B_\beta$. Hence, A is a rotation of B . Conversely, if A is a rotation of B , then there is a γ in $[0, 2\pi)$ such that $A = B_\gamma$. So, $D_9(A, B_\gamma) = 0$. Hence $D_{10}(A, B) = 0$.

$$\begin{aligned}
 \text{(iii)} \quad D_{10}(A, B) &= \inf_{\beta} D_9(A, B_\beta) \\
 &= \inf_{\beta} D_9(A_{2\pi-\beta}, B) \text{ from Proposition 5.6.2} \\
 &= \inf_{\gamma} D_9(A_\gamma, B) \\
 &= \inf_{\gamma} D_9(B, A_\gamma) \text{ from Proposition 5.6.1} \\
 &= D_{10}(B, A)
 \end{aligned}$$

$$\begin{aligned}
 \text{(iv)} \quad D_{10}(A, B) + D_{10}(B, C) &= D_9(A, B_\beta) + D_9(B, C_\gamma) \text{ for some } \beta \text{ and } \gamma \\
 &= D_9(A, B_\beta) + D_9(B_\beta, C_{\gamma+\beta}) \text{ from Proposition 5.6.2} \\
 &\geq D_9(A, C_{\gamma+\beta}) \text{ since } D_9 \text{ satisfies triangular inequality} \\
 &\geq D_{10}(A, C)
 \end{aligned}$$

Two regions A and B in $\mathcal{R}_2$ have the same shape if and only if A is a rotation of B . Hence, Proposition 5.6.4 leads to the following proposition.

Proposition 5.6.5 D_{10} defines a metric on $\mathcal{G}^3$.

To prove unambiguity, we have to show, for A, B, C belonging to $\mathcal{R}_2$, $D_{10}(A, B) = D_{10}(A, C)$ if C is a rotation of B (i.e., if B and C have the same shape). Let $C = B_\beta$. So, $D_{10}(A, C) = \inf_{\gamma} D_9(A, C_\gamma) = \inf_{\gamma} D_9(A, B_{\beta+\gamma}) = \inf_{\theta} D_9(A, B_\theta) = D_{10}(A, B)$.

The definition of D_{10} in (5.11) considers all possible rotations of the region B . In that case, the computations to find the value of D_{10} will be quite costly in terms of time. So, we propose another definition of shape distance D_{11} in (5.12) with the help of the major axis of a region. In that case, only two rotations of B are considered so that there is much reduction in computation time.

Definition 5.6.3 D_{11} is a distance function on $\mathcal{R}_2$ such that for A, B in $\mathcal{R}_2$

$$D_{11}(A, B) = \min \{ D_9(A, B_{\theta_1 - \theta_2}), D_9(A, B_{\theta_1 - \theta_2 + \pi}) \} \quad (5.12)$$

where θ_1 and θ_2 are the angles that the major axes of A and B respectively make with the x -axis. That is, B is rotated in such a way that its major axis coincides with that of A with the angle between the two major axes being either 0 or π and D_{11} is the smaller of the two D_9 values in those two orientations.

Our conjecture is that

$$D_{11}(A, B) = D_{10}(A, B)$$

when the regions A, B are convex.

Proposition 5.6.6 For A, B and C in $\mathcal{R}_2$,

- (i) $D_{11}(A, B) \geq 0$
- (ii) $D_{11}(A, B) = 0$ if and only if A is a rotation of B
- (iii) $D_{11}(A, B) = D_{11}(B, A)$
- (iv) $D_{11}(A, B) + D_{11}(B, C) \geq D_{11}(A, C)$

Proof (i) Trivial.

$$(ii) \quad D_{11}(A, B) = 0$$

$$\Rightarrow D_9(A, B_{\theta_1 - \theta_2}) = 0 \quad \text{or} \quad D_9(A, B_{\theta_1 - \theta_2 + \pi}) = 0$$

$$\Rightarrow A = B_{\theta_1 - \theta_2} \quad \text{or} \quad A = B_{\theta_1 - \theta_2 + \pi} \quad \text{from (ii)}$$

in Proposition 5.6.1

$$\Rightarrow A \text{ is a rotation of } B$$

Now, note that if a region is rotated by angle π , the value of θ of its major axis remains unchanged. That is, if $A = B_\beta$ then either $\theta_1 - \theta_2 = \beta$ or $\theta_1 - \theta_2 + \pi = \beta$.

So conversely,

A is a rotation of B

$$\Rightarrow A = B_{\theta_1 - \theta_2} \quad \text{or} \quad A = B_{\theta_1 - \theta_2 + \pi}$$

$$\Rightarrow D_9(A, B_{\theta_1 - \theta_2}) = 0 \quad \text{or} \quad D_9(A, B_{\theta_1 - \theta_2 + \pi}) = 0$$

$$\Rightarrow D_{11}(A, B) = 0$$

$$(iii) \quad D_{11}(B, A)$$

$$= \text{Min} \left\{ D_9(B, A_{\theta_2 - \theta_1}), D_9(B, A_{\theta_2 - \theta_1 + \pi}) \right\}$$

$$= \text{Min} \left\{ D_9(B_{\theta_1 - \theta_2}, A), D_9(B_{\theta_1 - \theta_2 + \pi}, A) \right\},$$

from Proposition 5.6.2

$$= \text{Min} \left\{ D_9(A, B_{\theta_1 - \theta_2}), D_9(A, B_{\theta_1 - \theta_2 + \pi}) \right\}$$

since D_9 is symmetric

$$= D_{11}(A, B)$$

(iv) There are four cases, that is, $D_{11}(A, B)$ is either $D_9(A, B_{\theta_1 - \theta_2})$ or $D_9(A, B_{\theta_1 - \theta_2 + \pi})$ and $D_{11}(B, C)$ is either $D_9(B, C_{\theta_2 - \theta_3})$ or $D_9(B, C_{\theta_2 - \theta_3 + \pi})$ where θ_3 is the angle that the major axis of C makes with the x-axis.

For the first case,

$$D_9(A, B_{\theta_1 - \theta_2}) + D_9(B, C_{\theta_2 - \theta_3})$$

$$= D_9(A, B_{\theta_1 - \theta_2}) + D_9(B_{\theta_1 - \theta_2}, C_{\theta_1 - \theta_3})$$

from Proposition 5.6.2

$$\geq D_9(A, C_{\theta_1 - \theta_3}), \text{ since } D_9 \text{ follows triangular inequality}$$

$$\geq D_{11}(A, C)$$

The proofs for the other three cases will be similar.

Since two regions A and B in $\mathcal{R}_2$ have the same shape if and only if A is a rotation of B , Proposition 5.6.6 leads to the following proposition.

Proposition 5.6.7 D_{11} defines a metric on $\mathcal{G}^3$.

To prove unambiguity, we have to show, for A, B, C belonging to $\mathcal{R}_2$, $D_{11}(A, B) = D_{11}(A, C)$ if B is a rotation of C (i.e. if B and C have the same shape). Let $C = B_\beta$.

Then $\theta_3 = \theta_2 + \beta$ from Proposition 5.5.1. So,

$$\begin{aligned} D_{11}(A, C) &= \text{Min} \{ D_g(A, C_{\theta_1 - \theta_3}), D_g(A, C_{\theta_1 - \theta_3 + \pi}) \} \\ &= \text{Min} \{ D_g(A, B_{\theta_1 - \theta_3 + \beta}), D_g(A, B_{\theta_1 - \theta_3 + \pi + \beta}) \} \\ &= \text{Min} \{ D_g(A, B_{\theta_1 - \theta_2}), D_g(A, B_{\theta_1 - \theta_2 + \pi}) \} \\ &= D_{11}(A, B). \end{aligned}$$

Proposition 5.6.8 $\sup_{A, B \in \mathcal{R}_2} D_{11}(A, B) \leq 2a$.

Proof. Recall that each region in $\mathcal{R}_2$ has area a .

So, $\text{Area}(A-B) \leq a$ and $\text{Area}(B-A) \leq a$.

$$\begin{aligned} \text{Thus, } D_g(A, B) &= \text{Area}(A-B) + \text{Area}(B-A) \\ &\leq 2a \end{aligned}$$

Consequently, $D_{11}(A, B) \leq 2a$.

Definition 5.6.4 μ_4 is a similarity measure between the shapes of two regions A and B in $\mathcal{R}_2$ such that

$$\mu_4(A, B) = 1 - D_{11}(A, B)/2a \quad (5.13)$$

Clearly, $1 \geq \mu_4 \geq 0$. Also, $\mu_4(A, B) = 1$ if and only if A and B have the same shape.

Computations (Parui, 1983E)

Now, we will present the computational techniques to find the values of D_{11} and μ_4 for digital regions.

Let S be a digital region. It can be represented as the set of its 1-pixels $\{(i_k, j_k) : k=1, 2, \dots, K\}$.

Definition 5.6.5 The centroid of S is (r, s) where

$$r = \frac{\sum_{k=1}^K i_k}{K} \text{ and } s = \frac{\sum_{k=1}^K j_k}{K} \text{ and } r, s \text{ are not necessarily integers.}$$

Definition 5.6.6 The major axis of S is the straight line passing through the centroid of S and making an angle β with the y -axis (i.e. with the vertical line since (i_k, j_k) are the co-ordinates of a matrix) where

$$(i) \quad \tan 2\beta = \frac{2 \sum_{k=1}^K (i_k - r)(j_k - s)}{\sum_{k=1}^K (i_k - r)^2 - \sum_{k=1}^K (j_k - s)^2}$$

$$\text{and (ii)} \quad \cos 2\beta \left[\sum_{k=1}^K (i_k - r)^2 - \sum_{k=1}^K (j_k - s)^2 \right]$$

$$+ 2 \sin 2\beta \sum_{k=1}^K (i_k - r)(j_k - s) > 0$$

Let S_1 and S_2 be two digital regions between which the value of the shape distance D_{11} is to be calculated. Let

$$\text{Area}(S_1) = K_1 \text{ and } \text{Area}(S_2) = K_2.$$

If $K_1 = K_2$, then propagation is not necessary for matching. Otherwise, assume $K_2 > K_1$. The smaller region S_1 is propagated by the factor $\sqrt{K_2/K_1}$ by Algorithm 2.1 proposed in Chapter 2.

Let S_3 be this propagation of S_1 . The areas of S_2 and S_3 will be approximately the same. Suppose, the centroids of S_2 and S_3 are (r_1, s_1) and (r_2, s_2) respectively. Let i_1, j_1, i_2 and j_2 be the nearest integers of r_1, s_1, r_2 and s_2 respectively. Let S_4 and S_5 be the translations of S_2 and S_3 by (i_1, j_1) and (i_2, j_2) respectively so that the centroids of S_4 and S_5 coincide (at least approximately) with $(0,0)$.

Suppose, β and γ are the angles that the major axes of S_4 and S_5 respectively make with the vertical line. A digital region is rotated with the help of the algorithm proposed in Chapter 2. Let S_6 and S_7 be the rotations of S_5 by angles $(\beta-\gamma)$ and $(\beta-\gamma+\pi)$ respectively. The pixels of S_7 can be directly obtained from the pixels of S_6 , since $(i_k, j_k) \in S_6$ if and only if $(-i_k, -j_k) \in S_7$. Note that (i) the areas of S_4, S_6 and S_7 are almost the same, (ii) the centroids of S_4, S_6 and S_7 almost coincide with $(0,0)$ and (iii) the major axis of S_4 almost coincides with the major axes of S_6 and S_7 .

Suppose, the binary picture matrices of S_4, S_6 and S_7 are N_1, N_2 and N_3 respectively. Let

$D_9^*(S_4, S_6)$ = the number of pixels (i, j) such that

$$N_1(i, j) \neq N_2(i, j)$$

$D_9^*(S_4, S_7)$ = the number of pixels (i, j) such that
 $N_1(i, j) \neq N_3(i, j)$

D_9^* is the discrete form of D_9 defined in (5.10) and gives the unmatched areas between S_4 and S_6 and S_4 and S_7 above.

Let $D_{11}^*(S_4, S_5) = \text{Min} \{ D_9^*(S_4, S_6), D_9^*(S_4, S_7) \}$
 and $\mu_4^*(S_4, S_5) = 1 - D_{11}^*(S_4, S_5) / (\text{Area}(S_4) + \text{Area}(S_5))$

D_{11}^* and μ_4^* are the discrete forms of D_{11} and μ_4 respectively. Note that $\text{Area}(S_4)$ and $\text{Area}(S_5)$ are equal to $\text{Area}(S_2)$ and $\text{Area}(S_3)$ respectively which have already been calculated.

5.7 Two Shape Distances on the Basis of Region Boundaries

Notations: If A is a region, $\bar{A}$ will denote the boundary of A . From the definition of a region (Definition 5.3.1), $\bar{A}$ is a closed curve. $\mathcal{L}^*$ will denote the set of all closed curves which are boundaries of the regions belonging to $\mathcal{R}^*$.

There is a one-to-one correspondence between $\mathcal{R}^*$ and $\mathcal{L}^*$. This is why we will essentially consider in this section only closed curves of $\mathcal{L}^*$ instead of regions of $\mathcal{R}^*$. Consequently, shape distances for regions will be defined on the basis of closed curves of $\mathcal{L}^*$.

It can be seen that a closed curve alongwith any fixed

point Q on it may be looked upon as an open curve in the sense that Q may be treated as the starting point as well as the end point of the curve. This is necessary because as will be clear later on, the shape distances defined for open curves in Chapter 4 will be made use of in this section. So, the problem is about the choice of the point Q on a closed curve belonging to $\mathcal{L}^*$. This problem can be solved in the following way. Any region A in $\mathcal{R}^*$ will have two reference points Q_1 and Q_2 (defined in Section 5.5) which lie on the boundary $\bar{A}$. Each Q_i can be used as the fixed point Q on the closed curve $\bar{A}$. Another problem is about the direction (clockwise or anticlockwise) in which we scan $\bar{A}$. To remove the ambiguity, we always scan a closed curve in $\mathcal{L}^*$ in the clockwise direction. Let A^i be the curve obtained by scanning the boundary $\bar{A}$ of a region A from (and ending at) the point Q_i in the clockwise direction ($i=1,2$). That is, A^i can be represented as $\{(u_i(\lambda L), v_i(\lambda L)) : \lambda \in [0, 1]\}$ where $(u_i(0), v_i(0)) = (u_i(L), v_i(L))$ is the point Q_i , u_i and v_i are real valued functions defined in Chapter 4 and L is the length of the closed curve $\bar{A}$ (scanning direction is clockwise). Though any one of A^1 and A^2 describes the boundary $\bar{A}$ of A uniquely, we will represent $\bar{A}$ as the set of two curves $\{A^1, A^2\}$. The reason for this will be clear later when shape distances are defined in this section.

We assume that the functions u_i and v_i here have the same properties as they have in the case of open curves in Chapter 4. We will now define a shape distance for regions on the basis of the slopes of their boundaries. Let two regions A and B belonging to $\mathcal{R}^*$ have boundaries $\bar{A}$ and $\bar{B}$. Suppose, the open curve representations of the boundaries are $\bar{A} = \{A^1, A^2\}$ and $\bar{B} = \{B^1, B^2\}$. Let the slope of A^i be $\{w_{1i}(\lambda) : \lambda \in (0, 1)\}$ and that of B^i be $\{w_{2i}(\lambda) : \lambda \in (0, 1)\}$ for $i = 1, 2$ (slope has been defined in Chapter 4). Recall the definition of D_3 in Chapter 4 (Definition 4.3.6). According to that definition, $D_3(A^1, B^1)$, $D_3(A^1, B^2)$, $D_3(A^2, B^1)$ and $D_3(A^2, B^2)$ are well-defined.

Proposition 5.7.1 B is a translation/dilation/rotation of A if and only if $\bar{B}$ is a translation/dilation/rotation of $\bar{A}$ if and only if either B^1 is a translation/dilation/rotation of A^1 and B^2 is a translation/dilation/rotation of A^2 or B^1 is a translation/dilation/rotation of A^2 and B^2 is a translation/dilation/rotation of A^1 .

Proof. It is trivial.

Definition 5.7.1 D_{12} is a distance function on $\mathcal{R}^*$ such that for A and B belonging to $\mathcal{R}^*$

$$D_{12}(A, B) = \min \left\{ D_3(A^1, B^1) + D_3(A^2, B^2), \right. \\ \left. D_3(A^1, B^2) + D_3(A^2, B^1) \right\} \quad (5.14)$$

Proposition 5.7.2 For A, B and C belonging to $\mathcal{Q}^*$,

- (i) $D_{12}(A, B) \geq 0$
- (ii) $D_{12}(A, B) = 0$ if and only if A and B have the same shape
- (iii) $D_{12}(A, B) = D_{12}(B, A)$
- (iv) $D_{12}(A, B) + D_{12}(B, C) \geq D_{12}(A, C)$

Proof (i) Trivial.

(ii) Suppose A and B have the same shape. So, from Proposition 5.7.1, either A^1 and B^1 have the same shape and A^2 and B^2 have the same shape or A^1 and B^2 have the same shape and A^2 and B^1 have the same shape. Again, from Proposition 4.3.12, either $D_3(A^1, B^1) = D_3(A^2, B^2) = 0$ or $D_3(A^1, B^2) = D_3(A^2, B^1) = 0$. Thus, $D_{12}(A, B) = 0$.

Conversely, suppose $D_{12}(A, B) = 0$. That is, either $D_3(A^1, B^1) = D_3(A^2, B^2) = 0$ or $D_3(A^1, B^2) = D_3(A^2, B^1) = 0$. So, from Proposition 4.3.12, either A^1 and B^1 have the same shape and A^2 and B^2 have the same shape or A^1 and B^2 have the same shape and A^2 and B^1 have the same shape.

Again from Proposition 5.7.1, A and B have the same shape.

- (iii) This is true since D_3 is symmetric (Proposition 4.3.12).
- (iv) Suppose, $D_{12}(A, B) = D_3(A^1, B^1) + D_3(A^2, B^2)$
and $D_{12}(B, C) = D_3(B^1, C^1) + D_3(B^2, C^2)$. (The other three cases can be treated in a similar way).

$$\begin{aligned}
\text{So, } D_{12}(A,B) + D_{12}(B,C) \\
&\geq D_3(A^1, C^1) + D_3(A^2, C^2) \text{ since triangular inequality} \\
&\quad \text{holds for } D_3 \\
&\geq D_{12}(A,C)
\end{aligned}$$

Hence Proposition 5.7.2.

It is clear from the above proposition that D_{12} defines a metric on $\mathcal{G}^3$.

Proposition 5.7.3 $D_{12}(A,B) \leq 4$ for all A and B in $\mathcal{R}^*$.

Proof. It is true because from Proposition 4.3.17, $D_3(A^r, B^s) \leq 2$ for $2 \geq r, s \geq 1$.

Definition 5.7.2. μ_5 is a similarity measure between two shapes in $\mathcal{G}^3$ such that

$$\mu_5(A,B) = 1 - D_{12}(A,B)/4 \quad (5.15)$$

Clearly, $1 \geq \mu_5 \geq 0$. Note that $\mu_5(A,B) = 1$ if and only if A and B have the same shape.

We shall now develop another shape distance measure for regions on the basis of their boundaries. Since the shape of a region does not depend on its position and size, we shall now consider only the regions whose boundaries have their centre of gravity at the origin $(0,0)$ and have a fixed length, say, L . Let $\mathcal{R}_3$ be the set of all such regions. Let A and B be two regions belonging to $\mathcal{R}_3$. Suppose, $\bar{A}$ and $\bar{B}$ are boundaries

of A and B such that $\bar{A} = \{A^1, A^2\}$ and $\bar{B} = \{B^1, B^2\}$. Recall the definition of D_6 in Chapter 4 (Definition 4.3.10).

According to that definition, $D_6(A^1, B^1)$, $D_6(A^1, B^2)$, $D_6(A^2, B^1)$ and $D_6(A^2, B^2)$ are well-defined.

Definition 5.7.3 D_{13} is a distance function on $\mathcal{R}_3$ such that for A and B belonging to $\mathcal{R}_3$

$$D_{13}(A, B) = \min \{ D_6(A^1, B^1) + D_6(A^2, B^2), D_6(A^1, B^2) + D_6(A^2, B^1) \} \quad (5.16)$$

Proposition 5.7.4 For A, B and C belonging to $\mathcal{R}_3$,

- (i) $D_{13}(A, B) \geq 0$
- (ii) $D_{13}(A, B) = 0$ if and only if B is a rotation of A
- (iii) $D_{13}(A, B) = D_{13}(B, A)$
- (iv) $D_{13}(A, B) + D_{13}(B, C) \geq D_{13}(A, C)$

Proof. (i) Trivial.

(ii) Suppose, B is a rotation of A . So, from the proof of Proposition 5.7.1, either B^1 is a rotation of A^1 and B^2 is a rotation of A^2 or B^2 is a rotation of A^1 and B^1 is a rotation of A^2 . So, from Proposition 4.3.26, either $D_6(A^1, B^1) = D_6(A^2, B^2) = 0$ or $D_6(A^1, B^2) = D_6(A^2, B^1) = 0$. Thus $D_{13}(A, B) = 0$. Conversely, suppose $D_{13}(A, B) = 0$. That is, either $D_6(A^1, B^1) = D_6(A^2, B^2) = 0$ or $D_6(A^1, B^2) = D_6(A^2, B^1) = 0$. So, from Proposition 4.3.26, either B^1 is a rotation of A^1 and B^2 is a rotation of A^2 or B^1 is a rotation of A^2 and B^2 is a rotation

of A^1 .

So from Proposition 5.7.1, B is a rotation of A .

(iii) This is true since D_6 is symmetric (Proposition 4.3.26).

(iv) Suppose, $D_{13}(A, B) = D_6(A^1, B^1) + D_6(A^2, B^2)$ and

$D_{13}(B, C) = D_6(B^1, C^1) + D_6(B^2, C^2)$. (The other three cases can be treated in a similar way).

So, $D_{13}(A, B) + D_{13}(B, C)$

$\geq D_6(A^1, C^1) + D_6(A^2, C^2)$ since triangular inequality holds

for D_6

$\geq D_{13}(A, C)$

Hence Proposition 5.7.4.

Now, it can be seen that two regions A and B belonging to $\mathcal{R}_3$ have the same shape if and only if B is a rotation of A . Thus, from the above proposition it is clear that D_{13} defines a metric on $\mathcal{S}^3$.

Proposition 5.7.5 $D_{13}(A, B) \leq 2/\sqrt{6}$ for all A and B in $\mathcal{R}_3$.

Proof. This is because from Proposition 4.3.28

$D_6(A^r, B^s) \leq 1/\sqrt{6}$ for $2 \geq r, s \geq 1$.

Definition 5.7.4 μ_6 is a similarity measure between two shapes in $\mathcal{S}^3$ such that

$$\mu_6(A, B) = 1 - \sqrt{6} D_{13}(A, B)/2 \quad (5.17)$$

Clearly, $1 \geq \mu_6 \geq 0$. Note that $\mu_6(A, B) = 1$ if and only if A and B have the same shape.

5.8 Conclusion

It is clear that the computation of D_8 or D_{11} takes more time than that of D_{12} or D_{13} . D_8 and D_{11} involve computations of order n^2 while D_{12} and D_{13} involve computations of order n . The values of shape similarities μ_3 , μ_4 , μ_5 and μ_6 defined on the basis of shape distances D_8 , D_{11} , D_{12} and D_{13} respectively for some standard shapes are given in Tables 5.2 and 5.3. Shapes of rectangles and ellipses with different parameters are considered. Shape similarities are computed between each of these shapes and a circle. It can be seen that the values of μ_3 , μ_4 , μ_5 and μ_6 increase monotonically as the ratio of the major axis to the minor axis of a rectangle or an ellipse decreases. These results are consistent with the human visual judgement.

As far as the extension of shape distance from 2 dimensions to 3 dimensions is concerned, D_8 and D_{11} have an advantage over D_{12} and D_{13} . Boundary in 2 dimensions becomes surface in 3 dimensions and it is quite difficult to define 3-dimensional shape distance on the basis of surfaces. On the other hand, it is easy to extend D_8 to 3 dimensions, since it is possible to define shape vector in 3 dimensions on the basis of concentric 3-dimensional rings around the centre of gravity. In order to extend D_{11} to 3 dimensions, one has to define the major axis and major plane for a 3-dimensional

object. The major axis is the line around which the moment of inertia is the minimum. The major plane is such that the sum of squared perpendicular distances from all points of the object to that plane is the minimum. The major axis and major plane can be used to normalize the orientation of 3 dimensional objects. Now, the propagation and shrinking algorithms proposed in Chapter 2 in this thesis can easily be extended to 3 dimensions. And these propagation and shrinking algorithms can be used to normalize the size of 3 dimensional objects. Position can be normalized on the basis of the centre of gravity. Finally, the shape distance between two 3 dimensional objects can be defined as the unmatched volume after normalization of position, size and orientation, in the same way as D_{11} has been defined in the case of 2 dimensions.

In the case of regular shapes like circle, square and other regular polygons, D_{11} , D_{12} and D_{13} have not been defined, since such a regular shape does not have a unique major axis and hence does not have a unique orientation. R_1 contains regions of such regular shapes. Now, a region with almost regular shape (not belonging to R_1) also may pose some problem. Slight noise in such a shape may drastically change the major axis. Such instability of the major axis is shown in the case of nearly equilateral triangular shapes in Fig.5.4.

