A Web-Based Automatic Outline Capturing of Images

BY

MUHAMMAD FAISAL ABDUL RAZZAK

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE
In

COMPUTER SCIENCE

December 2001

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Muhammad Faisal Abdul Razzak** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE.

Thesis Committee

Dr. Muhammad Sarfraz (Chairman)

Najib A Kofahi

Dr. Najib Kofahi (Member)

Dr. Moataz Ahmed (Member)

Kanaan Faisal

Dr. Kanaan A. Faisal
(Department Chairman)

Prof. Osama A. Jannadi
(Dean of Graduate Studies)

9/2/2002

Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedicated to

My Parents

and

Sisters

ACKNOWLEDGEMENTS

In the name of Allah, Most Gracious, Most Merciful

Alhamdulillah , All praise is due to Allah, the Lord of the Worlds. The Beneficent, the Merciful. Master of the Day of Judgment. Thee do we worship and Thee aid we seek.

Peace and blessing of Allah be upon last Prophet Muhammad (sallallaahu 'alaihi wa sallam).

Acknowledgement is due to King Fahd University of Petroleum and Minerals for providing support for this work.

I would like to thank the following people:

- Dr. Muhammad Sarfraz, my thesis advisor, for his support and patience during the long time this thesis took to be finished. Thanks for his guidance and encouragement throughout this study.

- Dr. Najib Kofahi and Dr. Moataz Ahmed, my thesis committee members, for their interest and valuable suggestions.

- my parents and sisters for their love and support during my whole life.

- my friends, Murtaza and Owais, for helping me in developing the code.

- Shafayat, for his valuable tips; Khaled, for his arabic translation of abstract; Fareed and Abdul Shakoor for their moral support and guidance.

- all my friends and relatives here and in my home country (Pakistan).

- Last, but not least, I want to thank the people at KFUPM, for their generous help.

Contents

List of Tables

List of Figures

THESIS ABSTRACT

Name: Muhammad Faisal Abdul Razzak

Title: A Web-Based Automatic Outline Capturing of Images

Degree: MASTER OF SCIENCE

Major Field: Computer Science

Date of Degree: December 2001

This thesis introduces a system which will automatically capture the outline of the images. It is a World Wide Web based system for converting bitmap images which specifies each individual pixel in the image to outline which specifies the image as a collection of mathematically-specified curves.

Designing of curves has been one of the significant problems of Computer Graphics. There are number of applications where finding a mathematical curve description of the desired shape is beneficial. Font designing, Capturing Hand drawn images on computer screens, Data Visualization and Cartooning are main motivation towards curve designing.

An efficient and modular design approach has been taken to automate the process of capturing outline of images. The aim is to find the minimal number of significant points to fit the curve keeping in mind to optimize the closeness of fit between original digitized curve and our parametric curve.

The system is also deployed over the World Wide Web to present an easy-to-use interface to the user. The design of the interface system is such that the user feels a sense of control over the system by varying some parameters and seeing the results online. Moreover, in addition to the standard images available, a user can upload his own image for testing purposes.

King Fahd University of Petroleum and Minerals, Dhahran.
December 2001

خلاصة الرسالة

الاســـم : محمد فيصل عبدالرزاق

العنـــوان: التحديد التلقائي للرسم الكفافي من خلال الشبكة العنكبوتيه

الدرجـــة: الماجستير في العلوم

التخصص الرئيسي: علوم الحاسب الآلي

تاريخ التخرج: كانون الأول "ديسمبر" ٢٠٠١

يقدم هذا البحث نظاماً يقوم بتحديد الخط الكفافي للرسم بصورة تلقائية. فهذا النظام مبنياً على الشبكة العنكبوتية لأجل تحويل صور الخارطة النقطية **bitmap images** وذلك بتحديد كل نقطة ضوئية **pixel** موجوده على الرسم ومن ثم استخلاص الخط الكفافي في شكل مجموعة من المنحنيات المحدده رياضياً.

إن تصميم هذه المنحنيات تمثل أحد أبرز المشكلات في مجال الرسم الحاسوبي **Computer Graphics**. فهناك العديد من التطبيقات تستفيد من ايجاد وصف المنحنى الرياضي للشكل المطلوب مثل تصميم الخط، التعرف على الرسم اليدوي على شاشة الحاسب، **Data Visualization** والرسومات الكرتونية **Cartooning** ، فهذه التطبيقات كانت الحافز الرئيسي باتجاه تصميم هذه المنحنيات.

تم طرح طريقة فعالة وبتصميم معياري لآلية عملية التقاط الخط الكفافي للرسم. فالهدف هو الحصول على أقل عدد من النقاط البارزه والتي تناسب المنحنى المكوّن للرسم مع المحافظه على أكبر قدر من التقارب ين المنحنى الأصلي والمنحنى القياسي الذي نريد ايجاده.

تم أيضاً نشر هذا النظام على الشبكة العنكبوتيه ليقدّم للمستخدم بطريقة سلسة وسهلة. فواجهة البرنامج تتيح للمستخدم حرية التحكم على النظام وتغيير القياسات ورؤية النتيجة مباشرةً. إلى جانب الصور المتوفرة في النظام، يمكن للمستخدم أن يقوم بإنزال صورة من عنده لغرض الاختبار.

جامعـــة الملك فهـــد للبترول والمعـــادن

كانون اول "ديسمبر" ٢٠٠١

Chapter 1

Introduction

Designing of curves has been one of the significant problems of Computer Graphics. There are number of applications where finding a mathematical curve description of the desired shape is beneficial. Moreover the curves, which are robust and easy to control and compute, are of more interest. Font designing, Capturing Hand drawn images on computer screens, Data Visualization and Cartooning are main motivation towards curve designing. In curve designing, the cubic functions are the most powerful tools as they can define space curves and curves with inflections. The ideas such as end point interpolation, detection of characteristic points, least square method, recursive subdivision and parameterization can be used for curve fitting. There is a fair amount of literature on this topic particularly in [1, 2, 3]. The readers can refer to [4, 5, 6, 7, 8, 9, 10, 11] as well.

This thesis deals with an algorithm to eliminate the human interaction in obtaining the outline of original digital image and the system's appearance over the web. In the traditional approaches [12], initially, a digitized image is obtained either by scanning or from some electronic device. From this digitized image, boundary or contour of the image is obtained. Then corner points of the image are determined from contour. These corner points can be obtained by some interactive method or by some automated corner detection algorithm [13, 14, 15]. Optimal curve fitting is done by segmenting the contour outline at the corner points. Normally, the curve fitting methods are based on Bézier cubics [6].

1.1 Methodology

The methodology, in this thesis, mainly differs to the traditional approaches in various ways and follows mainly the work of [1, 3]. Since, some times corners are not detected precisely and some times only corner points are not sufficient to fit the curve which represent the original image, some more points are needed to achieve a best fit. These points are called the break points and are used along with corner points to achieve the best fit by using the least square method. The subdivision methodology is used to conquest the desired solution. Another major difference lies in the curve model for the description of design curve. The outline capturing technique, instead of traditional Bézier cubics, is based upon a cubic model which has attracting features to control the curve segments.

1.2 Advantages of a Web based System

The growth of information technology and the World Wide Web motivates us to make the system appear over the Web. The advantages of a web based system are:

- a system will be widely accessible on the Internet
- share the knowledge with the whole Internet Community
- good and easy-to-use user interface using HTML technology
- no additional software is required at the client side except Web browser
- no special hardware platform is required - any client computer capable of running a web browser will suffice
- user can test their own images and get a quick and faster response

The client/server model is used to achieve our goal. The Matlab Web Server, Apache and Tomcat Web Server along with HTML, JSP and JavaBeans technology are used to implement the system.

1.3 Organization of chapters

This thesis is organized as follows. Chapter 2 provides some background information on cubic curves and curve fitting. Also some web terminologies are described and explained. The corner detection algorithm is discussed in chapter 3. The details of fitting parametric cubic model are given in chapter 4. The web application structure and implementation is discussed in chapter 5. Chapter 6 shows some comparison and the results obtained. Chapter 7 ends with a conclusion and future work.

Chapter 2

Background

This chapter gives a brief introduction of parametric cubic curves and discusses the significance of parametric representation of curves along with the past approaches for curve fitting. A more detailed presentation of this material can be found in [11] and in [1, 3]. Also the work described in this thesis is concerned with the design of a web-based interface and presentation system. It is therefore appropriate to describe also the context in which the system will exist - the World Wide Web, as well as the major tools such as Common Gateway Interface. Java programming language etc.

2.1 Parametric Representation of Curves

Polylines are first-degree, piecewise approximation to curves. If the curves being approximated are not piecewise linear then a large number of endpoint coordinates must be created and stored to achieve reasonable accuracy. This is expensive in terms of both computation and space. Higher degree functions can be used to approximate the desired curves. The higher-degree approximations can be based on one of three methods. (The discussion is restricted to only 2-D curves.)

- Explicit Functions
- Implicit Equations
- Parametric Representation

2.1.1 Explicit Functions

We can express y as explicit function of x (e.g. $y = f(x)$). The difficulties with this approach are (1) it is impossible to get multiple values of y for a single value of x, so curves such as circle and ellipse must be represented by multiple curve segments; (2) such curves are not rotationally invariant and may require breaking a curve segment into many segments; (3) describing curves with vertical tangents is difficult, because a slope of infinity is difficult to represent.

2.1.2 Implicit Equations

We can choose to model curves as solutions to implicit equations of the form $f(x, y) = 0$; The difficulties with this approach are: (1) the given equation may have more solutions than required, for example, in modeling a circle, we might want to use $x^2 + y^2 = 1$, which is fine. But how do we model one-half of a circle? We must add constraints such as $x \geq 0$ which cannot be contained within the implicit equation; (2) if two implicit defined curve segments are joined together, it may be difficult to determine whether their tangent directions agree at join point.

2.1.3 Parametric Representation

The parametric form overcomes the problems caused by functional and implicit forms. The points on a curve are represented as ordered set of values: $p_i = [x_i, y_i]$. There are corresponding parametric functions that may be used to represent arbitrary curves; these are of the form $Q(t) = [x(t), y(t)]$. The parameter t takes the values from a specified range; conventionally from 0 to 1. A curve represented in this way can be thought of as the projection of the curve in 3-D, t as the third dimension, perpendicular to x and y plane. A unit circle can be represented in parametric form $Q(t) = [cos(t), sin(t)]$. Parametric form of curve allows multiple values of y for single or more values of x. Parametric curves replace the use of geometric slopes (which may be infinite) with parametric tangent vectors (which are never infinite). A parametric curve is approximated by *piecewise polynomial curves.* Cubic polynomials are most often used because lower-degree polynomials give little flexibility in controlling the shape of the curve, and higher degree polynomials require more computation and can introduce unwanted wiggles. No lower-degree polynomials allow a curve segment to interpolate (pass through) two specified end points with specified derivates at each end point.

2.2 Overview of Curves

This section describes various forms of parametric cubic curves and the past approaches of curve fitting [11],[16] and [1]. A detailed description and comparison of these curves can be found in [11].

2.2.1 Spline

The term spline goes back to the long flexible strips of metal used by draftsmen to lay out the surfaces of ships, cars, aircrafts etc. Weight attached to splines called "ducks", were used to pull the spline in various directions. These metal splines had second-order continuity. Figure (2.1) shows draftmen's spline. The mathematical equivalent of these strips, the natural cubic spline, is a C^0, C^1, and C^2 continuous cubic polynomial that interpolates the control points.

A huge amount of work has been done towards the construction of splines in the last two decades. Various families of splines for different objectives have been discovered.

Figure 2.1: Draftmen's Spline

Curve fitting by Spline

The use of splines for approximation of functions appears in [17]. This method applied to single-valued functions. Reinsch [17] used cubic splines to fit the curve. Grossman [18] described a parametric curve fitting technique. He noted that ordinary least-square fitting method does not take into account the ordering of points, and can not handle the multi-valued functions. His approach is to treat the data points parametrically but without piecewise representation. In order to fit complex curves, higher-degree polynomials were required. Almost the same technique is the basis of curve fitting systems in [2] and [1]. Generalized cubic splines [7, 19, 20] provide a reasonable methodology for curve fitting and curve designing. Rational cubic splines [21-68] are also good candidates for curve fitting.

2.2.2 B-Spline

B-splines consist of curve segments whose polynomial coefficients depend on just a few control points. This behavior is called *local control*. Thus, moving a control point affects only a small part of the curve. This local behavior is due to the fact that each vertex is associated with a unique basis function. The B-spline basis allows the order of basis function and hence the degree of the resulting curve to be changed without changing the number of defining polygon vertices.

Letting $P(t)$ be the position vectors along the curve as a function of the parameter t, a B-spline curve is given by

$$P(t) = \sum_{i=1}^{n+2} B_i N_{i,k}(t) \quad t_{min} \leq t < t_{max}; \quad 2 \leq k \leq n+1 \tag{2.1}$$

where B_i are the control points(position vectors) of the $n+1$ defining vertices and $N_{i,k}$ are the normalized B-spline basis functions. The i^{th} normalized B-spline basis function of order k (degree $k-1$) is defined by the Cox-deBoor recursion formulas.

$$N_{i,1} = \begin{cases} 1 & \text{if} \quad x_i \leq t < x_{i+1}; \\ 0 & otherwise. \end{cases}$$

and

$$N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}} \tag{2.2}$$

In above equations, x_i's are *knots* , satisfying the relation $x_i \leq x_{i+1}$. The parameter t varies from t_{min} to t_{max} along the cure $P(t)$. Since the denominators in the recursive calculations can have a value 0, the convention 0/0=0 is assumed.

Formally a B-spline curve is defined as a polynomial spline function of order k (degree $k-1$). $P(t)$ and its derivatives of order $1, 2, \ldots, k-2$ are all continuous over the entire curve. Thus, for example, a fourth-order B-spline curve is a piecewise cubic curve (C^2 continuity).

From equation (2.2) it is clear that the choice of knot vector (x_i) has a significance influence on the B-spline basis function $N_{i,k}(t)$ and hence on the resulting B-spline curve. Fundamentally three types of knot vectors are used: *uniform, open-uniform (or open) and non-uniform.*

Curve fitting by B-spline

If data points and knot values are known then control points can be computed and thereby B-spline curve can be fitted. This is know as inversion method. Wu, Abel and Greenburg [69] used B-spline for surface representation. The user of the system has to choose knots from a digitized curve then by inversion method B-spline curve is approximated. Yamaguchi [70] developed a method of an interactive curve fitting using B-spline. A least square cubic B-spline curve fitting technique is described by Dierchx [71]. A curve fitting technique based on quadratic B-splines is described by Yang [72]. Vercken [73] has also described a system based on B-spline. Some evolutionary methods, using B-splines, can be seen in [74, 75, 76, 77, 78]. These methods are based upon knot selection.

2.2.3 Hermite

Hermite curves are defined by two points P_1 and P_4 and two tangent vectors R_1 and R_4. Figure (2.2) shows a series of Hermite parametric cubic curves.

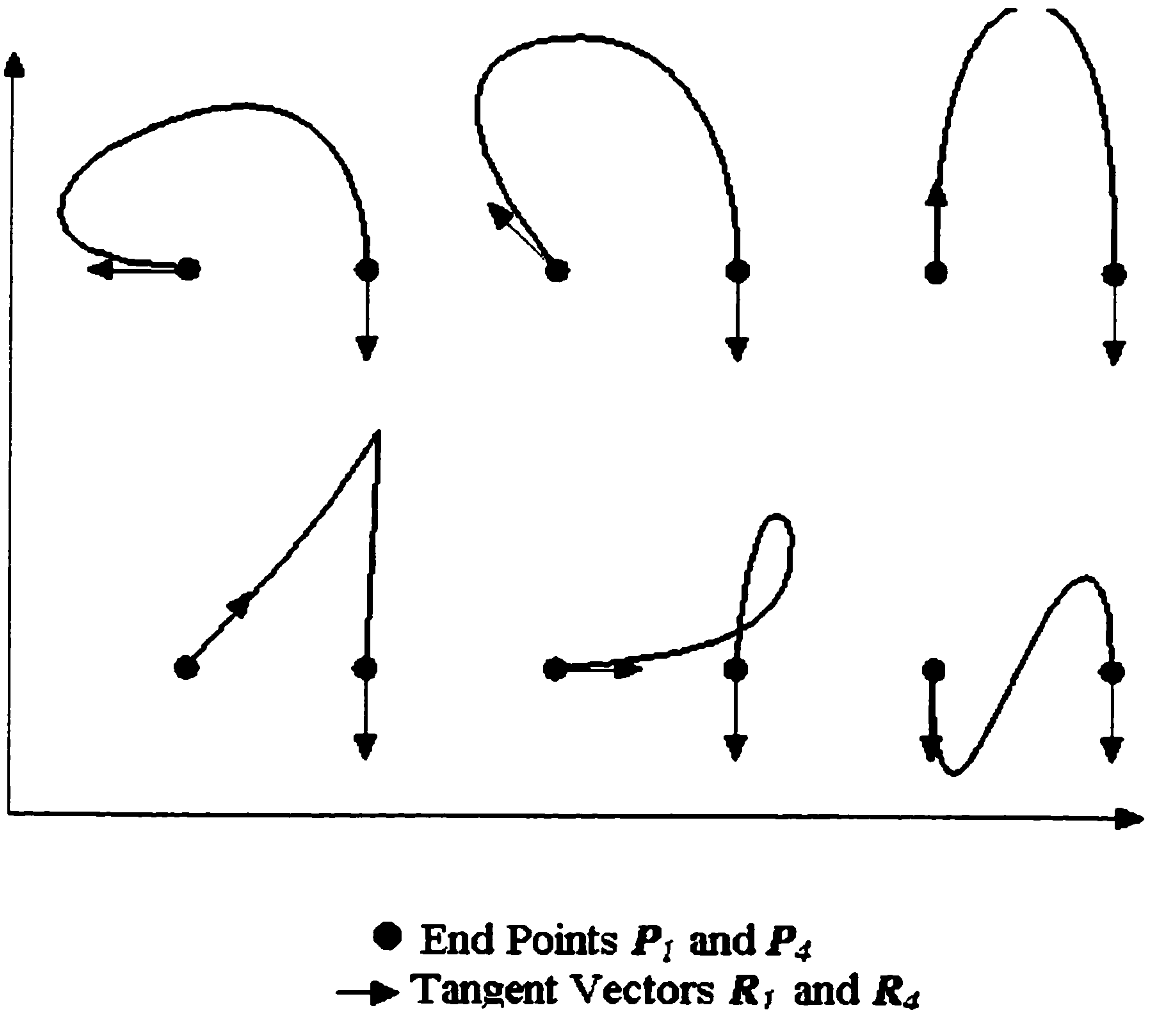

● End Points P_1 and P_4
→ Tangent Vectors R_1 and R_4

Figure 2.2: Hermite Curves

To find *Hermite basis matrix* M_H, which relates the Hermite geometry vector G_H to the polynomial coefficients, we write four equations, one for each of the constraints (P_1, P_4, R_1, R_4), in the four unknown polynomial coefficients, and then solve for the unknowns.

Defining G_{H_x} , the x component of the Hermite geometry matrix, as

$$G_{H_x} = \begin{bmatrix} P_{1_x} & P_{2_x} & P_{3_x} & P_{4_x} \end{bmatrix} \tag{2.3}$$

We express the curve as:

$$\begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x && (2.4) \\ &= G_{H_x}.M_H.T \\ x(t) &= G_{H_x}.M_H \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} && (2.5) \end{aligned}$$

The constraints on $x(0)$ and $x(1)$ are found by substituting $t = 0$ and $t = 1$ in equation (2.5)

$$x(0) = P_{1_x} \;=\; G_{H_x}.M_H \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{2.6}$$

$$x(1) = P_{4_x} \;=\; G_{H_x}.M_H \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T \tag{2.7}$$

To get $x'(t)$ we have to differentiate equation (2.5)

$$x'(t) \;=\; G_{H_x}.M_H \begin{bmatrix} 3t^2 & 2t & 1 & 0 \end{bmatrix} \tag{2.8}$$

Now we can find $x'(0)$ and $x'(1)$

$$x'(0) = R_{1_x} \;=\; G_{H_x}.M_H \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{2.9}$$

$$x'(1) = R_{4_x} \;=\; G_{H_x}.M_H \begin{bmatrix} 3 & 2 & 1 & 0 \end{bmatrix}^T \tag{2.10}$$

Now equation(2.3) can be written as follows

$$G_{H_x} \;=\; G_{H_x}.M_H \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \tag{2.11}$$

M_H is the inverse of 4x4 matrix in equation (2.11)

$$M_H = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \tag{2.12}$$

M_H can now be substituted in (2.5) to find $x(t)$. Similarly $y(t)$ and $z(t)$ can be found out. So we can write for Hermite polynomial $Q(t)$

$$Q(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix} = G_H.M_H.T \tag{2.13}$$

where

$$G_H = \begin{bmatrix} P_1 & P_4 & R_1 & R_4 \end{bmatrix}$$

Hermite blending function B_H can be written as

$$B_H = M_H.T \tag{2.14}$$

Now $Q(t)$ can be written as follows

$$Q(t) = G_H.B_H$$

$$Q(t) = (2t^3 - 3t^2 + 1)P_1 + (-2t^3 + 3t^2)P_4 + (t^3 - 2t^2 + t)R_1 + (t^3 - t^2)R_4 \quad (2.15)$$

These basis functions are shown in Figure(2.3)

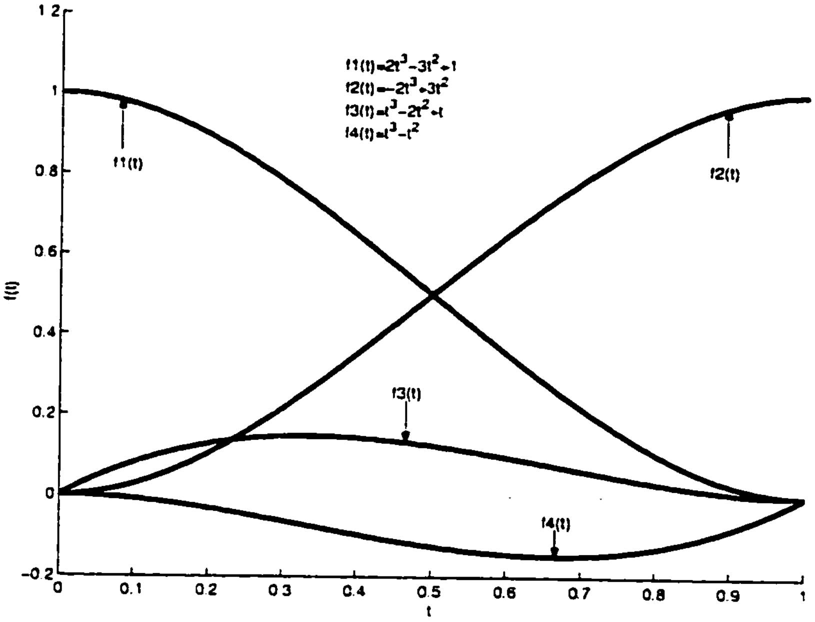


Figure 2.3: Hermite Basis Functions

Curve fitting by Hermite

Ichida [79] described curve fitting by one pass method. His method produces curve of C^1 continuity. Chong [80] described an automatic curve fitting algorithm, produces C^1 continuity. It is worth to mention here that the curve fitting model used in this thesis is a generalized form of Hermite cubics [81, 82]. The detailed description of this technique is given in chapter 4.

2.2.4 Bézier

Bézier curves are developed by Pierre Bézier [83] and [84] for use in designing automobiles at Renault.

The Bézier form of cubic polynomial curve segment has four control points P_0, P_1, P_2 and P_3. Two intermediate points P_1 and P_2 are not on the curve. The Bézier curve interpolates the two end control points P_0 and P_3 and approximates the two intermediate points P_1 and P_2. Some typical Bézier curves and their control polygon are shown in Figure(2.4)

The starting and end tangent vectors R_0 and R_3 are determined by P_0P_1 and P_2P_3 as follows:

$$R_0 = Q'(0) = 3(P_1 - P_0) \tag{2.16}$$

$$R_3 = Q'(1) = 3(P_3 - P_2) \tag{2.17}$$

The *Bézier geometry matrix* G_B is defined as

$$G_B = \begin{bmatrix} P_0 & P_1 & P_2 & P_3 \end{bmatrix} \tag{2.18}$$

The matrix M_{HB} defines the relation between *Hermite geometry matrix* G_H and

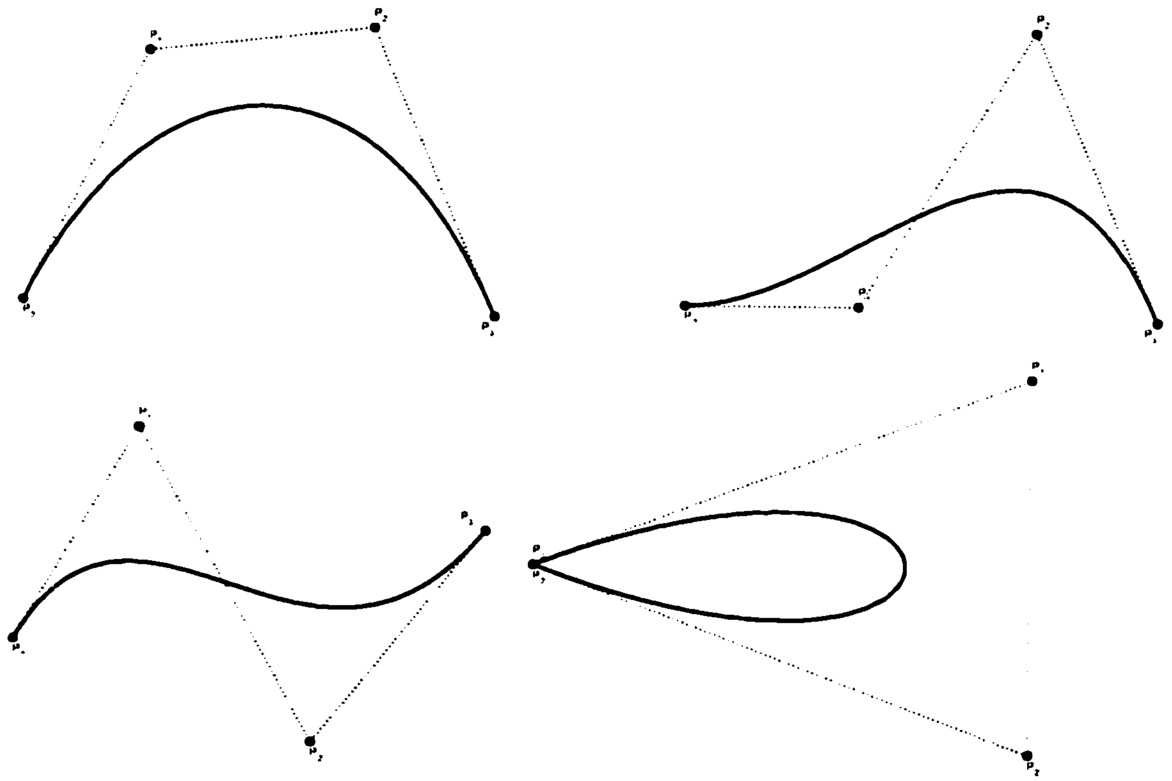


Figure 2.4: Bézier Curves and their Control Polygons

Bézier geometry matrix G_B:

$$\begin{aligned} G_H &= G_B.M_{HB} \\ &= \begin{bmatrix} P_0 & P_1 & P_2 & P_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{bmatrix} \end{aligned} \tag{2.19}$$

To find *Bézier basis matrix* M_B, we use equation (2.13) for the Hermite form

$$\begin{aligned} Q(t) &= G_H.M_H.T \\ &= (G_B.M_{HB}).M_H.T \\ &= G_B.(M_{HB}.M_H).T \end{aligned} \tag{2.20}$$

we define

$$M_B = M_{HB}.M_H = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.21}$$

Now we can write equation (2.20) as follows

$$Q(t) = G_B.M_B.T \tag{2.22}$$

Substituting the G_B, M_B and T yields

$$Q(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \tag{2.23}$$

Bernstein polynomials $B_0(t), B_1(t), B_2(t)$ and $B_3(t)$ can be defined as follows

$$\begin{aligned} B_0(t) &= (1-t)^3 \\ B_1(t) &= 3t(1-t)^2 \\ B_2(t) &= 3t^2(1-t) \\ B_3(t) &= 3t^3 \end{aligned} \tag{2.24}$$

Figure (2.5) shows Bézier basis functions.

General form of Bézier Bernstein polynomials

A Bézier curve of degree n is defined in terms of *Bernstein polynomials* as follows:

$$Q(t) = \sum_{i=0}^{n} P_i B_i^n(t) \tag{2.25}$$

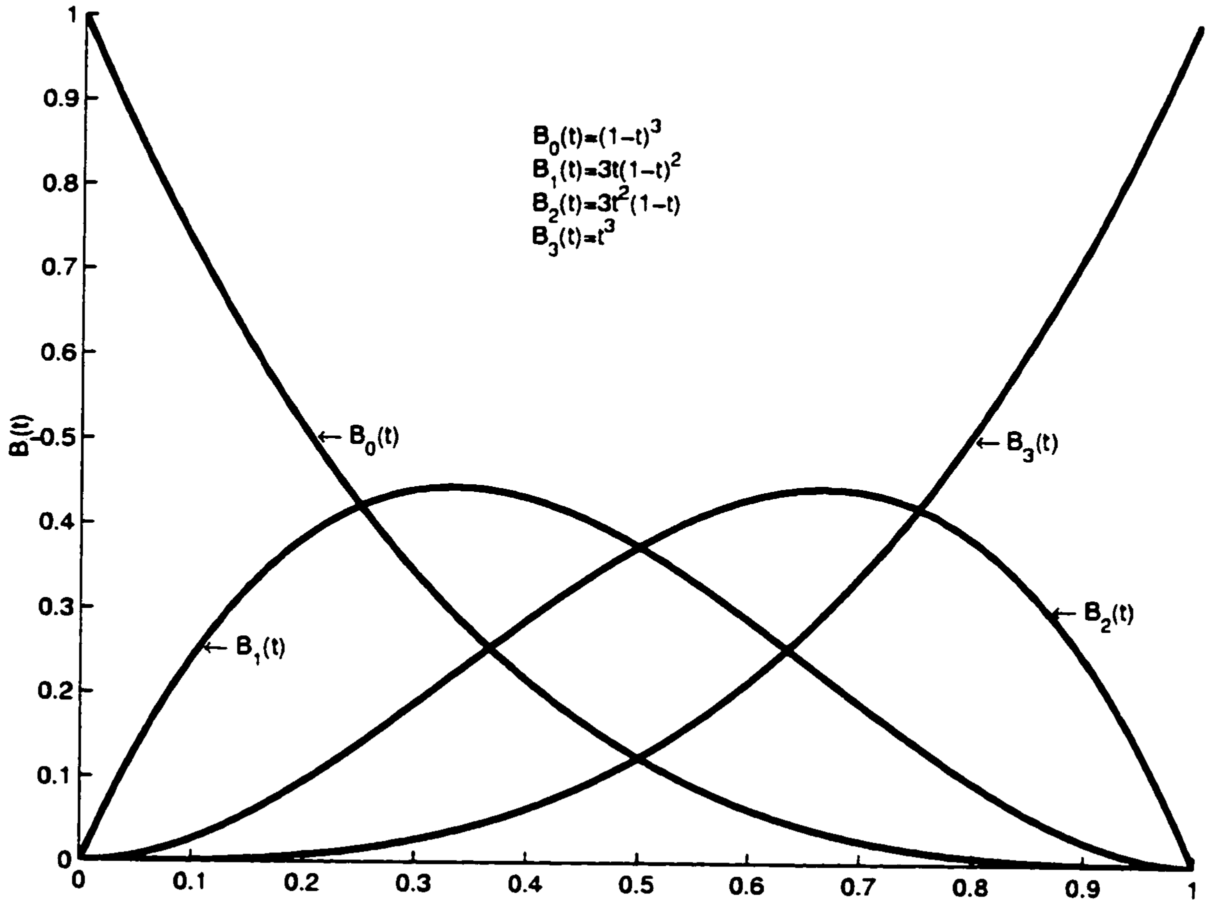


Figure 2.5: Bézier Basis Functions

$B_i^n(t)$ are *Bernstein polynomials*

$$B_i^n(t) = \left(\frac{n!}{i!(n-i)!}\right) t^i(1-t)^{n-i}, \quad i = 0, \ldots, n \tag{2.26}$$

P_i are the control points, and the sequence $P_0, P_2, \ldots, P_n$ form the control polygon of the curve segment.

