The Islamic University–Gaza
Research and Postgraduate Affairs
Faculty of Information Technology
Master of Information Technology

الجامعـــــــة الإســــلاميــة غـــزة
شئون البحث العلمي والدراسات العليا
كـليــــة تكنـــولـــوجيا المعلـــومات
ماجستير تكنولوجيا المعلومات

Efficient Arabic Word Outline by Combining Active Contour Model and Corner Detection

تمثيل الإطار الخارجي للكلمات العربية بكفاءة من خلال الدمج بين نموذج الكنتور النشط وتحديد ونقاط الزوايا

Amjad Yousef El-Bahnasawi

Supervised by

Dr. Ashraf Alattar

Assistant Professor of Computer Science

A thesis submitted in partial fulfillment

Of the requirements for the degree of

Master of Information Technology

March/2017

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Efficient Arabic Word Outline by Combining Active Contour Model and Corner Detection

تمثيل الإطار الخارجي للكلمات العربية بكفاءة من خلال الدمج بين نموذج الكنتور النشط وتحديد ونقاط الزوايا

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل الاخرين لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

Declaration

I understand the nature of plagiarism, and I am aware of the University's policy on this.

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted by others elsewhere for any other degree or qualification.

Student's name:	أمجد يوسف البهنساوي	**اسم الطالب:**
Signature:		**التوقيع:**
Date:	16/2/2017	**التاريخ:**

Abstract

Graphical curves and surfaces fitting are hot areas of research studies and application, such as artistic applications, analysis applications and encoding purposes. Outline capture of digital word images is important in most of the desktop publishing systems. The shapes of the characters are stored in the computer memory in terms of their outlines, and the outlines are expressed as Bezier curves. Existing methods for Arabic font outline description suffer from low fitting accuracy and efficiency.

In our research, we developed a new method for outlining shapes using Bezier curves with minimal set of curve points. A distinguishing characteristic of our method is that it combines the active contour method (snake) with corner detection to achieve an initial set of points that is as close to the shape's boundaries as possible. The method links these points (snake + corner) into a compound Bezier curve, and iteratively improves the fitting of the curve over the actual boundaries of the shape.

We implemented and tested our method using MATLAB. Test cases included various levels of shape complexity varying from simple, moderate, and high complexity depending on factors, such as: boundary concavities, number of corners.

Results show that our method achieved average 86% of accuracy when measured relative to true shape boundary. When compared to other similar methods (Masood & Sarfraz, 2009; Sarfraz & Khan, 2002; Ferdous A Sohel, Karmakar, Dooley, & Bennamoun, 2010), our method performed comparatively well.

Keywords: Bezier curves, shape descriptor, curvature, corner points, control points, Active Contour Model.

الملخص

تعتبر المنحنيات والأسطح الرسومية موضوعاً هاماً في الدراسات البحثية وفي التطبيقات البرمجية مثل التطبيقات الفنية، وتطبيقات تحليل وترميز البيانات. ويعتبر تخطيط الحدود الخارجية للكلمات عملية أساسية في غالبية تطبيقات النشر المكتبي. في هذه التطبيقات تخزن أشكال الأحرف في الذاكرة من حيث خطوطها الخارجية، وتمثل الخطوط الخارجية على هيئة منحنيات Bezier. الطرق المستخدمة حالياً لتحديد الخطوط الخارجية للكلمات العربية تنقصها دقة وكفاءة الملاءمة ما بين الحدود الحقيقية والمنحنى الرسومي الذي تقوم بتشكيله.

في هذا البحث قمنا بتطوير طريقة جديدة لتخطيط الحدود الخارجية للكلمات تعتمد على منحنيات Bezier بمجموعة أقل من المنحنيات الجزئية. تتميز طريقتنا بخاصية مميزة وهي الدمج بين آلية لاستشعار الزوايا مع آلية نموذج الكنتور النشط (الأفعى). يتم الدمج بين نقاط الزوايا ونقاط الأفعى لتشكيل مجموعة موحدة من النقاط المبدئية قريبة قدر الإمكان من الحدود الحقيقية للشكل المراد تحديده. يتشكل منحنى Bezier من هذه المجموعة المدمجة، وتتم عملية تدريجية على دورات لملاءمة المنحنى على الحدود الحقيقية للشكل.

قام الباحث بتنفيذ وتجربة الطريقة الجديدة باستخدام برنامج MATLAB. وتم اختيار أشكال رسومية كعينات اختبار تتصف بمستويات متباينة من التعقيد تتراوح ما بين بسيط إلى متوسط إلى عالي التعقيد على أساس عوامل مثل تقعرات الحدود، عدد نقاط الزوايا، الفتحات الداخلية، إلخ.

وقد أظهرت نتائج الاختبار أن طريقتنا الجديدة حققت دقة في الملائمة تصل نسبتها إلى ٨٦% مقارنة بالحدود الحقيقية للشكل المستهدف. وكذلك فقد كان أداء طريقتنا جيداً بالمقارنة مع طرق أخرى مماثلة.

.

Dedication

To the best jewels which their names have decorated my life: My beloved mother, to my dear father,

To my dear wife and beloved children,

To my sisters and brothers,

To my professors and mentor at the Islamic University-Gaza,

To my friends and colleagues,

To all those who inspired and encouraged me in my study endeavor,

I dedicate this work for what you have done to assist me achieve this goal.

For that, I am evermore grateful.

Acknowledgment

I would like to extend thanks to Dr. Ashraf Al-attar for his relentless efforts and vital contribution to accomplish this thesis and I would like to express my sincere graduate for his outstanding supervision Dr. Ashraf has held me in various means at all of our work phases. I her by a knowledge that I would never stand on this place without their assistance and valuable advice and comments.

Similar, profound graduated goes to Dr. Ashraf Y. Maghari and Dr. Samy Salama for their role in discussing this thesis and putting down their valuable comments to excel this job.

.

Table of contents

List of Tables

List of Figures

List of Abbreviations:

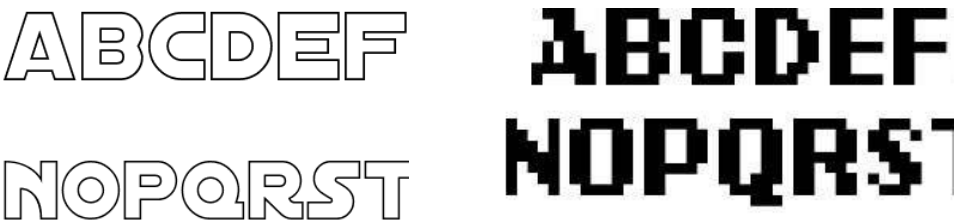
ACM	Active Contour Model
GAC	Geometric Active Contour
GVF	Gradient Vector Flow snake
PCACSS	Principle Component Analysis Based Curvature Scale Space
CP	Control Points
BCGSC	Bezier Curve Generic Shape Encoder

Chapter 1

Introduction

Chapter 1
Introduction

A computer font is an electronic data file containing an arrangement of glyphs, characters, or images, for example, dingbats. Although the term font first referred to a set of metal type sorts in one style and size. There are two major techniques for storing away fonts in computer. One is bitmap and other is outline. Outline is formed as a set of curves and vectors (TrueType Font described in is a typical member of the vector font family). We need little space to save and store fonts using vector approach and the obtained representation seem to have the ability to extend or shrink showing certain level of flexibility. Nevertheless, rasterization process is needed to switch- on adequate pixels, which demands additional processing power. Outline fonts use curves, drawing instructions and mathematical formulae to identify each glyph, which makes the character outlines scalable to any size (Haralambous, 2007). Show the outline fonts in this figure.