In order to find the major axis of a region, we need computations of order n^2 if all the pixels of the region are

considered. But if we find the major axis on the basis of only the boundary pixels of a region, computations will be of order n . However, the major axis found on the basis of the boundary pixels has a drawback. It is more sensitive to noise than the major axis found on the basis of all pixels. For the rectangle in Fig. 5.5(a), the major axis is the horizontal line passing through the centre of gravity (whether it is defined on the basis of all pixels or on the basis of only boundary pixels). Some noise is added to the top right part of the rectangle. The major axis gets disturbed accordingly. In Fig. 5.5(b), is shown the major axis defined on the basis of all the pixels of the noisy rectangle, while in Fig. 5.5(c) is shown the major axis defined on the basis of only the boundary pixels of the noisy rectangle. Disturbance is more in Fig. 5.5(c) than in Fig. 5.5(b) as far as the position of the major axis is concerned.

So far we have considered shape to be invariant under translation, dilation and rotation. The definition of shape can be extended by making shape invariant under reflection also. Reflection or mirror image of a region A is the set $\text{Ref1}(A) = \{(x, -y) : (x, y) \in A\}$. Of the four shape distances defined in this chapter, D_8 is already invariant under reflection, since the shape vector of a region is invariant under reflection. $D_{11}(A, B)$,

$D_{12}(A, B)$ and $D_{13}(A, B)$ can be modified as $\text{Min} \{D_{11}(A, B), D_{11}(A, \text{Ref1}(B))\}$, $\text{Min} \{D_{12}(A, B), D_{12}(A, \text{Ref1}(B))\}$ and $\text{Min} \{D_{13}(A, B), D_{13}(A, \text{Ref1}(B))\}$ respectively.

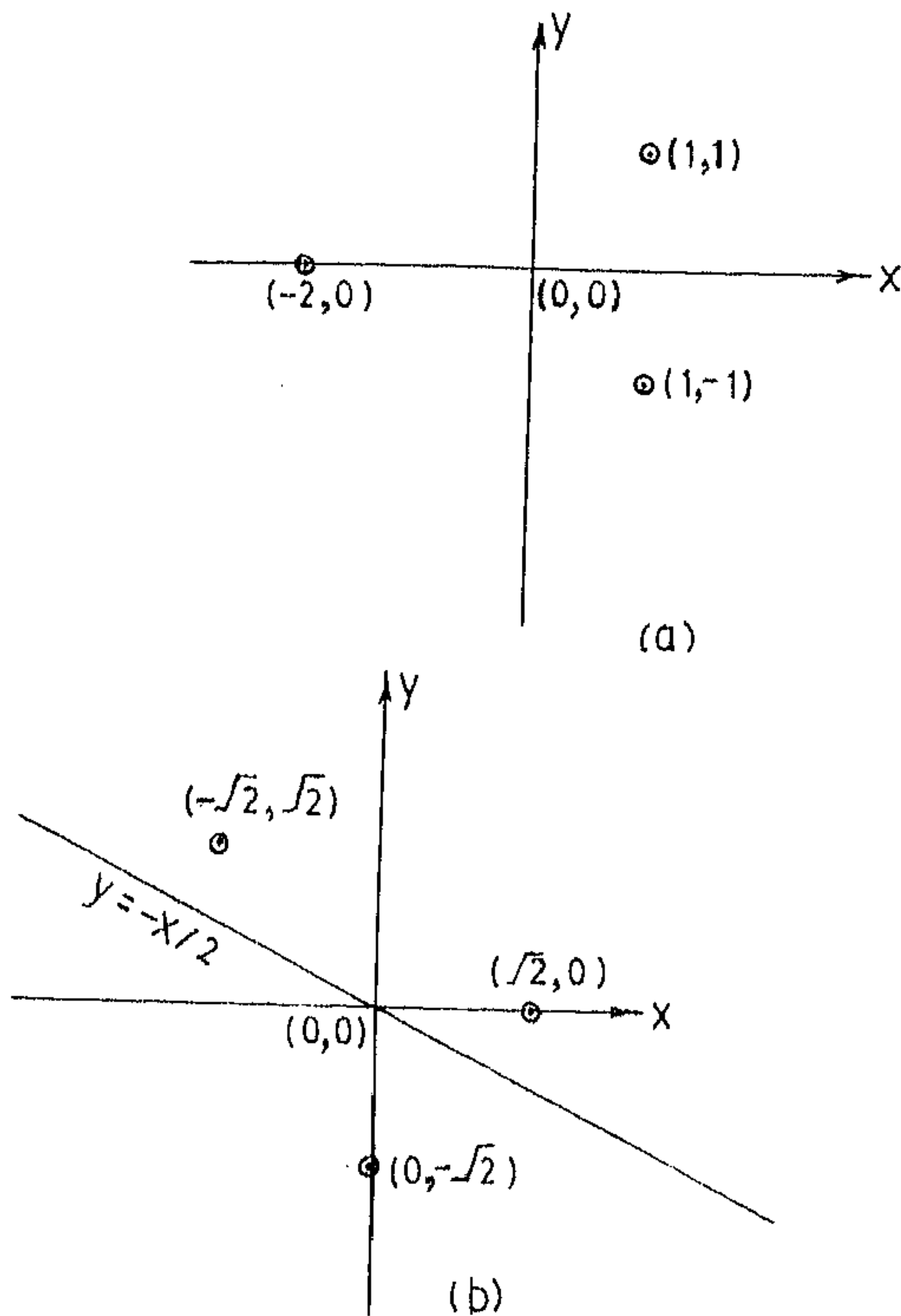


Fig.5.1 The three points in (b) are obtained by rotating the three points in (a) by angle 45° in the clockwise direction. The best fit in (a) is $y=0$. In (b) it is $y=-x/2$, not $y=-x$.

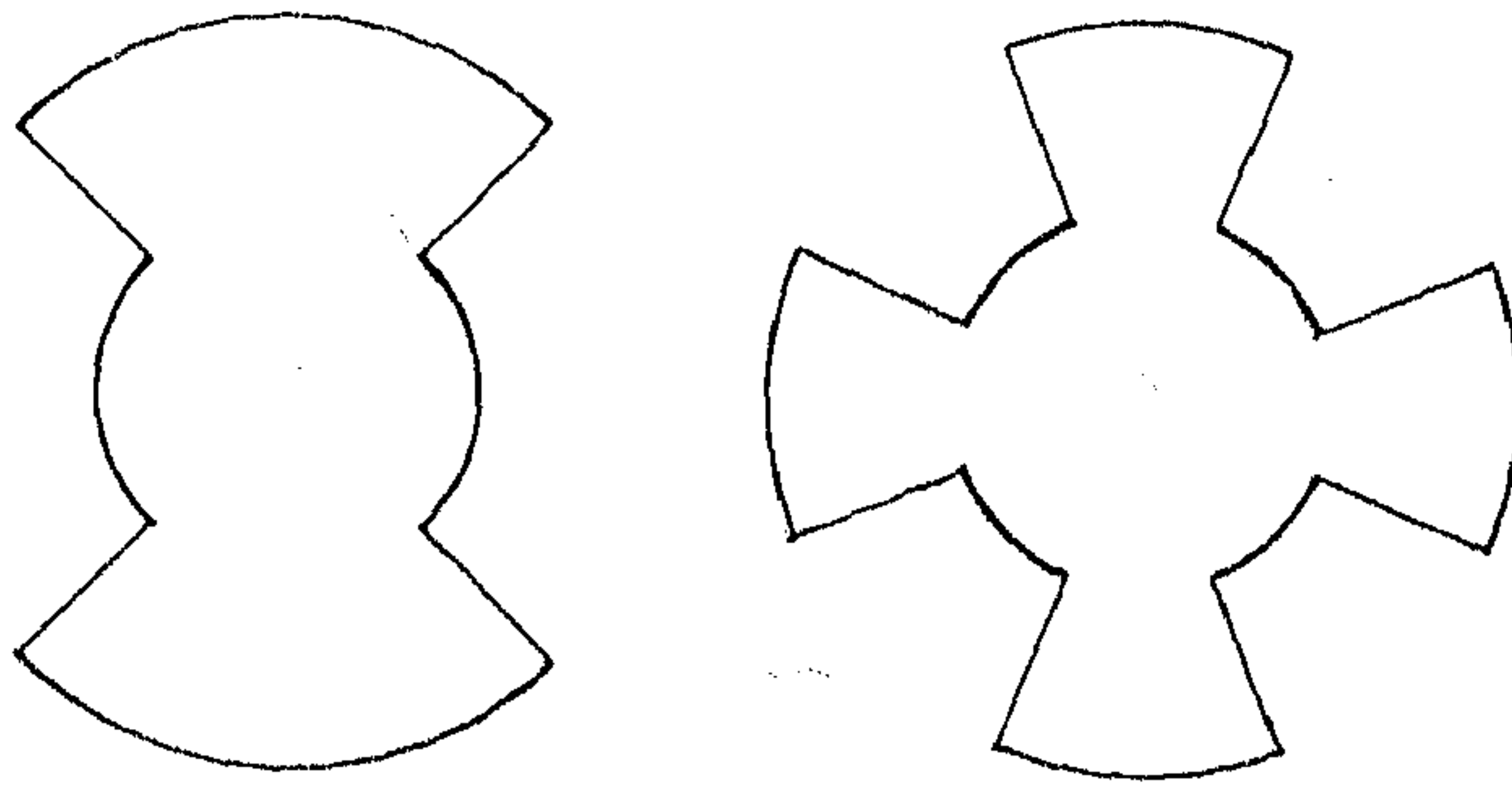


Fig.5.2 The two regions above have different shapes but their shape vectors are the same for any value (even) of n . In fact, in the shape vector the first $n/2$ entries will be 1 while the last $n/2$ entries will be 0.5.

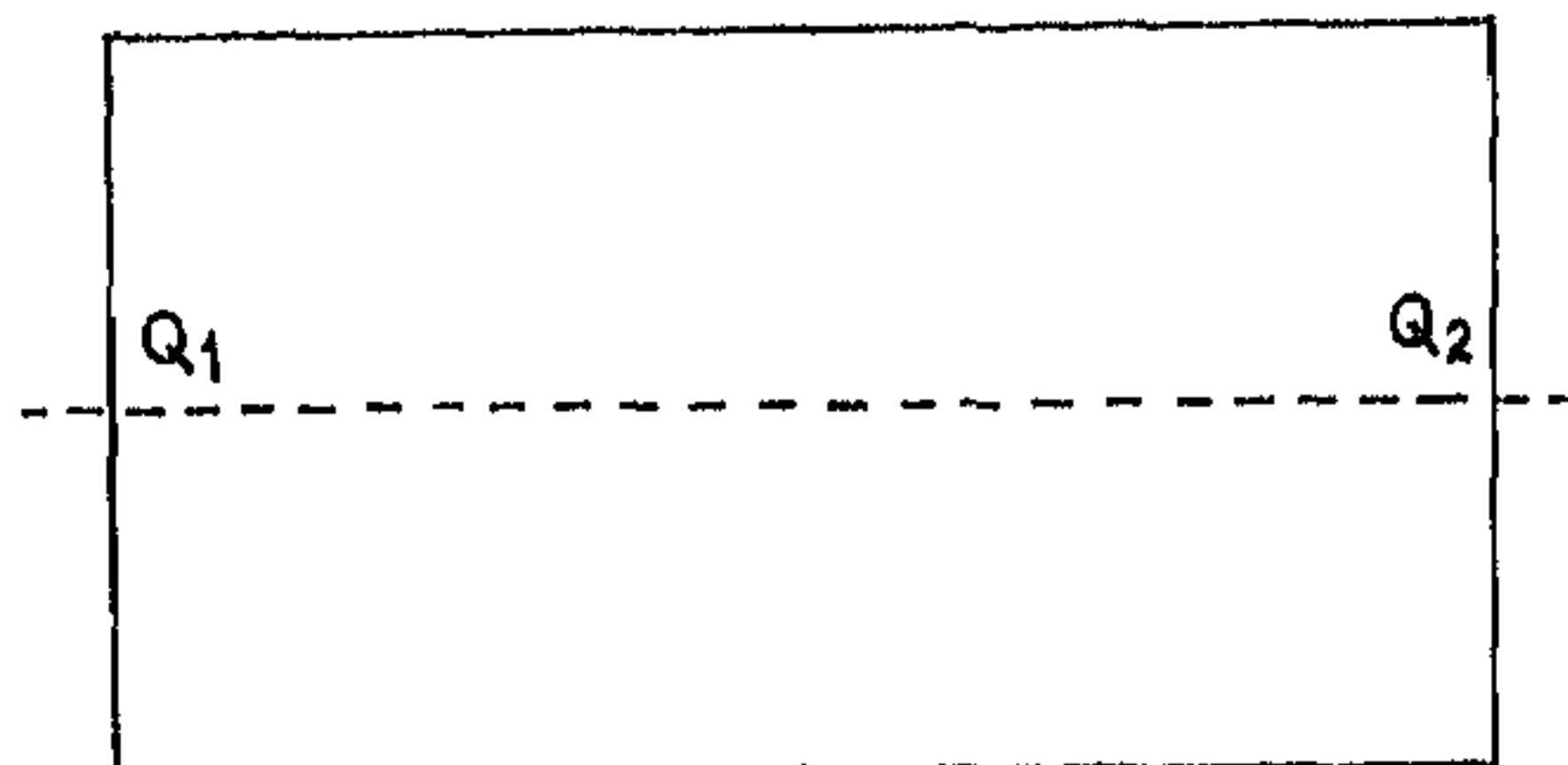


Fig. 5.3(a) Q_1Q_2 is the major axis of the above rectangle. The two points Q_1 and Q_2 are the two reference points of the above region.

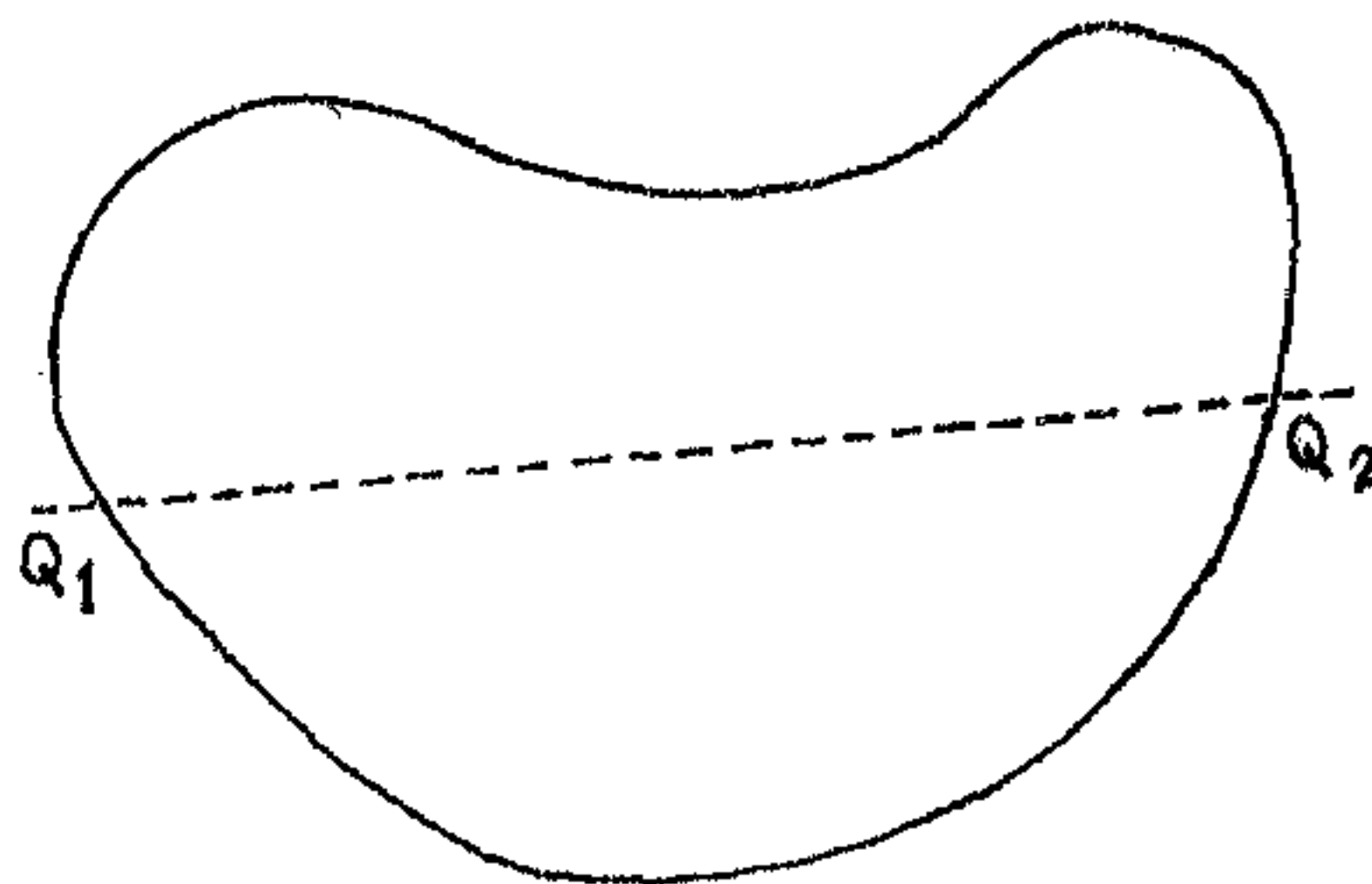
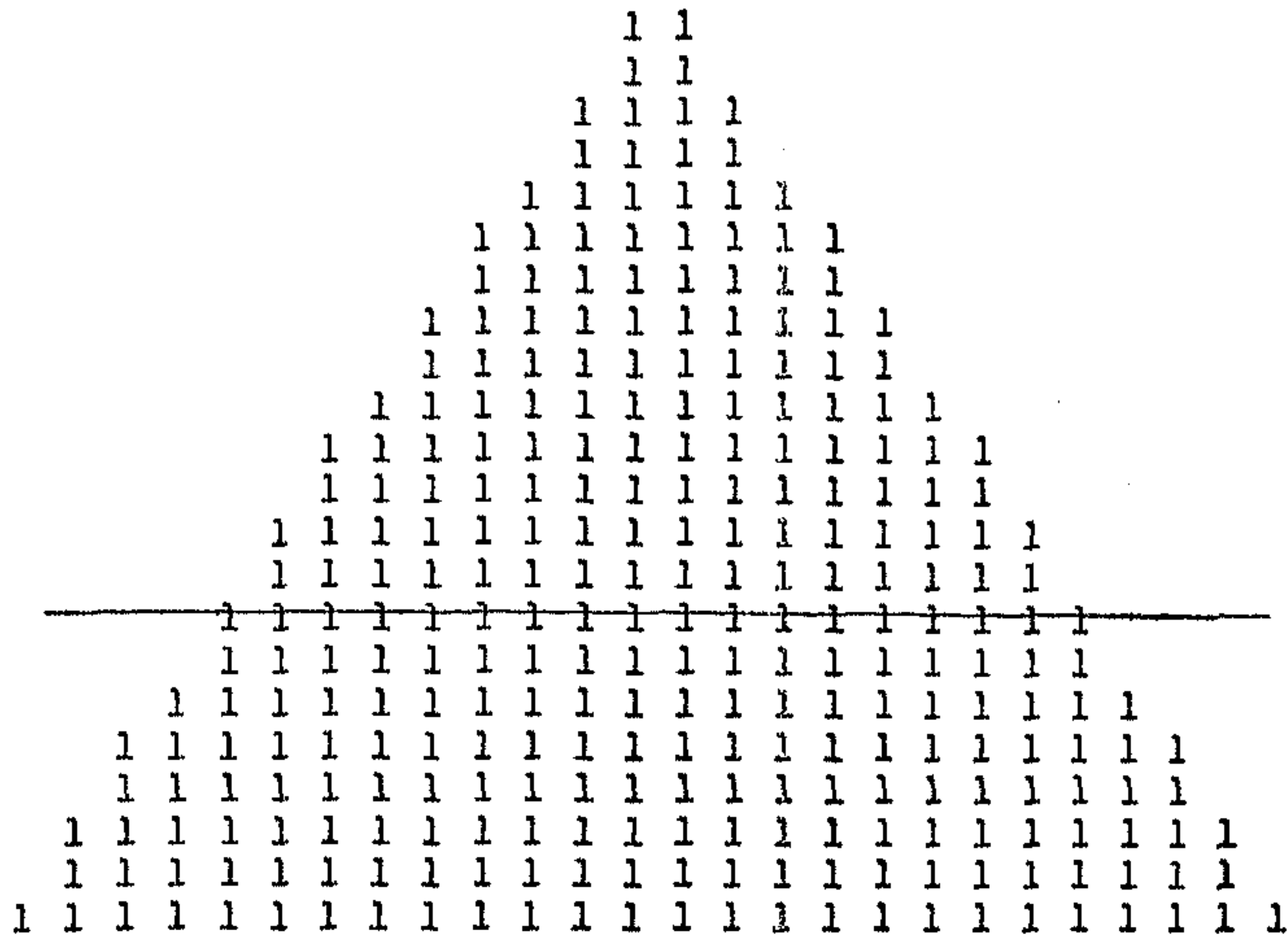
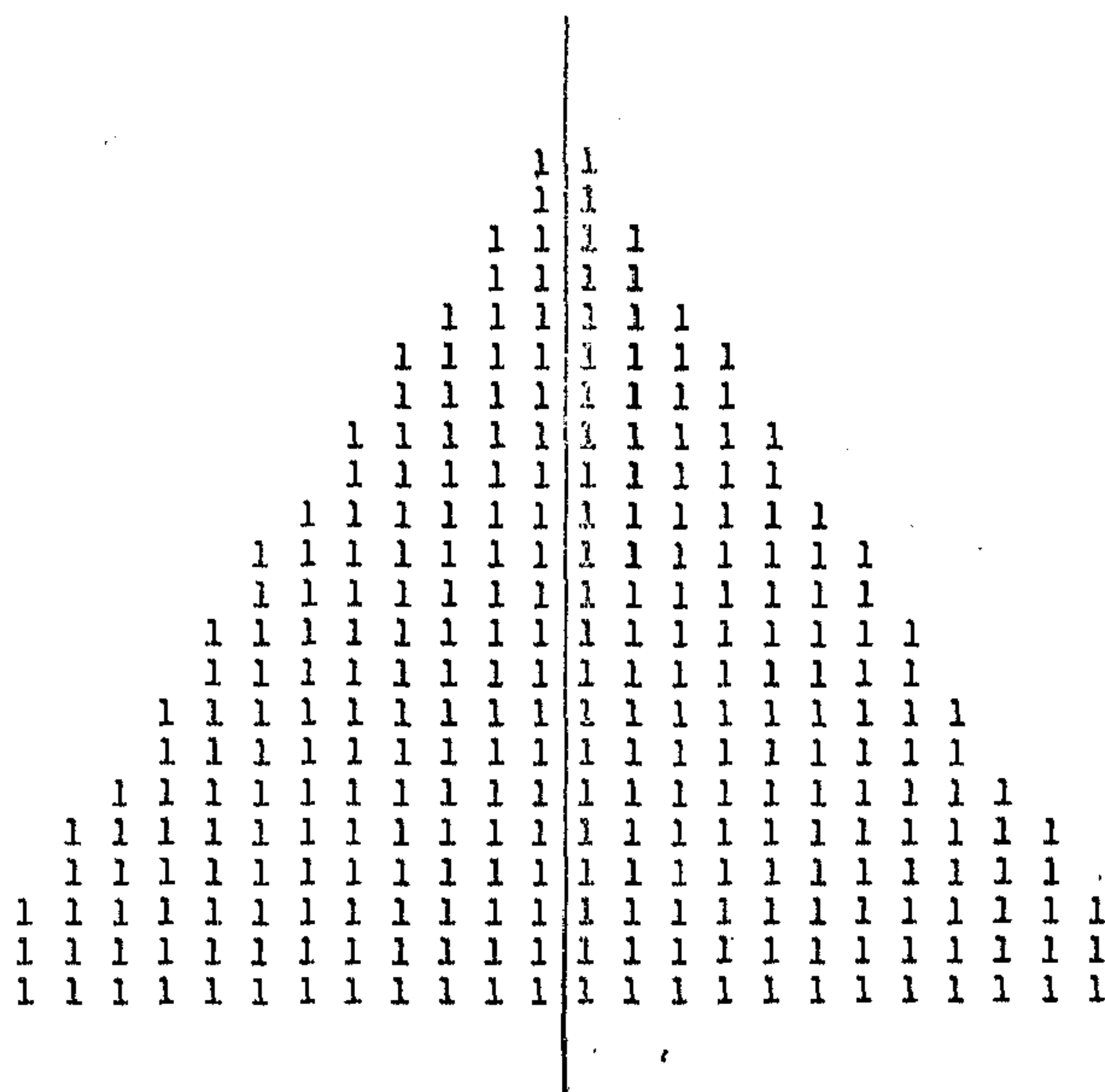


Fig. 5.3(b) Q_1Q_2 is the major axis of the above region with irregular shape whose reference points are Q_1 and Q_2 .



