A Short Title of the Thesis:

A Method for Measuring the Curvature of Quantised Lines

A METHOD FOR MEASURING THE CURVATURE OF A BOUNDARY ON A 2-DIMENSIONAL QUANTIZED GRID

by

LAWRENCE C.-F. PANG
B.Sc., McGill University, 1974

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

in

The School of Computer Science
McGILL UNIVERSITY
MONTREAL P. Q.

November 1975

ABSTRACT

A survey is given of various methods for extracting curvature and tangent angle measurements from boundary points of quantized shapes. An original method is proposed which uses the entropy of a histogram of the angles of line segments joining the discrete boundary points. This method is implemented experimentally using a computer and its validity and theoretical basis investigated and established. A relationship between the proposed measure and the classical measure of curvature is given. Some variations of the proposed method are suggested and their consequences are investigated. In addition, a method for data reduction of line drawings is proposed and investigated which is based on the entropy measure of curvature.

Index Terms: Curvature, Entropy Measure, Feature Extraction, Picture Processing, Preprocessing, Smoothing, Tangent Angle Measurement, Line Drawings, Data Reduction, Computer Graphics, Pattern Recognition.

Résumé

Une étude des différentes méthodes d'extraction de mesures de courbure et d'angles de tangence à partir des points délimitant les formes quantifiées est ici présentée. Une méthode nouvelle est proposée utilisant l'entropie d'un histogramme des angles formés par les segments de droite joignant les points discrets délimitant le contour. Cette méthode est réalisée experimentalement à l'aide d'un ordinateur et sa validité ainsi que ses fondements théoriques sont ici examinés et établis. Une relation est donnée entre la méthode de mesure proposée et la méthode classique de mesure de courbure. Sont suggerées également certaines variantes de la méthode proposeé et leurs conséquences sont étudiées. Finalement une méthode basée en principe sur la mesure par entropie de la courbure, et ayant pour but la réduction du volume de données requises pour tracés, est proposée et investiguée.

Répertoire: Courbure, Mesure d'Entropie, Extraction des caractéristiques, Traitement des images, Prétraitement, Filtrage, Mesure de l'angle de tangence, Tracés, Réduction des données, Graphiques d'Ordinateur, Reconnaissance des formes.

TABLE OF CONTENTS

List of Illustrations

Acknowledgement

I wish to extend my sincere thanks and appreciation to my thesis supervisor Dr. G. T. Toussaint for his guidance in this work and his assistance in reading and commenting on the initial version of this thesis. I would also like to thank Melvin Cohen for his assistance in the theory of a continuous model in Section 4.3.1 and Selim Akl for providing me with the French version of the abstract.

Finally, I would like to thank Pauline Zakaib for helping to proof-read the thesis and my parents for their love, understanding, encouragement and their earnest prayers.

1. INTRODUCTION

Ever since physiologists and experimental psychologists have conceptualized the picture processing mechanisms of visual systems, this knowledge has instigated the development of many preprocessing techniques and mathematical models to simulate these picture processing procedures and properties. Most of them process pictures that are digitized and quantized in various ways and then applied as a part of preprocessing and feature extraction to some visual pattern recognition system.

1.1 Curvature Measurement

The use of curvature as a feature or shape descriptor in Pattern kecognition arose after Attneave (1) pointed out that "points at which contours change direction maximally" provide the informational content that one would require to recognize shapes. Thus providing a description of a boundary by its curvature values as a function of the points on the boundary is one way by which one can study and try to recognize shapes.

Contours of patterns are used to provide measures of this geometric property. The curvature function may depend on the orientation of the original figure resulting in differences of partial phase shifts. This can be overcome by auto-correlation techniques in the classification procedure. Besides being affected by a variation in size, which can easily be removed by appropriate normalization procedures, the curvature function is literally unaffected by translation. This one dimensional function representing the original two dimensional shape sufficiently enough to make possible the reconstruction of the boundary is deemed a good procedure in data reduction. These attractive properties have made the extraction of curvature measurements from a boundary as a function of distance around the boundary points of its quantized shape a possibly good procedure in image-processing, as a preprocessor in visual pattern recognition systems.

1.2 Mathematical Concept of Curvature

With reference to Fig. 1.1, given a twice differentiable function $f(t)$, we denote $T(t)$ as the unit tangent to the point P at time t, and $\Theta(t)$ as the angle of the tangent at point P at time t. We want to know the rate of change of Θ as t increases, that is, as we move along the curve $f(t)$, as a measure of curvature of the function $f(t)$.

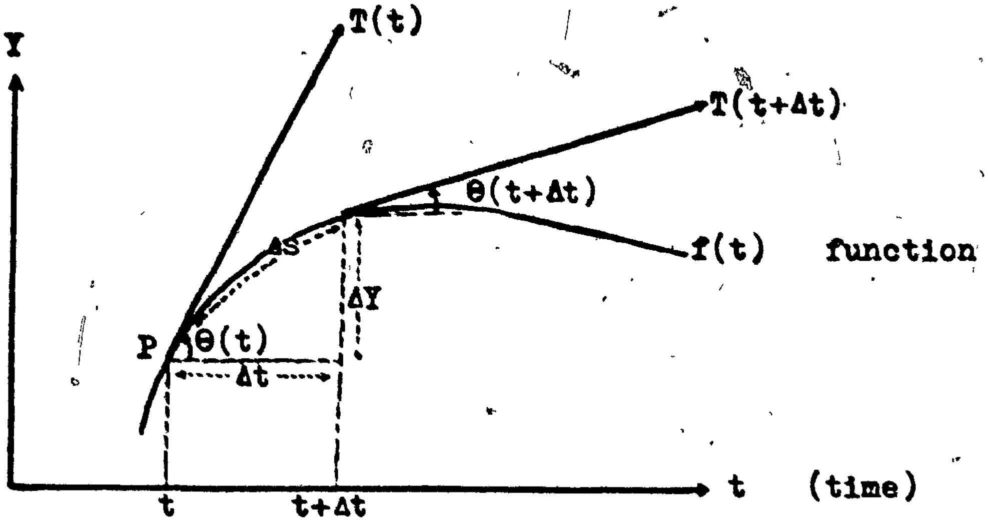


Fig. 1.1: The curvature of the function $f(t)$ is the rate θ changes as a function of t, time.

Formally, the curvature at point P is the value of the derivative of the polar angle (Θ) of the unit tangent vector T with respect to arc length at that point. Intuitively, one can also interpret curvature as the rate of change of arc length per unit change in inclination (Θ), but it is preferable to consider its reciprocal, that is, the rate of change of inclination $\Delta\Theta$ (or turn of the tangent vector) per Δs, the unit change in the arc length along the curve.

So, the curvature at P can be represented by:

$$\mathcal{C} = \frac{d\theta}{ds} = \lim_{\Delta s \to 0} \frac{\Delta\theta}{\Delta s}$$

If we denote the slope tan θ as $y' = \frac{dy}{dt}$, i.e. θ = arctan y', then:

$$\frac{d\theta}{dt} = \frac{y''}{1+(y')^2} \; ; \quad \text{thus, using} \quad \frac{d\theta}{ds} = \frac{d\theta}{dt} * \frac{dt}{ds}$$

where

$$\frac{dt}{ds} = (1+(\frac{dy}{dt})^2)^{\frac{1}{2}} = \sqrt{1-(y')^2}$$ is the reciprocal of the rate of change of arc length per unit change in time t, we get the expression for the curvature:

$$\mathcal{C} = \frac{y''}{\left[1+(y')^2\right]^{\frac{3}{2}}} \qquad (1.2.1)$$

In terms of polar coordinates, we substitute for t=r·cos θ and y=r·sin θ to obtain:

$$\mathcal{C}' = \frac{r^2 + 2(r')^2 - r*r''}{(r^2 + (r')^2)^{\frac{3}{2}}} \qquad (1.2.2)$$

So curvature measures the rate at which the tangent line turns per unit distance moved along the curve, that is, the rate of change of direction of the curve. This calculus notion of curvature will be referred to as the classical definition of curvature.

1.3 Some Shapes and their Curvature Functions

As an example, consider a few simple curves and their curvature functions.

(1) The Circle: its equation: $x^2 + y^2 = r^2$ (1.3.1)

To determine its curvature function, we determine their derivatives y' and y" and substitute them into Eqn. 1.2.1. For $f(x) = \sqrt{r^2 - x^2} = y$, we differentiate to get:

$$y' = \frac{-x}{(r^2 - x^2)^{\frac{1}{2}}} \quad \text{and} \quad y'' = \frac{-r^2}{(r^2 - x^2)^{\frac{3}{2}}}$$

Thus the curvature function:

$$\mathcal{C} = \begin{cases} -\frac{1}{r} & \text{for } y > 0, \text{ i.e. when } y'' < 0 \\ \frac{1}{r} & \text{for } y < 0, \text{ i.e. when } y'' > 0 \end{cases} \qquad (1.3.2)$$

(2) The Ellipse: its equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad (1.3.3)$$

Again, we determine the derivatives and substitute them into Eqn. 1.2.1. In this case, the function is:

$y = \frac{b}{a}\sqrt{a^2 - x^2}$. Thus their derivatives are:

$$y' = -\frac{b}{a} * \frac{x}{(a^2 + x^2)^{\frac{1}{2}}}, \text{ and } y'' = \frac{-a*b}{(a^2 - x^2)^{\frac{3}{2}}}$$

Thus, the curvature function:

$$\mathcal{C} = \frac{-b*a^4}{[a^4 - x^2(a^2-b^2)]^{\frac{3}{2}}} \qquad (1.3.4)$$

(3) The Cardioid: its equation: $r = a(1-\cos\theta)$ (1.3.5)

This time, we determine the polar coordinate derivatives and substitute them into Eqn. 1.2.2. So, we have:

$$r' = a*\sin\theta \quad \text{and} \quad r'' = a*\cos\theta$$

Thus, the curvature function:

$$\mathcal{C} = \frac{3}{2\sqrt{a*r}} \qquad (1.3.6)$$

(4) The Spiral of Archimedes: its equation: $r = a*\theta$ (1.3.7)

Again, we use Eqn. 1.2.2 for:

$$r' = a \quad \text{and} \quad r'' = 0$$

Thus the curvature function:

$$\mathcal{C} = \frac{\theta^2 + 2}{a\,(\theta^2 + 1)^{\frac{3}{2}}} \qquad (1.3.8)$$

(5) The Parabola: its equation: $y = x^2$ (1.3.9)

In the similar steps, we get its curvature function:

$$\mathcal{C} = \frac{2}{(1+4x^2)^{\frac{3}{2}}} \qquad (1.3.10)$$

1.4 Scope of Thesis

A brief survey is given in Chapter 2 on the methods of extracting curvature and tangent angle measurements and their use in obtaining features for visual and pictorial pattern recognition systems. In Chapter 3, a method is proposed to measure curvature and tangent angles using two-dimensionally quantized boundaries of shapes. This method involves using entropy measures as indicators of curvature, producing an entropy value graph plotted against the linear boundary point positions, from which one could easily distinguish higher curvature points from lower curvature points. These methods are implemented using a computer and their validity and dependability investigated and established.

The theoretical structure of this method is explored in Chapter 4 and its basis established by studies and experimental results. A congruous endeavour determines the tangent angle directions of the given boundary on the two-dimensional quantized grid.

The computer implementation provides for a useful tool from which investigative studies on the method is done. Some variations of it are studied in Chapter 5, section 5.1 to 5.3, and their effects determined. Section 5.4 outlines the effects of quantization on the method's performance. Finally, in section 5.5, a method based on using entropy measures as indicators of curvature is suggested for use in data reduction of line drawings. Then conclusions are drawn in Chapter 6 with regards to this method's performance in terms of its time involvement in its computer implementation.

1.5 Some Comments

Throughout this thesis, many terms have been used selectively at my discretion. Terms like "line-points" and "1-step staircases" will be defined when first encountered and subsequently used quite freely when required. Other terms like "noise", "picture retina", "transformation" and "entropy" have been adapted from concepts in picture processing and information theory. Some statistical terms like unbiasness, expected values, means, probability, frequency distribution and histograms have been used with the assumption that readers are familiar with them, and thus they will not be defined.

Some effort has been made to maintain consistency in the symbols used. A list of the main ones used throughout the thesis follows:

θ - the angle of the line (ℓ) between two points (P_1 & P_2) with reference to an arbitrary zero line (σ). In most cases, σ is chosen as the x-axis.

$\emptyset$ - subtending angle of an angle sector of a circle

r - the radius of the circle

ρ - perpendicular distance from the origin to a line

ζ - classical curvature

κ - curvature estimate obtained by taking the entropy of a histogram

ξ - an expression that gives a value indicating curvature

With reference to the programmed system, i.e. the computer implementation of this method, the k^{th} point is the midpoint of the segment under consideration and is the point under consideration. When referring to histograms, the terms "cell" and "slot" are synonymous.

2. Methods of Curvature and Tangent Angle Measurements and their Applications in Pattern Recognition

2.1 Introduction

Since this thesis follows two Ph. D. theses (2,3) related to the measurement of curvature, many references of physiological results and related background in Pattern Recognition involving the use of curvature and tangent angles for various purposes are omitted to reduce the amount of repetition. The intention here is to give a brief survey of some of the more commonly used methods from which the curvature and tangent angle measurements can be derived.

Many researchers have employed these measures to create a description with which to categorize two-dimensional binary shapes or silhouettes. Of all these works, only a few methods have been repetitiously used, although variations of them are usually adapted for each particular problem. Inadvertently, these curvature or tangent angle measures have been easy to derive and have proven to be useful as representatives of shapes and figures. Most often, they are extracted from digitized or quantized versions of the original shape or pattern. Methods involving digitized or quantized pictures with more than two gray levels is not considered in this thesis, (e.g. Symon (4)).

2.2 Representation of Curves

Most often one can easily provide adjectival descriptors of shapes like compact, jagged, complex, beautiful, etc. One can also correlate physical measures like perceivable complexity with more geometrical terms like the number of sides, angular variability, symmetry or lack of symmetry, perimeter^2/area, etc. For example, Kazmierczak (5) used some descriptive terms for line forms that produced shape criteria characteristics when using the potential distribution to characterize the shape. Here, he used terms like peaks, exposed marginal points, protected points, curve shape open to the left or to the right, straight line, closed lines, continued lines, etc. Also terms like convex, concave, straight, etc have been sufficient for some purposes (6). On the other hand, picture subsets can be completely determined by specifying their edges as a set of directed curves which could be represented by their equations.

2.2.1 Using Chains of Coded Directions

Attneave (1) established through some experiments that information of shapes was concentrated at points of higher contrast and where the contour changes direction most rapidly, the relative locations of the points and their connectivity. These points on the contour, having maximum curvature, were important for the representation and recognition of the shapes.

Attneave & Arnoult (7) approximated a curve by choosing sample points so that the difference between the original and the interpolated curves does not exceed a pre-specified amount. Alternately, a pre-specified number of sample points can be selected to minimize the difference.

Freeman (8) used the grid-intersection method to approximate a curve. The set of points approximating the line are those grid-intersection points for which the curve passes the closest or through some pre-specified neighbourhood about that point. In the polygonal approximation obtained in this way, each side of the approximated curve is either horizontal, vertical of unit length, or diagonal with length of the ratio of $\sqrt{2}$ to the unit length. This is the basis of chain encoding from which one can represent lines by their starting point and the sequence of slopes of its sides. These slopes, the tangent angle approximations, are usually coded into six (hexagonal grid representation) or eight numbered directions to produce a chain of coded directions which most often can be used as an efficient description with some invariance properties for the purpose of classification.

Others have done this by selecting breakpoints for which the edges or the borders of a picture subset may be segmented to produce polygonal approximations of the original contours. For instance, Freeman (9) chose points of inflections. Mason and Clemens (10) used local extremas of the contours; i.e. the locally lowest, highest, left-most, right-most points of the edge to generate a simple code for classification of alpha-numerics. Genchi, et al (11) segmented edges by locating shape features like straight strokes, bays, notches, spurs, etc, on the contour in seven stroke segment directions and used this sequence of geometrical features of the character for classification. Glucksman (12) and Symon (4) both chose points on the curve that have high curvature values. Guzman (13) chose intersections for his own purpose; where two or more lines with different slopes meet at one point, end points and isolated points. Sherman (14) chose nodes; where two or more arcs end or meet in their line-like figures. This breaks the contour into "branches" joined to one another at arc ends. A connection matrix is then used to describe the group joining these nodes and contains the sign, maximum degree and the amount of curvature of the branches joining the node.

2.2.2 As Functions of Picture Subsets

An important method of describing the edge of a picture subset is by its intrinsic equation which gives its curvature as a function of arc length measured from an arbitrary starting point. This method completely determines the edge and can be made independent of size and origin.

Alternatively, as can be observed in Fig. 2.1, one can use the slope of the edges (tangent angle estimates) rather than the curvature measure, as a function of arc length, although this would not be independent of orientation, or as in Fig. 2.2, take the angle between the radius vector, i.e. the line from the origin to a point on the edge, and the tangent to this edge at that point, as a function of arc length. This is independent of position, orientation and scale (15). Still another way, as in Fig. 2.3, is to take the radial angle as a function of the arc length (16).

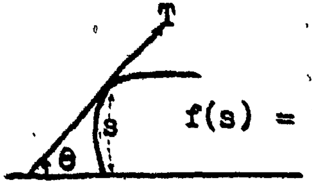


Fig. 2.1

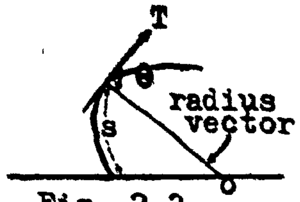


Fig. 2.2

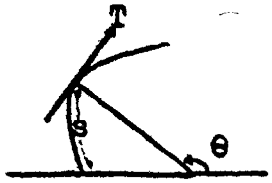


Fig. 2.3

The chain encoding of an arc, as a sequence of steps of pre-determined length in one of a given set of directions, can be thought of as a discrete intrinsic equation since it gives quantized slopes (tangent angle measurements) as a function of quantized arc length. The quantized curvature measure, the rate of change of the slope, as a function of arc length is then obtained by taking the differences of successive slopes (17-19). Cantoni (20) used this approach to obtain optimal curve fitting using a linear piecewise function.

2.2.3 Other Methods of Representing Curves

Hough (21) proposed an interesting and computationally quite efficient procedure to represent curves and eventually detect lines, even dotted lines in pictures. In cases of line-like figures, sequential methods adapted from his point to curve (slope-intercept) transformation scheme can be used to search for and follow curves to provide information about its shape; for instance, to obtain the curvature as a function of arc length.

Duda & Hart (22) showed that the use of the angle-radius rather than the slope-intercept parameters of Hough to represent curves would simplify computations even further. Their method is used for more general curve fitting and has been improved to detect lines in noisy pictures by Cohen & Toussaint (23).

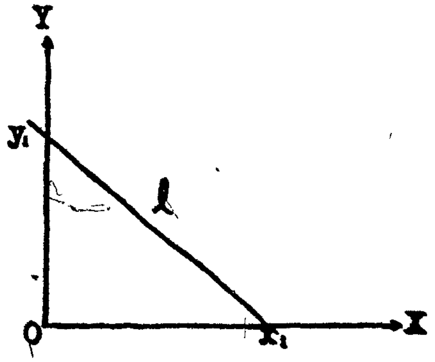


Fig. 2.4: Hough Transformation.
The line ℓ is represented by the parameters:

$$\left(\frac{y_1}{x_1}, y_1\right)$$

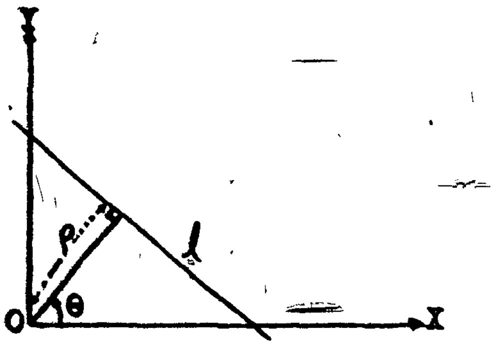


Fig. 2.5: Duda & Hart's Transformation
The line ℓ is represented by the parameters:

$$(\theta, \rho)$$

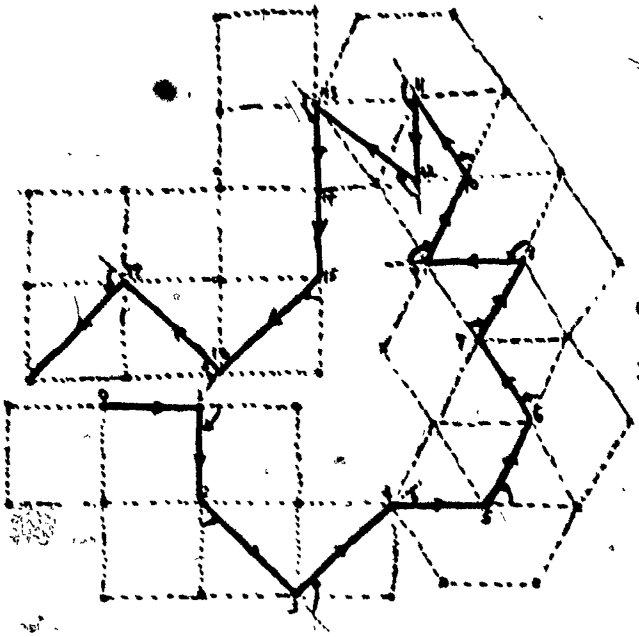


Fig. 2.6
An Arbitrary Shape quantized on:
i. Square Grid: •
ii. Hexagonal Grid: ×
and traversed in an anti-clockwise direction.

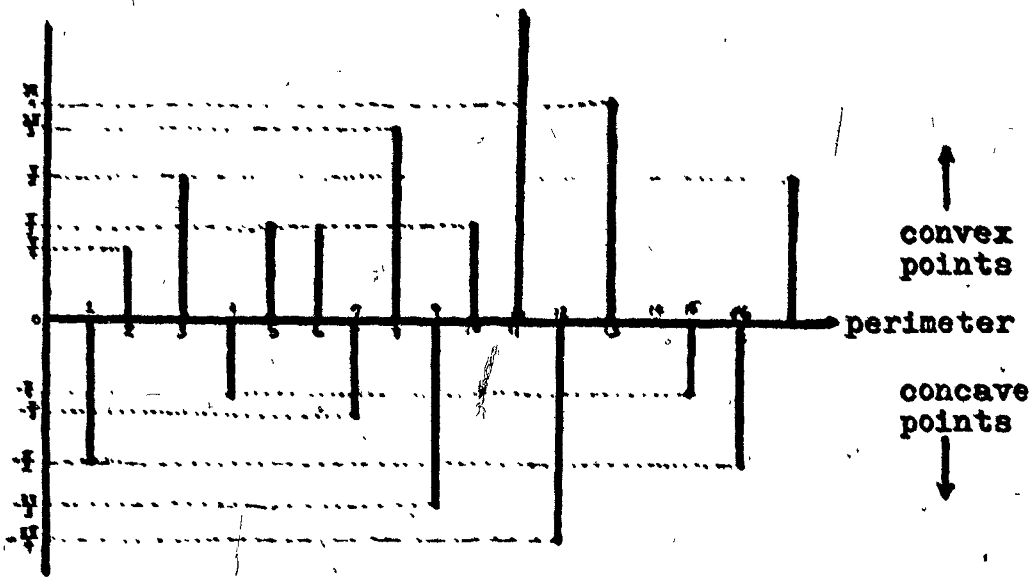


Fig. 2.7: The Curvature Graph.

Curvature values are expressed as the difference in the angles (in radians) of the trailing vector and the leading vector. For instance, trailing vector $\overline{01}$, leading vector $\overline{12}$ gives a clockwise difference in angle of $\pi/2$, thus establishes the value as negative — for a concave point.

2.3 Some Classes of Techniques for Measuring Curvature and Tangent Angles

We have already seen some ways of representing digitized or quantized lines and figures. In this section, we class some more commonly used ideas from which one could derive or extract measures of curvature and tangent angles. It should be kept in mind that some of these are not direct procedures from which one can obtain these measures; instead, they are means used to facilitate the extraction of the curvature and tangent angle measurements.

2.3.1 By the definition of an Imrulse of Curvature

The measure of curvature at a point of a linearly segmented arbitrary shape is literally taken as the difference in angular directions of the vector leading from the previous point and the vector leading to the next point, the figure being traversed in a pre-specified direction from which arbitrary standard notations may be set. For instance, if the shape is being traversed in an anti-clockwise direction, we can set clockwise difference in angle as being of negative curvature and counter-clockwise difference as being of positive curvature. Thus concave segments will have negative curvature values and convex segments will have positive values of curvature.

To illustrate this method, we quantized an arbitrary shape (Fig. 2.6) partly on square grids and partly on hexagonal grids. Distances between points are accounted for by different distances on the x-axis of the curvature graph, shown in Fig. 2.7. Possible values obtained from shapes quantized on square grids are: $\pm\pi$, $\pm 3\pi/4$, $\pm\pi/2$, $\pm\pi/4$ and 0. Similarly, possible values obtained from shapes quantized on hexagonal grids are: $\pm\pi$, $\pm 2\pi/3$, $\pm\pi/3$, and 0. (24)

2.3.2 Via Chain-Encoding

Here, a slope function is quantized into a set of eight possible standard slopes, or six, to produce representations based on a hexagonal grid. The curve is represented by a sequence of small vectors in a limited set of possible directions which describe the features of the shape by the associated chains and chainlets of coding used. It was Freeman (8,18,25) who first introduced the chain-encoding method that used the eight basic directions to quantize the directions along a curve traced out by a line follower. This technique has been adapted by many others (26-31) for use in the recognition of handwritten characters and numerals. Some defined chainlets as portions of larger chains that represent curves between slope discontinuities or inflection points, and describe the shapes of the pieces by features associated with the properties of the associated chains and chainlets.

In a similar way, Groner (28) quantized his curve using only the four main directions. Still others (32,33) used these principal techniques on their work to detect overlapping chromosomes, cells and other biological objects.

Essentially, curve following techniques require well-defined contrast boundaries which are usually obtained after some form of preprocessing if they are not initially present in the original figure. Kasvand (34) uses a simple technique that does not require a curve follower but it does not account for the effects of imperfect data or the presence of noise. He suggests quantizing a contour or edge using a hexagonal array as the basis of his operations on monochromatic two-dimensional objects and using curvature as a descriptor.

With reference to Fig. 2.8, if $a_{01} = 1$, the assembly of points is on the contour, and the expression:

$C = a_{01} + \sum_{i=1}^{6} a_{1i} + \sum_{i=1}^{12} a_{2i}$ is directly, although non-linearly related to the angle of curvature. This is then used as a basis for his coding.

Yamada & Fornanago(35) used another method where they quantized and labelled boundary points in a

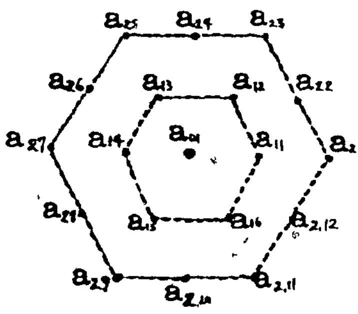


Fig. 2.8: Hexagonal grid assembly used by Kasvand

scene according to a scale of eight directions of a line through the point in question using a 5 by 5 quantized grid. This method also does not require the use of a curve follower.

2.3.3 Via the Angle versus Length (AVL) Technique

Attneave and Arnoult (7) outlined a few methods to describe contours so that they would be independent of their size, place and orientation. Generally, normalization of the boundary would render it independent of size while location and orientation are intrinsically dormant in these techniques.

(a) Plot the reciprocal of the radius of curvature versus the distance along the contour at each point. This gives a periodic function (of 2π if the contour is closed and convex) that can easily be made independent of the scale of the original shape. For instance, the function can be normalized by scaling the perimeter of the figure as one unit and expressing (r) the radius of curvature in comparable terms, or alternately, set the area under one period of the function to one unit.

(b) Plot the angle deviation (θ) versus the distance along the contour. This can be realized as in Fig. 2.9, by guiding a tricycle over the contour such that the point between the rear wheels follows the contour line. The angle deviation θ, the angle measured between the front wheel and the forward direction is then plotted against the distance travelled by the front wheel to give a periodic function descriptive of the contour. Since the front wheel moves in an arc concentric with the segment of the contour being followed, the function maintains a range of values between -90° and 90°. The radius of curvature is given by: $r = \cot \theta$, where L is the distance between the front and rear wheels. This function could also be normalized by giving the perimeter of the figure unit value and setting L at some small fraction; the smaller L is, the

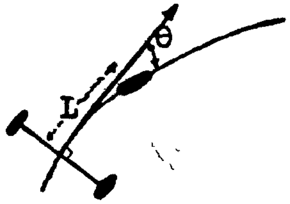


Fig. 2.9

more "continuous" would be the resulting function.

(c) We can divide the figure into individually homogeneous parts which are amenable to approximate descriptions in terms of a few standard dimensions.

Given a "complex" figure, we construct a polygon about it by drawing tangents

(i) at points of zero curvature, i.e. at straight segments of the boundary and at inflection points--when the curve changes from concave to convex or vice versa,

(ii) at points of minimal curvature, when decrease in curvature is followed by an increase, or vice versa,

(iii) at discontinuities of slopes or angles.

This makes it possible to describe the figure by stating successive sets of slopes and the length of each line in the constructed figure. It may be made orientation-free and scale-free by specifying instead, for each pair of adjacent segments,

(i) the change in direction in degrees or radians,

(ii) the change in length (in logarithms), as the contour is followed around in a clockwise direction,

which is what is essentially required for the description of shapes of successive segments of the polygon taken in pairs.

In this case, as can be seen in Fig. 2.10, a curve is approximated by an arc located tangent to two successive lines of the polygon, like a rounded-off angle. The size of the arc is limited by the shorter of the two segments. Hence the curvature can be expressed as the ratio s_1/s_2. Values of this range from zero, indicating an abrupt angle with the radius of curvature = 0, to one, indicating an arc which is tangent to the shorter segment at its end. It may also attain values greater than one for some series of bulbous projections that one might introduce or find in a figure.

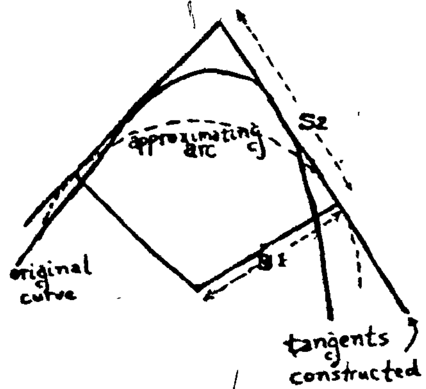


Fig. 2.10

This method provides descriptions that are invariant of size or orientation. Since general normalizing factors are avoided, part similarities of two objects are reflected in their numerical descriptions between same contours and repetitious sequences of elements, so, auto-correlation techniques could be used.

The number of terms required to adequately describe a curve is related to the complexity of the figure. Fehrer (36) showed that it is proportional to the difficulty in a reproduction-learning situation, while Attneave (7) showed that it is proportional to the number of sides. The major disadvantage of this method is that some figures like spirals do not yield unique descriptions since we approximate curves by straight lines and arcs and ignore higher order invariances.

(d) Plot the tangent angle versus the arc length. Starting at an arbitrary point and tracing the contour of the shape clockwise, the sum of the angular changes of the directed tangent to the boundary is plotted against the length along the boundary. If we let clockwise angular changes be considered negative and counter-clockwise, positive from the position of the initial tangent, the resulting wave-form would be a single valued continuous function of the boundary length. This method wasfirst used by Brill (37) and many others in the University of Ohio group, (see 3,24) and Zahn (38).

2.3.4 <u>By Syntax-Directed Methods</u>

Ledley and Rotolo (39) used a syntax-directed pattern recognition technique using the programming system FIDACSYS to classify objects by matching characteristic shapes and other aspects of the object with a syntax description. The boundary is characterized in terms of successive segments, each described by its direction and curvature.

Here, curvature is approximated as the angle between the leading (L) and trailing vector (T) that divides the arc length equally, see Fig. 2.11. The tangent angle is the sum of the two vectors.

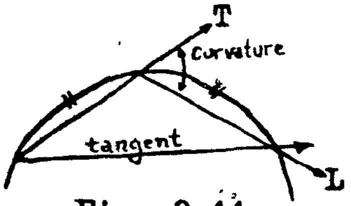


Fig. 2.11

Ledley (40) used this same method to obtain boundary curvature as a function of arc length for automatic chromosome analysis and classification of blood cells and other biological shapes. The karyotyping was performed by syntax-directed boundary analysis from which these measurements were made for the purpose of classification. A related technique was used by Ledley, et al (41) while using film as input to a digital automatic computer programmed to use an associated syntax-directed pattern recognition system. This same method of evaluating curvature and tangent angle was also used by Shirai (42) in the scene analysis of polyhedral objects and by Johnston & Rosenfeld (43).

The analysis of local curvature of the chromosome boundary pioneered by Ledley (44,45) has since been refined by many others. One of them, Gallus & Neurath (46) introduced a simplified form of curvature analysis which aims specifically at locating regions of negative curvature which indicates the end of the centromere line. This method includes analysis of the symmetry of the shape in relation to the various prospective centromere lines joining pairs of opposing concavities.

2.3.5 By Fourier Analysis

The Fourier Descriptors of Brill (37,47) are simply related to the Fourier components of the curvature function. The boundary is traced to produce a characteristic wave-form by plotting the tangent angle versus the arc length around the figure. This unique one-dimensional wave-form representing the exterior boundary of the shape is expanded into the Fourier series from which the coefficients of the series are used as descriptors of the character. This method is the first successful attempt to describe a character's boundary by its Fourier coefficients which are then used for their classification.

Rutovitz (16) described chromosome shapes as closed curve outlines in polar co-ordinates by a Fourier analysis of r, the radius, as a function of the polar angle, θ. In taking the origin of the coordinate system as the centroid of the enclosed area, the results would depend on the choice of the origin.

Moreover, this technique is restricted to single valued curves for any given value of Θ.

Bennett (3) extensively used and investigated the use of curvature and tangent angles as a function of arc length, in terms of distance around the boundary, and its resulting expansion of the periodic function in a Fourier series for shape recognition.

2.3.6 Via Lateral Inhibition and the Area Operator

The use of lateral inhibition models to obtain a form related to curvature has been extensively studied by Connor (2). It was used by Kasvand (48-50) to extract and smooth the curvature of boundary points, and using the concept of area operators, extract curvature values at isolated points in investigating the use of curvature in nerve endings and water droplet countings.

The concept of area operators stems from an enormous background of psychological and physiological results and mathematical models built on the basis of lateral inhibition networks. The area operator simulates the activity in a lateral inhibitory network and produces values when operated along an intensity boundary to give a description of the boundary related to its intrinsic description -- curvature as a function of arc length. These concepts and techniques were dealt with extensively by Connor (2) and Bennett (3) in their Ph. D. theses.

2.4 Some Feature Extraction Schemes

Some examples of tangent and curvature measure extraction procedures should provide a feeling of how the above methods are utilized for the purpose of feature extraction.