Properties of Bézier Curve

Some of the important properties of Bézier curves are following:

Affine invariance: Bézier curves are invariant under affine maps. It means that we can rotate, translate, or scale the set of control points, and the curve associated with them will retain its relationship with the control points.

Convex hull: For $t \in [0, 1]$, the curve lies within the convex hull of control polygon. A simple consequence of the convex hull property is that a planar control polygon always generates a planar curve.

Endpoint interpolation: The Bézier curve passes through first and last control points. This can be easily verified by substituting $t = 0$ and $t = 1$ in Equation (2.23). This property ensures that the curve passes through significant points of each segment.

Derivative of Bézier Curve: Derivatives of Bézier curve are used to enforce continuities. The k^{th} derivative of Bézier curve is given by

$$\frac{d^kQ(t)}{dt^k} = \frac{n!}{(n-k)!}\sum_{i=0}^{n-k} \Delta^k P_i B_i^{n-k}(t) \tag{2.27}$$

where

$$\begin{aligned} \Delta^1 P_i &= \Delta P_i = P_{i+1} - P_i \\ \Delta^k P_i &= \Delta^k P_{i+1} - \Delta^{k-1} P_i \end{aligned}$$

2.3 Web Terminologies

The work described in this thesis is also concerned with the design of a web-based interface and presentation system. It is therefore appropriate to briefly mention the World Wide Web, as well as the major tools that are used in realizing the system - hypertext transfer protocol (HTTP), hypertext markup language (HTML), the Common Gateway Interface (CGI), and the Java programming language. For brevity, the readers can refer to [85, 86, 87, 88] or the electronic references at the end of this thesis.

2.3.1 The World Wide Web

The World Wide Web (also known as WWW or simply the Web) is a distributed hypermedia information system. It allows hypermedia information to be located on a network of computers called Web servers around the world, which are connected through the Internet. User friendly tools called Web clients or Web browsers make the Web easily accessible to anybody who has access to a personal computer and a modem. For that reason it has experienced tremendous growth over the past three years and has become the most used portion of the Internet. The World Wide Web uses a client-server model. Web clients are programs that send simple request to Web servers. Web servers reply to these requests by either sending documents or error codes.

The World Wide Web integrates several Internet protocols and defines its own protocol for hypertext transfer called Hypertext Transfer Protocol (HTTP). To define the location of objects on the Web, it uses Uniform Resource Locators (URLs). The standard format for hypertext documents on the World Wide Web is the Hypertext Markup Language (HTML). To enable building gateways to databases and other applications the Common Gateway Interface (CGI) was defined. For more interactivity on the Web JavaScript and Java allow the integration of executable content into Web pages. These components of the World Wide Web are introduced in the next sections.

2.3.2 Hypertext Transfer Protocol (HTTP)

The Hypertext Transfer Protocol (HTTP) is an application-level protocol that provides a fast and flexible mechanism to retrieve units of information distributed on the Web. It is based on the Internet protocol TCP/IP and HTTP servers usually listen on port TCP 80. The name Hypertext Transfer Protocol name might be misleading as information units transported by HTTP may not only be hypertext, but every kind of data such as images or sounds. Like most protocols on the Internet HTTP is a simple client-server protocol. In a typical transaction an HTTP-client opens a connection to a server and sends a request. The server replies to that request and the connection is closed again. Every request stands on its own and no state can be maintained. Thus, HTTP is a connectionless protocol.

2.3.3 Hypertext Markup Language (HTML)

Documents on the World Wide Web may be of arbitrary format, but the format generic to the Web is the Hypertext Markup Language (HTML). The HyperText Markup Language can be said to be the language of the World Wide Web. It allows text data objects to embed simple formatting information and references to other objects. It is a markup language used to describe and encapsulate the content of web documents. HTML hypertext documents are portable from one platform to another, as they are plain (ASCII) text files formatted according to a specification.

There is no standard for HTML. The World Wide Web Consortium (W3C) defines recommendations for HTML and the current recommendation is HTML 4.0 [89].

2.3.4 Common Gateway Interface (CGI)

Documents delivered by Web servers are static. They are files that do not change. As the Web evolved, its users started to look for dynamic documents created on the fly on the server. For that reason, the Common Gateway Interface (CGI) was introduced. CGI defines a standard for interfacing external programs with Web servers. These programs are often called CGI scripts, although they are not necessarily scripts and can be written in arbitrary programming languages. CGI scripts are addressed like normal documents by the use of URLs. They output an HTTP response in real time, which is then sent back to the client by the Web server. By this means, dynamic Web objects of arbitrary type can be created.

A CGI program is an executable program that can be run independently, but it is most often started by the client browser when posting a form. The purpose of the CGI program is then to receive the posted data and act upon it. The range of possible applications of the combination of html forms and CGI scripts is wide. Databases can be made accessible through the World Wide Web. Simulations can be run based on user input and their results presented to the user. Companies can provide credit card product order services. In short, the Web becomes an interactive medium instead of a static information presentation system.

2.3.5 Java

In some cases not even the power of the CGI is sufficient. In our system a user needs to upload the image file which is not currently supported by the Matlab server and the Matlab programs which acts as a CGI. To implement the upload feature, JSP and JavaBeans technologies are chosen which are nothing but Java.

Java is a new object-oriented programming language developed at Sun Microsystems. It is designed to be particularly useful as a programming language for the World Wide Web as it is made compact and secure. In the context of the web it is used for creating portable mini-applications, called applets, which are downloaded to a client as the result of a special tag in the source code for the currently viewed web page. The code of the applet is stored on the server until a browser fetches it, much in the same way as a graphic or a picture would be fetched. It is brought into the web browser where the browser then starts executing the code.

2.3.6 JSP and JavaBeans

$JavaServerPages^{TM}$ (JSP^{TM}) technology allows web developers and designers to rapidly develop and easily maintain, information-rich, dynamic web pages that leverage existing business systems. As part of the $Java^{TM}$ family, JSP technology enables rapid development of web-based applications that are platform independent. JavaServer Pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content.

JavaServer Pages (JSP) is a web-scripting technology similar to Microsoft Active Server Pages (ASP). It is a presentation layer technology and allows mixing static HTML content with server-side scripting to produce dynamic output. JSP uses Java as its default scripting language and all Java capabilities and APIs can be used. JSP uses JavaBeans to implement the business and data logic for an application. JSP provides tags and scripting platform for exposing the content generated or returned by the beans in HTML pages. Web designers generally develop JSP pages that uses beans for presentation logic. Java developers/programmers create JavaBeans to implement business logic. By separating the page logic from its design and display and supporting a reusable component-based design, JSP technology makes it faster and easier than ever to build web-based applications. Figure 2.6 shows how JSP and JavaBeans work on the Internet/Intranet.

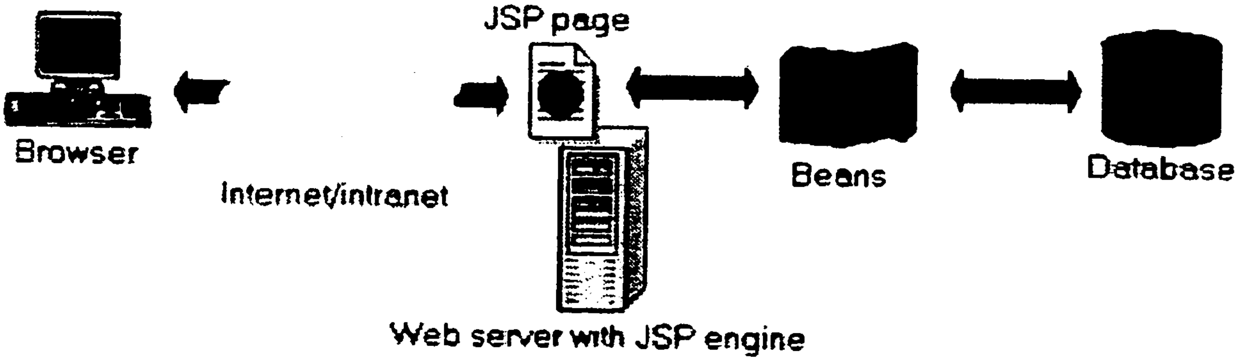


Figure 2.6: JSP and JavaBeans

2.3.7 Servers and Clients

There is a wide range of different Web servers available. Netcrafts Web server survey [90], in which more than 3 million servers were surveyed, shows that three server products hold more than 80 percent share of the market. The shareware Web server Apache holds more than 50 percent, the commercial products Microsoft Internet Information Server and Netscape Enterprise Server hold about 20 and 10 percent respectively. Microsoft's and Netscape's Web servers come with a range of tools, which help with information management and server administration, while Apache does not provide such tools. The Apache Web Server is chosen due to the simplicity in configuration and integration with the Matlab server.

There is a large number of World Wide Web clients or Web browsers, as they are often called, available. They differ in their support of the various features the World Wide Web offers. For example some support extensions made to HTTP and HTML, while others do not. The most common Web browsers are the Netscape Navigator and Microsoft Internet Explorer.

Chapter 3

Corner Detection

3.1 Introduction

Corners in digital images give important clues for shape representation and analysis. Since dominant information regarding shape is usually available at the corners, they provide important features for object recognition, shape representation and image interpretation. Corners are the robust features in the sense that they provide important information regarding objects under translation, rotation and scale change. If the corner points are identified properly, a shape can be represented in an efficient and compact way with sufficient accuracy in many shape analysis problem.

Corner detection schemes can be broadly divided into two categories based on their applications:

- binary (suitable for binary images) and

- gray level (suitable for gray level images)

Corner detection approaches for binary images usually involve segmenting the image into regions and extracting boundaries from those regions that contain them. The techniques for gray level images can be categorized into two classes: (a) Template-based and (b) gradient-based. The template-based technique utilizes correlation between a subimage and a template of a given angle. A corner point is selected by finding the maximum of the correlation output. Gradient-based techniques require computing curvature of an edge that passes through a neighborhood in a gray level image.

Corner detection is related to detection of high curvature points in planar curves. Various corner detection algorithms have been developed. Frequently cited approaches of corner detection are discussed and compared by [13] and [15]. For images a corner detection algorithm, based on the property of corners that the change of image intensity should be high in all directions is described by [91].

What is a corner? The notion of corner seems to be intuitively clear but no generally accepted mathematical definition exists, at least for digital curves. Different approaches give different but conceptually related computational definitions to a visual phenomenon. Since curvature measure is used to detect corner, therefore in this thesis some threshold value of angle is set to declare a point as corner point. If the corner points are detected precisely then the computation will be minimized in the later stages.

A curve consists of sequence of points $p_i = (x_i, y_i), i = 1, 2, \ldots, n$. For each point its cornerness (measure of corner strength) is determined. Points whose cornerness is above predefined threshold are declared as corner points. When processing a point p_i, the algorithm considers a number of subsequent points (p_k^+) and previous points (p_k^-) in the sequence, as candidates for potential corner points.

3.2 Corner Detection Algorithm

The proposed algorithm in [15] is used in this thesis where corner point is defined as a point where triangle of specified angle can be inscribed within specified distance from its neighbor points. The number of neighbor points to be checked are also predefined. It is a two pass algorithm. In the first pass the algorithm scans the sequence and selects candidate corner points. The second pass is post-processing to remove superfluous candidates.

First Pass: For each point p_i it is checked if triangle of specified size and angle is inscribed or not. Following three conditions are used.

$$d_{min}^2 \leq |p - p_k^+|^2 \leq d_{max}^2 \tag{3.1}$$

$$d_{min}^2 \leq |p - p_k^-|^2 \leq d_{max}^2 \tag{3.2}$$

$$\alpha \leq \alpha_{max} \tag{3.3}$$

where

p is the point under consideration for corner point.

p_k^+ is the k^{th} clockwise neighbor of p.

p_k^- is the k^{th} anti-clockwise neighbor of p.

Taking

$a = |p - p_k^+|$, the distance between p and p_k^+

$b = |p - p_k^-|$, the distance between p and p_k^-

$c = |p_k^+ - p_k^-|$, the distance between p_k^+ and p_k^-

The angle α can be computed by using cosine law

$$a^2 + b^2 - c^2 - 2ab\,cos\alpha = 0 \tag{3.4}$$

$$\alpha = cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \tag{3.5}$$

All the three conditions described in equations (3.1), (3.2) and (3.3) are necessary for the first pass. Now each point p may have zero, one or more than one alpha values. Among all alpha values, minimum value is taken as the alpha value of that point p.

Second Pass: Second pass removes some superfluous points. A candidate corner point p from the first pass is discarded if it has a sharper valid neighbor p_v:α(p) > α(p$_v$). A candidate point p_v is a valid neighbor of p if $|p - p_v|^2 \leq d^2_{max}$. As alternative definitions, one can use $|p - p_v|^2 \leq d^2_{min}$ or the points adjacent to p.

d_{min}, d_{max} and α_{max} are the parameters of the algorithm. Small values of d_{min} responds to fine corners. The upper limit d_{max} is necessary to avoid false sharp triangles formed by distant points in highly varying curves. α_{max} is the angle limit that determines the minimum sharpness accepted as high curvature.

The procedure of detecting corner points is given in the Flow chart of Figure (3.1).

Demonstration of the corner detection algorithm is shown from Figures (3.2) to (3.12). The pictures are selected from different categories to show the results with the default values of d_{min}, d_{max} and α_{max}. Alphabets of english and japanese language, flower, plane, pound sign and an arabic word 'Lillah' are the selected images. Also the arabic word 'Lillah' has been shown with different values of α_{max} to indicate the effect of changing α_{max}. Although the algorithm works fine and detects corner correctly in most of the images but it may not find all of the corners at the right position such as at the tail in the figure of plane (figure 3.11). But the method employed in this thesis is such that it will take care of these points and fit the curve correctly by inserting some break points in the later stage.

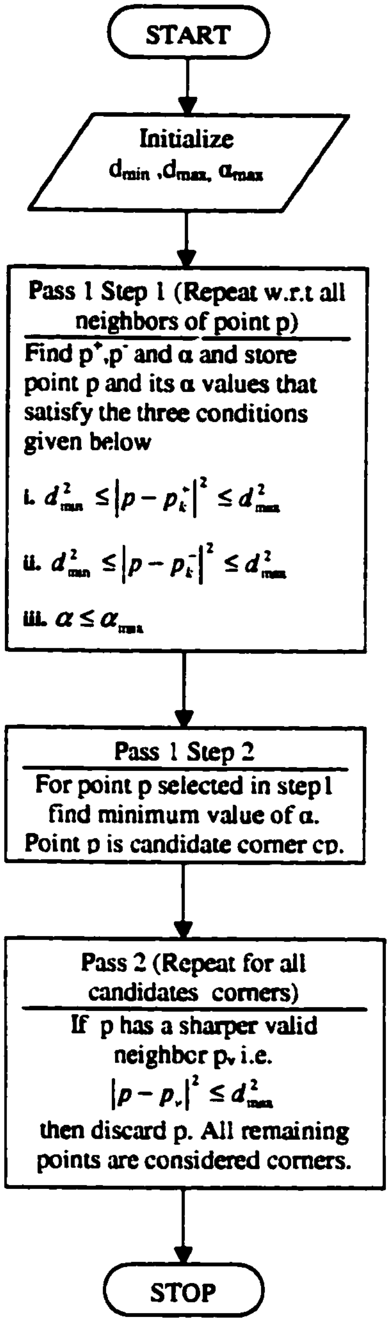

START
Initialize
d_{min} ,d_{max}, α_{max}
Pass 1 Step 1 (Repeat w.r.t all neighbors of point p)
Find p^+,p^- and α and store point p and its α values that satisfy the three conditions given below
i. $d_{min}^2 \le |p - p_k^+|^2 \le d_{max}^2$
ii. $d_{min}^2 \le |p - p_k^-|^2 \le d_{max}^2$
iii. $\alpha \le \alpha_{max}$
Pass 1 Step 2
For point p selected in step1 find minimum value of α. Point p is candidate corner cp.
Pass 2 (Repeat for all candidates corners)
If p has a sharper valid neighbcr p_v i.e.
$|p - p_v|^2 \le d_{max}^2$
then discard p. All remaining points are considered corners.
STOP

Figure 3.1: Flow Chart of Corner Detection Algorithm

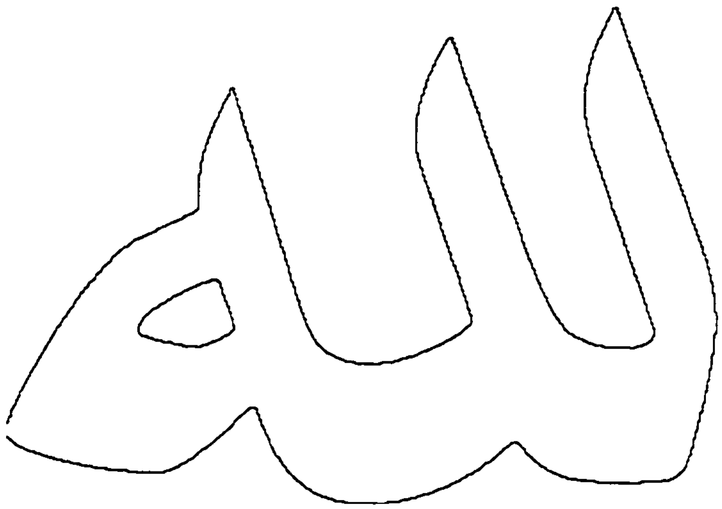


Figure 3.2: Contour of Image

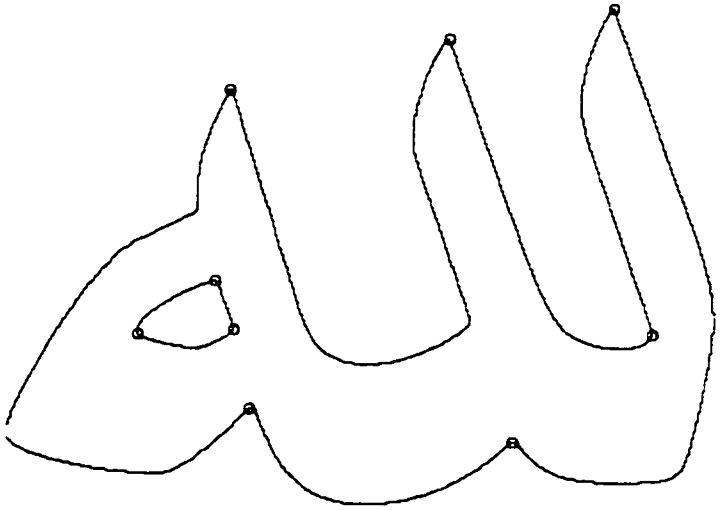


Figure 3.3: Corner candidates after Pass2. $d_{min} = 7$, $d_{max} = 9$, $\alpha_{max} = 120$

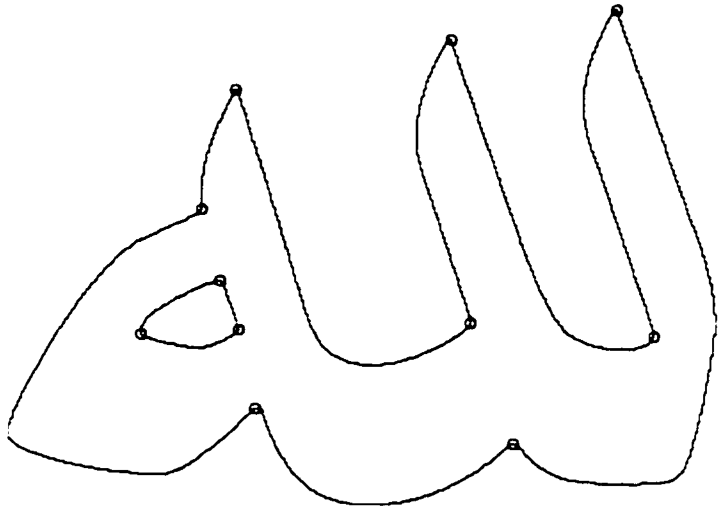


Figure 3.4: Corner candidates after Pass2. $d_{min} = 7$, $d_{max} = 9$, $\alpha_{max} = 130$

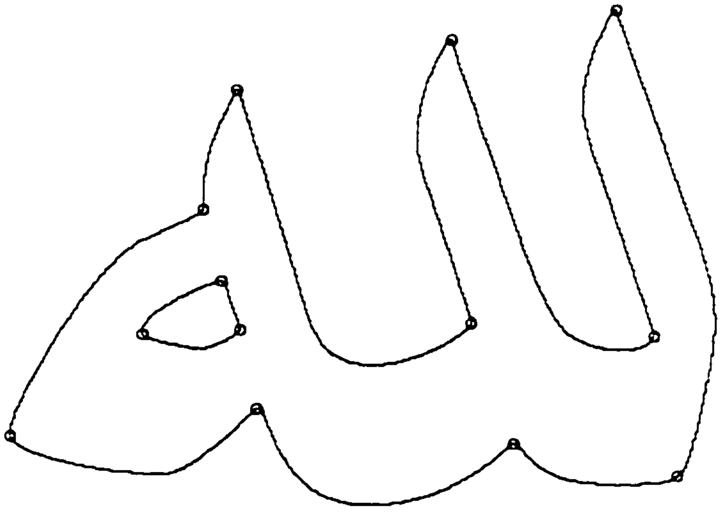


Figure 3.5: Corner candidates after Pass2. $d_{min} = 5$, $d_{max} = 8$, $\alpha_{max} = 150$

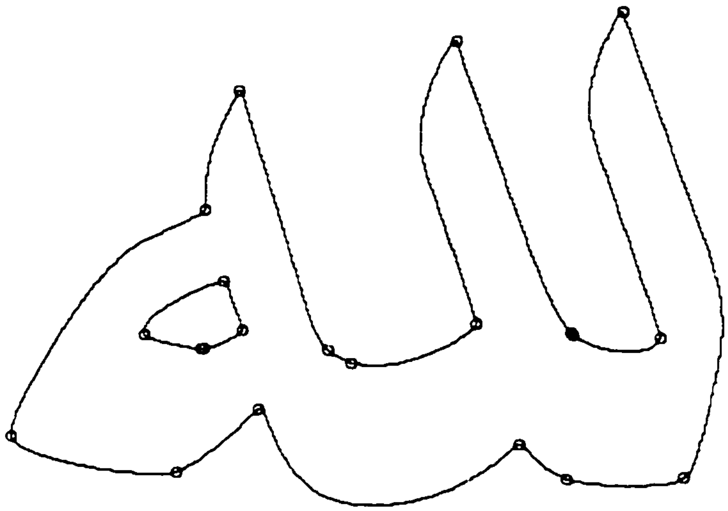


Figure 3.6: Corner candidates after Pass2. $d_{min} = 7$, $d_{max} = 9$, $\alpha_{max} = 160$

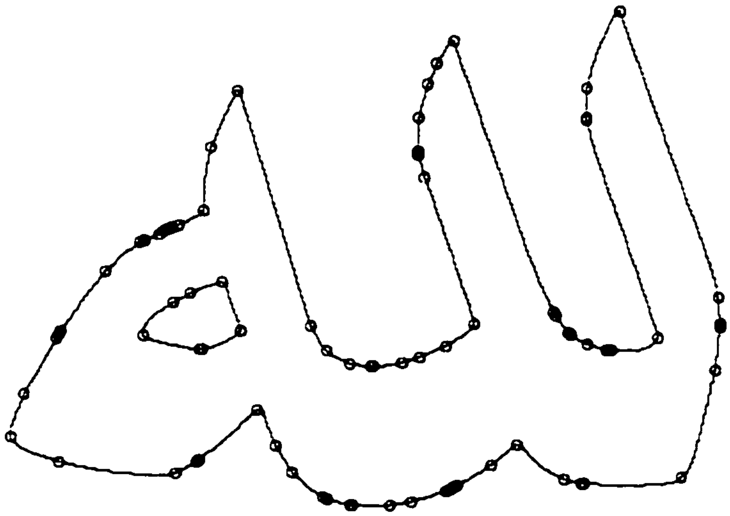


Figure 3.7: Corner candidates after Pass2. $d_{min} = 5$, $d_{max} = 8$, $\alpha_{max} = 170$

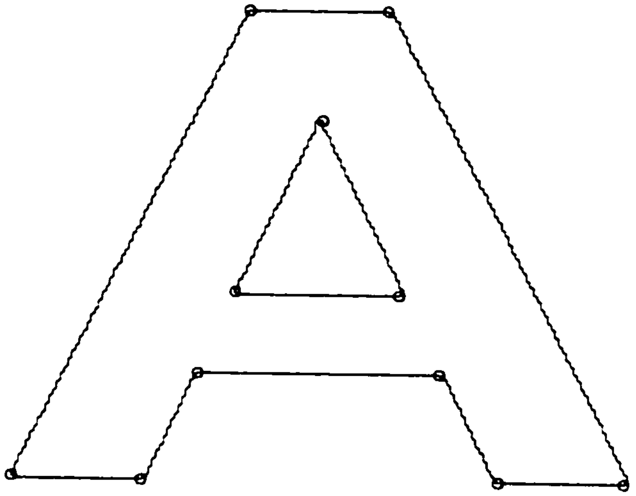


Figure 3.8: Corner candidates after Pass2. $d_{min} = 7$, $d_{max} = 9$, $\alpha_{max} = 150$

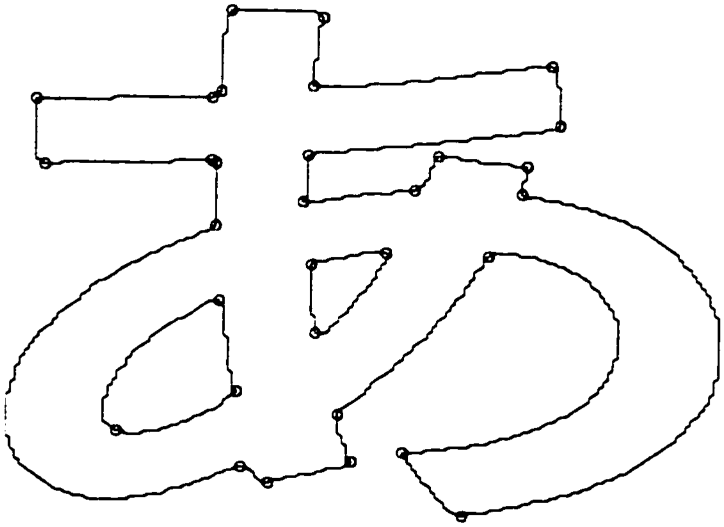


Figure 3.9: Corner candidates after Pass2. $d_{min} = 7$, $d_{max} = 9$, $\alpha_{max} = 150$

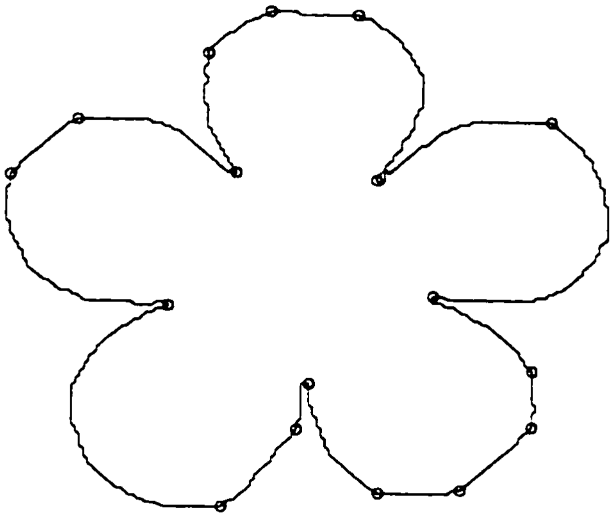


Figure 3.10: Corner candidates after Pass2. $d_{min} = 7$, $d_{max} = 9$, $\alpha_{max} = 150$

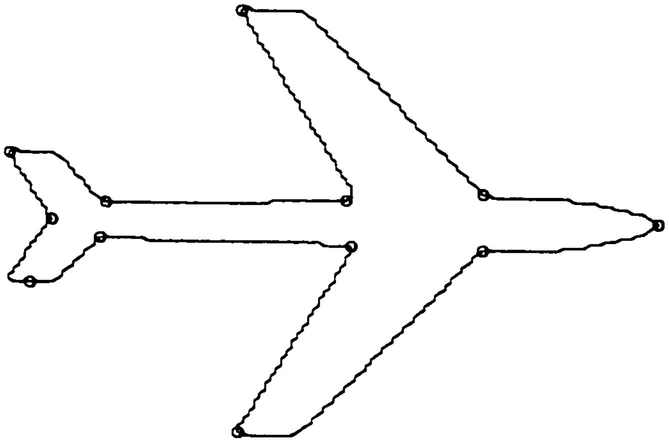


Figure 3.11: Corner candidates after Pass2. $d_{min} = 5$, $d_{max} = 8$, $\alpha_{max} = 150$

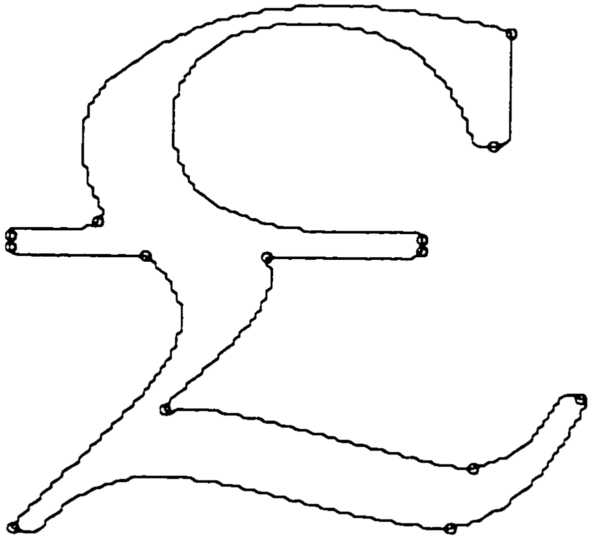


Figure 3.12: Corner candidates after Pass2. $d_{min} = 7$, $d_{max} = 9$, $\alpha_{max} = 150$

Chapter 4

Work Flow

4.1 Introduction

This chapter gives the details about the work flow of the system. Computer generated images are used in many applications, especially in font designing, data visualization etc. They also help in designing automobiles, ships, mechanical parts etc. The designer uses the iterative process to produce the physical design of the shape and it must be then conveyed to computer via user interface constrained to some parameters. To design a curve, many representation schemes exist nowadays. In traditional systems, the designer communicates the initial shape of the curve by specifying some control points interactively (with a mouse etc.). The computer then generates a curve whose shape depends on the control points. A curve may pass through (interpolate) control points, or may not, depending on the mathematical formulation defining the curve's shape. The shape of the curve depends on the placement of control points and the mathematical description that relates the curve [1,7,19,20-68]. Human interaction in designing the shape of the curve by moving some control points is not feasible as the users may not be familiar with the mathematics of the system. Automation needs to be done at various steps which we will see later in this chapter.

The work done in this thesis differs in various ways to the traditional approaches and the work done recently by Schneider [1] and Murtaza [3]. The main difference lies in the mathematical formulation of the curve which is a Hermite cubic or generalized Hermite cubic spline [81, 82] as compared to Bézier cubics used in previous work. There exists a continuity as well which was not present earlier. Segment breaking at the worst point error is also done using different approach discussed later in this chapter. However, the effect of noise filtering and reparameterization is not discussed in this thesis. Thus a modular and iterative procedure is developed to achieve the goal and the goal is to convert a bitmap image (which specifies each individual pixel in the image) to outline (which specifies the image as a collection of mathematically-specified curves). Specifically, the input is a bitmap image and the output is the outline of an image as shown in Figure (4.1).

The black box in Figure (4.1) consists of several steps. The output of one step is an input for another and all of them are automated to get the desired output. Figure 4.2 shows the steps involved in getting a good outline of bitmap image.

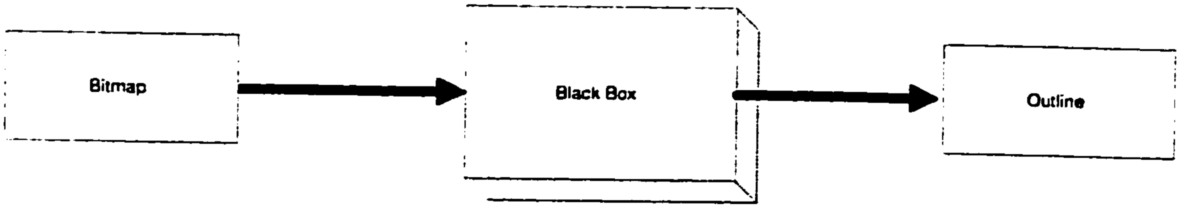


Figure 4.1: System showing input and output

4.2 Getting Digitized Image

Digitized image can be obtained directly from some electronic device or by scanning an image. The quality of digitized scanned image depends on various factors such as image on paper, scanner and attributes set during scanning. The quality of digitized image obtained directly from electronic device depends on the resolution of device, source of image, type of image etc. Some of the digitized images are shown in Figures (4.3) and (4.4).