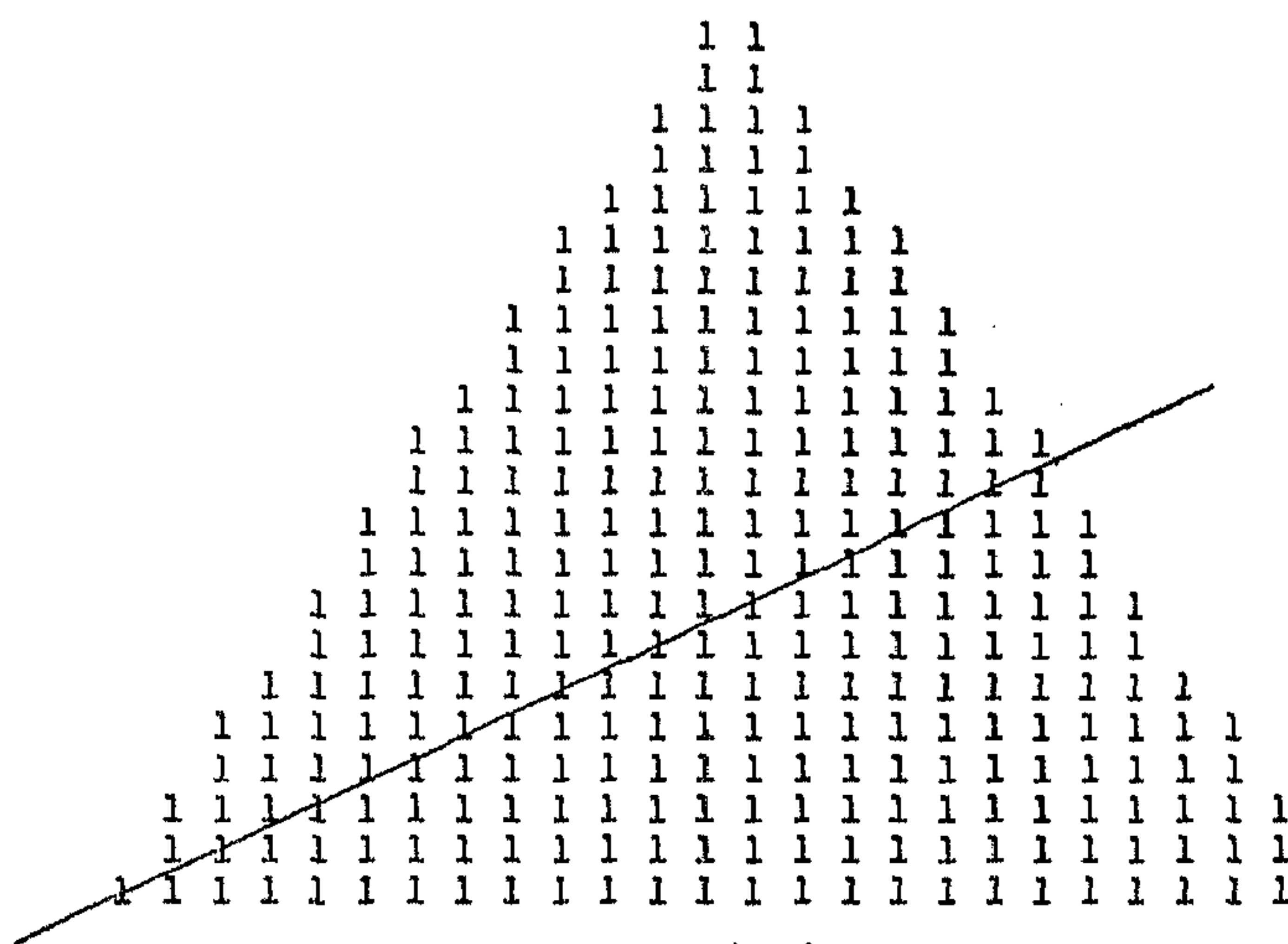
(a) The major axis is horizontal

Fig.5.4 (contd.)



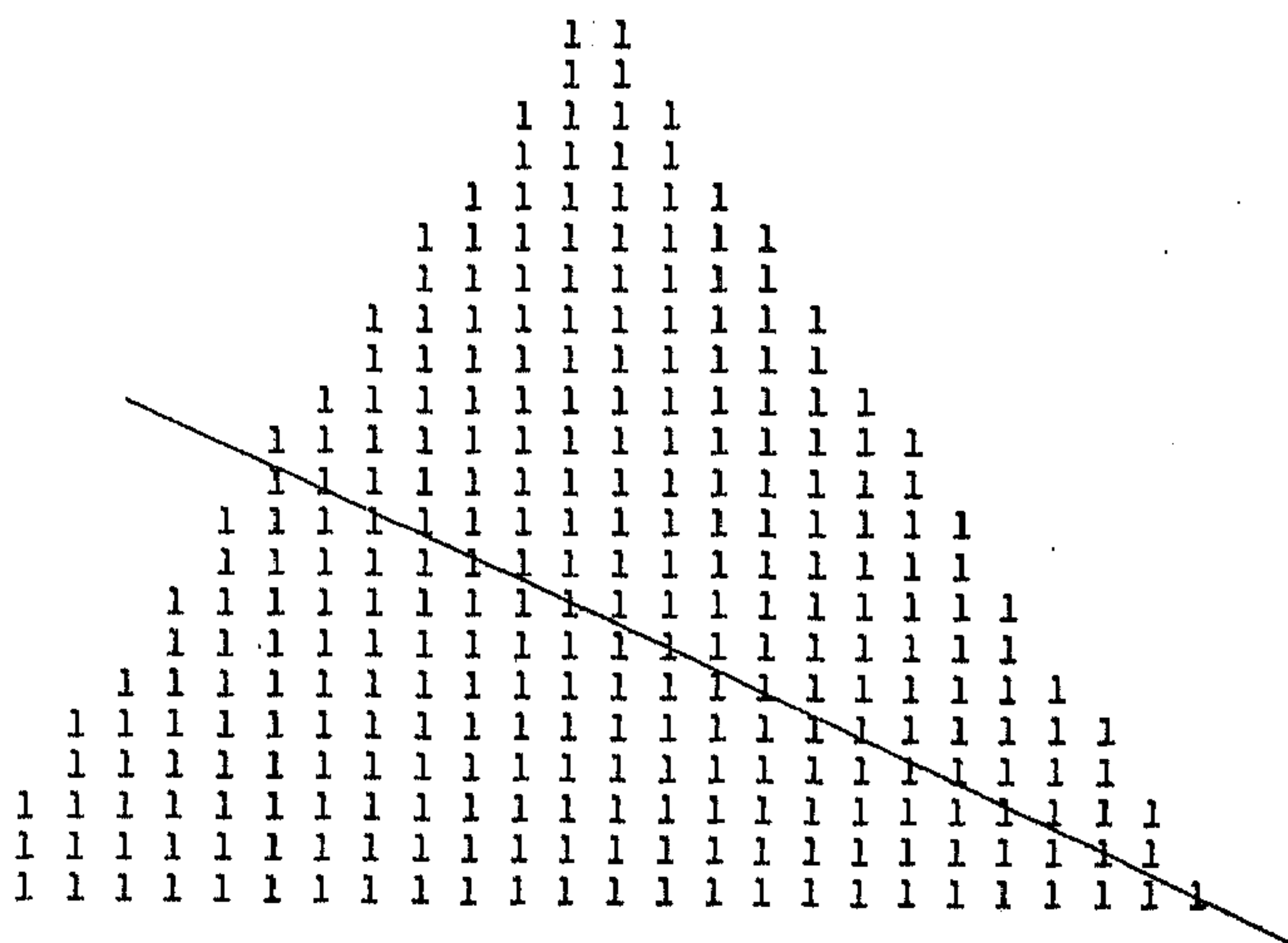
(b) The major axis is vertical

Fig.5.4 (contd.)



(c) The major axis is slant

Fig.5.4 (contd.)



(d) The major axis is slant

Fig.5.4 The above four triangular shapes are nearly equilateral. In each of these cases, the denominator in Equation (5.5), i.e. $\text{Variance}(X) - \text{Variance}(Y)$, is non-zero (though its absolute value is very small) and hence the major axis is unique. Note that although all the four pictures above are almost the same, slight noise has changed the major axis considerably. This is because all of them almost belong to $\mathcal{R}_1$ in the sense that for each of them, $\text{Covariance}(X, Y)$ and $\text{Variance}(X) - \text{Variance}(Y)$ are zero or almost zero.

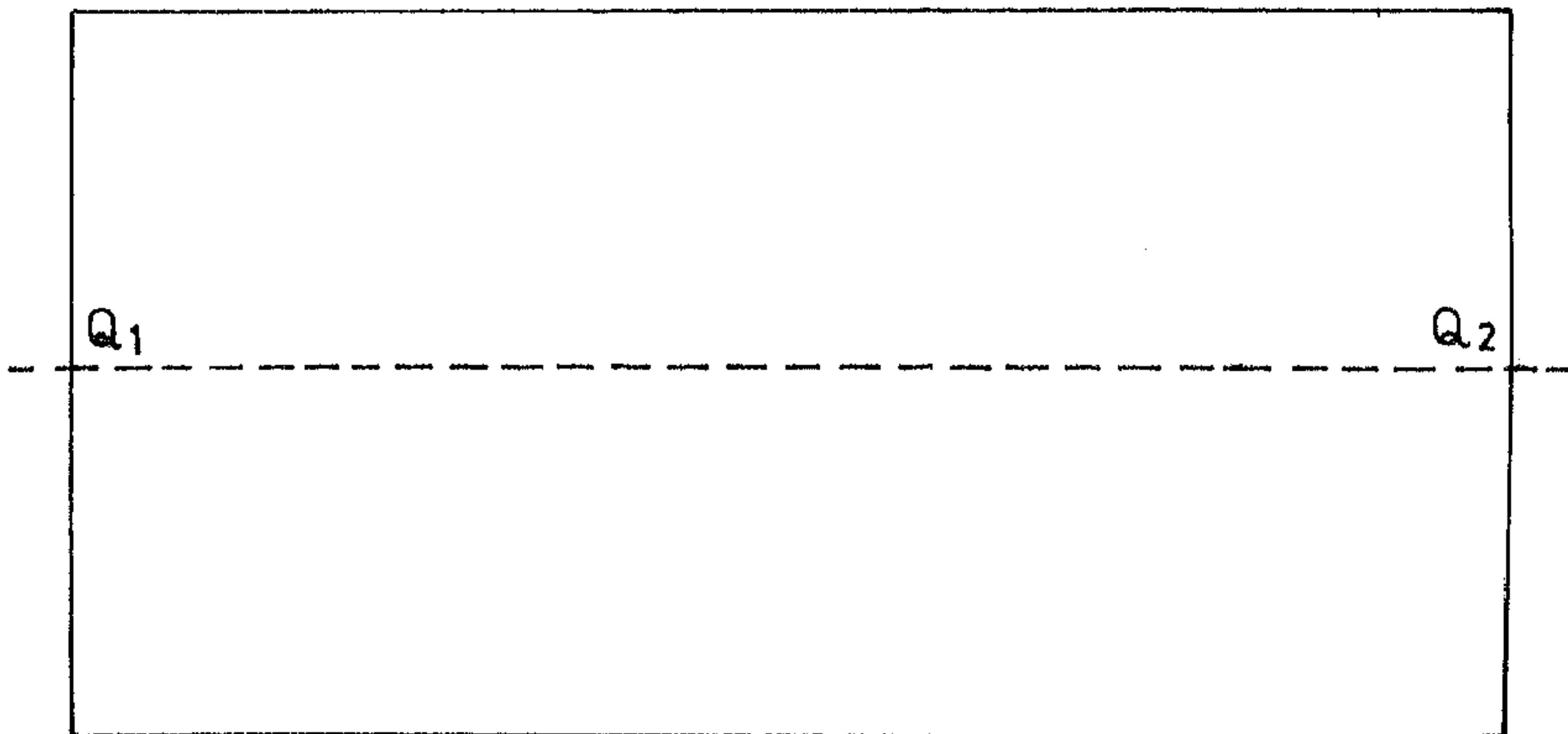


Fig. 5.5(a) For the rectangle above, Q_1Q_2 is the major axis (irrespective of whether it is defined on the basis of all pixels or on the basis of only boundary pixels). Some noise is added to the above rectangle and the noisy rectangle is shown in Figs. 5.5(b) and (c).

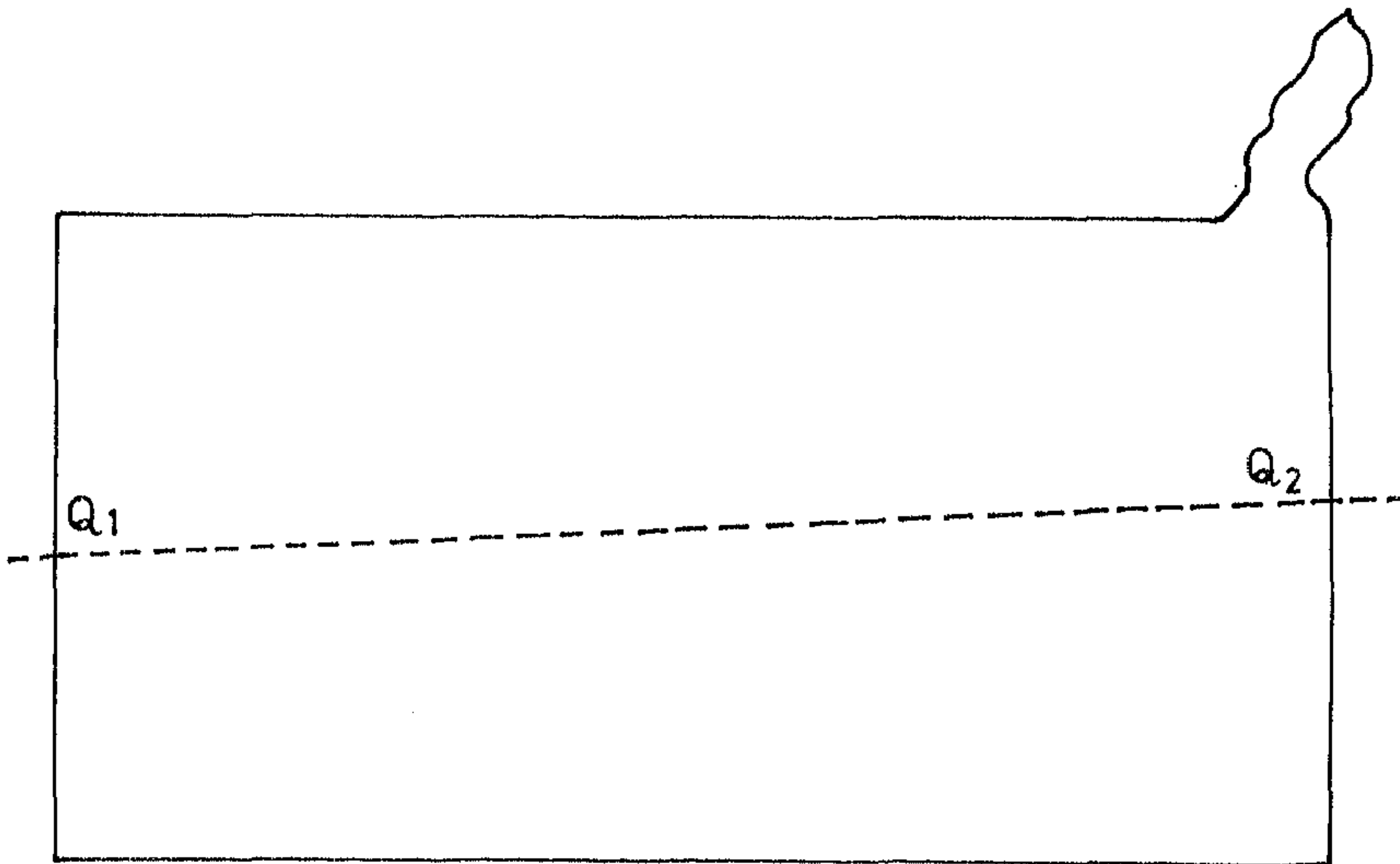


Fig. 5.5(b) For the noisy rectangle above, Q_1Q_2 is the major axis found on the basis of all pixels. It is clear that the major axis has been disturbed due to noise, because it is no more horizontal.

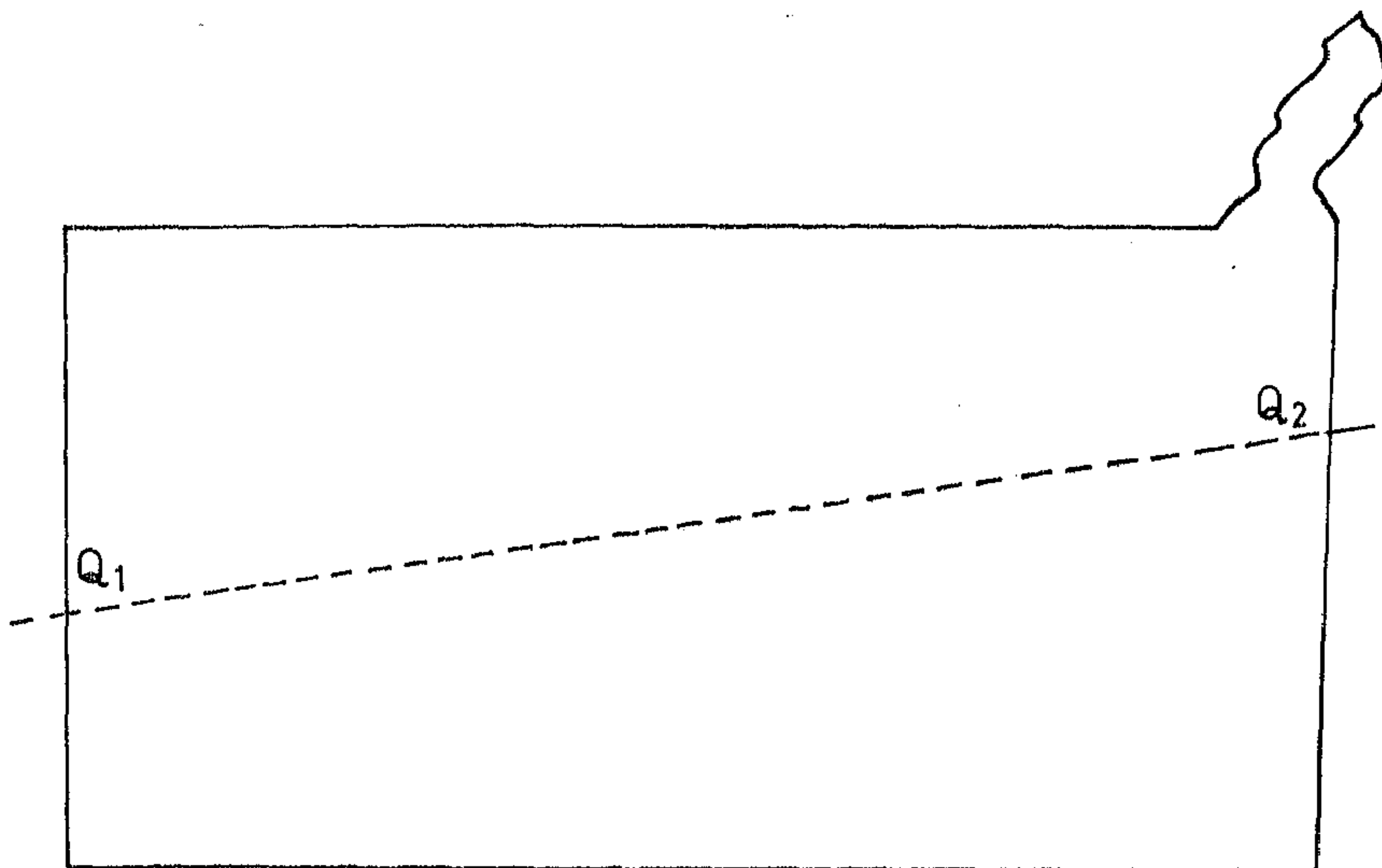


Fig. 5.5(c) For the same noisy rectangle as in Fig.5.5(b), Q_1Q_2 is the major axis found on the basis of only boundary pixels. Here also the major axis has been disturbed. But the disturbance is more in Fig.5.5(c) than in Fig.5.5(b). Thus, the major axis found on the basis of only boundary pixels is more sensitive to noise than the major axis found on the basis of all pixels.

Table 5.1

Shape vectors ($n=8$) of some standard shapes are given below

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
Circle	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Square	1.00	1.00	1.00	1.00	1.00	0.89	0.35	0.09
Ellipse(1.1)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.61
(1.2)	1.00	1.00	1.00	1.00	1.00	1.00	0.91	0.36
(1.5)	1.00	1.00	1.00	1.00	1.00	0.80	0.44	0.20
(2.0)	1.00	1.00	1.00	1.00	0.66	0.42	0.27	0.13
(2.5)	1.00	1.00	1.00	0.74	0.44	0.30	0.20	0.10
(3.0)	1.00	1.00	0.92	0.52	0.35	0.24	0.16	0.08
(4.0)	1.00	1.00	0.59	0.36	0.25	0.18	0.12	0.06
Rectangle(1.1)	1.00	1.00	1.00	1.00	1.00	0.87	0.36	0.09
(1.2)	1.00	1.00	1.00	1.00	1.00	0.78	0.39	0.09
(1.5)	1.00	1.00	1.00	1.00	0.88	0.60	0.43	0.10
(2.0)	1.00	1.00	1.00	0.91	0.59	0.45	0.37	0.14
(2.5)	1.00	1.00	0.99	0.66	0.46	0.36	0.30	0.16
(3.0)	1.00	1.00	0.87	0.52	0.38	0.30	0.25	0.16
(4.0)	1.00	0.99	0.58	0.37	0.28	0.23	0.19	0.14

The ratio of the major axis to the minor axis for ellipses and rectangles are given within brackets above.

Table 5.2

Values of μ_3 between some standard shapes (n=8)

		E L L I P S E R E C T A N G L E															
		CIRCLE	SQUARE	1.1	1.2	1.5	2.0	2.5	3.0	4.0	1.1	1.2	1.5	2.0	2.5	3.0	4.0
E L L I P S E	CIRCLE	1.00	0.79	0.95	0.91	0.81	0.68	0.60	0.53	0.44	0.79	0.78	0.75	0.68	0.62	0.56	0.47
	SQUARE	0.79	1.00	0.84	0.88	0.96	0.88	0.80	0.74	0.65	0.99	0.98	0.94	0.87	0.81	0.75	0.67
	1.1	0.95	0.84	1.00	0.96	0.85	0.73	0.65	0.58	0.49	0.84	0.83	0.80	0.73	0.67	0.61	0.52
	1.2	0.91	0.88	0.96	1.00	0.90	0.78	0.69	0.62	0.53	0.88	0.87	0.84	0.77	0.71	0.65	0.56
	1.5	0.81	0.96	0.85	0.90	1.00	0.88	0.79	0.73	0.64	0.97	0.98	0.94	0.88	0.81	0.75	0.67
	2.0	0.68	0.88	0.73	0.78	0.88	1.00	0.91	0.85	0.76	0.89	0.89	0.93	0.96	0.92	0.87	0.79
	2.5	0.60	0.80	0.65	0.69	0.79	0.91	1.00	0.94	0.84	0.81	0.81	0.85	0.92	0.96	0.93	0.86
	3.0	0.53	0.74	0.58	0.62	0.73	0.85	0.94	1.00	0.91	0.74	0.75	0.78	0.85	0.92	0.96	0.92
	4.0	0.44	0.65	0.49	0.53	0.64	0.76	0.84	0.91	1.00	0.65	0.66	0.69	0.76	0.83	0.88	0.96
R E C T A N G L E	1.1	0.79	0.99	0.84	0.88	0.97	0.89	0.81	0.74	0.65	1.00	0.99	0.94	0.88	0.81	0.75	0.67
	1.2	0.78	0.98	0.83	0.87	0.98	0.89	0.81	0.75	0.66	0.99	1.00	0.96	0.89	0.82	0.76	0.68
	1.5	0.75	0.94	0.80	0.84	0.94	0.93	0.85	0.78	0.69	0.94	0.96	1.00	0.92	0.85	0.80	0.71
	2.0	0.68	0.87	0.73	0.77	0.88	0.96	0.92	0.85	0.76	0.88	0.89	0.92	1.00	0.93	0.87	0.79
	2.5	0.62	0.81	0.67	0.71	0.81	0.92	0.96	0.92	0.83	0.81	0.82	0.85	0.93	1.00	0.94	0.86
	3.0	0.56	0.75	0.61	0.65	0.75	0.87	0.93	0.96	0.88	0.75	0.76	0.80	0.87	0.94	1.00	0.91
	4.0	0.47	0.67	0.52	0.56	0.67	0.79	0.86	0.92	0.96	0.67	0.68	0.71	0.79	0.86	0.91	1.00

Table 5.3

Shape similarity values μ_4, μ_5 and μ_6 between a circle and rectangles/ellipses ($n=32$ for μ_5 and μ_6)

Rectangles

Ratio of the major axis to the minor axis	μ_4	μ_5	μ_6
4.00	0.5543	0.8277	0.8150
3.00	0.6285	0.8407	0.8420
2.00	0.7465	0.8991	0.8866
1.50	0.8348	0.9049	0.9178
1.25	0.8906	0.9209	0.9346
1.20	0.9051	0.9222	0.9374
1.125	0.9059	0.9284	0.9406
1.10	0.9069	0.9297	0.9410

Ellipses

Ratio of the major axis to the minor axis	μ_4	μ_5	μ_6
4.00	0.5835	0.8101	0.8112
3.00	0.6610	0.8646	0.8413
2.00	0.7777	0.9339	0.8931
1.50	0.8687	0.9732	0.9308
1.25	0.9260	0.9904	0.9552
1.20	0.9422	0.9941	0.9612
1.125	0.9613	0.9959	0.9663
1.10	0.9694	0.9974	0.9708

CHAPTER 6

Symmetry of Curves and Regions

6.1 Introduction

In this chapter, we will deal with symmetry of 2-dimensional curves and regions. Symmetry can play many roles in understanding shapes. A symmetric shape can be represented economically. One need store a description of only one half of a symmetric shape alongwith the axis of symmetry. If this half also is symmetric then further economy is possible. Again, if a part of a symmetric shape is missing or noisy, then with the help of symmetry one can complete the shape or rid the shape of noise. The axis of symmetry is also useful in defining the orientation of a shape.

We will discuss two types of symmetry in this chapter. The first type is symmetry around an axis and is usually termed as axial or lateral symmetry. We will call this ~~only~~ 'symmetry' while the other type we will call 'antisymmetry' which is symmetry around a point and is usually known as rotational or radial symmetry.