(a) (b)

Figure (1.1) : (a) Outline fonts, (b) Bitmap fonts

Bitmaps can be represented by a 2D array, where every pixel involves 1 bit of storage space. Since numerous pixels inside the array are turned-off, a great deal of memory space is squandered. This shortcoming is much clearer when larger-size font styles with many characters (like Chinese characters) are utilized. Another genuine shortcoming of raster font styles is that every text dimension must be saved in isolation, which consequently leads to some troubles when devices with various resolutions are used. The utilization of geometric alterations, such as scaling and turning are exceptionally restricted (Globacnika, Žalikb, Globačnik, & Žalik, 2010). Figure (1.1) (b) show the bitmap fonts.

Bitmap representation requires the transformation of font size to be more elaborated that is to match a particular printer.

Moreover, Latin and non- Latin fonts for example are unlike , the previous used outline fonts , while the latter used script-like nature as outlines , brush strokes or offset curves(Karow, 1994).

Various types of objects are mostly needed to be shown on the screen in computer graphics. The shape of objects are not always flat that's why we need to draw curves on objects limits or boundaries. A curve can simply be defined as a set of cohesive points, each point has a start point and an end point with the exemption of endpoints. Curves can be categorized into three groups explicit, implicit and parametric curves.

Parametric curves are called so for having parametric form. On contrary, explicit and implicit curves are used when the function is obvious. There are many types of parametric curves, such as Bezier Curves, B-Spline Curves and etc(Sofiyanti, Fitmawati, & Roza, 2015).

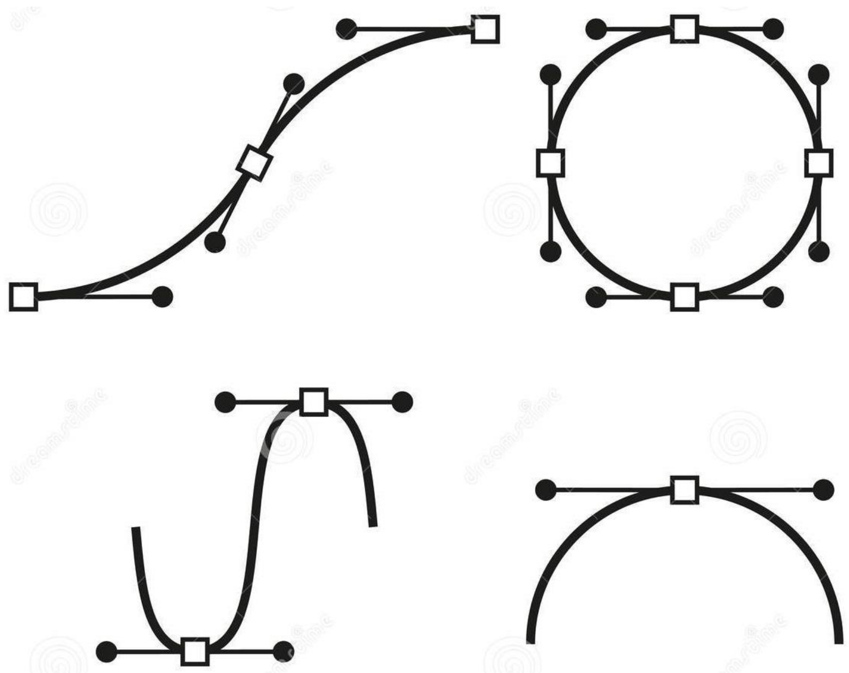


Figure (1.2) : Bezier Curve

It has been presented in previous literature diverse capturing techniques with different spline model, e.g. Bezier curve B-splines, Hermite interpolation and rational cubic interpolation. Some researcher suggested different ways that may reduce data points to approximately 5% of boundary points. Besides that, it has other merits and benefits like translation, rotation, scaling and deformation of shapes with certain quality standards. This might also drive us to new applications, such as: vector graphics, font designing, digitization of hand-drawn shapes, Data visualization, computer supported cartooning and different utilizations of Computer- Aided Design (CAD), Computer-Aided Manufacturing (CAM), and Computer-Aided Geometric Design (CAGD) (On, 2014).

Active contours, or snakes, are computer-generated curves that move within images to find object boundaries.

Computer vision and image analysis are frequently used to detect, locate and describe objects. A snake, for instance can be used to locate manufactured parts on an assembly line, one is to find the outline of the object and the other is to identify the character (Altarawneh et al. n.d. 2014). We use it to detect significant boundary points of Arabic characters.

a b c

Figure (1.3): (a)Initial Contour, (b) 50 iterations, (c) Final Contour

In this research, the Active Contours Model (ACM) combined with a corner detection technique would achieve the following advantages:

1 ACM Provides an initial set of spline points.

2 Corner detection identifies corner points.

3 Bezier Curve utilizes to construct an efficient spline, (adding/deleting points as necessary).

Snake is used to obtain significant points that help us arrive optimal fitting curve. Unlike other languages characters, Arabic language is unique because the Arabic characters are written cursively from right to left, and each character has two or more written forms depending on the character location in the word.

There should be more studies and concentration on Arabic script for its font formats richness and diversifications.

In this research, we developed a method for shape description of Arabic word outline, which would assist in the process of shape description. Moreover, we are going to adopt the Bezier Curve-based development processes for shape description of Arabic word outline, including Arabic language properties and its relationships. The outline will be the core of the approach to shape description of Arabic word outline by fitting curves. The developed method contains three basic modules namely the Pre-processing module, the Segment fitting and the Outline font generation module.

Different outline-capturing procedures that can be found in writing today. These strategies are computationally substantial and unsuitable for the use of analysis due to its utilization of a large number of end-points and lacks fitting accuracy and low level of efficiency. Moreover, there is limited number of researches related to Arabic language in this field.

Our research is intended to treat Arabic words, which differentiates it from the previous work. The research aims at developing a more comprehensive method by merging corner algorithm with Active Contour Modeling to achieve our ultimate goal with minimum end points.

Bezier curve and active contour model will be elaborated and discussed in Chapter 2 "Theoretical background".

1.1. Statement of the problem

Based on the above introduction and discussion we define our research problem in the following statement:

> *Existing outline fitting methods do not have high fitting efficiency (minimum number of points for shape description) and accuracy (minimum difference between fitting curve and actual shape contour) especially in shapes with boundary concavities. Furthermore, not enough number of methods cover Arabic words and characters.*

1.2. Objectives

Based on the above problem statement, the objectives of this research can be defined as specified in the following subsection.

1.2.1. Main objective

The main objective of this research is to develop an automatic shape outlining method by combining the snake technique, corner detection, and point minimization to generate highly efficient and accurate Bezier curves.

1.2.2. Specific objectives

1. Designing a method for automatic outline capture of digital word images.
2. Merging two algorithms (Active contour model & Corner detection) to outline capture of Arabic word.
3. Developing the method where the outline will be the focus.
4. Applying the method on Arabic character image to obtain the original outline of word.
5. Evaluating the results for accuracy in process of outlining.

1.3. Scope

1. There are many different outline capture fonts, but because of the importance of the capture Arabic fonts, we are going to concentrate on outline capture of Arabic fonts.
2. There are different fit curves to identify segment of contour points. However, we use cubic and linear Bezier curve.
3. We use two algorithms to determine corner points and find significant points which play an important role in the overall shape of the final word.
4. We use noise-free-images as objects of the research.