4.3 Boundary Extraction

The next step is to find all the cyclical outlines (i.e., closed curves) in the bitmap image. The resulting list is called a pixel outline list which consists of the pixel coordinates of each edge on the outline. For example, the pixel outline list for an 'i' has two elements: one for the dot, and one for the stem. The pixel outline list for an 'o' also has two elements: one for the outside of the shape, and one for the

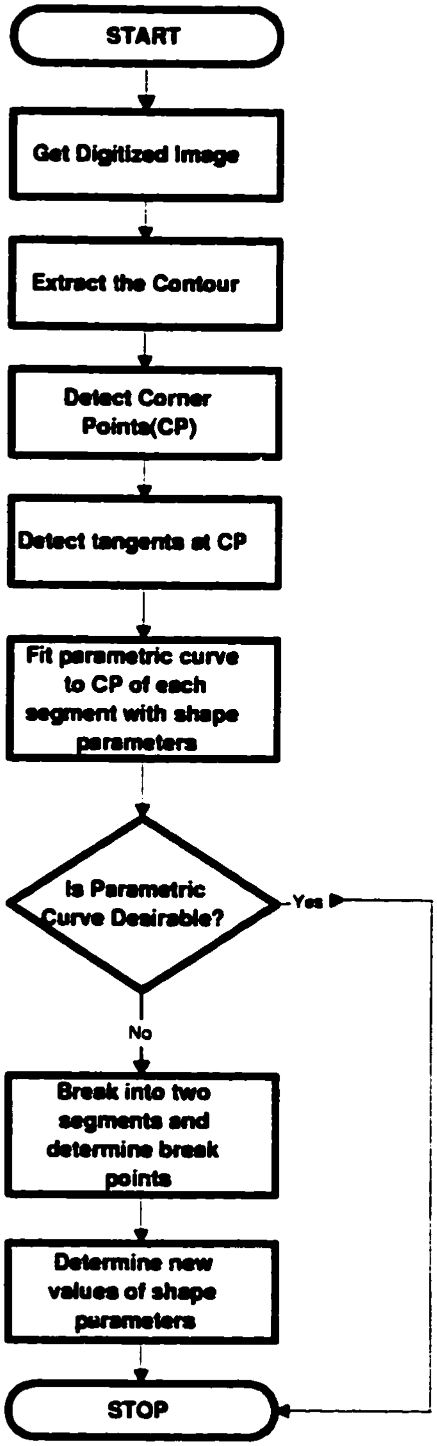

START
Get Digitized Image
Extract the Contour
Detect Corner Points(CP)
Detect tangents at CP
Fit parametric curve to CP of each segment with shape parameters
Is Parametric Curve Desirable?
Yes
No
Break into two segments and determine break points
Determine new values of shape parameters
STOP

Figure 4.2: Flow chart of the system

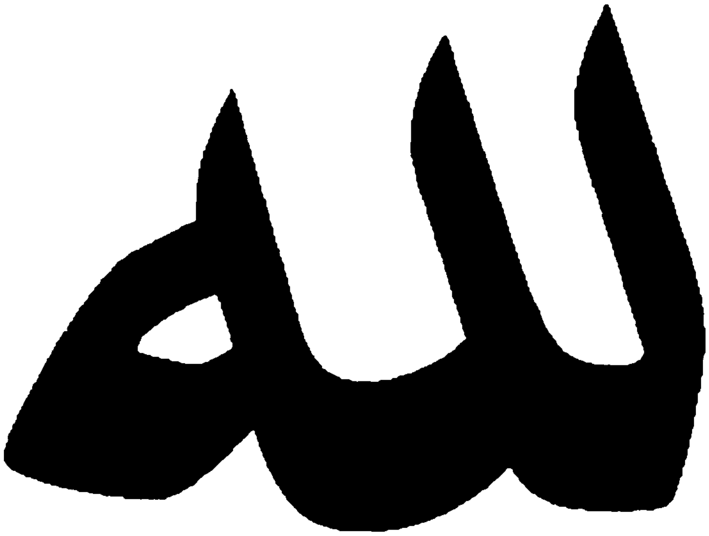


Figure 4.3: Digitized Image of ‘Lillah’ character

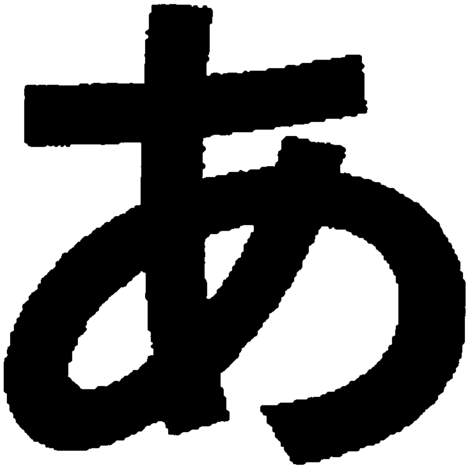


Figure 4.4: Digitized Image of ’Kanji’ character

inside.

Boundary of digitized image is extracted by using some boundary detection algorithm. There are numerous algorithms for detecting boundary [92]. The algorithm in [93] is used in this thesis. The input to this algorithm is a bitmap file. The algorithm returns number of pieces and for each piece number of boundary points and values of these boundary points $p_i = (x_i, y_i), i = 1, \ldots, N$. Figures 4.5 and 4.6 show detected boundary of the images of Figures 4.3 and 4.4 respectively. One of the text file obtained using the algorithm in [93] for figure 4.3 can be seen in appendix A.1.

Figure #	# of Pieces	# of Boundary Points
4.5	2	1522+116=1638
4.6	3	870+102+67=1039

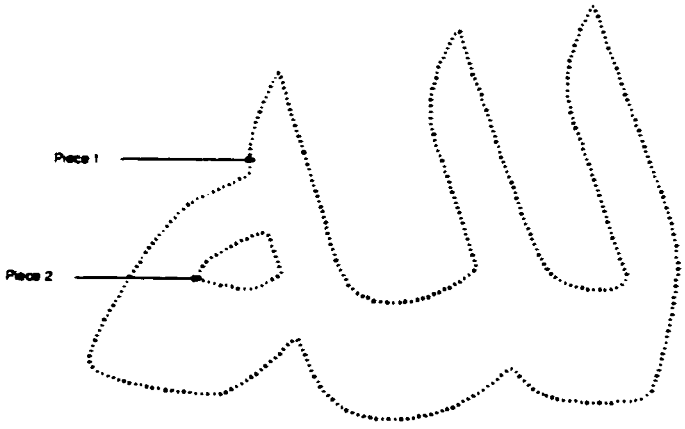


Figure 4.5: Detected Boundary consists of Two Pieces

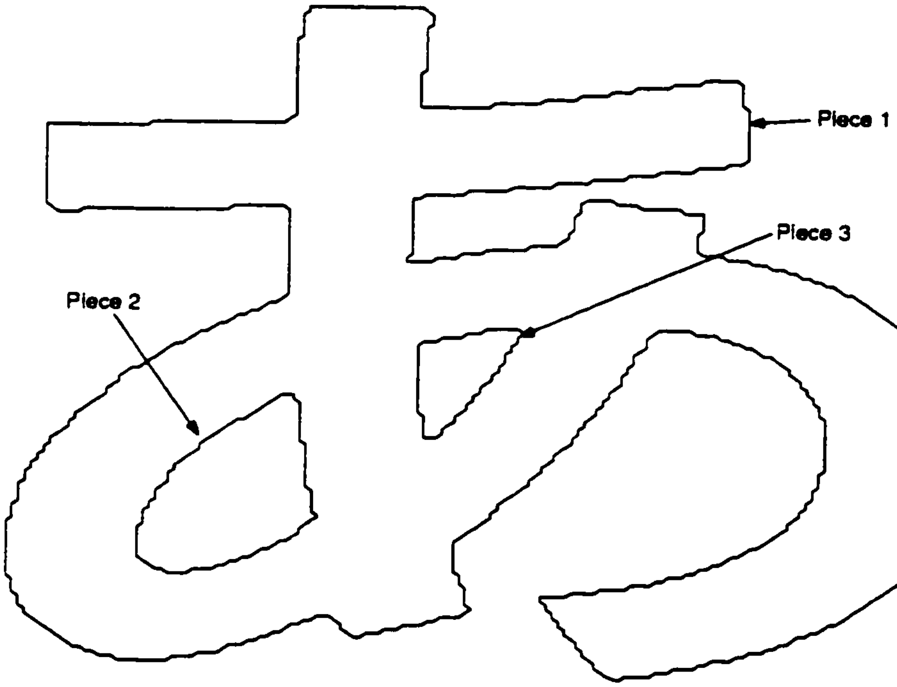


Figure 4.6: Detected Boundary consists of Three Pieces

4.4 Detecting Corner Points

The final goal is to fit continuous curves to the discrete bitmap image. We look for points where the outline makes such a sharp turn that a single curve cannot possibly fit well. These points are sometimes called characteristic points or corner points or significant points. If the corner points are detected precisely then the computation will be minimized in the later stages. The detail of this step is mentioned in chapter 3.

4.5 Cubic Interpolant

We divide the whole set of contour points into groups called segments such that each segment lies between the two consecutive corner points. The parametric representation of curves is then used to fit the curve piecewisely. Each segment of the overall curve is given by functions x and y which are cubic polynomials in the parameter t. Cubic polynomials are most often used because lower-degree polynomials give too little flexibility in controlling the shape of the curve, and higher-degree polynomials can introduce unwanted wiggles and also require more computation. No lower-degree representation allows a curve segment to interpolate (pass through) two specified endpoints with specified derivatives at each endpoint. Given a cubic polynomial with its four coefficients, four knowns are used to solve for the unknown coefficients. The four knowns might be the two endpoints and the derivatives at the endpoints.

Let $\mathbf{F}_i,\mathbf{F}_{i+1},i \in Z$ be the two end characteristic points given at the distinct knots $t_i,t_{i+1},i \in Z$ with interval spacing $h_i = t_{i+1} - t_i > 0$. Also let $\mathbf{D}_i,\mathbf{D}_{i+1},i \in Z$ denote the first derivative values defined at the knots. Then the generalized form of the cubic is defined by

$$\mathbf{P}|_{(t_i,t_{i+1})}(t) = (1-t)^3\mathbf{F}_i + 3t(1-t)^2\mathbf{V}_i + 3t^2(1-t)\mathbf{W}_i + t^3\mathbf{F}_{i+1}, \tag{4.1}$$

where

$$\mathbf{V}_i = \mathbf{F}_i + v_i h_i \mathbf{D}_i/3, \qquad \mathbf{W}_i = \mathbf{F}_{i+1} - w_i h_i \mathbf{D}_{i+1}/3 \tag{4.2}$$

The interpolation conditions are as follows:

$$\left.\begin{array}{ll} \mathbf{P}(t_i) = \mathbf{F}_i, & \mathbf{P}(t_{i+1}) = \mathbf{F}_{i+1} \text{ and} \\ \mathbf{P}^{(1)}(t_i) = v_i\mathbf{D}_i, & \mathbf{P}^{(1)}(t_{i+1}) = w_i\mathbf{D}_{i+1}, i \in Z. \end{array}\right\} \tag{4.3}$$

Equation (4.1) can be rewritten as

$$\mathbf{P}|_{(t_i,t_{i+1})}(t) = R_{0,i}(t)\mathbf{F}_i + R_{1,i}(t)\mathbf{V}_i + R_{2,i}(t)\mathbf{W}_i + R_{3,i}(t)\mathbf{F}_{i+1}, \tag{4.4}$$

where

$$\left.\begin{aligned} R_{0,i}(t) &= (1-t)^3, \\ R_{1,i}(t) &= 3t(1-t)^2, \\ R_{2,i}(t) &= 3t^2(1-t), \\ R_{3,i}(t) &= t^3 \end{aligned}\right\} \tag{4.5}$$

The functions $R_{j,i}$, $j = 0, 1, 2, 3$ are Bernstein Bézier like basis functions, such that

$$\sum_{j=0}^{3} R_{j,i}(t) = 1. \tag{4.6}$$

From the Bernstein-Bézier theory it follows that the curve segment $\mathbf{P}|_{[t_i,t_{i+1}]}$ lies in the convex hull of the control points $\{\mathbf{F}_i, \mathbf{V}_i, \mathbf{W}_i, \mathbf{F}_{i+1}\}$ and is variation diminishing with respect to the control polygon joining these points.

We observe the following properties defined by equation (4.1) and (4.2):

- The curve always passes through $\mathbf{F}_i$ and $\mathbf{F}_{i+1}$
- If $v \rightarrow 0$, then the curve exhibits the biased tension behavior to the left and is pulled towards the control point $\mathbf{F}_i$
- If $w \rightarrow 0$, then the curve exhibits the biased tension behavior to the right and is pulled towards the control point $\mathbf{F}_{i+1}$

- If $v, w \to 0$, then the curve exhibits the internal tension behavior and approaches to the linear interpolant

$$\mathbf{P}(t) = (1 - t)\mathbf{F}_i + t\mathbf{F}_{i+1}$$

- For $0 < v, w \leq 1$, the cubic ensures the convex hull property and hence the curve segment is more flexible than the traditional cubic Bézier curve.

4.5.1 Parameterization

There are number of parameterizations techniques in the literature [94, 95] such as uniform parameterization, linear or chord length parameterization, parabolic parameterization and cubic parameterization. Chord length parameterization scheme can be adapted to continuous sets of points and this scheme is used to estimate the parametric value t associated with each point $p_i = (x_i, y_i)$.

$$t_i = \begin{cases} 0 & \text{if } \ i = 1; \\ \frac{|p_1p_2| + |p_2p_3| + \cdots + |p_{i-1}p_i|}{|p_1p_2| + |p_2p_3| + \cdots + |p_{n-1}p_n|} & \text{if } \ 2 \leq i \leq n - 1; \\ 1 & \text{if } \ i = n \end{cases}$$

It should be noted that t_i is in normalized form and varies from 0 to 1, and hence h_i in our case is always equal to 1.

4.5.2 Estimation of TangentVectors

We define a distance based choice for tangents vectors $\mathbf{D}_i$'s at $\mathbf{F}_i$'s as follows:

For open curves:

$$\left.\begin{aligned} \mathbf{D}_0 &= 2(\mathbf{F}_1 - \mathbf{F}_0) - (\mathbf{F}_2 - \mathbf{F}_0)/2, \\ \mathbf{D}_n &= 2(\mathbf{F}_n - \mathbf{F}_{n-1}) - (\mathbf{F}_n - \mathbf{F}_{n-2})/2, \\ \mathbf{D}_i &= a_i(\mathbf{F}_i - \mathbf{F}_{i-1}) + (1 - a_i)(\mathbf{F}_{i+1} - \mathbf{F}_i), i = 1, .., n-1. \end{aligned}\right\} \tag{4.7}$$

For close curves:

$$\left.\begin{aligned} \mathbf{F}_{-1} = \mathbf{F}_{n-1}, \mathbf{F}_{n+1} = \mathbf{F}_1, \\ \mathbf{D}_i = a_i(\mathbf{F}_i - \mathbf{F}_{i-1}) + (1 - a_i)(\mathbf{F}_{i+1} - \mathbf{F}_i), i = 0, .., n. \end{aligned}\right\} \tag{4.8}$$

where

$$a_i = \frac{|\mathbf{F}_{i+1} - \mathbf{F}_i|}{|\mathbf{F}_{i+1} - \mathbf{F}_i| + |\mathbf{F}_i - \mathbf{F}_{i-1}|}, i = 0, .., n. \tag{4.9}$$

Also the tangents $\mathbf{D}_i$ and $\mathbf{D}_{i+1}$ in equation (4.1) can be computed by using Least Square method. The number of cases that are discussed in the next section uses both the techniques to calculate the tangents.

4.5.3 Optimal Design Curve

Now we have five cases to consider and they are discussed below one by one.

Case 1: $v_i = w_i = 1$ and tangents are estimated as described in §4.5.2.

It is a C^1 Hermite Spline curve and can be treated as a default design curve. This is a simplest case to consider and requires less computation initially. Since v_i,w_i and h_i are equal to 1, we have:

$$\mathbf{V}_i = \mathbf{F}_i + \mathbf{D}_i/3 \quad \text{and} \quad \mathbf{W}_i = \mathbf{F}_{i+1} - \mathbf{D}_{i+1}/3.$$

The results of this case for the four images are shown in chapter 6.

Case 2: $v_i = w_i$ and tangents are estimated as described in §4.5.2.

In this case the shape parameters are treated as equal and we have to solve the equation for only one variable using Least Square method. Suppose, for $i = 0, 1, 2, ..., n-1$, the data segments

$$\mathbf{P}_{i,j} = (x_{i,j}, y_{i,j}), \qquad j = 1, 2, ..., m_i \tag{4.10}$$

are given as the ordered sets of the universal set of the data points. Then the squared sums S_i's of distances between $\mathbf{P}_{i,j}$'s and their corresponding parametric points $\mathbf{P}(t_j)$'s on the curve are computed as:

$$S_i = \sum_{j=1}^{m_i} [\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}]^2, \qquad i = 0, 1, 2, ..., n-1 \tag{4.11}$$

where the parameterization over u's is in accordance with the chord length parameterization.

For the best fitting of the curve to the given data, we have to find out the parameter v_i so that sums S_i's are minimal and this can be done by using least square approximation. The minimum of S_i's occur if partial derivative of S_i's with respect to v_i's becomes zero. We have

$$\frac{\partial S_i}{\partial v_i} = 2\sum_{j=1}^{m_i} \frac{\partial \mathbf{P}_i(u_{i,j})}{\partial v_i} . [\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, \qquad i = 0, 1, 2, ..., n-1,$$

Since

$$\frac{\partial \mathbf{P}_i(u_{i,j})}{\partial v_i} = \frac{h_i}{3}(R_{1,i}(u_{i,j})\mathbf{D}_i - R_{2,i}(u_{i,j})\mathbf{D}_{i+1}),$$

therefore

$$\frac{\partial S_i}{\partial v_i} = \frac{2h_i}{3}\sum_{j=1}^{m_i}(R_{1,i}(u_{i,j})\mathbf{D}_i - R_{2,i}(u_{i,j})\mathbf{D}_{i+1}).[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, i = 0, 1, 2, ..., n-1, \tag{4.12}$$

Equation (4.12) can be simplified as

$$\begin{aligned}&\mathbf{D}_i.\sum_{j=1}^{m_i}(R_{1,i}(u_{i,j})\mathbf{P}_i(u_{i,j}) - \mathbf{D}_{i+1}.\sum_{j=1}^{m_i}(R_{2,i}(u_{i,j})\mathbf{P}_i(u_{i,j}) = \\ &\quad \mathbf{D}_i.\sum_{j=1}^{m_i}(R_{1,i}(u_{i,j})\mathbf{P}_{i,j} - \mathbf{D}_{i+1}.\sum_{j=1}^{m_i}(R_{2,i}(u_{i,j})\mathbf{P}_{i,j}, \qquad i = 0, 1, 2, ...n-1.\end{aligned} \tag{4.13}$$

This is equivalent to (by replacing $\mathbf{P}_i(u_{i,j})$ from equation (4.1)):

$$\begin{aligned}&\mathbf{D}_i.\mathbf{F}_i(a_{0,i} + a_{1,i}) + \mathbf{D}_i.\mathbf{F}_{i+1}(a_{2,i} + a_{3,i}) - \mathbf{D}_i.\sum_{j=1}^{m_i}(R_{1,i}(u_{i,j})\mathbf{P}_{i,j} - \mathbf{D}_{i+1}.\mathbf{F}_i(a_{2,i} + a_{4,i}) - \\ &\quad \mathbf{D}_{i+1}.\mathbf{F}_{i+1}(a_{5,i} + a_{6,i}) + \mathbf{D}_{i+1}.\sum_{j=1}^{m_i}(R_{2,i}(u_{i,j})\mathbf{P}_{i,j} = \\ &\quad \frac{h_i}{3}v_i(-|\mathbf{D}_i|^2 a_{1,i} + 2\mathbf{D}_i.\mathbf{D}_{i+1}a_{2,i} - |\mathbf{D}_{i+1}|^2 a_{5,i}), \qquad i = 0, 1, 2, ...n-1.\end{aligned} \tag{4.14}$$

where

$$\left.\begin{aligned}
a_{0,i} &= \sum_{j=1}^{m_i} R_{0,i}(u_{i,j})R_{1,i}(u_{i,j}),\\
a_{1,i} &= \sum_{j=1}^{m_i} R_{1,i}^2(u_{i,j}),\\
a_{2,i} &= \sum_{j=1}^{m_i} R_{1,i}(u_{i,j})R_{2,i}(u_{i,j}),\\
a_{3,i} &= \sum_{j=1}^{m_i} R_{1,i}(u_{i,j})R_{3,i}(u_{i,j}),\\
a_{4,i} &= \sum_{j=1}^{m_i} R_{0,i}(u_{i,j})R_{2,i}(u_{i,j}),\\
a_{5,i} &= \sum_{j=1}^{m_i} R_{2,i}^2(u_{i,j}),\\
a_{6,i} &= \sum_{j=1}^{m_i} R_{2,i}(u_{i,j})R_{3,i}(u_{i,j}),
\end{aligned}\right\} \qquad (4.15)$$

Equation (4.14) leads to the solution:

$$v_i = \frac{3(A_{2,i} - A_{3,i} + A_{4,i})}{h_i.A_{1,i}}, \qquad i = 0, 1, 2, ...n - 1. \qquad (4.16)$$

where

$$\left.\begin{aligned}
A_{1,i} &= |\mathbf{D}_i|^2 a_{1,i} - 2\mathbf{D}_i.\mathbf{D}_{i+1} a_{2,i} + |\mathbf{D}_{i+1}|^2 a_{5,i},\\
A_{2,i} &= \mathbf{D}_i . \sum_{j=1}^{m_i} R_{1,i}(u_{i,j})\mathbf{P}_{i,j} - \mathbf{D}_{i+1} . \sum_{j=1}^{m_i} R_{2,i}(u_{i,j})\mathbf{P}_{i,j},\\
A_{3,i} &= \mathbf{D}_i.\mathbf{F}_i(a_{0,i} + a_{1,i}) + \mathbf{D}_i.\mathbf{F}_{i+1}(a_{2,i} + a_{3,i}),\\
A_{4,i} &= \mathbf{D}_{i+1}.\mathbf{F}_i(a_{2,i} + a_{4,i}) + \mathbf{D}_{i+1}.\mathbf{F}_{i+1}(a_{5,i} + a_{6,i}).
\end{aligned}\right\} \qquad (4.17)$$

Thus the curve fitted using the above values of v_i's will be a good candidate of best fit.

Case 3: $v_i \neq w_i$ and tangents are estimated as described in §4.5.2.

In this case the shape parameters are not treated equal and we have to solve the equation for two variables using Least Square method. Suppose, for $i = 0, 1, 2, ..., n-1$, the data segments

$$\mathbf{P}_{i,j} = (x_{i,j}, y_{i,j}), \qquad j = 1, 2, ..., m_i \tag{4.18}$$

are given as the ordered sets of the universal set of the data points. Then the squared sums S_i's of distances between $\mathbf{P}_{i,j}$'s and their corresponding parametric points $\mathbf{P}(t_j)$'s on the curve are computed as:

$$S_i = \sum_{j=1}^{m_i} [\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}]^2, \qquad i = 0, 1, 2, ..., n-1 \tag{4.19}$$

where the parameterization over u's is in accordance with the chord length parameterization.

For the best fitting of the curve to the given data, we have to find out the parameters v_i's and w_i's so that sums S_i's are minimal and this can be done by using least square approximation. The minimum of S_i's occur if partial derivative of S_i's with respect to v_i's and w_i's becomes zero. We have

$$\frac{\partial S_i}{\partial v_i} = 2\sum_{j=1}^{m_i} \frac{\partial \mathbf{P}_i(u_{i,j})}{\partial v_i}.[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, \qquad i = 0, 1, 2, ..., n-1,$$

and

$$\frac{\partial S_i}{\partial w_i} = 2\sum_{j=1}^{m_i} \frac{\partial \mathbf{P}_i(u_{i,j})}{\partial w_i}.[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, \qquad i = 0, 1, 2, ..., n-1,$$

which implies

$$\frac{\partial S_i}{\partial v_i} = \frac{2h_i}{3}\sum_{j=1}^{m_i}(R_{1,i}(u_{i,j})\mathbf{D}_i).[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, i = 0, 1, 2, ..., n-1, \qquad (4.20)$$

and

$$\frac{\partial S_i}{\partial w_i} = -\frac{2h_i}{3}\sum_{j=1}^{m_i}(R_{2,i}(u_{i,j})\mathbf{D}_{i+1}).[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, i = 0, 1, 2, ..., n-1, \qquad (4.21)$$

Replacing $\mathbf{P}_i(u_{i,j})$ from equation (4.1) and simplifying equation (4.20) and (4.21), we get

$$\mathbf{D}_i.\mathbf{F}_i(a_{0,i} + a_{1,i}) + \mathbf{D}_i.\mathbf{F}_{i+1}(a_{2,i} + a_{3,i}) - \mathbf{D}_i.\sum_{j=1}^{m_i}(R_{1,i}(u_{i,j})\mathbf{P}_{i,j} =$$
$$\frac{h_i}{3}(-v_i|\mathbf{D}_i|^2 a_{1,i} + w_i\mathbf{D}_i.\mathbf{D}_{i+1}a_{2,i}), \qquad i = 0, 1, 2, ...n - 1. \tag{4.22}$$

and

$$\mathbf{D}_{i+1}.\mathbf{F}_i(a_{2,i} + a_{4,i}) + \mathbf{D}_{i+1}.\mathbf{F}_{i+1}(a_{5,i} + a_{6,i}) - \mathbf{D}_{i+1}.\sum_{j=1}^{m_i}(R_{2,i}(u_{i,j})\mathbf{P}_{i,j} =$$
$$\frac{h_i}{3}(-v_i\mathbf{D}_i.\mathbf{D}_{i+1}a_{2,i} + w_i|\mathbf{D}_{i+1}|^2 a_{5,i}), \qquad i = 0, 1, 2, ...n - 1. \tag{4.23}$$

where $a_{0,i}, ..., a_{6,i}$ are same as in equation (4.15) described in case 2.

Solving equation (4.22) and equation (4.23), we get

$$v_i = \frac{3}{h_i}\left[\frac{A_{3,i}B_{2,i} - A_{2,i}B_{3,i}}{A_{1,i}B_{2,i} - A_{2,i}B_{1,i}}\right] \tag{4.24}$$

and

$$w_i = \frac{3}{h_i}\left[\frac{A_{3,i}B_{1,i} - A_{1,i}B_{3,i}}{A_{2,i}B_{1,i} - A_{1,i}B_{2,i}}\right] \tag{4.25}$$

where

$$\left.\begin{aligned} A_{1,i} &= -|\mathbf{D}_i|^2 a_{1,i}, \\ A_{2,i} &= \mathbf{D}_i.\mathbf{D}_{i+1} a_{2,i}, \\ A_{3,i} &= \mathbf{D}_i.\mathbf{F}_i(a_{0,i} + a_{1,i}) + \mathbf{D}_i.\mathbf{F}_{i+1}(a_{2,i} + a_{3,i}) - \mathbf{D}_i.\sum_{j=1}^{m_i} R_{1,i}(u_{i,j})\mathbf{P}_{i,j}, \end{aligned}\right\} \quad (4.26)$$

and

$$\left.\begin{aligned} B_{1,i} &= -\mathbf{D}_i.\mathbf{D}_{i+1} a_{2,i}, \\ B_{2,i} &= |\mathbf{D}_{i+1}|^2 a_{5,i}, \\ B_{3,i} &= \mathbf{D}_{i+1}.\mathbf{F}_i(a_{2,i} + a_{4,i}) + \mathbf{D}_{i+1}.\mathbf{F}_{i+1}(a_{5,i} + a_{6,i}) - \mathbf{D}_{i+1}.\sum_{j=1}^{m_i} R_{2,i}(u_{i,j})\mathbf{P}_{i,j}, \end{aligned}\right\} \quad (4.27)$$

Thus the curve fitted using the above values of v_i's and w_i's will be a good candidate of best fit.

Case 4: $v_i = w_i = 1$ and tangents are calculated using Least Square method. In this case the shape parameters v_i's and w_i's are treated as constant for each segment and the tangents are approximated using Least Square method. Suppose, for $i = 0, 1, 2, ..., n-1$, the data segments

$$\mathbf{P}_{i,j} = (x_{i,j}, y_{i,j}), \qquad j = 1, 2, ..., m_i \quad (4.28)$$

are given as the ordered sets of the universal set of the data points. Then the squared sums S_i's of distances between $\mathbf{P}_{i,j}$'s and their corresponding parametric points $\mathbf{P}(t_j)$'s on the curve are computed as:

$$S_i = \sum_{j=1}^{m_i} [\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}]^2, \qquad i = 0, 1, 2, ..., n-1 \tag{4.29}$$

where the parameterization over u's is in accordance with the chord length parameterization.

We have to find out the tangents $\mathbf{D}_i$'s and $\mathbf{D}_{i+1}$'s so that sums S_i's are minimal and this can be done by using least square approximation. The minimum of S_i's occur if partial derivative of S_i's with respect to $\mathbf{D}_i$'s and $\mathbf{D}_{i+1}$'s becomes zero. We have

$$\frac{\partial S_i}{\partial \mathbf{D}_i} = 2\sum_{j=1}^{m_i} \frac{\partial \mathbf{P}_i(u_{i,j})}{\partial \mathbf{D}_i}.[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, \qquad i = 0, 1, 2, ..., n-1,$$

and

$$\frac{\partial S_i}{\partial \mathbf{D}_{i+1}} = 2\sum_{j=1}^{m_i} \frac{\partial \mathbf{P}_i(u_{i,j})}{\partial \mathbf{D}_{i+1}}.[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, \qquad i = 0, 1, 2, ..., n-1,$$

Since

$$\frac{\partial \mathbf{P}_i(u_{i,j})}{\partial \mathbf{D}_i} = \frac{h_i}{3}(R_{1,i}(u_{i,j}),$$

and

$$\frac{\partial \mathbf{P}_i(u_{i,j})}{\partial \mathbf{D}_{i+1}} = -\frac{h_i}{3}(R_{2,i}(u_{i,j}),$$

therefore

$$\frac{\partial S_i}{\partial \mathbf{D}_i} = \frac{2h_i}{3}\sum_{j=1}^{m_i} R_{1,i}(u_{i,j}).[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, i = 0, 1, 2, ..., n-1, \tag{4.30}$$

and

$$\frac{\partial S_i}{\partial \mathbf{D}_{i+1}} = -\frac{2h_i}{3}\sum_{j=1}^{m_i} R_{2,i}(u_{i,j}).[\mathbf{P}_i(u_{i,j}) - \mathbf{P}_{i,j}] = 0, i = 0, 1, 2, ..., n-1, \tag{4.31}$$

Replacing $\mathbf{P}_i(u_{i,j})$ from equation (4.1) and simplifying equation (4.30) and (4.31), we get

$$\begin{aligned}\mathbf{F}_i(a_{0,i} + a_{1,i}) + \mathbf{F}_{i+1}(a_{2,i} + a_{3,i}) - \sum_{j=1}^{m_i}(R_{1,i}(u_{i,j})\mathbf{P}_{i,j} = \\ \frac{h_i}{3}(-\mathbf{D}_i a_{1,i} + \mathbf{D}_{i+1}a_{2,i}), \qquad i = 0, 1, 2, ...n-1.\end{aligned} \tag{4.32}$$

and

$$\begin{aligned}&\mathbf{F}_i(a_{2,i} + a_{4,i}) + \mathbf{F}_{i+1}(a_{5,i} + a_{6,i}) - \sum_{j=1}^{m_i}(R_{2,i}(u_{i,j})\mathbf{P}_{i,j} = \\ &\frac{h_i}{3}(-\mathbf{D}_i a_{2,i} + \mathbf{D}_{i+1} a_{5,i}), \qquad i = 0, 1, 2, ...n - 1.\end{aligned} \tag{4.33}$$

where $a_{0,i}, ..., a_{6,i}$ are same as in equation (4.15) described in case 2.