Detection of symmetry is possible on the basis of both scalar measurements and structural information. Examples of the former are moments (Zusne, 1970) and Fourier co-efficients

(Zahn and Roskies, 1972). Structural information has been used by Davis(1977B), Chaudhuri(1980) and Parui(1983A). Davis considers symmetries of simple polygons first and then of figures that have a hierarchical side/angle structure. Chaudhuri detects symmetries on the basis of some correlation measure between curvatures of opposite points on a closed curve with respect to an axis. Parui detects symmetries in a hierarchical manner. At each level the best axis of symmetry is found. At a lower level, the detected best axis is a better approximation of the actual global axis of symmetry. The degree of symmetry is defined on the basis of the directional codes at equidistant points on a closed curve.

In Section 6.2 we discuss symmetry and antisymmetry of open curves. Various measures of degree of symmetry and antisymmetry of open curves are defined on the basis of distances D_3 and D_6 defined in Chapter 4. In Section 6.3 we discuss symmetry and antisymmetry of regions (without holes). Three measures of degree of symmetry and two measures of degree of antisymmetry are defined on the basis of the distances D_3 and D_6 and the major axis defined in Chapter 5. We consider symmetry and antisymmetry in the analogue domain in Sections 6.2 and 6.3. But in Section 6.4 we detect symmetry (of a region) directly in the digital domain. The axis of symmetry of a region is here found in a hierarchical manner and is better than the axis found in Section 6.3. Results

and discussion are given in Section 6.5.

6.2 Symmetry and Antisymmetry of Open Curves

Let $A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0, 1]\}$ be an open curve with length L and with slope $\{w(\lambda) : \lambda \in (0, 1)\}$ (defined in Chapter 4). Let $A_1 = \{(u_1(\lambda L/2), v_1(\lambda L/2)) : \lambda \in [0, 1]\}$ and $A_2 = \{(u_2(\lambda L/2), v_2(\lambda L/2)) : \lambda \in [0, 1]\}$ be two halves of A , each with length $L/2$, such that $u_1(\lambda L/2) = u(\frac{\lambda L}{2})$, $v_1(\lambda L/2) = v(\frac{\lambda L}{2})$, $u_2(\lambda L/2) = u(L - \frac{\lambda L}{2})$ and $v_2(\lambda L/2) = v(L - \frac{\lambda L}{2})$ for all $\lambda \in [0, 1]$. In fact, A_1 and A_2 are two open curves with the starting points $(u(0), v(0))$ and $(u(L), v(L))$ respectively and with the common end point $(u(L/2), v(L/2))$. Let the slopes of A_1 and A_2 be $\{w_1(\lambda) : \lambda \in (0, 1)\}$ and $\{w_2(\lambda) : \lambda \in (0, 1)\}$ respectively. Clearly, $w_1(\lambda) = w(\lambda/2)$ for all $\lambda \in (0, 1)$ and $w_2(\lambda) = w(1 - \lambda/2) + 4 \pmod{8}$ for all $\lambda \in (0, 1)$ except when $(u(L - \frac{\lambda L}{2}), v(L - \frac{\lambda L}{2}))$ is a corner point.

Definition 6.2.1 An open curve A is said to be symmetric if there exists a straight line φ such that φ is the perpendicular bisector of the line segment joining the two points $(u_1(\lambda L/2), v_1(\lambda L/2))$ and $(u_2(\lambda L/2), v_2(\lambda L/2))$ for all $\lambda \in [0, 1]$. The axis of symmetry of A is φ .

Proposition 6.2.1 Suppose an open curve A is symmetric with the axis of symmetry φ . Suppose, φ_1 is a rotation and translation of φ and B is the rotation and translation of A by the same parameters. Then B is symmetric around φ_1 .

Proof. It is clear since only the system of axes is being rotated and translated keeping the curve A unchanged.

Recall the definition of continuous directional codes in the interval $S_8 = (0, 8]$ defined in Chapter 4. Any straight line will have two directional codes in S_8 corresponding to two opposite directions of the line. One of them, say d , belongs to $(0, 4]$ and the other one is $d+4$.

Definition 6.2.2 The absolute directional code of a straight line is defined as the directional code d and belongs to $(0, 4]$. For example, the absolute directional code of the x -axis is 2.

Proposition 6.2.2 Suppose an open curve A is symmetric around the x -axis. Then $\frac{w_1(\lambda) + w_2(\lambda) \pmod{8}}{2} = 2$ for all λ in $(0, 1)$.

Proof. Since the x -axis is the perpendicular bisector of the line segment joining the points $(u_1(\lambda L/2), v_1(\lambda L/2))$ and $(u_2(\lambda L/2), v_2(\lambda L/2))$ for all $\lambda \in [0, 1)$, $u_1(\lambda L/2) = u_2(\lambda L/2)$ and $v_1(\lambda L/2) = -v_2(\lambda L/2)$ for all $\lambda \in [0, 1)$.

Recall the functions $f(\lambda)$ and $g(\lambda)$ defined for an open curve in Section 4.3 of Chapter 4. Let $f_1(\lambda), g_1(\lambda)$ and $f_2(\lambda), g_2(\lambda)$ be these functions for the two halves A_1 and A_2 respectively. For $\lambda \in (0, 1)$

$$f_1(\lambda) = \lim_{h \downarrow 0} \frac{v_1(\lambda L/2) - v_1(\lambda L/2 - h)}{u_1(\lambda L/2) - u_1(\lambda L/2 - h)}.$$

$$\begin{aligned}
&= -\lim_{h \downarrow 0} \frac{v_2(\lambda L/2) - v_2(\lambda L/2 - h)}{u_2(\lambda L/2) - u_2(\lambda L/2 - h)} \\
&= -f_2(\lambda)
\end{aligned}$$

Case I $|f_1(\lambda)| < \infty$.

$$\begin{aligned}
g_1(\lambda) &= \frac{\pi/2 - \tan^{-1} f_1(\lambda)}{\pi/4} \\
&= \frac{\pi/2 + \tan^{-1} f_2(\lambda)}{\pi/4} \\
&= \frac{\pi - [\pi/2 - \tan^{-1} f_2(\lambda)]}{\pi/4} \\
&= 4 - g_2(\lambda)
\end{aligned}$$

$$\text{So, } g_1(\lambda) + g_2(\lambda) = 4.$$

Since $u_1(\lambda L/2) = u_2(\lambda L/2)$ for all $\lambda \in (0, 1)$, $w_1(\lambda) = g_1(\lambda)$ if and only if $w_2(\lambda) = g_2(\lambda)$ and $w_1(\lambda) = g_1(\lambda) + 4$ if and only if $w_2(\lambda) = g_2(\lambda) + 4$. Thus, $w_1(\lambda) + w_2(\lambda) = g_1(\lambda) + g_2(\lambda)$ or $g_1(\lambda) + g_2(\lambda) + 8 = 4$ or 12 .

$$\text{So, } \frac{w_1(\lambda) + w_2(\lambda) \pmod{8}}{2} = 2$$

Case II $|f_1(\lambda)| = \infty$

That is, either $f_1(\lambda) = +\infty$ and $f_2(\lambda) = -\infty$

or $f_1(\lambda) = -\infty$ and $f_2(\lambda) = +\infty$

So, either $w_1(\lambda) = 8$ and $w_2(\lambda) = 4$

or $w_1(\lambda) = 4$ and $w_2(\lambda) = 8$

Thus, in this case also, $\frac{w_1(\lambda) + w_2(\lambda) \pmod{8}}{2} = 2$

Hence Proposition 6.2.2.

Proposition 6.2.3 Suppose, A is any symmetric open curve with φ as its axis of symmetry. Then

$\frac{w_1(\lambda) + w_2(\lambda) \pmod{8}}{2}$ gives the absolute directional code of φ for all λ in $(0,1)$.

Proof. φ can be rotated and translated so that it coincides with the x-axis. Let B be the corresponding rotation and translation of A . It can be easily seen that B is symmetric around the x-axis. Suppose, the slopes of the two halves of B are $\{w_3(\lambda) : \lambda \in (0,1)\}$ and $\{w_4(\lambda) : \lambda \in (0,1)\}$. Suppose, the rotation needed to get B from A is by an angle α in the anticlockwise direction. So, from Proposition 4.3.3, $w_1(\lambda) = w_3(\lambda) + 4\alpha/\pi \pmod{8}$ and $w_2(\lambda) = w_4(\lambda) + 4\alpha/\pi \pmod{8}$ for all λ in $(0,1)$ and the absolute directional code of φ is $2 + 4\alpha/\pi \pmod{4}$.

Now, from the above proposition, $w_3(\lambda) + w_4(\lambda) = 4$ or 12 for all λ in $(0,1)$. Now,

$$\begin{aligned} \frac{w_1(\lambda) + w_2(\lambda) \pmod{8}}{2} &= \frac{w_3(\lambda) + w_4(\lambda) + 8\alpha/\pi \pmod{8}}{2} \\ &= \frac{4 + 8\alpha/\pi \pmod{8}}{2} = 2 + 4\alpha/\pi \pmod{4} \end{aligned}$$

= absolute directional code of φ , for all λ in $(0,1)$.

Now, we will prove the converse of Proposition 6.2.2.

Proposition 6.2.4 Suppose an open curve

$A = \{(u(\lambda L), v(\lambda L)) : \lambda \in [0,1]\}$ is such that (i) $v(L/2) = 0$ and and (ii) $\frac{w_1(\lambda) + w_2(\lambda) \pmod{8}}{2} = 2$ for all $\lambda \in (0,1)$. Then A is

symmetric around the x-axis.

Proof. $w_1(\lambda) + w_2(\lambda) = 4$ or 12 for all $\lambda \in (0, 1)$ since $w_1(\lambda), w_2(\lambda)$ belong to $(0, 8]$. There are only 3 following cases:

$$\text{Case 1} \quad 4 > w_1(\lambda) > 0, \quad 4 > w_2(\lambda) > 0, \quad w_1(\lambda) + w_2(\lambda) = 4$$

$$\text{Case 2} \quad 8 > w_1(\lambda) > 4, \quad 8 > w_2(\lambda) > 4, \quad w_1(\lambda) + w_2(\lambda) = 12$$

$$\text{Case 3} \quad w_1(\lambda) = 4, \quad w_2(\lambda) = 8 \quad \text{or} \quad w_1(\lambda) = 8, \quad w_2(\lambda) = 4$$

(we assume, without loss of generality, that $L=1$)

$$\text{Case 1} \quad w_1(\lambda) = \frac{\pi/2 - \tan^{-1} f_1(\lambda)}{\pi/4}, \quad w_2(\lambda) = \frac{\pi/2 - \tan^{-1} f_2(\lambda)}{\pi/4}$$

$$w_1(\lambda) + w_2(\lambda) = 4 \text{ implies } \tan^{-1} f_1(\lambda) + \tan^{-1} f_2(\lambda) = 0$$

So, $f_2(\lambda) = -f_1(\lambda)$. From Appendix D.1,

$u_1'(\lambda/2) = u_2'(\lambda/2)$ and $v_1'(\lambda/2) = -v_2'(\lambda/2)$ for all but finitely many λ .

$$\text{Also,} \quad u_1(\lambda/2) = \int_0^{\lambda} u_1'(\lambda_1/2) d\lambda_1 + u_1(0)$$

$$v_1(\lambda/2) = \int_0^{\lambda} v_1'(\lambda_1/2) d\lambda_1 + v_1(0)$$

$$u_2(\lambda/2) = \int_0^{\lambda} u_2'(\lambda_1/2) d\lambda_1 + u_2(0)$$

$$v_2(\lambda/2) = \int_0^{\lambda} v_2'(\lambda_1/2) d\lambda_1 + v_2(0)$$

for $\lambda \in (0, 1)$.

At the end point of the two halves $v_1(1/2) = v_2(1/2) = v(1/2) = v(L/2) = 0$. Again, since $v_1'(\lambda/2) = -v_2'(\lambda/2)$, $\int_0^{\lambda} v_1'(\lambda_1) d\lambda_1 =$

$= \int_0^1 v_2'(\lambda_1) d\lambda_1$. So, $v_1(0) = -v_2(0)$. Similarly, $u_1(0) = u_2(0)$.

Thus $u_1(\lambda/2) = u_2(\lambda/2)$ and $v_1(\lambda/2) = -v_2(\lambda/2)$ for all λ in $(0,1)$.

Hence the x-axis is the perpendicular bisector of the line segment joining $(u_1(\lambda/2), v_1(\lambda/2))$ and $(u_2(\lambda/2), v_2(\lambda/2))$ for all λ in $[0,1)$. So, A is symmetric around the x-axis.

The proofs for the other two cases will be similar.

We will now prove the converse of Proposition 6.2.3.

Proposition 6.2.5 Suppose an open curve A is such that $\frac{w_1(\lambda) + w_2(\lambda) \pmod{8}}{2}$ is a constant k for all $\lambda \in (0,1)$.

Then A is symmetric around an axis whose absolute directional code is k.

Proof. Without loss of generality, we assume that the mid point of A, $(u(L/2), v(L/2))$, is at the origin $(0,0)$. Let A be rotated by an angle $\alpha = (k+2)\pi/4$ in the anticlockwise direction. Let B be the rotated open curve with the slopes of its two halves as $\{w_3(\lambda) : \lambda \in (0,1)\}$ and $\{w_4(\lambda) : \lambda \in (0,1)\}$. From Proposition 4.3.3, $w_3(\lambda) = w_1(\lambda) - (k+2) \pmod{8}$ and $w_4(\lambda) = w_2(\lambda) - (k+2) \pmod{8}$ all $\lambda \in (0,1)$. So, $w_3(\lambda) + w_4(\lambda) \pmod{8} = w_1(\lambda) + w_2(\lambda) - 2(k+2) \pmod{8} = 2k - 2(k+2) \pmod{8} = 4$.

Thus, from the above proposition, B is symmetric around the x-axis (whose absolute directional code is 2). So, from Proposition 6.2.1, A is symmetric around an axis whose absolute directional code $= 2 + 4\alpha/\pi \pmod{4} = 2 + (k+2) \pmod{4} = k$.

Thus from Proposition 6.2.3 and Proposition 6.2.5, we get the following proposition:

Proposition 6.2.6 Suppose, A is an open curve whose two halves have the slopes $\{w_1(\lambda): \lambda \in (0,1)\}$ and $\{w_2(\lambda): \lambda \in (0,1)\}$. Then A is symmetric around an axis whose absolute directional code is k if and only if $\frac{w_1(\lambda) + w_2(\lambda) (\text{mod } 8)}{2} = k$ for all $\lambda \in (0,1)$.

Definition 6.2.3 For an open curve A , $w^*(\lambda)$ is defined as $\frac{w_1(\lambda) + w_2(\lambda) (\text{mod } 8)}{2}$ for all $\lambda \in (0,1)$ and $\bar{w}$ is defined as $\int_0^1 w^*(\lambda) d\lambda$.

Definition 6.2.4 A measure of deviation from symmetry of an open curve A is defined as

$$\text{Dev}_1(A) = \text{Dev}(A_1, A_2) = \int_0^1 D(w^*(\lambda), \bar{w}) d\lambda \quad (6.1)$$

$$\text{where } D(w^*(\lambda), \bar{w}) = \text{Min} \{ |w^*(\lambda) - \bar{w}|, 4 - |w^*(\lambda) - \bar{w}| \} \text{ for all } \lambda \in (0,1).$$

Proposition 6.2.7 $\text{Dev}_1(A) \leq 2$.

Proof. Note that $4 \geq w^*(\lambda) > 0$ for all $\lambda \in (0,1)$. Also, $4 \geq \bar{w} > 0$. Hence $2 \geq D(w^*(\lambda), \bar{w}) \geq 0$ for all $\lambda \in (0,1)$. Thus, $\text{Dev}_1(A) \leq 2$.

On the basis of the above results a measure of degree of symmetry of an open curve is defined in the following way —

Definition 6.2.5 A measure of degree of symmetry of an open curve A is defined as

$$\mu_7(A) = 1 - \text{Dev}_1(A)/2 \quad (6.2)$$

Clearly, $1 \geq \mu_7(A) \geq 0$.

Proposition 6.2.8 An open curve A is symmetric if and only if $\mu_7(A) = 1$.

Proof. It is clear from Proposition 6.2.6.

We will propose two more measures of degree of symmetry of an open curve.

Definition 6.2.6 The reflection of a subset $\{(x_i, y_i) : i \in I\}$ of the plane is defined as the set $\{(x_i, -y_i) : i \in I\}$. Reflection is taken here around the x-axis.

Proposition 6.2.9 Suppose an open curve A with two halves A_1 and A_2 is symmetric. Then A_1 and the reflection of A_2 have the same shape or more specifically, A_1 is a translation and rotation of the reflection of A_2 .

Proof. From the earlier results, it is clear that we can always get an open curve B which is a translation or rotation of A such that B is symmetric around the x-axis. The two halves B_1 and B_2 of B also are translations and rotations of A_1 and A_2 respectively. From the definition, B_1 is the reflection of B_2 . Now, the reflection of B_2 is a translation and rotation of the reflection of A_2 . Since B_1 is a translation and rotation of A_1 , A_1 is a translation and rotation of the reflection of A_2 .

Proposition 6.2.10 Suppose, A_1 and A_2 are the two halves of an open curve A . If A_1 and the reflection of A_2 have the same shape, then A is symmetric.

Proof. Since A_1 and A_2 have the same length, A_1 is a translation and rotation of the reflection of A_2 . Now,
 $A_1 = \{(u_1(\lambda L/2), v_1(\lambda L/2)) : \lambda \in [0, 1]\}$ and the reflection of
 $A_2 = \{(u_2(\lambda L/2), -v_2(\lambda L/2)) : \lambda \in [0, 1]\}$. So, for some a , b and α ,

$$u_1(\lambda L/2) = u_2(\lambda L/2) \cos \alpha + v_2(\lambda L/2) \sin \alpha + a$$

$$\text{and } v_1(\lambda L/2) = u_2(\lambda L/2) \sin \alpha - v_2(\lambda L/2) \cos \alpha + b$$

for all $\lambda \in [0, 1]$.

Putting $\lambda=1$ we get $a = u_1(L/2)(1 - \cos \alpha) - v_1(L/2) \sin \alpha$

$$b = -u_1(L/2) \sin \alpha + v_1(L/2)(1 + \cos \alpha)$$

since $u_1(L/2) = u_2(L/2)$ and $v_1(L/2) = v_2(L/2)$.

From the above two expressions of a and b it can be seen that

$$a \cos \frac{\alpha}{2} + b \sin \frac{\alpha}{2} = 0 \quad (6.3)$$

Let B be the open curve obtained after rotating A by the angle $\alpha/2$ in the clockwise direction. That is, $B = A_{-\alpha/2}$.

We will now show that B is symmetric.

Let $B_1 = \{(u_3(\lambda L/2), v_3(\lambda L/2)) : \lambda \in [0, 1]\}$ and

$B_2 = \{(u_4(\lambda L/2), v_4(\lambda L/2)) : \lambda \in [0, 1]\}$ be the two halves of B . So,

$$u_3(\lambda L/2) = u_1(\lambda L/2) \cos \frac{\alpha}{2} + v_1(\lambda L/2) \sin \frac{\alpha}{2}$$

$$v_3(\lambda L/2) = -u_1(\lambda L/2) \sin \frac{\alpha}{2} + v_1(\lambda L/2) \cos \frac{\alpha}{2}$$

$$u_4(\lambda L/2) = u_2(\lambda L/2) \cos \frac{\alpha}{2} + v_2(\lambda L/2) \sin \frac{\alpha}{2}$$

and
$$v_4(\lambda L/2) = -u_2(\lambda L/2) \sin \frac{\alpha}{2} + v_2(\lambda L/2) \cos \frac{\alpha}{2}$$
for all $\lambda \in [0, 1]$.

Now,
$$\begin{aligned} w_3(\lambda L/2) &= \left\{ u_2(\lambda L/2) \cos \alpha + v_2(\lambda L/2) \sin \alpha + a \right\} \cos \frac{\alpha}{2} \\ &\quad + \left\{ u_2(\lambda L/2) \sin \alpha - v_2(\lambda L/2) \cos \alpha + b \right\} \sin \frac{\alpha}{2} \\ &= u_4(\lambda L/2) + a \cos \frac{\alpha}{2} + b \sin \frac{\alpha}{2} \\ &= u_4(\lambda L/2) \quad \text{from (6.3)} \end{aligned}$$

Similarly,
$$v_3(\lambda L/2) = -v_4(\lambda L/2) + b \cos \frac{\alpha}{2} - a \sin \frac{\alpha}{2}$$

i.e.
$$v_3(\lambda L/2) + v_4(\lambda L/2) = \text{constant}$$

for all $\lambda \in [0, 1]$.

Thus, from the definition of symmetry, B is symmetric around the line $y = \frac{1}{2} \left\{ b \cos \frac{\alpha}{2} - a \sin \frac{\alpha}{2} \right\}$. From Proposition 6.2.1, A also is symmetric.

From the above two propositions, follows the proposition below.

Proposition 6.2.11 A is symmetric if and only if A_1 and the reflection of A_2 have the same shape.

Definition 6.2.7 Another measure of deviation from symmetry of an open curve A is defined as

$$\text{Dev}_2(A) = D_3(A_1, A_3) \quad (6.4)$$

where A_3 is the reflection of A_2 and D_3 is a shape distance defined in Chapter 4. From Proposition 4.3.17 in Chapter 4, $\text{Dev}_2(A) \leq 2$.

Definition 6.2.8 On the basis of $\text{Dev}_2(A)$, another measure of degree of symmetry of an open curve A is defined as

$$\mu_8(A) = 1 - \text{Dev}_2(A)/2 \quad (6.5)$$

Clearly, $1 \geq \mu_8(A) \geq 0$.

Proposition 6.2.12 An open curve A is symmetric if and only if $\mu_8(A) = 1$.

Proof. From Proposition 6.2.11, A is symmetric if and only if A_1 and A_3 have the same shape. Again, from Proposition 4.3.12 in Chapter 4, A_1 and A_3 have the same shape if and only if $D_3(A_1, A_3) = 0$.

Definition 6.2.9 Still another measure of deviation from symmetry of an open curve A is defined as

$$\text{Dev}_3(A) = D_6(A_1, A_3) \quad (6.6)$$

where D_6 is a shape distance defined in Chapter 4. From Proposition 4.3.28 in Chapter 4, $\text{Dev}_3(A) \leq 1/\sqrt{6}$.

Definition 6.2.10 On the basis of $\text{Dev}_3(A)$, another measure of degree of symmetry of an open curve A is defined as

$$\mu_9(A) = 1 - \sqrt{6} \text{Dev}_3(A) \quad (6.7)$$

Clearly, $1 \geq \mu_9(A) \geq 0$.

Proposition 6.2.13 An open curve A is symmetric if and only if $\mu_9(A) = 1$.

Proof. From Proposition 6.2.11, A is symmetric if and only if A_1 and A_3 have the same shape. Again, from Proposition 4.3.26 in Chapter 4, A_1 and A_3 have the same shape

if and only if $D_6(A_1, A_3) = 0$.

Antisymmetry

If an open curve A is symmetric, then only one of its two halves A_1, A_2 alongwith its axis of symmetry can be stored without any information about A being lost. That is, A can be reconstructed from just one half and its axis of symmetry. Similarly, for an antisymmetric open curve A (Definition 4.2.14), only one half of A alongwith its mid-point is sufficient for the reconstruction of A .

From Proposition 4.2.12, an open curve A is antisymmetric if and only if A and A^{-1} have the same shape. So, two measures of degree of antisymmetry of an open curve can be defined on the basis of D_3 and D_6 .

Definition 6.2.11 Two measures of deviation from antisymmetry of an open curve A are defined as

$$\text{Dev}_4(A) = D_3(A, A^{-1}) \quad (6.8)$$

$$\text{and } \text{Dev}_5(A) = D_6(A, A^{-1}) \quad (6.9)$$

Definition 6.2.12 Two measures of degree of antisymmetry of an open curve A are defined as

$$\mu_{10}(A) = 1 - \text{Dev}_4(A)/2 \quad (6.10)$$

$$\text{and } \mu_{11}(A) = 1 - \sqrt{6} \text{Dev}_5(A) \quad (6.11)$$

Clearly, $1 \geq \mu_{10}(A), \mu_{11}(A) \geq 0$.

Proposition 6.2.14 An open curve A is antisymmetric

if and only if $\mu_{10}(A) = 1$ if and only if $\mu_{11}(A) = 1$.

Proof. It is immediate from Proposition 4.2.12.

The computations of $\text{Dev}_4(A)$ or $D_3(A, A^{-1})$ can be simplified. From Appendix D.2, if A is antisymmetric, then $w(\lambda) = w(1-\lambda)$ for all λ except when λ -th point is a corner point where $\{w(\lambda) : \lambda \in (0, 1)\}$ is the slope of A .

Now, $D_3(A, A^{-1})$

$$= D_3(w(\lambda), w(1-\lambda) \oplus 4) \text{ from Proposition 4.3.5}$$

$$= D_3(w(\lambda), w(1-\lambda))$$

which can be computed as $D_2(w(\lambda), w(1-\lambda))$ that is zero when A is antisymmetric, from above.

$D_2(w(\lambda), w(1-\lambda))$ is defined in Chapter 4 as

$$\int_0^1 D_1(w(\lambda), w(1-\lambda)) d\lambda \quad (6.12)$$

where $D_1(x, y) = \text{Min}\{|x-y|, 8-|x-y|\}$.

6.3 Symmetry and Antisymmetry of Regions

In this section, we will propose three measures of degree of symmetry and two measures of degree of antisymmetry of a region.

Definition 6.3.1 A subset A of the Euclidean plane is said to be symmetric around the x -axis if $(x, y) \in A$ implies $(x, -y) \in A$.