1.4. Limitations

1. Only solid shapes with no openings inside are considered.

2. Time: the study has neglected the impact of time.

2.1. Importance

Graphical curves and surfaces are hot areas of research studies and application (gaming, graphics animation, fonts and computer graphics), that is to achieve effeteness in identifying the significant corner points and maintain efficiency in fitness.

Font outlines have two general benefits:

1. Accuracy: outlines are more accurate in determining the optimal shape boundaries.
2. Storage size reduction.

In addition to the general benefits, font outlining is important for a wide range of specific practical purposes, such as:

1. Artistic applications: (Photoshop, Picasa, Perfect Effect and etc.)
2. Analysis application: (Image-Pro Premier, SigmaScan and etc.) Converting fonts from bitmap to outline (Vector) due to the reduction of size makes it easier for the user to control the words.
3. Font application: (FontAgent Pro, Font Pilot, Fontcase and etc.)
4. Encoding purposes: "Efficient encoding of the character outline is one of the prime challenges in mobile multimedia communications due to the inherent bandwidth limitations"(Ferdous Ahmed Sohel, Karmakar, & Dooley, 2007), which act on decreasing the requirements of storage space and transmission bandwidth.
5. Compression purposes: Decrease the requirements of storage space and transmission bandwidth.
6. Image indexing and retrieval.
7. Character recognition.
8. Conservation and Protection of historic and antique calligraphic characters.

2.2. Research Methodology

As a research project, our work involved a process of research activities performed by the researcher. We refer to this process as the “Research Methodology” and describe it here in this section in full detail. This is different from the “Method” which is the automated process proposed by the researcher as a solution to the research problem, and which is presented in great details in Chapter 4, “Method”.

The research project was organized into the following steps:

1. **The literature review**:

   At first, the researcher reviewed the academic literature available in local libraries and other internet resources including international database that is to identify and locate reliable resources about the Bezier Curve and the snake model.

2. **Problem statement**:

   Building upon the literature review, the researcher has made clear the problem statement which is confined to the problem of efficiency and accuracy in fitting especially in shapes boundaries and concavities. Existing outline fitting methods do not have high fitting efficiency and accuracy especially in shapes with boundary concavities.

3. **Designing the proposed method**: the researcher designed a new method by combining Active Contour Model and corner detection technique.
4. **Testing method**: The researcher used several measurements to check the efficiency and accuracy or the method, such as: standard deviation, Fitting ratio and Euclidian Distance using Matlab.
5. **Phrasing the solutions**.
6. **Presenting conclusion.**

2.3. Thesis Format

The remainder of this thesis is organized as follows: **Chapter 2** provides a theoretical background on Bezier curves, the active contour model, and corner detection techniques. **Chapter 3,** presents a number of research works related to our research. In **Chapter 4,** we present our research methodology including a detailed description and explanation of our proposed fitting method. In **Chapter 5,** we describe how we evaluated our proposed method, present and discuss the results. **Chapter 6**, conclusion of our research work and suggestions for future research followed by bibliography and appendixes.

Chapter 2

Theoretical Background

Chapter 2
Theoretical Background

2.1. Bezier curves:

The Bernstein polynomial basis has been used as the main function in Bezier model. It, Bernstein Polynomial foundation, demonstrates the performance of the model which also helps design, detect and draw certain curves of different arbitrative objects with different sizes depending on a group of control points.

Bezier curves of any degree can be defined as a degree N Bezier curve has n+1 control points whose merging functions are denoted $B_n^i(t)$. Figure (2.1) shows sample curves of degree one through four.

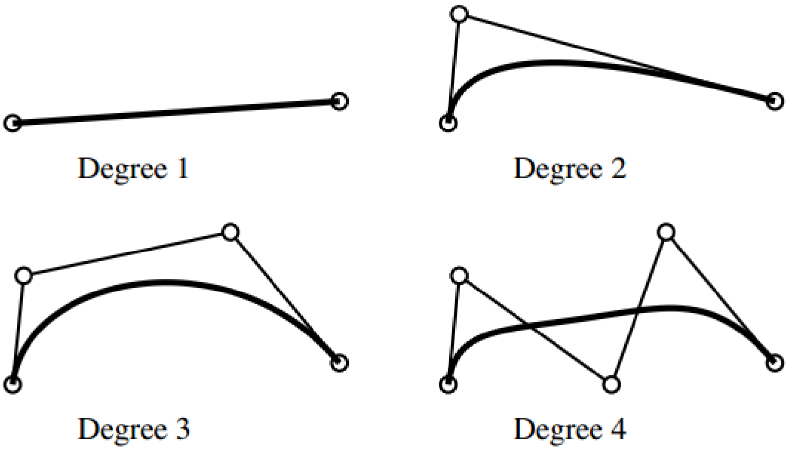


Figure (2.1): Bezier curves of various degree.

A Bezier curve is defined as a set of control points P_0 through P_n, where n is called its order (n = 1 for linear, 2 for quadratic, etc.). The first and last control points are always the end points of the curve; however, the intermediate control points (if any) generally do not lie on the curve. Linear Bezier curves gives points P_0 and P_1, A linear Bezier curve is simply a straight line between those two points. The curve can be stated as following:

$$B(t) = P_0 + t(P_1 - P_0) = (1-t)P_0 + tP_1 \; , 0 \le t \le 1 \qquad \text{Eq(1)}$$

And it is equivalent to linear interpolation. Cubic Bezier curves four points P0, P_1, P_2 and P_3 in the plane or in higher-dimensional space define a cubic Bezier curve. The curve starts at P_0 going toward P_1 and arrives at P_3 coming from the direction of P_2. Usually, it will not pass through P_1 or P_2; these points are only there to provide directional information. The distance between P0 and P1 determines "how far" and "how fast" the curve moves towards P1 before turning towards P2 (Khan & Lobiyal, 2017).

Writing $B_{P_i,P_j,P_k}(t)$ for the quadratic Bezier curve defined by points P_i, P_j, and P_k, the cubic Bezier curve can be defined as a linear combination of two quadratic Bezier curves:

$$B(t) = (1-t)B_{p0,p1,p2}(t) + tB_{p0,p1,p2}(t), 0 \leq t \leq 1. \qquad \text{Eq(2)}$$

P control points , $t \in [0 \quad 1]$

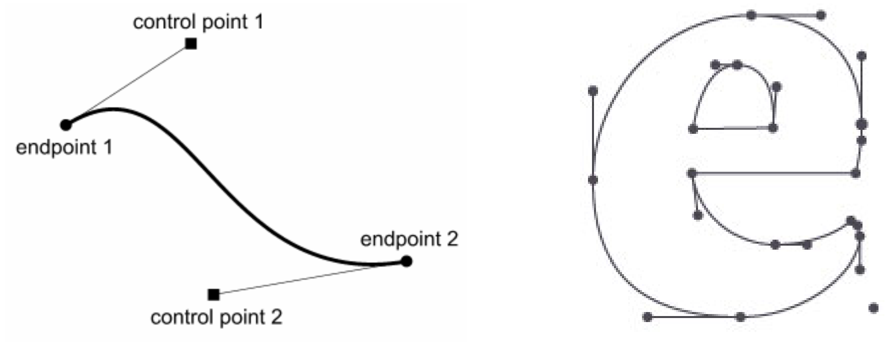
The explicit form of the curve is:

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^3 P_2 + t^3 P_3 \, , 0 \leq t \leq 1 \qquad \text{Eq(3)}$$

For some choices of P_1 and P_2 the curve may intersect itself, or contain a cusp.