Kazmierczak (5) used two-dimensional fields of flow in which to represent characters and then extracted shape descriptors classed as gross convexity open towards the left or right. Guzman (13) pointed out that two curves have the same shape if the function that gives the radius of curvature as a function of s', the normalized distance along the curve, is the same for both. Although this function of s' is independent of size, position and orientation of the curve, visualizing the curves described by it is rather difficult. So, he chose intersections -- where two or more lines with different slopes meet at one point, end-points of lines and isolated points. Then he determined the main axis of the curve, the tangent angle direction, by finding the maximum total contribution of this curve in each of the 64 directional axes used. The function of the curve is obtained by coding the direction using two terms: the slope code, i.e. the tangent angle direction code, and the midpoint of the curve. With this, the curve was normalized to make the function independent of curve sise and its origin. A criterion used for two curves to have the same shape is that they must have the same total contribution at the same angles.

Barrow & Popplestone (51) represented curves in the s-θ coordinate, (see Fig. 2.1), where θ was the angle that the tangent to the curve makes with a reference line, usually the x-axis, and s is the length along the curve from an arbitrary starting point. This function $\theta(s)$ is single-valued and is suitable for curves of arbitrary complexity. For a closed curve it is periodic with a period of 2π. If we consider $\phi(s) = \theta(s)-2\pi s/s_o$, where s_o is the length of the perimeter, we subtract the steadily rising component as s increases. This function is also single-valued and cyclic, since $\phi(0+s)=\phi(ns+s)$ for any integer $n \geqslant 1$, thus, well-suited for Fourier analysis.

For example, a circle has constant $d\theta/ds = 2\pi/s_o = 1/r$, so $\theta(s)$ is a straight line of slope $2\pi/s_o$ and $\phi(s)$ is also constant. Thus an arc of a circle transforms to a straight line on the s-ϕ or s-θ plot. The magnitude and direction of the slope gives the radius and sense of the arc. Straight lines also transform to a straight line on the s-ϕ or s-θ plot. Then the average curvature measure is computed by fitting the best straight line to the $\theta(s)$ representation of the boundary curve and normalizing it by dividing it by $2\pi/s_o$, the average curvature of the whole boundary.

3. The Proposed Method

3.1 Introduction

This method came about initially through intuitive speculations on how a higher proportion of all possible lines that can be drawn through pairs of points on a segment of a digitized curve seem to give us the general or axial direction of the segment in question. Then if one were to count the number of lines in each direction to give a histogram, the entropy value of this histogram would reflect the curvature of the curve.

Ideally, a preprocessed and eight-way connected* digitized boundary of a shape is used. For each boundary point, we consider a string of connected points on each side. In determining a good size for the neighbourhood, heuristic approximations and considerations of the shape in question are employed to obtain a value that will give the most realistic set of curvature and tangent angle measurements.

The necessity of the boundary points being all connected is rather arbitrary since the importance of this method lies in working with a linear list of boundary point coordinates. If a shape or pattern happens to have more than one boundary, then more than one linear list of boundary point coordinates would have to be considered. Also, depending on the classification scheme involved, it may be sufficient to consider only the "major" boundary of the given shape or pattern.

3.2 The Procedures Involved

This method can be segmented into three steps to clearly illustrate the technique involved.

* Picture cells are termed four-way connected if they are defined to be connected only to their four surrounding cells with which they share common edges. Otherwise, they can be eight-way connected, i.e. they are defined to be also connected to those four cells with which they share common vertices. This obviously applies only to picture cells on quadruled grids, as picture cells on hexagonal grids which have common vertices also share common sides.

Step 1: Given a segment of n points, we connect all possible pairs of points in the segment to obtain a total of $n*(n-1)/2$ such lines. We shall refer to these lines drawn between pairs of points as "line-points" as each of these lines will be mapped into a one-dimensional space.

Step 2: We arbitrarily select the origin as the center of the picture retina, a 120 by 120 array of points. A perpendicular is drawn from this center to each line-point of the segment under consideration. The radial angle θ of the perpendicular is determined and entered as a count into an appropriate cell of the θ - histogram; that is, we map all $n*(n-1)/2$ line-points of the segment into a one-dimensional θ space.

Step 3: With the completed histogram containing at most $n*(n-1)/2$ points, we can now determine the tangent to the k th coordinate at the midpoint of the segment and the measure of curvature at point k using the following:

(a) At the θ-histogram cell with the highest value, we can say that the estimate of the tangent angle lies in this vicinity. So, we take the expected value of θ using the counts of each θ cell as its probability, that is,

$$\text{Tangent estimate} = E(\theta) = \sum \theta_i * \hat{P}(\theta_i) \qquad (3.2.1)$$

where

$$\hat{P}(\theta_i) = \frac{\text{count in the } \theta_i{}^{th} \text{ slot}}{\text{total number of line-points}} \qquad (3.2.2)$$

is an estimate of the probability of a line-point being mapped into the $\theta_i{}^{th}$ slot of the histogram.

(b) To evaluate the curvature measure at point k, we calculate the entropy of the histogram, that is,

$$\text{Curvature estimate} = \hat{H} = -\sum \hat{P}(\theta_i) * \log \hat{P}(\theta_i) \qquad (3.2.3)$$

where $\hat{P}(\theta_i)$ is as defined in Eqn. 3.2.2.

In the next chapter, theoretical results will be given to support these intuitive approaches to obtain the tangent angle and curvature measurements. Let us now consider some examples.

Eg. 1: For a given straight segment of a boundary, see Fig. 3.1, we will get only one value of θ for all the $n*(n-1)/2$ line-points. This would result in that the θ_i th slot of the histogram contains all the points and correctly implying, by our method of estimation, that the tangent to the k th point is the line with the transformed coordinate of θ_i. To calculate the curvature at the k th point, we substitute $\hat{P}(\theta_i) = 1$ and $\hat{P}(\theta_j) = 0$ for all other $j \neq i$, into Eqn. 3.2.3 to get a value of zero, which also is a correct estimate of the curvature of a straight line.

Eg. 2: With a (nicely) curved boundary segment like an arc of a circle, see Fig. 3.2, we will get the estimate of the tangent angle as the angle of the line with the transformed coordinate of the expected value of θ, using Eqn. 3.2.1. The curvature estimate obtained using Eqn. 3.2.3 is always a real non-negative value.

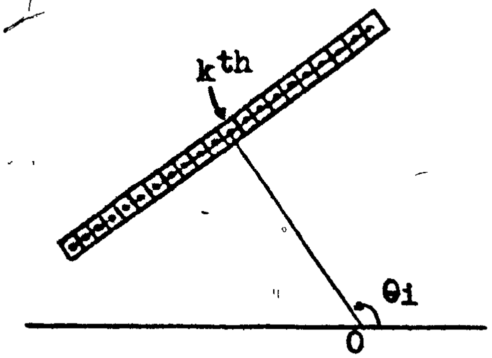


Fig. 3.1 Example 1

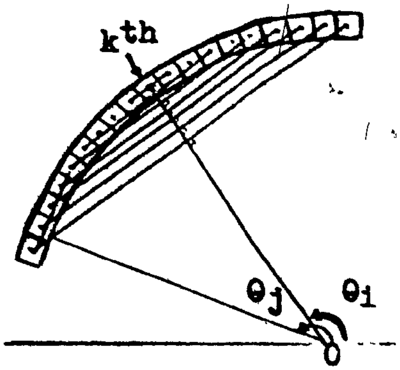


Fig. 3.2 Example 2

3.3 Definition of the Transformation Used

Essentially, as can be seen in Fig. 3.3, we map line-points (ℓ, ℓ') to their radial angles (θ) of their perpendiculars from the center with respect to an arbitrarily chosen reference line γ. In our case, it was convenient to use the x-axis as the reference line γ. This mapping will be referred to as the line to angle ($\ell - \theta$) transformation.

Given a line (ℓ) joining two points P_1 and P_2 on the picture retina, as defined in Section 3.2 Step 2, a perpendicular

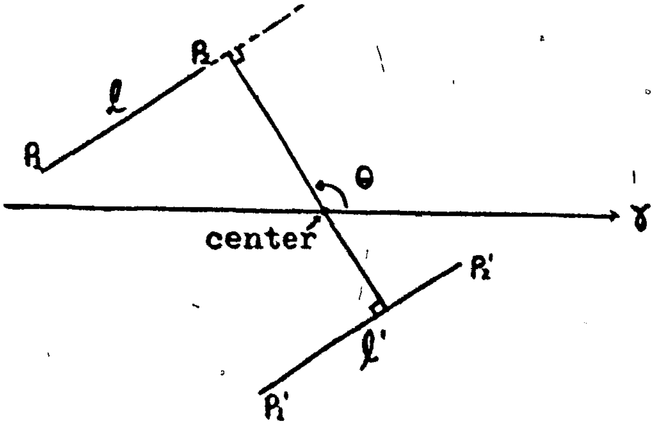


Fig. 3.3 The ℓ- θ Transformation

to this line is constructed from the center of the retina. The perpendicular to the line (ℓ) or the extension of it makes an angle of θ with respect to the reference line γ in a pre-specified angular direction. Thus, θ represents the line going through points P_1 and P_2; moreover, it represents all lines parallel to it.

The range of values for θ is defined to be between 0 and π radians. This would make the θ histogram obtained independent of the position of the origin, the center of the retina, as will be proven in Section 4.7.

3.4 The Programming Design of the System

Those who are not interested in a system design and computer implementation of this method may skip this section, the purpose of which is primarily to provide a sufficiently complete documentation of a program used to study and investigate this proposed method.

In this method, a picture is quantized on a 120 by 120 grid frame. The use of simply connected (in the topological sense) single-boundary test patterns is assumed. When studying the effects of quantization, pictures of varying sizes are presented on this same size array. Smaller pictures give the effect of gross quantization and produces similar results to that obtained when the picture is actually quantized on a larger sized grid frame.

Some variations to the original system will be discussed later to provide some alternate ideas for the purpose of comparison. Models of some sub-processes used will also be discussed.

We consider programming this system with a few variables that may eventually be fixed for any given problem. We define:

(i) NB as the size of the neighbourhood about a point. This makes the number of points in the segment = 2*NB+1, where the point under consideration is the (NB+1) st point — the point at the middle of the segment.

(ii) NS as the number of equal size angle partitions between 0 and π radians to provide NS distinct cells in the θ histogram.

(iii) XX as a threshold, a fraction of one, determined heuristically and used to determine the θ range of the histogram in the evaluations of the curvature measures. The reascns for including a threshold variable in this implementation will become clear in the following sections.

3.4.1 The General Subsystems

The process is divided into three main components outlined here. Each will be covered in more detail in the subsequent sections. Detailed algorithms and listings can be found in Appendix 1.

(I) Read in and determine the boundary points of a given test pattern. The set is then preprocessed and the sign of the curvature determined for each point; convex points being positive and concave points negative, denoted by +1 and -1 respectively in the array called "SGN".

(II) Read in the system variables (NB,NS,XX) and produce the measures of curvature and tangent angle estimates for each point on the boundary of the test pattern.

(III) For research and evaluation purposes, graphs are produced from the results obtained. The original and reconstructed pictures of the test patterns are also drawn using the CALCOMP Digital Plotter.

3.4.2 Subsystem I

The algorithms used to determine the boundary points, the sign of their curvature and the pre-processing procedure are briefly outlined here. A simple contour following method from Duda and Hart (52) is used to determine the boundary since we do not anticipate disconnected boundary points. Obviously, for patterns with more than one set of connected boundary points, this procedure is merely repeated for each.

The coordinates of the starting boundary point can be manually supplied or determined by a scan from the left edge. The boundary tracing algorithm is as follows:

> If the point is part of the picture, take a step left, otherwise, take a step right. Terminate when the boundary of the test pattern has been circumvented. See Fig. 3.4.

This bug would trace out the boundary in a clockwise direction. In mathematical notations, the coordinates of the next point (x_{k+1},y_{k+1}) are determined as follows:

$$\begin{aligned} x_{k+1} &= x_k + \{[y_k - y_{k-1}][1-2R(x_k,y_k)]\} \\ y_{k+1} &= y_k + \{[x_k - x_{k-1}][2R(x_k,y_k)-1]\} \end{aligned} \qquad (3.4.1)$$

where (x_k,y_k) is the present location, (x_{k-1},y_{k-1}) is the immediate previous location and $R(x_k,y_k)=\begin{cases}1 & \text{if } (x_k,y_k) \text{ is part of the picture,}\\ 0 & \text{otherwise.}\end{cases}$

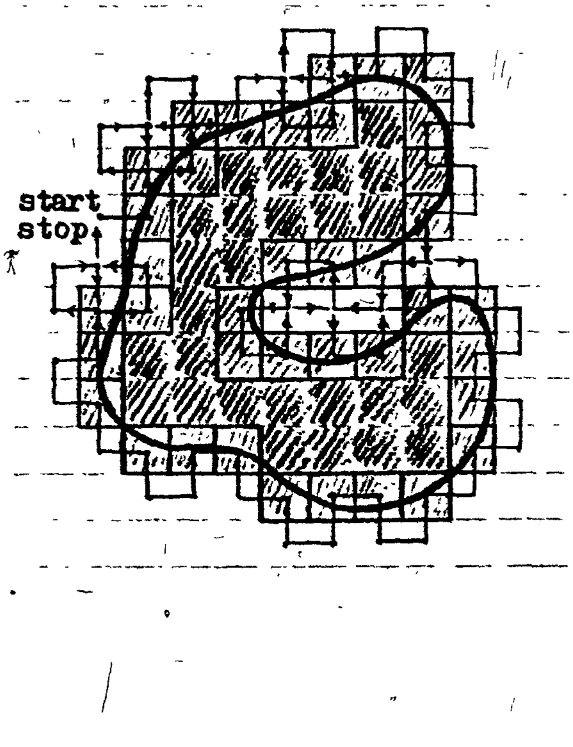


Fig. 3.4

Contour tracing by a "bug" in the clockwise direction.

After the set of boundary points has been established, they are preprocessed to remove points that we believe produce "noise" in the θ-histogram. Before going any further, consider a definition. Given three consecutive points on the boundary, 1, 2 and 3, if this ordered sequence of points fits in a 2 by 2 window, then it is called a "1-step staircase". All configurations of 1-step staircases are illustrated in Fig. 3.5.

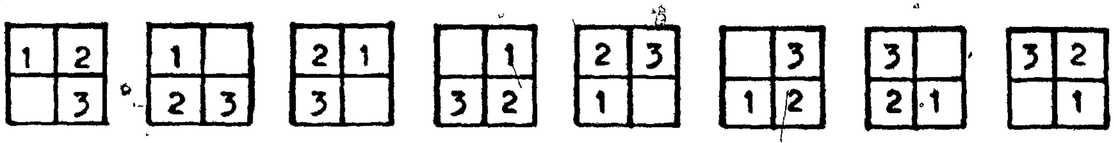


Fig. 3.5: 1-step staircases -- 3 points in a 2 x 2 window

We want a minimum 8-way connected boundary as much as possible, so we remove the second point of all 1-step staircases that can be found on the boundary. The algorithm for this preprocessing procedure is as follows:

1. Take 3 consecutive points and check for the existence of a 1-step staircase.
2. If it exists, then remove the middle point and set point(3) as point(1), otherwise, set point(2) as point(1) for the next sequence of three consecutive points.
3. Repeat step 1 until the boundary has been traversed.

The logic for checking for the existence of 1-step staircases is best represented as a flowchart in Fig. 3.6, where the coordinates of the three points are: (i_1,j_1), (i_2,j_2) and (i_3,j_3).

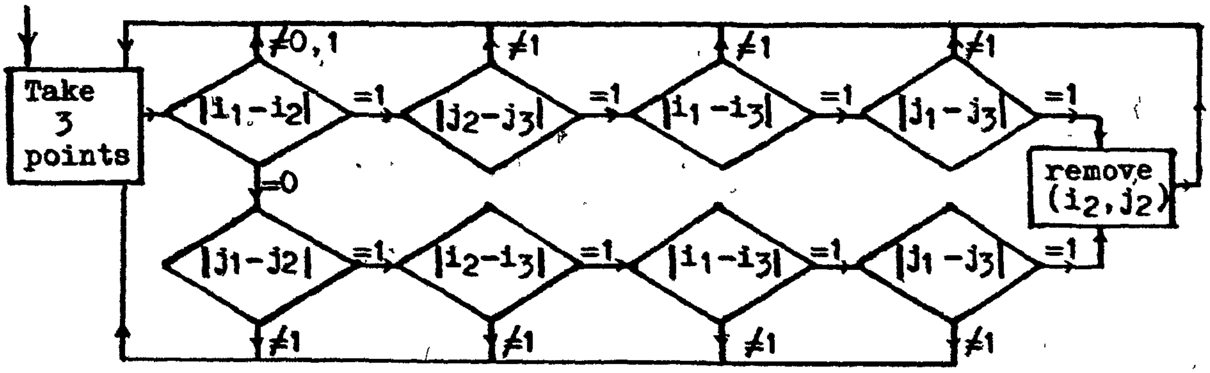


Fig. 3.6: Flow logic for detecting 1-step staircases

In programming this logic, we use the knowledge that approximately 15 % of the points will be removed, so that at about 85 % of the time, only two evaluations and comparisons out of the possible eight are done. This has shown to be relatively more efficient than other logics for this procedure.

To determine the sign of the curvature, we comply with the standard (classical) convention of convexity and concavity. As boundary tracing is done in a clockwise direction around the test pattern, the following heuristic algorithm was adequate for the determination of the signs of the curvature for each boundary point.

Considered, in Fig. 3.7, three ordered points, P_1, P_2 and P_3 with coordinates (x_1,y_1), (x_2,y_2) and (x_3,y_3) respectively and the vectors $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_2P_3}$. We want to determine the sign of the curvature for point P_2. Let Θ_1 be the angle between $\overrightarrow{P_1P_2}$ and the initial coordinate system's x-axis $\{e_1\}$ as shown. If vector $\overrightarrow{P_2P_3}$ deviates negatively, i.e. in a clockwise direction from the vector $\overrightarrow{P_1P_2}$, then P_2 is a convex point; otherwise, P_2 is a concave point. Knowing this, we only need to determine the y-coordinate value of point P_3 with respect to the coordinate system having point P_1 as the origin and vector $\overrightarrow{P_1P_2}$ indicating the direction of its x-axis.

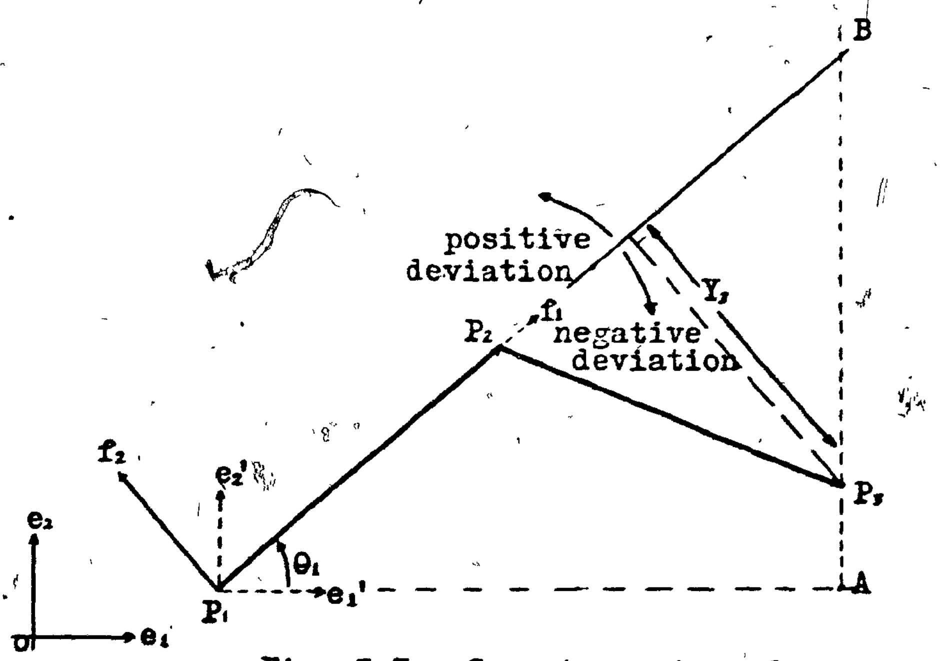


Fig. 3.7: Curvature sign of P_2 = sign of Y

First, we translate the origin to point P_1, to give us the system with basis $\{e_1',e_2'\}$. Then we rotate the e_1'-$\hat{e}_2'$ axis θ_1 degrees so that the positive x-axis aligns with the vector $\overrightarrow{P_1P_2}$ to give us the system with the basis $\{f_1,f_2\}$. On this coordinate system, we want to determine the y-component of point P_3.

Let $(\hat{x}_3,\hat{y}_3)$ be the coordinates of P_3 in the $\{e_1',e_2'\}$ system. Then the coordinates (X_3,Y_3) of P_3 in $\{f_1,f_2\}$ are given by:

$$\hat{x}_3 = X_3 * \cos\theta_1 - Y_3 * \sin\theta_1 \qquad (3.4.2)$$

$$\hat{y}_3 = X_3 * \sin\theta_1 + Y_3 * \cos\theta_1 \qquad (3.4.3)$$

To extract the y-component value, Y_3, from the above equations, we perform the operations: $\cos\theta_1 *$ Eqn. 3.4.3 - $\sin\theta_1 *$ Eqn. 3.4.2 to get: $\hat{y}_3 * \cos\theta_1 - \hat{x}_3 * \sin\theta_1 = Y_3\,(\cos^2\theta_1 + \sin^2\theta_1)$. Thus,

$$Y_3 = (y_3-y_1)*\cos\theta_1 - (x_3-x_1)*\sin\theta_1 \qquad (3.4.4)$$

So, we have that if $Y_3 \leqslant 0$, P_2 is convex, otherwise, P_2 is concave.

To realize the significance of Eqn. 3.4.4, we can reduce its terms and interpret it as: (with reference to Fig. 3.7),

$$Y_3 = (y_3-y_1)*(x_3-x_1)/|\overrightarrow{P_1B}| - (x_3-x_1)*|\overrightarrow{AB}|/|\overrightarrow{P_1B}|$$

i.e. $Y_3 = \left(\frac{x_3-x_1}{|P_1B|}\right) * \left\{(y_3-y_1) - |\overrightarrow{AB}|\right\}$

In effect, if $|\overrightarrow{AB}| > |\overrightarrow{AP_3}|$ then Y_3 is negative, thus P_2 is convex. Otherwise, Y_3 will be positive, thus P_2 is concave.

3.4.3 Subsystem II

Subsystem I presents a list of boundary point coordinates and the signs of their curvature to Subsystem II which then continue processing using this reduced data from the original pattern by going through the following steps:

i. Read in the system variables and establish the first n points as the segment to be used for computation, and the middle point as the k th point under consideration.

ii. Determine the $n*(n-1)/2$ values of θ (as defined in Section 3.3), from this segment of n points.

iii. Map these line-points into their proper cells in the θ-value histogram.

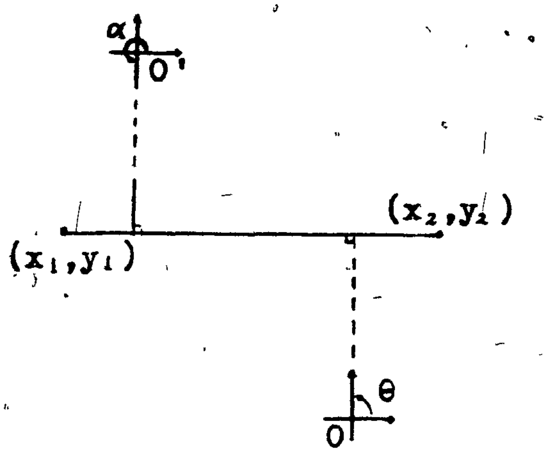


Fig. 3.8: Case 1
when $(y_2-y_1)=0$
$\therefore\ \theta = \begin{cases} \pi/2 & \text{(for 0)} \\ \alpha - \pi/2 & \text{(for 0')} \end{cases}$

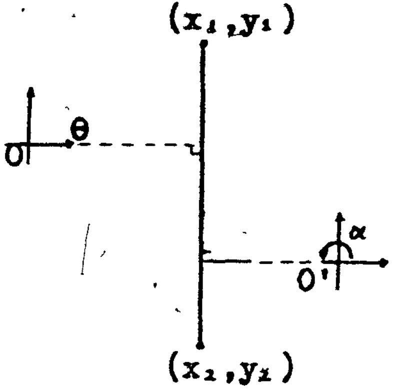


Fig. 3.9: Case 2
when $(x_2-x_1)=0$
$\therefore\ \theta = \begin{cases} 0 & \text{(for 0)} \\ \alpha - \pi/2 & \text{(for 0')} \end{cases}$

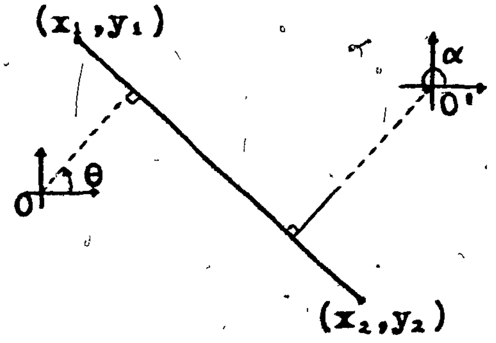


Fig. 3.10: Case 3
when slope < 0
$\therefore\ \theta = \begin{cases} \pi/2 - \arctan(-\text{slope}) \\ \alpha - \pi/2 \quad \text{(for 0')} \end{cases}$

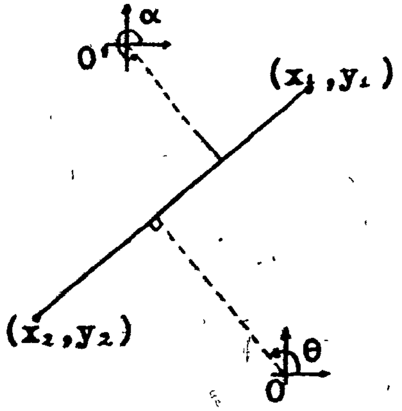


Fig. 3.11: Case 4
when slope > 0
$\theta = \begin{cases} \pi/2 + \arctan(\text{slope}) \\ \alpha - \ /2 \quad \text{(for 0')} \end{cases}$

iv. Determine the measures of curvature and tangent angle estimates from this histogram.

v. Move the segment "ahead" by one point and "recopy" the "matrix of θ-values", leaving out values in the first row and determining values for the last column. This step will be explained in more detail later.

vi. Repeat step (iii) until the k th point reaches the starting position, indicating that the boundary has been traversed.

vii. Produce graphs of the measures and copy the arrays of values onto disk or card files to be used in Subsystem III.

We can briefly discuss an algorithm used to determine the θ value for each line-point. By the definition of the ℓ-θ transformation, the range of the radial angle is from 0 to π, i.e. modulo π. Given two points, (x_1,y_1), and (x_2,y_2) on the boundary of the test pattern, we use the slope $(y_2-y_1)/(x_2-x_1)$ to determine the radial angles. Points marked 0' in Figures 3.8 - 3.11 denote arbitrary alternate centers of the picture retina. Consider the four cases:

Case 1. The line-point is horizontal, i.e. $(y_2-y_1)=0$, or the slope is zero. Then the radial angle $\theta = \pi/2$

Case 2. The line-point is vertical, i.e. $(x_2-x_1)=0$ and thus $\theta = 0$

Case 3. If the slope is less than 0, then $\theta = \pi/2 - \arctan(-\text{slope})$

Case 4. If the slope > 0, then $\theta = \pi/2 + \arctan(\text{slope})$

For a known NS, the number of equal size partitions between 0 and π, a given value of θ will be mapped into the $\left\lfloor \frac{\theta}{\pi/NS} + 1 \right\rfloor^{th}$ slot on the θ-axis of the histogram. That is, one would be added to the present histogram count of that slot.

Before explaining Step v, a point on the storage structure and data manipulation technique of the θ values ought to be made. For a given segment of n = 2*NB+1 points, we have to store at most n*(n-1)/2 θ values. These are stored in a virtual upper triangular matrix where one can easily identify each θ value by its location in the array; a θ value in the (i,j) th position of the array is the radial angle of a line-point between point i and point j of the segment of n points, i,j $\leq$ n and i $\neq$ j. Obviously, the use of the whole matrix would be inefficient in terms of the storage space used. Moreover, the use of an array structure is unnecessary. Since there is no need to refer to the θ values by their position in the matrix, we use a linear vector of size $n*(n-1)/2$ with an index k that is related to the matrix structure indices i and j by the relations: $k = (i-1)*(n-1) + j$.

In Step v, when we move the segment "ahead" by one point, only the values related to the previous first point become obsolete and the new values from all the remaining points to the new one just added to the end of the segment are calculated. In the procedure to "recopy" the "matrix", we simply omit the first row and calculate values for the last column of this imaginary matrix; thus, only (n-1) new values are needed each time it is moved ahead by one point. This technique results in a substantial reduction in the amount of computations required in this program.

3.5 Analysis of the Parameters Involved

The presence of three variables, NB, NS and XX defined in Sec. 3.4 each ought to be justified and the effects of their varying values investigated in order to arrive at an ideal set of these variable values that would give the best or most desirable results. Two of these, NB and NS are an integral part of the proposed method. The purpose of NB is to segment the boundary into n point partitions from which the described procedures could be applied to obtain values that would be assigned to the point at the midpoint of each partition.

The purpose of the parameter NS, is to provide a histogram with the indicated NS number of cells into which the θ values of line-points can be mapped. The third parameter XX, the threshold, was introduced in order to make the technique attain a higher degree of accuracy and correlation to intuitive senses of curvature. The exact reason as to why it was introduced and the way this parameter is used will be given when we get to the analysis of this variable.

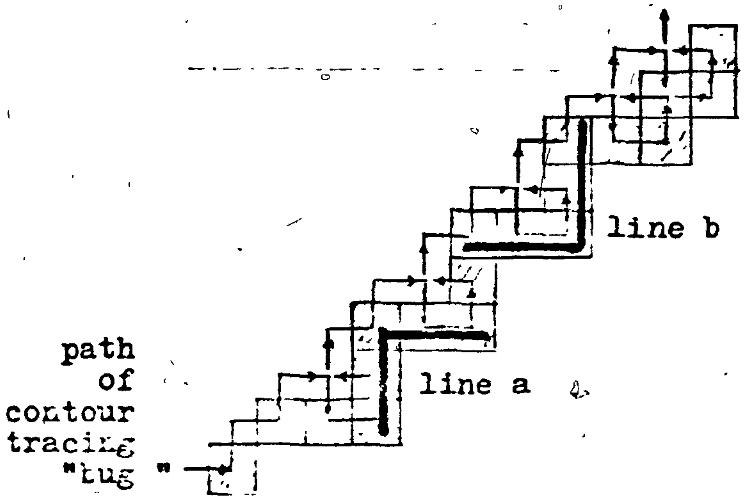


Fig. 3.12
An Unprocessed Boundary

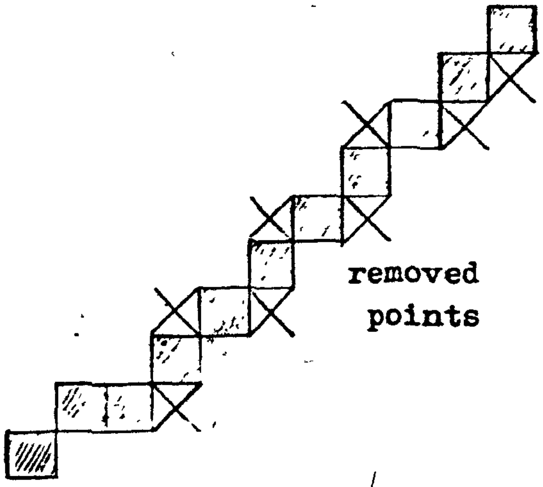


Fig. 3.13
The Processed Boundary

3.5.1 NB - The Size of the Neighbourhood

Given an unprocessed boundary like in Fig. 3.12, a small size NB of say, two, would produce results for the boundary that could be mis-interpreted as part of a line that is relatively convex (like line a) or concave (like line b). If this same boundary segment was preprocessed to remove 1-step staircases, using a small neighbourhood size would not produce locally convex or concave sections of the boundary, see Fig. 3.13. Thus measures from a preprocessed boundary would be relatively more accurate.

On the other hand, the size of the neighbourhood would also reflect the amount of noise present; a larger NB size used on the unprocessed boundary could reduce the amount of noise in the resulting measures. So, we want to select a suitable neighborhood size that might provide more realistic correlations to the actual curvature and tangent angle measures.

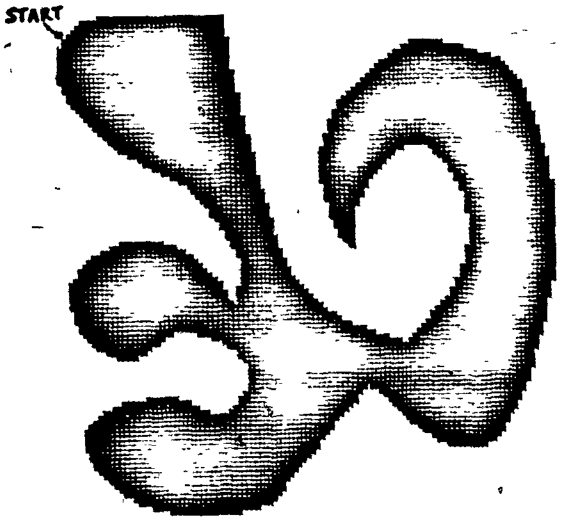
In order to avoid or reduce the effects of noise, we can do one or some of the following:

a) Quantize the given pattern using a much smaller grid in order to obtain more details. Then use its preprocessed version. Repeated use of this is undesirable in terms of the storage space required. Besides, it would involve a lot more computations than one can afford or wish to have.

b) Increase the size of the neighbourhood to smooth out some of the noise. This size should also be limited to a proportion of the size of the smallest bulbous appendage that one might have on the pattern and would want to have recognized and present in the resulting pattern. A large size NB would tend to give axial directions instead of local tangent directions.

c) Strive to obtain a minimum 8-way connected boundary. This implies eliminating as many 1-step staircases as we can. With this change, we obtain a much less noisy picture with less boundary points, and a "good" size of NB, the neighbourhood, would tend to be smaller, thus, also reducing the amount of computations.