It can be seen from Appendix D.3 that a region A is symmetric around the x -axis if and only if the boundary $\bar{A}$ is symmetric around the x -axis. So, in this chapter, we will study symmetry of a region through symmetry of its boundary.

Let A be a region and P be a point on the boundary $\bar{A}$. Let L be the length of $\bar{A}$. Let $\{(u(s), v(s)) : s \in [0, L]\}$ be the closed curve (clockwise) $\bar{A}$ such that its starting point and end point coincide with P . That is, the co-ordinates of $P = (u(0), v(0)) = (u(L), v(L))$. Let A_1 and A_2 be the two halves of $\bar{A}$ as defined in the beginning of Section 6.2. That is, A_1 describes the open curve from $(u(0), v(0))$ to $(u(L/2), v(L/2))$ along $\bar{A}$ in the clockwise direction and A_2 does the same thing in the anticlockwise direction. In other words, $A_1 = \{(u_1(\lambda L/2), v_1(\lambda L/2)) : \lambda \in [0, 1]\}$ and $A_2 = \{(u_2(\lambda L/2), v_2(\lambda L/2)) : \lambda \in [0, 1]\}$ such that $u_1(\lambda L/2) = u(\frac{\lambda L}{2})$, $v_1(\lambda L/2) = v(\frac{\lambda L}{2})$, $u_2(\lambda L/2) = u(L - \frac{\lambda L}{2})$ and $v_2(\lambda L/2) = v(L - \frac{\lambda L}{2})$ for all $\lambda \in [0, 1]$. The point $(u(L/2), v(L/2))$ is called the point opposite to P and is denoted by P^* .

We will use the result (from Section 6.2) that an open curve A is symmetric if and only if A_1 and the reflection of A_2 have the same shape, in defining symmetry of a region.

Definition 6.3.2 Let A be a region and P be a point on the boundary $\bar{A}$. Let A_1 and A_2 be the two halves of $\bar{A}$ with respect to P . Then the region A is said to be symmetric around the line joining P and P^* if A_1 and the reflection of A_2 have

the same shape. The line joining P and P^* is the axis of symmetry.

In order to detect symmetry of a region, we study the two halves of the boundary. So, the problem is to find the point on the boundary with respect to which the boundary is to be halved. The major axis (defined in Chapter 5) provides a solution to this problem.

Proposition 6.3.1 The major axis of a symmetric region A either coincides with or is perpendicular to the axis of symmetry, provided A belongs to $\mathcal{R}^*$.

Proof. We know that if a symmetric region is rotated by an angle β , then its axis of symmetry and its major axis also are rotated by the same angle β in the same direction. That is why we assume that A is symmetric around the x -axis. Since, $(x, y) \in A$ implies $(x, -y) \in A$, the centroid of A will be $(c_1, c_2) = (c_1, 0)$. Now, from the definition, the major axis of A passes through (c_1, c_2) and makes the angle β with the x -axis where

$$\tan 2\beta = \frac{2 \int_A (x-c_1)(y-c_2) da}{\int_A (x-c_1)^2 da - \int_A (y-c_2)^2 da} \quad (6.13)$$

Since $(x, y) \in A$ implies $(x, -y) \in A$, the numerator of (6.13) is zero. The denominator can not be zero, because A belongs to $\mathcal{R}^*$.

Now, $\tan 2\beta = 0$ gives two solutions for β . One is $\beta=0$ and the other is $\beta=\pi/2$. If $\beta=0$, the major axis coincides with the

x-axis, since $c_2=0$. If $\beta=\pi/2$, the major axis is perpendicular to the x-axis and passes through (c_1, c_2) .

Hence Proposition 6.3.1 .

Let A be a region in $\mathcal{Q}^*$. Let Q_1 and Q_2 be the reference points of A (defined in Chapter 5). That is, Q_1 and Q_2 are the two farthest points of intersection between the major axis of A and the boundary $\bar{A}$. Let the minor axis of A be the straight line which passes through the centroid of A and is perpendicular to the major axis. Let Q_3 and Q_4 be the two farthest points of intersection between the minor axis of A and the boundary $\bar{A}$. From the above proposition, it is clear that if A is symmetric, its axis of symmetry passes through either Q_1 and Q_2 or Q_3 and Q_4 . Now, for any arbitrary (symmetric or otherwise) region A , we will halve the boundary $\bar{A}$ with respect to each Q_i . Suppose, the corresponding halves are $\{A1_i, A2_i\}$, $i=1,2,3,4$. Let $A3_i$ be the reflection of $A2_i$. Recall the definition of D_3 in Chapter 4 (Definition 4.3.6). $D_3(A1_i, A3_i)$ are well-defined. If A is symmetric, then from Proposition 6.3.1, either $D_3(A1_i, A3_i)=0$ for $i=1,2$ or $D_3(A1_i, A3_i)=0$ for $i=3,4$.

Definition 6.3.3 A measure of deviation from symmetry of a region A is defined as

$$\text{Dev}_6(A) = \min_{i=1 \text{ to } 4} D_3(A1_i, A3_i) \quad (6.14)$$

The above results involving D_3 hold for D_6 (Defined in

Chapter 4) also.

Definition 6.3.4 Another measure of deviation from symmetry of a region A is defined as

$$\text{Dev}_7(A) = \min_{i=1 \text{ to } 4} D_6(A1_i, A3_i) \quad (6.15)$$

Definition 6.3.5 Still another measure of deviation from symmetry of a region A is defined as

$$\text{Dev}_8(A) = \min_{i=1 \text{ to } 4} \text{Dev}(A1_i, A3_i) \quad (6.16)$$

where $\text{Dev}(A1_i, A3_i)$ has been defined in Section 6.2.

We know, $D_3 \leq 2$, $D_6 \leq 1/\sqrt{6}$ and $\text{Dev} \leq 2$.

Definition 6.3.6 Three measures of degree of symmetry of a region A are defined as

$$\mu_{12}(A) = 1 - \text{Dev}_6(A)/2 \quad (6.17)$$

$$\mu_{13}(A) = 1 - \sqrt{6}\text{Dev}_7(A) \quad (6.18)$$

$$\text{and } \mu_{14}(A) = 1 - \text{Dev}_8(A)/2 \quad (6.19)$$

Clearly, $1 \geq \mu_{12}(A), \mu_{13}(A), \mu_{14}(A) \geq 0$

Proposition 6.3.2 A region A is symmetric if and only if $\mu_{12}(A)=1$ if and only if $\mu_{13}(A)=1$ if and only if $\mu_{14}(A)=1$.

Proof. It is clear from the previous results.

Antisymmetry

Definition 6.3.7 Let A be a region with boundary $\bar{A}$. A is said to be antisymmetric if if A_1 and A_2^{-1} have the same shape, where A_1 and A_2 are the two halves of $\bar{A}$ with respect to any point on $\bar{A}$.

Recall the halves $\{A_{1i}, A_{2i}\}$ of $\bar{A}$ with respect to the points $Q_i (i=1,2,3,4)$ defined earlier in this section (Section 6.3). Clearly, if A is antisymmetric, then $D_3(A_{1i}, A_{2i}^{-1}) = D_6(A_{1i}, A_{2i}^{-1}) = 0$.

Definition 6.3.8 Two measures of deviation from antisymmetry of a region A belonging to $\mathcal{R}^*$ are defined as

$$\text{Dev}_9(A) = \min_{i=1 \text{ to } 4} D_3(A_{1i}, A_{2i}^{-1}) \quad (6.20)$$

$$\text{and } \text{Dev}_{10}(A) = \min_{i=1 \text{ to } 4} D_6(A_{1i}, A_{2i}^{-1}) \quad (6.21)$$

Definition 6.3.9 Two measures of degree of antisymmetry of a region A belonging to $\mathcal{R}^*$ are defined as

$$\mu_{15}(A) = 1 - \text{Dev}_9(A)/2 \quad (6.22)$$

$$\text{and } \mu_{16}(A) = 1 - \sqrt{6} \text{Dev}_{10}(A) \quad (6.23)$$

Clearly, $1 \geq \mu_{15}(A), \mu_{16}(A) \geq 0$.

Computations of $\mu_i (i=7,8,\dots,16)$ in the digital case involve the following

- (1) Finding equidistant points on an open curve or on the

boundary of a region,

- (ii) Finding reflection of an open curve,
 - (iii) Finding the slope of an open curve,
 - (iv) Finding the values of D_3 and D_6
- and (v) Finding the points Q_1, Q_2, Q_3, Q_4 on the boundary of a region.

Methods for (i), (iii) and (iv) have been given in Chapter 4 while the method for (v) has been given in Chapter 5.

As far as the reflection of an open curve A is concerned, $(x_i, -y_i)$ are taken to be the equidistant points on the reflection of A where (x_i, y_i) are the equidistant points on A .

6.4 Hierarchical Search For the Best Axis of Symmetry Of a Region.

For a region that is not perfectly symmetric, several degrees of symmetry have been proposed in Section 6.3. In each case, the axis of symmetry is fixed (that is, either the major axis or the minor axis) on the basis of which degrees of symmetry are found using Dev_6, Dev_7 and Dev_8 . Now, for a region that is not perfectly symmetric, its best axis of symmetry may not coincide with its major or minor axis. In other words, there may be an axis of the region around which the deviation from symmetry (Dev_1 or Dev_2 or Dev_3) is less than the deviation around the major

or minor axis.

The best axis of symmetry of a region A can be defined on the basis of any of the deviations $\text{Dev}_1, \text{Dev}_2$ and Dev_3 defined in (6.1), (6.4) and (6.6) respectively. Let P be any point on the boundary $\bar{A}$ and $\{A_1, A_2\}$ be the corresponding halves of $\bar{A}$. Suppose, A_3 is the reflection of A_2 .

Definition 6.4.1 Three measures of deviations from symmetry of region A are defined as

$$\text{Dev}_{11}(A) = \inf_{P \in \bar{A}} D_3(A_1, A_3) \quad (6.24)$$

$$\text{Dev}_{12}(A) = \inf_{P \in \bar{A}} D_6(A_1, A_3) \quad (6.25)$$

$$\text{and } \text{Dev}_{13}(A) = \inf_{P \in \bar{A}} \text{Dev}(A_1, A_3) \quad (6.26)$$

The corresponding axes (i.e. the lines joining P and the opposite point P^*) are the best axes of symmetry in these three cases. Clearly, $\text{Dev}_6 \geq \text{Dev}_{11}, \text{Dev}_7 \geq \text{Dev}_{12}$ and $\text{Dev}_8 \geq \text{Dev}_{13}$ (equality holds for perfect symmetry). These best axes of symmetry are in general better than the major or minor axis.

Definition 6.4.2 Three measures of degree of symmetry of region A are defined as

$$\mu_{17}(A) = 1 - \text{Dev}_{11}(A)/2 \quad (6.27)$$

$$\mu_{18}(A) = 1 - \sqrt{6}\text{Dev}_{12}(A) \quad (6.28)$$

$$\text{and } \mu_{19}(A) = 1 - \text{Dev}_{13}(A)/2 \quad (6.29)$$

Clearly, $1 \geq \mu_{17}(A), \mu_{18}(A), \mu_{19}(A) \geq 0$.

Proposition 6.4.1 A region A is symmetric if and only if $\mu_{17}(A)=1$ if and only if $\mu_{18}(A)=1$ if and only if $\mu_{19}(A)=1$.

Proof. It is clear from ^{the} previous results.

Now, we will describe an algorithm to find the best axis of symmetry of a region and compute the corresponding degree of symmetry μ_{17} or μ_{18} or μ_{19} . Here, the boundary of the input digital region is represented as a polygon in a hierarchical manner. First, the boundary is scanned from any boundary point in the clockwise direction and the chain of directional codes (belonging to I_8) of the boundary is obtained such that the length of each directional code is almost the same (an algorithm for this has been given in Section 4.5 of Chapter 4). Let $d_i, i=1, \dots, N$ be the directional codes of the closed boundary which can be looked upon as a polygon with N sides. The sides will be denoted by their directional codes d_i . The starting points of d_i are the vertices v_i of the polygon. Here, contrary to the common definition of a polygon, three consecutive vertices can be collinear. That is, change of direction need not occur at every vertex. Call this polygon P_1 which is the lowest level in the hierarchy and which does not lose any information about the digital input. A higher level is defined as a polygon $P_r (N > r > 1)$ whose number of sides is $N_r = N/r$ and whose vertices are $v_j^r, j=1, \dots, N_r$, where $v_j^r = v_{(j-1)r+1}$. Correspondingly, d_j^r are defined as the directional codes (belonging to S_8) of the sides $\overrightarrow{v_j^r v_{j+1}^r} \pmod{N_r}$ of the

polygon P_r (algorithm to compute d_j^r is given in Appendix B.9). Though the sides of P_r are not necessarily of equal length, the distance between any two consecutive vertices along the boundary is the same. Higher values of r indicate higher levels of hierarchy. From the definition of P_r , it is clear that P_r contains only the coarse and global features of the boundary for higher values of r .

We will now compute the deviation from symmetry of the input figure at level r of the hierarchy. At this level, we consider $N_r/2$ lines for the axis of symmetry. These are lines a_j^r joining the vertices v_j^r and $v_{j+N_r/2}^r$, $j=1, \dots, N_r/2$. For each of these lines, a deviation from symmetry (Dev_1 or Dev_2 or Dev_3) is computed. (Note that $\text{Dev}_1, \text{Dev}_2$ and Dev_3 correspond to $\text{Dev}_{13}, \text{Dev}_{11}$ and Dev_{12} respectively). Here, we will describe computation of Dev_{13} only. Computations of Dev_{11} and Dev_{12} essentially involve computations of Dev_2 and Dev_3 which have been described in Chapter 4.

Consider the line a_j^r . The absolute directional code d_0 of a_j^r is computed from the vertices v_j^r and $v_{j+N_r/2}^r$. The slope of one half of the boundary is d_{j-1+k}^r , $k=1, \dots, N_r/2$ and the slope of the other half is $d_{j-k(\text{mod } N_r)}^r + 4(\text{mod } 8)$, $k=1, \dots, N_r/2$. So,

$$d_k^* = \frac{d_{j-1+k}^r + d_{j-k(\text{mod } N_r)}^r + 4(\text{mod } 8)}{2}$$

where d_k^* is the discrete version of w^* (Definition 6.2.3). In fact, d_k^* is the absolute directional code of the angular bisector of the angle between the sides d_{j-1+k}^r and $d_{j-k}^r \pmod{N_r}$ (Fig. 6.1).

Here, we will measure deviation around d_0 , the absolute directional code of a_j^r . The deviation is

$$E_j^r = 2 \sum_{k=1}^{N_r/2} e_k / N_r \quad (6.30)$$

where $e_k = \min \{ |d_0 - d_k^*|, 4 - |d_0 - d_k^*| \}$ and $\pi e_k / 4$ gives the acute angle between d_k^* and d_0 . If a_j^r is a perfect axis of symmetry of P_r at this level of hierarchy, then $e_k = 0$ for all k , so that $E_j^r = 0$. Since $2 \geq e_k \geq 0$, $2 \geq E_j^r \geq 0$.

The search for the best axis of symmetry of a region is carried out hierarchically. In the higher levels of the hierarchy, the polygon P_r will have small number of sides and hence the choice of the axis of symmetry is from among a small number of lines while in the lower levels the choice is from among a larger number of lines. Clearly, the latter will be a better choice. We start with a high level in the hierarchy, i.e., with a polygon P_r for a suitably high value of r . The deviations $E_j^r, j=1, \dots, N_r/2$ are first calculated and then the local minima among E_j^r are found. $E_{j_0}^r$ is said to be a local minimum if

$E_{j_0-1}^r \pmod{N_r/2} > E_{j_0}^r < E_{j_0+1}^r \pmod{N_r/2}$. Among the local minima, there is an $E_{j_0}^r$ such that the corresponding axis $a_{j_0}^r$ gives an

approximate idea of the position of the best axis of symmetry. In order to avoid a futile search for the best axis symmetry for a figure that is quite asymmetric, a threshold t_r is used at level r of the hierarchy. If all the local minima are greater than t_r , then we say that the figure is asymmetric and do not go for lower levels of the hierarchy. Otherwise, we choose those local minima $E_{j_0}^r$ that are less than t_r and go down to a lower level, say s . We assume, $r=ps$ where p is an integer greater than 1. At level s we do not have to check all the lines a_m^s or the deviations E_m^s , $m=1, \dots, N_s/2$ of the polygon P_s to get a better axis of symmetry. Since, at the level r , $a_{j_0}^r$ is the closest line to the globally best axis of symmetry, the best axis must lie between the lines $a_{j_1}^r$ and $a_{j_2}^r$ where $j_1 = j_0 - 1 \pmod{N_r/2}$ and $j_2 = j_0 + 1 \pmod{N_r/2}$. Note that $a_{j_1}^r = a_{w-p}^s \pmod{N_s/2}$ and $a_{j_2}^r = a_{w+p}^s \pmod{N_s/2}$ where $w = (j_0 - 1)p + 1 \pmod{N_s/2}$. So, we have to check only $2p-1$ lines a_m^s , $m=w-p+1, \dots, w+p-1 \pmod{N_s/2}$ and find out the corresponding deviations E_m^s on the basis of v_m^s and d_m^s , $m=1, \dots, N_s$. If the minimum among these $(2p-1)$ E_m^s is $E_{m_0}^s$, then the corresponding line $a_{m_0}^s$ is the best axis of symmetry at level s . In fact, $a_{m_0}^s$ gives a better (or equally good at the worst) approximation of the globally best axis of symmetry than $a_{j_0}^r$. In a similar fashion, we go down to a still lower level in the

hierarchy. If a^s is the best axis of symmetry at level s , then a^s tends to the globally best axis of symmetry as s decreases. In practice, we stop at a level where a desired degree of accuracy in the position of the axis of symmetry is achieved. Suppose, E is the minimum deviation computed above and let μ_{19}^* be the degree of symmetry of the input region where $\mu_{19}^* = 1 - E/2$. It can be seen that E and μ_{19}^* are the discrete forms of Dev_{13} and μ_{19} respectively.

The above algorithm can be used to compute Dev_{11} , Dev_{12} , μ_{17} and μ_{18} also. In that case, E_j^r will involve computations of D_3 and D_6 whose computational techniques have been given in Chapter 4.

6.5 Results and Discussion

The hierarchical steps of the algorithm described in the previous section are illustrated using a simple input picture (Fig.6.2(a)). The directional codes d_i 's are shown in Fig.6.2(d). We start with a high level of hierarchy $r=12$ and d_j^r , E_j^r and a_j^r are shown in Fig.6.2(e). The axis a_2^r is the best axis of symmetry (shown by dots) at this level (Fig.6.2(b)). Next a low level, $s=6$, is considered and d_m^s , E_m^s and a_m^s are shown in Fig.6.2(f). The axis a_4^s is the best axis of symmetry (shown by dots) at this level (Fig.6.2(c)).

From Fig.6.2(b) and (c) it is clear that the best axis of symmetry at level $s=6$ is better than the best axis of symmetry at level $r=12$.

In Fig.6.3, errors E_j^r are plotted for a rectangle with side ratio 2:5. E_j^r are computed by the algorithm described in the previous section at a medium level of hierarchy. Among the errors there are two local minima which correspond to the two axes of symmetry of the rectangle.

The values of degrees of symmetry μ_7 , μ_8 and μ_9 for shapes A_1 , A_2 , A_3 and A_4 in Fig.6.4 are given in Table 6.1. Monotonicity of these values is consistent with human visual judgement. Similar results are obtained for antisymmetry (Table 6.2) where the values of μ_{10} and μ_{11} are given for shapes B_1 , B_2 and B_3 in Fig.6.5.

A right angled triangle is shown in Fig.6.6 (the ratio of the lengths of the base and the perpendicular of the triangle is $2/3$). All the measures of degree of symmetry of a region discussed in this chapter are computed for this triangle. Q_1Q_2 and Q_3Q_4 are the major and minor axes respectively. Q_i^* is the point on the boundary such that the arc distances between Q_i and Q_i^* in the clockwise and anticlockwise directions are equal ($i=1, \dots, 4$). We have to choose the best axis of symmetry from among the four lines $Q_iQ_i^*$.

($i=1, \dots, 4$) on the basis of D_3 , D_6 and Dev (equation numbers 6.14, 6.15, 6.16). For the right angled triangle in Fig.6.6, the best axis of symmetry is $Q_2 Q_2^*$ in all the three cases. The values of Dev_6 , Dev_7 , Dev_8 , μ_{12} , μ_{13} and μ_{14} are given in Table 6.3. In Fig.6.7 are shown the three best axes of symmetry found hierarchically (at a low level of hierarchy) on the basis of D_3 , D_6 and Dev (equation numbers 6.24, 6.25, 6.26). Among these three best axes of symmetry, only the one based on D_6 coincides with the axis of symmetry found in Fig.6.6 (on the basis of D_6). The two axes of symmetry based on D_3 in Fig.6.6 and Fig.6.7 differ only slightly while the two axes of symmetry based on Dev in Fig.6.6 and Fig.6.7 differ very much. However, each of the above three best axes of symmetry is very close to either the major axis or the minor axis. It is also seen that the three best axes of symmetry need not coincide with one another for an asymmetric or nearly symmetric region. The values of Dev_{11} , Dev_{12} , Dev_{13} , μ_{17} , μ_{18} and μ_{19} for the region in Fig.6.7 are given in Table 6.3. Note that μ_{12} , μ_{13} and μ_{14} are less than or equal to μ_{17} , μ_{18} and μ_{19} respectively.

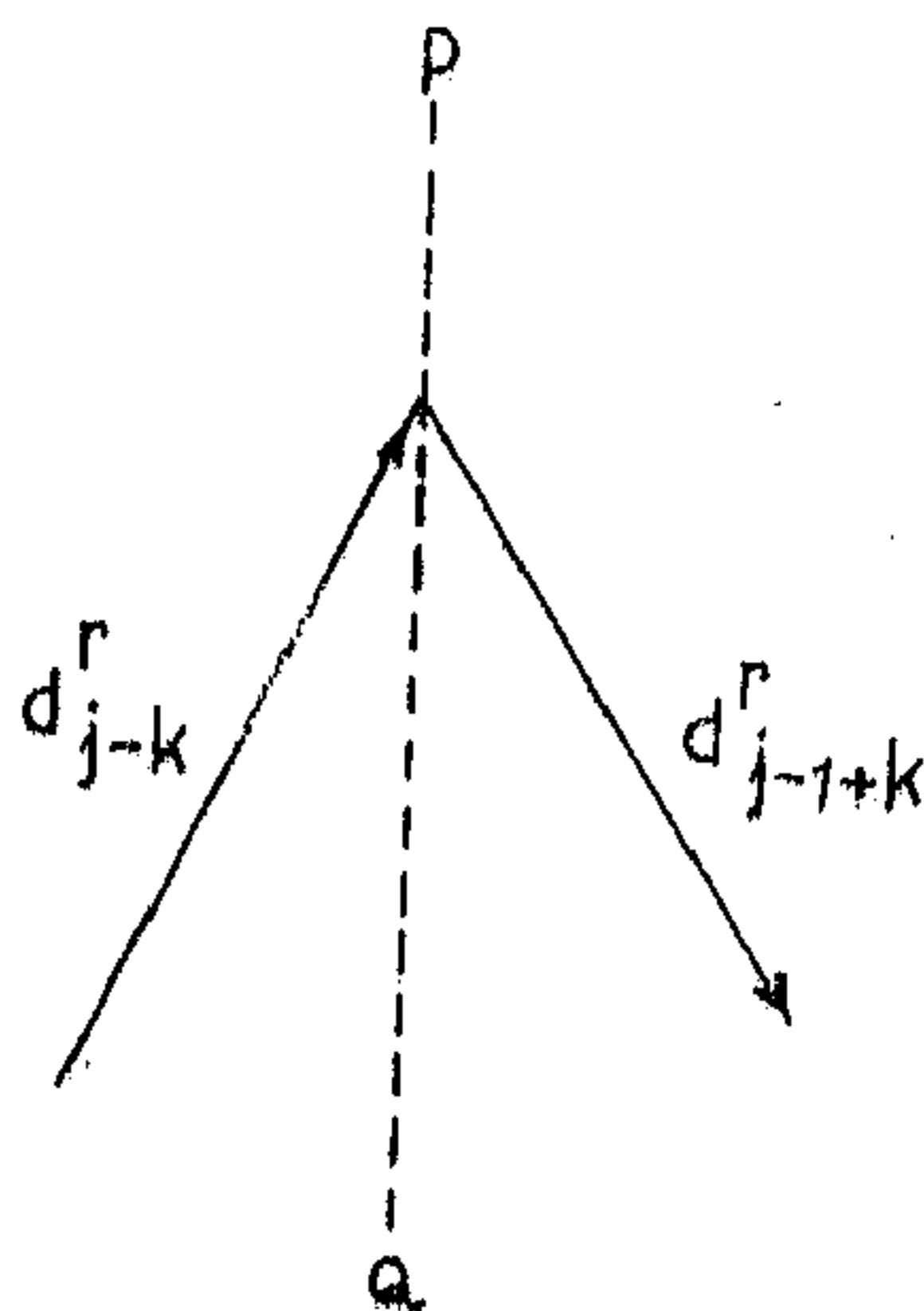


Fig. 6.1 The absolute directional code of the angular bisector PQ is d_k^* .