Despite its wide use for free-form drawing, Bezier Curve can be used to shorten the distance and approximate the data points derived from various functions. It is applicable that any series of distinct points to be modified or altered to a cubic Bezier curve that matches the four points together in order. Some cubic Bezier curves function is to give the starting and ending points of some cubic Bezier curves, and the points along the curve comparing to t = 1/3 and t = 2/3, the control points for the first Bezier curve can be recuperated. The problem of approximation function is the estimation of control points from a data set. There is a difference between Drawing function and approximation function. In surface designs and curves the approximation error is not vital or vivid. However the aesthetics or visual effect of the shape is the main objective. Therefore, it should be noted how the drawn objects depicts the shape of the target object(On, 2014) .

Bernstein approximation , which is the line drawing between control points, of data points and goes to higher order(quadratic or cubic) approximation , until it mimics the shape of object. Via interactions, a designer can conduct corrections when required to perfect the shape of the object. On the other hand the, we account for the data approximation by the error in approximation. It's a mathematical problem as we are not concerned with the graphics but with the approximation. Furthermore, if the data set corresponds to a gray level image, the error in approximation becomes subjective (Lu & Xiang, 2016). The Figure (2.2) **(b)** shows Bezier curve fitting fonts.



(a) (b)

Figure (2.2) : Bezier Curve Fitting.

We accept small or large error depending on the nature of applications. Such approximation of image data points is useful in compression and feature extraction. The concept of control points in Bezier -Bernstein spline is implicit in the definition of the Bernstein polynomial and it was Bezier who made it explicit. Later on, the concept of control points was generalized to knots in B-spline to keep the interaction locally confined, so that the global shape of curves and surfaces is least affected. The generalization, therefore, introduces more drawing flexibility in the B-spline model(Sofiyanti et al., 2015). Problem when using Bezier curves some points are not necessary.

2.1.1 Properties of Bezier Curves

The Bezier Curves have distinct characteristics which can be summarized in the following points:

- Following the shape of control polygon which consists of segments joining the control points.
- Passing from the beginning, first corner point, to the last control point.
- Having the structure of convex hull in defining the control points.
- Having followed polynomial in defining the curve segment, the degree is one point less than the polygon point.
- Following the shape of polygon in defining the shape.

- Having the same direction of the last tangent vector as that vector determined by the first and last segments.
- Ensuring the smooth following of the control points.
- Avoiding intersection between the Bezier cure repeatedly in away exceeds the intersections in the control polygon.
- Maintaining steadiness under relative changes.
- Exhibiting alteration in the shape of the curve as a whole.
- Bezier curve can be subdivided at a point t=t0 into two Bezier segments which join together at the point corresponding to the parameter value t=t0(Sofiyanti et al., 2015).

2.2. Active Contours Models

The Active Contour Model (ACM), also called snakes, is a framework in computer vision for delineating an object outline from a possibly noisy 2D image. The ACM was first defined by Kass et al. as an energy minimizing, deformable spline influenced by constraints and image forces that pull it towards object contours. The snake model is popular in computer vision, and snakes are greatly used in applications like object tracking, shape recognition, segmentation, edge detection and stereo matching (Liljeback, Pettersen, Stavdahl, & Gravdahl, 2012).

Snakes may be understood as a special case of the general technique of matching a deformable model to an image by means of energy minimization(Kass, Witkin, & Terzopoulos, 1988).

In two dimensions, the active contour model is represented by a discrete version taking advantage of the point distribution model to restrict the shape range to an explicit domain learned from a training set. In the discrete formulation, the ACM model is composed of a set of snake points $v_i = (x_i, y_i)$ for $i = 1,2, \ldots, n$ where x_i and y_i are the x and y coordinates, respectively, and n is the total number of snake points. The points are initialized around the object of interest as shown in Figure (2.3).

Each snake point has an energy value determined by the effect of two types of forces: a) **external** forces based on image features such as edge gradient, and b) **Internal** forces based on snake features such as continuity and curvature. Overall, the

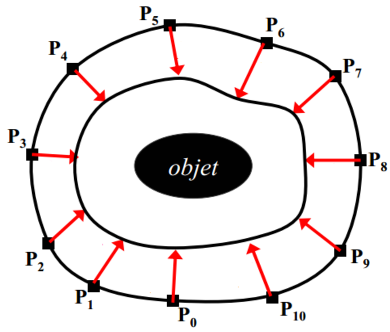


Figure (2.3): Active Contour Model (snake).

snake's total energy is the sum of its points' energies, and is typically formulated as follows:

$$E_{snake} = \sum_{i=1}^{n} (\alpha E_{cont}(v_i) + \beta E_{curv}(v_i) + \gamma E_{ext}(v_i)) \qquad \text{Eq. 1}$$

Whereα,β, and γ are relative weights, $E_{cont} = |\, v_s(s)|^2$, $E_{curv} = |\, v_{ss}(s)|^2$, and E_{ext} is typically edge gradient or any other user interaction. The choice of coefficient values in Eq.1, depends on what type of features to be extracted:

- Set α high if there is a deceptive Image Gradient.
- Set β high if smooth edged Feature, low if sharp edges.
- Set γ high if contrast between Background and Feature is low.

Through an optimization process the snake's energy is minimized and typically settles at the object's boundaries. The most commonly used algorithm for implementing the optimization process is the greedy technique which leads to the greedy snake.

2.2.1 Greedy Snake

The greedy snake is called greedy because it looks for the optimal solution locally at the snake point level. As Figure (2.4) shows, the movement of each snake point is determined by calculating the energy of candidate points in its $n \times n$ neighborhood in terms of Eq. 1, and selecting the minimum energy location. This procedure is performed for each snake point iteratively until the points stop moving to new locations (Saqer, 2015).

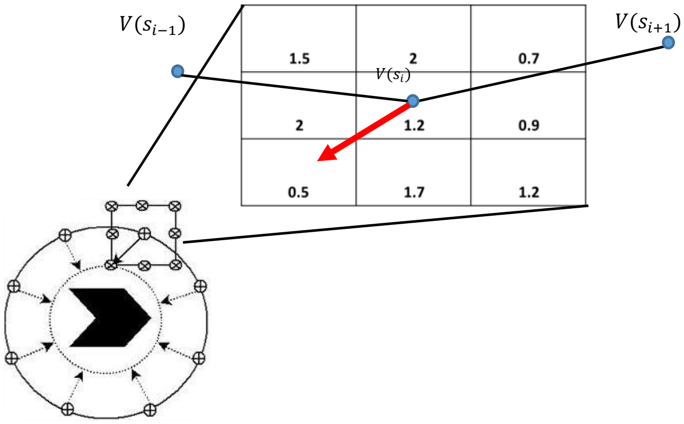


Figure (2.4): Greedy snake neighborhood energy minimization.

A major shortcoming of the classical snake is the inability of snake points to converge on concave boundaries of the shape. Figure (2.5) demonstrates this shortcoming with an active contour with points which failed to reach inside the concavities of the work.

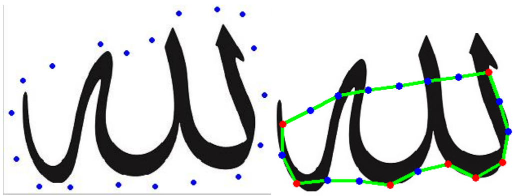


Figure (2.5): Snake points fail to reach concave boundaries of the shape

The following section describes one of the modified ACM models which overcomes the concavity problem.