One cannot say if there is a neighbourhood size that would produce optimal performance for all evaluations since it all



THE ORIGINAL PICTURE

Fig. 3.14

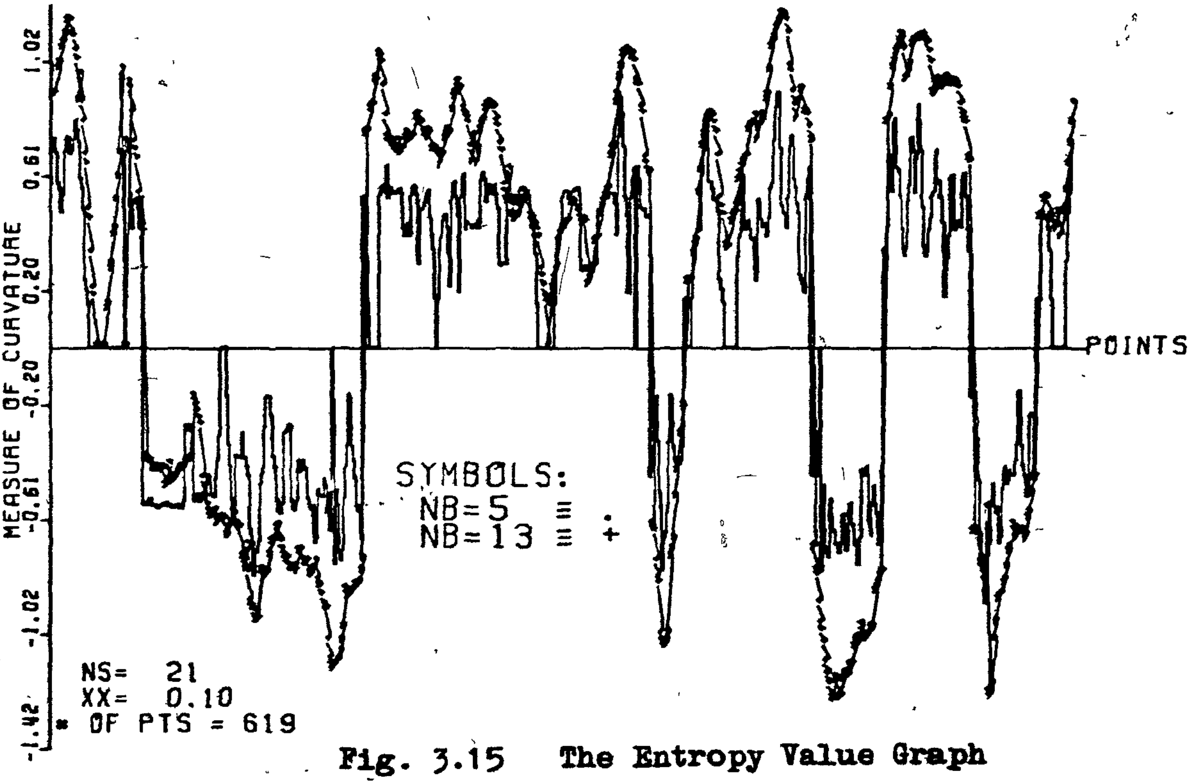


Fig. 3.15 The Entropy Value Graph

depends on the test pattern and the quantization level of it. But for a given class of test patterns, it can be derived by heuristic methods or by trial and error; that is, by varying the size of NB and objectively deciding which produces results that most closely resemble and correlate with theoretical measures. This would be rather time consuming since for a segment of size n, at most n*(n-1)/2 line-points are being considered for each point on the boundary, thus the amount of computation increases by an order of n^2. This process of objectively selecting the best results has been applied to other recognition procedures like selecting the best skeletonization algorithm or image processing techniques. So it may well be a suitable and practical method for our case.

We use a test pattern shown in Fig. 3.14, that has several different curves having different degrees of curvature to generally represent the class of patterns which may include Chinese characters, chromosomes, handwritten alphabets, etc. The best NB size obtained from this could also then be suitable for other patterns in this category of shapes. To demonstrate the difference in the results when using different neighbourhood sizes, consider the cases for NB = 5 and 13.

With reference to the curvature value graphs in Fig. 3.15, some observations specific to these variables can be made. One quickly observes that the range of the values for NB = 13 is larger than that for NB = 5, and that NB = 13 produces a much smoother and regularly varying graph as opposed to a more rapidly varying graph of NB = 5. We also note that the NB = 5 curve attains the zero value more frequently than the NB = 13 curve.

This phenomenon of a more rounded curve is evidently caused by a larger neighbourhood size which tends to average up and smooth out the curvature measurements giving slight changes to correspondingly slight variations, whereas the graph of the smaller NB size exhibited sensitivity to local changes on the boundary, giving a jagged-looking curve. We can conclude that the graph of the larger NB size can primarily be usedto identify points of higher curvature although it is quite far from a realistic point by point representation of the pattern's curvature. A smaller NB

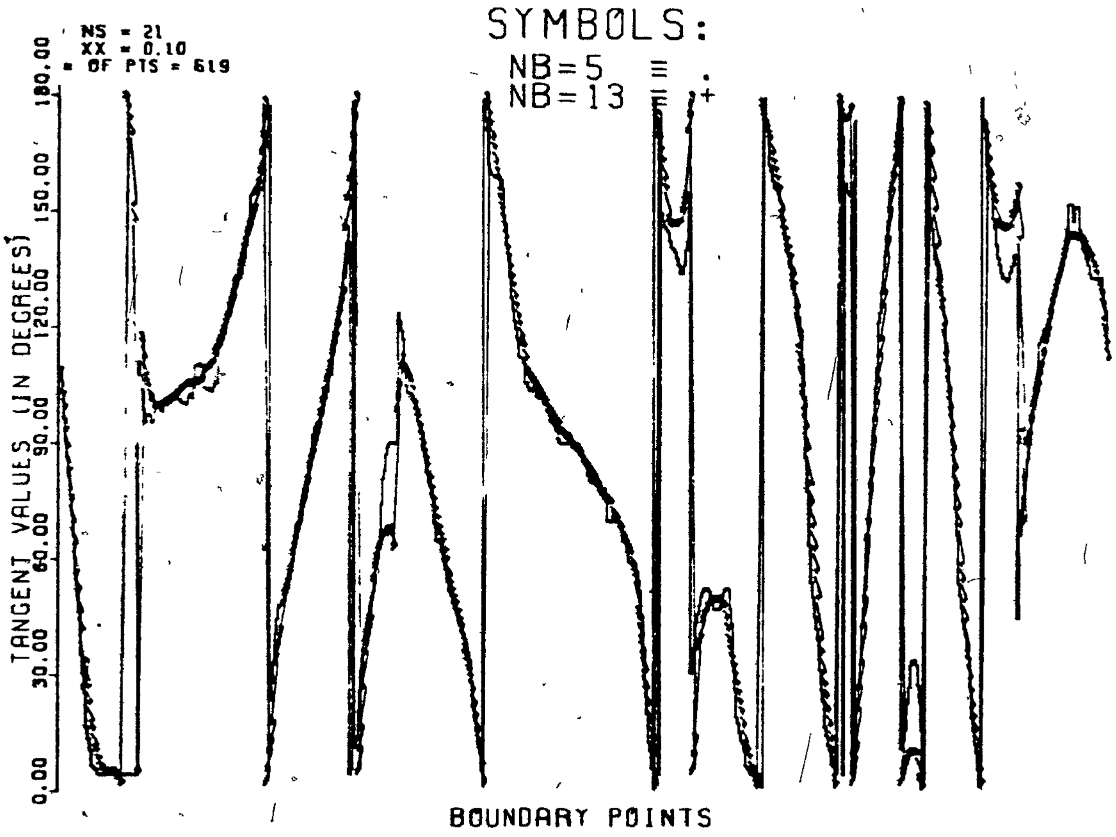


Fig. 3.16 The Tangent Value Graph

gives a more realistic local representation of the curvature, although a smoothed version of it would practically be more useful.

For the tangent estimates, we observe in Fig. 3.16 that a larger NB size gives more averaged-out estimates of the tangent angle. When reconstructing the picture by drawing a short line through each point with the estimated tangent angle, we observe in Fig. 3.17 that the larger NB size gives more locally mis-directed lines, whereas the smaller NB size gives a locally more realistic set of tangent estimates. So, a smaller size neighbourhood like 5 has provided more accurate and realistic tangent estimates.

3.5.2 NS - The Number of Angle Sectors

This divides π radians into NS equal size partitions from whence calculated θ values can be placed into their respective cells of the θ histogram. In using this histogram, the cell with the highest count is first located. Then a scan of the counts of cells on either side of it is done until a zero cell is encountered. This region of non-zero cells on both sides of the cell with the highest count will be known as the high frequency region and it would be used in the calculation of the curvature and tangent angle measures.

In assuming a uniform distribution of θ values in each angular partition, the mean, being the value of the angle at the center of the partition, would sufficiently represent the angle sector. Hence, an odd number of partitions would be ideal to ensure the availability of the value $\pi/2$ radians, i.e. 90°. If NS is large, like 180, then each angle degree would be represented by a histogram slot. The only disadvantage in this is that the resulting histogram could have a generally known high frequency region with embedded zero-value cells. These would indicate cut-off points when using the above procedures, causing subsequent cells to be ignored in the computations. Furthermore, due to the presence of any vertically or horizontally aligned quantized boundary points, the possibility of selecting an incorrect high frequency region is higher; this would evidently result in bias and/or incorrect measurements.

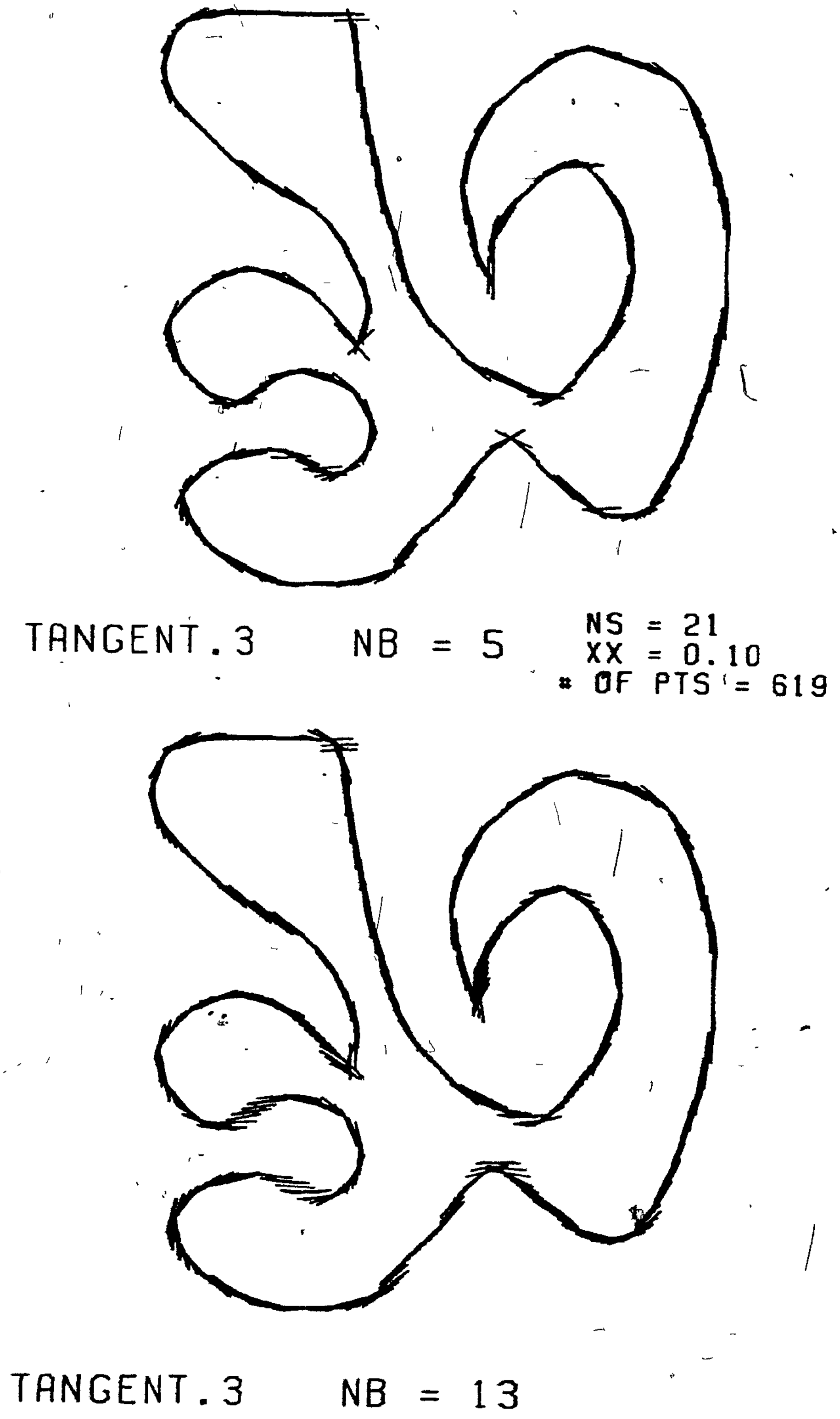


Fig. 3.17 The Reconstructed Shape

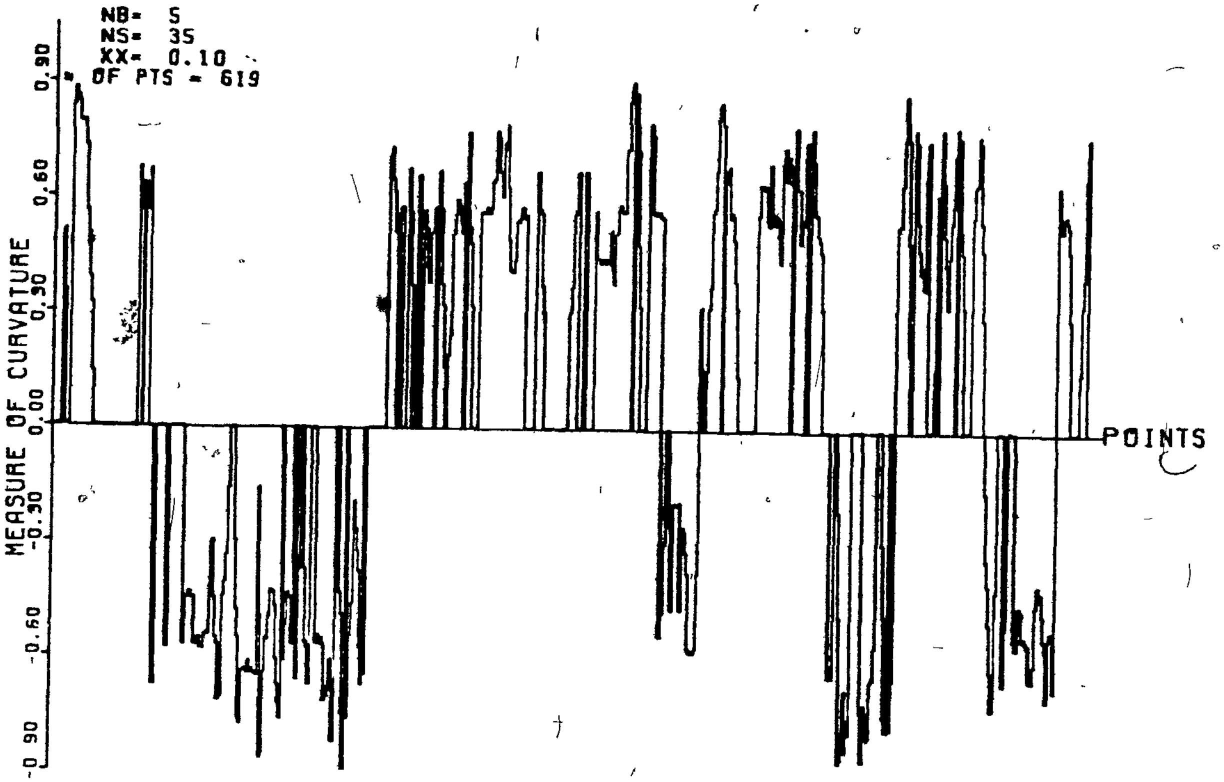


Fig. 3.18 The Entropy Value Graph

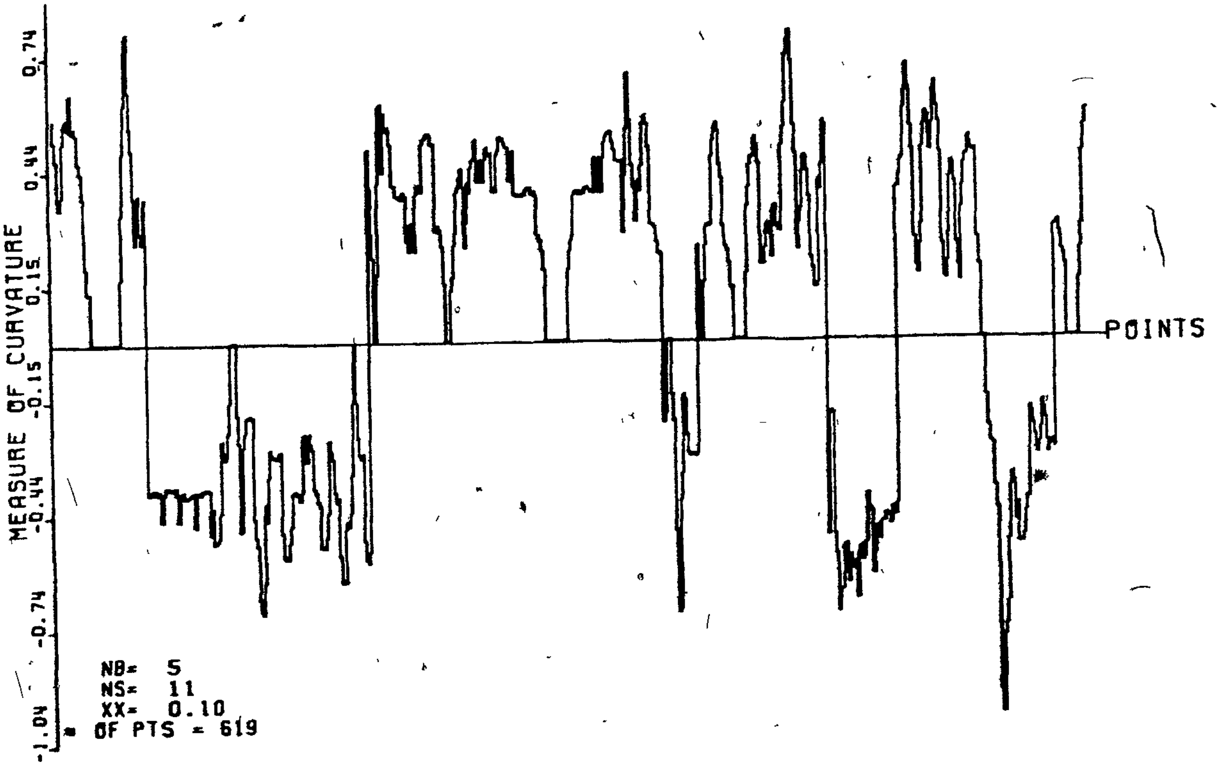


Fig. 3.19 The Entropy Value Graph

On the other hand, a small value of NS may not enable one to discriminate Θ values sufficiently well, thus, it may not provide enough information to determine which Θ value would best represent the tangent angle.

Let us observe the effects of using different NS values by briefly considering the three cases:

i. Using a small NB with a large NS, e.g. (5,35)

ii. Using a small NB with a small NS, e.g. (5,11)

iii. Using a large NB with a large NS, e.g. (13,35).

i) Small NB and large NS: As can be seen in Fig. 3.18, we notice an irregular graph that attains the value zero quite frequently and often without any sort of gradual approach to it. The occurrence of this erratic back and forth movement from zero to a non-zero value and back at some regions can be explained as follows.

A large NS value implies small histogram partitions for Θ values. Thus, only lines with very small Θ value differences accumulate in the same histogram slot. Having a small NB value is like considering a small segment of a large curve. Consequently, at times, they are perceived as straight lines. This would result in line-points being classed in only a few slots of the histogram. The small size Θ partitions would make the possibility of having the highest slot separated on both sides by zero valued slots even higher. Consequently, the curvature measure would be evaluated on a high frequency region containing only a single slot and thus, the resulting value would be zero. So only at regions where the curve varies regularly in a small segment would this measure behave differently. This combination of a small NB and a large NS has shown to have made the measuring mechanism too sensitive to vary proportionately with differences in curvature on the digitized curve.

ii) Small NB and small NS: Reducing the value of NS, as can be observed in Fig. 3.19, causes the curvature graph to become relatively smoother. This is due to the greatly reduced occurrence of having the highest slot separated on both sides

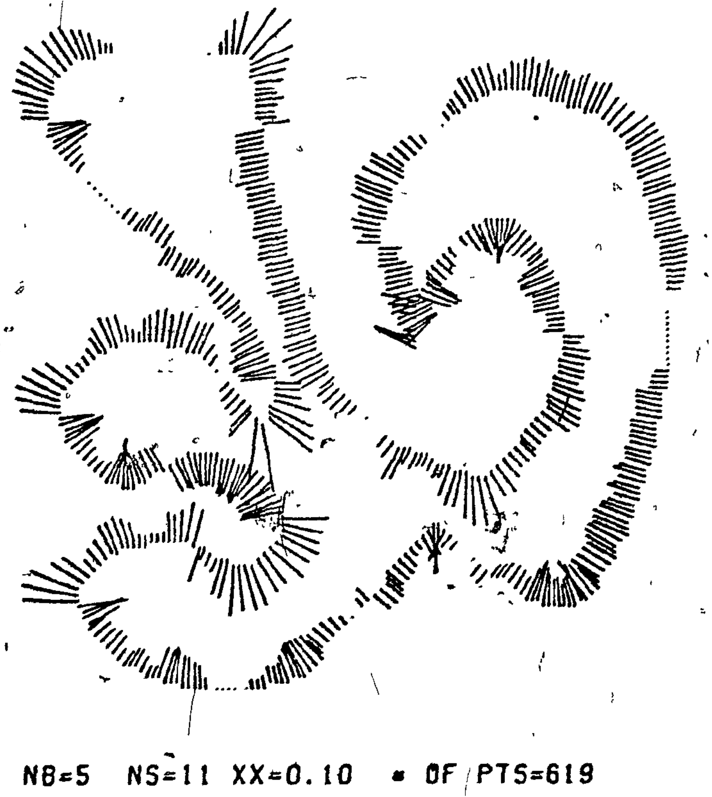


Fig. 3.20 The length of the lines drawn indicates the boundary points' relative degree of curvature

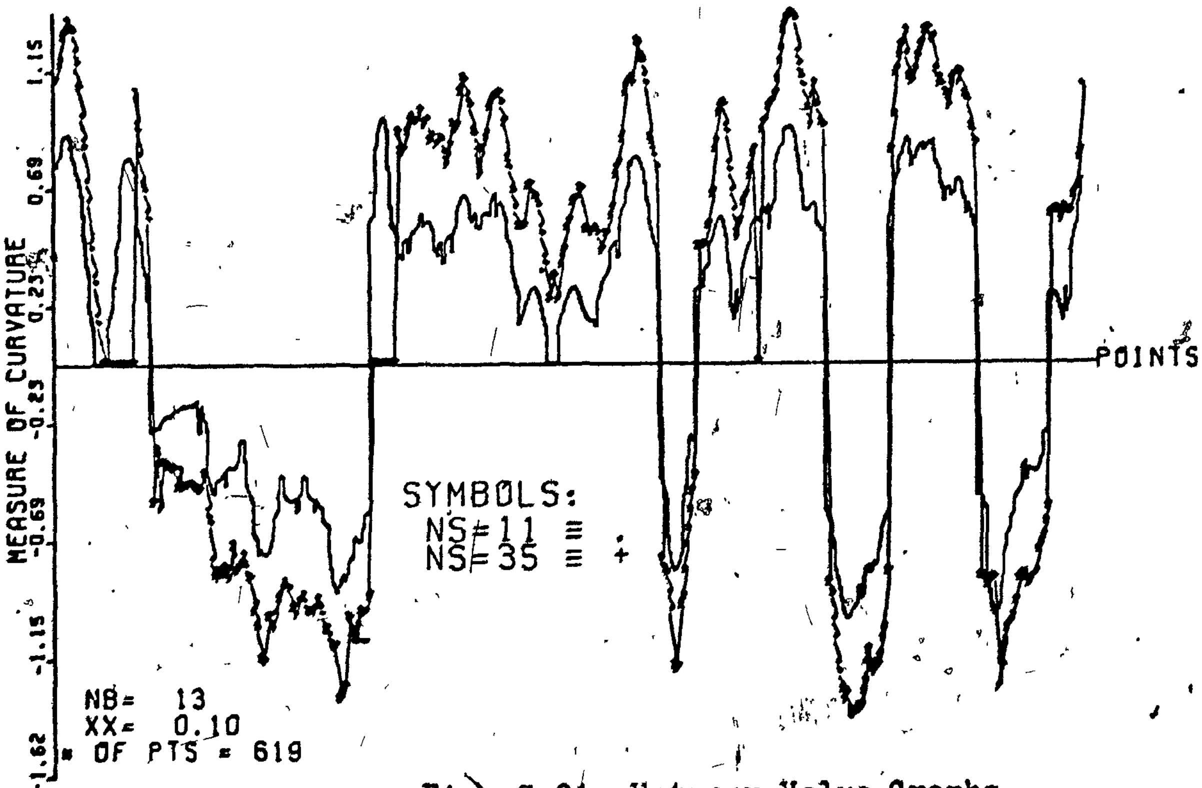


Fig. 3.21 Entropy Value Graphs

by zero-valued slots. Slight variations at some stretches of the graph can be interpreted as points having similar curvature. Likewise, prominent peaks are seen as gradual increases and decreases in curvature. In other words, one can perceive some continuity in the calculated curvature measures. Thus, a smaller NS provides a much better representation of the curvature, as can be observed by the indicator lines in Fig. 3.20.

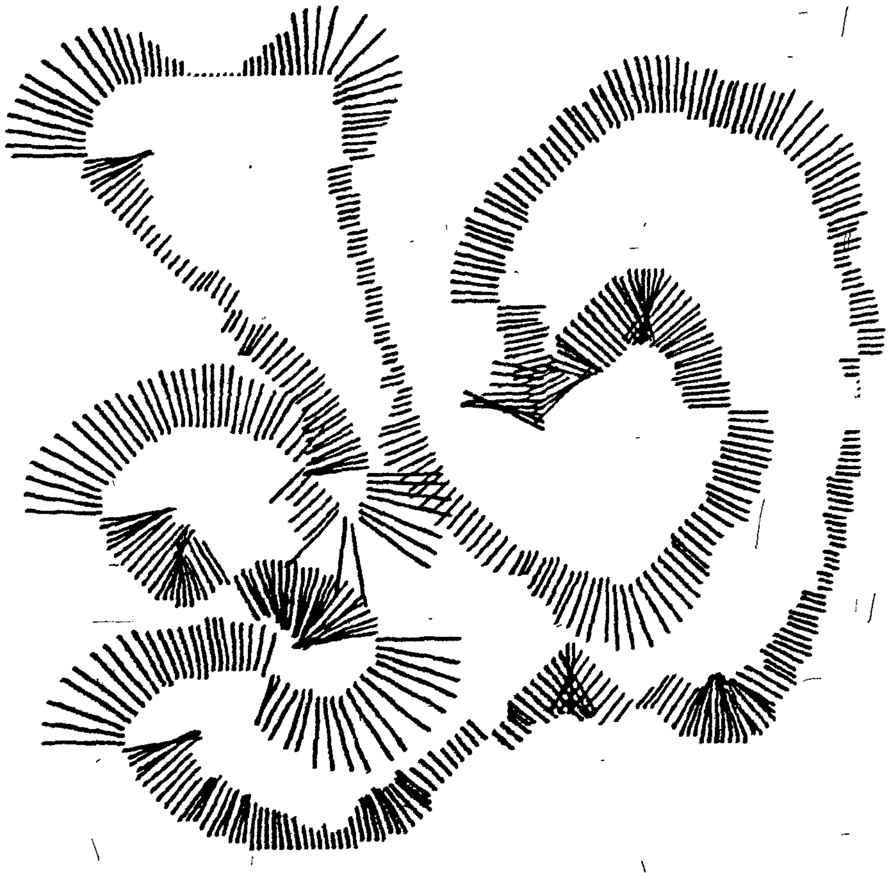
iii) Large NB and large NS: When using a large neighbourhood size, a large NS size produces nicely prominent peaks as can be observed in Fig. 3.21, highlighting the regions, and not points of higher curvature. We notice a lack of low curvature regions which we can assume exist somewhere between the high curvature regions. So, this combination of higher values of NB and NS can be used for the purpose of detecting high curvature regions although a smaller size NS would not drastically reduce its performance; compare Fig. 3.22 with Fig. 3.20.

For tangent estimates: Generally, a large NB value would produce relatively averaged-out versions of the original shape. In our example in Fig. 3.23, the tangent tends to take on the general direction of the section, resulting in mis-directed lines at points of higher curvature and points of inflection. A smaller NS would only give some smoothly varying tangent vectors at some regions of higher curvature.

When NB is smaller, like in Fig. 3.24 when NB=5, reasonable local estimates of the tangents are generally obtained, but a smaller size NS would give relatively better values, even at regions of higher curvature. For a larger NS, the main axial directions obtained give more locally mis-directed lines. Thus a smaller NB and NS has shown to produce better tangent estimates. In this case, it was when NB = 5 and NS = 11.

3.5.3 XX - The Acceptable Fraction

In our method of locating a higher probability region on the histogram, where the neighbouring slots of the highest slot are scanned until a zero slot is encountered, there is always a



NB=13 NS=11 XX=0.10 * OF PTS=619

Fig. 3.22 The length of the lines drawn indicates the boundary points' relative degree of curvature.
The direction of the lines is immaterial; it changes as a consequence of the algebraic manipulations used to draw them perpendicular to the boundary.

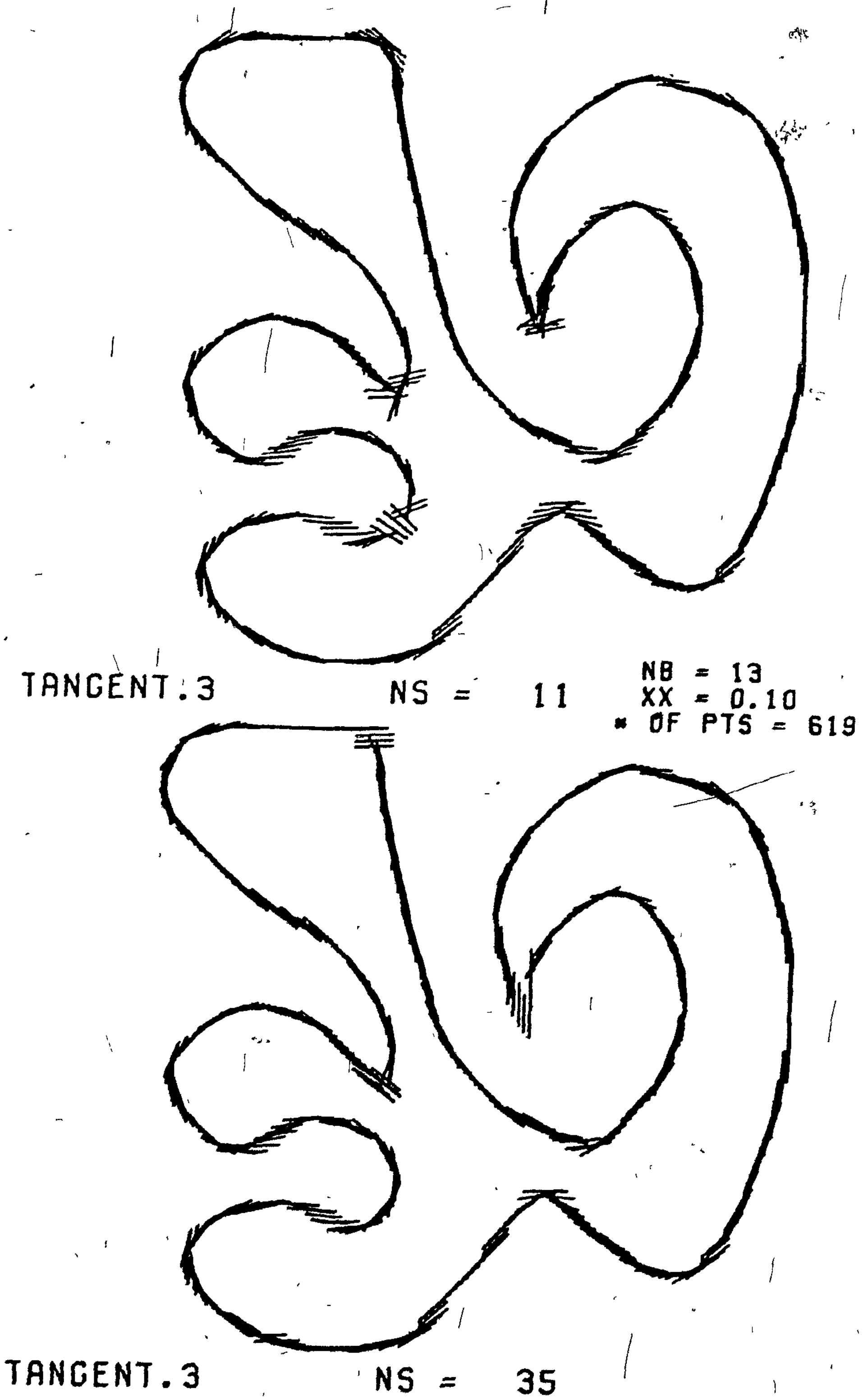


Fig. 3.23 The Reconstructed Shape

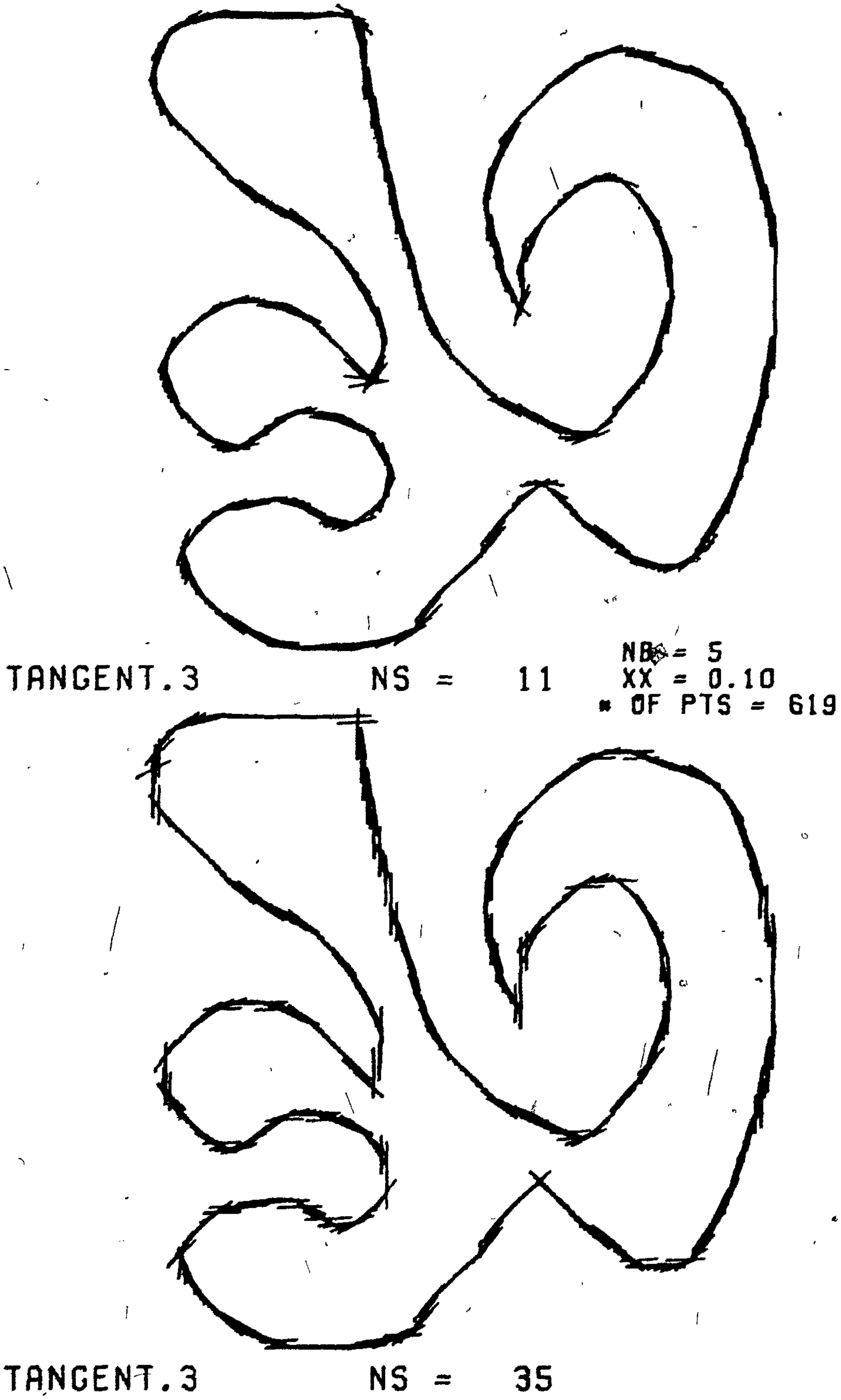


Fig. 3.24 The Reconstructed Shape

chance that the whole histogram becomes the high frequency region. All that needs to happen is that every slot has one count or more. In evaluating the entire histogram, chances are that the estimates of curvature or tangent would not be as realistic. If we can isolate a more specific higher probability region, then the estimated tangents and curvature values would obviously tend to be closer to actual values.