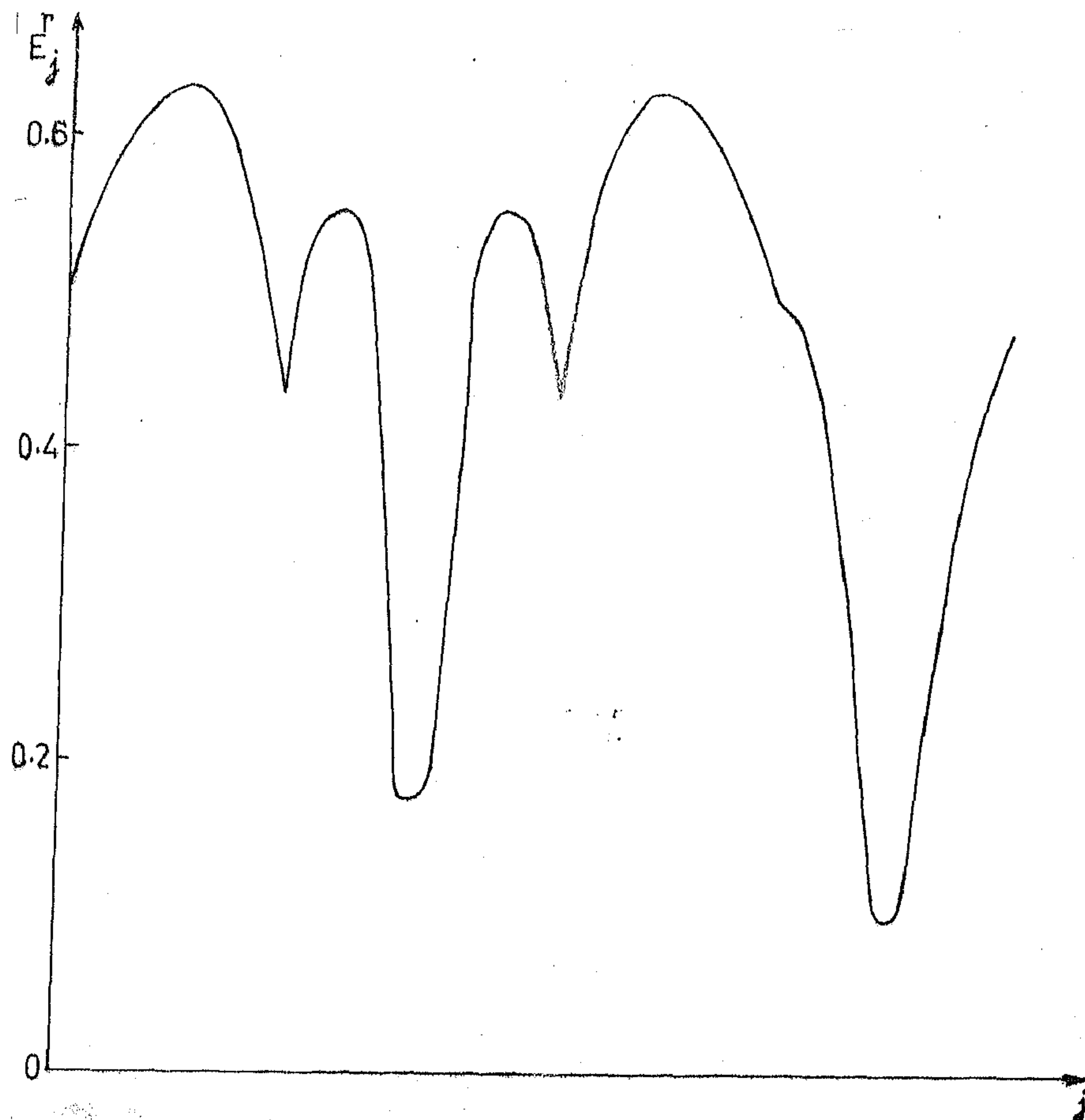


Fig. 6.3 Errors E_j^r at a medium level of hierarchy for a rectangle with side ratio 2:5. The two local minima in errors with sufficiently small absolute values (0.18 and 0.10) indicate the two axes of symmetry of the rectangle.

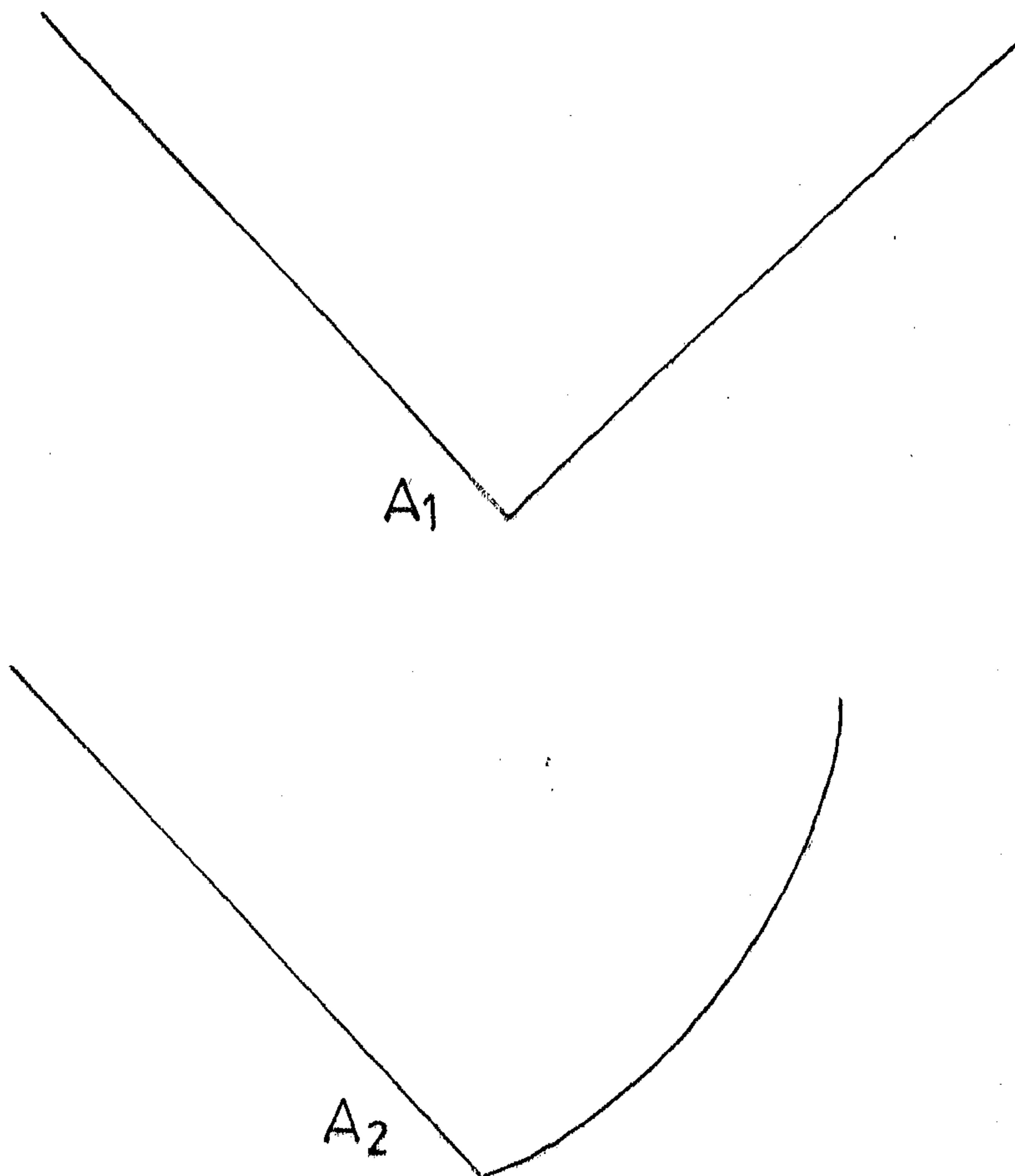


Fig.6.4(a) The values of μ_7, μ_8 and μ_9 for the above shapes are given in Table 6.1.

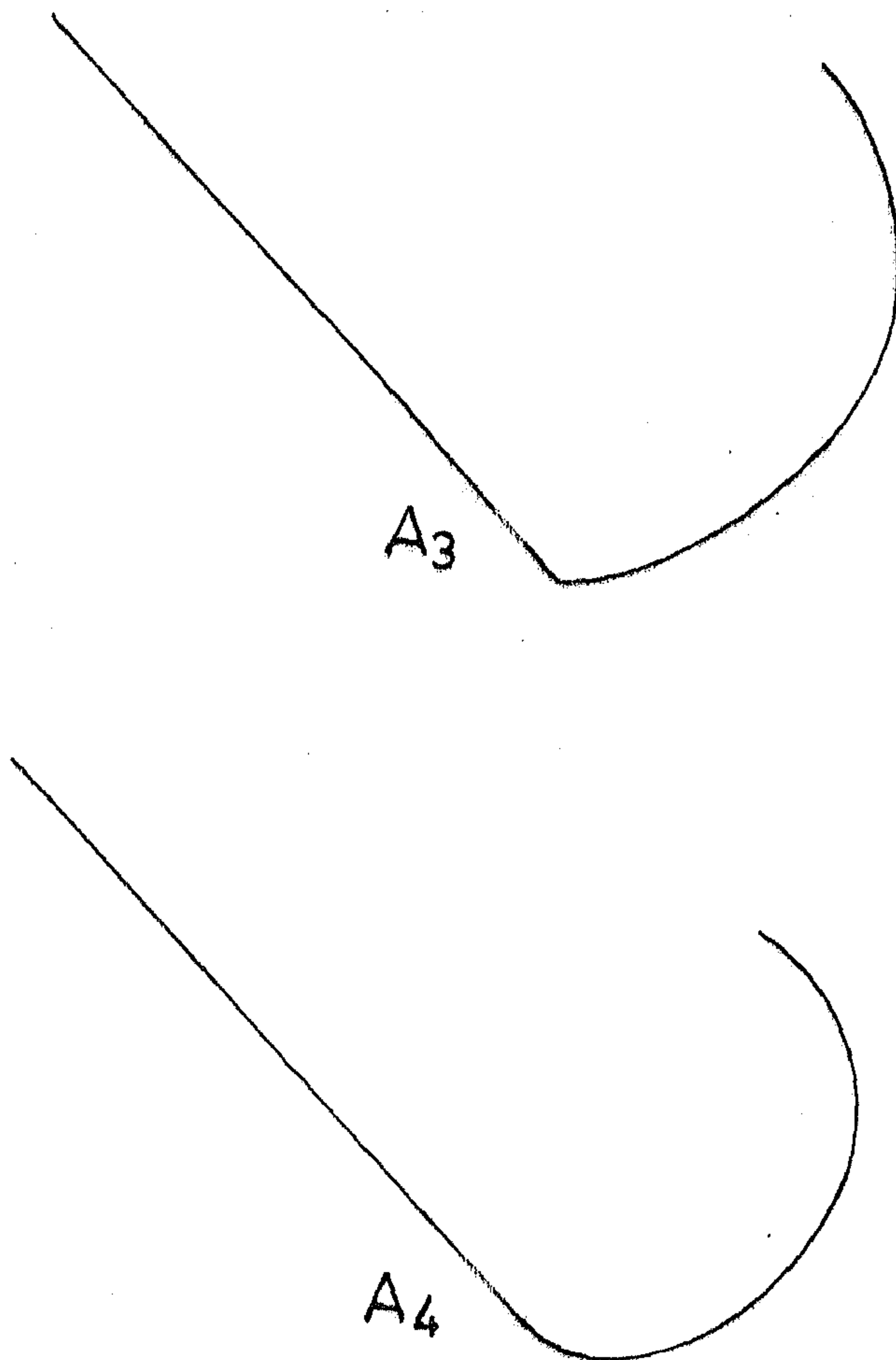


Fig.6.4(b) The values of μ_7 , μ_8 and μ_9 for the above shapes are given in Table 6.1.

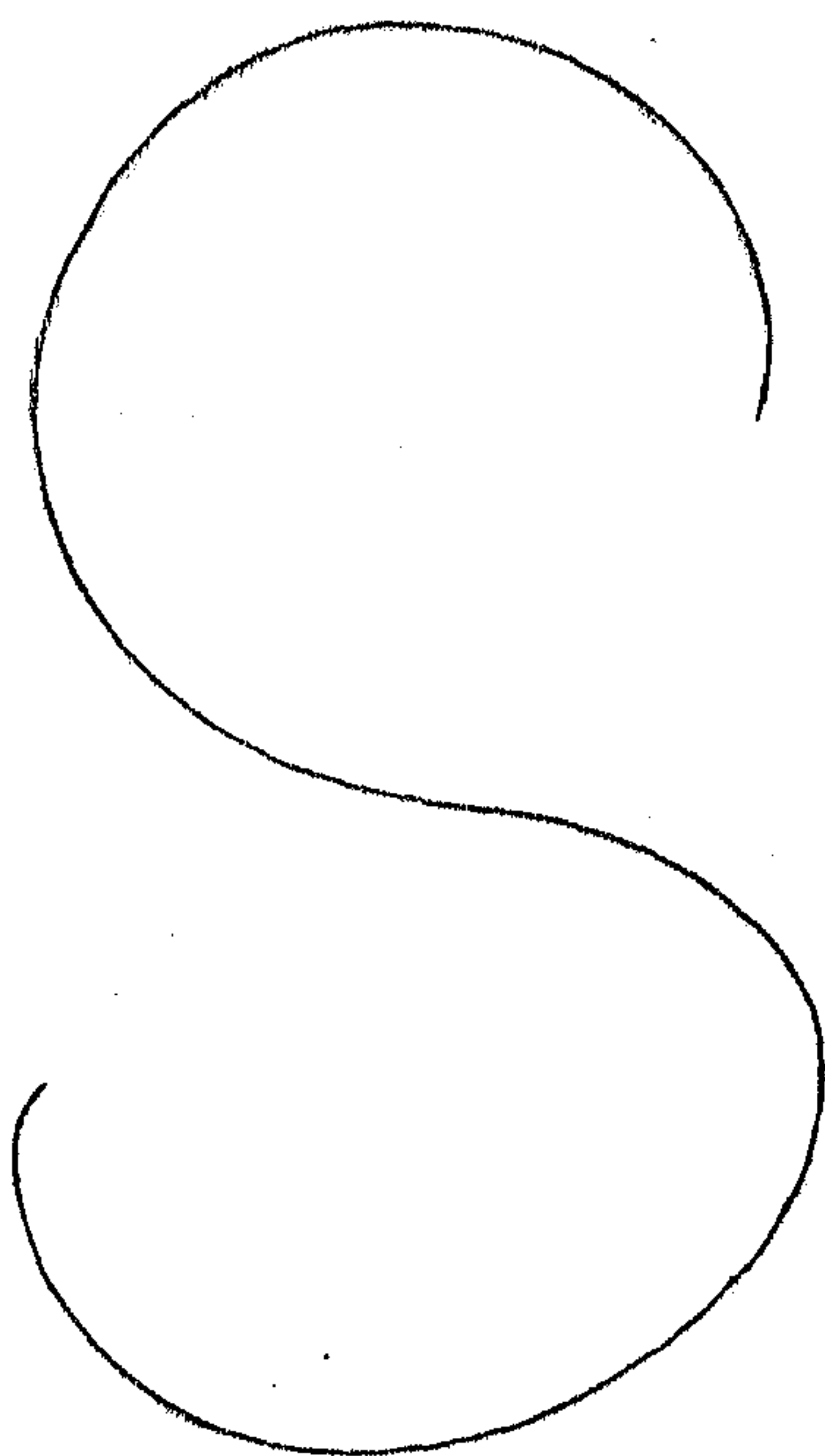
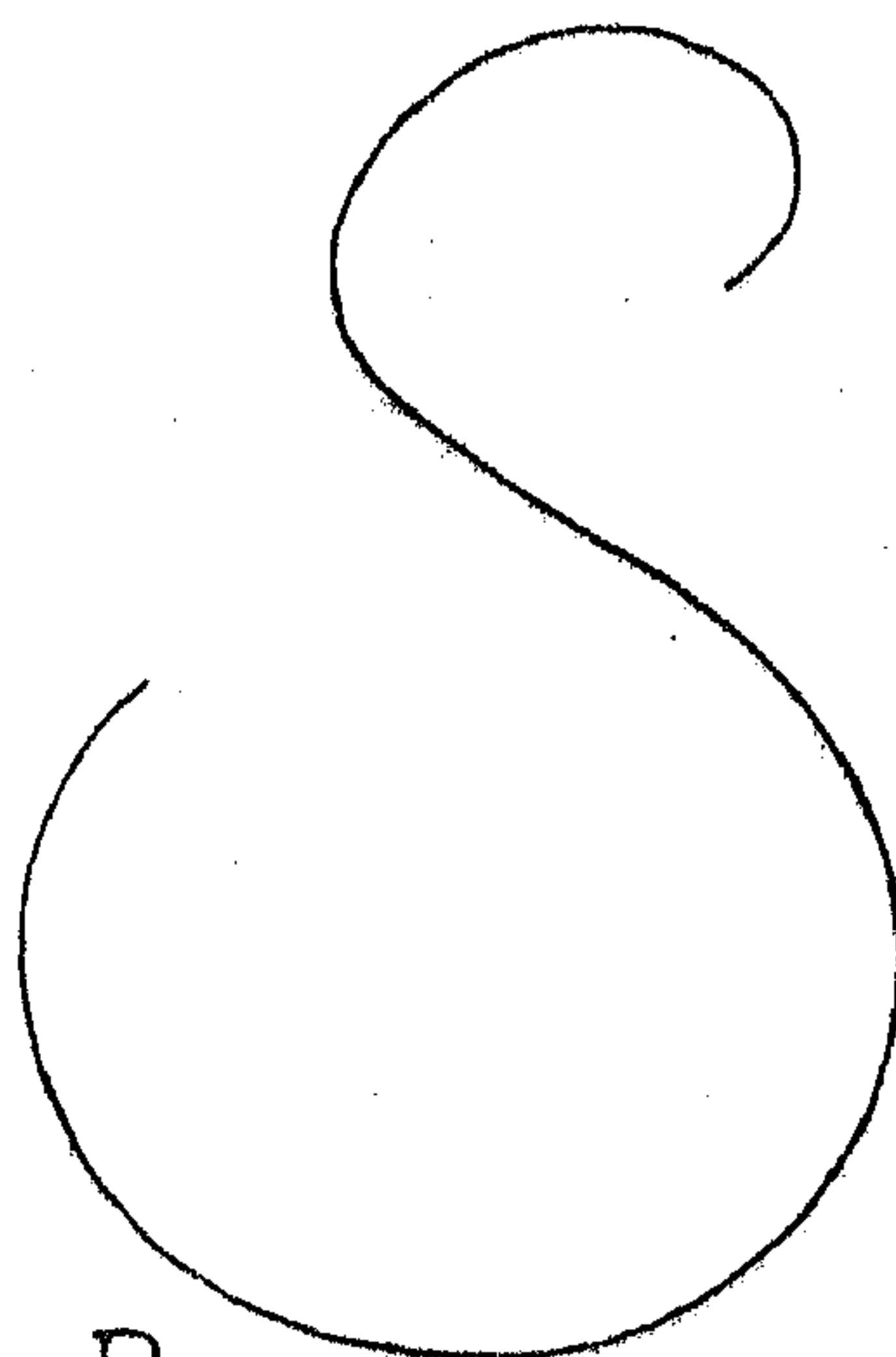
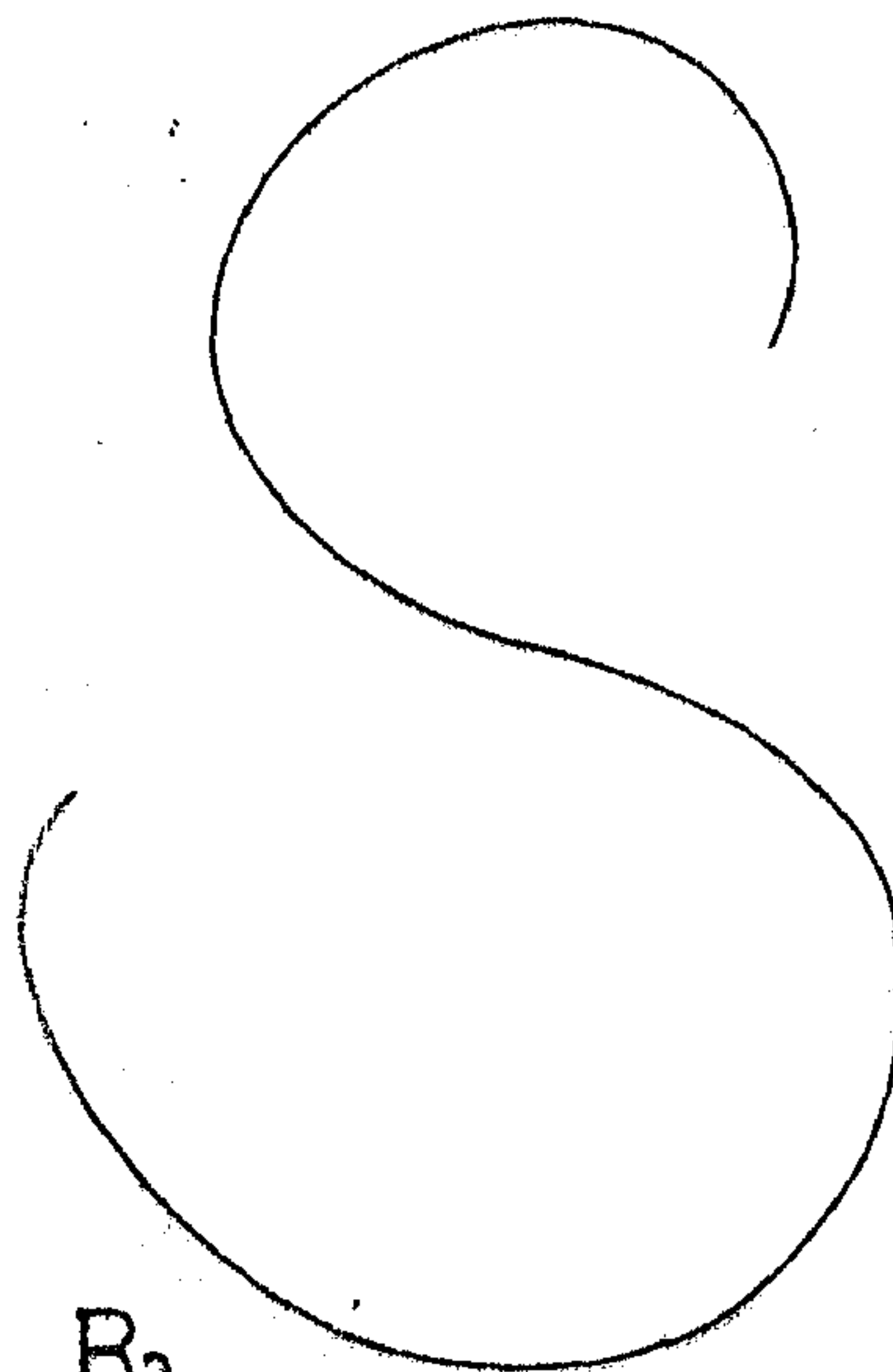
 B_1  B_3  B_2

Fig.6.5 The values of μ_{10} and μ_{11} for the above shapes are given in Table 6.2.

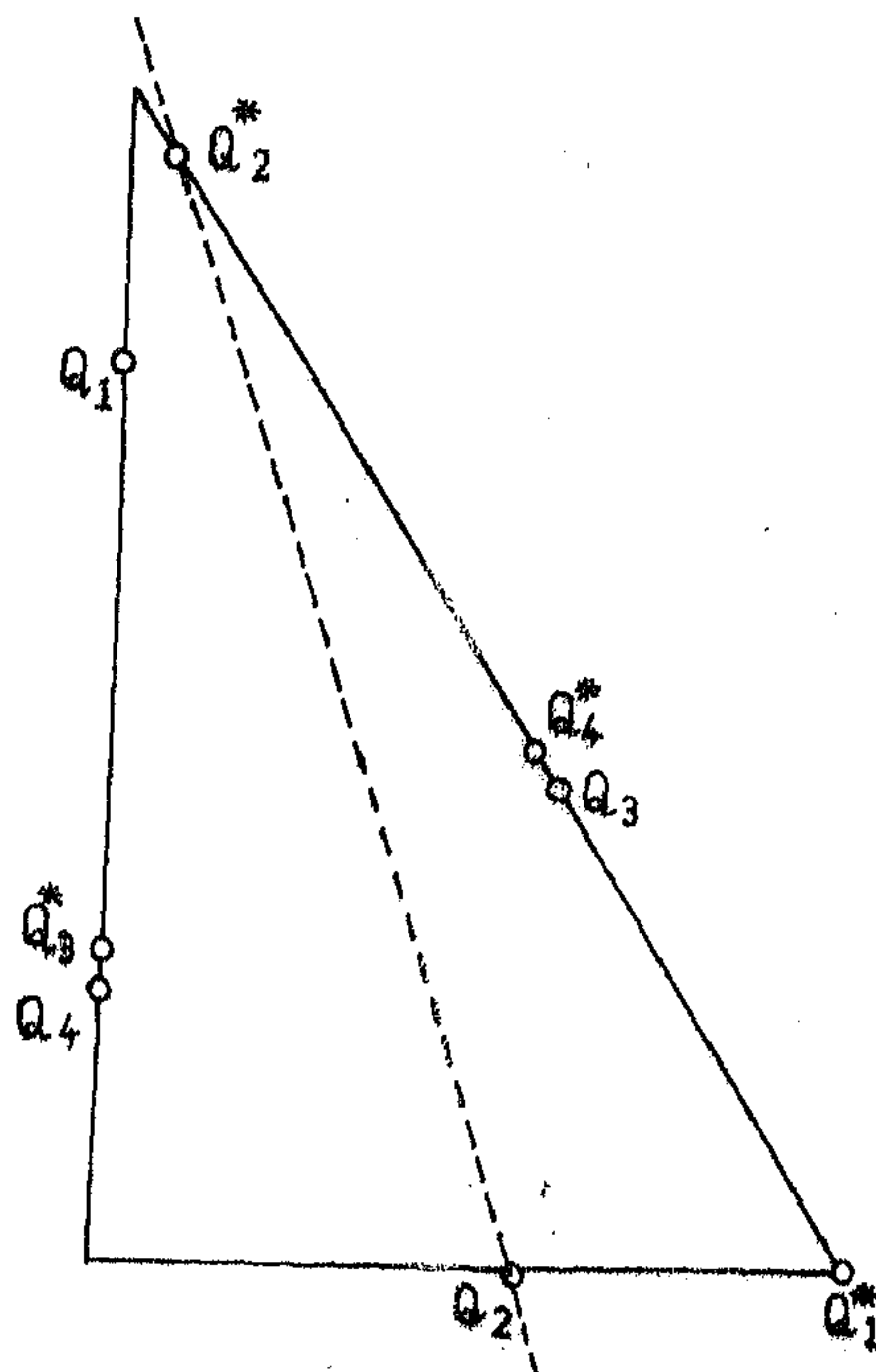


Fig.6.6 The region above is a right angled triangle whose height is 1.5 times its base, and hence not perfectly symmetric. Q_1Q_2 and Q_3Q_4 are the major and minor axes respectively. Q_1^* is the point on the boundary such that the arc distances between Q_1 and Q_1^* in the clockwise and anticlockwise directions are equal ($i=1, \dots, 4$). On the basis of each of D_3, D_6 and Dev , $Q_2Q_2^*$ is found to be the best axis of symmetry among all $Q_iQ_i^*$ ($i=1, \dots, 4$). The corresponding values of $Dev_6, Dev_7, Dev_8, \mu_{12}, \mu_{13}$ and μ_{14} are given in Table 6.3.