2.2.2 Gradient Vector Flow (GVF) Model

Many approaches where proposed to overcome traditional ACM shortcomings. Gradient Vector Flow (GVF) was proposed by Chenyang Xu et al (2008).

GVF is computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image and used as an external force of snake and called a GVF snake.

The advantages of using GVF in snake model are:

- The snake becomes insensitive to the initial position of snake, if it's inside or outside object.

- The ability to move into boundary concavities.
- The large capture range, which means that, barring interference from other objects, it can be initialized far away from the boundary.

These advantages make the GVF snake more suitable for finding boundaries of buildings, and we find it the right choice to compare and evaluate our proposed method(Cheng & Liu, 2008).

Compared to classical feature extraction techniques, snakes have multiple advantages (K. Zhang, Song, & Zhang, 2010)**:**

- They autonomously and adaptively search for the minimum state.
- External image forces act upon the snake in an intuitive manner.
- Incorporating Gaussian smoothing in the image energy function introduces scale sensitivity.
- They can be used to track dynamic objects.
- The key drawbacks of the traditional snakes are
- They are sensitive to local minima states, which can be counteracted by simulated annealing techniques.
- Minute features are often ignored during energy minimization over the entire contour.
- Their accuracy depends on the convergence policy.

Snakes are a model-driven approach for solving many image understanding problems that are difficult- if not impossible- to tackle has using classical approaches. Just like human vision, snakes start with an a priori model of what an object should look like. By using the smoothness constraints of the splines, they are able to fill in for missing and noisy boundary information. As a result, they are more robust than non-model based methods, which make little use of image structure. Much of the current research in active contour models deals with generalizing the form of the contours and overcoming the convergence and stability problems encountered during the energy minimization process(Berger, N, Cede, & Mohr, 1990).

2.3. Corner Detection

Corner detection is an approach used within computer vision systems to extract certain kinds of features and infer the contents of an image. Corner detection is frequently used in motion detection, image registration, video tracking, image mosaicing, panorama stitching, and 3D modelling and object recognition. Corner detection overlaps with the topic of interest, which is point detection.

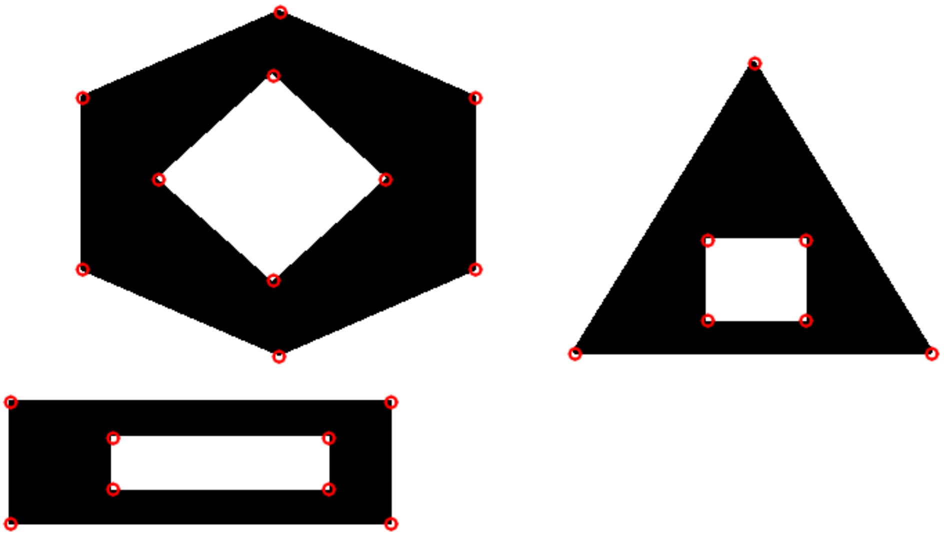


Figure (2.6): Corner Detection.

A corner can be characterized as the convergence of two edges. A corner can likewise be characterized as a point for which there are two overwhelming and diverse edge headings in a nearby neighborhood of the point.

An interest point is a point in an image which has a well-defined position and can be robustly detected. This means that an interest point can be a corner but it can also be, for example, an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal.

In practice, most so-called corner detection methods detect interest points in general, and in fact, the term "corner" and "interest point" are used more or less interchangeably through the literature(Willis, 2009). As a consequence, if only corners are to be detected it is necessary to do a local analysis of detected interest points to determine which of these are real corners. Examples of edge detection that can be used with post-processing to detect corners are the Kirsch operator and the Frei-Chen masking set.

Corner interest point and feature are used interchangeably in literature, confusing the issue. Specifically, there are several blob detectors that can be referred to as "interest point operators", but which are sometimes erroneously referred to as "corner detectors". Moreover, there exists a notion of ridge detection to capture the presence of elongated objects.

Corner detectors are not usually very robust and often require large redundancies introduced to prevent the effect of individual errors from dominating the recognition task. One determination of the quality of a corner detector is its ability to detect the same corner in multiple similar images, under conditions of different lighting, translation, rotation and other transforms.

A simple approach to corner detection in images is using correlation, but this gets very computationally expensive and suboptimal. An alternative approach used frequently is based on a method proposed by Harris and Stephens, which in turn is an improvement of a method by Moravec(Groch, 2011).

Extraction of corner points from digital image is an important research content in the field of digital image processing and pattern recognition. Corner point is pixel with a high curvature on the image edge, and it retains key local features of objects, therefore corner point has more information than any other pixel on image. Compared with other methods, image-processing algorithm based on corner point greatly reduces the computational complexity. At present, the basic starting point of most computer vision processing tasks, such as shape recognition, target detection, 3-D reconstruction, etc, is to extract corner point first. Consequently, corner detection algorithm has been a hot topic on the research of digital image processing.

The existing corner detection algorithms can be divided into two categories. One category is using grayscale image to process directly, such methods include Susan algorithm(Smith & Brady, 1997), Harris algorithm (Cg Harris, 1987) , etc. In addition, the main idea for this category of algorithms is using the gray scale correlation between one pixel and its neighborhood pixels, and then analyzing the area around the corner before achieving the final corner set. The other category is using the edge of an image, such methods include Cooper algorithm (Kitchen, 1993), Rosenfield algorithm (Teh & Chin, 1989) , etc. Moreover, the main idea for this category of algorithms is using the geometric topology of edge pixel to determine the position of the corner points after image edge detection. The first method mentioned above has some defects, such as high calculation complexity and poor real-time.

The Harris & Stephens corner detection algorithms

Harris and Stephens improved upon Moravec's corner detector by considering the differential of the corner score with respect to direction directly, instead of using shifted patches. (This corner score is often referred to as autocorrelation, since the term is used in the paper in which this detector is described. However, the mathematics in the paper clearly indicate that the sum of squared differences is used(Chris Harris & Stephens, 1988).

The Wang and Brady corner detection algorithm:

The Wang and Brady detector considers the image to be a surface, and looks for places where there is large curvature along an image edge. In other words, the algorithm looks for places where the edge changes direction rapidly. The corner score, C, is given by:

$$C = \nabla^2 I - c|\nabla I|^2 \quad \text{Eq(4)}$$

Where c determines how edge-phobic the detector is. The authors also note that smoothing (Gaussian is suggested) is required to reduce noise. In this case, the first term of C becomes the Laplacian (single-scale) blob detector. Smoothing also causes displacement of corners, so the authors derive an expression for the displacement of a 90 degree corner, and apply this as a correction factor to the detected corners(Wang & Brady, 1995).