It is with this in mind that the use of a fractional variable, XX, was introduced. This fraction functions by only allowing slots with counts that are more than this fraction of the highest slot count to be included in the high frequency region of the histogram. In the scan from both sides of the slot with the highest count, as soon as a slot is encountered with a count that is less than this fraction of the highest count, the high frequency region ends.

The initial idea on this thresholding value is that it should vary proportionately with the size of NB, since for a larger NB, there is a higher tendency that the larger number of line-points would produce a histogram with no zero value slots. A small value of XX would tend to return a larger θ region, if not the entire θ range. If XX is higher, the θ region singled out as the high probability region would tend to be smaller and very probably not the entire histogram θ range. Without these other slots that cause estimates to deviate from the actual values, the measures obtained would be much more realistic.

As XX approaches 1, the region reduces to one or very few slots, all having very close or equal values. This results in tangent estimates showing a step-like graph due to its limited field of definition. Moreover, curvature estimates would often be zero if not close to it. So, a high value of XX is undesirable. Furthermore, it has been observed that as XX becomes less than 0.5, more reasonable regions were chosen and more realistic measures of curvature and tangent angle estimates were produced.

From all the above observations, we can now provide some suitable sets of variables for the system. If we only want to determine the regions of higher curvature, we use a large value of

NB and NS, like 13 and 35 respectively. If only points of higher curvature are required, a smaller NB value and a large value of NS would do, like 7 and 35 respectively. If we want locally good estimates of curvature, the graph of which is quite smooth to represent gradual changes in curvature, we use a small value of NB and NS, like 5 and 11 respectively. If we only want locally good and accurate tangent estimates, the small NB and NS, like 5 and 11 would also be suitable.

Since we want curvature measures to be as smoothly varying as possible and showing differences of higher and lower curvature points distinctively, and tangent estimates to show more details in terms of small variations of angles on the boundary to realistically represent the intended shape of the test pattern, we have selected the set of variables NB = 5, NS = 11 and XX = 0.1 as the best to most reasonably achieve our requirements. This value of XX had shown to produce the best result for the NB = 5, NS = 11 combination.

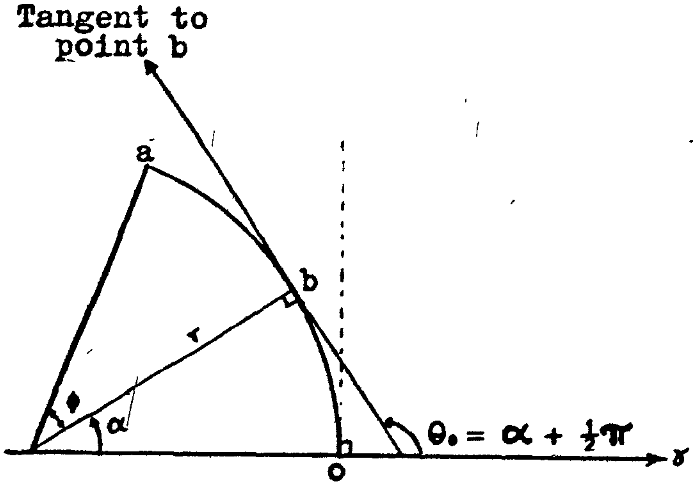


Fig. 4.1 If the length of the arc $\widehat{ob}$ is Δs, then
$\alpha \cong \Delta \emptyset = \Delta s/r$

If b is at the midpoint of arc $\widehat{ao}$, then $\alpha = \emptyset$, and the tangent to point b is
$\theta_0 = \frac{1}{2}\pi + \frac{1}{2}\emptyset'$
where $\emptyset' = \alpha + \emptyset$ is the subtending angle of arc $\widehat{ao}$.

4. Some Theoretical Considerations

4.1 Introduction

In this chapter we consider a few models of the proposed method of measuring curvature in order to establish some background that might give us more insight as to how and why this method works. Section 4.2 gives a discrete model and its simulated results, Section 4.3 gives the theoretical background to this system with respect to the continuous environment and finally, Sections 4.4 to 4.7 give some theoretical implications from our method.

Arcs of circles of various radii have been ideal as the object of our consideration because they give us a regularity that is rather unique; they are geometrically easy to handle, their curvature is the same at every point on the arc and classically known to be 1/radius. Furthermore, if we can consider an alternate way of measuring the curvature of a segment of an arbitrary curve as taking the curvature of the circle that best fits the segment, then circles are a basis for measuring curvature. In this respect, our results using arcs of circles can be considered as bases for the proposed method of measuring curvature.

4.2 Discrete Case

Given n consecutive boundary points, we employ a histogram to show the frequency of the radial angles of all the line-points. Now, we want to determine the frequency distribution of the radial angles generated by all minor chords of an arc of a circle of radius r and subtended by an angle of size $\emptyset$. In Fig. 4.1, the tangent to the arc at point b is a vector of angle

$$\theta_0 = (\alpha + \frac{\pi}{2}) \qquad (4.2.1)$$

with reference to the horizontal line γ as the zero direction. Obviously, if $\alpha = 0$, the angle of the tangent to point b would be

$$\theta_0 = \pi/2 \qquad (4.2.2)$$

In our considerations, we can assume $\alpha = 0$.

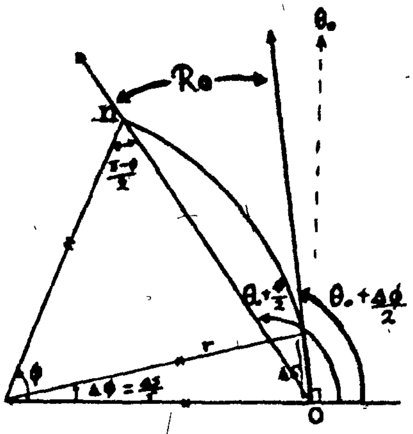


Fig. 4.2 The range from the first point.

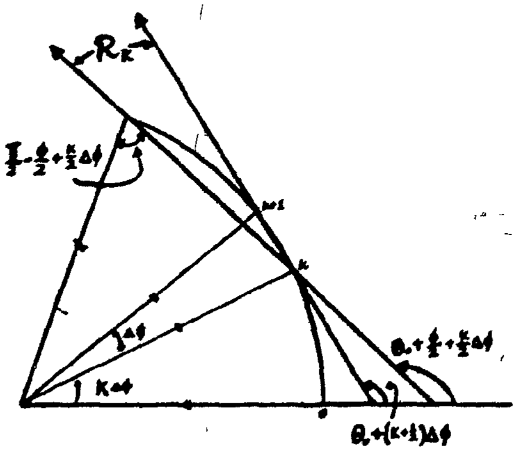


Fig. 4.3 The range from the k th point

* We chose (n+1) points in order to get n equal sized (Δ s) segments on the arc, also implying that the subtending angle $\emptyset$ is divided into n equal sized angle partitions, each of size $\Delta\emptyset = \Delta s/r$

We now resort to a discrete process to simulate the proposed method for a digitized version of an arc $\widehat{ab}$. If we have (n+1) equally spaced points*, Δs distance apart on the arc, we would have n*(n+1)/2 possible line-points and their corresponding θ values. An easy way to account for all these θ values would be to establish their ranges in an orderly manner.

If consecutive points are Δs distance apart, then the angle between the radii drawn to any two consecutive points on the arc is $\Delta\phi = \Delta s/r$, where r is the radius of the arc. If we number the (n+1) points on the arc sequentially as point 0 to n and define $\mathcal{R}_i$ as the range of the θ values of all line-points originating from point i to all points $j > i$, then $\mathcal{R}_0 = \{ \theta_o + \frac{1}{2}\Delta\phi,\ \theta_o + \frac{1}{2}\phi \}$ is the range from the first point, point 0, as can be verified in Fig. 4.2. In general,

$$\mathcal{R}_k = \left\{ \theta_o + (k+\tfrac{1}{2})\Delta\phi,\ \theta_o + \tfrac{1}{2}\phi + \tfrac{1}{2}k\Delta\phi \right\} \tag{4.2.3}$$

for all k = 0 to (n-1), as can be verified in Fig. 4.3. Note that the range from point k = n-1 is a single value of $\theta_o + \phi - \frac{1}{2}\Delta\phi$, and $\phi = n\Delta\phi$, thus, the range of point k = n is not defined. We also note that the left limit of the ranges moves to the right by $\Delta\phi$ as the right limits move by $\frac{1}{2}\Delta\phi$ for each consecutive pair of ranges. Thus the width of the ranges decreases as in the sequence:

$$\left(\frac{\phi}{2} - \frac{\Delta\phi}{2},\ \ldots,\ \frac{\phi}{2} - \frac{k+1}{2}\Delta\phi,\ \ldots,\ \frac{\Delta\phi}{2},\ 0 \right)$$

and the entire range for the θ values is:

$$\mathcal{R} = \left(\theta_o + \frac{\Delta\phi}{2},\ \theta_o + \phi - \frac{\Delta\phi}{2} \right) \tag{4.2.4}$$

where θ_o is as defined in Eqn. 4.2.1. As $\Delta s' \rightarrow 0$, $\Delta\phi \rightarrow 0$, implying that $\mathcal{R} \rightarrow (\theta_o,\ \theta_o + \phi)$ (4.2.5)

Now consider the following example with an arc of a circle of radius r, subtended at the center by an angle of size ϕ. The entire range of the θ values is then $\mathcal{R} = (\theta_o,\ \theta_o + \phi)$ from Eqn. 4.2.5. If this arc was divided into 20 partitions by 21 equally spaced points Δs distance apart, we can represent the ranges for each point, k = 0 to 19 as defined in Eqn. 4.2.3, using the discrete process described, as in Fig. 4.4. If we sum up the number of bars representing the θ ranges over each slot on the

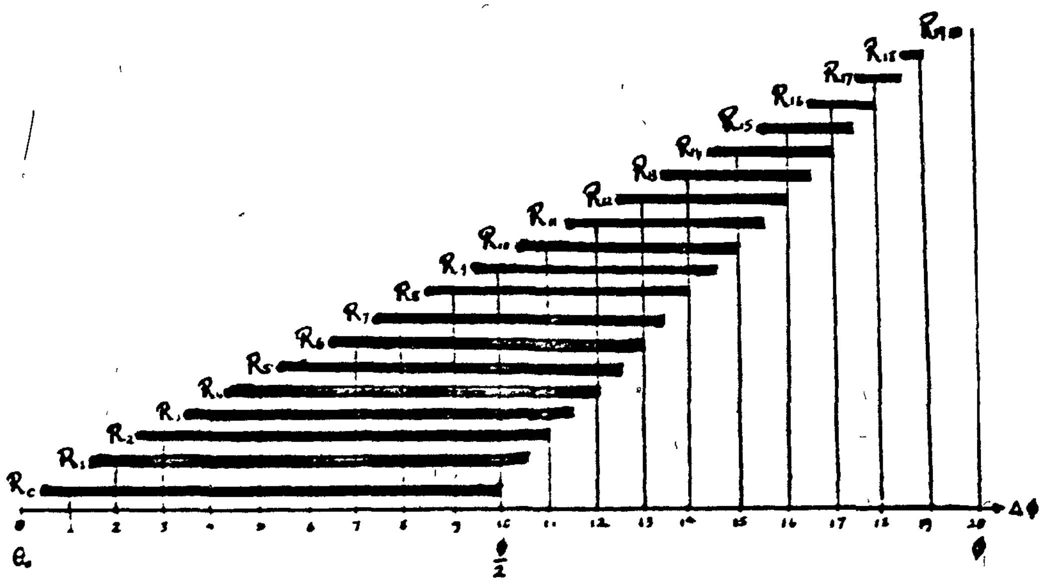
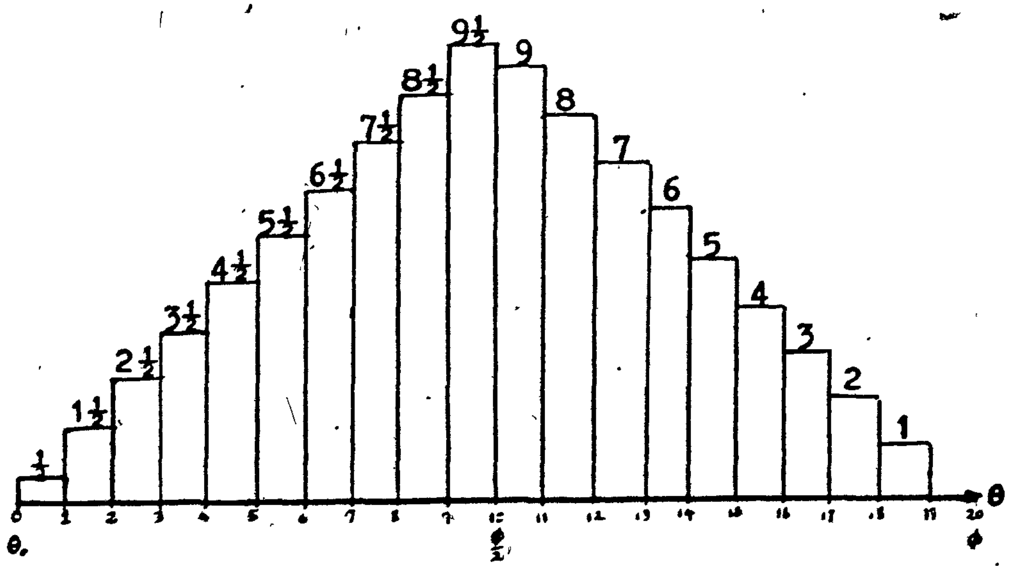


Fig. 4.4 The ranges $\mathcal{R}_i$ are represented by horizontal bars over their domain of definition.



The θ Histogram over the entire range

$\mathcal{R} = (\theta_0 , \theta_0 + \phi)$

x-axis, we obtain a model histogram showing the frequency distribution of θ over the entire range. We note that if $\Delta s \rightarrow 0$, the histogram will approach a continuous symmetrical shape over the entire range $\mathcal{R}$.

With a symmetrically shaped distribution, the mean, or the expected value of θ, will be at the peak of the histogram, i.e. $\theta \rightarrow \theta_0 + \frac{1}{2}\emptyset$ as $\Delta s \rightarrow 0$. We note that this is actually the θ value of the tangent at the mid-point of the arc; see Fig. 4.1.

4.2.1 The Probability Distribution

We generalize the above example by taking s partitions on the arc, instead of 20, giving us s slots on the histogram. In the limiting case when $s \rightarrow \infty$, the probability at the center slot approaches $2/\emptyset$, the maximum height of the density function. Using this as our guideline, with $\frac{1}{2}s$ slots on each side of the histogram's center, the value of each slot has to differ by $\delta = \frac{4}{s\emptyset}$ units. Thus, the generalized slot counts are: (for s even),

$$c_i = \begin{cases} (i-\frac{1}{2}) * \delta, & i = 1, \ldots \ldots, \frac{1}{2}s \\ \frac{2}{\emptyset} - (i-\frac{s}{2} - 1) * \delta, & i = \frac{1}{2}s+1, \ldots \ldots, s \end{cases} \tag{4.2.6}$$

We normalize these counts by their sum $\sum_i c_i = \frac{s+1}{\emptyset}$ to obtain their discrete probability:

$$p_i = \begin{cases} \frac{4i-2}{s(s+1)} & i = 1, \ldots \ldots, \frac{1}{2}s \\ \frac{4(s+1-i)}{s(s+1)} & i = \frac{1}{2}s+1, \ldots \ldots, s \end{cases} \tag{4.2.7}$$

4.2.2 The Estimate of the Mean

The expected value of this frequency distribution is our estimate of the tangent angle. We evaluate, for any value of s -- the number of partitions using values of p_i from Eqn. 4.2.7,

$$\sum_{i=1}^{s} p_i * \frac{i}{s} \emptyset = \frac{\emptyset}{4} * \frac{2s+1}{s} * \frac{s+2}{s+1} .$$

Taking the limit as $s \rightarrow \infty$, we get: $= \frac{\emptyset}{2}$. This implies that the estimate of of the tangent angle approaches the midpoint of the angular range as the number of slots increases. Thus, the estimate is unbiased.

4.2.3 The Entropy Value

The entropy value of this frequency distribution is our estimate of the curvature of the arc under consideration, that is, $-\sum_{i=1}^{s} p_i * \log p_i$. Substituting the p_i's from values in Eqn. 4.2.7, into the positive component of the entropy, yields, after much algebra:

$$= \frac{s}{2(s+1)} \log\left(\frac{s}{s(s+1)} \right) + \frac{s+2}{2(s+1)} \log\left(\frac{4}{s(s+1)} \right)$$

$$+ \frac{2}{s(s+1)} \sum_{i=1}^{s/2} (2i-1) * \log (2i-1)$$

$$+ \frac{4}{s(s+1)} \sum_{i=\frac{s}{2}+1}^{s} (s+1-i) * \log(s+1-i)$$

$$= \frac{2}{s+1} \log s + \log\left(\frac{s}{s(s+1)} \right) + \frac{2}{s(s+1)} \sum_{i=1}^{s-1} i * \log i$$

Thus, the estimate of the curvature is the negative value of the above expression.

4.3 Continuous Case

Although the primary aim, as the title of the thesis suggests, is the measurement of curvature on a quantized grid, i.e. the discrete case, in this section, we would like to gain insight into the method from the theoretical point of view. One way to do this is to have a continuous model. A continuous model assumes that the quantization of the grid is extremely fine, or alternatively, that a particular segment of the contour of a pattern essentially consists of an infinite number of points.

In the discrete case, the tangent angle estimate is the average value of Θ obtained from the Θ- histogram. In the continuous model, the problem of measuring the tangent angle becomes: given an arc $\widehat{ab}$, what is the expected value of Θ when two points are chosen at random on the arc to produce a random segment. As before, the analysis below is based on a circle for simplicity.

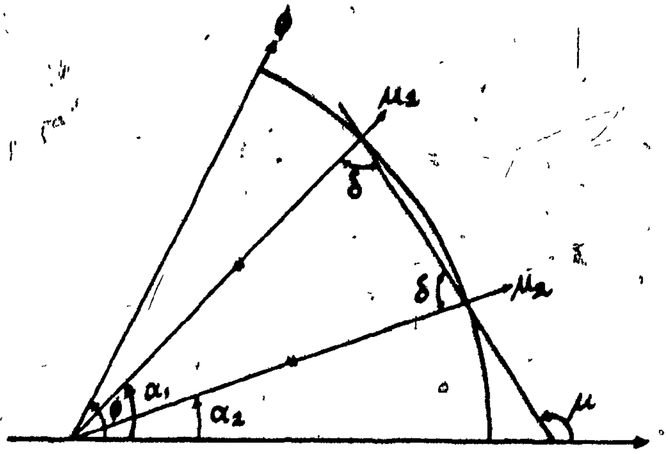


Fig. 4.5 To determine the probability distribution of μ when α_1 and α_2 are uniform random variables.

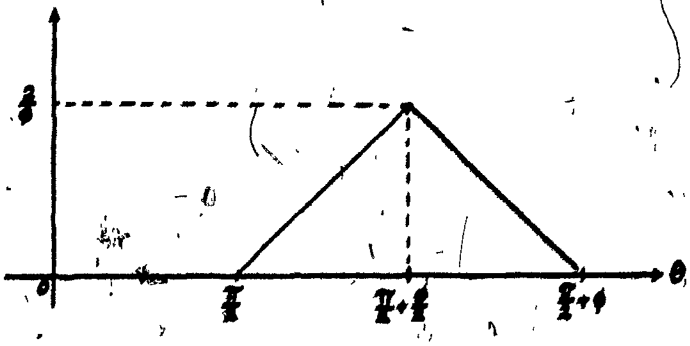


Fig. 4.6 Probability Density Function of $(\frac{\mu_1+\mu_2}{2}+\frac{\pi}{2})$ where $\mu_1, \mu_2 \in (0,\emptyset)$ are two uniform random variables.

4.3.1 A Continuous Model

We want to know the probability density function of the radial angle from a line-point between two randomly placed points on the arc of a sector of a circle subtended by an angle of size $\emptyset$. With this, one would know the frequency of line-points for any radial angle Θ. In our considerations, we let α as defined in Eqn. 4.2.1 be zero to make things easier to manipulate, thus understand.

Let α_1 and α_2 be two uniformly distributed angle values between 0 and $\emptyset$ radians. With reference to Fig. 4.5, we have:

$$\delta = \tfrac{1}{2}\{\pi - |\alpha_1 - \alpha_2|\}, \quad \text{then by simple geometry,}$$

$$\begin{aligned}\mu &= \delta + \max(\alpha_1, \alpha_2) \\ &= \frac{\pi}{2} - \frac{|\alpha_1 - \alpha_2|}{2} + \max(\alpha_1, \alpha_2)\end{aligned}$$

If we let $\mu_1 = \max(\alpha_1, \alpha_2)$ and $\mu_2 = \min(\alpha_1, \alpha_2)$, then μ_1 and μ_2 are two uniform random variables within the range of (0 , $\emptyset$). Again, by simple geometry, we have:

$$\begin{aligned}\mu &= \frac{\pi}{2} - \frac{\mu_1 - \mu_2}{2} + \mu_1 \\ &= \frac{\pi}{2} + \frac{\mu_1 + \mu_2}{2}\end{aligned}$$

Hence, we have that the probability distribution of the variable μ is the distribution of $(\frac{1}{2}\{\mu_1 + \mu_2\} + \pi/2)$, where $\pi/2$ is a constant, affecting the density function by a shift or displacement of that value on the Θ axis.

We know what the distribution of the sum of two uniforms each in the range of $(0,\emptyset)$ would look like (53), thus, we have the probability density function of half the sum of two uniforms, each being in this range, in the range of $(0,\emptyset)$ as shown in Fig. 4.6.

4.3.2 The Probability Density Function

To determine the density function, we have, for any given $\emptyset$, $f_\emptyset(\Theta) \geq 0$, $\int_{-\infty}^{\infty} f_\emptyset(\Theta)\, d\Theta = 1$, and the general equation:

$$f_\emptyset(\Theta) = \begin{cases} c\,\Theta & 0 \leq \Theta \leq \tfrac{1}{2}\emptyset \\ c(\emptyset - \Theta) & \tfrac{1}{2}\emptyset < \Theta \leq \emptyset \\ 0 & \text{elsewhere, where } c \text{ is the slope.} \end{cases} \qquad (4.3.1)$$

Geometrically, the height of this function, by symmetry of the two equal size triangles on both sides of $\frac{1}{2}\emptyset$, is obtained by solving for the area equation: $\frac{1}{2}$ base x height = 1. With the base = $\emptyset$, the height = $2/\emptyset$, the slope = height/$\frac{1}{2}$*base, is $4/\emptyset^2$. That is:

$$c = \frac{4}{\emptyset^2} \tag{4.3.2}$$

Analytically, we confirm this by solving for the system: $\int_{-\infty}^{\infty} f_\phi(\Theta)\, d\Theta = 1$. That is, $\int_0^{\frac{\phi}{2}} c\ \Theta\ d\Theta + \int_{\frac{\phi}{2}}^{\phi} c\ (\emptyset-\Theta)\ d\Theta = 1$.

Taking the integrals, $\frac{1}{2}c\ \Theta^2\Big|_0^{\frac{\phi}{2}} + c\ \Theta(\emptyset-\frac{1}{2}\Theta)\Big|_{\frac{\phi}{2}}^{\phi} = 1$

Substituting, we have $(c\ \emptyset^2)/4 = 1$

implying that $c = 4/\emptyset^2$, and the height is the value of the function $f_\phi(\Theta)$ at $\Theta = \emptyset/2$, and is equals to $2/\emptyset$.

The distribution function $D(\Theta) = \int_0^{\phi} f_\phi(\Theta)\, d\Theta$ is worked out to be:

$$D(\Theta) = \begin{cases} 2*(\frac{\Theta}{\emptyset})^2 & \text{for } 0 \leq \Theta \leq \frac{1}{2}\emptyset \\ 1 - 2*(1-\frac{\Theta}{\emptyset})^2 & \text{for } \frac{1}{2}\emptyset < \Theta \leq \emptyset \end{cases} \tag{4.3.3}$$

It is interesting to note that the density of Fig. 4.6 is indeed the continuous version of the histogram of Fig. 4.4 for the discrete case.

4.3.3 Calculation of the Mean

Given the probability density function from Eqn. 4.3.1, we calculate the expected value as $\int_{-\infty}^{\infty} \Theta\ f_\phi(\Theta)\, d\Theta$. That is,

$= \int_0^{\frac{\phi}{2}} c\ \Theta^2\ d\Theta + \int_{\frac{\phi}{2}}^{\phi} c\ \Theta\ (\emptyset-\Theta)\ d\Theta$. Taking the integrals,

$= \frac{c}{3}\Theta^3\Big|_0^{\frac{\phi}{2}} + c\ \Theta^2\ (\frac{\emptyset}{2} - \frac{\Theta}{3})\Big|_{\frac{\phi}{2}}^{\phi}$. Substituting,

$= \frac{c}{3}*\frac{\emptyset^3}{8} + \frac{c}{12}\emptyset^3$. Adding and substituting for c,

$= \emptyset/2$. Thus the mean is unbiased and is at $\emptyset/2$.

Again, this value agrees with the value obtained in the discrete case.

4.3.4 Calculation of the Entropy

Since this probability density function is symmetric about $\Theta = \frac{1}{2}\emptyset$, we need only evaluate the entropy for the range

$(0, \frac{1}{2}\emptyset)$ and multiply this by two to give us the entropy of the whole density function. That is, entropy $= -2\int_0^{\frac{\emptyset}{2}} f_\phi(\Theta) * \log f_\phi(\Theta)\, d\Theta$. So, substituting $f_\phi(\Theta) = c\,\Theta = 4\Theta/\emptyset^2$ into

$\int_0^{\frac{\emptyset}{2}} f_\phi(\Theta) * \log f_\phi(\Theta)\, d\Theta$, we get, after considerable algebra,

$= \frac{1}{2} (\log 2 - \log \emptyset - \frac{1}{2})$. Thus, the entropy indicating curvature is:

$$K = \frac{1}{2} + \log (\emptyset/2) \tag{4.3.4}$$

4.4 The Theoretical Discrete Case

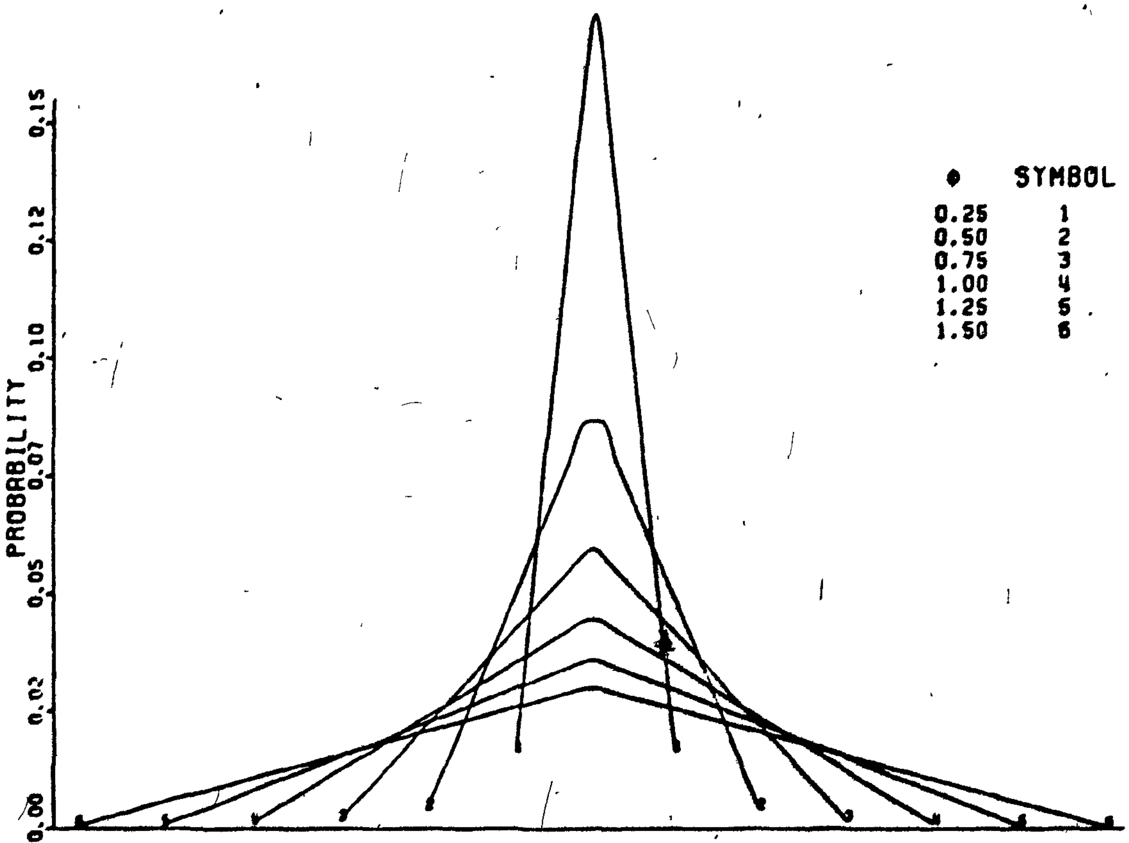
Since the distribution of the angles of lines formed between two uniformly random points on the arc between 0 and $\emptyset$ radians is evidently the same as the limiting case of the frequency distribution of line-point angles, the process of taking line-point angles is equivalent in some way to taking angles of lines between two randomly selected points on the arc. Thus we can impose a discrete framework on the results of the continuous model to investigate the theoretical discrete case. Furthermore, in addition to being able to relate the curvature measure (Eqn. 4.3.4) with the classical measure in the continuous domain, we can also relate the curvature measure of the theoretical discrete case to the classical curvature measure.

We start by partitioning the distribution function of Eqn. 4.3.3 into n partitions to give us a frequency distribution of the angles, each value $p_i = D(x_i) - D(x_{i-1})$, where $x_0 = 0$. The resulting histogram would have the values: (for n even),

$$p_i = \begin{cases} \frac{2}{n} * \frac{2i-1}{n} & \text{for } i=1, \ldots \ldots, \frac{1}{2}n \\ \frac{2}{n} * \frac{2(n-i)+1}{n} & \text{for } i=\frac{1}{2}n+1, \ldots, n \end{cases} \tag{4.4.1}$$

and for n odd,

$$p_i = \begin{cases} \frac{2}{n} * \frac{2i-1}{n} & \text{for } i=1, \ldots \ldots, \frac{n-1}{2} \\ \frac{2n-1}{n} & \text{for } i= \frac{n+1}{2} \\ \frac{2}{n} \quad \frac{2(n-i)+1}{n} & \text{for } i=\frac{n+3}{2}, \ldots, n \end{cases} \tag{4.4.2}$$



HISTOGRAM OF ANGULAR FREQUENCY DISTRIBUTION
FOR SOME VALUES OF Φ

Fig. 4.7

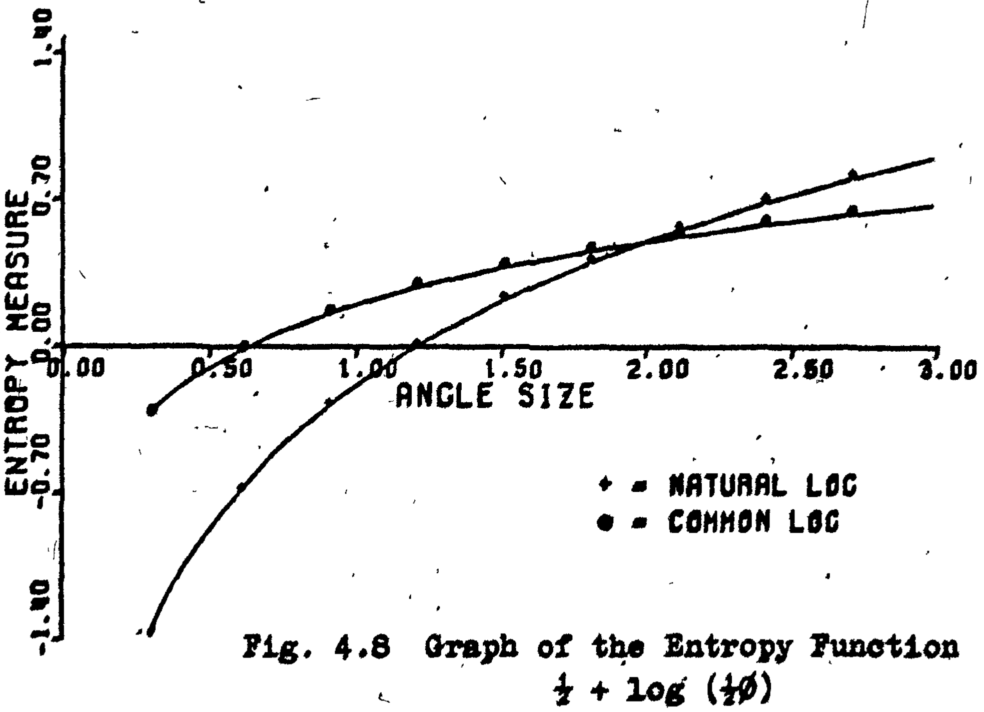


Fig. 4.8 Graph of the Entropy Function

$\frac{1}{2} + \log (\frac{1}{2}\phi)$

If we take these values of p_i to evaluate the entropy, we would get, after much algebraic manipulation:

$$-K = \sum_{i=1}^{n} p_i \log p_i = \sum_{i=1}^{n/2} \frac{2}{n} * \frac{2i-1}{n} \log \left(\frac{2}{n} * \frac{2i-1}{n}\right) + \left[\frac{2n-1}{n^2} \log \left(\frac{2n-1}{n^2}\right)\right]_{n\ odd}$$

$$+ \sum_{i=\frac{n}{2}+1}^{n} \frac{2(2(n-i)+1)}{n^2} \log \left(\frac{2(2(n-i)+1)}{n^2}\right)$$

$$\xi(n) = \frac{2}{n} \log 2 - \log n^2 + \frac{4}{n^2} * \left\{\sum_{k=1}^{n-1} k \log k - 2* \sum_{k=1}^{n/2} k \log k\right\}$$

for n being even (4.4.3)

$$\xi(n) = \frac{2(n-1)}{n^2} \log 2 - \log n^2 + \left(\frac{2n-1}{n}\right) \log (2n-1)$$

$$+ \frac{4}{n^2} * \left\{\sum_{k=1}^{n-1} k \log k - 2* \sum_{k=1}^{n-1} k \log k\right\}$$ for n odd (4.4.4)

Thus the entropy = $-\xi(n)$.