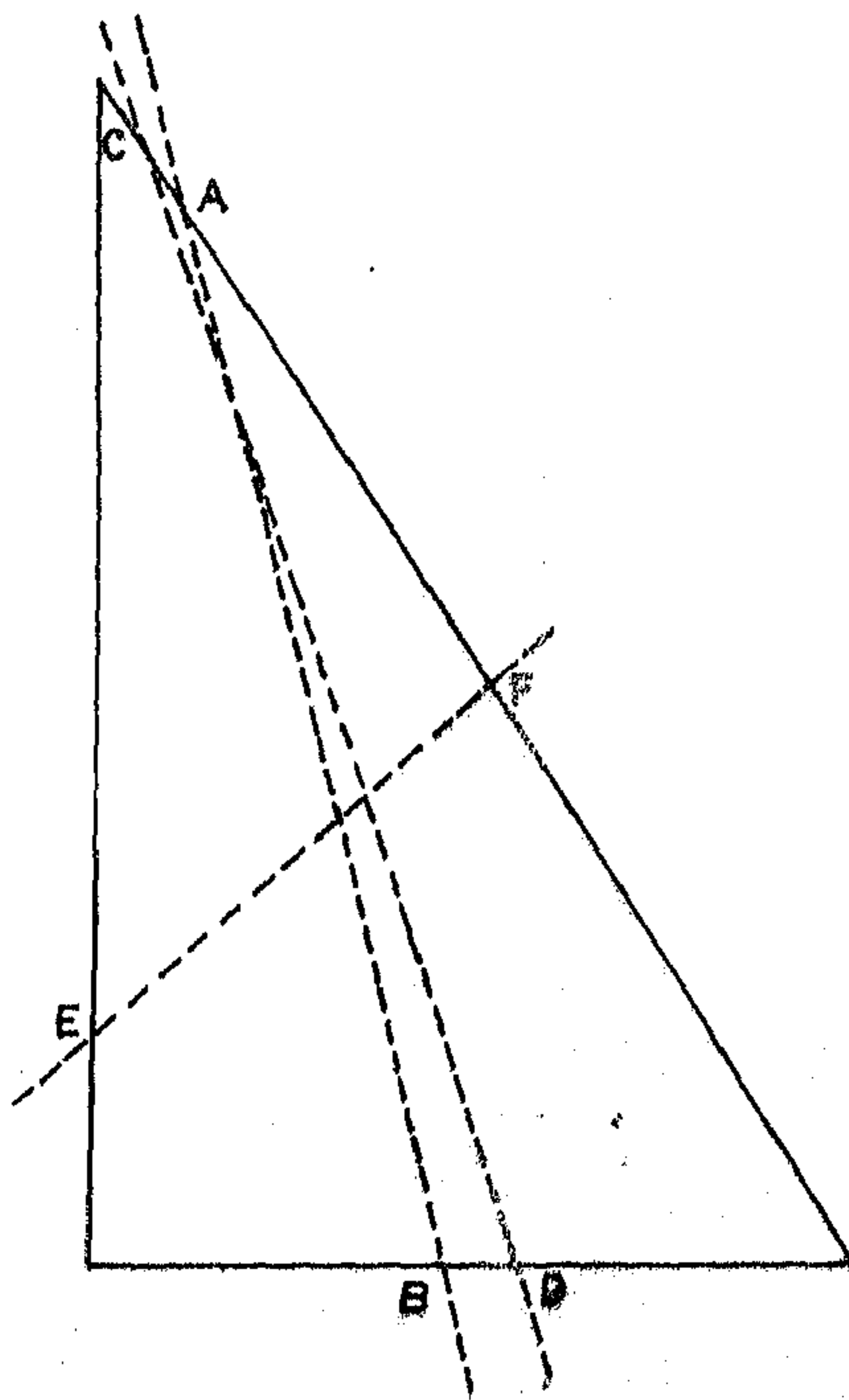


Fig.6.7 The region above is the same as in Fig.6.6. AB, CD and EF are the best axes of symmetry found hierarchically (at a low level of hierarchy) on the basis of D_3 , D_6 and Dev respectively. Note that AB is very close to $Q_2Q_2^*$ (Fig.6.6) while CD coincides with $Q_2Q_2^*$ and EF is far away from $Q_2Q_2^*$. The corresponding values of Dev_{11} , Dev_{12} , Dev_{13} , μ_{17} , μ_{18} and μ_{19} are given in Table 6.3.

Table 6.1

The values of degrees of symmetry μ_7 , μ_8 and μ_9 for shapes in Fig.6.4

	A_1	A_2	A_3	A_4
μ_7	1.0000	0.8944	0.7556	0.6498
μ_8	1.0000	0.9007	0.7484	0.6159
μ_9	1.0000	0.8908	0.7743	0.7090

Table 6.2

The values of degrees of antisymmetry μ_{10} and μ_{11} for shapes in Fig.6.5

	B_1	B_2	B_3
μ_{10}	0.9387	0.8394	0.7230
μ_{11}	0.9513	0.8438	0.8067

Table 6.3

For the right angled triangle in Fig.6.6 and Fig.6.7, the values of different deviations from symmetry and degrees of symmetry alongwith the corresponding best axes of symmetry are given below.

$$\text{Dev}_6 = 0.4294, \mu_{12} = 0.7853$$

$$\text{Dev}_7 = 0.0348, \mu_{13} = 0.9148$$

$$\text{Dev}_8 = 0.3128, \mu_{14} = 0.8436$$

In all the three cases above, the corresponding best axis of symmetry is $Q_2Q_2^*$ in Fig.6.6.

$$\text{Dev}_{11} = 0.4170, \mu_{17} = 0.7915$$

The corresponding best axis of symmetry is AB in Fig.6.7.

$$\text{Dev}_{12} = 0.0348, \mu_{18} = 0.9148$$

The corresponding best axis of symmetry is CD in Fig.6.7.

$$\text{Dev}_{13} = 0.2104, \mu_{19} = 0.8948$$

The corresponding best axis of symmetry is EF in Fig.6.7.

CHAPTER 7

Conclusion

In the present thesis we have dealt with 2-dimensional shapes giving emphasis to defining various measures of shape distance and shape similarity for 2-dimensional objects. Already existing shape distances are mainly based on scalar transforms of shapes. But in this thesis all the shape distances except one involve structural information of shapes and are intuitively appealing.

Scope of future research lies in extending some of the definitions and algorithms proposed in the present thesis to 3 dimensions. In order to extend the shape distance D_4 of 2-dimensional curves to 3 dimensions, one has to define 3-dimensional continuous directional codes on the basis of the 26 discrete directional codes in 3 dimensions. The extension of the other shape distance D_7 of 2-dimensional curves to 3 dimensions is straightforward.

The shape distance D_8 of 2-dimensional regions (based on shape vector) also can easily be extended to 3 dimensions. For this purpose, concentric spheres in 3 dimensions instead of concentric circles in 2 dimensions are to be considered. In order to extend the shape distance D_{11} of 2-dimensional regions (based on unmatched area), one has to consider the major axis and

the major plane of a 3-dimensional object and the unmatched volume (discussed in Section 5.8). But the extension of the shape distances D_{12} and D_{13} of 2-dimensional regions to 3 dimensions is not straightforward. A possible solution to the problem is as follows. Two 3-dimensional solids of equal volume are standardized so that their centres of gravity, their major axes and their major planes coincide. Let the minor plane be the plane which is perpendicular to the major plane and contains the major axis. Now, several equidistant planes which are parallel to the common major plane are considered. Each such plane gives rise to a closed boundary (of the cross section) for each of the two solids. A shape distance between these two closed boundaries can be defined on the basis of D_3 or D_6 (depending on whether D_{12} or D_{13} is to be extended). The problem of finding the starting point on these boundaries can be solved with the help of the straight line where the plane in question intersects with the minor plane. Thus, for each of the parallel planes, the value of D_3 or D_6 is obtained. An average of these values over all the equidistant parallel planes provides a shape distance between the two 3-dimensional solids.

The extension to 3 dimensions of the algorithms for propagation, shrinking and rotation proposed in Chapter 2 of the present thesis is straightforward as already mentioned in Section 2.4.

The thinning algorithms proposed in Chapter 3 of the present thesis also can be extended to 3 dimensions. For this purpose, one has to consider 26 directional codes in 3 dimensions instead of 8 directional codes in 2 dimensions. Skeletal voxels can be sequentially found in a similar fashion. In order to obtain the starting skeletal voxel, first a voxel is chosen at random and then three mutually perpendicular cross sections (along xy , yz and xz planes) containing that voxel are considered. The cross section with the minimum area or minimum perimeter is considered and some voxel in the middle of that cross section is chosen as the starting skeletal voxel. From this starting voxel, the sequential search for the other skeletal voxels is made in the two directions (in 3 dimensions) perpendicular to the plane of the cross section with the minimum area or minimum perimeter. In 2 dimensions, at the time of finding the skeletal pixels sequentially, two directions (DIR_{-1} and DIR_1) apart from the original direction DIR are considered. In 3 dimensions, apart from the original direction DIR , 8 other (3-dimensional) directions are to be considered.

The extension to 3 dimensions of some of the definitions of measure of degree of symmetry and antisymmetry proposed in Chapter 6 of the present thesis is possible. The degrees of symmetry and antisymmetry $\mu_7, \mu_8, \mu_9, \mu_{10}$ and μ_{11} for 2-dimensional curves can be extended for 3-dimensional curves, since D_3 and D_6

can be extended to 3 dimensions as mentioned earlier. But it is quite difficult to extend the hierarchical search for the best axis of symmetry of a 2-dimensional region to a hierarchical search for the best plane of symmetry of a 3-dimensional solid. However, the extension to 3 dimensions of the degrees of symmetry of a 2-dimensional region based on its major axis is possible. For this purpose, one needs the following result: if a 3-dimensional solid is symmetric, then its plane of symmetry is either its major plane or its minor plane or the plane which is perpendicular to both its major and minor planes and passes through its centre of gravity. Several equidistant planes parallel to the major plane are considered (as in the case of extension of D_{12} and D_{13}). On both sides of the major plane, closed boundaries (of the cross sections) are obtained. Each pair of the closed boundaries which are equidistant from the major plane is considered and a similarity measure between the two is computed through some matching. Thus, a measure of degree of symmetry of the solid is obtained with respect to the major plane. Similarly, the other two mutually perpendicular planes (i.e., the minor plane and the plane which is perpendicular to both the major and minor planes) are considered and the plane of symmetry is chosen as the best among the three.

Appendix A

Appendix A.1 Since A_i is bounded, there exists a bounded disc D containing A_i . So, D^c is a subset of $A_i^c = \bigcup_{j=1}^{m_i} B_{ij}$. Since D^c is connected, D^c is contained in exactly one B_{ij} because $B_{ij} \cup B_{ik}$ is disconnected for $j \neq k$. Wolog, let D^c be a subset of B_{i1} . Since D^c is unbounded, B_{i1} is unbounded.

Now, consider any B_{ij} such that $j > 1$. Since $B_{ij} \cup B_{i1}$ is disconnected, B_{ij} is a subset of B_{i1}^c . Again since D^c is a subset of B_{i1} , B_{i1}^c is a subset of D . So, B_{ij} is a subset of D which is bounded. Hence B_{ij} is bounded for $j > 1$.

Appendix A.2 If no A_r is a subset of some B_{ij} ($j > 1$), then $\bigcup_{j=2}^{m_i} B_{ij} = \bigcup_{i=1}^n A_i = \bigcup_{j=2}^{m_i} B_{ij}$. In that case Proposition 1.2.3 is immediate from condition (e) in (1.1).

Suppose, some A_r is a subset of some B_{ij} ($j > 1$). So, A_r can not have intersection with other B_{ij} 's.

Enough to show that $(B_{ij} - A_r)$ can be expressed as a finite union $\bigcup_{p=1}^v D_{ip}$ where D_{ip} satisfies conditions (a), (b) and (c) in (1.1) and $D_{ip} \cup D_{iq}$ is disconnected for $p \neq q$.

$$\begin{aligned}
B_{ij} - A_r &= B_{ij} \cap A_r^c \\
&= B_{ij} \cap (B_{r1} \cup B_{r2} \cup \dots \cup B_{rm_r}) \\
&= (B_{ij} \cap B_{r1}) \cup (B_{ij} \cap B_{r2}) \cup \dots \cup (B_{ij} \cap B_{rm_r}) \\
&= D_{i1} \cup D_{i2} \cup \dots \cup D_{im_r}
\end{aligned}$$

Result 1 For any A_i and B_{ij} , $A_i \cup B_{ij}$ is connected.

Proof Consider the limit points of B_{ij} . Since B_{ij} is open, some of these limit points fall outside B_{ij} . They can not belong to any B_{ik} ($k \neq j$) since $B_{ij} \cup B_{ik}$ is disconnected. So, some limit points of B_{ij} belong to A_i . Hence $A_i \cup B_{ij}$ is connected.

(i) So, from Result 1, $A_r \cup B_{rp}$ is connected for all $p=1, 2, \dots, m_r$. Since B_{ij} is open and contains A_r , $B_{ij} \cap B_{rp}$ is non-empty for all $p=1, 2, \dots, m_r$. Hence D_{ip} is non-empty. Since D_{ip} is open, D_{ip} satisfies condition (a) in (1.1).

(ii) To show that D_{ip} satisfies condition (b) in (1.1) for all p .

First, we will show D_{ip} satisfies condition (b) for $p > 1$. Enough to show that $D_{ip} = B_{rp}$ for $p > 1$. That is, to show that B_{rp} is a subset of B_{ij} for $p > 1$. Since A_r is a subset of B_{ij} , B_{ij}^c is a subset of A_r^c . So, $(\bigcup_{k \neq j} B_{ik}) \cup A_i$ is a subset of $\bigcup_{p=1}^{m_r} B_{rp}$. Now, A_i is a subset of exactly one B_{rp} (from Proposition 1.2.2), say B_{rp_0} . Take any B_{ik} ($k \neq j$). Since $B_{ik} \cup A_i$ is connected (from Result 1) and

B_{rp_0} contains A_i , $B_{ik} \cap B_{rp_0}$ is non-empty. We will show that B_{ik} is a subset of B_{rp_0} . Suppose not, i.e., suppose B_{ik} has some intersection with B_{rp_1} where $p_0 \neq p_1$. Let, wlog, $B_{rp_0} \cup B_{rp_1}$ contain B_{ik} .

Now, $B_{ik} = (B_{ik} \cap B_{rp_0}) \cup (B_{ik} \cap B_{rp_1})$ is connected.

Again, $B_{rp_0} \cup B_{rp_1} = (B_{ik} \cap B_{rp_0}) \cup (B_{ik}^c \cap B_{rp_0}) \cup (B_{ik} \cap B_{rp_1}) \cup (B_{ik}^c \cap B_{rp_1})$

Note that $(B_{ik} \cap B_{rp_0}) \cup (B_{ik} \cap B_{rp_1}) = B_{ik}$ is connected.

Also, $(B_{ik} \cap B_{rp_0}) \cup (B_{ik}^c \cap B_{rp_0}) = B_{rp_0}$ and $(B_{ik} \cap B_{rp_1}) \cup (B_{ik}^c \cap B_{rp_1}) = B_{rp_1}$ are connected. Hence $B_{rp_0} \cap B_{rp_1}$ is connected. It is a contradiction because $B_{rp_0} \cap B_{rp_1}$ is disconnected since $p_0 \neq p_1$.

Hence $\bigcup_{k \neq j} B_{ik}$ and A_i are subsets of B_{rp_0} . Since $j > 1$, B_{i1} is a subset of B_{rp_0} . Since B_{i1} is unbounded, B_{rp_0} also is unbounded. So, $B_{rp_0} = B_{r1}$. Thus A_i and $\bigcup_{k \neq j} B_{ik}$ are subsets of B_{r1} . So, B_{r1}^c is a subset of the complement of $(\bigcup_{k \neq j} B_{ik}) \cup A_i$ which is B_{ij} .

Hence B_{rp} is a subset of B_{ij} for $p > 1$.

Above we have shown that D_{ip} satisfies condition (b)

for $p > 1$. Now, to show that $D_{i1} = B_{ij} \cap B_{r1}$ satisfies condition (b).

Result 2 If a set A satisfies condition (b) with some $d > 0$, then A satisfies condition (b) with any d_0 where $d > d_0 > 0$.

Proof. This is because any disc D of diameter d can be expressed as union of discs of diameter d_0 which are contained in D .

Since $B_{r1} \cup B_{rp}$ is disconnected for $p > 1$ and $B_{ij} \cup B_{ik}$ is disconnected for $k \neq j$, any disc which is not a subset of D_{i1} but has intersection with D_{i1} must have intersection with either A_i or A_r or both. Since A_i and A_r are disjoint compact sets, $\text{Dist}(A_i, A_r) > 0$. Let $\mu = \min\{\text{Dist}(A_i, A_r), \lambda(B_{ij}), \lambda(B_{r1})\}$. Since B_{r1} is unbounded, $\lambda(B_{r1})$ may not exist. In that case, we ignore $\lambda(B_{r1})$ in defining μ above. We will show that D_{i1} satisfies condition (b) with any d less than $\mu/4$. Let x belong to D_{i1} . Let $d_i = \text{Dist}(x, A_i)$ and $d_r = \text{Dist}(x, A_r)$. Note that $d_i + d_r \geq \text{Dist}(A_i, A_r) \geq \mu$. So, either $d_i \geq \mu/2$ or $d_r \geq \mu/2$. Let, wolg, $d_i \geq \mu/2$. Now, since B_{r1} satisfies condition (b) with $d = \lambda(B_{r1})$, from Result 2 above B_{r1} satisfies condition (b) with $\mu/4$ which is less than d . So, there is a closed disc D of diameter $\mu/4$ such that $x \in D$ and D is a subset of B_{r1} . Note that D has no intersection with A_r . Since $\text{Dist}(x, A_i) \geq \mu/2$, D cannot have intersection with A_i . Hence, D is contained in D_{i1} . Proof will be similar if $d_r \geq \mu/2$. Hence D_{i1} satisfies condition (b).

Result 3 Suppose, a set A satisfies conditions (b) and (c) in (1.1) with some $d > 0$. Then A satisfies conditions (b) and (c)

with any d_0 where $d > d_0 > 0$.

Proof. Result concerning condition (b) has been proved in Result 2. Now to show that $C_{d_0}(A)$ is path-connected. Let $B = \{x; \text{there exists } y \text{ in } C_d(A) \text{ such that } (d-d_0)/2 \geq d(x,y)\}$. It can be seen that $B = C_{d_0}(A)$. Enough to show that B is path-connected. Let x_1 and x_2 be two elements in B . So, there exist $y_i (i=1,2)$ in $C_d(A)$ such that $(d-d_0)/2 \geq d(x_i, y_i) = \varepsilon_i$, say. So, x_i and y_i belong to the closed disc with centre at y_i and diameter ε_i , which is a subset of B . So, there is a path in B joining x_i and $y_i (i=1,2)$. Again, since y_1, y_2 belong to $C_d(A)$ which is path-connected, there is a path in B joining y_1 and y_2 . Thus, there is a path in B joining x_1 and x_2 . Hence Result 3.

To show that D_{ip} satisfies condition (c) in (1.1) for all p . In (ii) we have shown that $D_{ip} = B_{rp}$ for $p > 1$. Hence, D_{ip} satisfies condition (c) for $p > 1$. To show that D_{i1} satisfies condition (c). That is, to show that $C_d(D_{i1})$ is path-connected for some $d > 0$. Above we have shown that D_{i1} satisfies condition (b) for some d , say d_0 . Let $C_{d_0}(D_{i1})$ be not path-connected. Since D_{i1} is open, $C_{d_0}(D_{i1})$ is open. Wlog, let $C_{d_0}(D_{i1}) = E_1 \cup E_2$ where E_1 and E_2 are open and connected (for an open set, connectedness means path-connectedness) and E_1 and E_2 are disjoint. Let F_j be the union of all closed discs of diameter d_0 with centre in $E_j (j=1,2)$.

Since E_j is open, F_j is open. Now, $D_{i1} = F_1 \cup F_2$ is connected. Hence $F_1 \cap F_2$ is non-empty. Let x belong to $F_1 \cap F_2$ which is open. So, there is a disc D of positive diameter such that $x \in D$ and D is a subset of $F_1 \cap F_2$. Thus, $d_0/2 > \text{Dist}(x, E_j)$ for $j=1,2$. Now, there exists y_j in E_j such that $d_0/2 > d(x, y_j)$. Let $\varepsilon/2 = \max_j d(x, y_j)$ which is less than $d_0/2$. Let $d_1 = d_0 - \varepsilon$. We will show that $C_{d_1}(D_{i1})$ is path-connected. We have seen above $E_j = C_{d_0}(F_j)$ is path-connected. From Result 3, $C_{d_1}(F_j)$ is path-connected. Also, from the proof of Result 3, $C_{d_1}(F_j) = \{z : \text{there exists } y \text{ in } E_j \text{ such that } (d_0 - d_1)/2 > d(y, z)\}$. Now, $(d_0 - d_1)/2 = \varepsilon/2$. So, $x \in C_{d_1}(F_j)$ since $\varepsilon/2 > d(x, y_j)$. Thus, $C_{d_1}(F_1) \cap C_{d_1}(F_2)$ is non-empty. So, $C_{d_1}(F_1) \cup C_{d_1}(F_2)$ is path-connected. But $C_{d_1}(F_1) \cup C_{d_1}(F_2) = C_{d_1}(D_{i1})$. Hence the result.

Appendix B

Appendix B.1 Let $-x = x \cos \alpha - y \sin \alpha$ (1)

and $-y = x \sin \alpha + y \cos \alpha$ (2)

such that at least one of x and y is non-zero and α belongs to $[0, 2\pi]$.

Without loss of generality assume, $x \neq 0$.

From (1) $y \sin \alpha = x + x \cos \alpha$ (3)

From (2) $-y \sin \alpha = x \sin^2 \alpha + y \sin \alpha \cos \alpha$ (4)

Using (3) and (4),

$$-x - x \cos \alpha = x \sin^2 \alpha + (x + x \cos \alpha) \cos \alpha$$

i.e. $x(\sin^2 \alpha + \cos^2 \alpha + \cos \alpha + 1 + \cos \alpha) = 0$

i.e. $x(1 + \cos \alpha) = 0$

i.e. $1 + \cos \alpha = 0$

Hence $\alpha = \pi$.

Appendix B.2 Let $A = \{(u(s), v(s)) : s \in [0, L]\}$ be an open curve.

Since the curve is piecewise smooth, $u'(s)$ and $v'(s)$ exist and are continuous everywhere except at finitely many points. It is sufficient to show that if $u'(s)$ and $v'(s)$ exist then $(u(s), v(s))$ can not be a corner point. Now, it can be seen that $(u'(s))^2 + (v'(s))^2 = 1$ for all s . So, $u'(s)$ and $v'(s)$ can not be zero simultaneously.

Suppose, $u'(s) \neq 0$. Then it is easy to see that

$\lim_{h \rightarrow 0} \frac{v(s) - v(s-h)}{u(s) - u(s-h)} = \frac{v'(s)}{u'(s)}$. So, the first condition in

Definition 4.3.1 is not satisfied. So, $(u(s), v(s))$ is not a corner point.

Suppose, $u'(s) = 0$. So, $v'(s) \neq 0$.

Now, $\lim_{h \downarrow 0} \frac{v(s) - v(s-h)}{u(s) - u(s-h)} = \lim_{h \downarrow 0} \frac{(v(s) - v(s-h))/h}{(u(s) - u(s-h))/h}$

Since the denominator goes to zero and the numerator goes to a non-zero finite number as $h \downarrow 0$, then the limit goes to either $+\infty$ or $-\infty$, depending on the sign of $v'(s)$. So, here the second condition in Definition 4.3.1 is not satisfied. Hence, $(u(s), v(s))$ is not a corner point.