Chapter 3

Literature review

Chapter 3
Literature review

3.1 Introduction

In this section, this work is going to review the previous attempts in handling the study field. The previous literature of the study help us in our endeavor to resolve the research problem and design our method.

At first, the project is going to discuss works of researchers related to corner point and Bezier curve and how these attempts helped us proceed and go on in this work.

Then, the active contour model is going to be put for discussion elaborately that is to draw lines and hints for our work and going to summarize how these works would contribute to the proposed methods.

3.2 Bezier Curve Outline

Edine and Elmiad (2014) proposed approach in the field of recognition of multi-font Arabic characters. Draws on the semi-cursive Arabic characters and consists in assimilating them to a small number of checkpoints equipped with their derivatives; it consists in choosing for primitives the dots where the shape of the character presents a significant variation (change of direction, inflection dots and cusps). These dots with their derivatives represent the primitives that characterize the character. The choice of their approach is motivated by the possibility to approximate the shape of the character using only these primitives. Indeed, the theory of Bezier curves from data on a few dots (coordinates and derivatives) to draw shapes that conserve the properties of data (On, 2014).

Karmakar and Bennamoun (2010) aimed in their the study to introduce a novel Bezier curve-based generic shape encoder (BCGSE) that partitions of an object contour turn into contiguous segments based upon its cornerity, before generating the control point for each segment using relevant shape curvature information. Although control points encoding has generally been ignored, BCGSE embeds an efficient vertex-based encoding strategy exploiting the latent equidistance between consecutive control points. A non-linear optimization technique has also been presented to enable the encoder to automatically adapt to bit-rate constraints. Their efficient encoding considerably aid in reducing the descriptor length (Ferdous A Sohel et al., 2010).

Masood and Sarfraz (2009) Proposed technique produces set of data points which are the control points of approximating Bezier curve. An efficient search algorithm producing optimal curves determines the control points. Approximation process is simplified by decomposition of outline into smaller curves. The decomposition/subdivision is performed on detected corner points as a preprocessing step. Further subdivision is done by recursive algorithm during the approximation process. Proposed algorithm has various advantages like computational efficiency, better shape representation, low approximation error and high compression ratio (Masood & Sarfraz, 2009).

Zhang and Lin (2007) the aim of the study was to introduce a novel methodology to capture character contours from images of ancient Chinese calligraphy which mainly includes two steps: feature points detecting from character contours and contour segment approximating. A new feature-point-detection method called PCACSS (Principle Component Analysis Based Curvature Scale Space) is proposed. Compared with several existing methods for feature-point-detection, it is robust to noise and can accurately detect feature points from character contours automatically. With feature points determined by PCACSS, the character contours are divided into some contour segments, each contour segment is then approximated with a straight line or Bezier curve depending on the least square error (J. Zhang, Yu, & Lin, 2007).

Yahya and Ali (2006) proposed the Arabic character is represented as an outline font fitted with G1 rational Bezier cubic curves. The outline font representation is done in several phases - contour extraction of font image, corner point's detection and segmentation and lastly Contour segment fitting. A rational cubic Bezier with weights is used in the last step. The weights are adjusted automatically to get curves that are as close as we want to the digitized data points. The weights of the rational cubics are adjusted to get the G0 and G1 curves to be as close as we want to the digitized data points. The whole process is folly automatic (Yahya, Md Ali, Majid, & Ibrahim, 2006).

Sarfraz and Khan (2002) presented a simple and effective method for optimal curve deign which is particularly useful for capturing outline of fonts and they alleged that the methodology that they adopted can be used for other applications. The corners are not detected precisely in the traditional approaches. In addition to the corner points, they identified some other significant points; these points are identified by introducing intermediate points as well as high curvature points. They presented a method for capturing outlines of fonts, particularly suitable for non-Roman languages like Arabic. A hand-drawn character is scanned from paper to obtain a gray level image. From this gray-level image, contour of the character is obtained. Then, corner points of the character are determined from contour. These corner points can be obtained by some interactive method or by some automated process. Optimal curve fitting is done by segmenting the contour outline at the corner points. The curve fitting methods are based on conics or Bezier cubics (Sarfraz & Khan, 2002).

Yang and Lee (2001) proposed a method to vectorize Chinese characters in calligraphy documents, the system can prevent the zigzag phenomena when the characters are enlarged. The system contains two modules: contour segment extraction and description. In the former, high curvature, points on contours are detected as corner points, which divide the contour into several segments. In the latter, a contour segment can be described either by a straight line or a cubic Bezier curve. According to relations between the contour segment and the Bezier curve, control points are adjusted to fit the contour segment better. When the curve fitness cost is small enough, the shape is described well (Yang, Lu, & Lee, 2001).

Sarfraz and Raza (2001) proposed a method to convert the original problem into a discrete combinatorial optimization problem and solve it by a genetic algorithm. They also incorporate a corner detection algorithm to detect significant points which are necessary to capture a pleasant looking spline fitting for shapes such as fonts. A parametric B-Spline has been approximated to various characters and symbols. The chromosomes have been constructed by considering the candidates of the locations of knots as genes. In order to aid They GA (Genetic Algorithm) a corner detection algorithm has also been used to determine the significant points, which are of great importance in capturing the outline of closed curves in which accuracy of approximation is important (Sarfraz & Raza, 2001).

Comments on the Bezier curve studies:

There are various outline-capturing techniques which can be found in the literature today. These methods (Masood & Sarfraz, 2009; Sarfraz & Khan, 2002; Ferdous A Sohel, Karmakar, Dooley, & Bennamoun, 2010) are computationally heavy and are not suitable because it used large number of end point. Some of the researchers have proposed techniques to minimize this computation time. (Yahya et al., 2006) Presented enhanced Bezier curve model, with no increase in order of computational complexity, to reduce the distance between Bezier curve and its control polygon. It was ultimately applied to reduce distortion of approximating curves. Similarly. (Sarfraz & Khan, 2002) in their outline capturing scheme, calculated the ratio between two intermediate control points and used this to estimate their position. This caused reduction of computation in subsequent phase of approximation. Few other techniques include use of control parameters, genetic algorithms(Sarfraz & Raza, 2001), and wavelets (Yang et al., 2001).

This is how our work is different in which we intend to develop a more comprehensive method by merging corner algorithm with Active Contour Modeling "Snake" to get mid-point with lowest cost, straight line between corner point used line Bezier curves.

We can conclude that the related works touch the idea discussed in terms of the use of a Bezier curve for shape description of Arabic word outline, but we are going focus on the Arabic word Outline. Additionally, their works lacked Arabic word outline that can help in shape description. We differ from them in the construction of Arabic word outline that includes merge corner detection and snake to obtain on optimal fitting shape.

3.3 Active contour Model Shape Detection:

Xu and Prince (1998), in their work, they proposed a new external force for active contours, largely solving both problems. This external force, which we call gradient vector flow (GVF), is computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image. It differs fundamentally from traditional snake external forces in that it cannot be written as the negative gradient of a potential function, and the corresponding snake is formulated directly from a force balance condition rather than a variational formulation (Xu & Prince, 1998).

Cheng and Liu, (2008), in their work, they proposed a new image segmentation algorithm based on Susan approach and GVF Snake model. First, we check corner points at the edge using Susan approach and mark them as energy minimization points, then, we use GVF Snake model to capture object boundary after setting initial snake curve (Cheng & Liu, 2008).