Digressing a little, we note that when using the values of p_i from Eqn. 4.4.1 and Eqn. 4.4.2, the expected value of these frequencies for a given Ø is:

$$\sum_{i=1}^{n} p_i * \frac{i}{n} Ø = \sum_{i} \frac{2(2i-1)}{n^2} * \frac{i}{n} Ø + \left[\frac{2n-1}{n^2} * \frac{n+1}{2n} Ø\right]_{n\ odd}$$

$$+ \sum_{i} \frac{2(2n+1-2i)}{n^2} * \frac{i}{n} Ø$$

After much algebraic manipulation, we get:

$$= \frac{Ø}{2} * \frac{n+1}{n}$$ for n even, and

$$= \frac{Ø}{2} + \frac{1}{n}$$, for n odd.

We note that they are equivalent and the limit as n → ∞ is Ø/2. which is the unbiased estimate of the tangent angle.

An illustration of some frequency distributions for some values of Ø can be seen in Fig. 4.7. Here, one can observe what happens to the density function as Ø increases. This in effect produces the differences in the entropy measures to show the differences in their curvature values.

Getting back to the entropy value of the continuous model obtained in Eqn. 4.3.4, one sees in Fig. 4.8 that the graph of the entropy function $\frac{1}{2} + \log(\frac{1}{2}Ø)$ is negative for some values of $Ø < Ø_0$. Realizing that the entropy of the histogram indicating curvature

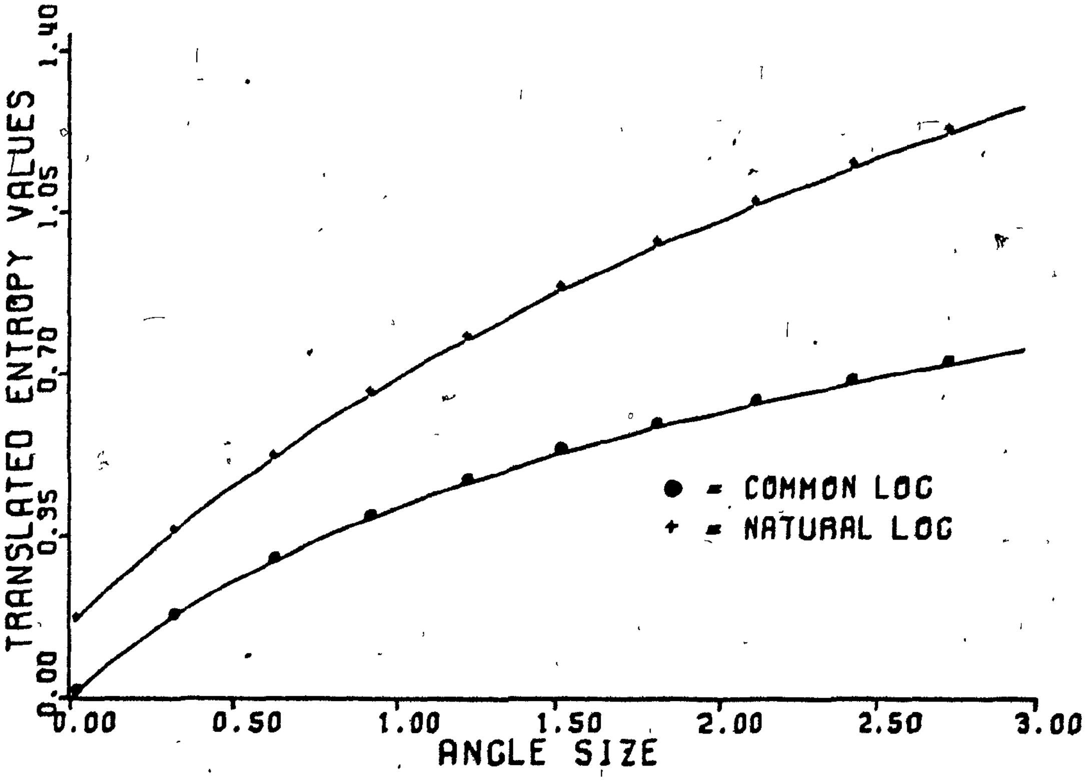


Fig. 4.9 Graph of the Translated Entropy

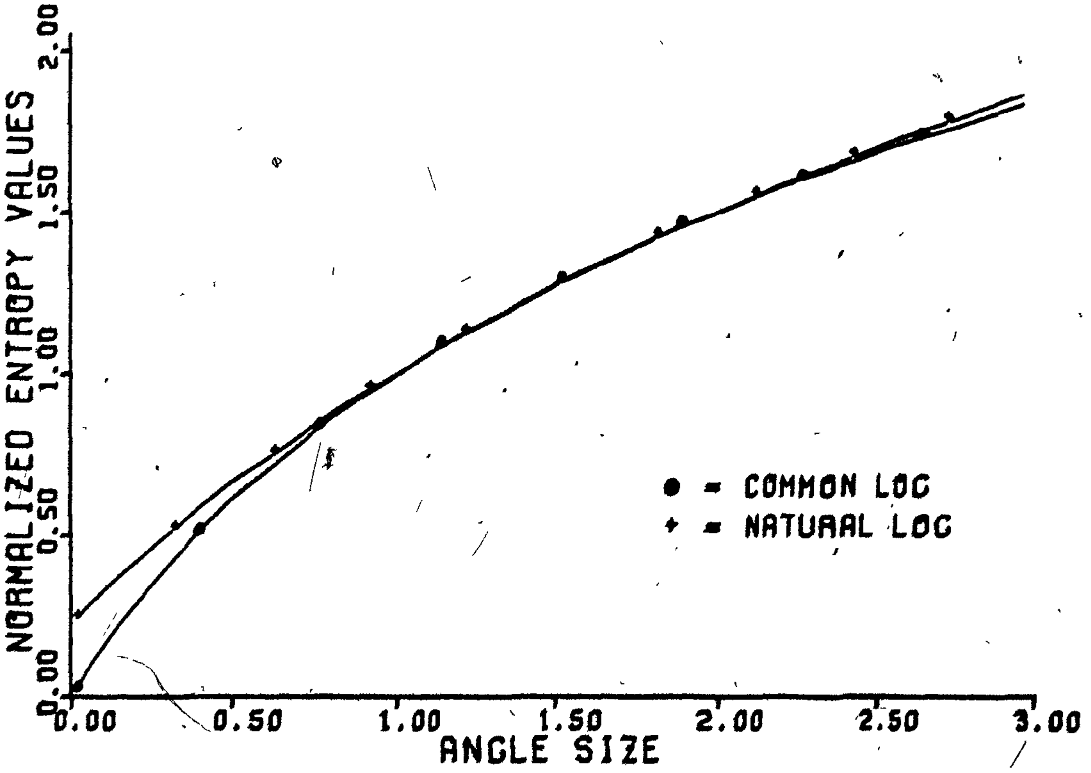


Fig. 4.10 Graph of the Normalized Entropy

is always non-negative, we suggest a translation of the graph by ϕ_0 to give a non-negative function of the entropy. To determine the value of ϕ_0, one simply solves for its root, i.e. solve for $\frac{1}{2} + \log(\frac{1}{2}\phi) = 0$, or $\log \phi = \log 2 - \frac{1}{2}$. Thus, we obtain $\phi_0 = 2/\sqrt{b}$ where b is the base of the logarithms. Now we can equate the curvature to non-negative values of the entropy by the equation:

$$\kappa_t(\phi) = \frac{1}{2} + \log\left(\frac{1}{2}(\phi+\phi_0)\right) \quad \text{(Translated Entropy)} \qquad (4.4.5)$$

indicating that the curvature for $\phi = 0$ is 0. As such, we obtain the graph as illustrated in Fig. 4.9. With this, we still do not have the curvature of $\phi = 1$ as one which is the case for circles. We can obtain this by normalizing the values of the entropy by its value at $\phi = 1$. Thus, the normalized values of entropy indicating its curvature is: (Normalized Entropy)

$$\kappa_n(\phi) = \frac{\frac{1}{2} + \log\left(\frac{1}{2}(\phi+\phi_0)\right)}{\frac{1}{2} + \log\left(\frac{1}{2}(1+\phi_0)\right)} = \frac{\kappa_t(\phi)}{\kappa_t(1)} \qquad (4.4.6)$$

The graph of the normalized values of entropy can be observed in Fig. 4.10.

Now let us consider the entropy values obtained in Eqn. 4.4.3 and 4.4.4 by using the histogram frequencies of Eqn. 4.4.1 and Eqn. 4.4.2. We note that this entropy expression is independent of ϕ. At first this may seem inconsistent after knowing that the entropy measures indicating curvature are directly related to the size of ϕ. But we will see in the next section that for an initial (arbitrary) value of ϕ, a decision will be made as to how many slots in the histogram--partitions of the density function, are to be used for this value of ϕ. Thus, for all subsequent values of ϕ encountered, this same sized partition will be used. This implies that the number of slots directly reflects on the size of ϕ. Since the expressions in Eqn. 4.4.3 and 4.4.4 are functions of the number of partitions, n, it indicates that it is related to the size of the angle ϕ after all.

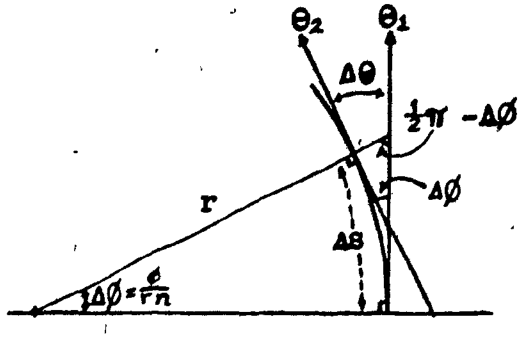


Fig. 4.11 The rate the angle of the tangent to the arc changes (Δ θ) is uniform for fixed size changes in the arc length (Δ s).

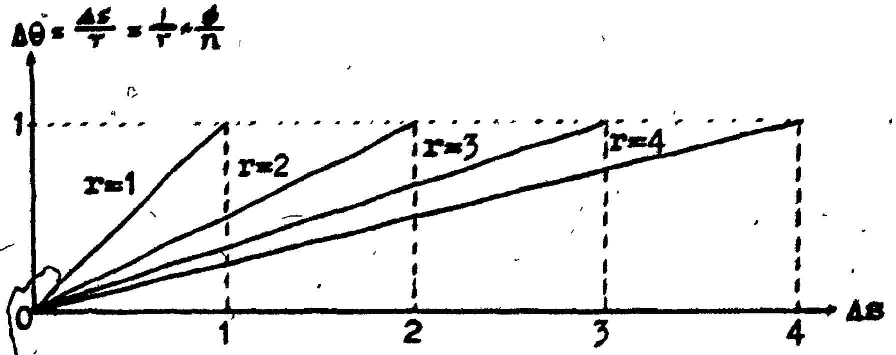


Fig. 4.12 The difference in the rate of change of Θ with respect to arc length is the size of the radius of the arc, r.

4.5 Entropy Measure versus Classical Measure of Curvature

Classically, curvature is defined as the rate of change of the angle Θ with respect to its arc length; see Section 1.2. So, we want to see in a discrete way, the graph of the rate of change of Θ as a function of the arc length s.

Consider a sector of radius $r = 1$. We divide this arc into n equal size sectors each having an arc length of Δs, and subtended by an angle of $\Delta\phi = \phi/n$ as illustrated in Fig. 4.11. Since $r = 1$, $\Delta s = \Delta\phi = \phi/n$, and the difference in the angle between vectors Θ_1 and Θ_2 is $\Delta\Theta = \Delta\phi = \phi/n$, easily verified by simple geometry - using the fact that the sum of the angles of a triangle is π radians. Thus we see that $\Delta\Theta = \Delta\phi = \Delta s = \phi/n$, implying that the change in Θ ($\Delta\Theta$), is uniform for equal sized changes in s (Δs). This is represented in Fig. 4.12 by the r = 1 graph.

In general, we consider the case for any integer value of $r > 1$. Using the same size Δs as before, we now partition the arc into $(n*r)$ equal size sectors, each sector now being subtended by an angle of $\Delta\phi = \Delta s/r = \phi/(n*r)$. Similarly, the difference in the angles of the two vectors Θ_1 and Θ_2 is $\Delta\Theta = \Delta\phi = \phi/(n*r)$ like as before. Thus the graph showing the uniform change of Θ for equal size changes in s can be represented as in Fig. 4.12. for various values of r.

In essence, the change in Θ, $\Delta\Theta = \Delta s/r = \phi/(n*r)$, when differentiated with respect to its arc length, Δs, gives us the rate of the change of Θ with respect to the arc length, i.e. the classical curvature function. That is, $d\Theta/ds = 1/r$, which is the classical curvature function for a circle.

To elaborate briefly on our concept of curvature measure, let us consider a model. As a basis for reference, consider a one radian sector of a unit circle. Classically, the curvature for $\phi = 1$, $r = 1$ is one at any point on the arc regardless of the size of the arc length taken into consideration, and for any other values of r extending from this one radian sector, the curvature $\kappa = 1/r$. With reference to Fig. 4.13, we present the model.

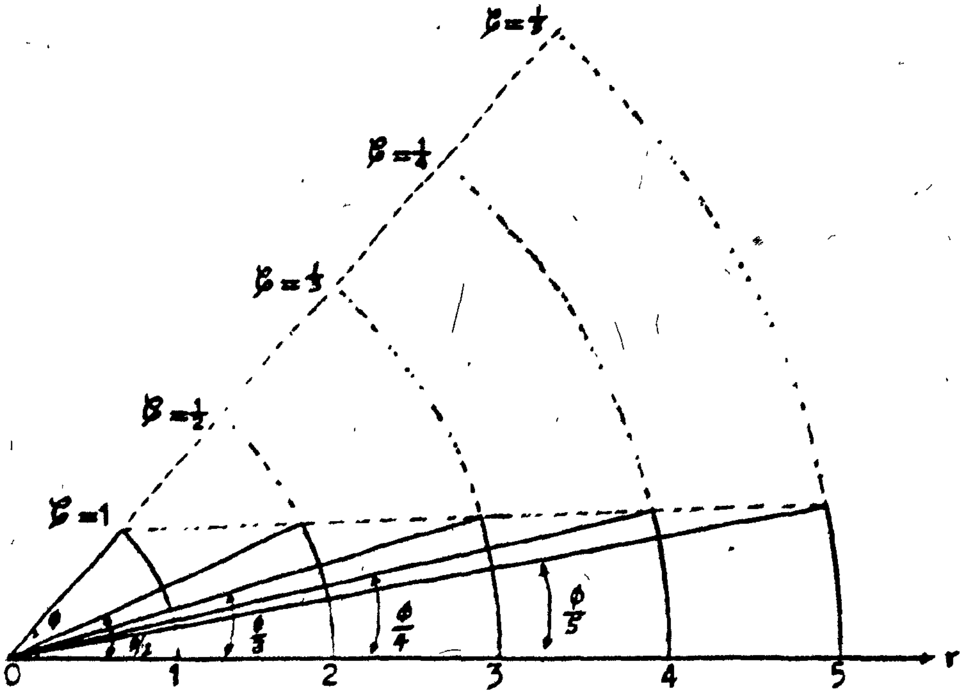


Fig. 4.13 Classically, the curvature of the arcs, being 1/radius, show a decreasing trend as the radius increases. In using the proposed method, only a fixed sized segment of each arc is considered, whereby the decreasing values of the subtending angles would result in a corresponding decrease in the entropy measure to correlate with the classical measures.

We set the curvature of a $\phi = 1$, $r = 1$ arc to one by appropriate normalization of the entropy, like as was done in Section 4.4 for the discretized continuous model. Then for any other value of r, we use the same size arc for our evaluations. As r increases, the angles of the fixed arc-length sectors (ϕ) successively decrease. Subsequently, there is a corresponding decrease in the entropy measure. Thus we see a correlation between the entropy and classical measure of curvature.

This sets the importance of using a fixed length segment as the basis for measuring curvature in the proposed method; a fixed size neighbourhood of the point under consideration is used for our evaluations. Having established this, we now seek to determine a suitable size of Δs which will remain fixed for all cases of ϕ being considered. Analogously, the neighbourhood size is to remain constant for all evaluations done in the quantized case. Note that all calculations done in the proposed method have been made independently of the size of the radius, and the entropy, our curvature estimate, remained solely a function of the size of the angle ϕ.

The constraint which Δs has to satisfy is that n of them on a one radian sector would give a frequency histogram that has an entropy of one or close to it. That is, we solve for the p_i's in the equation: $-\sum_{i=1}^{n} p_i * \log p_i = 1$, p_i as defined in Eqn. 4.4.1 and Eqn. 4.4.2. When using common logarithms, we get $n = 12$ for a unit ϕ to give an entropy value of 0.9994. We note that this is determined independently of the actual size of ϕ or the radius r. We also note that when using our proposed method, the curvature estimates of all the complete arcs in Fig. 4.13 would be equal since they all give identical histograms. This would come about if each arc is divided into n (or 12) equal sized segments, each subtended by an angle of ϕ/n regardless of the size of the radius. With an equal number of angular sectors, the histograms obtained, and indeed the resulting entropy values would be equal. Thus we would want to establish norms for the values of ϕ and Δs from which all other measures can be normalized using the resulting entropy value obtained from this set of norms.

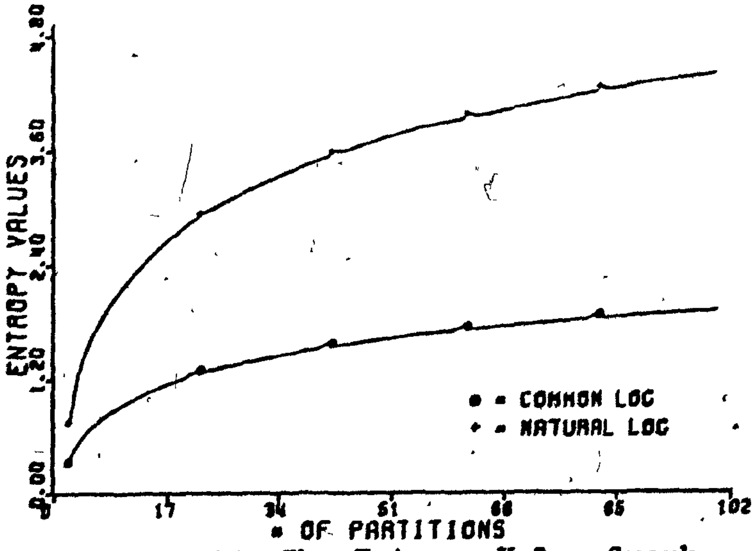


Fig. 4.14 The Entropy Value Graph

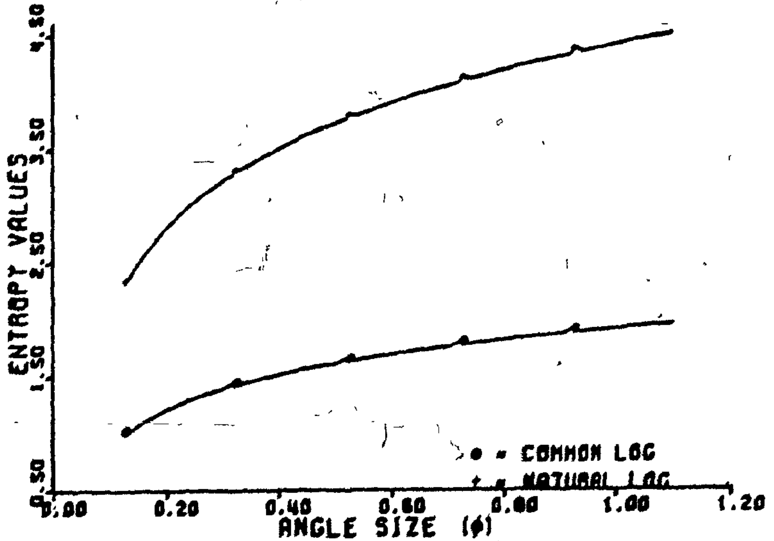


Fig. 4.15 The Entropy Value Graph

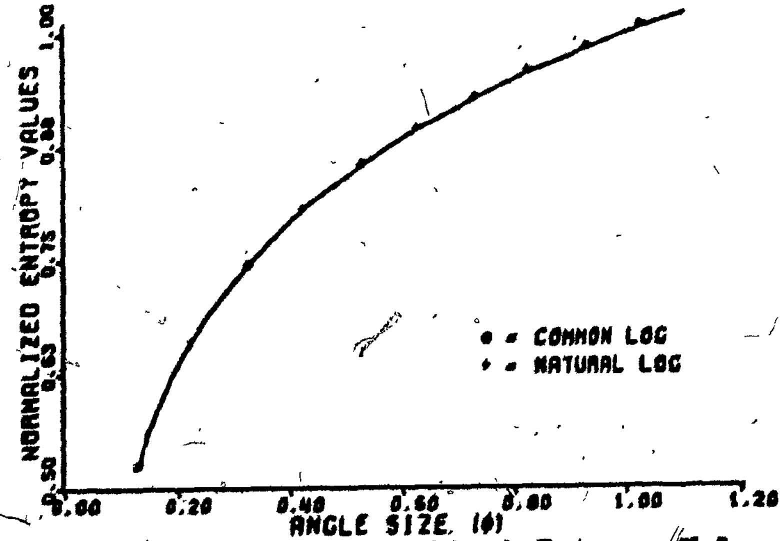


Fig. 4.16 The Normalized Entropy Value Graph

We can choose Δs with respect to $\phi = 1$ radian and with $\Delta\phi = \phi/12$. We would then be able to partition any given angle ϕ' into $(\phi'/\phi)*12$ equal sized angular sectors, and for varying sizes of ϕ', the corresponding variations in the number of angular sectors would give entropy measures of the histograms that depict their curvature as a non-linear function as can be seen in Fig. 4.14. So, in fact, we are using the entropy value of the θ-distribution histogram for some ϕ' with reference to an initial ϕ which we have established as the norm. Fig. 4.15 shows the entropy values for various angle sizes. And after normalization by the value of the entropy when $\phi' = 1$, so that we can get a curvature indicating one when the size of the angle is actually one radian, we note that the two curves in Fig. 4.16, obtained when using common and natural logarithms coincide. It can be shown that regardless of the base of the logarithms used, normalized entropy values will always be equivalent.

Now that we have seen the theoretical equivalence of this method, we want to know what these results imply when they are related to the actual application of the method. The range of the θ definition is from 0 to π. If NS = n $\geq$ 2 is used, then $\lceil \frac{n}{\pi} \rceil$ partitions would enclose an angle of one radian. We can thus set the entropy of a histogram with $\lceil \frac{n}{\pi} \rceil$ cells as the value of the norm, i.e. equivalent to a curvature measure of one. Thus, we can relate the proportional factor that associates entropy values and curvature in terms of n, i,e.

$$\text{Curvature} \propto \frac{\text{Entropy from } n' \ (\leq n) \text{ slots}}{\text{Entropy from } \lceil \frac{n}{\pi} \rceil \text{ slots}} \qquad (4.5.1)$$

Obviously, different values of n would give different values of the norm. Using the result that the entropy is one for n = 12 when using common logarithms, each histogram cell would actually represent 1/12 (or 0.0833) radians. Thus NS should be $12*\pi$ in order that the one radian sector would give an entropy measure that is close to its curvature value of one. With this, the theoretical number of angular partitions (NS) would be 37.699. A larger sized partition and an odd number of them is ideal, so 37 would satisfy both ideals.

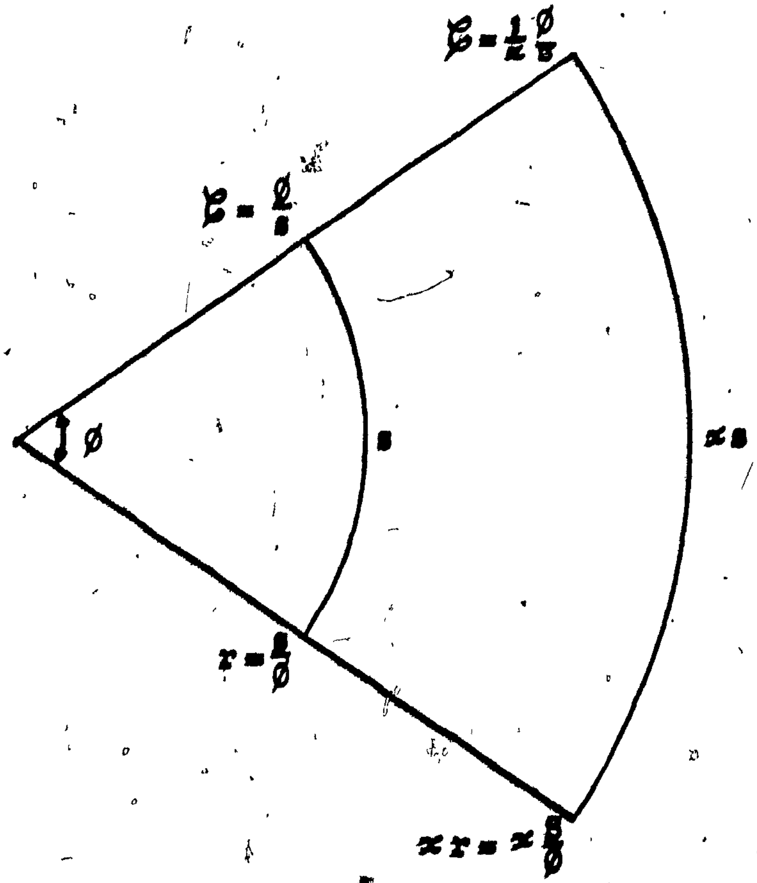

$\mathcal{C} = \frac{1}{x}\frac{\emptyset}{s}$
$\mathcal{C} = \frac{\emptyset}{s}$
$\emptyset$
s
xs
$r = \frac{s}{\emptyset}$
$xr = x\frac{s}{\emptyset}$

Fig. 4.17 Classically, the curvature of the two arcs have different values, whereas the entropy evaluated on the size of the subtending angle Ø would give the same value for both arcs.

It is also of interest to note that when the sector is partitioned into 12 equal sectors, we equivalently get an arc length of 12 units which means 13 points on the arc is required. This implies a neighbourhood size of 6 which is what we can consider as the theoretical neighbourhood size (NB).

4.6 General Relationships Between the Proposed and the Classical Measure of Curvature

In Section 4.3, we derived an expression for the entropy of the probability density function obtained from a continuous model of our proposed method. Eqn. 4.3.4 gives us:

$$\mathcal{K}(\phi) = \tfrac{1}{2} + \log(\tfrac{1}{2}\phi) \qquad (4.6.1)$$

We can easily relate this result implied by the continuous model of the proposed method to classical curvature.

Consider an angle sector, radius r, subtended by an angle ϕ with an arc length s. The classical curvature $\kappa = 1/r = \phi/s$, thus $\phi = s\kappa$. Substituting this equivalence relation into Eqn. 4.6.1, we get:

$$\mathcal{K}(\phi) = \tfrac{1}{2} + \log\left(\frac{s\kappa}{2}\right) \qquad (4.6.2)$$

This shows an exact relationship between the entropy measure of the continuous case and classical curvature.

On second thought, we realize that the entropy function in the continuous case at times had negative values which do not depict the discrete histogram's entropy consistently, i.e. they had to be transformed and normalized before they showed any direct resemblance to histogram entropies. With this in mind, we want to establish some general relationships to classical curvature by using results of the theoretical discrete model of Section 4.4 which best describes our proposed method.

There are two differences between the classical and entropy measure of curvature.

1. In Fig. 4.17, we consider the case of an angle ϕ having different size arcs. Classically, the curvature is taken as the reciprocal of the radius; for any size arc length s, the curvature $\kappa(s) = \phi/s$. For any other arc of length $s' = x s$,

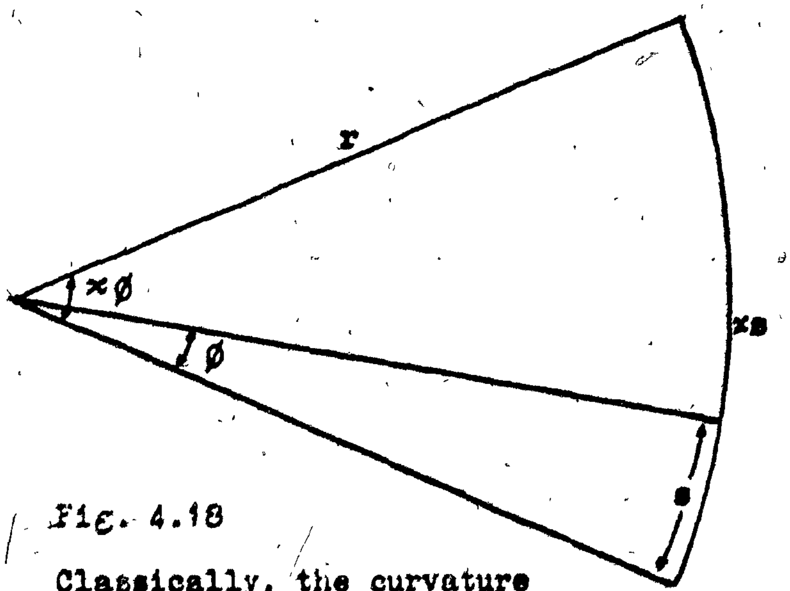


Fig. 4.18

Classically, the curvature of both arcs is the same, whereas the entropy values evaluated on the size of their subtending angles would be different.

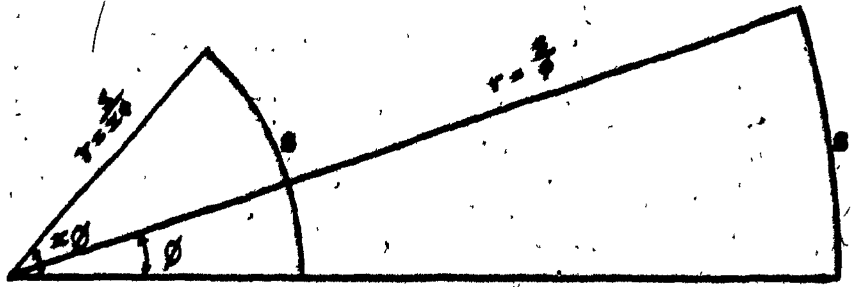


Fig. 4.19 In this case, the values of the classical curvature and the entropy measure correlate positively.

the curvature $\mathcal{C}(s') = \frac{\phi}{x s} = \frac{1}{x}\mathcal{C}(s)$. We note that if $f(x) = xs$, then the curvature

$$\mathcal{C}(xs) = \mathcal{C}(f(x)) = \frac{1}{x}\mathcal{C}(f(1)) = \frac{1}{x}\mathcal{C}(s)$$

In evaluating the entropy of a histogram from a sector of angle size ϕ divided into n partitions, we would get an entropy value of $\mathcal{H}(\phi) = -\mathcal{B}(n)$, where $\mathcal{B}$ is from Eqn. 4.4.3 or 4.4.4 depending on whether n is even or odd respectively. When normalized by $\mathcal{H}(\phi=1)$, the normalized entropy is:

$$\mathcal{H}_n(\phi) = \frac{\mathcal{B}(n)}{\mathcal{B}(\lceil \frac{n}{\phi} \rceil)} \qquad (4.6.3)$$

So the first difference is that $\mathcal{H}_n(\phi)$ is independent of the size of s and dependent on the size of ϕ, whereas, classical curvature $\mathcal{C}(s)$ is dependent on the size of s or r, the radius.

2. In Fig. 4.18, we consider a fixed sized r with varying sized angles and their proportionately varying arc lengths. Thus, if a sector of angle ϕ has an arc length of s on a circle of radius r, a sector of angle $x\phi$, for x a positive real number, would have an arc length of xs on the same circle, because $\phi = s/r$. Classically, the curvature of both arcs is $1/r$.

On the other hand, the normalized entropy measure from a histogram of angle size ϕ having n slots would be like in Eqn. 4.6.3, whereas, from an angle of size $x\phi$ having xn slots, the entropy measure would be:

$$\mathcal{H}_n(x\phi) = \frac{\mathcal{B}(xn)}{\mathcal{B}(\lceil \frac{n}{\phi} \rceil)}$$

Clearly, if $x \neq 1$, $\mathcal{H}_n(\phi)$ and $\mathcal{H}_n(x\phi)$ are not equal, whereas, classical curvature indicates both arcs to have the same curvature. This is the second difference.

Now we want to consider the situation where both entropy and curvature measures behave similarly. In Fig. 4.19, two arcs having the same length s, are subtended by different sized angles, ϕ and $x\phi$, where x is a real positive number. Classically, the curvature of each arc is the reciprocal of their radii, s/ϕ, $s/(x\phi)$.