Appendix B.3 Suppose, A and B have the same length. Suppose, λ -th point is not a corner point. Then $w(\lambda) = w_1(\lambda)$ implies $u'(\lambda L) = u_1'(\lambda L)$ and $v'(\lambda L) = v_1'(\lambda L)$ since $(u'(\lambda L))^2 + (v'(\lambda L))^2 = (u_1'(\lambda L))^2 + (v_1'(\lambda L))^2 = 1$. That is, $u'(s) = u_1'(s)$ and $v'(s) = v_1'(s)$.

Since, there are only finitely many corner points,
 $u(s_0) = \int_0^{s_0} u'(s) ds + \text{const} = \int_0^{s_0} u_1'(s) ds + \text{const} = u_1(s_0) + \text{const}$.

Similarly, $v(s_0) = v_1(s_0) + \text{const}$ for s_0 in $[0, L]$. Thus, B is a translation of A. If A and B have different lengths, then B is a translation and dilation of A.

Appendix B.4 Suppose, $(u(\lambda L), v(\lambda L))$ is not a corner point. So,

either LHS and RHS in Definition 4.3.1 are equal or one of them is $+\infty$ and the other $-\infty$. Suppose, $u'(\lambda L)$ and $v'(\lambda L)$ exist. Now, $e(\lambda) \geq 0$ and $\delta(\lambda) \geq 0$ implies $u'(\lambda L) = 0$ and $e(\lambda) \leq 0$ and $\delta(\lambda) \leq 0$ implies $u'(\lambda L) = 0$. So, $u'(\lambda L) \neq 0$ implies either $e(\lambda) = 1$ and $\delta(\lambda) = -1$ or $e(\lambda) = -1$ and $\delta(\lambda) = 1$.

Suppose, $u'(\lambda L) \neq 0$. So LHS = RHS. So,

$$\begin{aligned} f_1(1-\lambda) &= \lim_{h \downarrow 0} \frac{v_1(L-\lambda L) - v_1(L-\lambda L-h)}{u_1(L-\lambda L) - u_1(L-\lambda L-h)} \\ &= \lim_{h \downarrow 0} \frac{v(\lambda L) - v(\lambda L+h)}{u(\lambda L) - u(\lambda L+h)} \\ &= \lim_{h \downarrow 0} \frac{v(\lambda L) - v(\lambda L-h)}{u(\lambda L) - u(\lambda L-h)} \\ &= f(\lambda) \end{aligned}$$

Also, $e_1(1-\lambda) = \delta(\lambda)$. So from above either $e(\lambda) = 1$ and $e_1(1-\lambda) = -1$ or $e(\lambda) = -1$ and $e_1(1-\lambda) = 1$. So, from the construction of w and w_1 , $w_1(1-\lambda) = w(\lambda) + 4 \pmod{8}$.

Suppose, $u'(\lambda L) = 0$. So, $u'_1(L-\lambda L) = 0$. Also,

$$\begin{aligned} v'_1(L-\lambda L) &= \lim_{h \downarrow 0} \frac{v_1(L-\lambda L) - v_1(L-\lambda L-h)}{h} \\ &= \lim_{h \downarrow 0} \frac{v(\lambda L) - v(\lambda L+h)}{h} \\ &= -v'(\lambda L) \end{aligned}$$

So, either $\text{RHS} = +\infty$ and $\text{LHS} = -\infty$ or $\text{RHS} = -\infty$ and $\text{LHS} = +\infty$. From the construction of w and w_1 , either $w_1(1-\lambda) = 8$ and $w(\lambda) = 4$ or $w_1(1-\lambda) = 4$ and $w(\lambda) = 8$. So, $w_1(1-\lambda) = w(\lambda) + 4 \pmod{8}$.

Appendix B.5 Let $D_2(A, B) = 0$. That is, $\int_0^1 D_1(w_1(\lambda), w_2(\lambda)) d\lambda = 0$. But $f(\lambda) = D_1(w_1(\lambda), w_2(\lambda))$ is a non-negative and continuous function of λ . So, $f(\lambda) = 0$ for all λ in $(0, 1)$. Thus, $w_1(\lambda) = w_2(\lambda)$ for all λ . Hence from Appendix B.3, B is a translation and dilation of A .

Appendix B.6 Let $f(\alpha)$ be the function $D_2(A, B_\alpha)$ or $D_5(A, B_\alpha)$ or $D_9(A, B_\alpha)$ in the closed interval $[0, 2\pi]$. Since $f(\alpha)$ is continuous in $[0, 2\pi]$, $T = \{f(\alpha) : \alpha \in [0, 2\pi]\}$ is a closed set. So, $\sup_{\alpha \in [0, 2\pi]} f(\alpha)$ belongs to T . Hence $\sup_{\alpha \in [0, 2\pi]} f(\alpha) = f(\beta)$ for some β in $[0, 2\pi]$, since $f(0) = f(2\pi)$.

Appendix B.7 Let $A = \{(u(s), v(s)) : s \in [0, L]\}$ be an open curve with centroid at $(0, 0)$. So,

$$\int_0^L \{(u(s)-x)^2 + (v(s)-y)^2\} ds \geq \int_0^L (u^2(s) + v^2(s)) ds \quad (1)$$

for any point (x, y) . Let $u_0 = u(L/2)$ and $v_0 = v(L/2)$.

$$\text{So, } \int_0^L \{(u(s)-u_0)^2 + (v(s)-v_0)^2\} ds \leq \int_0^L (s-L/2)^2 ds, \text{ since } |s-L/2| \text{ gives the curve distance}$$

which is greater than or equal to the Euclidean distance between two points.

$$= \int_{-L/2}^{L/2} t^2 dt = L^3/12$$

From (1) $\int_0^L (u^2(s) + v^2(s)) ds \leq L^3/12$

Hence, $\int_0^1 (u^2(\lambda L) + v^2(\lambda L)) d\lambda \leq L^2/12$

Appendix B.8 Let $\underline{P} = \{P_j\}_0^n$ be the description of an open curve. Let $A_1 = \{(u_1(s), v_1(s)) : s \in [0, L]\}$ and $A_2 = \{(u_2(s), v_2(s)) : s \in [0, L]\}$ be two open curves having the same description $\underline{P}$. Now,

$$\begin{aligned} & \int_{jL/n}^{(j+1)L/n} \{(u_1(s) - u_2(s))^2 + (v_1(s) - v_2(s))^2\} ds \\ & \leq \int_{jL/n}^{(j+1)L/n} (4L^2/n^2) ds = 4L^3/n^3 \end{aligned}$$

because the two points $(u_1(s), v_1(s))$ and $(u_2(s), v_2(s))$ lie within a circle of radius L/n with centre at P_j .

So,
$$\begin{aligned} & \int_0^L \{(u_1(s) - u_2(s))^2 + (v_1(s) - v_2(s))^2\} ds \\ & \leq n \cdot 4L^3/n^3 = 4L^3/n^2 \end{aligned}$$

Thus, open curves giving rise to the same description $\underline{P}$ differ at most by $4L^3/n^2$ which goes to zero as n increases. And the distance within the class being zero means the class is a singleton.

Appendix B.9 To find the directional code (belonging to S_8) of the line $\overrightarrow{PQ}$ for any two points P and Q in the plane.

Suppose, the co-ordinates of P and Q are (x_1, y_1) and (x_2, y_2) respectively. We will represent the output directional code by d.

If $(x_2 - x_1)$ is zero, then $d=4$ if $(y_2 - y_1)$ is negative and $d=8$ if $(y_2 - y_1)$ is positive. If $(x_2 - x_1)$ is non-zero, calculate

$\theta = \tan^{-1} \left\{ (y_2 - y_1) / (x_2 - x_1) \right\}$. Set $d=2-4\theta/\pi$ if $(x_2 - x_1)$ is positive and $d=6-4\theta/\pi$ if $(x_2 - x_1)$ is negative.

Appendix C

Appendix C.1 Suppose $A \neq B$. So, either $A-B$ or $B-A$ is non-empty. Wolog, let us assume that $A-B$ is non-empty. $A-B = A \cap B^c$ and $\text{Int}(A)-B = \text{Int}(A) \cap B^c$. Now, if E and F are disjoint open sets, then $\text{Cl}(E)$ and F are disjoint. So, $A-B$ is non-empty implies $\text{Int}(A)-B$ is non-empty, since $A = \text{Cl}(\text{Int}(A))$. But since $\text{Int}(A)-B$ is an open set, $\text{Area}(\text{Int}(A)-B) > 0$. So, $\text{Area}(A-B) > 0$. Thus $D_g(A, B) > 0$.

Appendix D

Appendix D.1 Since $4 > w_1(\lambda), w_2(\lambda) > 0$, $|f_i(\lambda)| < \infty$ and $e_i(\lambda) = 1$ for $i=1,2$. Now, $f_2(\lambda) = -f_1(\lambda)$ implies $v_2'(\lambda/2)/u_2'(\lambda/2) = -v_1'(\lambda/2)/u_1'(\lambda/2)$ since $L \neq 1/2$. Again, $(u_1'(\lambda/2))^2 + (v_1'(\lambda/2))^2 = 1$ for $i=1,2$. So, $e_i(\lambda) = 1$ implies $u_1'(\lambda/2) = u_2'(\lambda/2)$ and $v_1'(\lambda/2) = -v_2'(\lambda/2)$. This is true for all but finitely many λ , since the curve is piecewise smooth.

Appendix D.2 Since the curve is piecewise smooth, $u'(\lambda L)$ and $v'(\lambda L)$ exist and are continuous for all except finitely many λ . Suppose, $(u(\lambda L), v(\lambda L))$ is not a corner point.

If $u'(\lambda L) \neq 0$, LHS = RHS in Definition 4.3.1.

$$\begin{aligned} \text{Now, } u'(L-\lambda L) &= \lim_{h \downarrow 0} \frac{u(L-\lambda L) - u(L-\lambda L - h)}{h} \\ &= \lim_{h \downarrow 0} \frac{u(\lambda L + h) - u(\lambda L)}{h} \\ &= u'(\lambda L) \end{aligned}$$

Similarly, $v'(L-\lambda L) = v'(\lambda L)$.

$$\text{So, } f(1-\lambda) = \frac{v'(L-\lambda L)}{u'(L-\lambda L)} = \frac{v'(\lambda L)}{u'(\lambda L)} = f(\lambda)$$

$$\text{Now, } e(1-\lambda) = 1 \iff \delta(\lambda) = -1 \iff e(\lambda) = 1$$

$$\text{Similarly, } e(1-\lambda) = -1 \iff \delta(\lambda) = 1 \iff e(1-\lambda) = -1$$

$$\text{Thus, } w(\lambda) = w(1-\lambda)$$

Suppose, $u'(\lambda L) = 0$. Then $u'(L-\lambda L) = 0$.

Also, $v'(\lambda L) = v'(L - \lambda L)$. If $v'(\lambda L) > 0$, $f(\lambda) = f(1 - \lambda) = +\infty$ and $w(\lambda) = w(1 - \lambda) = 8$. If $v'(\lambda L) < 0$, $f(\lambda) = f(1 - \lambda) = -\infty$ and $w(\lambda) = w(1 - \lambda) = 4$.

Appendix D.3 Let a region A be symmetric. So, $(x, y) \in A$ if and only if $(x, -y) \in A$. Let $(x, y) \in \text{Int}(A)$. So, there is an open disc D containing (x, y) such that D is contained in $\text{Int}(A)$. Now, by symmetry, $-D = \{(a, -b) : (a, b) \in D\}$ is a subset of A . But $-D$ is an open disc containing $(x, -y)$. So, $(x, -y) \in \text{Int}(A)$. Thus, $(x, y) \in \text{Int}(A)$ if and only if $(x, -y) \in \text{Int}(A)$. So, $(x, y) \in \bar{A}$ if and only if $(x, -y) \in \bar{A}$. Hence $\bar{A}$ is symmetric.

Conversely, suppose $\bar{A}$ is symmetric. Suppose, A is not symmetric. So, there exists (x, y) in $\text{Int}(A)$ such that $(x, -y)$ does not belong to $\text{Int}(A)$. In fact, $(x, -y) \in A^c$. Since both $\text{Int}(A)$ and A^c are open sets, there exists an open set D containing (x, y) such that D is contained in $\text{Int}(A)$ and $-D$ is contained in A^c . Note that $(x, -y) \in -D$. Let E be the largest open set satisfying (i) $(x, y) \in E$, (ii) E is a subset of $\text{Int}(A)$ and (iii) $-E$ is a subset of A^c . Let $F = -E$.

We will show that $A^c = F \cup (A^c - F)$ is disconnected. $A^c - F$ is a subset of F^c . So, the closure of $A^c - F$ is a subset of F^c , and hence, cannot have intersection with F . Now, to show that the closure of F does not have any intersection with $A^c - F$. Suppose, (a, b) is a

limit point of F such that $(a,b) \in A^c$ and $(a,b) \notin F$. Note that $(a,-b)$ is a limit point of E and $(a,-b) \notin E$. Now $(a,-b)$ can not belong to $\bar{A}$ since (a,b) does not belong to $\bar{A}$. So, $(a,-b) \in \text{Int}(A)$. Hence, there is an open set D contained in $\text{Int}(A)$ such that $(a,-b) \in D$ and $-D$ is a subset of A^c . EUD satisfies (i), (ii) and (iii) above. It is a contradiction to the fact that E is the largest open set satisfying (i), (ii) and (iii).

It is clear that F is non-empty. $A^c - F$ is also non-empty because F is bounded. Thus A^c is disconnected. It is a contradiction to the fact that A is a region. Hence A is symmetric.

REFERENCES

- Apostol, T.M., Mathematical Analysis, Addison-Wesley, Reading, Massachusetts, 1979.
- Arcelli, C., 'Pattern thinning by contour tracing', Comput. Graph. Image Proc., Vol.17, 1981, pp.130-144.
- Ballard, D.H. and Brown, C.M., Computer Vision, Prentice-Hall, Englewood Cliffs, N.J., 1982.
- Bennet, J.R. and McDonald, J.S., 'On the measurement of curvature in a quantized environment', IEEE Trans. Comput., Vol.C-24, 1975, pp.803-820.
- Blum, H., 'A transformation for extracting new descriptions of shape', in Symposium on Models for the Perception of Speech and Visual Form, M.I.T. Press, 1964.
- Blum, H., 'Biological shape and visual science', Int.J.Theor. Biol., Vol.38, 1973, pp.205-287.
- de Boor, C., A Practical Guide to Splines, Springer-Verlag, New York, 1978.
- Bribiesca, E. and Guzman, A., 'How to describe pure form and how to measure differences in shapes using shape numbers', Pattern Recognition, Vol.12, No.2, 1980, pp.101-112.
- Chaudhuri, B.B., 'A simple method of thinning', Jour. Instn. Electron. and Telecom. Engrs., Vol.24, 1978, pp.264-265.
- Chaudhuri, B.B. and Dutta Majumdar, D., 'Recognition and fuzzy description of sides and symmetries of figures by computer', Int.J.Systems Sci., Vol.11, 1980, pp.1435-1445.

Davis, L.S., 'Understanding shape : angles and sides', IEEE Trans. Comput. Vol.C-26, 1977A, pp.236-242.

Davis, L.S., 'Understanding shape : II. symmetry', IEEE Trans. Vol.SMC-7, 1977B, pp.204-212.

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

Dudani, S.A., Breeding, K.J. and McGhee, R.B., 'Aircraft identification by moment invariants', IEEE Trans. Comput., Vol.C-26, 1977, pp.39-46.

Freeman, H., 'On the encoding of arbitrary geometric configurations', IEEE Trans. Elec. Comput., Vol.EC-10, 1961, pp.260-268.

Freeman, H., 'Boundary encoding and processing', in Picture Processing and Psychopictories, ed. by Lipkin and Rosenfeld, Academic Press, New York, 1970, pp.241-266.

Freeman, H., 'Computer processing of line drawing images', Computing Surveys, Vol.6, 1974, pp.57-97.

Freeman, H. and Davis, L.S., 'A corner finding algorithm for chain coded curves', IEEE Trans. Comput. Vol.C-26, 1977, pp.297-303.

Freeman, H. and Saghri, A., 'Generalized chain codes for planar curves', in Proc. 4th ICPR, Kyoto, Japan, 1978, pp.701-703.

Fu, K.S., Ed., Pattern Recognition and Machine Learning, Plenum, New York, 1971.

Fu, K.S. and Swain, P.H., 'On syntactic pattern recognition', in Software Engineering, Vol.2, J.T. Tou, ed., Academic Press, New York, 1971, pp.155-182.

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

- Fu, K.S. and Rosenfeld, A., 'Pattern recognition and image processing', IEEE Trans. Comput., Vol.C-25, 1976, pp. 1336-1346.
- Fu, K.S., Ed., Digital Pattern Recognition, Springer-Verlag, New York, 1976.
- Fu, K.S., Ed., Syntactic Pattern Recognition, Applications, Springer-Verlag, New York, 1977.
- Fu, K.S., Syntactic Pattern Recognition and Applications, Prentice-Hall, Englewood Cliffs, N.J., 1982.
- Granlund, G.H., 'Fourier preprocessing for handprint character recognition', IEEE Trans. Comput., Vol.C-21, 1972, pp.195-201.
- Hilditch, C.J., 'Linear skeletons from square cupboard', Machine Intelligence, Vol.4, 1969, pp.403-420.
- Hu, M.K., 'Visual pattern recognition by moment invariants', IRE Trans. Inf. Theory, Vol.IT-8, 1962, pp.179-187.
- Kashyap, R.L. and Oommen, B.J., 'A geometrical approach to polygonal dissimilarity and shape matching', IEEE Trans. Vol.PAMI-4, 1982, pp. 649-654.
- Ledley, R.S., 'High speed automatic analysis of biomedical pictures', Science, Vol.146, 1964, pp.216-223.
- Lipkin, B.S. and Rosenfeld, A. [eds.], Picture Processing and Psychopictories, Academic Press, New York, 1970.
- Montanari, U., 'Continuous skeletons from digitized images', Journal ACM, Vol.16, 1969, pp.534-549.
- Mott-Smith, J.C., 'Medial axis transformations', in Picture Processing and Psychopictories, ed. by Lipkin and Rosenfeld, Academic Press, New York, 1970, pp.267-283.

- Parui, S.K., Chaudhuri, B.B. and Dutta Majumder, D., 'Some algorithms on preprocessing of two-tone pictures for image analysis', Journal of Inst. of Elect. and Telecom. Engg., Vol.26, 1980, pp.21-28.
- Parui, S.K., Chaudhuri, B.B. and Dutta Majumder, D., 'A procedure for recognition of connected handwritten numerals', Int. J. Systems Science, Vol.13, 1982A, pp.1019-1029.
- Parui, S.K. and Dutta Majumder, D., 'A new definition/^{of} shape similarity', Pattern Recognition Letters, Vol.1, 1982B, pp.37-42.
- Parui, S.K. and Dutta Majumder, D., 'Symmetry analysis by computer', Pattern Recognition, Vol.16, 1983A, pp.63-67.
- Parui, S.K., Sarma, S.E. and Dutta Majumder, D., 'Shape analysis by computer', Proc. Annual Convention of Computer Society of India, 1983B, pp.A3.02.01-A3.02.06.
- Parui, S.K. and Dutta Majumder, D., 'Shape similarity measures for open curves', Pattern Recognition Letters, Vol.1, 1983C, pp.129-134.
- Parui, S.K. and Dutta Majumder, D., 'Two methods for quantification of shape distance between open curves', Proc. IEEE SMC Int. Conf., Bombay-Delhi, 1983D, pp.581-585.
- Parui, S.K., 'Some studies in shape analysis of 2-dimensional regions', Indian Statistical Institute Technical Report, ECSU/TR/02/1983, 1983E.
- Parui, S.K. and Dutta Majumder, D., 'How to quantify shape distance for 2-dimensional regions', Proc. 7th Int. Conf. Pattern Recognition, Montreal, 1984, pp. 72-74.

- Pavlidis, T., 'Analysis of set patterns', Pattern Recognition, Vol.1, 1968, pp.165-178.
- Pavlidis, T., 'Representations of figures by labelled graphs', Pattern Recognition, Vol.4, 1972, pp.5-17.
- Pavlidis, T. and Horowitz, S.L., 'Segmentation of plane curves', IEEE Trans. Comput., Vol.C-23, 1974, pp.860-870.
- Pavlidis, T., Structural Pattern Recognition, Springer-Verlag, Berlin, 1977.
- Pavlidis, T., 'Polygonal approximations by Newton's method', IEEE Trans. Comput., Vol.C-26, 1977, pp.800-807.
- Pavlidis, T., 'A review of algorithms for shape analysis', Comput.Graph. Image Proc., Vol.7, 1978, pp.243-258.
- Pavlidis, T., 'Algorithms for shape analysis of contours and waveforms', IEEE Trans. Vol.PAMI-2, 1980, pp.301-312.
- Pavlidis, T., 'Shape analysis and digitization', in Pattern Recognition in Practice, ed. by E.S. Gelsema and L.N. Kanal, North-Holland, Amsterdam, 1980, pp.123-129.
- Pavlidis, T., 'A thinning algorithm for discrete binary images', Comput. Graph. Image Proc., Vol.13, 1980, pp.142-157.
- Pavlidis, T., 'An asynchronous thinning algorithm', Comput. Graph. Image Proc., Vol.23, 1982, pp.133-157.
- Pavlidis, T., Algorithms for Graphics and Image Processing, Springer-Verlag, New York, 1982.
- Persoon, E. and Fu, K.S., 'Shape discrimination using Fourier descriptors', IEEE Trans. Vol.SMC-7, 1977, pp.170-179.
- Pfaltz, J.L. and Rosenfeld, A., 'Computer representation of planar regions by their skeletons', Communication ACM, Vol.10, 1967, pp.119-125.

- Pferd, W. and Ramachandran, K., 'Computer aided automatic digitizing of engineering drawings', Proc. IEEE COMPSAC, 1978, pp.630-635.
- Ranade, S., Rosenfeld, A. and Samet, H., 'Shape approximation using quadtrees', Pattern Recognition, Vol.15, 1982, pp.31-40.
- Rosenfeld, A. and Pfaltz, J.L., 'Sequential operations in digital picture processing', Journal ACM, Vol.13, 1966, pp.471-494.
- Rosenfeld, A. and Pfaltz, J.L., 'Distance functions on digital pictures', Pattern Recognition, Vol.1, 1968, pp.33-62.
- Rosenfeld, A., Picture Processing by Computer, Academic Press, New York, 1969.
- Rosenfeld, A. and Johnston, E.G., 'Geometrical operations on digitized pictures', in Picture Processing and Psychopictories [B.S. Lipkin and A. Rosenfeld, eds.], Academic Press, New York, 1970, pp.217-240.
- Rosenfeld, A., 'Connectivity in digital pictures', Journal ACM, Vol.17, 1970, pp.146-160.
- Rosenfeld, A., 'Arcs and curves in digital pictures', Journal ACM, Vol.20, 1973, pp.81-87.
- Rosenfeld, A., 'Adjacency in digital pictures', Information and Control, Vol.26, 1974, pp.24-33.
- Rosenfeld, A., 'A characterization of parallel thinning algorithms', Information and Control, Vol.29, 1975, pp.286-291.
- Rosenfeld, A. and Kak, A.C., Digital Picture Processing, Academic Press, New York, 1976.

- Rosenfeld, A. and Weszka, J.S., 'Picture recognition and scene analysis', Computer, Vol.9, 1976, pp.28-38.
- Rosenfeld, A., 'Digital topology', American Mathematical Monthly, Vol.86, 1979, pp.621-630.
- Rosenfeld, A. and Kak, A.C., Digital Picture Processing [2nd Ed.], Vol.2, Academic Press, New York, 1982.
- Seuffert, P., 'An application of line and character recognition in cartography', Proc. Pattern Recognition Image Processing Conf., 1977, pp.338-343.
- Shapiro, B., Pisa, J. and Sklansky, J., 'Skeletons generation from x,y boundary sequences', Comput. Graph. Image Proc., Vol.15, 1981, pp.136-153.
- Shapiro, L.G., 'A structural model of shape', IEEE Trans. Vol. PAMI-2, 1980, pp.111-126.
- Sklansky, J. and Kibler, D.P., 'A theory of non-uniformly digitizing binary pictures', IEEE Trans. Vol.SMC-6, 1976, pp.637-647.
- Smith, F.W. and Wright, M.H., 'Automatic ship photo interpretation by the method of moments', IEEE Trans. Comput. Vol.C-20, 1971, pp.1089-1094.
- Stefanelli, R. and Rosenfeld, A., 'Some parallel thinning algorithms for digital pictures', Journal ACM, Vol.18, 1971, pp.255-264.
- Tamura, H., 'A comparison of line thinning algorithms from a digital geometry viewpoint', Proc. Fourth Int. Joint. Conf. Pattern Recognition, 1978, pp.715-719.

- Yokoi, S., Toriwaki, J.I. and Fukumura, T., 'Generalized distance transformation of digitized binary images', Proc. Fifth Intern. Conf. Pattern Recognition, 1980, pp.1201-1203.
- You, K.C. and Fu, K.S., 'A syntactic approach to shape recognition using attributed grammars', IEEE Trans. Vol.SMC-9, 1979.
- You, K.C. and Fu, K.S., 'Distorted shape recognition using attributed grammars and error-correcting techniques', Comput. Graph. Image Proc., Vol.13, 1980, pp.1-16.
- Young, I.T., Peverini, R.L., Verbeek, P.W. and Otterloo, P.J., 'A new implementation for the binary and Minkowski operators', Computer Graphics and Image Processing, Vol.17, 1981, pp.189-210.
- Zahn, C.T. and Roskies, R.Z., 'Fourier descriptors for plane closed curves', IEEE Trans. Comput. Vol.C-21, 1972, pp.269-281.
- Zusne, L., Visual Perception of Form, Academic Press, New York, 1970.