Solving equation (4.32) and equation (4.33), we get

$$\mathbf{D}_i = \frac{3}{h_i(a_2^2 - a_1 a_5)}[a_5 \mathbf{A}_{1,i} - a_2 \mathbf{A}_{2,i}] \tag{4.34}$$

and

$$\mathbf{D}_{i+1} = \frac{3}{h_i(a_2^2 - a_1 a_5)}[a_2 \mathbf{A}_{1,i} - a_1 \mathbf{A}_{2,i}] \tag{4.35}$$

where

$$\left.\begin{aligned} A_{1,i} &= \mathbf{F}_i(a_{0,i} + a_{1,i}) + \mathbf{F}_{i+1}(a_{2,i} + a_{3,i}) - \sum_{j=1}^{m_i} R_{1,i}(u_{i,j})\mathbf{P}_{i,j}, \\ A_{2,i} &= \mathbf{F}_i(a_{2,i} + a_{4,i}) + \mathbf{F}_{i+1}(a_{5,i} + a_{6,i}) - \sum_{j=1}^{m_i} R_{2,i}(u_{i,j})\mathbf{P}_{i,j} \end{aligned}\right\} \tag{4.36}$$

Thus the curve fitted using the above values of $\mathbf{D}_i$'s and $\mathbf{D}_{i+1}$'s will be a good candidate of best fit.

Case 5: v_i's, w_i's and $\mathbf{D}_i$'s are calculated using Least Square method.

In this case the shape parameters v_i's and w_i's and the tangents are approximated using Least Square method. Here we combined the results of case 3 and 4. So we use equations (4.24) and (4.25) to calculate the best values of shape parameters and equations (4.34) and (4.35) to approximate the tangents at the characteristic points. The results at the end shows that this case gives good results in terms of less number of break points as compared to other cases.

4.5.4 Breaking Segments

A fitted Bezeir curve to a segment may not satisfy the threshold tolerance limit. The curve is then to be subdivided in two at the point of worst error - the point where the fitted spline is farthest from the digitized curve. In the literature [9, 10] it is computed as the squared distance between each point p_i of digitized curve and its corresponding point $p(t_i)$ of parametric curve. A different approach is taken in this thesis and each segment of the curve is broken at the maximum difference of x or y coordinates. The new break point will be considered as a significant point and the curve is again fitted between these characteristic points. The distance 'd' (in terms of pixels) between original curve points $\mathbf{P}_{i,j}$ and their corresponding points

$\mathbf{P}(t_{i,j})$ on the parametric curve is given by

$$d = \max(|\mathbf{P}_{x_{i,j}} - \mathbf{P}_x(t_{i,j})|, |\mathbf{P}_{y_{i,j}} - \mathbf{P}_y(t_{i,j})|)$$

If d exceeds predefined error tolerance limit then the segment is broken into two segments at the point of maximum distance and the point corresponding to maximum distance is added to list of significant points. Number of segments and number of Significant points are increased by one.

The process is repeated for each segment until all the segments of all the pieces meet the threshold tolerance limit.

4.6 Conclusion

In this chapter the details of the system has been described. The first few basic steps of the system were mentioned briefly. The curve fitting step is the main step of the system and discussed in detail. The generalized Hermite cubic is used to fit the curve. The two corner points for each segment are taken as the two end control points of the curve whereas the least square method is used to get the optimal shape parameter values.

Chapter 5

Web Application

The recent explosion in the popularity of the world wide web and its associated hypertext markup Language (HTML) and hypertext transfer protocol (HTTP) presents an exciting new opportunity to provide widely distributed access to sophisticated software applications [96].

A system or application appearance over the World Wide Web is nowadays common and sometimes essential and economical. Our aim is to implement and test the curve fitting approaches that are discussed in chapter 4 providing end user with an easy-to-use interface. Another reason for making the system Web based is to share the knowledge amongst the whole Internet community.

Initially two options were available for deploying the system over the Web. Either to use Matlab Web Server with some httpd server or to use Microsoft IIS or Tomcat Web Server in combination with ASP, JSP and JavaBeans. All the cases mentioned in chapter 4 were implemented on Matlab, so the Matlab Web Server option along with Apache Web Server is used. Since the Matlab Web Server does not support the uploading of files, the combination of JSP, JavaBeans and Tomcat Web Server to implement the upload feature is used. Here the Matlab Web Server Environment, it's components and how to build the applications in it are briefly discussed. The following information is extracted from [97].

5.1 MATLAB Web Server Environment

The *MATLAB*® Web Server enables to create MATLAB applications that use the capabilities of the World Wide Web to send data to MATLAB for computation and to display the results in a Web browser. The MATLAB Web Server depends upon TCP/IP networking for transmission of data between the client system and MATLAB. The required networking software and hardware must be installed on the system prior to using the MATLAB Web Server. In the simplest configuration, a Web browser runs on client workstation, while MATLAB, the MATLAB Web Server (matlabserver), and the Web server daemon (httpd) run on another machine.

In a more complex network, the Web server daemon can run on a machine apart from the others.

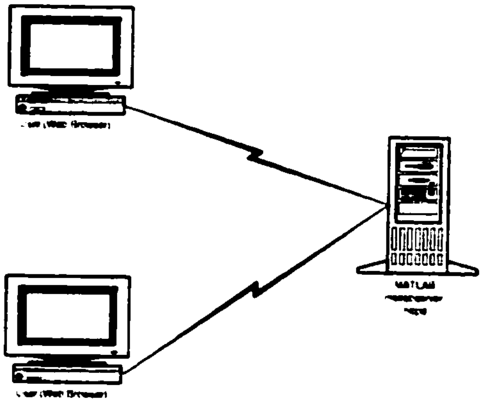


Figure 5.1: Simple Configuration

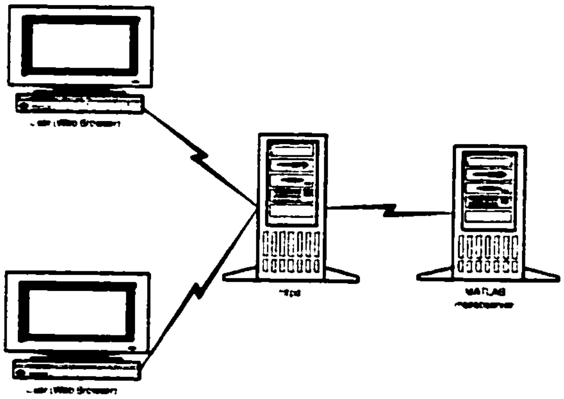


Figure 5.2: Complex Configuration

5.2 Building MATLAB Web Server Applications

MATLAB Web Server applications are a combination of M-files (Matlab Files), Hypertext Markup Language (HTML), and graphics.

The process of creating a MATLAB Web Server application involves the creation of:

1. An HTML input document for data submission to MATLAB.
2. An HTML output document for display of MATLAB's computations.
3. A MATLAB M-file to process input data and compute results. The task of this file is to:
   - receives the data entered in the HTML input form.
   - analyzes the data and generates any requested graphics.
   - places the output data into a MATLAB structure.
   - calls Matlab function 'htmlrep' to place the output data into an HTML output document template.
4. List the application name and associated configuration data in the configuration file matweb.conf.

5.3 MATLAB Web Server Components

The MATLAB Web Server consists of a set of programs that enable MATLAB programmers to create MATLAB applications and access them on the Web:

- matlabserver: Manages the communication between the Web application and MATLAB. matlabserver is a multithreaded TCP/IP server. It runs the MATLAB program (M-file) specified in a hidden field named mlmfile contained in the HTML document. matlabserver invokes matweb.m, which in turn runs the M-file. matlabserver can be configured to listen on any legal TCP/IP port by editing the matlabserver.conf file on Windows NT or running webconf on Solaris/ Linux.

- matweb: A TCP/IP client of matlabserver. This program uses the Common Gateway Interface (CGI) to extract data from HTML documents and transfer it to matlabserver.

- matweb.m: Calls the M-file that we want the Web application to run.

- matweb.conf: A configuration file that matweb needs for connecting to matlabserver. Applications must be listed in matweb.conf.

The following figure shows how Matlab operates over the Web.

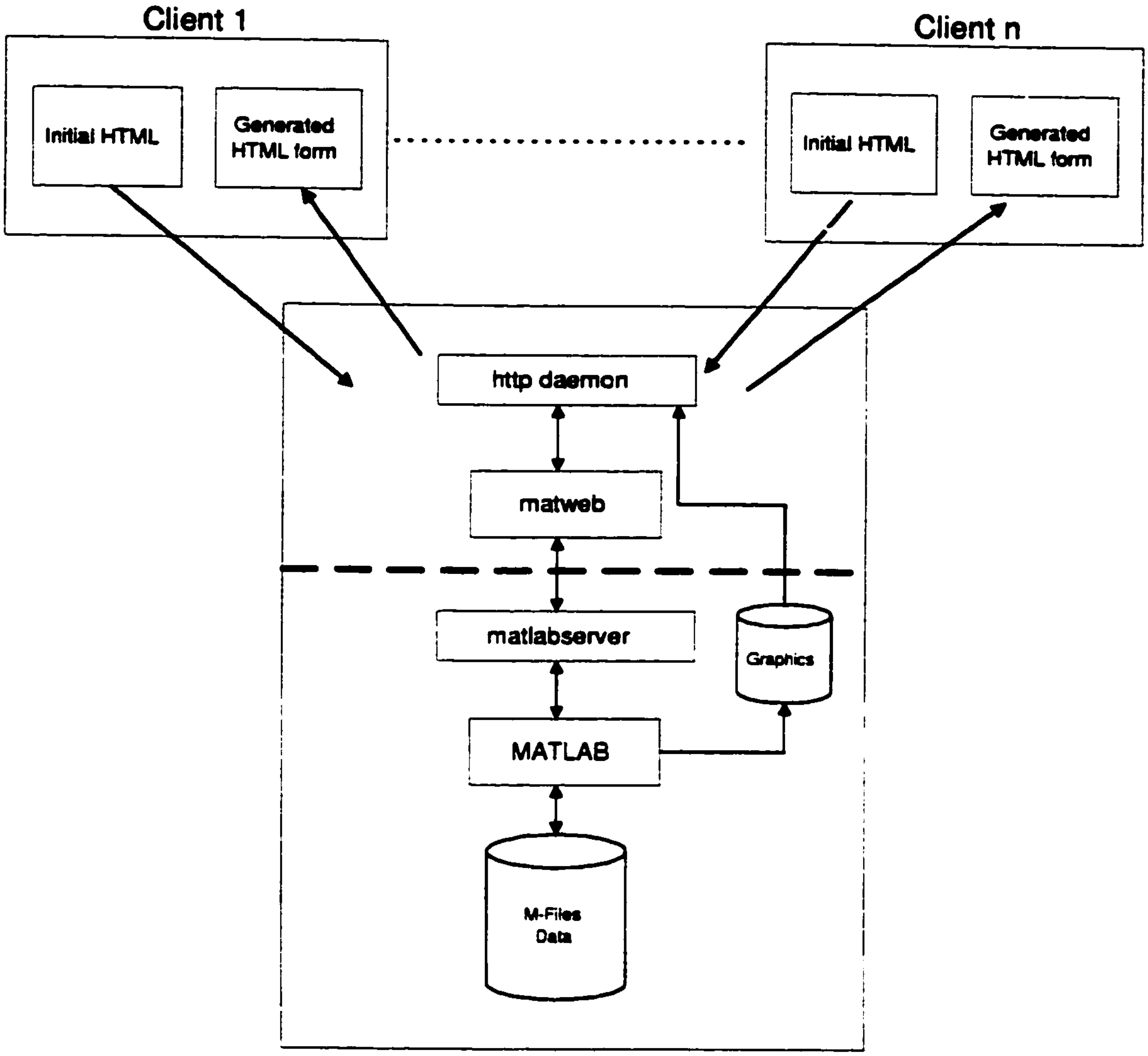


Figure 5.3: Matlab on the Web

5.4 Structure and Implementation

The structure of our system is similar to the one depicted in Figure 5.1. The Apache Web Server is used as the httpd daemon and the web system works as shown in Figure 5.3. Five cases that were discussed in chapter 4 were implemented and made available for testing over the Internet. Several different gray scale bitmap images, set of arabic and english alphabets are provided for testing. In addition to this a user can upload his own bitmap image for testing with any of the five cases. Figures 5.5 and 5.6 shows the set of arabic and english characters. Figure 5.7 shows the miscellaneous bitmap images page. Also a user uploaded images page have the same structure and appearance. A user has to select an image by clicking on the radio button next to the images and then select two more things from the list of values. The first one is the threshold value in pixels which specify how much error can be tolerated as described in chapter 4. The default value of threshold is 3 pixels. Another one is the selection of any one case from the five available cases discussed in chapter 4. By default, case 5 will be selected and the user has to press the 'Go' button to start the whole process described in chapter 4. A flow chart showing the client and server flow is depicted in figure (5.4).

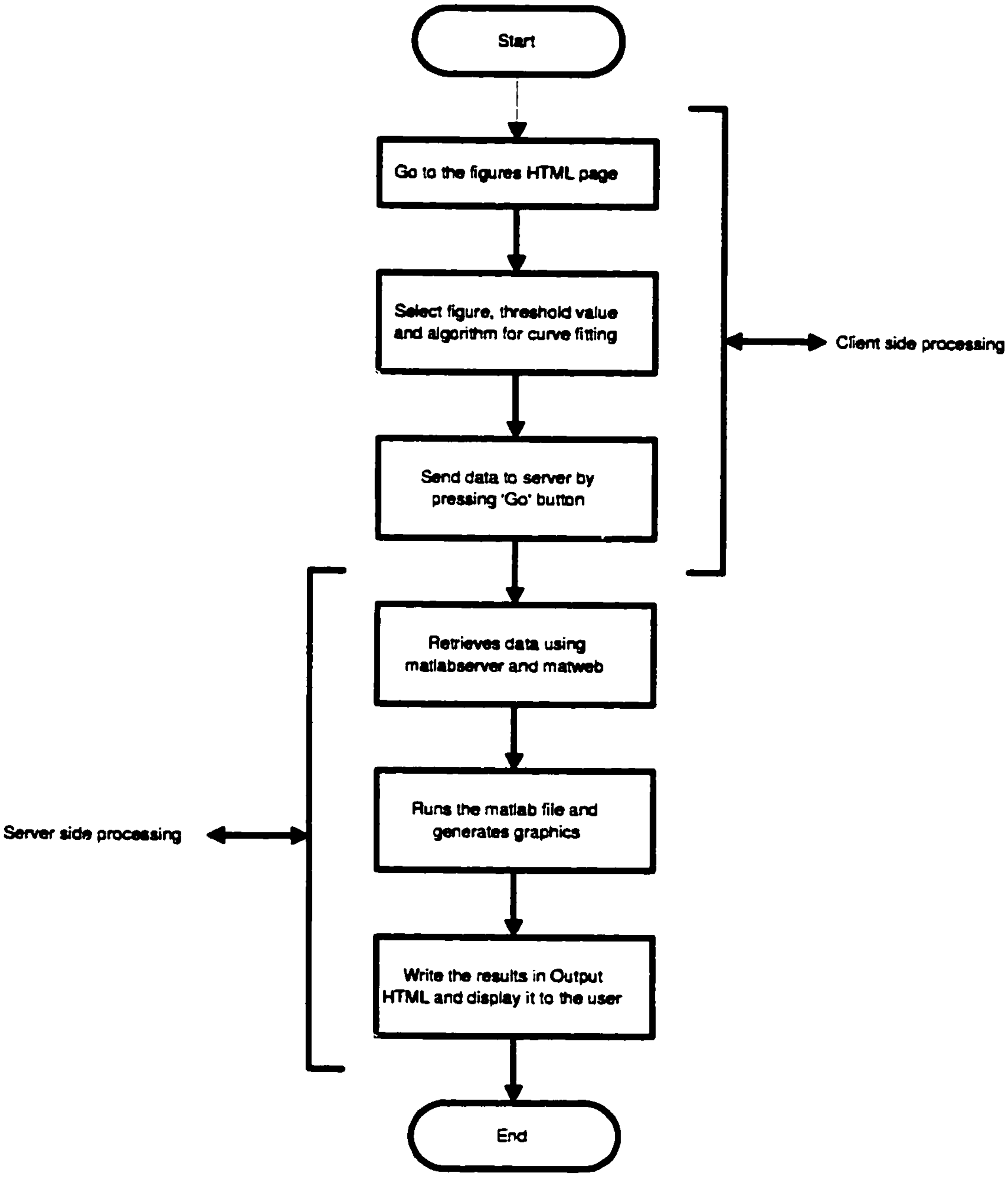

Start
Go to the figures HTML page
Select figure, threshold value and algorithm for curve fitting
Send data to server by pressing 'Go' button
Retrieves data using matlabserver and matweb
Runs the matlab file and generates graphics
Write the results in Output HTML and display it to the user
End
Client side processing
Server side processing

Figure 5.4: Flow chart of Matlab Web Application

Figure 5.8 shows the final result screen that appears after the processing. It shows some statistics such as total number of points, total number of corner and break points. It also shows the digitized image, outline of the curve and the fitted curve with corner and break points.

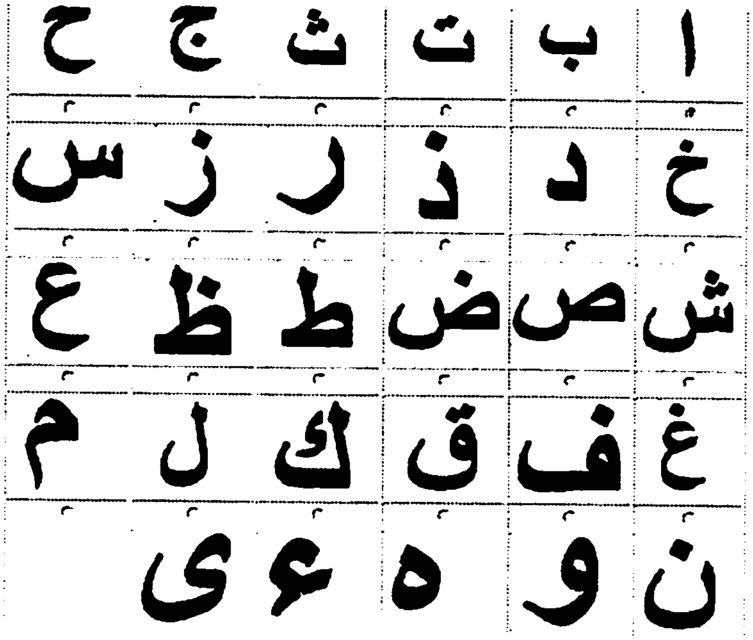


Figure 5.5: Set of Arabic characters

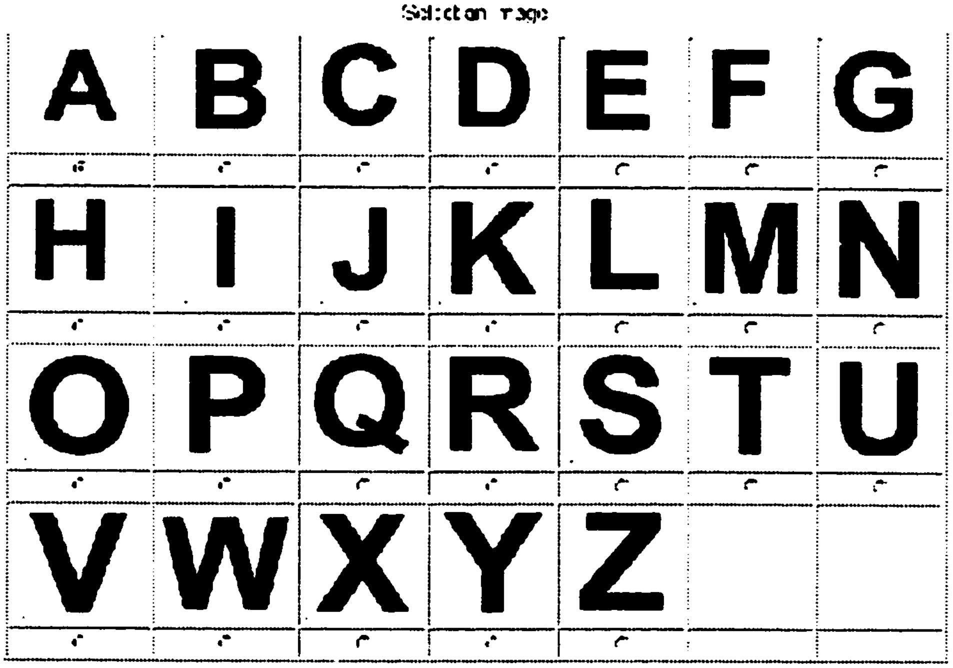
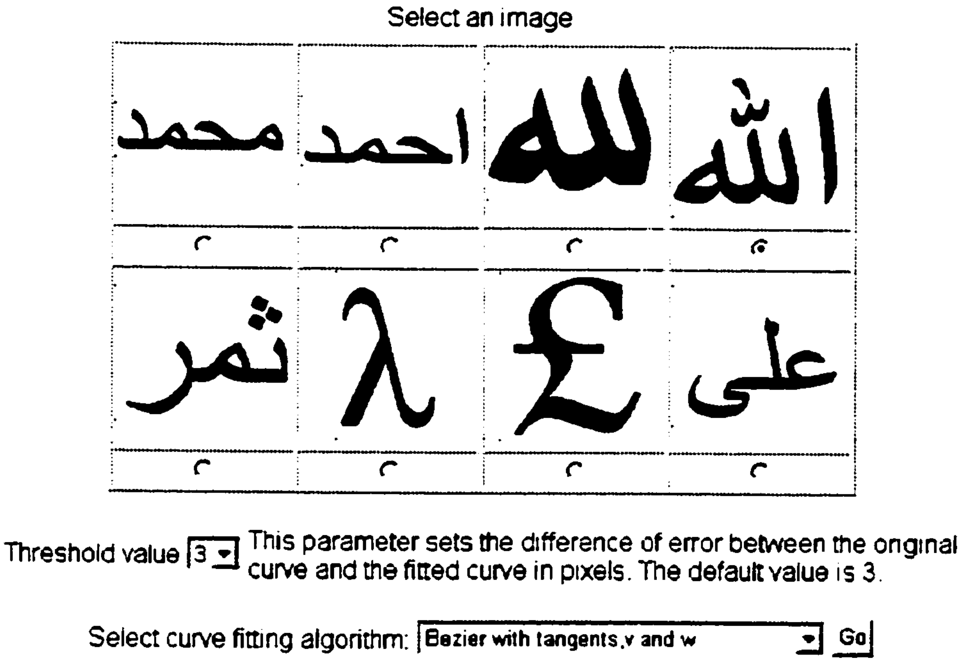


Figure 5.6: Set of English characters



Select an image

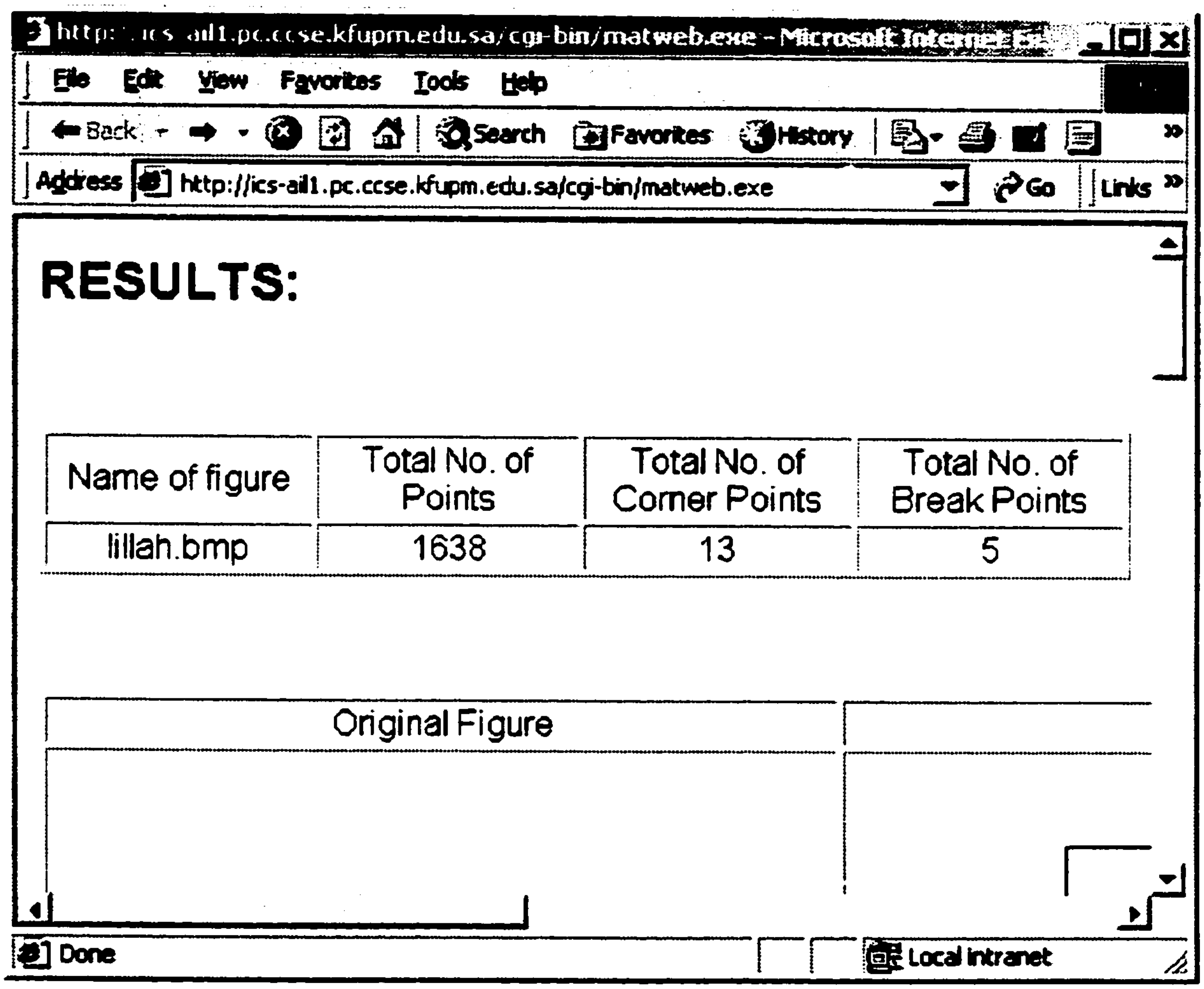
○ ○ ○ ●

○ ○ ○ ○

Threshold value 3 ▾ This parameter sets the difference of error between the original curve and the fitted curve in pixels. The default value is 3.

Select curve fitting algorithm: Bezier with tangents,v and w ▾ Go

Figure 5.7: Misc. figures page


http://ics-ail1.pc.ccse.kfupm.edu.sa/cgi-bin/matweb.exe - Microsoft Internet Ex
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://ics-ail1.pc.ccse.kfupm.edu.sa/cgi-bin/matweb.exe
Go Links
Done
Local intranet

RESULTS:

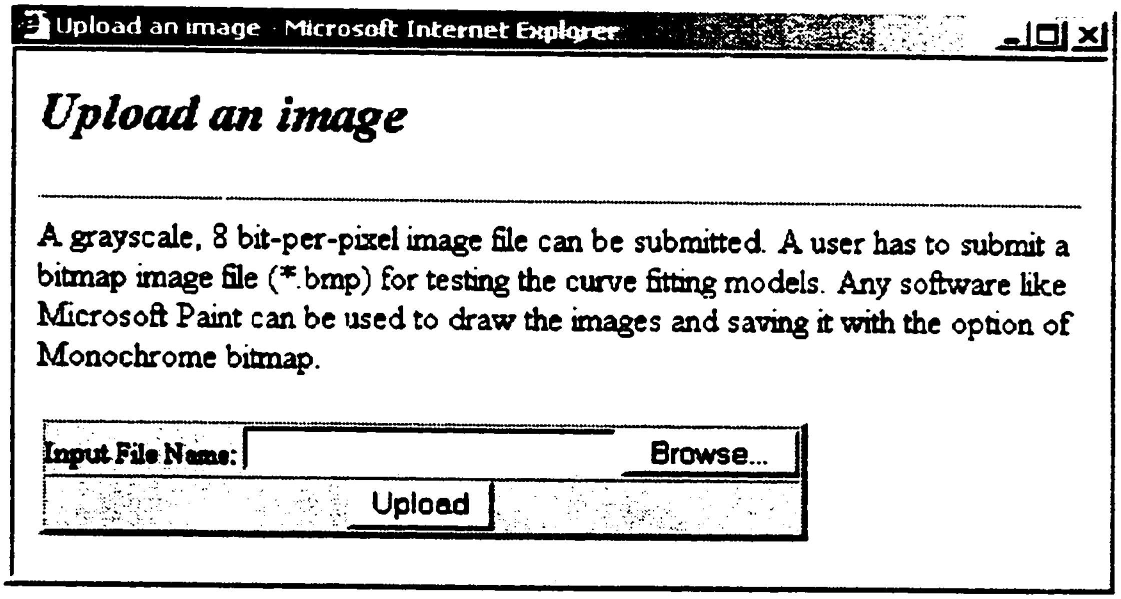
Name of figure	Total No. of Points	Total No. of Corner Points	Total No. of Break Points
lillah.bmp	1638	13	5

Original Figure	

Figure 5.8: Final result screen

5.4.1 Uploading Feature

In addition to providing some standard images and sets of alphabets, a user can upload his own bitmap image. Since Matlab Web Server does not support the uploading feature, the Tomcat Web Server is used as it supports JSP and JavaBeans technology. By using JSP and JavaBean technology, a user bitmap image file is uploaded and then one html file is also updated according to it, so that a user can see and test his image. Although the uploading feature uses different Web Server, this is completely transparent to the novice user. All he has to do is to select the file and then upload it by clicking a button. A sample screen is shown in Figure 5.9.

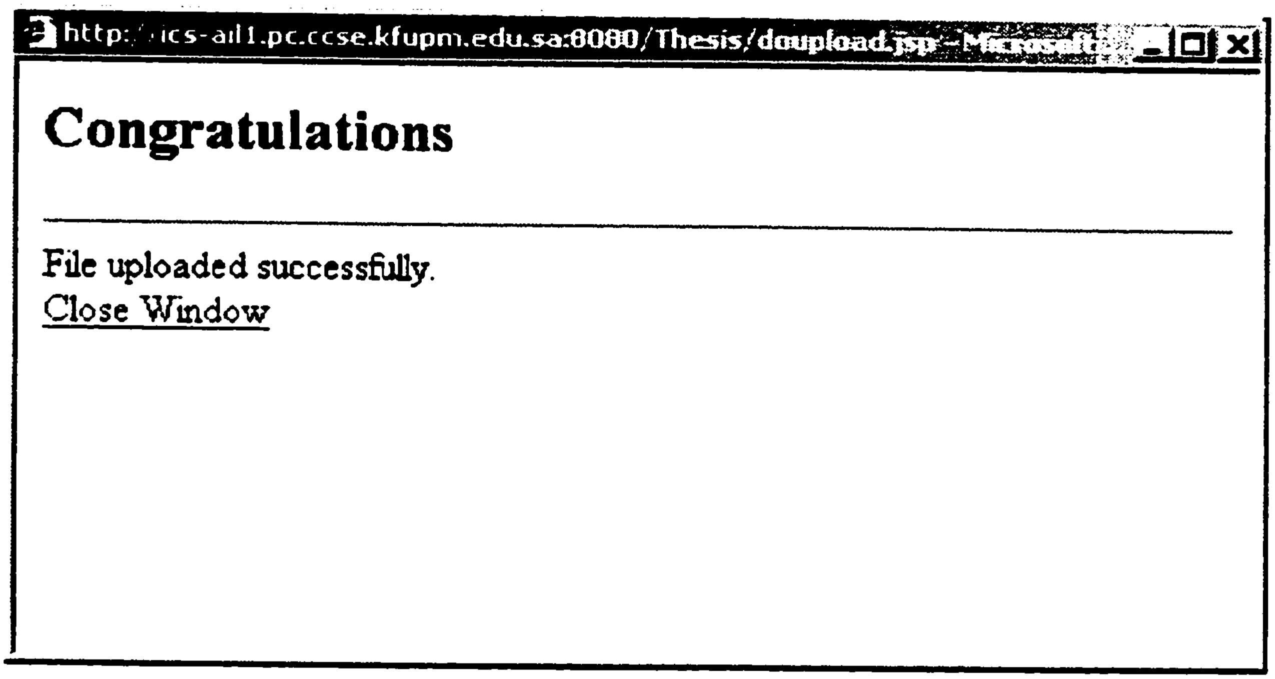


Upload an image

A grayscale, 8 bit-per-pixel image file can be submitted. A user has to submit a bitmap image file (*.bmp) for testing the curve fitting models. Any software like Microsoft Paint can be used to draw the images and saving it with the option of Monochrome bitmap.