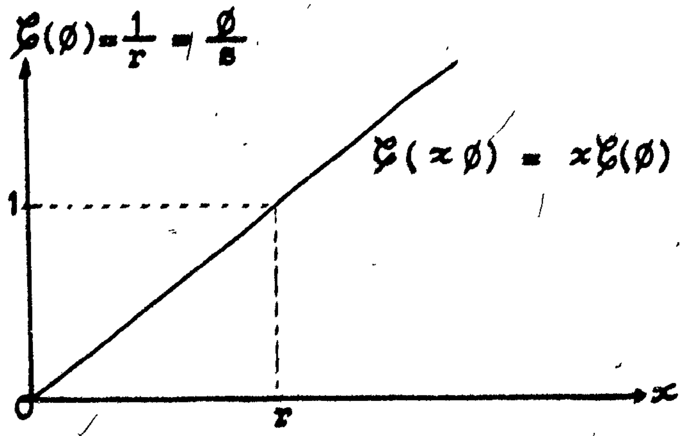


Fig. 4.20 The Graph of the Classical $\mathcal{C}$ as a function of the angle ∅

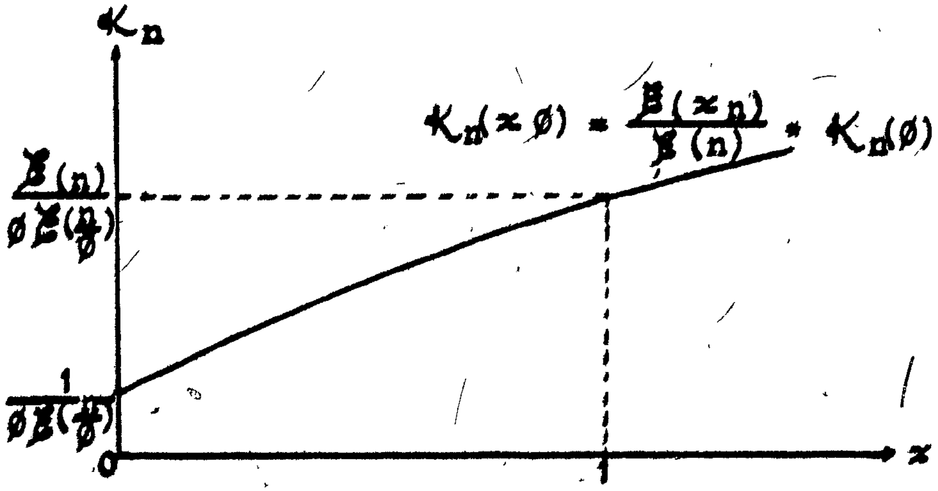


Fig. 4.21 The Graph of the Entropy as a function of the angle ∅

That is, the relationship between $\mathcal{C}(\phi)$ and $\mathcal{C}(\varkappa\phi)$, where $\mathcal{C}(\phi) = 1/r = \phi/s$, is:

$$\mathcal{C}(\varkappa\phi) = \frac{\varkappa\phi}{s} = \varkappa\,\mathcal{C}(\phi).$$

As before, classical curvature proves to be linear; the ratio of the angles is the same as the ratio of their curvatures; $1 : \varkappa$. This can be observed in Fig. 4.20 which shows a graph of the classical curvature as a function of ϕ.

If the histogram entropy for the range ϕ is taken over n slots, we denote it by $\mathcal{K}(\phi) = \mathcal{H}(n)$. When normalized by $\mathcal{K}(\phi{=}1)$, it becomes $\mathcal{K}_n(\phi) = \dfrac{\mathcal{H}(n)}{\mathcal{H}(\lceil \frac{n}{\phi} \rceil)}$ (4.6.4)

From a histogram with a range of $\varkappa\phi$, with the size of the angular partitions (or equivalently, the size of the arc) remaining constant, the number of slots is then $\varkappa n$. Thus the entropy when normalized is $\mathcal{K}_n(\varkappa\phi) = \dfrac{\mathcal{H}(\varkappa n)}{\mathcal{H}(\lceil \frac{n}{\phi} \rceil)}$ (4.6.5)

In this case, a linear increase over the size of ϕ does not result in a linear increase in the entropy value, but in both situations, there exists a corresponding increase if $\varkappa > 1$ or decrease if $\varkappa < 1$.

The relationship between $\mathcal{K}_n(\phi)$ and $\mathcal{K}_n(\varkappa\phi)$ is thus derived to be: (from Eqn. 4.6.4 and Eqn. 4.6.5)

$$\mathcal{K}_n(\varkappa\phi) = \frac{\mathcal{H}(\varkappa n)}{\mathcal{H}(\lceil \frac{n}{\phi} \rceil)} = \frac{\mathcal{H}(\varkappa n)}{\mathcal{H}(n)} \cdot \frac{\mathcal{H}(n)}{\mathcal{H}(\lceil \frac{n}{\phi} \rceil)}$$

$$= \frac{\mathcal{H}(\varkappa n)}{\mathcal{H}(n)} \cdot \mathcal{K}_n(\phi)$$

This relationship can be expressed as a graph as shown in Fig. 4.21.

We can now establish an exact algebraic relationship between classical curvature and entropy measured from a histogram. We denote curvature as $\mathcal{C}(\phi)$ and normalized entropy as $\mathcal{K}_n(\phi)$. For $\varkappa = 1$, we have:

$$\mathcal{C}(\phi) = \frac{\phi}{s} \quad \text{and} \quad \mathcal{K}_n(\phi) = \frac{\mathcal{H}(n)}{\mathcal{H}(\lceil \frac{n}{\phi} \rceil)}$$

Thus we have:

$$\mathscr{C}(\emptyset) = \left(\frac{\emptyset}{s} * \frac{\mathscr{B}(\lceil \frac{n}{\emptyset} \rceil)}{\mathscr{B}(n)} \right) * \mathcal{K}_n(\emptyset) \qquad (4.6.6)$$

or equivalently,

$$\mathcal{K}_n(\emptyset) = \left(\frac{s}{\emptyset} * \frac{\mathscr{B}(n)}{\mathscr{B}(\lceil \frac{n}{\emptyset} \rceil)} \right) * \mathscr{C}(\emptyset) \qquad (4.6.7)$$

In general, for any x, we have:

$$\mathscr{C}(x\emptyset) = x \frac{\emptyset}{s} \quad \text{and} \quad \mathcal{K}_n(x\emptyset) = \frac{\mathscr{B}(xn)}{\mathscr{B}(\lceil \frac{n}{\emptyset} \rceil)}$$

and thus:

$$\mathscr{C}(x\emptyset) = \left(\frac{x\emptyset}{s} * \frac{\mathscr{B}(xn)^2}{\mathscr{B}(n)\,\mathscr{B}(\lceil \frac{n}{\emptyset} \rceil)} \right) * \mathcal{K}_n(\emptyset) \qquad (4.6.8)$$

or equivalently,

$$\mathcal{K}_n(x\emptyset) = \left(\frac{s}{\emptyset} * \frac{\mathscr{B}(xn)}{\mathscr{B}(\lceil \frac{n}{\emptyset} \rceil)} \right) * \mathscr{C}(\emptyset) \qquad (4.6.9)$$

With this relationship established, we have shown that the histogram entropy is theoretically sound as an indicator of curvature. In this context, entropy measures the amount of change in the tangent angle (Θ) to the arc as a function of $\emptyset$. It should be remembered that entropy is solely dependent on the size of the radial (or subtending) angle. It is with this dependence on $\emptyset$ that similarities of this measure can be drawn to classical curvature by using the known fact that $\emptyset = s/r$ for circles.

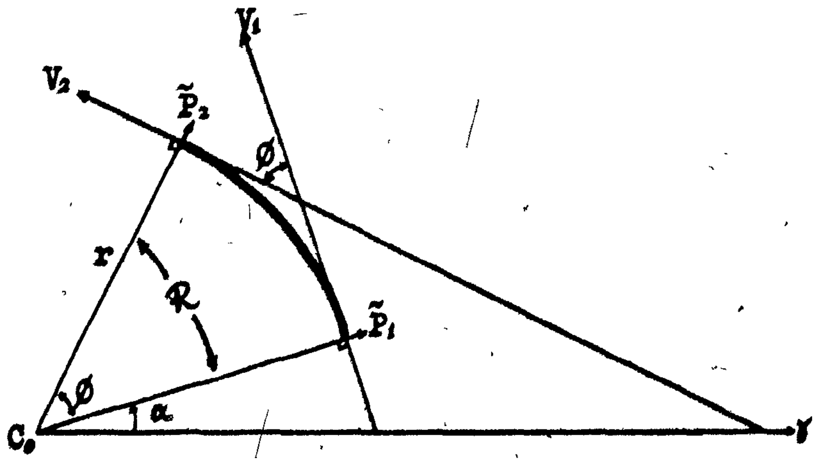


Fig. 4.22 To determine the Θ histogram we first consider the extent of the range.

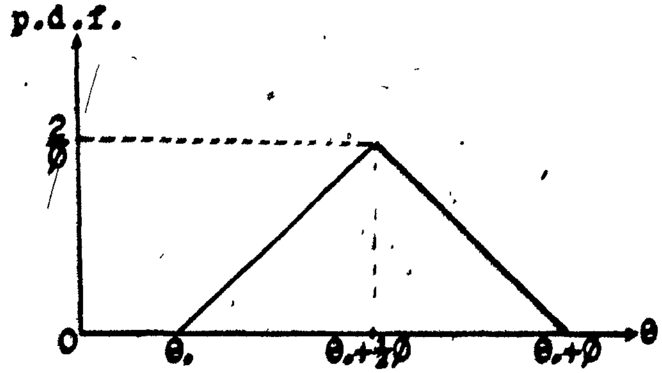


Fig. 4.23 The probability density function using C_0 as the "origin". In terms of radial angles, $\theta_0 = \alpha$, and in terms of line-point angles, $\theta_0 = \frac{1}{2}\pi + \alpha$

4.7 A Relevant Theorem

When we defined the ℓ- θ transformation in Sec. 3.3, reference was made to an "origin" from which the perpendiculars to line-points originate. This theorem states that:

> The θ histogram obtained from a given arc is independent of the position of this origin.

In Fig. 4.22, we consider an arc of a circle of radius r subtended at the center C_0 by an angle of size ϕ. We know from results in Sec. 4.4 that the histogram of θ values obtained while using this center C_0 as the origin looks like the probability density function in Fig. 4.23. The range of the θ values ($\mathcal{R}$) in the histogram is defined to be from the perpendicular ($\tilde{P}_1$) to vector V_1 to the perpendicular ($\tilde{P}_2$) to vector V_2. We note that it is equivalent to the difference in angle between vector V_1 and V_2, i.e. the range of line-point angles is also of size ϕ.

First we want to show that the range defined with respect to center C_0 is equal in size to the ranges defined for any other origins. This can most easily be verified geometrically, keeping in mind that line-point angles, thus radial angles θ, are expressed relative to π radians, i.e. modulus π. In Fig. 4.24, let C_1 and C_2 be alternate origins. Consider the perpendiculars from C_1 and C_2 to vectors V_1 and V_2. By similar triangles congruence relations, we see that both the ranges are of size equal to ϕ.

Now consider an arbitrary line-point on the arc. With respect to C_0 as the origin, let the difference between vector $\tilde{P}_1$ and the perpendicular to this line-point be φ. Consider the perpendiculars from the alternate origins, C_1 and C_2 to this line-point. It can easily be shown by simple geometry that the difference in angle between the perpendicular to this line-point and the perpendicular to vector V_1 from either C_1 or C_2 is equal to φ. From Fig. 4.23 we know the frequency distribution of this line-point's radial angle (φ from $\tilde{P}_1$), and this is also the frequency value of line-point's radial angle with respect to the other origins. So we see a 1-1 mapping of the θ frequency values from the range $\mathcal{R}_0$ to the ranges $\mathcal{R}_1$ and $\mathcal{R}_2$. Since the size of the range is exactly the same, it implies that the histograms are identical in shape and size. The only difference between them is merely in a lateral shift.

<u>COROLLARY</u> The entropy value of a given arc obtained by this proposed method is independent of the position of its origin.

<u>Proof</u>: Trivial. Since entropy is taken over histograms that are of the same shape and size, their values must be equal.

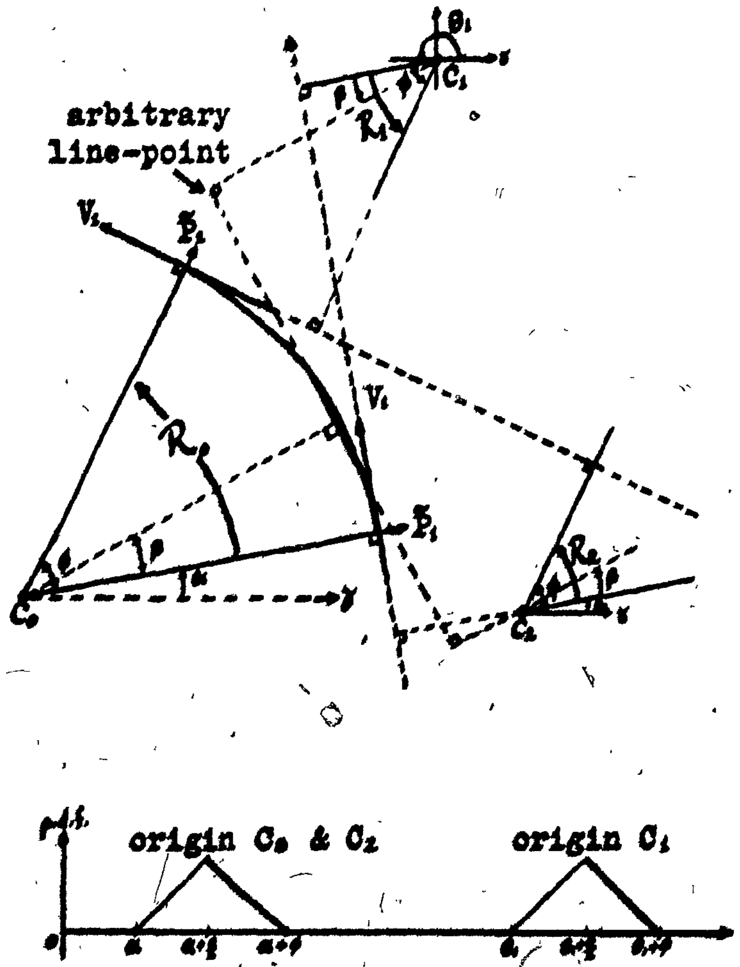


Fig. 4.24 The θ histograms from different "origins" differ only in a lateral shift on the horizontal axis.

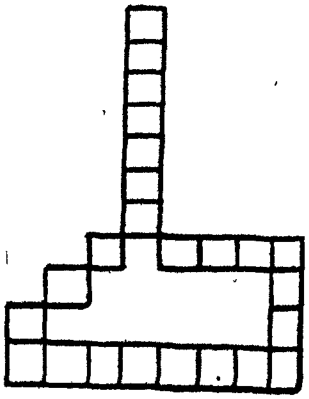


Fig. 5.1

An example of a
"Single Boundary Appendage".

5. Investigative Studies on the Implementation of this Method

5.1 Some Variations to the System

In this section, minor variations made to the present system are studied and their results briefly discussed. Individual modifications or combinations of them have been implemented for this study.

5.1.1 Use of logarithms to any base

Since entropy values were not normalized in the implementation of this system, this merely implies a difference in the order of an exponential function between values from different bases used. In our system common logarithms was arbitrarily chosen for use.

5.1.2 Absence of Single Boundary Appendages

If we can assume that the patterns or shapes that are being considered do not have "single boundary appendages" like in the example in Fig. 5.1, then the present system can be simplified to some extent. We can remove the "mechanism" that checks for these edges and handles them as boundary points on different sides of the appendage. If a pattern has i boundary points and m of these points belong to the appendage's boundary, we can determine the approximate amount of computation incurred by this mechanism. Let n be the size of the neighbourhood and assume a simply-connected boundary. With reference to the line numbers of the program listings in Appendix 1 and the superscripts representing c for comparisons, a for assignments and g for go to's, we have:

Line 29-32: $\frac{2n(2n-1)}{2} * \left\{ 1\tfrac{1}{2}^{c} + \tfrac{1}{2}^{a} + (m+\tfrac{1}{2})^{g} \right\}$

Line 37-39: $i * \left\{ \frac{2n(2n-1)}{2} * (1^{c} + 1^{g}) \right\}$

Line 119-122: $i * (2n-1) * \left\{ 1\tfrac{1}{2}^{c} + \tfrac{1}{2}^{a} + (m+\tfrac{1}{2})^{g} \right\}$

Adding the above yields:

$$(2n-1) * \left\{ \tfrac{3}{2}(n+i) + mi \right\} \quad \text{comparisons} \qquad (5.1.1)$$

$$(n - \tfrac{1}{2}) * (n + 1) \quad \text{assignments} \qquad (5.1.2)$$

and $$(2n-1) * \left\{ n(i+\tfrac{1}{2}) + \tfrac{1}{2}i + m(n+i) \right\} \quad \text{go to's} \qquad (5.1.3)$$

Note that if m = 0, only Eqn. 5.1.3 would show any change, implying that the remaining computations are wasted. In removing this "mechanism", the reduction in the amount of computations would be the sum of Eqns. 5.1.1, 5.1.2 and 5.1.3.

5.1.3 Incorporate Pre-processing into the Main Program.

This implies the incorporation of the contour or line follower and smoothing algorithm for boundary points into the main program. There are two ways about which one could do this:

(a) A physical concatenation of the two systems. This means literally placing the pre-processing procedure before the main procedure and making sure that the preprocessed boundary points are stored and carried over into the main procedure. In this case, there is no need for disk or card storage of the boundary points and no disk or card input is required for the main program. Although it will be one single unit, the space required increases due to the existence of a 120 x 120 picture matrix. This matrix is stored in one byte character form to ensure minimal storage; access to individual bits are not possible when using the Fortran H compiler except when logical (bit) variables are used, in which case, reading, outputting and manipulating them become very inefficient and undesirable in terms of time.

(b) An algorithmic merge of the two systems. This means doing both procedures at the same time as the boundary is being traversed. Again this requires more storage space and incurs more computation. For clarity and easier understanding of the system, this approach was not used. The following algorithm would show its relative complexity when compared to the actual implementation of the method.

Given a 120 x 120 picture matrix of the quantized shape with a single boundary stored on data cards, the algorithm follows:

1. Read the picture into a 120 x 120 array and print it out for the purpose of documentation.
2. Locate the starting point by a scan or read in the manually located coordinates of the starting point from data cards.
3. Read in the system's variables (NB,NS,XX).

4. Determine the first n points using the contour follower, (n > 1 and approximately 125% of 2*NB), and smooth the boundary points by removing 1-step staircases.
5. Check to see if the number of remaining points satisfy the required segment length of 2NB+1. If not, continue to obtain more boundary points, smooth them and repeat step 5.
6. Determine and store the sign of the curvature of the (NB+1)st point -- this is the point under consideration.
7. Determine the first set of angles from the line-points.
8. Map the angles into the Θ histogram.
9. Establish the high frequency region, determine and store the tangent angle and entropy measures.
10. Zero out the histogram
11. Move to the next boundary point; recopy relevant data, determine a few more boundary points, smooth them and execute step 5 and 6.
12. Determine new line-point angle values.
13 Go to step 8 and repeat until the first evaluated point is next in the process to be evaluated, i.e. the boundary has been traversed and all resulting values stored.
14. For documentation purposes, print the smoothed boundary point coordinates and their curvature signs.

Although we know that this algorithm is practical enough to be implemented, the number of times processes like in step 5 and 11 are done and the required storage for arrays presents additional problems. Also since the use of character or logical variables in Fortran H is unavailable and undesirable respectively, points have to be stored in full words, i.e. four bytes format, instead of one byte for character format and one bit for logical formats. This implies the involvement of a large amount of required space that would be badly utilized. Moreover, if one insists on using character variables, then the WATFIV compiler has to be used. Consequently, execution time would greatly increase by at least 200% which can be deemed unnecessary and to our disadvantage in terms of performance.

5.1.4 No Pre-processing of Boundary Points

The consequences of using an unprocessed boundary are few and individually rather insignificant, but each deserves some mention.

1. If the process to remove 1-step staircases is omitted, there is an approximate decrease of about 10 % in CPU time when using the WATFIV compiler. This first sub-system usually takes about eight to ten seconds, thus omitting the pre-processing does not significantly reduce the amount of computation time. On the other hand, doing this preprocessing reduces the number of boundary points by an average of 5% for a shape having about 600 boundary points. This reduction in the total number of points would affect the amount of computation time required in the main system.
2. To obtain a similar "quality" of tangent and entropy measures as in the case when boundary points were processed, the value of NB and XX have to slightly increase while NS should slightly decrease. NB, the neighbourhood size has to increase to reduce the influence of more horizontally and vertically aligned boundary points which generate "noise" for the segment under consideration. XX has to increase to exclude these line-points that produce a more "noisy" histogram. A slightly larger value of XX would ensure that we disregard any potentially present "noise" peaks in the histogram. NS, the number of angular partitions, should decrease to obtain better measures as it ensures that the slot with the highest count is not always the slots where horizontal and vertical line-points accumulate.

   The increase in the size of NB would evidently cause an increase in computational time. But the increase of XX and the decrease of NS actually help to reduce it, although not sufficient enough to warrant any significant difference in the overall system. The total increase in time would be approximately 15 % more than the original system.
3. For research purposes, a tangent was drawn at each boundary point to aid in the objective evaluations of the method's performance. An increased number of boundary points caused the

resulting pictures to look more untidy as more tangent lines were drawn per absolute length on the boundary. Of course, more time was involved.

In general, using an unprocessed boundary would result in an overall increase in computational time, a risk of less accurate tangent and entropy estimates and an increase in the number of points being considered.

In this same unprocessed environment, some changes in its logic have been studied. We can ignore the Θ angle values of line-points that are between points less than $\sqrt{2}$ or 2, etc units apart. In this way, we tried to reduce the amount of noise that originates from horizontally and vertically aligned boundary points. Instead, we observed an increase in the number of misdirected tangent lines because this logic makes the system pay more attention to line-points that are farther apart. Angles of line-points that are farther apart represent the axial direction of the segment better than the local tangent direction. Thus this system would work better for a smaller neighbourhood size and larger angle partitions, i.e. smaller NS.

5.2 Other Methods of Tangent Estimation

For the purpose of performance comparison, two alternate methods have been studied. These methods also use the Θ histogram. In fact they are minor variations of the existing method; their effectiveness would be briefly discussed.

1. Take the angle at the center of the angle partition in the histogram that has the highest count. Theoretically, this is acceptable as we can refer to results obtained in Chapter 4. However, in the quantized situation, as only NS values of the tangent are possible, we get a step-like and discontinuous tangent graph showing values that are approximated to half the size of the angle partitions. In this case, one might want to increase NS, but results have shown that for NS greater than some value, we may get varying values with smaller step size but they occasionally fluctuate even in regions of lower curvature. This is due to the sensitivity of the histogram when mapping

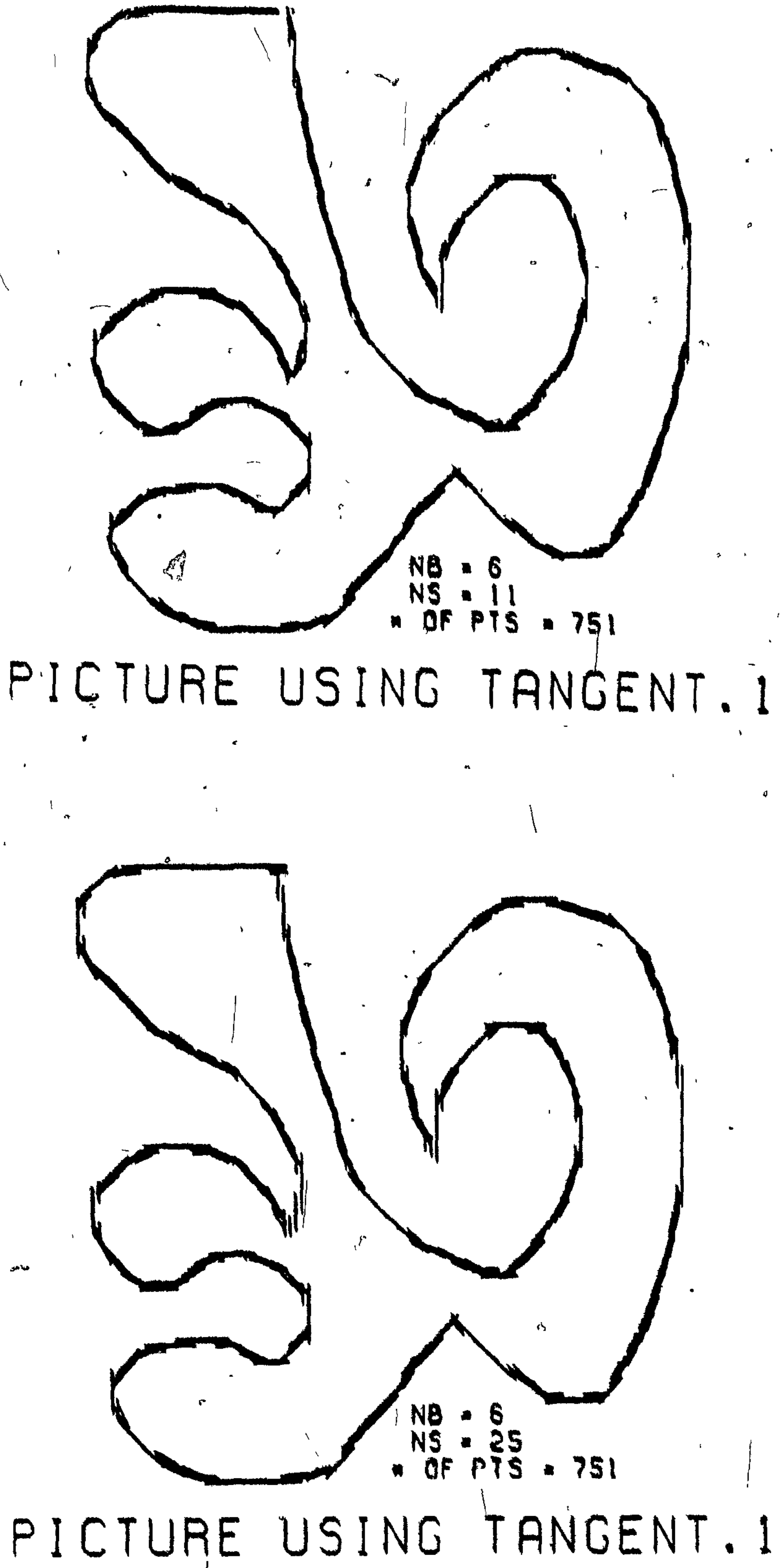


Fig. 5.2 The Reconstructed Shape

line-points into smaller angle partitions. This method is not recommended if pre-processing is not done, otherwise, it is fairly reliable except for its non-continuity in angle estimates. Tangent estimates obtained in this way have shown reasonably good pictures for smaller values of NS, see Fig. 5.2.

2. This method was heuristically formulated as an extension of the above case with the basic assumption that the tangent lies close to the angular value of the highest cell of the histogram. It has been studied and tested in depth and found to be faster and in some ways better than the implemented method. A brief account of this study is discussed here. The only disadvantage of this method is that it cannot handle noisy pictures as well as the present system.

   The value of the threshold XX is initially assumed to be large, say greater than 0.4. Thus only a few histogram slots are likely to be selected for use in the calculations. If we denote the highest cell as the k th slot and its count as C_k, then the angular value representing the k th slot is $(k-\frac{1}{2}) * \eta$, where η is the size of an angle partition for a given value of NS; i.e. $\eta = \pi/NS$. In general, let C_i be the count of the i^{th} cell of the histogram. Then the tangent estimate using the selected slots is

$$= (k-\tfrac{1}{2}) * \eta + \sum_{i \neq k} \frac{C_i}{C_i + C_k} * \frac{\eta}{i-k} \qquad (5.2.1)$$

Consider the following example: take an arbitrary histogram from which four cells with counts, 12,22,11,11 have been selected for calculation. Then using Eqn. 5.2.1, we substitute the counts to get:

$$= (k-\tfrac{1}{2}) * \eta \quad + (\tfrac{12}{34} * \tfrac{\eta}{-1}) + (\tfrac{11}{33} * \tfrac{\eta}{1}) + (\tfrac{11}{33} * \tfrac{\eta}{2})$$

$$= (k - \tfrac{12}{34}) * \eta \quad \cong \quad (k - 0.353) * \eta$$

As a comparison, we note that using the first method would return a tangent estimate of $(k-\frac{1}{2}) * \eta$ and while using the original method, the tangent would be estimated at:

$$\sum_{i=k-1}^{NS} \hat{p}_i * i = \left\{ \tfrac{11}{56} * (k+2) + \tfrac{11}{56} * (k+1) + \tfrac{22}{56} * k + \tfrac{12}{56} * (k-1) \right\} * \eta$$

$$= (k + 0.375) * \eta$$

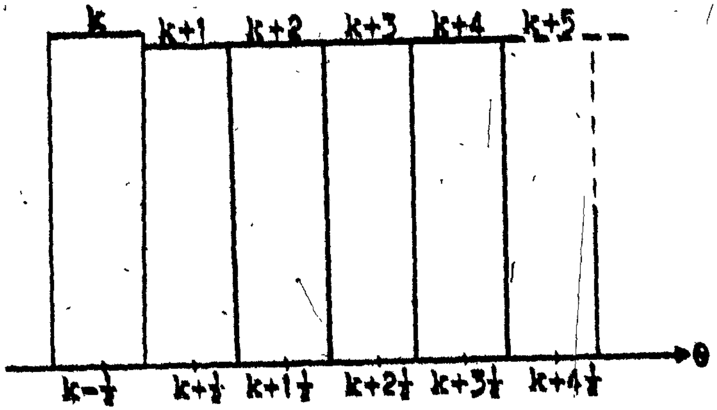


Fig. 5.3 Tangent Estimate Nethod 2
The extreme case of a histogram having the cell with the highest count flank on its side by many cells with counts slightly less than it.

showing that they all are within the range of one slot, η radian.

If we have two equal-valued slots, one of which is designated as the highest slot, then the tangent estimate using this method would be evaluated as $= (k-\frac{1}{2})*\eta + \frac{1}{2}*\eta = k*\eta$ which is the angle at the boundary common to both cells. This is of course the extension of the first method and is intuitively sound. We observe that if symmetry is encountered on both sides of the highest histogram cell, this method would return the value of $(k-\frac{1}{2})*\eta$ as the tangent estimate since $\sum \frac{C_i}{C_i+C_k} * \frac{\eta}{i-k} = 0$. This result coincides with both of the other methods of tangent estimation.

We use the expression in Eqn. 5.2.1 to estimate the tangent angle from a given histogram, so we can simply estimate the range of its values if we consider the extreme cases. In Fig. 5.3, we consider the highest cell flank on its right by many cells with values slightly less than it. In this case, the term $\sum \frac{C_i}{C_i+C_k} * \frac{\eta}{i-k}$ shows a geometric progression in $(\frac{1}{i-k})$ and since $C_i \leqslant C_k$, the expression $\frac{C_i}{C_i+C_k} \leqslant \frac{1}{2}$ for all values of i, so the upper bound becomes: $\leqslant \sum_{i=1}^{m} \frac{1}{2} * \frac{\eta}{i} = \eta \sum_{i=1}^{m} \frac{1}{2i}$. Although $\sum \frac{1}{2i}$ diverges, for a small m, say $\leqslant 10$, the upper bound becomes

$$\begin{aligned}(k-\tfrac{1}{2} + \sum_{i=1}^{m} \frac{1}{2i})*\eta &= (k - \tfrac{1}{2} + 1.465)*\eta \\ &\leqslant (k+1)*\eta \qquad (5.2.2)\end{aligned}$$

Similarly, the lower bound is estimated for $m \leqslant 10$ at:

$$\geqslant (k-2)*\eta \qquad (5.2.3)$$

With this assumption, the range of the tangent estimate is: $\{(k-2)*\eta, (k+1)*\eta\}$ whose extreme values can only be attained on these extreme cases where we have less than or equals to ten cells consecutively adjacent to the highest cell with values slightly less than the k th slot. Generally, the bounds can be assumed at $\{(k-1\frac{1}{2})*\eta, (k+\frac{1}{2})*\eta\}$, that is, the range of the tangent estimates is less than the size of one angle partition. Thus, this method would place the tangent estimate very close to the slot with the highest histogram count.

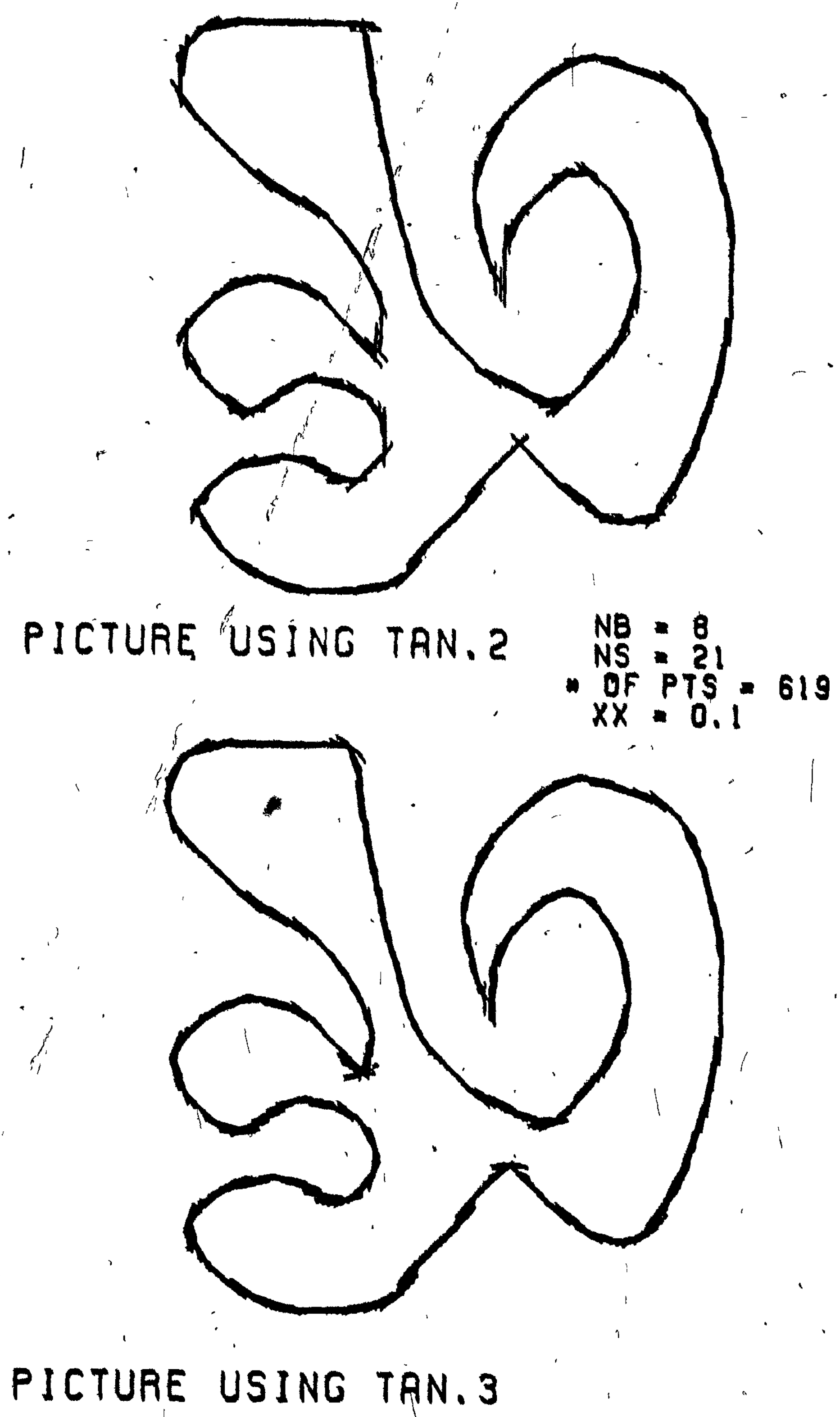


Fig. 5.4 The Reconstructed Shape

Now that the extent of the range of the tangent estimates has been realised to be small, the value of XX does not have to be restricted in any way as it will not substantially change the results. In comparing the results obtained by this method that we refer to as TAN.2 in Fig. 5.4 and our implemented method, (TAN.3), we can make some general observations. We note that TAN.3 gives a more averaged-up and rounded set of values than TAN.2, i.e. its graph would show a relatively smoother curve in terms of its rate of increase and decrease. For a smaller value of NB, the neighbourhood size, the picture from TAN.2 in Fig. 5.5 shows locally more accurate tangent estimates. However, the lack of smoothly varying lines at regions of higher curvature results in a rather squarish-looking picture. As NB or NS increases, it would get more untidy at higher curvature regions. This method, if used, has the advantage of being faster to compute and the results obtained are relatively just as good.