Discussing new image segmentation arithmetic based on SUSAN and GVF Snake. GVF snake model is employed widely in image segmentation and computer vision. It has a larger capture range and stronger convergence ability to boundary concavities than traditional snake. But in the energy minimization process some corner points can't be found. Because of this, the object boundary is not accurate to solve the problem.

Altarawneh and other (2014), in their work, they developed a global threshold-based active contour model. This model deploys a new edge-stopping function to control the direction of the evolution and to stop the evolving contour at weak or blurred edges. An implementation of the model requires the use of Selective Binary and Gaussian Filtering Regularized Level Set method (SBGFRLS). The method used either a selective local or global segmentation property. It penalizes the level set function to force it to become a binary function. This procedure is followed by using a regularization Gaussian. The Gaussian filters smooth the level set function and stabilizes the evolution process. One of the merits of this proposed model stems from the ability to initialize the contour anywhere inside the image to extract object boundaries. The proposed method is found to perform well, notably when the intensities inside and outside the object are homogenous. This method is applied on various types of images, including synthetic, medical and Arabic-characters images (Altarawneh, Luo, Regan, Sun, & Jia, 2014).

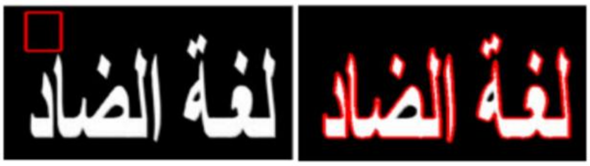


Figure (3.1): The left image shows the initial contour; the right image.

Figure (3.1) exhibits the performance of the proposed method in the case of Arabic-characters segmentation.

A global threshold-based active contour model with a new edge-stopping function has been presented. The main merits of this approach consist of its ability to control the direction of the evolving contour and to stop it on the weak or blurred edges. This model is implemented using the SBGFRLS method. They tested this method on several categories of images including synthetic, medical and Arabic-characters where a satisfactory performance was attained.

Comments on the Snake studies:

Most if not all of the previous studies indicted the significance point of using snake in outlining accurately the shapes of images.

There are some other studies that have been conducted on Arabic language in which merged the corner points detection techniques achieve a better images shaping identification and other (Altarawneh et al., 2014)).

To identify the curves properly, it is difficult to achieve this using the traditional snake model. Building upon that, we preferred to use snake model with corner points detection technique to enhance our capability in getting the point that describe the image by using Bezier curve.

3.4 Summary

The points of similarity and differences between our research and the related work:

Table (3.1): Similarity and differences between our research and the related work

Similarities	Differences
1. Employed a Bezier curve and Corner Point to Obtained outline. 2. Employed significant point to fit between the contour points and the Bezier Curve. 3. Applied their studies on printed written forms of languages. 4. Most of them applied the Bezier Curve on independent characters.	1. The current research is employing two types of Bezier curve (linear & cubic) 2. Employed Snake (Active Contour Model) to obtain significant point. 3. The study is going to apply the Bezier Curve on both printed and calligraphy. 4. The study is going to apply the Bezier Curve on words.

The researchers can summarize the benefit that can obtain from the related works in the following points:

1. Enriching the theoretical part of the Arabic word outline.
2. Identifying and understanding the obstacles and the problems that they have encountered during the Arabic word outline study progression and the ways to overcome them.
3. Finding appropriate ways to design the code in optimal results by combining between corner points and Active Contour Model.
4. Developing new ways to resolve problems and proposing recommendations for Arabic word outline.

Chapter 4

Methodology

Chapter 4
Proposed Method

4.1 Research method

This chapter describes in great details our developed method which is the outcome of Stage 4 “Design and Implementation” of our research project as outlined previously in Section 1.5 “Research Methodology”.

The basic idea of our method is as follows:

a) Use a corner detection technique and an active contour model, and combine the two sets of points produced by these two processes into one set.
b) Take each pair of adjacent points in the combined set as a Bezier curve and perform a curve fitting operation on the segment to the actual boundary of the shape.
c) Optimize the overall set of curve segments by merging adjacent segments.

We developed a method to minimize the human interaction in obtaining the outline of original word through merging two algorithms to identify corner points and significant points. Using Snake (Active Contour Model) and existing algorithms corner points helps us identify and determine the significant points in the shape which can easily enables us to put the appropriate fitting for the curve. The shape is divided into segments and treated in isolation from other segments to get the best fitting.

Figure (4.1) shows the developed method. The developed method contains three basic modules namely edge map, Contour and Corner detection (set of contour and edge points) and segment fitting (accurate and efficient outline).

1. Boundary Extraction.
2. Corner Detection (Using an existing corner detection technique and Snake point)
3. Segment Fitting: the corner points separate sets of data-points into segments.

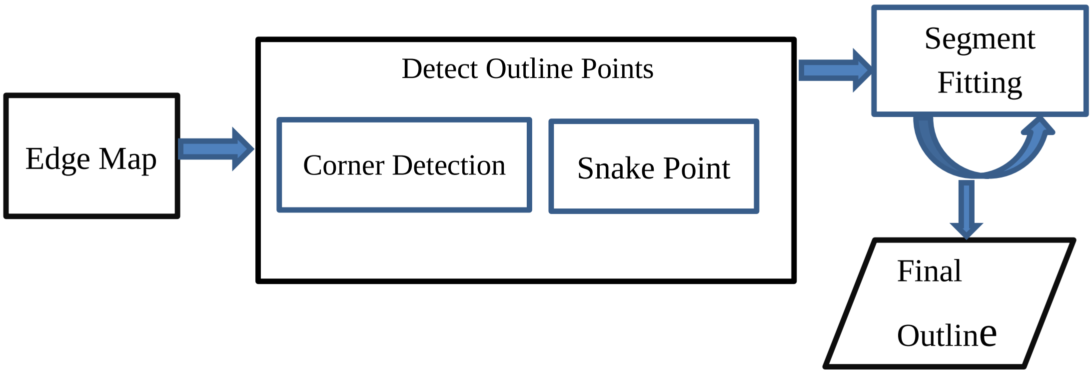


Figure (4.1): The developed method

To make the flow clear and complete, we illustrate the followed scenario for the consequent outline as shown in Figure (4.2).

4.2 Steps of the developed Method

The following steps summarize our method:

1. Edge detection using an existing algorithm such as canny edge detector.

2. Outline detection

- Using an existing corner detection technique such as Harris corner detector.
- Using Active Contour Model.

3. Contour segment fitting:

- Initially curve is fitted to only corner points.
- Fitting a Bezier curve for each segment by Euclidean distance. (Iterative process)
- Calculating fitness by least square method.
- Fitting up to desired tolerance limit.