Figure 5.9: Screen shot of an upload page

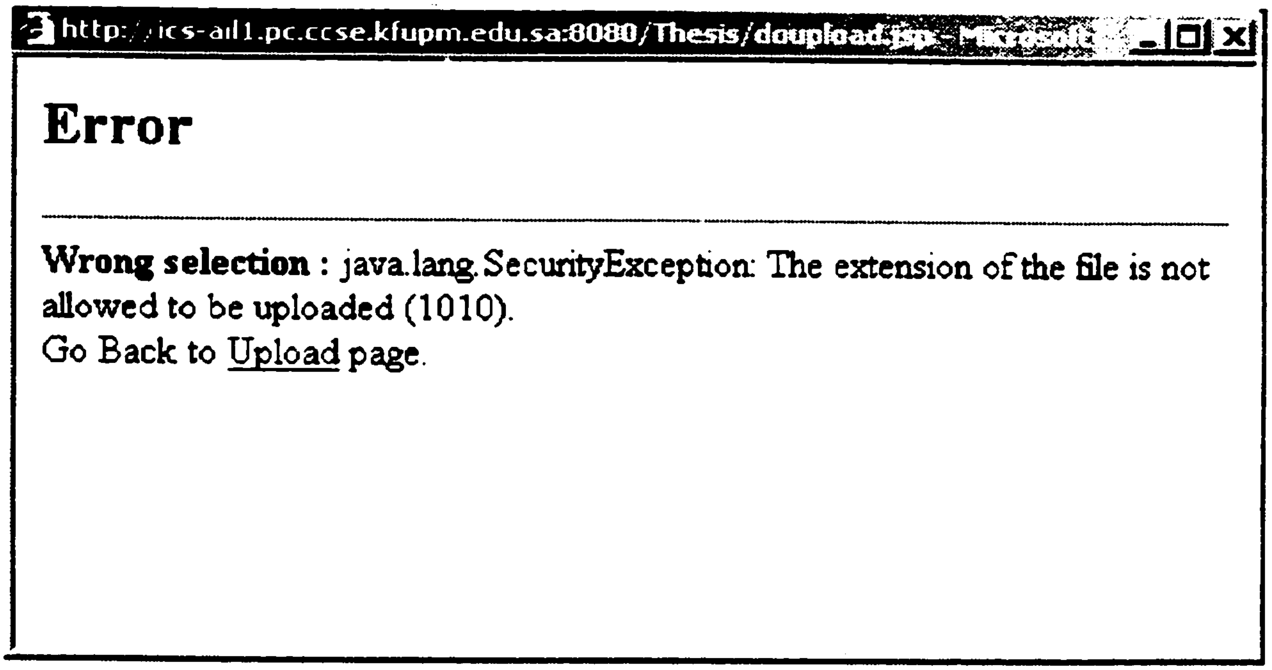
Currently only the monochrome bitmaps files are supported for testing. If the user tries to upload a file other than the bitmap then he will get an error message. Figure 5.11 shows the error screen that a user will see and he will be asked to repeat the process. If an image is successfully uploaded then a screen shown in figure 5.10 will be displayed.


http://ics-ail1.pc.ccse.kfupm.edu.sa:8080/Thesis/doupload.jsp - Microsoft

Congratulations

File uploaded successfully.
Close Window

Figure 5.10: Screen shot of an upload success page


http://ics-ai11.pc.ccse.kfupm.edu.sa:8080/Thesis/doupload.jsp - Microsoft

Error

Wrong selection : java.lang.SecurityException: The extension of the file is not allowed to be uploaded (1010).
Go Back to Upload page.

Figure 5.11: Screen shot of an upload failure page

Chapter 6

Comparison and Results

In this chapter the results of the five cases that were discussed in chapter 4 are shown. Also the comparison is done for the five cases with varying threshold values. In the end a comparison with a previous work is also done. Four images are being used for the purpose. One is an arabic word 'Lillah' (Figure 6.1) and the other is the 'Kanji' (Figure 6.2) character from Japanese language. The other two are the images of an aeroplane (Figure 6.3) and a flower (Figure 6.4). They are selected to emphasize that not only the approach is useful for font designing, it is appropriate for any other image (currently only monochrome bitmap images are used). The results are shown in the order from worst to excellent case in terms of the number of break points required to fit the curve. Also the images obtained with Murtaza's method [3] are also shown.

Figures 6.5-6.18 show the final outlines for 'Lillah' (Figure 6.1) image. For all the cases shown, two values, 1 and 3 are used for the threshold. The results of case 1 are depicted in Figures 6.5 and 6.6. Figures 6.7 and 6.8 show the results for case 2 whereas Figures 6.9 and 6.10 displays the results for case 3. It can be seen that we are getting the best fit with minimum number of significant points as we move along from case 1 to case 3. The final outlines obtained using case 4 are shown in Figures 6.11 and 6.12 whereas Figures 6.13 and 6.14 are the final results for case 5. Case 5 can be regarded as the best in terms of number of significant points required to fit the curve. Figures 6.15 and 6.16 are the final outlines obtained using Murtaza's method [3] when no filtering and reparameterization are used. Finally the results for the case with filtering but without reparameterization are shown in Figures 6.17 and 6.18.

Figures 6.19-6.32 show the final outlines for 'Kanji' (Figure 6.2) image. For all the cases shown, two values, 1 and 3 are used for the threshold. The results of case 1 are depicted in Figures 6.19 and 6.20. Figures 6.21 and 6.22 show the results for case 2 whereas Figures 6.23 and 6.24 displays the results for case 3. It can be seen that we are getting the best fit with minimum number of significant points as we move along from case 1 to case 3. The final outlines obtained using case 4 are shown in Figures 6.25 and 6.26 whereas Figures 6.27 and 6.28 are the final results for case 5. Case 5 can be regarded as the best in terms of number of significant points required to fit the curve. Figures 6.29 and 6.30 are the final outlines obtained using Murtaza's method [3] when no filtering and reparameterization are used. Finally the results for the case with filtering but without reparameterization are shown in Figures 6.31 and 6.32.

Figures 6.33-6.46 show the final outlines for aeroplane (Figure 6.3) image. For all the cases shown, two values, 1 and 3 are used for the threshold. The results of case 1 are depicted in Figures 6.33 and 6.34. Figures 6.35 and 6.36 show the results for case 2 whereas Figures 6.37 and 6.38 displays the results for case 3. It can be seen that we are getting the best fit with minimum number of significant points as we move along from case 1 to case 3. The final outlines obtained using case 4 are shown in Figures 6.39 and 6.40 whereas Figures 6.41 and 6.42 are the final results for case 5. Case 5 can be regarded as the best in terms of number of significant points required to fit the curve. Figures 6.43 and 6.44 are the final outlines obtained using Murtaza's method [3] when no filtering and reparameterization are used. Finally the results for the case with filtering but without reparameterization are shown in Figures 6.45 and 6.46.

Figures 6.47-6.60 show the final outlines for flower (Figure 6.4) image. For all the cases shown, two values, 1 and 3 are used for the threshold. The results of case 1 are depicted in Figures 6.47 and 6.48. Figures 6.49 and 6.50 show the results for case 2 whereas Figures 6.51 and 6.52 displays the results for case 3. It can be seen that we are getting the best fit with minimum number of significant points as we move along from case 1 to case 3. The final outlines obtained using case 4 are shown in Figures 6.53 and 6.54 whereas Figures 6.55 and 6.56 are the final results for case 5. Case

5 can be regarded as the best in terms of number of significant points required to fit the curve. Figures 6.57 and 6.58 are the final outlines obtained using Murtaza's method [3] when no filtering and reparameterization are used. Finally the results for the case with filtering but without reparameterization are shown in Figures 6.59 and 6.60.

Some statistics in tabular form are shown from Tables 6.1 to 6.9. Table 6.1 shows the number of pieces and data points in each of the four figures. Tables 6.2 to 6.9 shows the comparison of different cases. Tables 6.11 to 6.14 shows the shape parameters and tangent values for all the five cases.

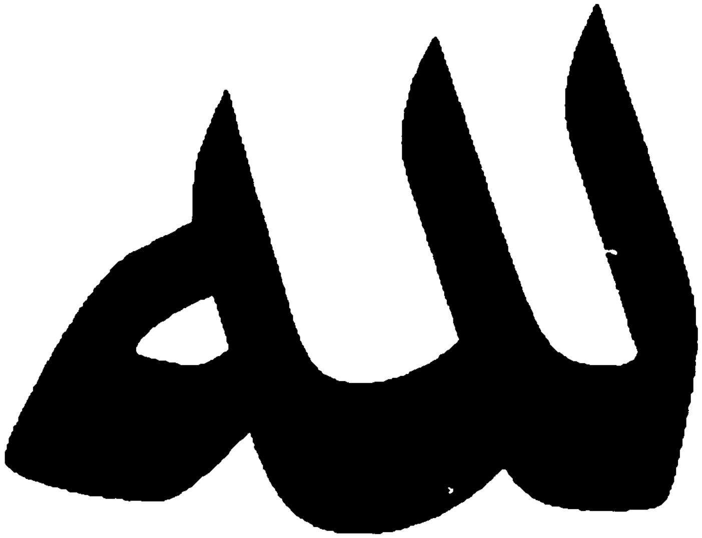


Figure 6.1: Digitized Image of 'Lillah' character

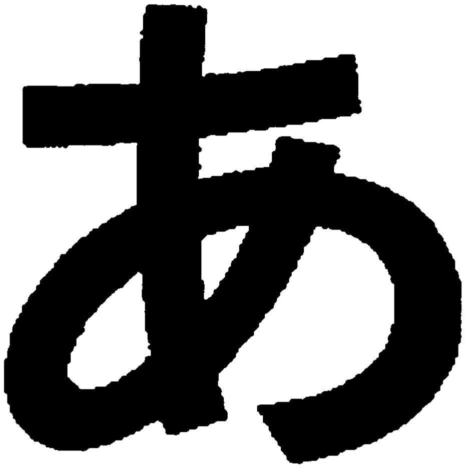


Figure 6.2: Digitized Image of 'Kanji' character

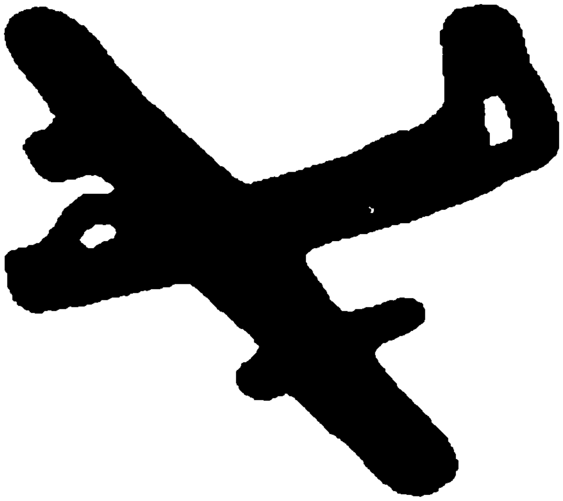


Figure 6.3: Digitized Image of an aeroplane

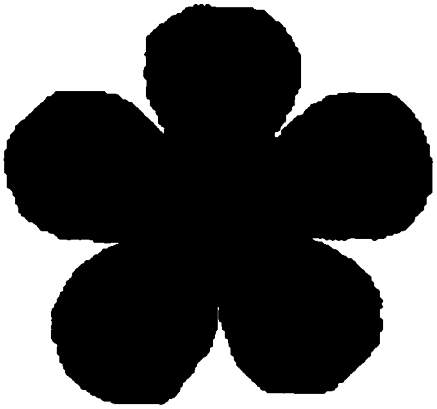


Figure 6.4: Digitized Image of flower

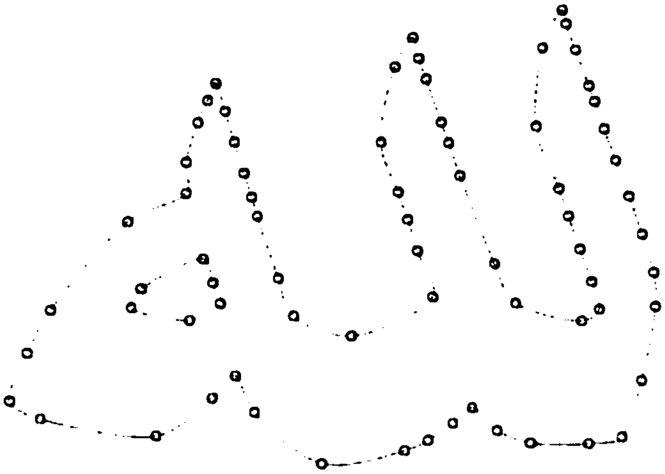


Figure 6.5: Fitted curve with corner and break points, threshold=3 (case 1).

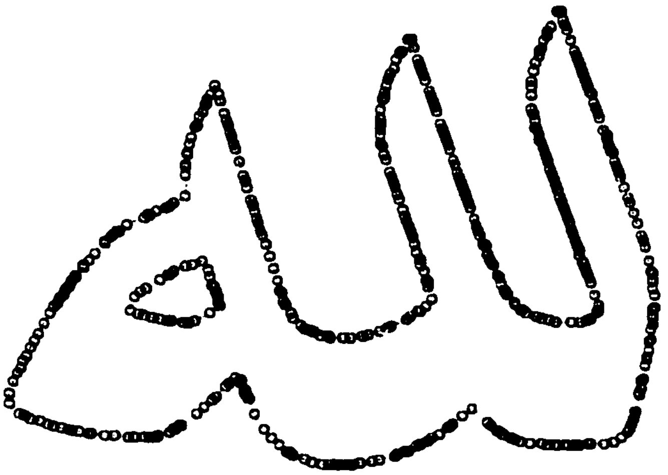


Figure 6.6: Fitted curve with corner and break points, threshold=1 (case 1).

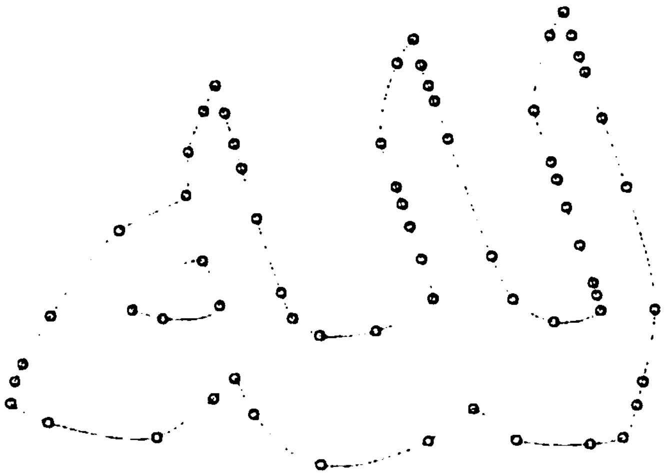


Figure 6.7: Fitted curve with corner and break points, threshold=3 (case 2).

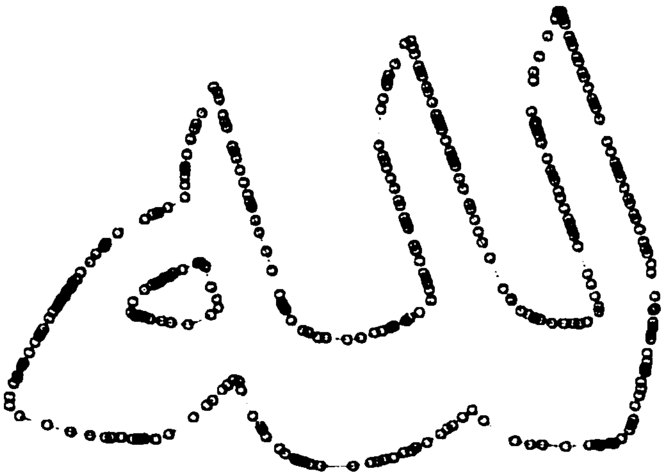


Figure 6.8: Fitted curve with corner and break points, threshold=1 (case 2).

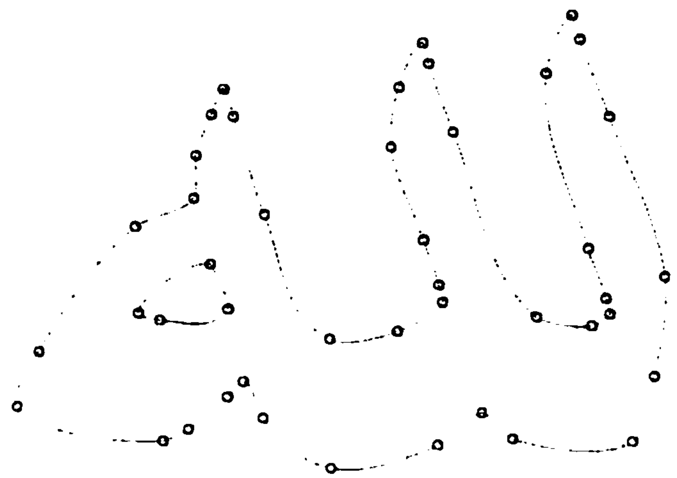


Figure 6.9: Fitted curve with corner and break points, threshold=3 (case 3).

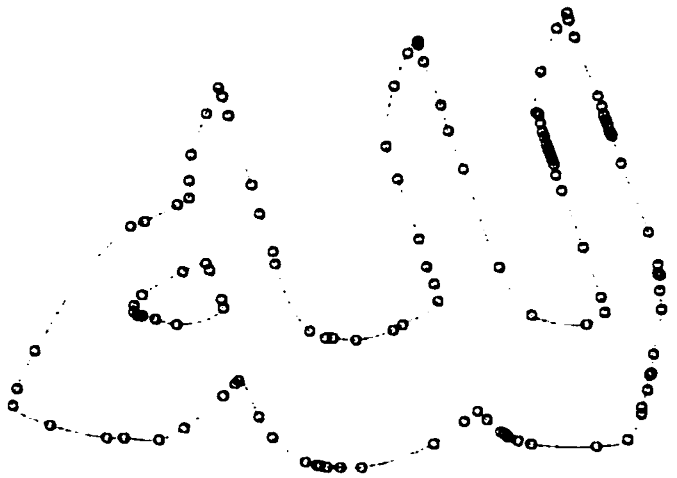


Figure 6.10: Fitted curve with corner and break points, threshold=1 (case 3).

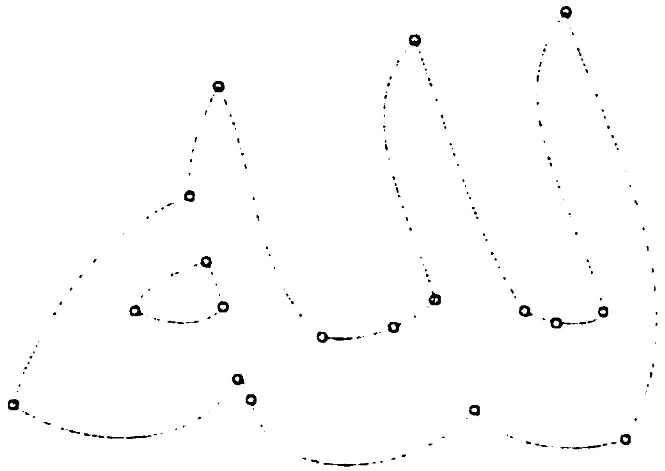


Figure 6.11: Fitted curve with corner and break points, threshold=3 (case 4).

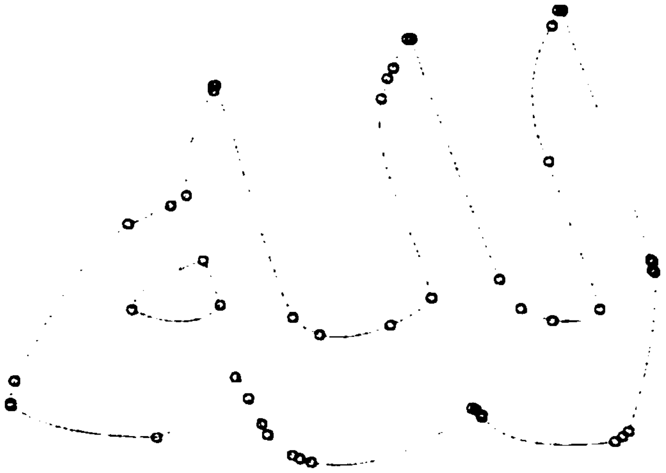


Figure 6.12: Fitted curve with corner and break points, threshold=1 (case 4).

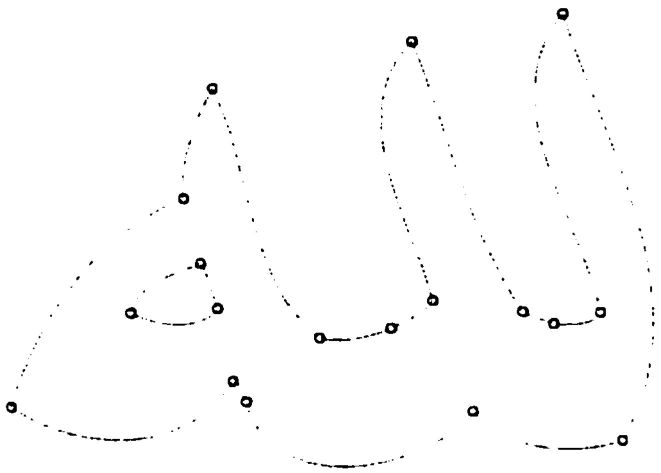


Figure 6.13: Fitted curve with corner and break points, threshold=3 (case 5).

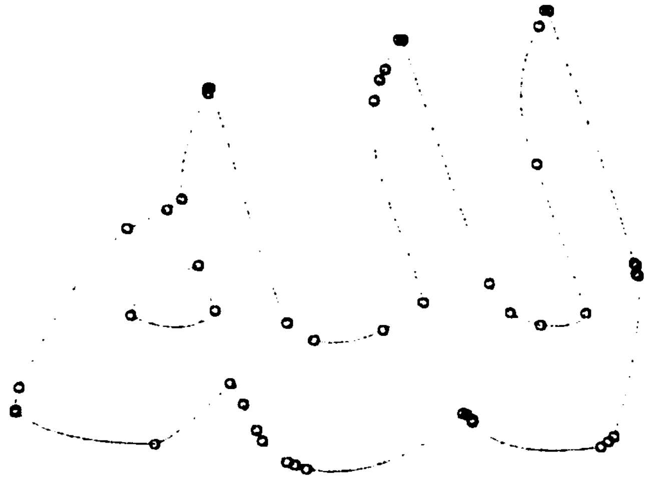


Figure 6.14: Fitted curve with corner and break points, threshold=1 (case 5).

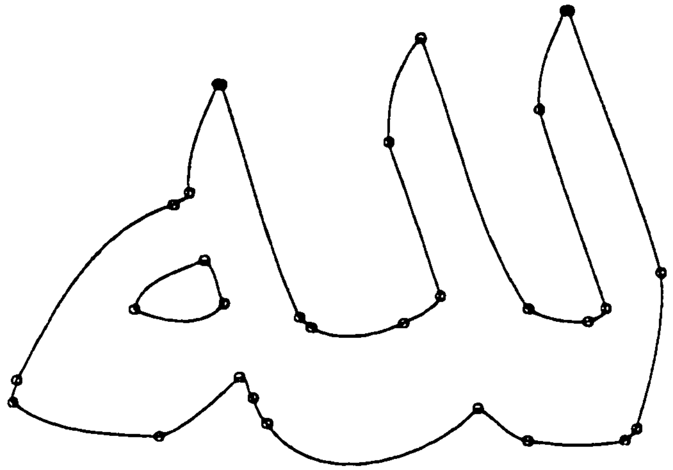


Figure 6.15: Final Outline without filtering and reparameterization, threshold=3

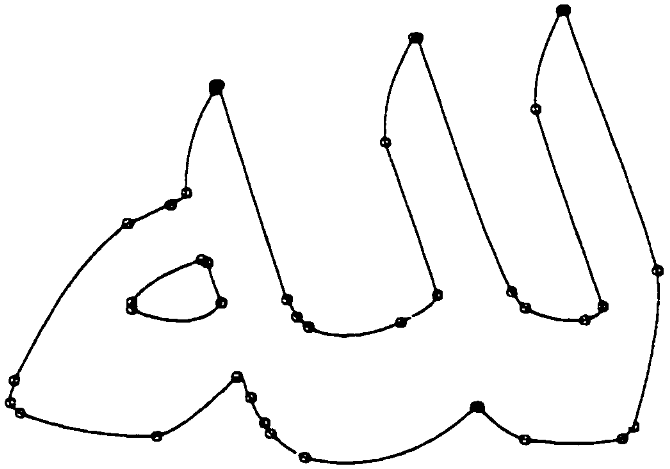


Figure 6.16: Final Outline without filtering and reparameterization, threshold=1

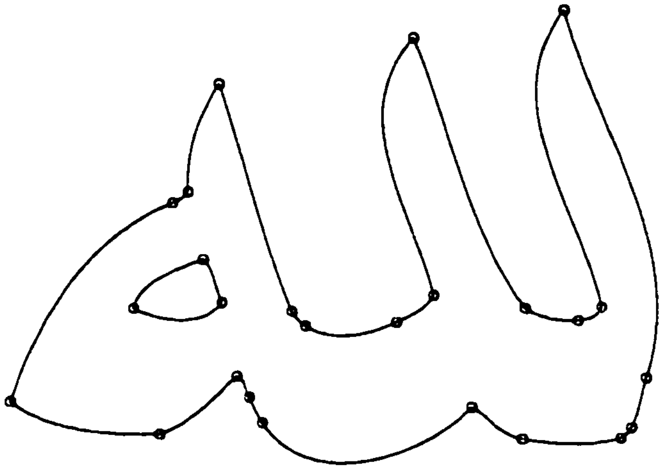


Figure 6.17: Final Outline without reparameterization, threshold=3

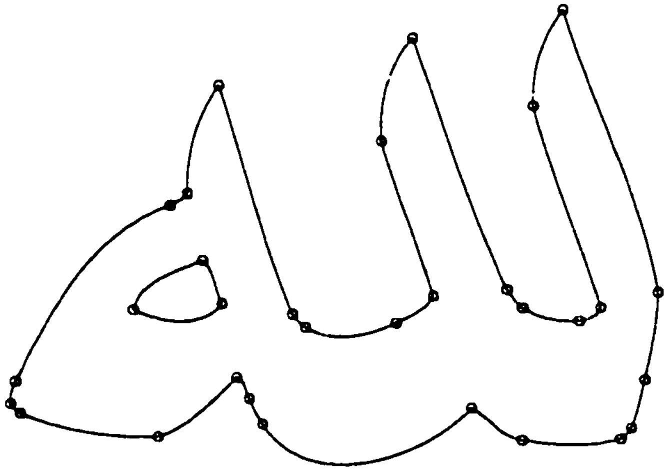


Figure 6.18: Final Outline without reparameterization, threshold=1

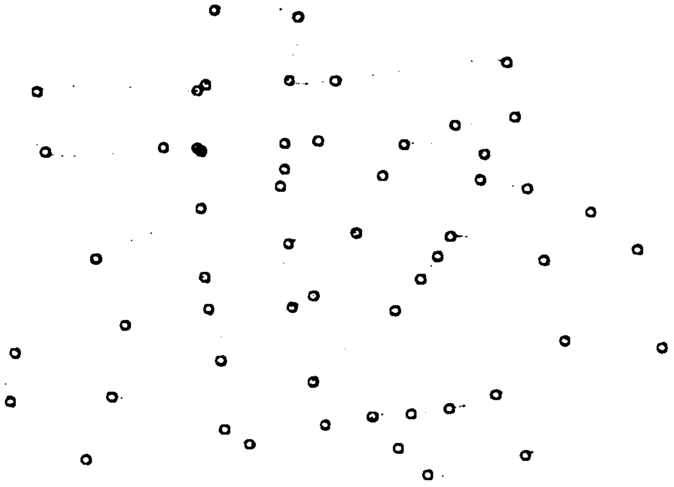


Figure 6.19: Fitted curve with corner and break points, threshold=3 (case 1).

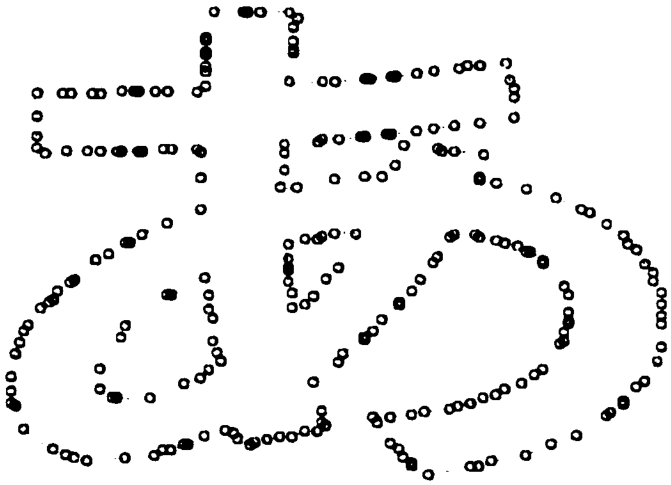


Figure 6.20: Fitted curve with corner and break points, threshold=1 (case 1).

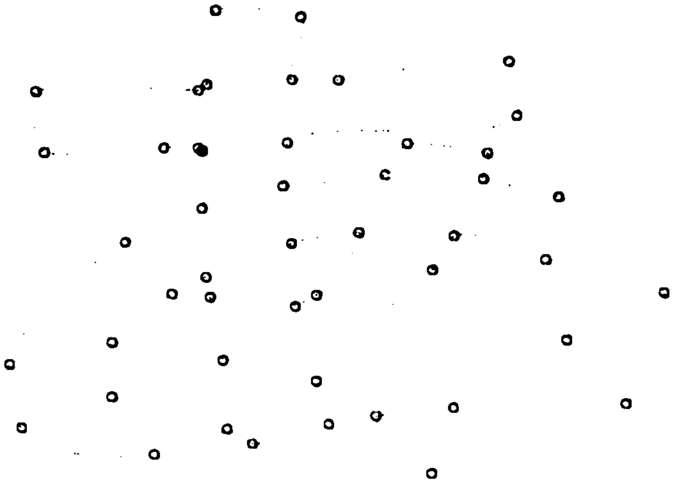


Figure 6.21: Fitted curve with corner and break points, threshold=3 (case 2).

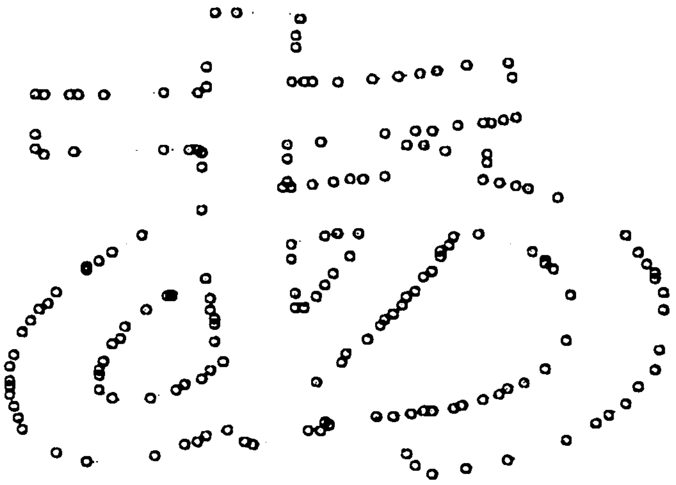


Figure 6.22: Fitted curve with corner and break points, threshold=1 (case 2).

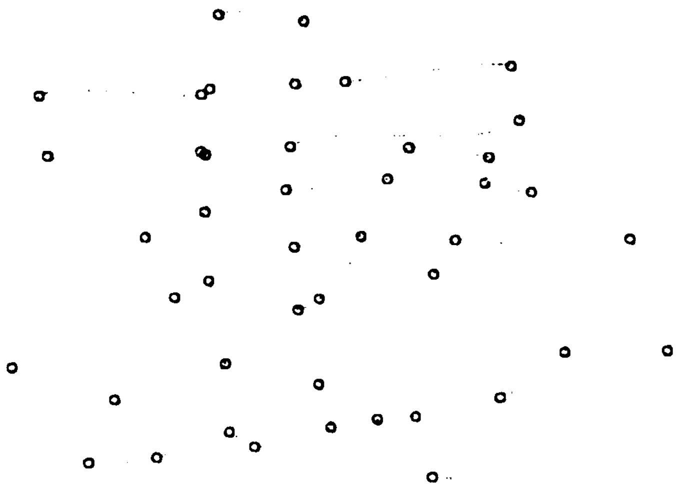


Figure 6.23: Fitted curve with corner and break points, threshold=3 (case 3).