5.3 The Use of a $\rho - \theta$ Transformation

A study was done on the measurement of curvature using what we will refer to as the $\rho - \theta$ transformation, represented in Fig. 5.6, similar to the transformation used in our method. For each pair of points of the segment on the quantized boundary, the line joining them, the line-point, is mapped into the $\rho - \theta$ space, where θ is the radial angle and ρ is the perpendicular distance to the line or its extension. Clearly, for a straight segment, all the θ's and ρ's are constant and hence the $\rho - \theta$ histogram is perfectly peaked giving zero entropy and hence indicating no curvature.

This section discusses this heuristic method with some general experimental results. Theoretical considerations show the method to be unsound. Thus, the purpose of this presentation is primarily for the sake of those who might think of researching it as a feasible extension of our method to measure curvature.

The use of this transformation entails an additional system variable called NP, the number of equal-width partitions on the picture retina, independent of orientation. This is used

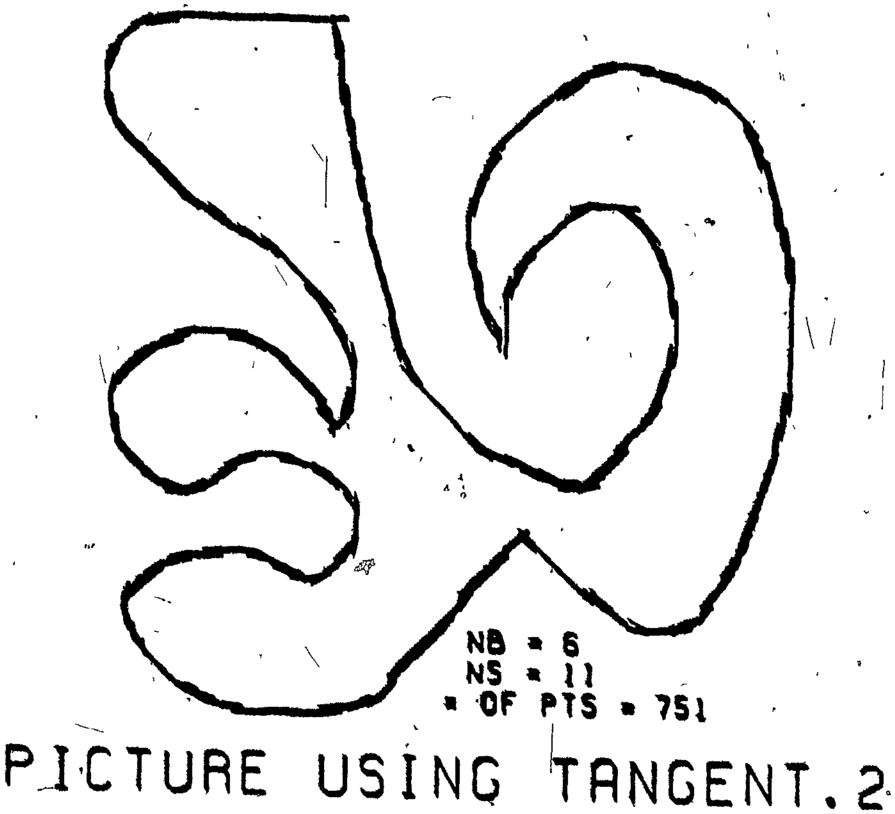


Fig. 5.5

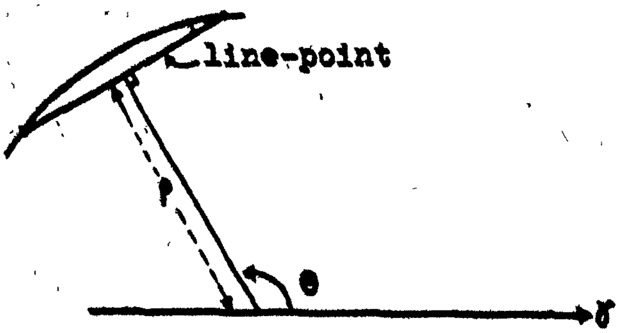


Fig. 5.6 The ρ - θ Transformation. The line-point is represented by the parameters (ρ, θ).

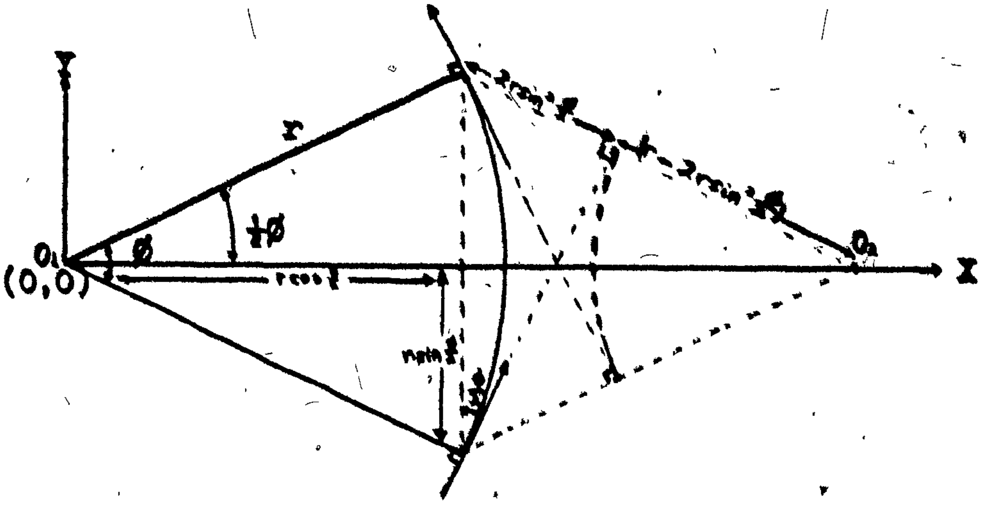


Fig. 5.7 An example showing the range of ρ to differ when taken with respect to the two origins, O_1 and O_2 .

to class line-points at different distances away from the center of the retina, which we refer to as the origin. The additional use and evaluation of ρ with this variable for each line-point results in an increment of over 400 % in C.P.U. time in the tests done on it. Using this variable also requires the use of a large array of size NS x NP as the histogram. The estimation of the tangent is done independently of this variable and the curvature estimate is taken as the entropy of the two-dimensional histogram.

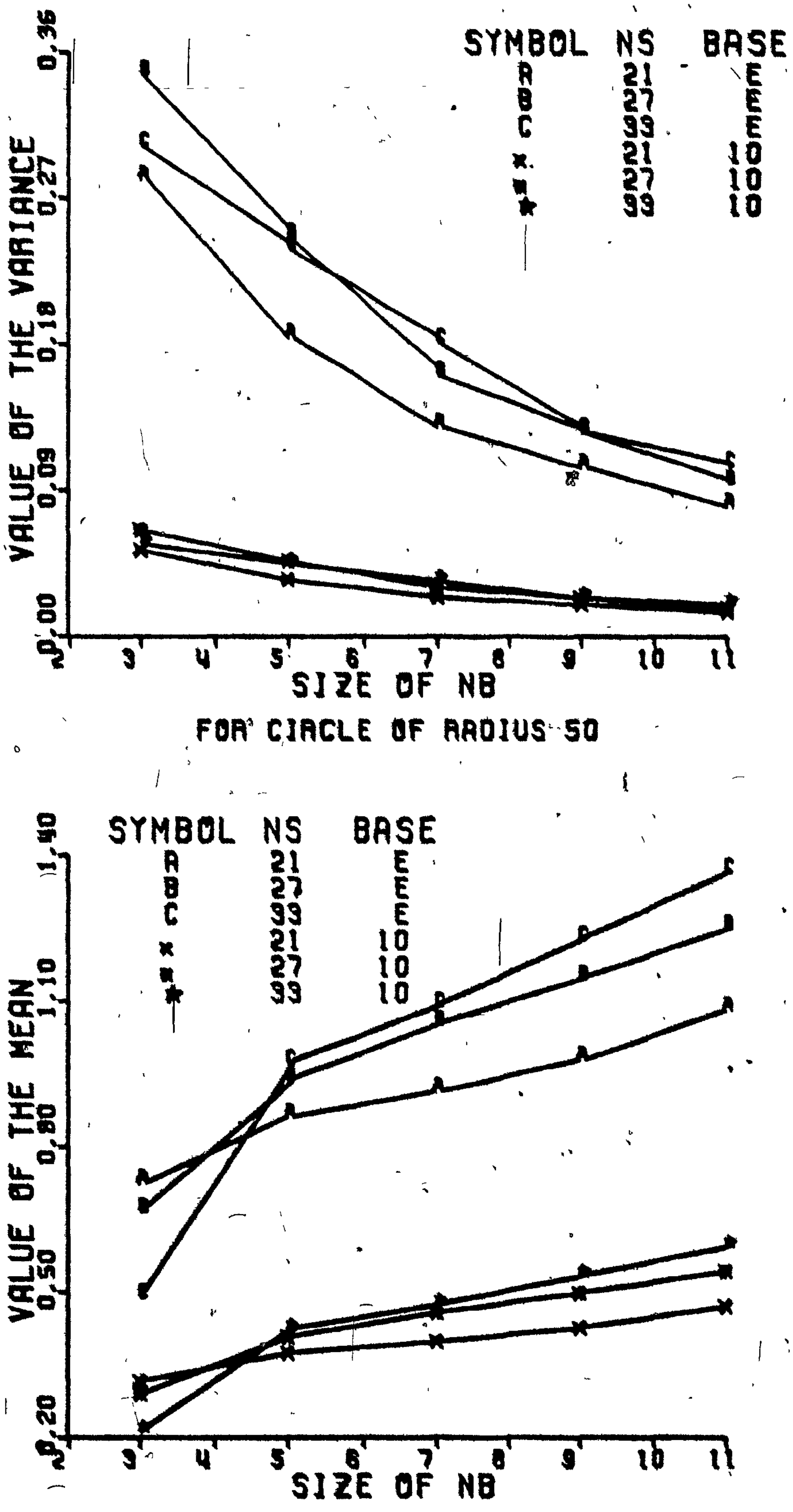
From experimental results, a larger size NP produces entropy values with a larger range which substantially aids in differentiating high from low curvature points. A larger NB size will give smoother curvature graphs. Now, for the theoretical aspect of this method that makes it undesirable.

Consider the following theorem:

> The entropy measure of the two-dimensional $\rho - \theta$ histogram obtained by using the $\rho - \theta$ transformation on line-points is NOT independent of the position of its origin.

One way to prove this is by assuming independence and producing a counter-example to contradict the assumption. With reference to Fig. 5.7, given an arc of a circle of radius r subtended at the center by an angle of ϕ, construct the two origins at O_1, $(0,0)$ and O_2, $(2*r*\cos(\frac{1}{2}\phi), 0)$. We know that entropy measures from histograms having a different number of cells are not equal. Furthermore, since the quantization of ρ remains fixed in a given problem, a different range for ρ determines a different number of histogram cells. We now only need to show that the ranges of ρ from the two origins O_1 and O_2 have different sizes.

The range of ρ with respect to O_1 is clearly $(r\cos\frac{1}{2}\phi, r)$. Thus the size of it is: $r * (1 - \cos(\frac{1}{2}\phi))$ (5.3.1)
Similarly, the range of ρ with respect to the origin O_2 is calculated to be $(-r+2r*\sin^2(\frac{1}{2}\phi), -r*\cos(\frac{1}{2}\phi))$. Thus the extent is: $r * (\cos(\frac{1}{2}\phi) + 2*\sin^2(\frac{1}{2}\phi) - 1)$ (5.3.2)
Obviously, the size of the ranges is not equal in general. It is equal for all ϕ such that $\cos(\frac{1}{2}\phi)+\sin^2(\frac{1}{2}\phi) = 1$, i.e. $\cos(\frac{1}{2}\phi) = \pm 1$. That means that ϕ must be zero or π for the size of the two ranges to be equal.



FOR CIRCLE OF RADIUS 50

FOR CIRCLE OF RADIUS 50

Fig. 5.8 Graphs of the Entropy Means and Variances

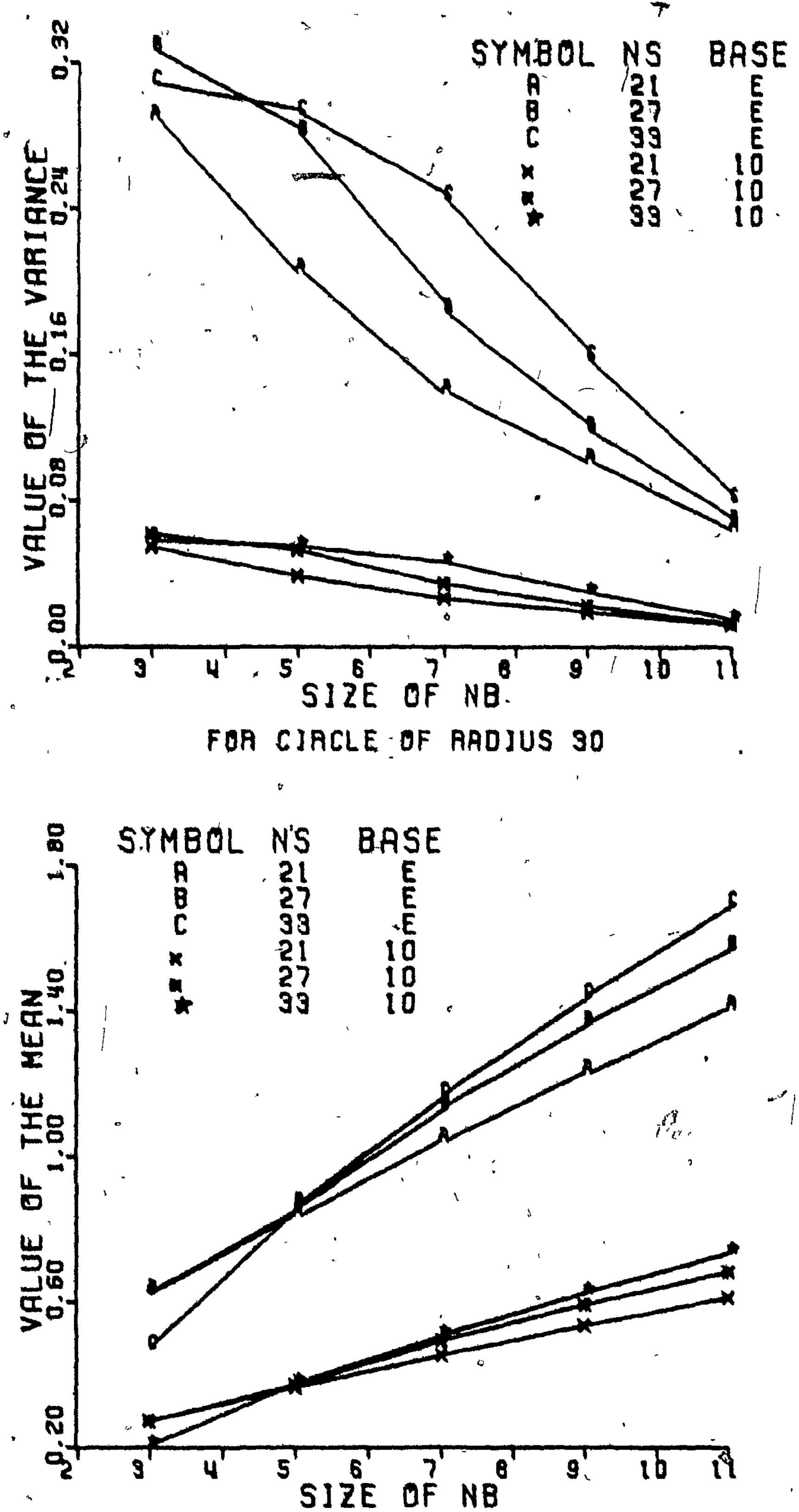
5.4 Some Results on the Effects of Quantization

When digitizing pictures using quantized grids, a shape is represented differently and in different degrees of accuracy by using different sized grids; a picture digitized using a smaller grid size would produce more detailed silouhettes which would resemble the original shape much better. In our case, to simulate quantization on different sized grids, we need only to vary the size of our shapes on the same size 120 x 120 picture frame used for input into the system.

The motivating basis of this study is to see how this proposed method's performance is affected by different levels of quantization when the same parameters are used. A shape suitably chosen for this study is the circle. Since digitization effectively removes the property of continuity, resulting measures when compared to theoretical results tend to look rather discontinuous and unpredictably fluctuating instead of the rigidly consistent value of the reciprocal of the radius. Thus we have a good basis from which to compare entropy values obtained from circles with their classical curvature values. Since consistency of values can be realized if we know its mean and variance, these statistics of the entropy values will enable us to determine which set of variables for the system would produce "better" measurements in terms of smaller variations away from the mean. Thus we can relate the corresponding differences of values obtained from different sized circles. Results of these statistics when plotted will easily show their relationship in terms of performance.

In the first case, in Fig. 5.8, we use a circle of radius 50; with the process of digitization and preprocessing of the boundary, we ended up with a 282 point boundary. Then we determine its entropy values for all points on the boundary for various values of NB and NS. We observed, as expected*, that the values obtained using common logarithms attain much smaller variances and generally lower mean values. We also note that for

* since values of logarithm to the base 2 is larger than values of the logarithm to the base 10.



FOR CIRCLE OF RADIUS 30

FOR CIRCLE OF RADIUS 30

Fig. 5.9 Graphs of the Entropy Means and Variances

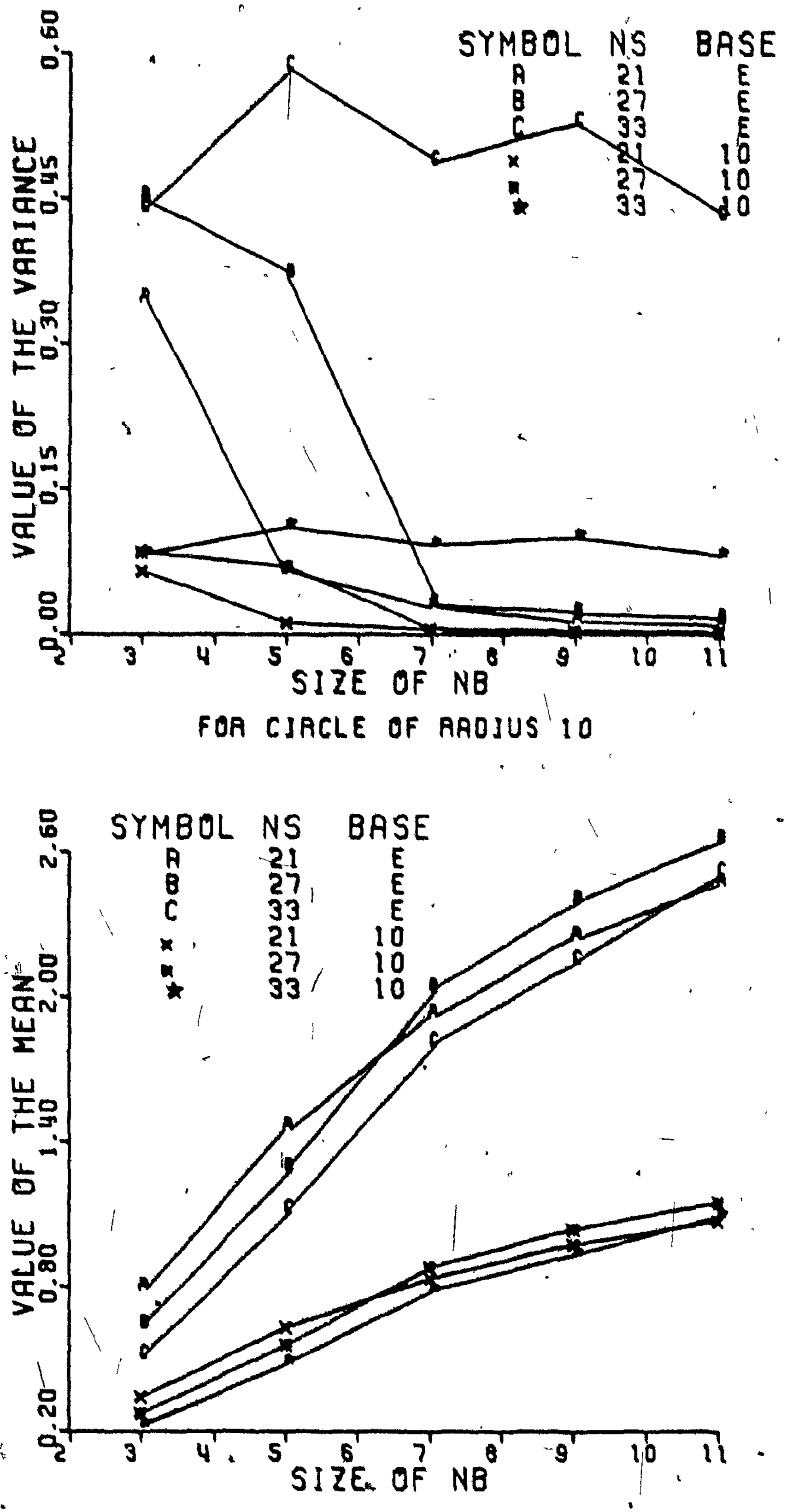
a fixed size NS, as NB increases, the means increase and at the same time, the variances decrease. For a fixed size NB, as NS increases, both the means and variances increase. These results reinforce our intuitive idea that the variance would decrease and become small for suitably chosen values of NB and NS.

Similarly, the same was done to digitized circles of radius 30 and 10. We observe that as the size of the radius increases, the means of the entropy values decrease as expected of them. As NB (or NS) increases, the values of the mean also increase. As for the variances, they generally decrease as the radius increases and also as the variable NB increases. The variances increase as the variable NS decreases. Thus we see that for a fixed set of variables, they correspond relatively well to the differences of their theoretical curvature measurements of 1/radius.

To sum these observations, we can say that the propposed method produces entropy values--curvature indicators that correlate very well to theoretical results, and different levels of quantization merely causes the estimates to vary slightly depending on the values of the system variables used. Gross quantization, like in the case when the radius equals 10, in Fig. 5.10, evidently does not produce as consistent or accurate results as finer quantization. This is exhibited in the graph when the variances of the entropy values do not show much consistency for improvement with varying values of NB and NS like the graphs in Fig. 5.8 and Fig. 5.9. Most evidently, finer quantization would produce better results.

5.5 Application to Data Compression of Line Drawings

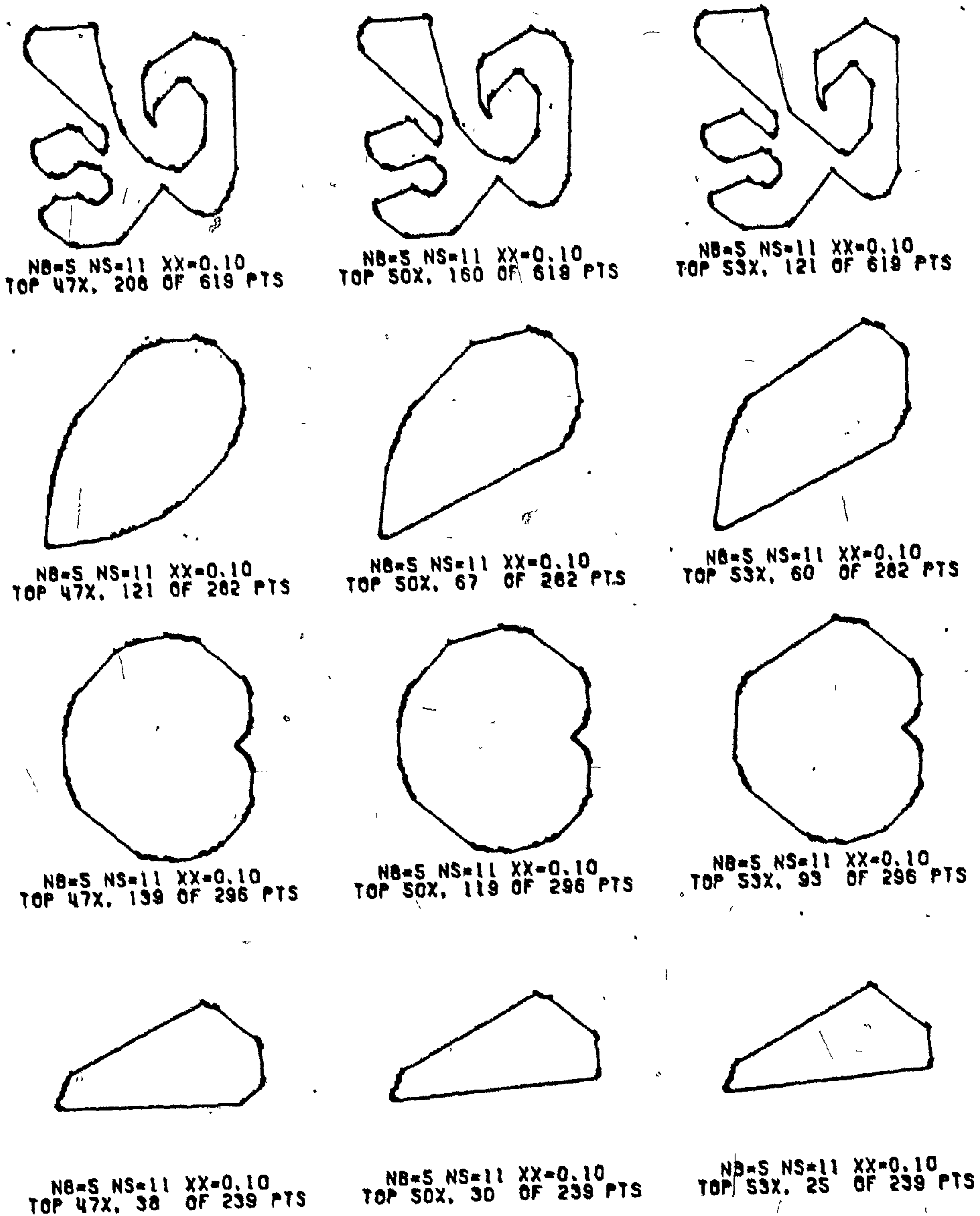
In many applications dealing with line drawings, data compression is of crucial importance. For example, in communication theory we would have to transmit much less information if we consider only a small subset of the points making up a line drawing (54). In the rapidly expanding field of automated cartography, it is important to store maps with elevation profiles



FOR CIRCLE OF RADIUS 10

FOR CIRCLE OF RADIUS 10

Fig. 5.10 Graphs of the Entropy Means and Variances



NB=5 NS=11 XX=0.10
TOP 47X, 206 OF 619 PTS

NB=5 NS=11 XX=0.10
TOP 50X, 160 OF 619 PTS

NB=5 NS=11 XX=0.10
TOP 53X, 121 OF 619 PTS

NB=5 NS=11 XX=0.10
TOP 47X, 121 OF 282 PTS

NB=5 NS=11 XX=0.10
TOP 50X, 67 OF 282 PTS

NB=5 NS=11 XX=0.10
TOP 53X, 60 OF 282 PTS

NB=5 NS=11 XX=0.10
TOP 47X, 139 OF 296 PTS

NB=5 NS=11 XX=0.10
TOP 50X, 119 OF 296 PTS

NB=5 NS=11 XX=0.10
TOP 53X, 93 OF 296 PTS

NB=5 NS=11 XX=0.10
TOP 47X, 38 OF 239 PTS

NB=5 NS=11 XX=0.10
TOP 50X, 30 OF 239 PTS

NB=5 NS=11 XX=0.10
TOP 53X, 25 OF 239 PTS

Fig. 5.11 Results obtained by thresholding the entropy graph of various shapes

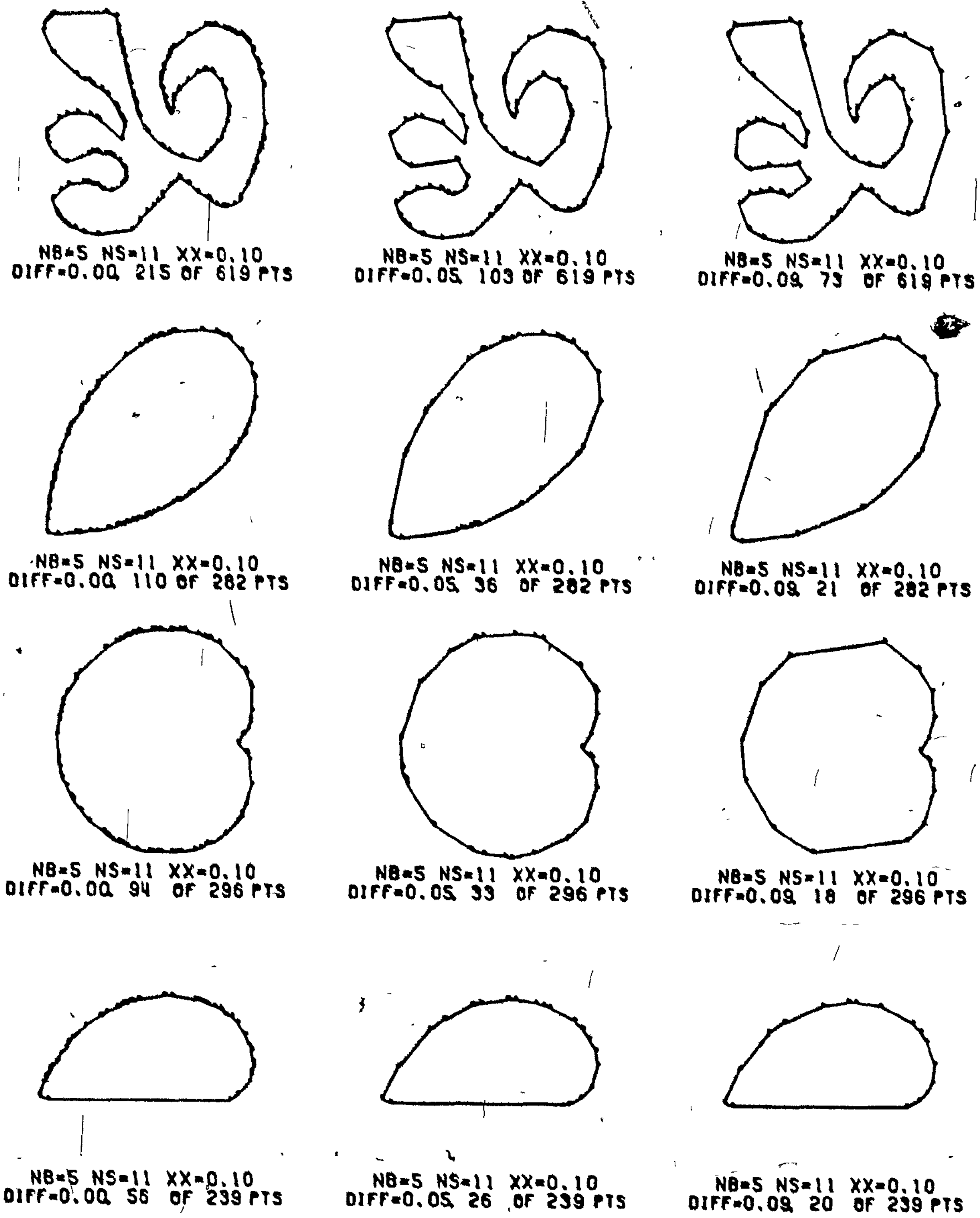
efficiently (55,56). Finally, in computer graphics, data reduction is important to (a) reduce storage requirements for the display file, (b) provide for faster image transmission and buildup, and (c) provide for faster transformations (57). In this section, we propose and investigate a method for reducing the number of points of a line drawing based on the entropy measure of curvature.

As originally intended, the measurements produced by the proposed method are supposed to enable one to distinguish low curvature points (regions) from high curvature points (regions). So, one could use the resulting entropy values in two ways:

(1) Apply a thresholding process to the values obtained from a given shape. This actually selects all points above a prespecified fraction of the maximum absolute entropy value for a given shape. No attention is given to the sign of the entropy values in selecting these points which are then joined by straight lines indicating that the region between them have lower curvature.

Experimental results have shown that a reasonably good resemblance of the original shape is obtained by using a threshold level of approximately 50% of its absolute maximum entropy value. A lower value would increase the number of points selected making the resulting picture look more like the original. Of course this is acceptable unless one were to be looking for a minimal number of points to best represent a picture, then one would have more points than necessary or required. A higher value will result in a reduction of the number of points selected and it may also omit some crucial points on the boundary necessary for the connected set of points to resemble the original shape. If a process using this method has no need of a joined set of the selected points, then a higher threshold level would essentially exhibit regions of higher curvature.

Fig. 5.11 shows some results of this thresholding process applied to some shapes. The second shape is the first quadrant of a four-leaved rose, the third is a cardioid and the fourth is the top section of the Spiral of Archimedes. We note that when



NB=5 NS=11 XX=0.10
DIFF=0.00, 215 OF 619 PTS

NB=5 NS=11 XX=0.10
DIFF=0.05, 103 OF 619 PTS

NB=5 NS=11 XX=0.10
DIFF=0.09, 73 OF 619 PTS

NB=5 NS=11 XX=0.10
DIFF=0.00, 110 OF 282 PTS

NB=5 NS=11 XX=0.10
DIFF=0.05, 36 OF 282 PTS

NB=5 NS=11 XX=0.10
DIFF=0.09, 21 OF 282 PTS

NB=5 NS=11 XX=0.10
DIFF=0.00, 94 OF 296 PTS

NB=5 NS=11 XX=0.10
DIFF=0.05, 33 OF 296 PTS

NB=5 NS=11 XX=0.10
DIFF=0.09, 18 OF 296 PTS

NB=5 NS=11 XX=0.10
DIFF=0.00, 56 OF 239 PTS

NB=5 NS=11 XX=0.10
DIFF=0.05, 26 OF 239 PTS

NB=5 NS=11 XX=0.10
DIFF=0.09, 20 OF 239 PTS

Fig. 5.12 Results Obtained by Selecting Peaks of Entropy Graphs of Various Shapes

these shapes are threshold at 47% of its absolute maximum entropy value, the reduction in the number of points ranged from 53% to 84%. At the threshold level of 50%, the amount of reduction ranged from 60% to 87%, and at the threshold level of 53%, it ranged from 69% to 90%. Objectively, it seems that a threshold level of 50% is the "border-line" case, where if it gets greater, the amount of distortion may not make the connected remaining points represent the original shape as well.

(2) Select peaks of the entropy value graph and designate them as locally higher curvature points. If one intends to select regionally higher curvature points, as opposed to global, so that higher curvature points of each region or segment are retained to enable the connected set of selected points to better resemble the original, then one could introduce a difference factor to iron out the presence of fluctuating values within a small range; that is, we choose more prominent peaks whose values differ from neighbouring values by a certain amount.

Fig. 5.12 shows two levels of difference applied to the entropy value graphs of some shapes. When DIFF = 0, all local peaks are selected. We observe that as this value increases, less points would be selected thus reducing its resemblance to their original shapes. This technique enables one to obtain a relatively more complete reduced shape. We observe that if all local peaks are selected (i.e. DIFF = 0), the reduction in the number of points ranged from 61% to 77%. With DIFF = 0.05, the reduction ranged from 83% to 89%, and with DIFF = 0.09, it ranged from 88% to 94%. We note that while the number of points are reduced to a greater extent than in the first technique, the reduced shape still seems to resemble the original figure much more.

The first technique essentially selects higher curvature regions depending on how high the threshold level is; the higher the threshold, the higher would the curvature of the remaining points be. Thus it would be useful for the extraction of higher curvature regions. The second technique pin-points regionally or locally higher curvature points instead of regions depending on the value of the difference factor. These techniques could be used as a data reduction process for line drawings in picture processing Pattern Recognition applications.

	Neighbourhood Size	No. of Angular Sectors	Acceptable Fraction	No. of Points	Time (secs)
	NB	NS	XX		
1)	5	11	0.10	619	2.0667
	5	13	0.10	619	2.250
	5	15	0.10	619	2.13333
	5	17	0.10	619	2.13333
	5	19	0.10	619	2.250
	5	21	0.10	619	2.28333
	5	23	0.10	619	2.2333
	5	25	0.10	619	2.11667
	5	27	0.10	619	2.10
	5	29	0.10	619	2.1667
	5	31	0.10	619	2.21667
	5	35	0.10	619	2.08333
	5	21	0.33	619	2.2333
	5	35	0.33	619	2.08333
	6	19	0.10	619	2.58333
	6	21	0.10	619	2.650
	6	23	0.10	619	2.6333
	7	21	0.10	619	3.4500
	9	21	0.10	619	5.350
	9	35	0.10	619	5.41667
	13	11	0.10	619	8.31667
	13	21	0.10	619	8.1667
	13	35	0.10	619	8.38333
	13	21	0.33	619	8.4333
	13	35	0.33	619	7.950
2)	5	11	0.10	54	0.2333
3)	5	11	0.10	166	0.48333
4)	5	11	0.10	282	0.83333
5)	5	11	0.10	239	0.7500
6)	5	11	0.10	296	0.98333

Fig. 6.1

A Table showing the C.P.U. time involved for various values of the parameters NB, NS, XX.

6. Summary and Conclusions

In this thesis, a method was proposed to measure the curvature of boundaries or lines quantized on two-dimensional square grids. An implementation of this method on the computer showed desirable results while the study and investigation into the theoretical aspect of this method established relationships to the classical definition of curvature. A suggestion was also made as to the use of these entropy values in a data reduction process for line drawings.

While comparative evaluations of this method have yet to be done with other direct methods of measuring curvature, we can only express its performance in terms of the time taken to evaluate all the entropy values for the boundary points of a shape. These measured times shown in the table in Fig. 6.1 are C.P.U. times and do not include preliminary input-output functions. That is to say, that if this system is incorporated into a picture processing application, these will be the additional amounts of time involved. Also note that the times listed (in seconds) should be considered in conjunction with the number of points the shape possesses.

In this thesis, a means of measuring and expressing the curvature of digitized lines, curves or contours have been established through the use of the θ transformation and the θ histogram's entropy. The proposed method in Chapter 3 has been substantially supported by its theoretical equivalence and implications given in Chapter 4 and 5; its equivalence to the classical definition of curvature, and its independence of the position of the origin used, implying independence of the position and orientation of the shape. Furthermore, we showed that the $\rho - \theta$ transformation is not as effective or theoretically and practically as desirable as the proposed θ transformation.

A relationship between classical curvature and the entropy measure as defined has been established. This forms the backbone to the validity of our method which can also be used to determine the axial directions of quantized segments. Furthermore, a smoothing process, the removal of 1-step staircases, has successfully improved and enhanced the results and performance of our method. Thus the results of this thesis form a contribution to the area of curvature measurement, tangent angle measurement, boundary pre-processing, feature extraction and information or data reduction, and more generally, to the field of picture processing and computer graphics.

The approach to this thesis has mainly been in the direction of a proposed technique, the study into its validity, an investigation into its practicality and the establishment of its theoretical basis. A variety of extensions of the work described in this thesis are possible. The most immediate ones are comparative evaluations of this method to other methods of measuring curvature, and other methods of data reduction process. Then one could work on the implementation of this method into picture processing and other pattern recognition applications that could use a curvature measuring or data reduction process.

APPENDIX 1

Algorithm for Subsystem I: Preprocessing

1. Read in from data cards a connected digitized picture of size 120 x 120 and print the original shape.
2. Read in or determine the coordinates of the starting point.
3. Determine the boundary by the contour tracing method and store their coordinates in two linear vectors (IB & JB).
4. Blank out input array.
5. Do procedure to remove "1-step staircases".
6. Recopy the remaining boundary points into a linear list.
7. Determine and record the curvature sign of each point.
8. Store the coordinates and signs of the boundary point.

Algorithm for Subsystem II: Main Program

1. Read in boundary coordinates and signs from storage device.
2. Start CLOCK1 procedure to begin timing of process.
3. Read in the 3 system variables and establish constants to be used in the computations.
4. Copy first segment of points into the working array (IL & JL).
5. Determine the first set of angle values while checking for coinciding boundary points.
6. Map angle values into the histogram.
7. Determine the range for computation by scanning from the slot with the highest count.
8. Determine and store the curvature and tangent angle measures.
9. Zero out the histogram and move ahead by one point.

10. Determine the new values of the angles and go to step 6 and repeat until each boundary point has been considered.
11. Call CLOCK2 to terminate the timing subroutine.
12. Store resulting arrays of the measures.

Algorithm for Subsystem III:

1. Draw the original shape by indicating positions of the boundary points.
2. Draw the tangent angle at each boundary point.
3. Plot the graphs of the measures.
4. Draw indicators of curvature at each boundary point and reconstruct the picture by thresholding curvature values.

```
LEVEL 21.6 ( DEC 72 )                                     OS/360  FORTRAN H

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF
                 C
                 C       VARIABLES REQUIRED FOR THE UNIQUE STATE OF THE SYSTEM
                 C       -----------------------------------------------------
                 C         NB   -  SIZE OF NEIGHBOURHOOD RADIUS  <.14 ON EACH SIDE
                 C         NS   -  NUMBER OF EQUAL ANGLE SECTORS IN TOP HEMISPHERE  < 39
                 C         XX   -  VALUE OF THE ACCEPTABLE FRACTION
                 C
    ISN 0002           DIMENSION TAN3(750),CUR3(750),ANGLE(435)
    ISN 0003           INTEGER HIST(39),I3(750),J3(750),SGN(750),IL(29),JL(29)
    ISN 0004           DATA TAN3,CUR3,ANGLE,HIST,IL,JL /1935*0.0,97*0/
    ISN 0005           COMMON /AREA3/ R90
                 C
                 C       CONSTANT VARIABLES USED IN PROGRAM
                 C       ----------------------------------
                 C      IB,JB -  ARRAYS CONTAINING I & J COORDINATES OF THE BOUNDARY POINTS
                 C      IL,JL -  ARRAYS CONTAINING COORDINATES OF THE SEGMENT POINTS
                 C      R90    -  VALUE OF HALF * PI  (90 DEGREES IN RADIANS)
                 C      II     -  NUMBER OF POINTS ON THE BOUNDARY
                 C
    ISN 0006           R90 = 11./7.
    ISN 0007           READ (1)  IB,JB,SGN
    ISN 0008           II = IB(750)
                 C
                 C       NOW READ IN REQUIRED SYSTEM VARIABLES & DETERMINE ITS CONSTANTS
                 C       ---------------------------------------------------------------
                 C       NB1  -  LENGTH OF SEGMENT OF BOUNDARY UNDER CONSIDERATION
                 C       NB2  -  POINTS TO THE SECOND LAST ELEMENT OF THE SEGMENT
                 C       CNS  -  SIZE OF ANGLE OF SECTORS IN DEGREES
                 C       RNS  -  ANGLE OF EACH SECTOR ( IN RADIANS)
                 C       IN   -  POSITION OF POINT UNDER CONSIDERATION
                 C       IP   -  POSITION OF LAST SEGMENT POINT IN BOUNDARY LIST
                 C
    ISN 0009           CALL CLOCK1
    ISN 0010           READ 1,NB,NS,XX
    ISN 0011         1 FORMAT (2I3,F5.2)
    ISN 0012           PRINT 2, NB,NS,XX,II
    ISN 0013         2 FORMAT (5X,'SIZE OF SEGMENT RADIUS =',I3,4X,'# OF ANGLES =',I3,4X,
                      *        'ACCEPTABLE FRACTION =',F5.2,4X,'# OF POINTS =',I4)
    ISN 0014           NB1 = 2*NB+1
    ISN 0015           NB2 = NB1-1
    ISN 0016           NB = NB + 1
    ISN 0017           CNS = 180./NS
    ISN 0018           RNS = 22./(7.*NS)
    ISN 0019           IN = NB
    ISN 0020           IP = NB1
    ISN 0021           DO 3 I=1,NB1
    ISN 0022              IL(I)=IB(I)
    ISN 0023         3    JL(I)=JB(I)
                 C
                 C       DETERMINE THE FIRST SET OF ANGLES
                 C       N    -  # OF ACCEPTABLE LINE PAIRS IN THE SEGMENT (<=NB1*NB2/2)
                 C
    ISN 0024           N = 0
    ISN 0025           DO 5 K1=1,NB2
    ISN 0026              J = K1 + 1
    ISN 0027              DO 5 K2=J,NB1
ISN 0028                      N = N + 1
ISN 0029                      IF (IL(K1).NE.IL(K2)) GO TO 4
ISN 0031                      ANGLE(N) = 999.
ISN 0032                      IF (JL(K1).EQ.JL(K2)) GO TO 5
ISN 0034            4         CALL FIND(IL(K1),JL(K1),IL(K2),JL(K2),ANGLE(N))
ISN 0035            5    CONTINUE
                  C
                  C        TO FIT THE ANGLE VALUES INTO THE HISTOGRAM
                  C
ISN 0036           10    DO 20 I=1,N
ISN 0037                   IF (ANGLE(I).NE.999.) GO TO 15
ISN 0039                   GO TO 20
ISN 0040           15      K1=ANGLE(I)/RNS + 1
ISN 0041                   HIST(K1) = HIST(K1) + 1
ISN 0042           20    CONTINUE
                  C
                  C        WE NOW HAVE THE HISTOGRAM, FIND THE POSITION OF THE HIGHEST SLOT.
                  C      K1  =  VALUE OF HIGHEST HISTOGRAM ANGLE COUNT
                  C      K2  =  ROW IN HISTOGRAM WITH HIGHEST ANGLE COUNT
                  C
ISN 0043                 K1 = 0
ISN 0044                 DO 22 I=1,NS
ISN 0045                   IF (HIST(I).LE.K1) GO TO 22
ISN 0047                   K1 = HIST(I)
ISN 0048                   K2 = I
ISN 0049           22    CONTINUE
                  C
                  C     INCLUDE NEIGHBOURING ROW COUNTS GREATER THAN THE ACCEPTABLE FRACTION O
                  C      TO DETERMINE THE RANGE OF THE ROWS TO BE INCLUDED IN THE CALCULATION
                  C
ISN 0050                 K = 1
ISN 0051                 JN = K1 * XX
ISN 0052                 IF (JN.EQ.0 ) JN = 1
ISN 0054                 XK = K1
ISN 0055                 I = K2
ISN 0056                 J = K2 - 1
ISN 0057           23    IF (J.EQ.0)  J = NS
ISN 0059                 IF (HIST(J).LE.JN) GO TO 24
ISN 0061                 XK = XK + HIST(J)
ISN 0062                 K = K + 1
ISN 0063                 I = J
ISN 0064                 J = J - 1
ISN 0065                 GO TO 23
ISN 0066           24    J = K2 + 1
ISN 0067           25    IF (J.GT.NS) J=1
ISN 0069                 IF (HIST(J).LE.JN) GO TO 26
ISN 0071                 XK = XK + HIST(J)
ISN 0072                 K = K + 1
ISN 0073                 J = J + 1
ISN 0074                 GO TO 25
                  C
                  C        NOW, DETERMINE TANGENT.3 AND CURVATURE.3
                  C          TAN.3 =  EXPECTED VALUE OF THE ANGLES USING ROW COUNTS AS WEIGHTS
                  C          CUR.3 = TAKE ENTROPY MEASURE ON THE ENTIRE HISTOGRAM
                  C
ISN 0075           26    SLP = 0.
ISN 0076                 CURV = 0.
ISN 0077                 JN = 1
ISN 0078          DO 27 J=1,K
ISN 0079             SLP = SLP + HIST(I)*(JN-0.50)
ISN 0080             PR = HIST(I)/XK
ISN 0081             CURV = CURV - PR*ALOG10(PR)
ISN 0082             JN = JN + 1
ISN 0083             I = I + 1
ISN 0084             IF (I.GT.NS) I = 1
ISN 0086      27  CONTINUE
ISN 0087          SLP = (SLP/XK) * CNS
ISN 0088          SLP = AMOD(SLP,180.0)
ISN 0089          TAN3(IN) = SLP
ISN 0090          CUR3(IN) = CURV * SGN(IN)
ISN 0091          IN = IN + 1
ISN 0092          IF (IN.GT.II) IN = 1
ISN 0094          IF (IN.EQ.N3) GO TO 40
               C
               C     ZERO OUT HISTOGRAM IN PREPARATION FOR THE NEXT POINT
               C
ISN 0096          DO 28 I=1,NS
ISN 0097      28     HIST(I) = 0
               C
               C     MOVE TO THE NEXT CELL, I.E. MOVE SEGMENT AHEAD BY 1 CELL BOUNDARY U
               C       DELETE THE 1ST ROW AND DETERMINE THE LAST COLUMN OF THE ANGLE MAT
               C
ISN 0098          DO 30 I=1,N32
ISN 0099             IL(I)=IL(I+1)
ISN 0100      30     JL(I)=JL(I+1)
ISN 0101          IP = IP + 1
ISN 0102          IF (IP.GT.II)  IP=1
ISN 0104          IL(NB1)=IB(IP)
ISN 0105          JL(NB1)=JB(IP)
               C
               C     TO DETERMINE NEW VALUES OF ANGLE
               C
ISN 0106          K1=IL(NB1)
ISN 0107          K2=JL(NB1)
ISN 0108          JP = 1
ISN 0109          JN = NB2
ISN 0110          K = NB2
ISN 0111          J = 1
ISN 0112      32  JN = JN - 1
ISN 0113          IF (JN.EQ.0) GO TO 34
ISN 0115          DO 33 I=1,JN
ISN 0116             K = K + 1
ISN 0117             ANGLE(J)=ANGLE(K)
ISN 0118      33     J = J + 1
ISN 0119          IF (IL(JP).NE.K1) GO TO 34
ISN 0121          ANGLE(J) = 999.
ISN 0122          IF (JL(JP).EQ.K2) GO TO 35
ISN 0124      34  CALL FIND (IL(JP),JL(JP),K1,K2,ANGLE(J))
ISN 0125      35  JP = JP + 1
ISN 0126          J = J + 1
ISN 0127          IF (JP.LE.NB2) GO TO 32
ISN 0129          GO TO 10
               C-                              ******  END OF JOB  ******
ISN 0130      40  CUR3(747) = NB-0.5
ISN 0131          CUR3(748) = NS+0.5
ISN 0132          CUR3(749) = XX
```

```
   ISN 0133              CUR3(750) = II + 0.5
   ISN 0134              TAN3(747) = NB-0.5
   ISN 0135              TAN3(748) = NS+0.5
   ISN 0136              TAN3(749) = XX
   ISN 0137              TAN3(750) = II+0.5
   ISN 0138              CALL CLOCK2 (N)
   ISN 0139              XX = N/60.
   ISN 0140              PRINT 42,XX
   ISN 0141       42     FORMAT (10X,'TIME =',F9.5,' SECONDS')
   ISN 0142              WRITE (7,45) TAN3
   ISN 0143              WRITE (7,45) CUR3
   ISN 0144       45     FORMAT (20A4)
   ISN 0145              STOP
   ISN 0146              END

*OPTIONS IN EFFECT*       NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,

*OPTIONS IN EFFECT*      SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =      145 ,PROGRAM SIZE =    20052

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                              55K BYTES OF CORE NC


                  C
                  C        THIS SUBROUTINE FINDS THE THETA VALUES GIVEN TWO POINTS
                  C
                  C
   ISN 0002              SUBROUTINE FIND(I1,J1,I2,J2,THET)
   ISN 0003              COMMON /AREA3/ R90
   ISN 0004              AX=J2-J1
   ISN 0005              IF (AX.EQ.0) GO TO 1
   ISN 0007              BX=I2-I1
   ISN 0008              IF (BX.EQ.0) GO TO 2
   ISN 0010              SLP = AX/BX
   ISN 0011              IF (SLP.LT.0.) GO TO 3
   ISN 0013              THET=ATAN(SLP) + R90
   ISN 0014              RETURN
   ISN 0015        1     THET=R90
   ISN 0016              RETURN
   ISN 0017        2     THET = 0.
   ISN 0018              RETURN
   ISN 0019        3     THET=R90 - ATAN(-SLP)
   ISN 0020              RETURN
   ISN 0021              END

*OPTIONS IN EFFECT*       NAME=  MAIN,OPT=02,LINECNT=60,SIZE=0000K,

*OPTIONS IN EFFECT*      SOURCE,EBCDIC,NOLIST,NODECK,LOAD,NOMAP,NOEDIT,ID,NOXREF

*STATISTICS*     SOURCE STATEMENTS =       20 ,PROGRAM SIZE =      544

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                              79K BYTES OF CORE NC

*STATISTICS*  NO  DIAGNOSTICS THIS STEP
```

```
$WATFIV ,TIME=10,PAGES=10,LINES=65,NOEXT
C
C          GIVEN ANY BINARY VALUED (0-1) CONNECTED PICTURE, THIS PROGRAM
C
C             1.  OBTAIN THE BOUNDARY POINTS
C             2.  PREPROCESS THEM MAINTAINING 8-WAY CONNECTIVITY
C             3.  DETERMINE EACH BOUNDARY POINT AS A CONVEX OR CONCAVE POINT
C
C          THE STARTING COORDINATE (IS,JS) IS GIVEN AS A POINT ON THE BOUNDARY
C
      CHARACTER P(120,120) , SP /' '/
      INTEGER IB(750) /750*0/, JB(750) /750*0/, SGN(750) /750*1/
      COMMON /AREA1/ R90,R180,R270,R360
      R90 = 11./7.
      R180 = 22./7.
      R270 = 33./7.
      R360 = 44./7.
      NB = 9
      READ, IS,JS
C
C          READ & PRINT THE ORIGINAL PICTURE
C
      READ 1,((P(I,J),J=1,120),I=1,120)
    1 FORMAT (60A1,20X)
      PRINT 2
    2 FORMAT ('1',50X,'THE ORIGINAL PICTURE')
      PRINT 99,((P(I,J),J=1,120),I=1,120)
   99 FORMAT ('0',3X,126('*'),/,4X,'*',124X,'*',/,120(4X,'*  ',120A1,
     1       '  *',/),4X,'*',124X,'*',/,4X,126('*'),/,'1')
C
C          DETERMINE THE BOUNDARY POINTS AND STORE THEIR COORDINATES
C             IN THE ARRAYS  IB(.) & JB(.)
C
      IB(1) = IS
      JB(1) = JS
      I = IS
      J = JS
      IP = IS
      JP = JS - 1
      II = 1
    3 KVAL = 0
      IF (P(I,J).EQ.SP) GO TO 5
      KVAL = 1
      IF (I.NE.IB(II)) GO TO 4
      IF (J.EQ.JB(II)) GO TO 5
    4 II = II + 1
      IB(II) = I
      JB(II) = J
    5 IN = I+(J-JP)*(1-2*KVAL)
      JN = J+(I-IP)*(2*KVAL-1)
      IP = I
      JP = J
      I = IN
      J = JN
      IF (I.NE.IS) GO TO 3
      IF (J.NE.JS) GO TO 3
```

```
C
C          BLANK OUT INPUT ARRAY AND SET UP CIRCULARLY LINKED LIST OF POINT
C            COORDINATES.  THEN PROCESS TO REMOVE "CORNER" POINTS OF BOUNDARY
C
          DO 6 I=1,120
             DO 6 J=1,120
    6           P(I,J) = ' '
          DO 7 I=1,2
             IB(II+I) = IB(I)
    7        JB(II+I) = JB(I)
          I = 0
    8     I = I + 1
    9     IF (I.GT.II) GO TO 13
          IF (IABS(IB(I)-IB(I+1))-1) 12,10,8
   10     IF (IABS(JB(I+1)-JB(I+2)).NE.1) GO TO 8
          IF (IABS(IB(I)-IB(I+2)).NE.1)  GO TO 8
          IF (IABS(JB(I)-JB(I+2)).NE.1)  GO TO 8
   11     IB(I+1) = 999
          I = I + 2
          GO TO 9
   12     IF (IABS(JB(I)-JB(I+1)).NE.1) GO TO 8
          IF (IABS(IB(I+1)-IB(I+2)).NE.1)  GO TO 8
          IF (IABS(IB(I)-IB(I+2)).NE.1)  GO TO 8
          IF (IABS(JB(I)-JB(I+2)).NE.1)  GO TO 8
          GO TO 11
C
C                  RECOPY THE GOOD BOUNDARY POINTS
C
   13     K = 0
          DO 15 I=1,II
             IF (IB(I).EQ.999) GO TO 15
             K = K + 1
             IB(K) = IB(I)
             JB(K) = JB(I)
             P(IB(K),JB(K)) = '*'
   15     CONTINUE
          II = K
          IB(750) = II
C
C            DETERMINE IF EACH BOUNDARY POINT IS CONVEX OR CONCAVE
C
          J = 2*NB
          K = NB
          DO 17 I=1,II
             J=J+1
             K=K+1
             IF (J.GT.II) J=1
             IF (K.GT.II) K=1
             THET = DIR(IB(I),JB(I),IB(K),JB(K))
             IF ((JB(J)-JB(I))*COS(THET)-(IB(J)-IB(I))*SIN(THET).LE.0.)
     *            GO TO 17
             P(IB(K),JB(K))= '0'
             SGN(K) =-1
   17       CONTINUE
C
C            PRINT BOUNDARY POINT COORDINATES
C
          PRINT 109,IS,JS,NB
  109     FORMAT ('      STARTING POINT COORDINATE :',2I3,'  NB=',I3)
          PRINT 18,II
   18     FORMAT (5X,'WE HAVE',I4,' BOUNDARY POINTS',5X,'* :: CONVEX POINTS'
     X           ,5X,'0 :: CONCAVE POINTS')
          PRINT 99, ((P(I,J),J=1,120),I=1,120)
          PRINT 19,(I,IB(I),JB(I),SGN(I),I=1,II)
   19     FORMAT (5X,'THE POINT NUMBER, COORDINATES AND SIGNS ARE LISTED AS
     *FOLLOWS :',//,126( 5X,5('#',I3,'(',I3,',',I3,'):',I3,3X),/))
          WRITE (2) IB,JB,SGN
          STOP
          END
```

```
C
C
C
C      THIS FUNCTION DETERMINES THE VECTOR DIRECTION ANGLE OF TWO ORDERED
C           POINTS (I1,J1) & (I2,J2)  AND RETURNS THE ANGLE IN RADIANS

       FUNCTION DIR(I1,J1,I2,J2)
       INTEGER I1,J1,I2,J2
       COMMON /AREA1/ R90,R180,R270,R360
       Y = J2-J1
       X = I2-I1
       IF (Y) 1,2,3
    1  IF (X) 4,5,6
    4  DIR = R180+ATAN(Y/X)
       RETURN
    5  DIR = R270
       RETURN
    6  DIR = R360-ATAN(-Y/X)
       RETURN
    2  IF (X) 7,8,8
    7  DIR = R180
       RETURN
    8  DIR = 0.
       RETURN
    3  IF (X) 9,10,11
    9  DIR = R180-ATAN(-Y/X)
       RETURN
   10  DIR = R90
       RETURN
   11  DIR = ATAN(Y/X)
       RETURN
       END

$DATA
```

REFERENCES

1. Attneave, F., "Some Informational Aspects of Visual Perception", Psychol. Rev. 61, 1954. pp. 183-193.

2. Connor, D.J., "Lateral Inhibition and the Area Operator in Visual Pattern Recognition", Ph. D. Thesis, Dept. of Electr. Engrg., U.B.C., 1969.

3. Bennett, J.R., "On the Measurement of the Curvature of the Boundaries of 2-Dimensional Quantized Shapes", Ph. D. Thesis, Dept. of Electr. Engrg., U.B.C., 1972

4. Symon, M., "A new Self-Organizing Pattern Recognition System", Conf. on P.R. 42, IEE London, 1968. pp. 11-20.

5. Kazmierczak, H., "The Potential Field as an aid to Character Recognition", Proc. Internl. Conf. Info. Proc., Paris, UNESCO, June 1959. pp. 244-247

6. Gibson, J.J., "The Perception of the Visual World", Houghton Mifflin, Boston, Mass., 1950. p. 195

7. Attneave, F. & Arnoult, M.D., "The Quantitative Study of Shape and Pattern Perception", Psychol. Bull. 53, 1956. pp. 452-471. (Reprinted in P.R., L. Uhr (Ed), pp. 123-141)

8. Freeman, H., "On the Encoding of Arbitrary Geometric Configuration", IEEE Trans Elect. Comp., Vol EC-10, June 1961. pp. 260-268.

9. Freeman, H., "On the Classification of line drawing data", in Models for Perception of Speech & Visual Forms, W. Wathen-Dunn, (Ed). MIT Press, Cambridge, Mass., 1967. pp. 408-412.

10. Mason, S.J. & Clemens, J.K., "Character Recognition in an Experimental Reading Machine for the Blind", in Recognizing Patterns, Kolers & Eden, (Eds), MIT Press, Cambridge, Mass., 1968, pp. 155-167.

11. Genchi, H., Mori, K*I., Watanabe, S., Katsuragi, S., "Recognition of Handwritten Numerical Characters for Automatic Letter Sorting", Proc. IEEE Vol. 56, Aug. 1968. pp. 1292-1301.

12. Glucksman, A.A., "Classification of Mixed Font Alphabets by Characteric Loci", Proc. IEEE Comp. Conf., Sept. 1967, pp. 138-141.

13. Guzman, A., "Analysis of Curved Line Drawings Using Context & Global Information", in Machine Intelligence 6, Meltzer & Mitchie (Eds), Edinburgh Univ. Press, 1971, pp. 325-375.

14. Sherman, H., "A Quasi-Topological Method for the Recognition of Line Patterns", Proc. IFIP Congress, 1959. pp. 232-237.

15. Brouilette, J.W. & Johnsons, C.W., "Pattern Recognition Proc.", 4th National Convention of Military Electronics, June 1960, pp. 179-182.

16. Rutovitz, D., "Centromere: Some Shape Descriptors for Small Chromosome Outlines", in Machine Intelligence 5, Meltzer & Mitchie, (Eds), Edinburgh Univ. Press, Edinburgh, 1970. pp. 435-462.

17. Freeman, H., "A Technique for Classification Recognition of Geometric Patterns", Proc. 3rd Internl. Conf. on Cybergenetics, Sept. 1961. pp. 348-369.

18. Freeman, H., "On the Digital Computer Classification of Geometric Line Patterns", Proc. Nat. Elect. Conf., Chicago. Ill., Vol. 18, Oct. 1962. pp. 312-324.

19. Suguira, T. & Higashiuwatoko, T., "A Method for the Recognition of Japanese Hiragana Characters", IEEE Trans. Info Theory, Vol. IT-14, March 1968. pp. 226-233.

20. Cantoni, A., "Optimal curve fitting with Piecewise Linear Functions", IEEE Trans. on Comp., Vol. C-20, Jan. 1971, pp. 59-67.

21. Hough, P.V.C., "Methods and Means for Recognizing Complex Patterns", U.S. Patents 3069654, Dec. 18, 1962.

22. Duda, R.O. & Hart, P.E., "Use of the Hough Transformation to Detect Lines and Curves in Pictures", Comm. of the ACM, Vol. 15, Jan. 1972, pp. 11-15

23. Cohen, M. & Toussaint, G.T., "Detection of lines and curves in Noisy Pictures", Preliminary Draft, May 1975.

24. Bennett, J.R. & MacDonald, J.S., "On the Measurement of Curvature in a Quantized Environment", IEEE Trans. on Comp., Vol. C-24, Aug. 1975. pp. 803-820.

25. Freeman, H., "Techniques for the Digital Computer Analysis of Chain Encoded Arbitrary Plane Curves", Proc. Nat. Elect. Conf., Chicago, Ill., Vol. 17, Oct. 1961. pp. 421-432.

26. Greanias, E.C., Meagher, P.F., Norman, R.J., Essinger, P., "The Recognition of Handwritten Numerals by Contour Analysis", IBM Journal 7, 1963. pp. 14-21.

27. Tomita, S., Nonguichi, S., "Recognition of Handwritten Katakana Characters", Electronics & Communications in Japan, Journal Inst. Electr. Comm. Engrs of Japan, Vol. 50, April 1967. pp. 174-183.

28. Groner, G.F., "Real Time Recognition of Handprinted Text", The RAND Corp., Santa Monica, Calif., RAND Memo. RM-5016-ARPA Oct. 1966, (Also in Proc. Fall Jt. Comp. Conf. 1966, pp. 591-601.)

29. Golshan, N. & Hsu, C.C., "A Recognition Algorithm for Handwritten Arabic Numerals", IEEE Trans. on System, Science and Cybernetics, Vol. SSC-6 3, July 1970. pp. 246-249.

30. Kuhl, F., "Classification and Recognition of Handprinted Characters", IEEE Internl. Conf. Record, 1963, Part 4. pp. 75-93.

31. Bernstein, M.I., "Computer Recognition of On-line Handwritten Characters", The RAND Corp., Santa Monica, Calif., RAND Memo. RM-3753-ARPA, Oct. 1964.

32. Ledley, R.S., Ruddle, F.H., Wilson, J.B., Belson, M., Albanan, J., "The Case of Touching and Overlapping Chromosomes", in Pictorial Pattern Recognition, Cheng, et al. (Eds), Wash., D.C., Thompson, 1968. pp. 87-97.

33. Rintala, W.M. & Hsu, C.C., "A Feature Detection Program for Pattern with Overlapping Cells", IEEE Trans. on System, Science and Cybernetics, Vol. SSC-4, March 1968. pp. 16-23.

34. Kasvand, T., "Recognition of Mono-Chromatic 2-Dimensional Objects", Report available from author at NRC of Canada, Ottawa, Ontario. 1968.

35. Yamada, S. & Fornanago, J.P., "Experimental Results for Local Filtering of Digitized Pictures", Dept. of Comp. Sci., Univ. of Illinois, Urbana, Ill., Report 184, June 1965.

36. Fehrer, E.V., "An Investigation of the Learning of Visually Perceived Forms", Amer. J. Psychol., 47, 1935. pp. 187-221.

37. Brill, E.L., "Character Recognition via Fourier Descriptors", IEEE Wescon Tech. Papers, Session 25, Qualitative P. R. through Image Shaping, 1968.

38. Zahn, C.T., "2-Dimensional Picture Description and Recognition via Curvature Points", IEEE Trans. on Comp., Vol. C-21, March 1973, pp. 269-281.

39. Ledley, R.S. & Rotolo, L.S., "Application of Pattern Recognition to Biomedical Problems", in Automatic Interpretation and Classification of Images, Grasselli, A., (Ed), 1969. Chap. 16, pp. 323

40. Ledley, R.S., "Automatic Pattern Recognition for Clinical Medicine", Proc. of IEEE, Vol. 57 11, Nov. 1969, pp. 2017-2025.

41. Ledley, R.S., Rotolo, L.S., Golab, T.J., Jacobsen, J.D., Ginsberg, M.D., Wilson, J.B., "FIDAC: Film Input to Digital Automatic Computer and Associated Syntax Directed Pattern Recognition Programming System", in Optical and Electro-Optical Information Processing, Tippert, et al, (Eds), MIT Press, Cambridge, Mass., 1965. pp. 591-614.

42. Shirai, Y., "A Heterarchical Program for Recognition of Polyhedra", Bull. Electrotech. Lab., Vol. 36-10, Tokyo, Japan, 1972.

43. Johnston, E. & Rosenfeld, A., "Angle Detection on Digital Curves", IEEE Trans. on Comp., Vol. C-22, 1973, pp. 875-878.

44. Ledley, R.S., "Use of the Computer in Biology and Medicine", McGraw Hill, N.Y., 1965. pp. 339-384.

45. Ledley, R.S., "Pattern Recognition Studies in the Biomedical Sciences", Sprint Jt. Comp. Conf., AFIPS Conf., Proc. 29, Spartan, Wash., D.C., 1966. pp. 411-430.

46. Gallus, G., Neurath, P.W., "Improved Computer Chromosome Analysis Incorporating Pre-processing and Boundary Analysis", 1970. (Available from the authors).

47. Brill, E.L., Heydorn, R.P. & Hill, J.D., "Some Approaches to Character Recognition for Postal Address Reader Applications", Proc. of Automatic P.R., May 1969. pp 19-40.

48. Kasvand, T., "Computer Simulation of Pattern Recognition by Using Cell Assemblies and Contour Projection", Proc. of IFAC, Tokyo Symp. System Engrg. Control System Design., 1965. pp. 203-210.

49. Kasvand, T., "Pattern Recognition Applied to the Counting of Nerve Fibre Cross-sections and Water Droplets", in Methodologies of Pattern Recognition, Watanabe, S., (Ed), Academic Press, N.Y., 1969. pp. 333-339.

50. Kasvand, T., "Pattern Recognition Applied to the counting of Nerve Fiber Cross-sections", National Research Council Report, 1969.

51. Barrow, H.G. & Popplestone, J.R., "Relational Descriptions in Picture Processing", in Machine Intelligence 6, Meltzer & Mitchie, (Eds), Edinburgh Univ. Press, Edinburgh, 1971, pp. 377-396.

52. Duda, R.O. & Hart, P.E., "Pattern Classification and Scene Analysis", John Wiley, N.Y., 1973. pp. 290-292.

53. Marsaglia, G., Course notes in "Computer Methods in Probability and Statistics - 308-690A", Fall 1973.

54. Freeman, H. & Glass, J.M., "On the quantization of Line-Drawing Data", IEEE Trans. on Systems, Science and Cybernetics, Jan. 1969

55. Freeman, H., & Morse, S.P., "On Searching a Contour Map for a Given Terrain Elevation Profile", Journal of the Franklin Institute, Vol. 284, July 1967.

56. Douglas, D.H. & Peuker, T.K., "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", The Canadian Cartographer, Vol 10, Dec. 1973. pp. 112-122.

57. Reumann, K. & Witkam, A.P.M., "Optimizing curve segmentation in Computer Graphics", International Computing Symposium 1973, North Holland Publishing Co., 1974.