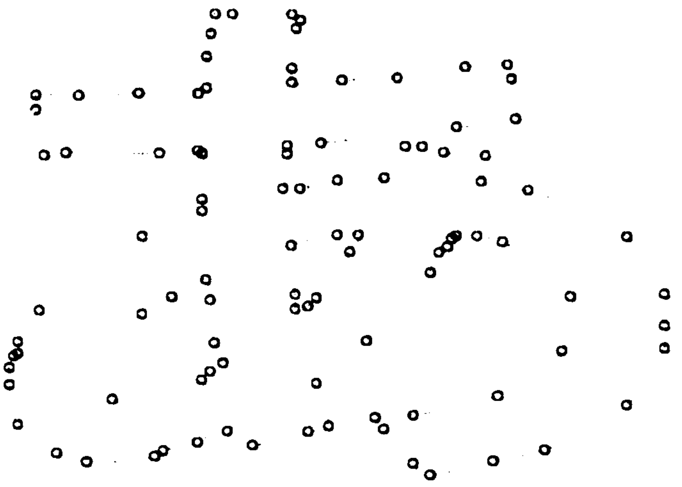


Figure 6.24: Fitted curve with corner and break points, threshold=1 (case 3).

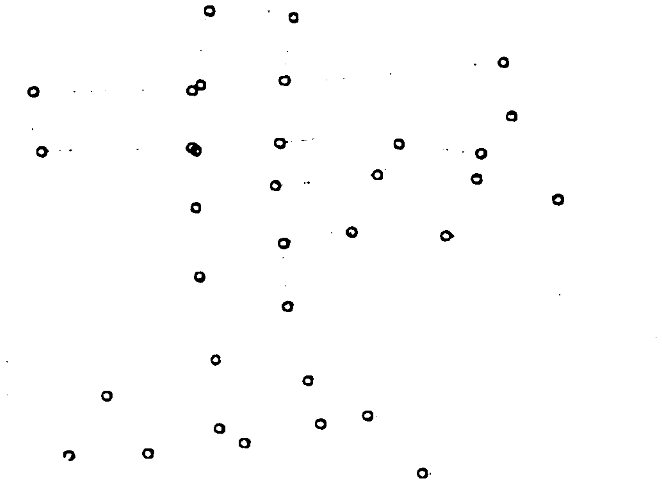


Figure 6.25: Fitted curve with corner and break points, threshold=3 (case 4).

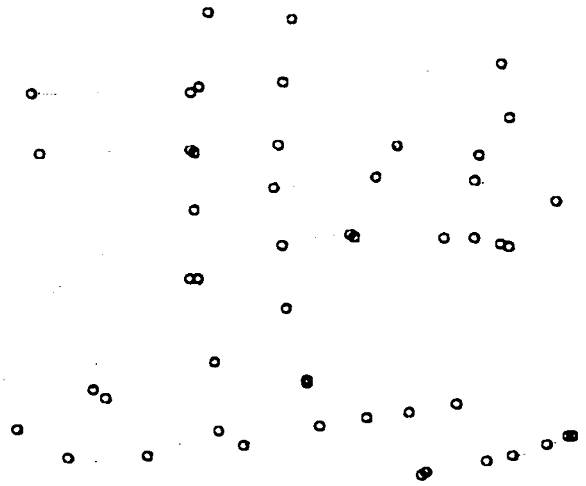


Figure 6.26: Fitted curve with corner and break points, threshold=1 (case 4).

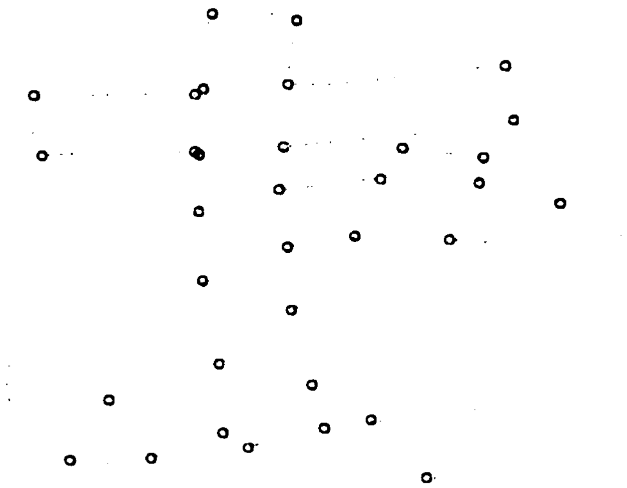


Figure 6.27: Fitted curve with corner and break points, threshold=3 (case 5).

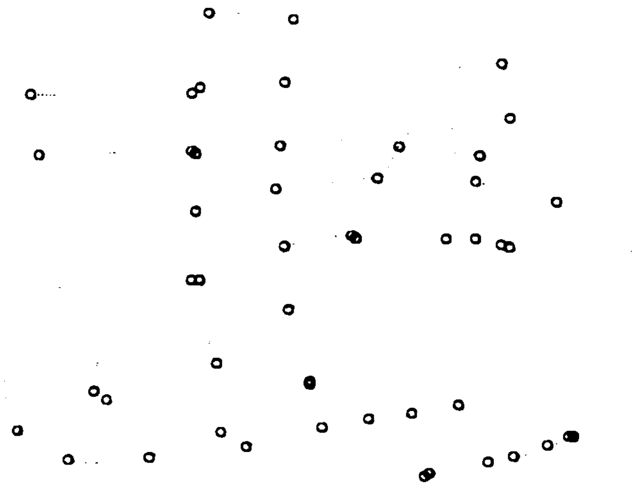


Figure 6.28: Fitted curve with corner and break points, threshold=1 (case 5).

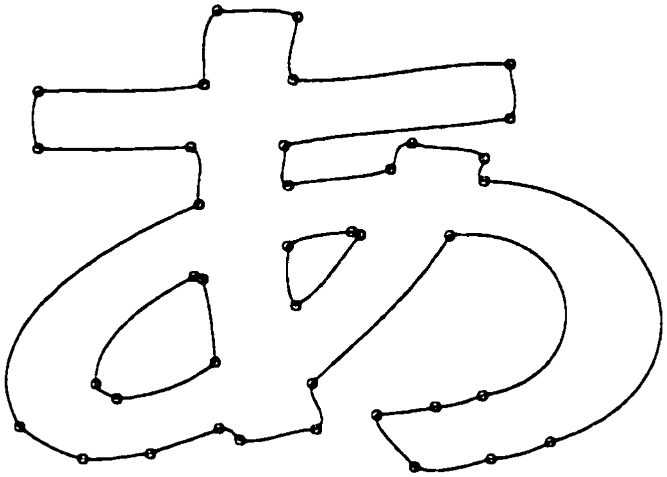


Figure 6.29: Final Outline without filtering and reparameterization, threshold=3

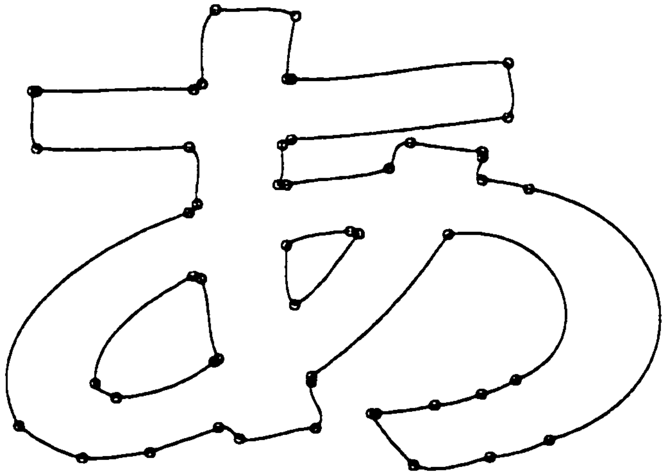


Figure 6.30: Final Outline without filtering and reparameterization, threshold=1

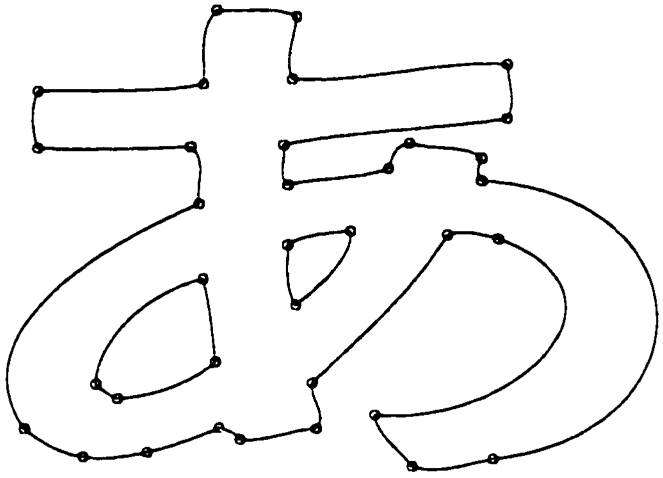


Figure 6.31: Final Outline without reparameterization, threshold=3

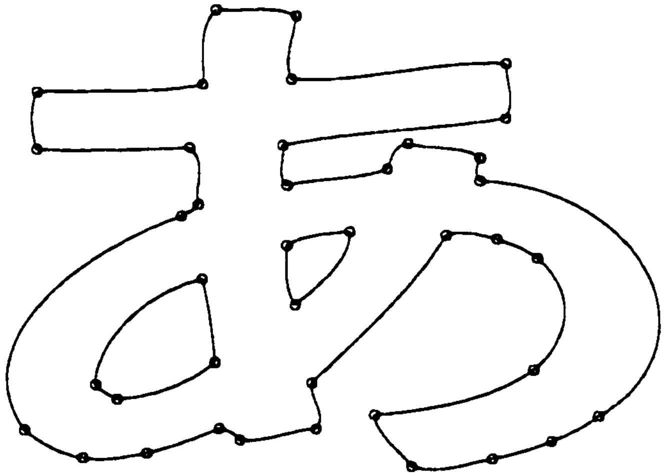


Figure 6.32: Final Outline without reparameterization, threshold=1

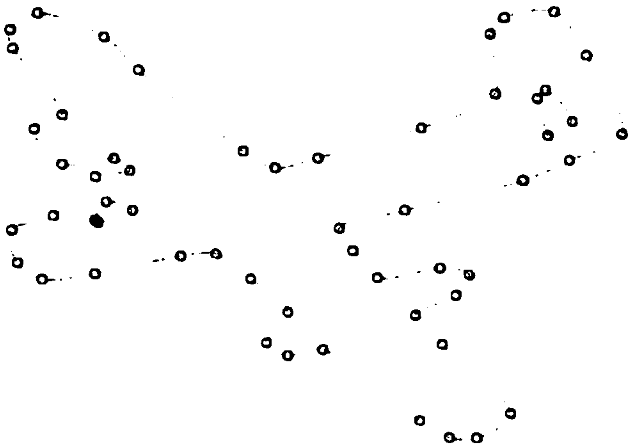


Figure 6.33: Fitted curve with corner and break points, threshold=3 (case 1).

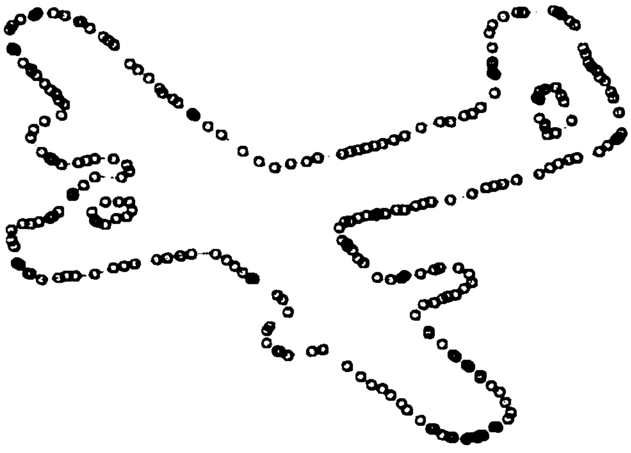


Figure 6.34: Fitted curve with corner and break points, threshold=1 (case 1).

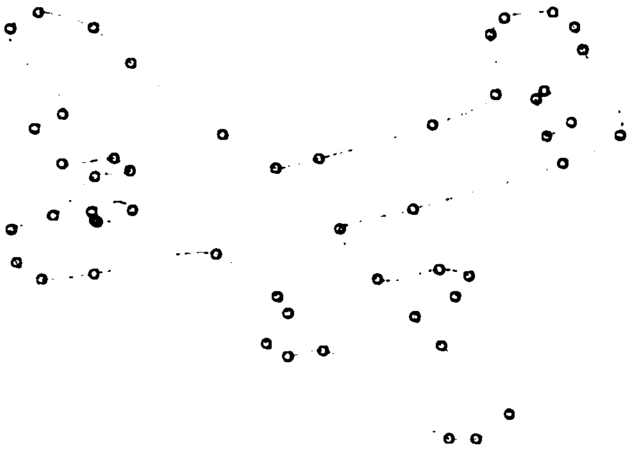


Figure 6.35: Fitted curve with corner and break points, threshold=3 (case 2).

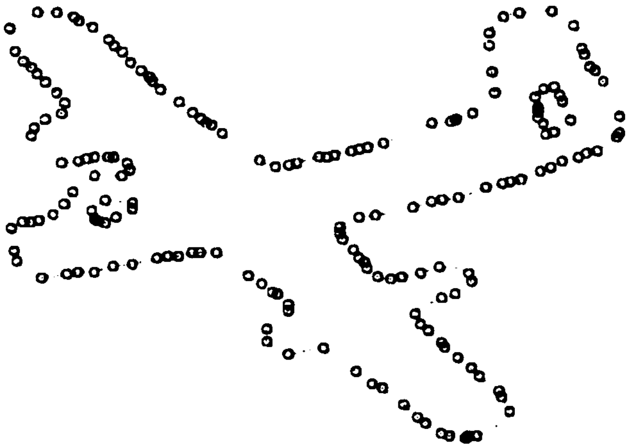


Figure 6.36: Fitted curve with corner and break points, threshold=1 (case 2).

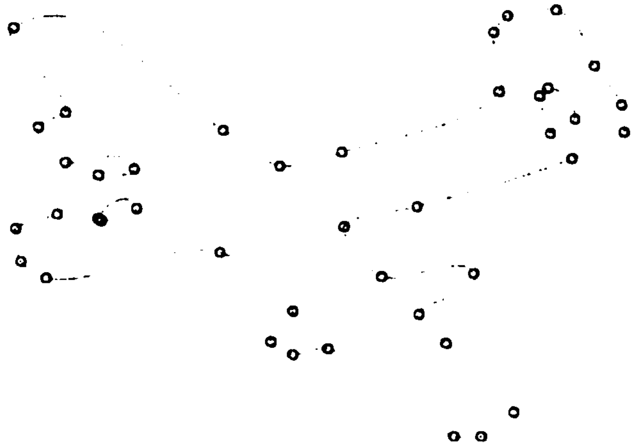


Figure 6.37: Fitted curve with corner and break points, threshold=3 (case 3).

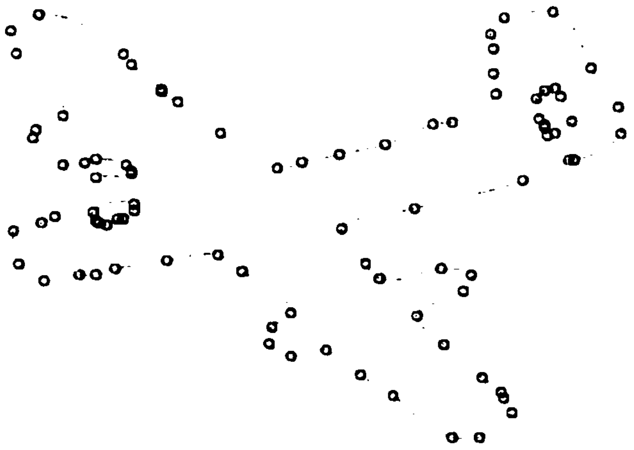


Figure 6.38: Fitted curve with corner and break points, threshold=1 (case 3).

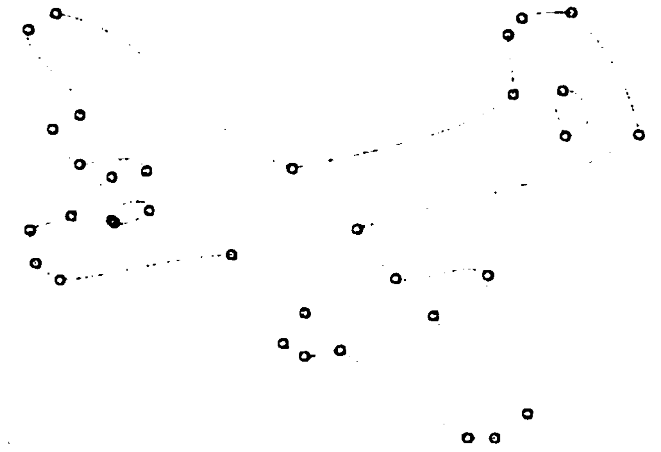


Figure 6.39: Fitted curve with corner and break points, threshold=3 (case 4).

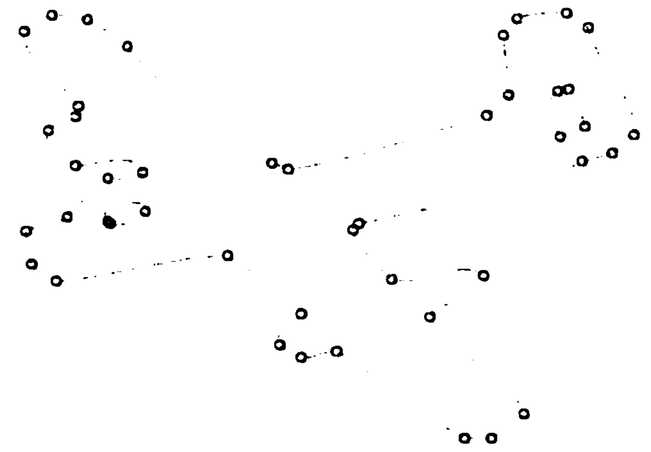


Figure 6.40: Fitted curve with corner and break points, threshold=1 (case 4).

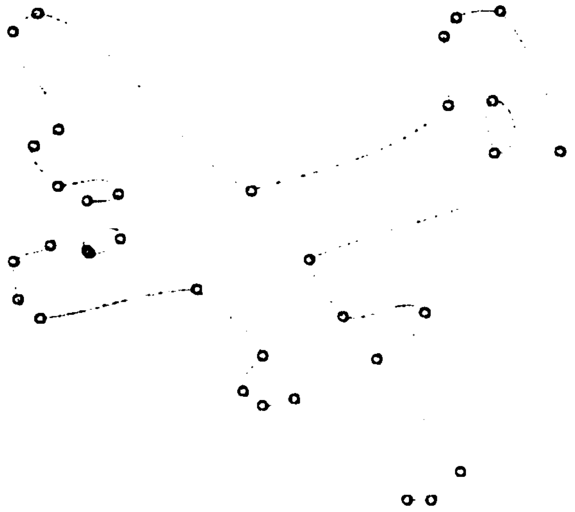


Figure 6.41: Fitted curve with corner and break points, threshold=3 (case 5).

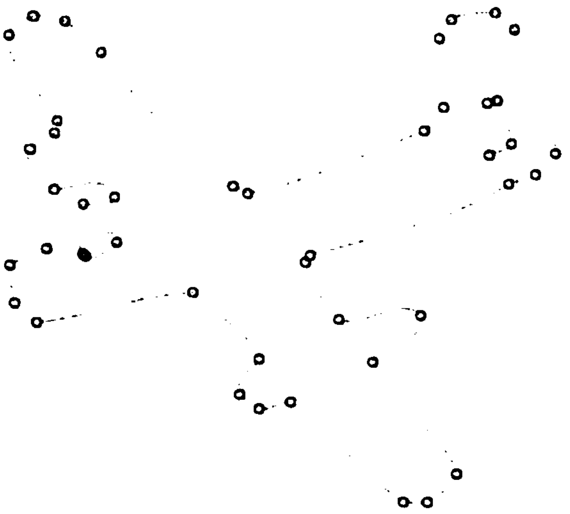


Figure 6.42: Fitted curve with corner and break points, threshold=1 (case 5).

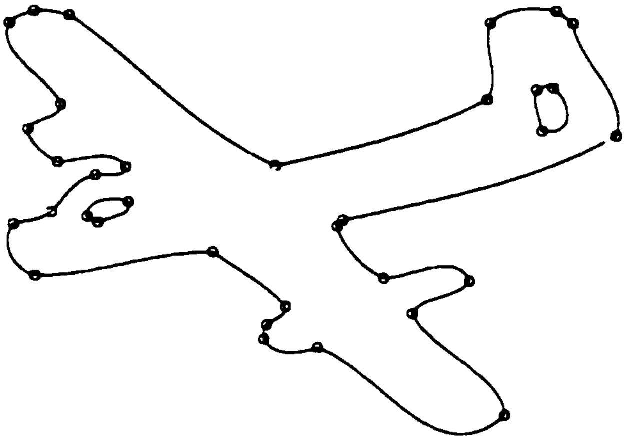


Figure 6.43: Final Outline without filtering and reparameterization, threshold=3

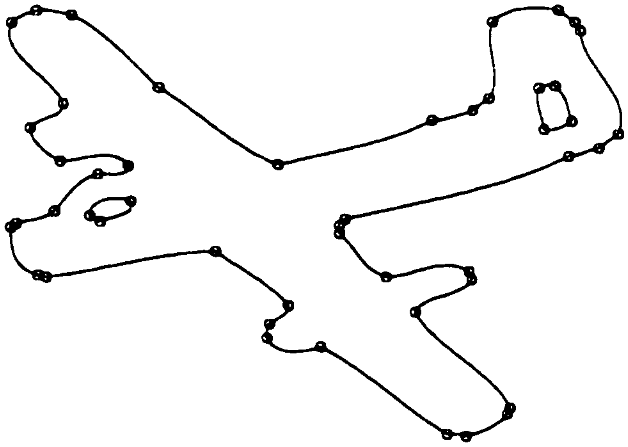


Figure 6.44: Final Outline without filtering and reparameterization, threshold=1

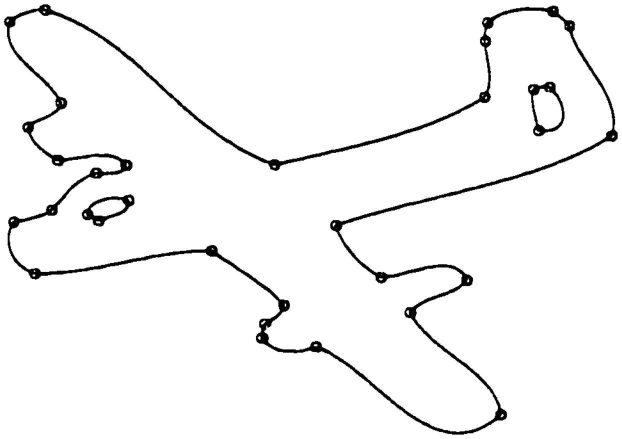


Figure 6.45: Final Outline without reparameterization, threshold=3

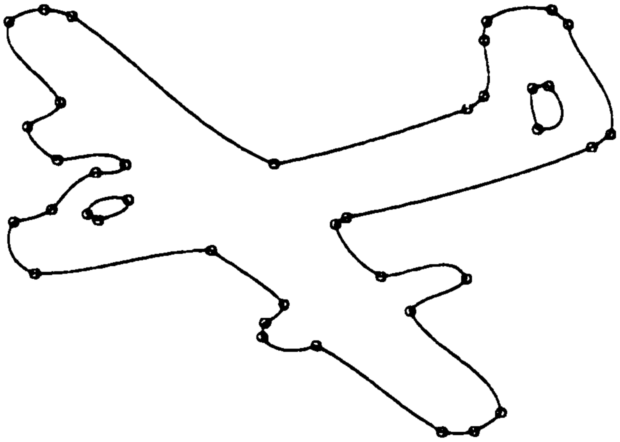


Figure 6.46: Final Outline without reparameterization, threshold=1

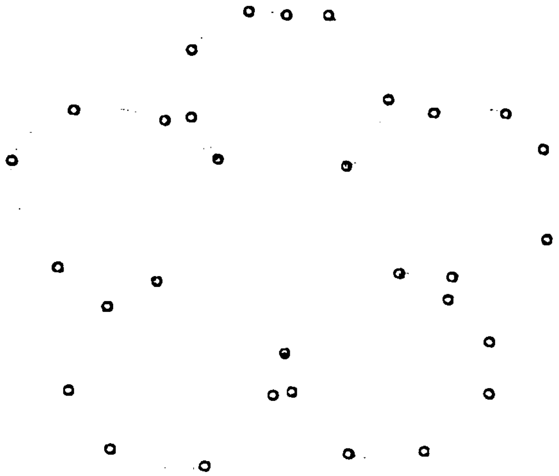


Figure 6.47: Fitted curve with corner and break points, threshold=3 (case 1).

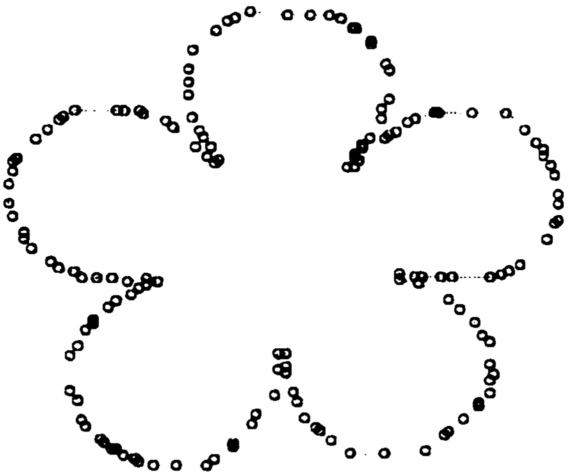


Figure 6.48: Fitted curve with corner and break points, threshold=1 (case 1).

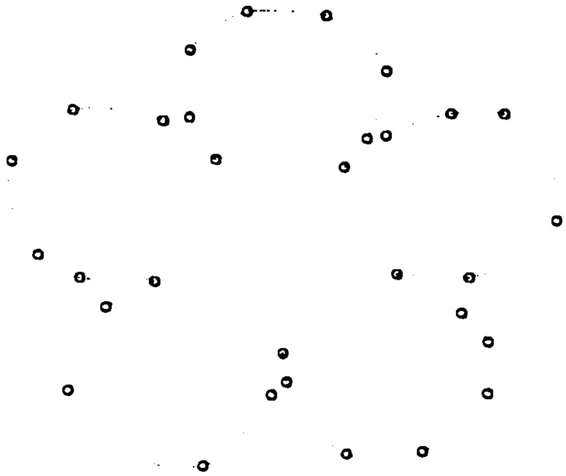


Figure 6.49: Fitted curve with corner and break points, threshold=3 (case 2).

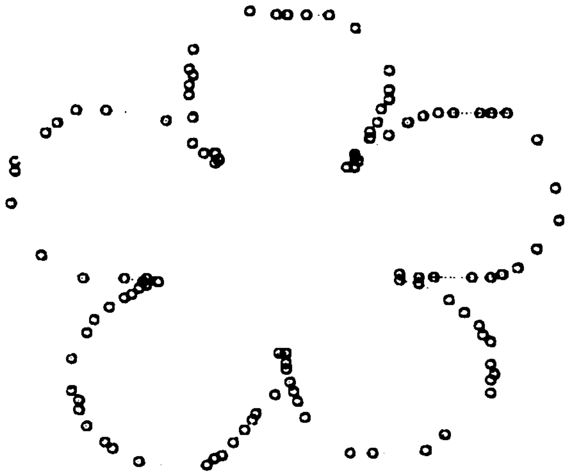


Figure 6.50: Fitted curve with corner and break points, threshold=1 (case 2).

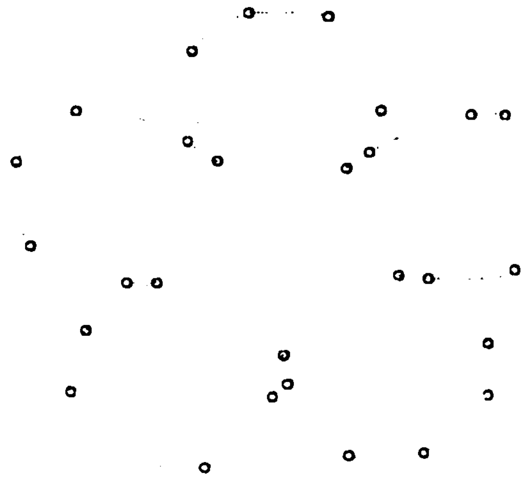


Figure 6.51: Fitted curve with corner and break points, threshold=3 (case 3).

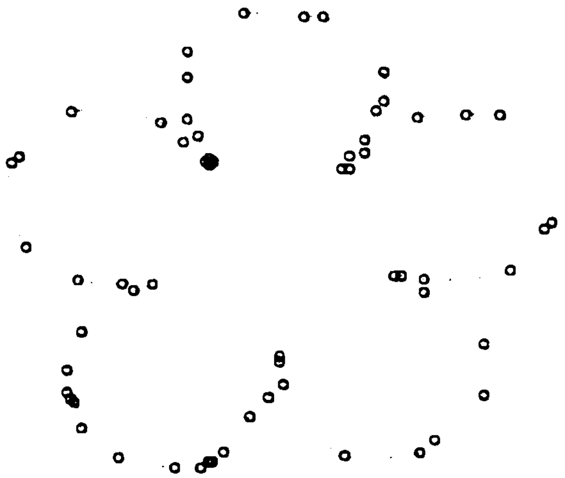


Figure 6.52: Fitted curve with corner and break points, threshold=1 (case 3).

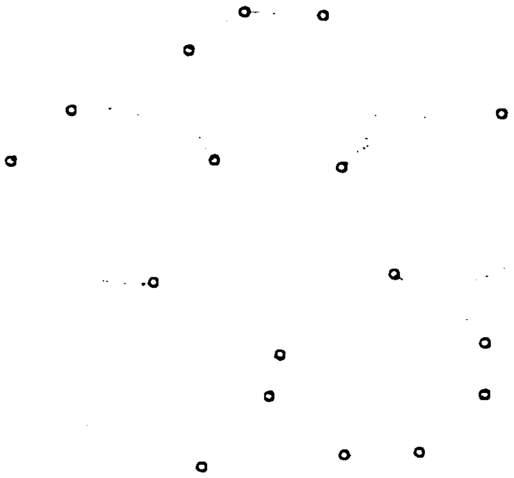


Figure 6.53: Fitted curve with corner and break points, threshold=3 (case 4).

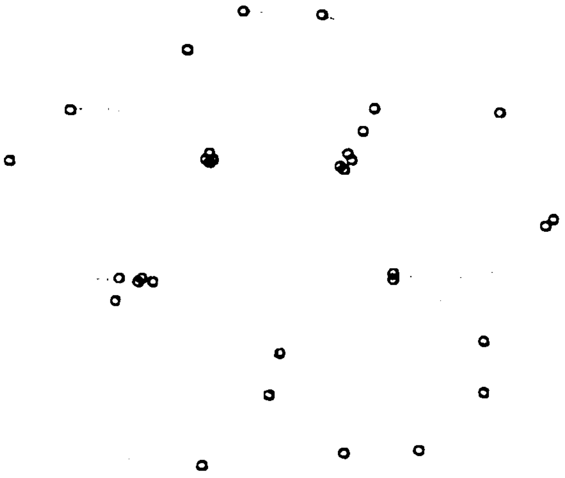


Figure 6.54: Fitted curve with corner and break points, threshold=1 (case 4).

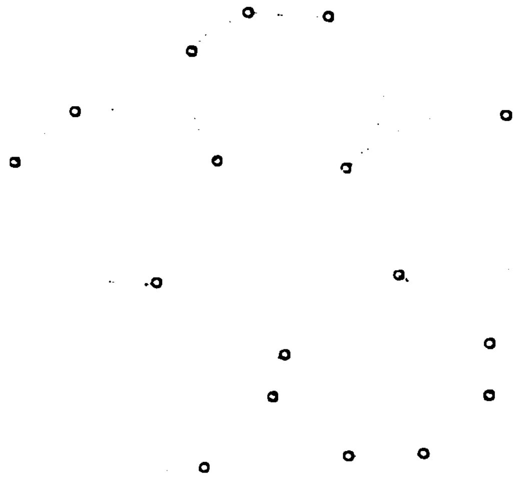


Figure 6.55: Fitted curve with corner and break points, threshold=3 (case 5).

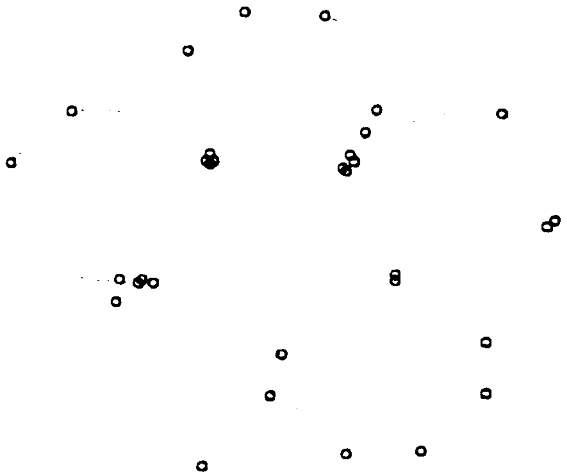


Figure 6.56: Fitted curve with corner and break points, threshold=1 (case 5).

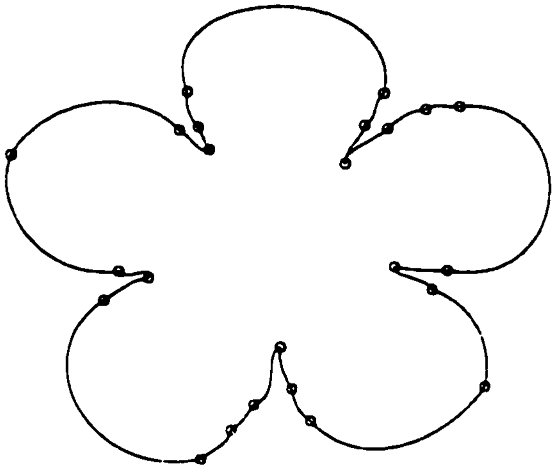


Figure 6.57: Final Outline without filtering and reparameterization, threshold=3

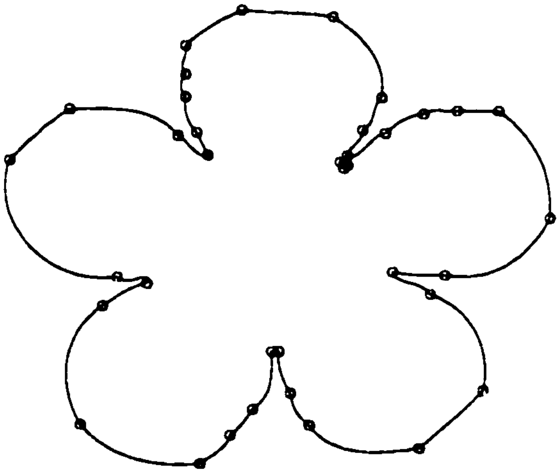


Figure 6.58: Final Outline without filtering and reparameterization, threshold=1

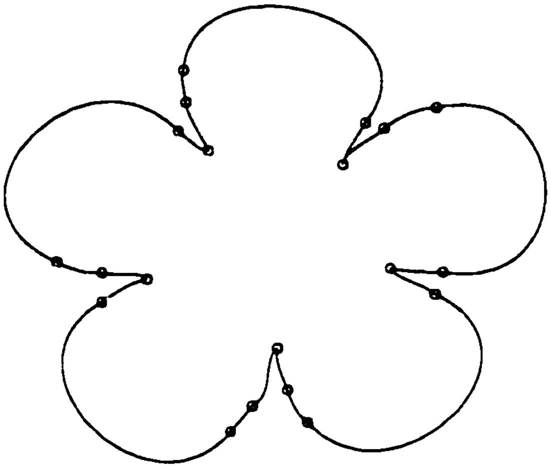


Figure 6.59: Final Outline without reparameterization, threshold=3

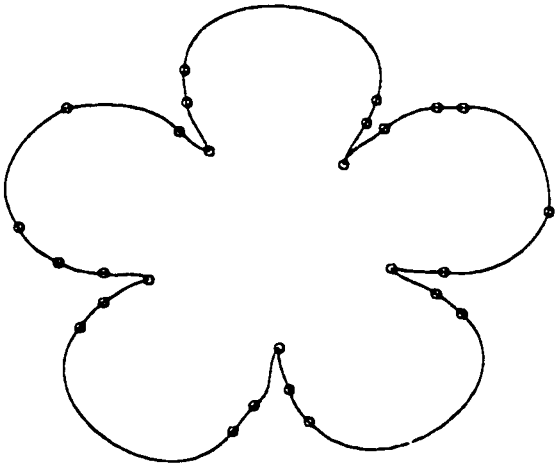


Figure 6.60: Final Outline without reparameterization, threshold=1

Figure #	# of Pieces	# of Boundary Points
6.1	2	1522+116=1638
6.2	3	870+102+67=1039
6.3	3	913+38+55=1006
6.4	1	587

Table 6.1: Statistics of images

	# of Corner Points	# of Break Points	Total # of Significant Points
case 1	10+3	55+3	71
case 2	10+3	51+1	65
case 3	10+3	30+1	44
case 4	10+3	5+0	18
case 5	10+3	5+0	18
without filtering and reparameterization [3].	9+3	19+0	31
with filtering and without reparameterization [3].	9+3	13+0	25

Table 6.2: Cases comparison for 'Lillah' image, threshold=3

	# of Corner Points	# of Break Points	Total # of Significant Points
case 1	10+3	593+33	639
case 2	10+3	269+21	303
case 3	10+3	110+10	133
case 4	10+3	39+1	53
case 5	10+3	39+1	53
without filtering and reparameterization [3].	9+3	27+2	41
with filtering and without reparameterization [3].	9+3	20+0	32

Table 6.3: Cases comparison for 'Lillah' image, threshold=1

	# of Corner Points	# of Break Points	Total # of Significant Points
case 1	25+3+3	23+2+1	57
case 2	25+3+3	13+3+1	48
case 3	25+3+3	12+1+1	45
case 4	25+3+3	3+0+0	34
case 5	25+3+3	3+0+0	34
without filtering and reparameterization [3].	23+3+3	6+1+1	37
with filtering and without reparameterization [3].	23+3+3	5+0+0	34

Table 6.4: Cases comparison for 'Kanji' image, threshold=3

	# of Corner Points	# of Break Points	Total # of Significant Points
case 1	25+3+3	235+16+13	295
case 2	25+3+3	126+20+9	186
case 3	25+3+3	60+6+5	102
case 4	25+3+3	19+2+1	53
case 5	25+3+3	19+2+1	53
without filtering and reparameterization [3].	23+3+3	19+3+1	52
with filtering and without reparameterization [3].	23+3+3	9+1+0	39

Table 6.5: Cases comparison for 'Kanji' image, threshold=1

	# of Corner Points	# of Break Points	Total # of Significant Points
case 1	28+3+2	20+1+2	56
case 2	28+3+2	16+1+2	52
case 3	28+3+2	7+0+2	42
case 4	28+3+2	1+0+0	34
case 5	28+3+2	1+0+0	34
without filtering and reparameterization [3].	23+3+3	5+0+0	34
with filtering and without reparameterization [3].	23+3+3	1+0+0	30

Table 6.6: Cases comparison for plane image, threshold=3

	# of Corner Points	# of Break Points	Total # of Significant Points
case 1	28+3+2	256+7+12	308
case 2	28+3+2	130+5+10	178
case 3	28+3+2	41+5+7	86
case 4	28+3+2	10+0+2	45
case 5	28+3+2	10+0+2	45
without filtering and reparameterization [3].	23+3+3	16+0+1	46
with filtering and without reparameterization [3].	23+3+3	10+0+0	39

Table 6.7: Cases comparison for plane image, threshold=1

	# of Corner Points	# of Break Points	Total # of Significant Points
case 1	17	14	31
case 2	17	14	31
case 3	17	11	28
case 4	17	0	17
case 5	17	0	17
without filtering and reparameterization [3].	6	17	23
with filtering and without reparameterization [3].	6	14	20

Table 6.8: Cases comparison for flower image, threshold=3

	# of Corner Points	# of Break Points	Total # of Significant Points
case 1	17	169	186
case 2	17	94	111
case 3	17	46	63
case 4	17	17	34
case 5	17	17	34
without filtering and reparameterization [3].	6	27	33
with filtering and without reparameterization [3].	6	19	25

Table 6.9: Cases comparison for flower image, threshold=1

S.No.	D_x	D_y
1	-18.4626	23.708
2	-11.136	1.68873
3	-17.5601	-15.9436
4	-40.3653	-10.2558
5	-26.3287	9.7598
6	-6.22969	17.0346
7	10.5441	26.4559
8	20.0743	33.0743
9	33.1195	29.1315
10	10.2893	17.657
11	2.5856	20.1547
12	5.36947	16.6947
13	4.43607	11.3082
14	4.39117	-0.17051
15	5	-16.9459
16	5	-18
17	4.43801	-15.752
18	3.43917	-12.3175
19	4.85981	-16.8119
20	9.15029	-27.368
21	17.3024	-17.3488
22	34.8027	2.60756
23	10.5154	25.5188
24	-6.19647	21.5894
25	-5	16.9459
26	-6.42278	20.624
27	-2.51514	35.0796
28	8.3969	25.1418
29	5.34826	-0.6501
30	3.49441	-12
31	5.30075	-16.2274
32	5.30075	-16.2274
33	4.77664	-14.7182
34	9.2307	-27.6152
35	13.2426	-31.9706
36	20.6231	-17.5608
37	15.0852	2.86204
38	3.68625	10.6788
39	-4.90575	17.3586
40	-6	19
41	-5.45691	17.3707
42	-7.14477	22.1279
43	-5.22732	40.5151
44	7.59246	30.2544
45	4.03533	-0.367517
46	3.02831	-10.3994
47	5.83333	-17.5
48	4.2	-12.6
49	3.72281	-11.5298
50	5.46907	-16.9381
51	6.46154	-19.3846
52	7	-21.4895
53	6.49691	-22
54	3.33781	-20.9351
55	-1.51922	-27.2428
56	-8.67459	-37.418
57	-14.6466	-13.7498
58	-21.7833	-2.52822
59	-21.9396	4.34021
60	-15	10
61	-11.2677	0.296313
62	-11.3519	-9.45063
63	-12.4499	-7.7998
64	-19.2769	-6.46948
65	-39.0769	10.1442
66	21.3177	3.54277
67	4.02683	11.1973
68	-4.45971	12.9194
69	-12.7816	4.77737
70	-11.6855	-12.7333
71	4.86052	-9.87308

Table 6.10: Tangent values for 'Lillah' image (case 1).

S.No.	v	D_x	D_y
1	0.6215	-18.4626	23.708
2	-0.0215	-10.5883	1.5869
3	0.6348	-16.7192	-15.3111
4	1.2517	-40.4148	-10.5823
5	0.6829	-29.3449	10.1612
6	0.5819	-5.8676	12.2507
7	0.1193	3.0996	11.3506
8	0.2336	6.5598	14.6076
9	1.2824	21.1196	35.4586
10	0.6944	35	32.6219
11	0.5306	13.9211	23.4799
12	1.0006	4.48065	24.5028
13	0.5178	6.7795	18.5075
14	0.1195	5.5092	-0.2139
15	0.8773	5	-16.9459
16	0.4289	4.438	-15.752
17	0.4301	5.3046	-18.8922
18	1.1503	10.0054	-34.6151
19	0.63	7.8516	-22.4062
20	0.605	9.8742	-12.5786
21	0.7995	19.5667	-5.1755
22	0.9922	29	10.3087
23	0.4925	8.2359	21.373
24	0.8424	-6	20.824
25	0.7628	-4.8114	15.4342
26	0.2431	-3.4343	11.3028
27	0.3659	-4.4228	14.2684
28	1.6271	-2.3188	32.8116
29	0.574	8	22.3402
30	-0.0938	5.9621	-0.775
31	0.7834	4	-13.3469
32	0.2263	3.4286	-10.2857
33	0.15	4.165	-12.7862
34	1.7594	10.8936	-33.194
35	0.6197	14.3074	-36.8516
36	0.9972	16.2514	-18.6984
37	1.3467	22.4909	-3.0607
38	0.3504	5.005	8.4612
39	-0.0306	-2	7.8825
40	0.3436	-3.1987	10.5961
41	0.9111	-7	22
42	0.8005	-5.8413	18.5239
43	0.2653	-3.7676	12.3027
44	0.4414	-4.5	15
45	1.3859	-1.9979	35.7664
46	0.5827	7.2593	21.7778
47	-0.1619	5.4458	-0.5061
48	0.7949	4	-12.9298
49	0.1649	3.4286	-10.2857
50	0.2848	4.5	-13.5
51	0.5607	10.6143	-32.2465
52	1.4101	13.7339	-51.3747
53	0.6323	1.7479	-52.6994
54	0.6752	-3.7569	-21.065
55	1.2465	-4.6569	-16.0711
56	0.5839	-12.3691	-10.9463
57	1.0177	-23.6061	-2.1126
58	0.6505	-28.8414	11.1586
59	0.3082	-22.4879	-0.053
60	1.2346	-34.0254	-17.2773
61	1.2736	-43.8943	9.8772
62	0.3278	9.323	17.5432
63	1.7848	-19.3373	5.6945
64	0.2738	2.0688	-11.3085
65	1.6773	20.9085	-0.4421

Table 6.11: Tangent & shape parameter (v) values for 'Lillah' image (case 2).

S. No.	v	w	D_x	D_y
1	0.698353	1.94471	-18.4626	23.708
2	0.672609	-0.17392	-8.68222	1.23336
3	1.59595	1.11002	-11.6464	-12.0387
4	0.640459	0.326515	-15.4404	-11.1836
5	2.63962	1.75417	-22.9357	-2.8334
6	0.806852	0.317219	-15.1679	28.0444
7	1.81093	1.86182	21.5606	43.3943
8	0.639846	1.16362	35.3525	32.7757
9	0.628886	0.380151	13.1942	21.6361
10	1.00067	0.99976	4.48065	24.5028
11	0.5152	0.672943	6.77945	18.5075
12	1.36447	0.0606299	5.50923	-0.213925
13	2.17796	0.747035	7.42731	-25.0473
14	1.17353	2.13096	23.6761	-63.3968
15	0.859512	0.661014	34.6925	-18.6769
16	0.757827	1.68284	28.3663	11.6337
17	0.507297	-0.00703357	4.57112	11.8399
18	1.69266	0.518342	-3.63601	14.3627
19	1.43485	1.1422	-10.921	35.0876
20	0.887761	0.915105	-4.0551	42.2879
21	0.572011	2.70994	8.41289	30.0355
22	0.441366	-0.140856	5.71489	-0.537142
23	2.0803	0.54379	5.27257	-18.3632
24	1.6231	2.51829	21.2651	-57.8633
25	0.576594	1.27645	31.1573	-25.128
26	0.784132	0.865922	14.4369	3.56614
27	0.804303	-0.0134002	2.91797	8.10582
28	2.04918	0.486117	-3.63038	13.6582
29	2.29478	2.12559	-11.9303	45.4546
30	0.398861	2.2157	3.9481	51.5865
31	0.2368	-0.121551	6.57143	-0.285714
32	2.06102	0.621343	6.5835	-21.2808
33	1.45786	1.2557	19.5846	-60.7185
34	0.838425	1.10478	7.71793	-71.092
35	0.717735	0.775623	-8.57239	-46.092
36	1.45963	1.38163	-30.6683	-22.6587
37	0.620531	0.674384	-27.8993	11.298
38	0.798676	0.321786	-18.9658	0.594818
39	1.41071	1.16926	-34.0254	-17.2773
40	1.24825	1.56227	-43.8943	9.87716
41	0.445517	0.181201	10.153	17.7294
42	1.60753	2.83696	-19.3373	5.69454
43	1.03551	-0.00894506	1.43137	-8.98366
44	2.15041	2.25427	16.9267	-1.2836

Table 6.12: Tangent & shape parameters (v,w) values for 'Lillah' image (case 3).

S. No.	D_{x1}	D_{y1}	D_{x2}	D_{y2}
1	-9.18167	10.0901	-11.7315	-3.20473
2	-72.0141	-136.029	-93.2907	75.7112
3	37.0497	145.831	142.641	61.2406
4	-6.2534	71.347	24.1931	50.7751
5	63.782	-135.085	113.954	-100.745
6	38.3418	-9.91867	25.6115	13.0955
7	21.522	11.397	10.201	19.6185
8	-33.0239	141.746	105.744	129.587
9	56.6044	-156.22	84.6548	-139.624
10	13.8382	-9.7964	11.3133	-3.91049
11	26.4554	-1.19687	8.45233	23.6315
12	-33.0393	165.876	109.348	157.321
13	66.7496	-234.117	-123.845	-243.59
14	-74.4781	-30.8481	-44.9859	68.1507
15	-83.801	-114.652	-34.7117	164.072
16	-8.69017	25.2321	-11.9742	18.9057
17	-40.6759	-14.4403	-4.89211	-43.6354
18	40.8642	-24.2447	30.2858	40.6132

Table 6.13: Tangent values for 'Lillah' image (case 4).

S. No.	v	w	D_{x1}	D_{y1}	D_{x2}	D_{y2}
1	1	1	-9.18167	10.0901	-11.7315	-3.20473
2	1	1	-72.0141	-136.029	-93.2907	75.7112
3	1	1	37.0497	145.831	142.641	61.2406
4	1	1	-6.2534	71.347	24.1931	50.7751
5	1	1	63.782	-135.085	113.954	-100.745
6	1	1	38.3418	-9.91867	25.6115	13.0955
7	1	1	21.522	11.397	10.201	19.6185
8	1	1	-33.0239	141.746	105.744	129.587
9	1	1	56.6044	-156.22	84.6548	-139.624
10	1	1	13.8382	-9.7964	11.3133	-3.91049
11	1	1	26.4554	-1.19687	8.45233	23.6315
12	1	1	-33.0393	165.876	109.348	157.321
13	1	1	66.7496	-234.117	-123.845	-243.59
14	1	1	-74.4781	-30.8481	-44.9859	68.1507
15	1	1	-83.801	-114.652	-34.7117	164.072
16	1	1	-8.69017	25.2321	-11.9742	18.9057
17	1	1	-40.6759	-14.4403	-4.89211	-43.6354
18	1	1	40.8642	-24.2447	30.2858	40.6132

Table 6.14: Tangent & shape parameters (v,w) values for 'Lillah' image (case 5).

6.1 Comparison with Previous Work

The work presented in this thesis is motivated by [3] and is an alternate approach. In addition to some small variations, the overall methodology mainly differs at a major step of curve fitting technique. A comparative study with one of the image is presented in the next paragraph.

The mathematical model of traditional Bézier was used in [3] whereas a generalized form of Hermite cubic is used in this thesis. Along with the shape parameters, it is believed that one can have a more control over the shape of the curve. Also there exists a C^1 continuity in fitting the curve piecewisely which was not present in [3]. It is believed that this will be quite useful especially in the case of smooth curves. Another difference lies in breaking the segment when the error between the digitized and fitted curve is larger than some threshold value. In [3], distance formula is used to calculate the error whereas the x and y pixel differences in both the horizontal and vertical directions is used in this thesis. Although in a high computing environment it does not make much difference but calculating the square values has been saved. Also no filtering and reparameterization is done to get the optimal curve fit. As a comparison for the 'Lillah' (Figure 6.1) image, there were 30 total number of significant points without filtering and 25 with filtering when the threshold value was 3. In the best case here it is fitted with 18 total number of significant points when the threshold value is 3. The detailed comparison for this figure along with three more images can be seen in Tables 6.2 to 6.9. Moreover the application has been deployed over the Web and potential users can test their images with ease and varying some parameters.

Chapter 7

Conclusion and Future Work

An algorithm for the approximation of boundary of digital images has been presented which can be particularly used in font designing method. In addition to the detection of corner points, a strategy to detect a set of significant points is also explained to optimize the outline. A more flexible class of cubic functions is used to fit the curve. It eliminates the human interaction in obtaining the outline of digitized image. Because of a modular design, the system is flexible and adaptable for future enhancements. Such a work is still in progress and some more elegant results can be achieved by using different parametric technique, segment subdivision and cubic model. A World Wide Web based interface has been created for automatic outline capturing of images. An easy-to-use interface is developed whereby users can test the cases discussed in this thesis. In addition to the images available, users can upload their own images. It is a simple interface and more features will be added in the future.

7.1 Future Work

- **Web Application Enhancement**: Several enhancements to the described system will be feasible within short. In addition to the uploading of images a user can be given a free hand to draw any kind of shape, save it and then submit to test the algorithm. Also a user can vary more parameters such as d_{min}, d_{max} and α_{max} of corner detection algorithms and can select edge detection method from various available methods.

- **Distributed/Parallel processing**: The nature of the system makes it suitable for distributed/parallel processing. It can be applied to each piece of the curve and corner detection, breaking the segments,fitting curve to segments can be done separately for each piece.

- **Going for three dimensional and Open curves**: It will be interesting to investigate and extend the method so that it can work for 3D images. Currently the method is applicable for only closed curves. It can be changed with slight ease to tackle with the Open curves and it might be necessary when the Web application is enhanced, so that user can draw any type of image.

- **Higher Order or Rational cubic curves**: The generalized Hermite cubic is used in this thesis. It is possible to use curves of degree four or five. Although using higher order curves will be expensive in terms of computation, but it can be worth to investigate if higher order curves can reduce the number of segments required to produce the outline. Also a curve can be fitted using rational cubics.

- **Colored Images**: Another possibility is to test the colored images. The main change might occur in detecting the edges and an additional field of color must be saved in the file.

Appendix A

Extracted Boundary File

A.1 Lillah.txt

The original data file consist of three columns but to save space it is depicted here in twelve columns. The first column corresponds to the x coordinate of the image and the second column corresponds to the y coordinate of the image. The third column is the index of these x and y coordinates. The same sequence is repeated in the next nine columns.

```
2       -----   Total number of segments
1522    -----   Number of data points in first segment
116     -----   Number of data points in second segment
```

164	25	1	163	25	2	162	25	3	161	25	4
160	26	5	159	26	6	158	26	7	157	27	8
156	27	9	155	27	10	154	28	11	153	28	12
152	29	13	151	29	14	150	30	15	149	31	16
148	31	17	147	32	18	146	33	19	145	34	20
144	35	21	143	36	22	142	37	23	141	38	24
140	39	25	139	40	26	138	41	27	138	42	28
137	43	29	136	44	30	136	45	31	135	46	32
135	47	33	134	48	34	133	49	35	133	50	36
132	51	37	132	52	38	131	53	39	131	54	40
130	55	41	130	56	42	130	57	43	129	58	44
129	59	45	128	60	46	128	61	47	128	62	48
127	63	49	127	64	50	126	65	51	126	66	52
126	67	53	125	68	54	125	69	55	125	70	56
124	71	57	124	72	58	124	73	59	123	74	60
122	74	61	121	74	62	120	73	63	119	72	64
118	71	65	117	70	66	116	69	67	115	68	68
114	67	69	113	66	70	113	65	71	112	64	72
111	63	73	110	62	74	109	61	75	108	60	76
107	59	77	106	58	78	105	57	79	104	56	80
103	55	81	102	54	82	101	53	83	100	52	84
99	52	85	98	51	86	97	50	87	96	49	88
95	48	89	94	47	90	93	46	91	92	46	92
91	45	93	90	44	94	89	44	95	88	43	96
87	42	97	86	42	98	85	41	99	84	41	100
83	40	101	82	40	102	81	40	103	80	39	104
79	39	105	78	39	106	77	39	107	76	39	108
75	39	109	74	39	110	73	39	111	72	39	112
71	39	113	70	39	114	69	39	115	68	39	116
67	39	117	66	39	118	65	39	119	64	39	120
63	39	121	62	40	122	61	40	123	60	40	124
59	40	125	58	40	126	57	40	127	56	40	128
55	40	129	54	40	130	53	40	131	52	41	132
51	41	133	50	41	134	49	41	135	48	41	136
47	41	137	46	41	138	45	42	139	44	42	140
43	42	141	42	42	142	41	42	143	40	43	144
39	43	145	38	43	146	37	43	147	36	43	148
35	44	149	34	44	150	33	44	151	32	44	152
31	45	153	30	45	154	29	45	155	28	45	156
27	46	157	26	46	158	25	46	159	24	47	160
23	47	161	22	47	162	21	48	163	20	48	164

19	48	165	18	49	166	17	49	167	16	49	168
15	50	169	14	50	170	13	51	171	12	51	172
11	52	173	10	52	174	9	53	175	8	54	176
7	55	177	6	56	178	5	57	179	5	58	180
5	59	181	5	60	182	5	61	183	5	62	184
5	63	185	5	64	186	6	65	187	6	66	188
6	67	189	7	68	190	7	69	191	7	70	192
7	71	193	8	72	194	8	73	195	9	74	196
9	75	197	9	76	198	10	77	199	10	78	200
11	79	201	11	80	202	11	81	203	12	82	204
12	83	205	13	84	206	13	85	207	14	86	208
14	87	209	15	88	210	15	89	211	16	90	212
16	91	213	17	92	214	17	93	215	18	94	216
18	95	217	19	96	218	19	97	219	20	98	220
20	99	221	21	100	222	21	101	223	22	102	224
22	103	225	23	104	226	23	105	227	24	106	228
24	107	229	25	108	230	25	109	231	26	110	232
26	111	233	27	112	234	27	113	235	28	114	236
28	115	237	29	116	238	30	117	239	30	118	240
31	119	241	31	120	242	32	121	243	33	122	244
33	123	245	34	124	246	34	125	247	35	126	248
36	127	249	36	128	250	37	129	251	37	130	252
38	131	253	39	132	254	39	133	255	40	134	256
41	135	257	41	136	258	42	137	259	43	138	260
43	139	261	44	140	262	45	141	263	46	142	264
46	143	265	47	144	266	48	145	267	49	146	268
49	147	269	50	148	270	51	149	271	52	150	272
53	151	273	54	152	274	54	153	275	55	154	276
56	155	277	57	156	278	58	157	279	59	158	280
60	159	281	61	159	282	62	160	283	63	161	284
64	162	285	65	163	286	66	163	287	67	164	288
68	164	289	69	165	290	70	165	291	71	166	292
72	166	293	73	167	294	74	167	295	75	168	296
76	169	297	77	169	298	78	170	299	79	170	300
80	171	301	81	171	302	82	172	303	83	172	304
84	173	305	85	173	306	86	174	307	87	174	308
88	175	309	89	176	310	90	176	311	91	177	312
92	177	313	93	178	314	94	179	315	95	180	316
95	181	317	95	182	318	95	183	319	95	184	320
95	185	321	95	186	322	95	187	323	95	188	324
95	189	325	95	190	326	95	191	327	95	192	328

95	193	329	95	194	330	95	195	331	95	196	332
95	197	333	95	198	334	96	199	335	96	200	336
96	201	337	96	202	338	96	203	339	96	204	340
96	205	341	97	206	342	97	207	343	97	208	344
97	209	345	97	210	346	98	211	347	98	212	348
98	213	349	98	214	350	99	215	351	99	216	352
99	217	353	100	218	354	100	219	355	100	220	356
101	221	357	101	222	358	101	223	359	102	224	360
102	225	361	102	226	362	103	227	363	103	228	364
104	229	365	104	230	366	104	231	367	105	232	368
105	233	369	106	234	370	106	235	371	106	236	372
107	237	373	107	238	374	108	239	375	108	240	376
109	241	377	109	242	378	109	243	379	109	244	380
110	244	381	111	244	382	111	243	383	111	242	384
112	241	385	112	240	386	112	239	387	113	238	388
113	237	389	113	236	390	113	235	391	114	234	392
114	233	393	114	232	394	115	231	395	115	230	396
115	229	397	115	228	398	116	227	399	116	226	400
116	225	401	116	224	402	117	223	403	117	222	404
117	221	405	118	220	406	118	219	407	118	218	408
118	217	409	119	216	410	119	215	411	119	214	412
120	213	413	120	212	414	120	211	415	120	210	416
121	209	417	121	208	418	121	207	419	122	206	420
122	205	421	122	204	422	122	203	423	123	202	424
123	201	425	123	200	426	124	199	427	124	198	428
124	197	429	124	196	430	125	195	431	125	194	432
125	193	433	125	192	434	126	191	435	126	190	436
126	189	437	127	188	438	127	187	439	127	186	440
127	185	441	128	184	442	128	183	443	128	182	444
129	181	445	129	180	446	129	179	447	129	178	448
130	177	449	130	176	450	130	175	451	131	174	452
131	173	453	131	172	454	131	171	455	132	170	456
132	169	457	132	168	458	132	167	459	133	166	460
133	165	461	133	164	462	134	163	463	134	162	464
134	161	465	134	160	466	135	159	467	135	158	468
135	157	469	136	156	470	136	155	471	136	154	472
136	153	473	137	152	474	137	151	475	137	150	476
138	149	477	138	148	478	138	147	479	138	146	480
139	145	481	139	144	482	139	143	483	139	142	484
140	141	485	140	140	486	140	139	487	141	138	488
141	137	489	141	136	490	141	135	491	142	134	492

142	133	493	142	132	494	143	131	495	143	130	496
143	129	497	143	128	498	144	127	499	144	126	500
144	125	501	145	124	502	145	123	503	145	122	504
146	121	505	146	120	506	146	119	507	147	118	508
147	117	509	148	116	510	148	115	511	149	114	512
149	113	513	150	112	514	150	111	515	151	110	516
151	109	517	152	108	518	153	107	519	154	106	520
155	105	521	156	104	522	157	103	523	158	103	524
159	102	525	160	102	526	161	101	527	162	101	528
163	100	529	164	100	530	165	99	531	166	99	532
167	99	533	168	99	534	169	99	535	170	98	536
171	98	537	172	98	538	173	98	539	174	98	540
175	98	541	176	98	542	177	98	543	178	98	544
179	98	545	180	98	546	181	98	547	182	99	548
183	99	549	184	99	550	185	99	551	186	99	552
187	99	553	188	100	554	189	100	555	190	100	556
191	101	557	192	101	558	193	101	559	194	102	560
195	102	561	196	102	562	197	103	563	198	103	564
199	104	565	200	104	566	201	105	567	202	105	568
203	106	569	204	106	570	205	107	571	206	107	572
207	108	573	208	108	574	209	109	575	210	109	576
211	110	577	212	111	578	213	111	579	214	112	580
215	113	581	216	114	582	217	114	583	218	115	584
219	116	585	220	117	586	221	118	587	222	119	588
223	120	589	223	121	590	223	122	591	223	123	592
223	124	593	222	125	594	222	126	595	222	127	596
221	128	597	221	129	598	221	130	599	221	131	600
220	132	601	220	133	602	220	134	603	219	135	604
219	136	605	219	137	606	218	138	607	218	139	608
218	140	609	217	141	610	217	142	611	217	143	612
217	144	613	216	145	614	216	146	615	216	147	616
215	148	617	215	149	618	215	150	619	214	151	620
214	152	621	214	153	622	213	154	623	213	155	624
213	156	625	213	157	626	212	158	627	212	159	628
212	160	629	211	161	630	211	162	631	211	163	632
210	164	633	210	165	634	210	166	635	209	167	636
209	168	637	209	169	638	208	170	639	208	171	640
208	172	641	208	173	642	207	174	643	207	175	644
207	176	645	206	177	646	206	178	647	206	179	648
205	180	649	205	181	650	205	182	651	204	183	652
204	184	653	204	185	654	204	186	655	203	187	656

203	188	657	203	189	658	202	190	659	202	191	660
202	192	661	201	193	662	201	194	663	201	195	664
200	196	665	200	197	666	200	198	667	199	199	668
199	200	669	199	201	670	199	202	671	198	203	672
198	204	673	198	205	674	198	206	675	197	207	676
197	208	677	197	209	678	197	210	679	196	211	680
196	212	681	196	213	682	196	214	683	196	215	684
196	216	685	196	217	686	196	218	687	196	219	688
196	220	689	196	221	690	196	222	691	196	223	692
196	224	693	196	225	694	196	226	695	196	227	696
196	228	697	196	229	698	196	230	699	196	231	700
197	232	701	197	233	702	197	234	703	197	235	704
197	236	705	197	237	706	198	238	707	198	239	708
198	240	709	198	241	710	199	242	711	199	243	712
199	244	713	199	245	714	200	246	715	200	247	716
200	248	717	200	249	718	201	250	719	201	251	720
201	252	721	202	253	722	202	254	723	203	255	724
203	256	725	203	257	726	204	258	727	204	259	728
204	260	729	205	261	730	205	262	731	206	263	732
206	264	733	207	265	734	207	266	735	208	267	736
208	268	737	209	269	738	209	270	739	210	271	740
210	272	741	211	272	742	212	272	743	212	271	744
212	270	745	213	269	746	213	268	747	213	267	748
214	266	749	214	265	750	214	264	751	215	263	752
215	262	753	215	261	754	215	260	755	216	259	756
216	258	757	216	257	758	217	256	759	217	255	760
217	254	761	218	253	762	218	252	763	218	251	764
219	250	765	219	249	766	219	248	767	220	247	768
220	246	769	220	245	770	221	244	771	221	243	772
221	242	773	222	241	774	222	240	775	222	239	776
223	238	777	223	237	778	223	236	779	224	235	780
224	234	781	224	233	782	225	232	783	225	231	784
225	230	785	225	229	786	226	228	787	226	227	788
226	226	789	227	225	790	227	224	791	227	223	792
228	222	793	228	221	794	228	220	795	229	219	796
229	218	797	229	217	798	230	216	799	230	215	800
230	214	801	231	213	802	231	212	803	231	211	804
232	210	805	232	209	806	232	208	807	233	207	808
233	206	809	233	205	810	234	204	811	234	203	812
234	202	813	235	201	814	235	200	815	235	199	816
236	198	817	236	197	818	236	196	819	236	195	820

237	194	821	237	193	822	237	192	823	238	191	824
238	190	825	238	189	826	239	188	827	239	187	828
239	186	829	240	185	830	240	184	831	240	183	832
241	182	833	241	181	834	241	180	835	242	179	836
242	178	837	242	177	838	243	176	839	243	175	840
243	174	841	244	173	842	244	172	843	244	171	844
245	170	845	245	169	846	245	168	847	246	167	848
246	166	849	246	165	850	246	164	851	247	163	852
247	162	853	247	161	854	248	160	855	248	159	856
248	158	857	249	157	858	249	156	859	249	155	860
250	154	861	250	153	862	251	152	863	251	151	864
251	150	865	252	149	866	252	148	867	252	147	868
253	146	869	253	145	870	253	144	871	254	143	872
254	142	873	255	141	874	255	140	875	255	139	876
256	138	877	256	137	878	257	136	879	257	135	880
258	134	881	258	133	882	258	132	883	259	131	884
259	130	885	260	129	886	260	128	887	261	127	888
261	126	889	262	125	890	262	124	891	263	123	892
264	122	893	264	121	894	265	120	895	266	119	896
266	118	897	267	117	898	268	116	899	269	115	900
270	115	901	271	114	902	272	113	903	273	113	904
274	112	905	275	112	906	276	111	907	277	111	908
278	110	909	279	110	910	280	110	911	281	109	912
282	109	913	283	109	914	284	109	915	285	108	916
286	108	917	287	108	918	288	108	919	289	108	920
290	108	921	291	108	922	292	108	923	293	108	924
294	108	925	295	108	926	296	108	927	297	108	928
298	108	929	299	108	930	300	108	931	301	109	932
302	109	933	303	109	934	304	110	935	305	110	936
306	111	937	307	112	938	308	113	939	308	114	940
309	115	941	309	116	942	309	117	943	309	118	944
308	119	945	308	120	946	308	121	947	307	122	948
307	123	949	307	124	950	306	125	951	306	126	952
306	127	953	305	128	954	305	129	955	305	130	956
305	131	957	304	132	958	304	133	959	304	134	960
303	135	961	303	136	962	303	137	963	302	138	964
302	139	965	302	140	966	301	141	967	301	142	968
301	143	969	300	144	970	300	145	971	300	146	972
299	147	973	299	148	974	299	149	975	299	150	976
298	151	977	298	152	978	298	153	979	297	154	980
297	155	981	297	156	982	296	157	983	296	158	984

296	159	985	295	160	986	295	161	987	295	162	988
294	163	989	294	164	990	294	165	991	294	166	992
293	167	993	293	168	994	293	169	995	292	170	996
292	171	997	292	172	998	291	173	999	291	174	1000
291	175	1001	290	176	1002	290	177	1003	290	178	1004
289	179	1005	289	180	1006	289	181	1007	288	182	1008
288	183	1009	288	184	1010	288	185	1011	287	186	1012
287	187	1013	287	188	1014	286	189	1015	286	190	1016
286	191	1017	285	192	1018	285	193	1019	285	194	1020
284	195	1021	284	196	1022	284	197	1023	283	198	1024
283	199	1025	283	200	1026	283	201	1027	282	202	1028
282	203	1029	282	204	1030	281	205	1031	281	206	1032
281	207	1033	280	208	1034	280	209	1035	280	210	1036
279	211	1037	279	212	1038	279	213	1039	278	214	1040
278	215	1041	278	216	1042	278	217	1043	277	218	1044
277	219	1045	277	220	1046	276	221	1047	276	222	1048
276	223	1049	276	224	1050	276	225	1051	275	226	1052
275	227	1053	275	228	1054	275	229	1055	275	230	1056
274	231	1057	274	232	1058	274	233	1059	274	234	1060
274	235	1061	274	236	1062	274	237	1063	274	238	1064
274	239	1065	274	240	1066	274	241	1067	274	242	1068
274	243	1069	274	244	1070	274	245	1071	274	246	1072
274	247	1073	274	248	1074	275	249	1075	275	250	1076
275	251	1077	275	252	1078	275	253	1079	275	254	1080
276	255	1081	276	256	1082	276	257	1083	276	258	1084
276	259	1085	277	260	1086	277	261	1087	277	262	1088
278	263	1089	278	264	1090	278	265	1091	278	266	1092
279	267	1093	279	268	1094	279	269	1095	280	270	1096
280	271	1097	280	272	1098	281	273	1099	281	274	1100
282	275	1101	282	276	1102	282	277	1103	283	278	1104
283	279	1105	284	280	1106	284	281	1107	284	282	1108
285	283	1109	285	284	1110	286	285	1111	286	286	1112
287	287	1113	287	288	1114	287	289	1115	288	289	1116
289	289	1117	289	288	1118	289	287	1119	290	286	1120
290	285	1121	290	284	1122	291	283	1123	291	282	1124
291	281	1125	292	280	1126	292	279	1127	292	278	1128
293	277	1129	293	276	1130	293	275	1131	294	274	1132
294	273	1133	294	272	1134	295	271	1135	295	270	1136
295	269	1137	296	268	1138	296	267	1139	296	266	1140
297	265	1141	297	264	1142	297	263	1143	298	262	1144
298	261	1145	298	260	1146	299	259	1147	299	258	1148

299	257	1149	300	256	1150	300	255	1151	300	254	1152
301	253	1153	301	252	1154	301	251	1155	302	250	1156
302	249	1157	302	248	1158	303	247	1159	303	246	1160
303	245	1161	304	244	1162	304	243	1163	304	242	1164
305	241	1165	305	240	1166	305	239	1167	306	238	1168
306	237	1169	306	236	1170	307	235	1171	307	234	1172
307	233	1173	308	232	1174	308	231	1175	308	230	1176
309	229	1177	309	228	1178	309	227	1179	310	226	1180
310	225	1181	310	224	1182	311	223	1183	311	222	1184
311	221	1185	311	220	1186	312	219	1187	312	218	1188
313	217	1189	313	216	1190	313	215	1191	313	214	1192
314	213	1193	314	212	1194	314	211	1195	315	210	1196
315	209	1197	315	208	1198	316	207	1199	316	206	1200
316	205	1201	317	204	1202	317	203	1203	317	202	1204
318	201	1205	318	200	1206	318	199	1207	319	198	1208
319	197	1209	319	196	1210	320	195	1211	320	194	1212
320	193	1213	321	192	1214	321	191	1215	321	190	1216
322	189	1217	322	188	1218	322	187	1219	323	186	1220
323	185	1221	323	184	1222	324	183	1223	324	182	1224
324	181	1225	325	180	1226	325	179	1227	325	178	1228
326	177	1229	326	176	1230	326	175	1231	327	174	1232
327	173	1233	327	172	1234	328	171	1235	328	170	1236
328	169	1237	329	168	1238	329	167	1239	329	166	1240
330	165	1241	330	164	1242	330	163	1243	331	162	1244
331	161	1245	331	160	1246	331	159	1247	332	158	1248
332	157	1249	332	156	1250	333	155	1251	333	154	1252
333	153	1253	333	152	1254	334	151	1255	334	150	1256
334	149	1257	334	148	1258	335	147	1259	335	146	1260
335	145	1261	335	144	1262	336	143	1263	336	142	1264
336	141	1265	336	140	1266	336	139	1267	336	138	1268
337	137	1269	337	136	1270	337	135	1271	337	134	1272
337	133	1273	337	132	1274	337	131	1275	337	130	1276
337	129	1277	337	128	1278	338	127	1279	338	126	1280
338	125	1281	338	124	1282	338	123	1283	338	122	1284
338	121	1285	338	120	1286	338	119	1287	338	118	1288
338	117	1289	337	116	1290	337	115	1291	337	114	1292
337	113	1293	337	112	1294	337	111	1295	337	110	1296
337	109	1297	337	108	1298	337	107	1299	337	106	1300
336	105	1301	336	104	1302	336	103	1303	336	102	1304
336	101	1305	336	100	1306	336	99	1307	336	98	1308
335	97	1309	335	96	1310	335	95	1311	335	94	1312

335	93	1313	335	92	1314	334	91	1315	334	90	1316
334	89	1317	334	88	1318	334	87	1319	334	86	1320
333	85	1321	333	84	1322	333	83	1323	333	82	1324
333	81	1325	333	80	1326	332	79	1327	332	78	1328
332	77	1329	332	76	1330	332	75	1331	331	74	1332
331	73	1333	331	72	1334	331	71	1335	331	70	1336
330	69	1337	330	68	1338	330	67	1339	330	66	1340
330	65	1341	329	64	1342	329	63	1343	329	62	1344
329	61	1345	328	60	1346	328	59	1347	328	58	1348
328	57	1349	328	56	1350	327	55	1351	327	54	1352
327	53	1353	327	52	1354	326	51	1355	326	50	1356
326	49	1357	325	48	1358	325	47	1359	325	46	1360
324	45	1361	324	44	1362	323	43	1363	322	42	1364
321	41	1365	320	40	1366	319	39	1367	318	39	1368
317	38	1369	316	38	1370	315	38	1371	314	38	1372
313	37	1373	312	37	1374	311	37	1375	310	37	1376
309	37	1377	308	37	1378	307	37	1379	306	37	1380
305	37	1381	304	37	1382	303	36	1383	302	36	1384
301	36	1385	300	36	1386	299	36	1387	298	36	1388
297	36	1389	296	36	1390	295	36	1391	294	36	1392
293	36	1393	292	36	1394	291	36	1395	290	36	1396
289	36	1397	288	36	1398	287	36	1399	286	36	1400
285	36	1401	284	36	1402	283	36	1403	282	37	1404
281	37	1405	280	37	1406	279	37	1407	278	37	1408
277	37	1409	276	37	1410	275	37	1411	274	37	1412
273	38	1413	272	38	1414	271	38	1415	270	38	1416
269	38	1417	268	39	1418	267	39	1419	266	39	1420
265	40	1421	264	40	1422	263	41	1423	262	41	1424
261	42	1425	260	42	1426	259	43	1427	258	44	1428
257	44	1429	256	45	1430	255	46	1431	254	47	1432
253	48	1433	252	49	1434	251	50	1435	250	51	1436
249	52	1437	249	53	1438	248	54	1439	247	55	1440
246	56	1441	245	57	1442	244	57	1443	243	57	1444
242	56	1445	241	55	1446	240	54	1447	239	53	1448
238	52	1449	237	51	1450	236	50	1451	235	49	1452
234	48	1453	233	47	1454	232	46	1455	231	46	1456
230	45	1457	229	44	1458	228	43	1459	227	42	1460
226	42	1461	225	41	1462	224	40	1463	223	40	1464
222	39	1465	221	38	1466	220	38	1467	219	37	1468
218	36	1469	217	36	1470	216	35	1471	215	35	1472
214	34	1473	213	33	1474	212	33	1475	211	32	1476

210	32	1477	209	32	1478	208	31	1479	207	31	1480
206	30	1481	205	30	1482	204	29	1483	203	29	1484
202	29	1485	201	28	1486	200	28	1487	199	28	1488
198	27	1489	197	27	1490	196	27	1491	195	26	1492
194	26	1493	193	26	1494	192	26	1495	191	25	1496
190	25	1497	189	25	1498	188	25	1499	187	25	1500
186	25	1501	185	24	1502	184	24	1503	183	24	1504
182	24	1505	181	24	1506	180	24	1507	179	24	1508
178	24	1509	177	24	1510	176	24	1511	175	24	1512
174	24	1513	173	24	1514	172	24	1515	171	24	1516
170	24	1517	169	24	1518	168	24	1519	167	24	1520
166	24	1521	164	25	1522						

89	106	1	90	106	2	91	106	3	92	106	4
93	106	5	94	106	6	95	106	7	96	106	8
97	106	9	98	107	10	99	107	11	100	107	12
101	108	13	102	108	14	103	109	15	104	109	16
105	110	17	106	110	18	107	111	19	108	112	20
109	112	21	110	113	22	111	114	23	112	115	24
113	116	25	113	117	26	112	118	27	112	119	28
112	120	29	112	121	30	111	122	31	111	123	32
111	124	33	110	125	34	110	126	35	110	127	36
109	128	37	109	129	38	109	130	39	108	131	40
108	132	41	108	133	42	107	134	43	107	135	44
107	136	45	107	137	46	106	138	47	106	139	48
106	140	49	105	141	50	104	142	51	103	142	52
102	142	53	101	141	54	100	141	55	99	140	56
98	140	57	97	140	58	96	139	59	95	139	60
94	138	61	93	138	62	92	137	63	91	137	64
90	136	65	89	136	66	88	135	67	87	134	68
86	134	69	85	133	70	84	133	71	83	132	72
82	131	73	81	131	74	80	130	75	79	129	76
78	129	77	77	128	78	76	127	79	75	126	80
74	126	81	73	125	82	72	124	83	71	123	84
70	122	85	69	121	86	69	120	87	68	119	88
68	118	89	67	117	90	67	116	91	67	115	92
67	114	93	67	113	94	68	112	95	69	111	96

70	111	97	71	111	98	72	110	99	73	110	100
74	110	101	75	109	102	76	109	103	77	109	104
78	109	105	79	108	106	80	108	107	81	108	108
82	108	109	83	108	110	84	107	111	85	107	112
86	107	113	87	107	114	88	107	115	89	106	116

Bibliography

[1] Philip J. Schneider. Phoenix:An Interactive Curve Design System Based on the Automatic Fitting of Hand Sketched Curves. Master's thesis, University of Washington, 1988.

[2] Michael Plass and Maureen Stone. Curve-fitting with piecewise parametric cubics. *Computer Graphics*, 17(3):229–239, July 1983.

[3] Murtaza Ali Khan. An Efficient Font Design Method. Master's thesis, KFUPM, January 2001.

[4] Abuhaiba I. S. I., Datto S., and Holt M. J. J. Straight Line Approximation and ID representation of Off-line Hand-Written Text. *Image and Vision Computing, Elsevier Science*, 13(10):755–769, July 1994.

[5] Chang H. and Yan H. Vectorization of Hand-Drawn Image using Piecewise Cubic Bezier Curves Fitting. *Pattern Recognition, Elsevier Science*, 31(11):1747–1755, 1998.

[6] Hussain Fiaz. *On the Capture and Representation of Fonts.* PhD thesis, Brunel University, England, December 1991.

[7] M. Sarfraz. Cubic Spline Curves with Shape Control. *Comuter and Graphics,* 18(5):707–713, 1994.

[8] M. Sarfraz, M. N. Haque, and M. A. Khan. Capturing Outlines of 2D Images. *PDPTA' 2000 Conference, Las Vegas, USA.*, 2000.

[9] M. Sarfraz and M. A. Khan. Towards Automation of Capturing Outlines of Arabic Fonts. *Proceeding of WICS 2000 Workshop*, 2000.

[10] M. Sarfraz and M. A. Khan. Automatic Outline Capture of Arabic Fonts. *Journal of Information Sciences*, 140(3-4), 2001.

[11] Foley, Van Dam, Feiner, and Hughes. *Computer Graphics: Principles and Practice.* Addison Wesley Publishing Company, 1999.

[12] Koichi Itoh and Yoshio Ohno. A curve fitting algorithm for character fonts. *Electronic Publishing*, 6(3):195–198, September 1993.

[13] H. C. Liu and M.D. Srinath. Corner detection from chain-code. *Pattern Recognition*, pages 51–68, 1990.

[14] H.L. Beus and S.S.H. Tiu. An improved Corner Detection Algorithm based on chain coded plane curves. *Pattern Recognition*, 20(3):291–296, 1987.

[15] Dmitry Chetverikov and Zsolt Szabo. A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves. *Proc. 23rd Workshop of the Australian Pattern Recognition Group*, pages 175–184, 1999.

[16] Foley, Van Dam, Feiner, Hughes, and Phillips. *Introduction to Computer Graphics*. Addison Wesley Publishing Company, 1997.

[17] Christian H. Reinsch. Smoothing by spline functions. *Numerical Mathematik*, pages 177–183, 1967.

[18] M. Grossman. Parametric curve fitting. *The Computer Journal*, 14(2):169–172, 1970.

[19] M. Sarfraz. Designing of 3D Rectangular Objects. *Lecture Notes in Computer Science 1024: Image Analysis Applications and Computer Graphics*, pages 411–418, 1995. Springer-Verlag.

[20] M. Sarfraz. A Mathematical Model for Computer Graphics. *J. Sc. Res.*, 25(1 & 2), 1996.

[21] M. Sarfraz. C^2 Rational B-spline Surfaces with Tension Control. *New advances in CAD & Computer Graphics*, I:314–320, 1993.

[22] M. Sarfraz. Some Remarks on a Rational Cubic Spline for the Visualization of Monotonic Data. *Comuter and Graphics*, 26(1), 2002.

[23] M. Sarfraz, S. Butt, and M. Z. Hussain. Visualization of Shaped Data by a Rational Cubic Spline Interpolation. *Comuter and Graphics*, 25(5):833–845, 2001.

[24] M. Sarfraz and Z. Habib. Rational Cubic and Conic Representation: A Practical Approach. *IIUM Engineering Journal*, 1(2):7–15, 2000.

[25] M. Sarfraz. A Rational Cubic Spline for the Visualization of Monotonic Data. *Comuter and Graphics*, 24(2):509–516, 2000.

[26] R. Qu and M. Sarfraz. Efficient Method for Curve Interpolation with Monotonicity Preservation and Shape Control, Neural, Parallel & Scientific Computations. *Special Issue on Computer Aided Geometric Design*, 5(1-2):275–288, 1997.

[27] M. Sarfraz, M. Al-Mulhem, and F. Ashraf. Preserving Monotonic Shape of the Data using Piecewise Rational Cubic Functions. *Computers Graphics*, 21(1):5–14, 1997.

[28] M. Sarfraz. Curves and surfaces for CAD using C2 rational cubic splines. *Engineering with Computers*, 11(2):94–102, 1995.

[29] M. Sarfraz. Freeform Rational Bicubic Spline Surfaces with Tension Control. *FACTA UNIVERSITATIS, Ser. Mathematics and Informatics*, 9:83–93, 1994.

[30] M. Sarfraz. Rational Interpolation with Shape Control. *Journal of Transfigural Mathematics*, 1(1):19–28, 1994.

[31] M. Sarfraz. Generalized Geometric Interpolation for Rational Cubic Splines. *Computers and Graphics*, 18(1):61–72, 1994.

[32] J. A. Gregory, M. Sarfraz, and P. K. Yuen. Interactive Curve Design using C^2 Rational Splines. *Computers and Graphics*, 18(2):153–159, 1994.

[33] M. Sarfraz. A C^2 Rational Cubic Spline which has Linear Denominator and Shape Control. *Annales Univ. Sci. Budapest*, 37:53–62, 1994.

[34] M. Sarfraz. Rational Interpolation Preserving the Monotonic Shape of the Data. *FACTA UNIVERSITATIS, Ser. Mathematics and Informatics*, 8:87–95, 1993.

[35] M. Sarfraz. Shape Preserving Rational Cubic Interpolation. *Extracta Mathematicae*, 8(2-3):106–111, 1993.

[36] M. Sarfraz. Designing of Curves and Surfaces using Rational Cubics. *Computers and Graphics*, 17(5):529–538, 1993.

[37] M. Sarfraz. A Geometric Rational Spline with Tension Controls: an alternative to the Weighted Nu-spline. *The Punjab University Journal of Mathematics*, 26:27–40, 1993.

[38] M. Sarfraz. A Geometric Characterization of Parametric Rational Quadratic Curves. *The Punjab University Journal of Mathematics*, 26:41–48, 1993.

[39] M. Sarfraz. Interpolatory Rational Cubic Spline with Biased, Point and Interval Tension. *Computers and Graphics*, 16(4):427–430, 1992.

[40] M. Sarfraz. A C^2 Rational Cubic alternative to the NURBS. *Computers and Graphics*, 16(1):69–77, 1992.

[41] M. Sarfraz. Convexity Preserving Piecewise Rational Interpolation for Planar Curves. *Bull. Korean Math. Soc.*, 29(2):193–200, 1992.

[42] M. Sarfraz. Interpolatory Rational Bicubic Spline Surface with Shape Control. *J. Sc. Res.*, 20(1 & 2):43–64, 1991.

[43] J. A. Gregory and M. Sarfraz. A Rational Spline with Tension. *Computer Aided Geometric Design*, 7:1–13, 1990.

[44] M. Sarfraz and Z. Habib. A rational cubic spline for the visualization of convex data. *The Proceedings of IEEE International Conference on Information Visualization-IV 2001-UK, IEEE Computer Society Press, USA*, pages 744–748, 2001.

[45] M. Sarfraz. Visualization of Monotone Data with Cubic Splines. *The Proc. International Conference on Imaging Science, Systems, and Technology (CISST 2001), Las Vegas, Nevada, USA, CSREA Press, USA*, pages 764–770, 2001.

[46] M. Sarfraz. Capturing Outlines of Hand-Drawn Shapes using Recursive Subdivision. *The Proceedings of the International Conference on Computer Graphics and Imaging, Las Vegas, Nevada, USA, ACTA Press, USA*, pages 204–209, 2000.

[47] M. Sarfraz, S. Butt, and M. Z. Hussain. A Rational Spline for Visualizing Positive Data. *The Proceedings of IEEE International Conference on Information Visualization-IV 2000-UK, IEEE Computer Society Press, USA*, pages 57–62, 2000.

[48] M. Sarfraz and A. Raheem. Curve Designing using a Rational Cubic Spline with Point and Interval Shape Control. *The Proceedings of IEEE International Conference on Information Visualization-IV 2000-UK, IEEE Computer Society Press, USA*, pages 63–68, 2000.

[49] M. Sarfraz. Piecewise Rational Interpolation Preserving Positive Data. *The Proc. International Conference on Imaging Science, Systems, and Technology (CISST 2000), Las Vegas, Nevada, USA, CSREA Press, USA*, pages 103–109, 2000.

[50] M. Sarfraz. A Smooth Rational Spline for Visualizing Monotone Data. *The Proceedings of IEEE International Conference on Information Visualization-IV'99-UK, IEEE Computer Society Press, USA*, pages 372–377, 1999.

[51] M. Sarfraz and Z. Habib. Conic Representation of a Rational Cubic Spline. *The Proceedings of IEEE International Conference on Information Visualization-IV 1999-UK, IEEE Computer Society Press, USA*, pages 232–237, 1999.

[52] M. Sarfraz. Designing of Objects using rational Quadratic Spline with Interval Shape Control. *Proc. International Conference on Imaging Science, Systems, and Technology (CISST 99), Las Vegas, Nevada, USA, CSREA Press, USA*, pages 558–564, 1999.

[53] A. Raheem and M. Sarfraz. Designing using a Rational Cubic Spline with Point Shape Control. *Proc. International Conference on Imaging Science, Systems, and Technology (CISST 99), Las Vegas, Nevadu, USA, CSREA Press, USA*, pages 565–571, 1999.

[54] M. Sarfraz, Z. Habib, and M. Hussain. Piecewise Interpolation for Designing of Parameteric Curves. *The Proceedings of IEEE International Conference on Information Visualization-IV 98-UK, IEEE Computer Society Press, USA*, pages 307–313, 1998.

[55] M. Sarfraz, M. Al-Mulhem, J. Al-Ghamdi, and A. Hussain. Quadratic Representation to a C1 Rational Cubic Spline with Interval Shape Control. *Proc International Conference on Imaging Science, Systems, and Technology (CISST'98), Las Vegas, Nevada, USA, CSREA Press, USA*, pages 322–329, 1998.

[56] M. Sarfraz, M. Al-Mulhem, J. Al-Ghamdi, and A. Raheem. Modeling by a Rational Spline with Interval Shape Control. *Proc Computer Graphics International '98, Hanover (Germany), IEEE Computer Society Press*, pages 730–737, 1998.

[57] M. Sarfraz, M. Z. Hussain, and S. Butt. Nonnegative Rational Spline Interpolation. *The Proceedings of International Conference on Information Visualization-IV'97-UK, IEEE Computer Society Press, USA*, pages 200–204, 1997.

[58] M. Sarfraz, M. Hussain, and Z. Habib. Local Convexity Preserving Rational Cubic Spline Curves. *The Proceedings of International Conference on Information Visualization-IV 97, UK, IEEE Computer Society Press, USA*, pages 211–218, 1997.

[59] M. Sarfraz. Rational Spline Interpolation Preserving the Shape of the Monotonic Data. *The Proceedings of Computer Graphics International'97-Belgium, IEEE Computer Society Press, USA*, pages 238–244, 1997.

[60] M. Sarfraz, M. Z. Hussain, and S. Butt. Simulating a Model for the Scientific Data. *The Proceedings of International Conference on Operations and Quantitative Management, India*, 1997.

[61] M. Sarfraz. An alternative to the NURBS of degree three. *The Proceedings of ISCIS XI: The Eleventh International Symposium on Computer and information Sciences, Turkey*, 2:735–744, 1996.

[62] M. Sarfraz, F. Ashraf, and M. Al-Mulhem. Preserving Shape of Scientific Data with Monotonic Behaviour: A Curve Method and Algorithm. *The Proceedings of the CADEX'96: The International Conference and Exhibition on Computer Aided Design, Austria*, 1996.

[63] M. Sarfraz and F. Hussain. Preserving the Monotonic Shape of the Scientific Data using Rational Cubics. *The Proceedings of IASTED International Conference on Modelling and Simulation, USA*, pages 169–173, 1996.

[64] M. Sarfraz and F. Hussain. A Rational Quadratic B-Spline Method with Shape Control. *The Proceedings of ISCIS X: The Tenth International Symposium on Computer and information Sciences, Turkey*, 2:573–58C, 1995.

[65] M. Sarfraz. Designing of Curves with Shape Control using Rational Cubic Splines. *The Proceedings of 4th International Conference on CAD and Graphics, SPIE*, 2644:20–27, 1995.

[66] M. Sarfraz. Efficiently Visualizing the Scientific Data Preserving the Monotonicity. *The Proceedings of 4th International Conference on CAD and Computer Graphics, SPIE*, 2644:757–764, 1995.

[67] M. Sarfraz, S. Butt, and H. Zawwar. Interpolation for the Positive Data using Rational Cubics. *The Proceedings of the Fourth International Conference of ISOSS, Pakistan*, 8:251-261, 1994.

[68] M. Sarfraz. Geometric Continuity does work for Computer Aided Geometric Design. *The Proceedings of the First National Symposium on artificial Intelligence, Pakistan*, pages 63–70, 1991.

[69] Sheng-Chuan Wu, John F. Abel, and Donald P. Greenburg. An interactive computer graphics approach to surface representation. *Communications of ACM*, 20(10):703–712, October 1977.

[70] Fujio Yamaguchi. A new curve fitting method using a crt computer display. *Computer Graphics and Image Processing*, pages 425–437, 1978.

[71] P. Dierchx. Algorithms for smoothing data with periodic and parametric splines. *Computer Vision, Graphics and Image Processing*, 1982.

[72] Mark Yang, Chong-Kyo Kim, Kuo-Young Cheng, Chung-Chin, and S.S. Liu. Automatic curve fitting with quadratic B-spline functions and its applications of computer-aided animation. *Computer Vision, Graphics and Image Processing*, pages 346–363, 1986.

[73] Christine Vercken, Christine Potier, and Sylvie Vignes. Spline curve fitting for an interactive design environment. *Theoretical Foundations of Computer Graphics and CAD*, 1987.

[74] M. Sarfraz and Arshad Raza. Visualization of Data using Genetic Algorithm. *Soft Computing in Industrial Applications - Recent Advances*, 2001. Springer Engineering, UK.

[75] M. Sarfraz and Arshad Raza. Visualization of Data using Genetic Algorithm. *The Proceedings of the Fourth KFUPM Workshop on Information and Computer Science: Internet Computing (WICS 2002)*, 2002.

[76] M. Sarfraz and Arshad Raza. Genetic Algorithm and Data Visualization. *The 6th Online World Conference on Soft Computing in Industrial Applications, WSC6: Multimedia and Internet Session*, 2001.

[77] M. Sarfraz and Arshad Raza. Capturing Outline of Fonts using Genetic Algorithm and Splines. *The Proceedings of IEEE International Conference on Information Visualization-IV 2001-UK, IEEE Computer Society Press, USA*, pages 738–743, 2001.

[78] Arshad Raza. Genetic Algorithms for Visualization. Master's thesis, KFUPM, 2001.

[79] Kozo Ichida, Takeshi Kiyono, and Fujiichi Yoshimoto. Curve fitting by one pass method with a piecewise cubic polynomial. *ACM Transactions on Mathematical Software*, 3(2):164–174, June 1977.

[80] Won L Chong. Automatic curve fitting using an adaptive local algorithm. *ACM Transactions on Mathematical Software*, 6(1):45–57, March 1980.

[81] M. Sarfraz and M. F. A. Razzak. An Algorithm for Automatic Capturing of Font Outlines. *Comuter and Graphics*, 26, 2002.

[82] M. Sarfraz and M. F. A. Razzak. Automatic Font Outline Capturing Through WEB. *The Proceedings of the Fourth KFUPM Workshop on Information and Computer Science: Internet Computing (WICS 2002)*, 2002.

[83] Bezier P. *Emploi des Machines a Commande Numerique/Numerical Control - Mathematics and Applications*. Wiley, 1972.

[84] Bezier P. *Mathematical and Practical Possibilities of UNISURF in Barnhill, R.E., and R.F. Riesenfeld, eds. Computer Aided Geometric Design*. Academic Press, New York, 1974.

[85] Fredrik Espinoza. A World Wide Web Based Presentation System For An Adaptive Help System. Master's thesis, Uppsala University, November 1996.

[86] Mattias Moser. Web Based Training Systems and Document Annotation Implementations for Hyperwave. Master's thesis, University of Auckland, New Zealand, November 1998.

[87] Gillian D. Elcock. Web-Based User Interface for a Simple Distributed Security Infrastructure (SDSI). Master's thesis, Massachusetts Institute of Technology, June 1997.

[88] James Rice, Adam Farquhar, Philippe Piernot, and Thomas Gruber. Lessons Learned Using the Web as an Application Interface. *CHI'96*, September 1995.

[89] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. HTML 4.0 Specification. http://www.w3.org/TR/REC-html40, 1998.

[90] Netcrafts Web survey. http://www.netcraft.com/survey, October 1998.

[91] Miroslav Trajkovic and Mark Hedley. Fast corner detection. *Image and Vision Computing*, 16(2):75–87, February 1998.

[92] G. Avrahami and V. Pratt. Sub-pixel edge detection in character digitization. *Raster Imaging and Digital Typography II*, pages 54–64, 1991.

[93] Azhar Quddus. *Curvature Analysis Using Multiresolution Techniques*. PhD thesis, K.F.U.P.M, 1998.

[94] G. E. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, New York, 1994.

[95] J.M. Brun, S. Fouofu, A. Bouras, and P. Arthaud. In search of an optimal parameterization of curves. *COMPUGRAPHICS95*, pages 135–142, December 1995.

[96] World Wide Web Consortium. http://www.w3.org/.

[97] MATLAB Web Server. The Language of Technical Computing. *The MathWorks, Inc.*, September 2000. 3 Apple Hill Drive Natick, MA 01760-2098, http://www.mathworks.com/access/helpdesk/help/pdf_doc/webserver/webserver.pdf.

Electronic references

Berners-Lee, T., WorldWideWeb: Proposal for a HyperText Project,

http://www.w3.org/pub/WWW/Proposal.html, Nov1990.

The Common Gateway Interface,

http://hoohoo.ncsa.uiuc.edu/cgi/,7/4 1996.

HyperText Markup Language,

http://www.w3.org/pub/WWW/MarkUp/, 20/10 1996.

HyperText Transfer Protocol,

http://www.w3.org/pub/WWW/Protocols/, 20/11 1996.

Java, by Sun Microsystems,

http://java.sun.com, 10/10 1996.

SGML, http://www.w3.org/pub/WWW/MarkUp/SGML/, 19/12 1996.

W3C - The World Wide Web Consortium,

http://www.w3.org/pub/WWW/, 15/7 1996.

Vita

- Muhammad Faisal Abdul Razzak.
- Born in Karachi, Pakistan on February 22, 1974.
- Received Bachelor of Engineering (B.E) degree in Computer Systems from N.E.D University of Engineering and Technology, Karachi, Pakistan in 1998.
- Joined King Fahd University of Petroleum and Minerals in August 1999.
- Currently two Publications.
- Email: faisalar@ccse.kfupm.edu.sa