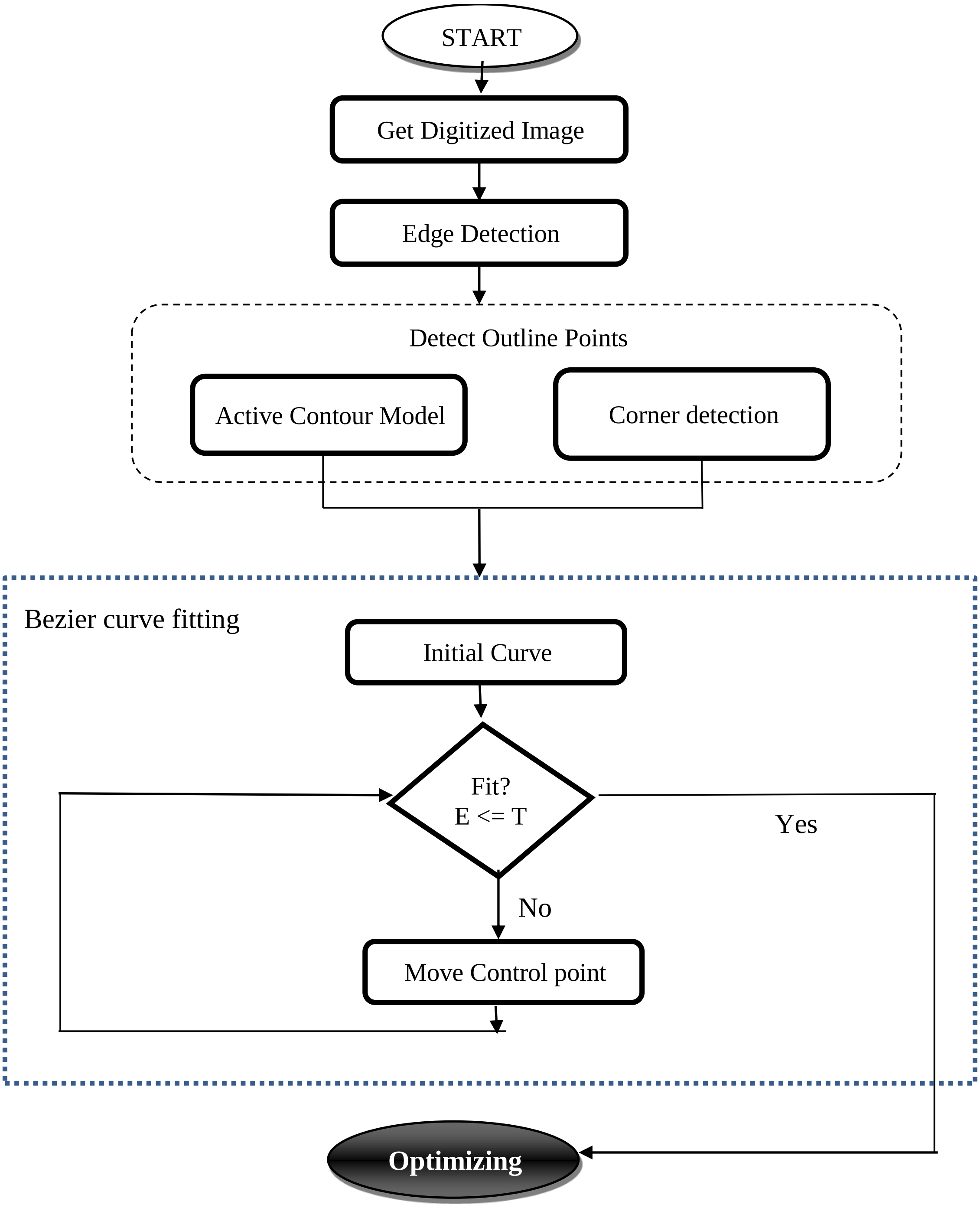


Figure (4.2): Stages of the Outline process based on the method.

E: error = $\sqrt{\left(X_{Contour\ point} - X_{Bezier\ curve}\right)^2 + \left(Y_{CContour\ point} - Y_{Bezier\ curve}\right)^2}$

T: Thresholds

4.2.1 Description of the fitting method

Fitting process between contour points and Bezier curve for each segment can be described as the following:

The fitting process performs the fitting iteratively on a segment-by-segment basis. Each segment is fitted in an iteration as follows:

- Edge detection by using the existing algorithm.
- Outline detection.
    a. By using the existing corner detection technique and Possibility determined number of corners.
    b. Using Active Contour Model to draw initial points.
    c. Merging corner points and snake points and arrange them in a clockwise order.
- Fitting a Bezier curve for each segment. (Iterative process)
    a. Each segment is defined by two points (start, end).
    b. While(Dis>=T)
        i. Create Bernstein formula using four control points p0=p1, p2=p3 straight line initially.
        ii. Set 3 point on Bezier curve (Left, Mid and Right) it shown in Figure (4.3)
        iii. Draw orthogonal line on **3** points (Left, Mid and Right).

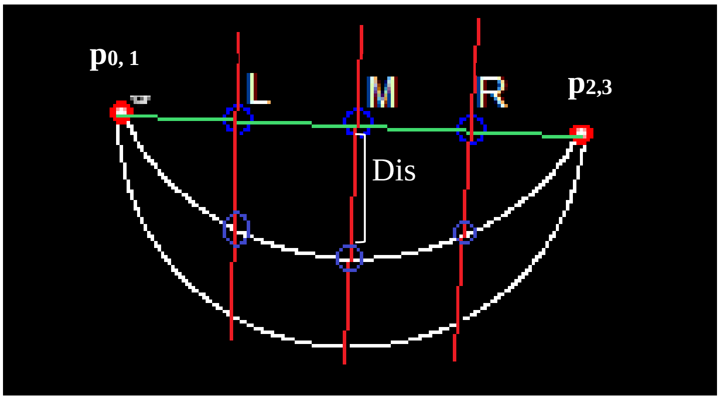


Figure (4.3): Fitting Process.

        iv. Find the nearest points on contour curve for each point (Left, Mid and Right) on extend orthogonal line.

- Find Yc By known slope " gradient" for orthogonal line and Compensation with X_c , c: *contour point*

$$Y_C = slope * (X_i - X_C) \qquad \text{Eq(5)}$$

v. Calculates the Euclidian distance the 3 points (Left, Mid and Right)

$$D(B,C) = \sqrt{(X_C - X_B)^2 + (Y_C - Y_B)^2} \qquad \text{Eq(6)}$$

B: *bezier point* , C: *contour point*

vi. Move Control point p1, p2 depending on the distances (D_{left}, D_{mid} and D_{right}).

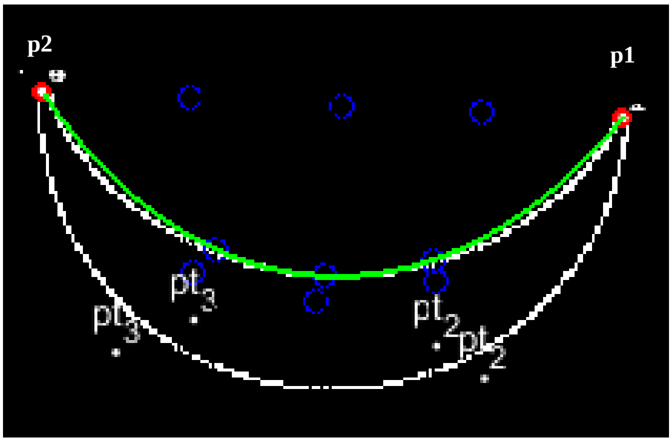


Figure (4.4): Fitting Process finished

- Move Right, Left

$$P_1(x) = P_1 \mp (P_1 \mp P_L) \div 2 \qquad \text{Eq(7)}$$
$$P_2(x) = P_2 \mp (P_2 \mp P_R) \div 2 \qquad \text{Eq(8)}$$

- Move Top, Down

$$P_1(y) = P_1 \mp Dis.L \div 2 \qquad \text{Eq(9)}$$
$$P_2(y) = P_2 \mp Dis.R \div 2 \qquad \text{Eq(10)}$$

vii. Calculate error =Number of Bezier curve points / Number of Contour Points

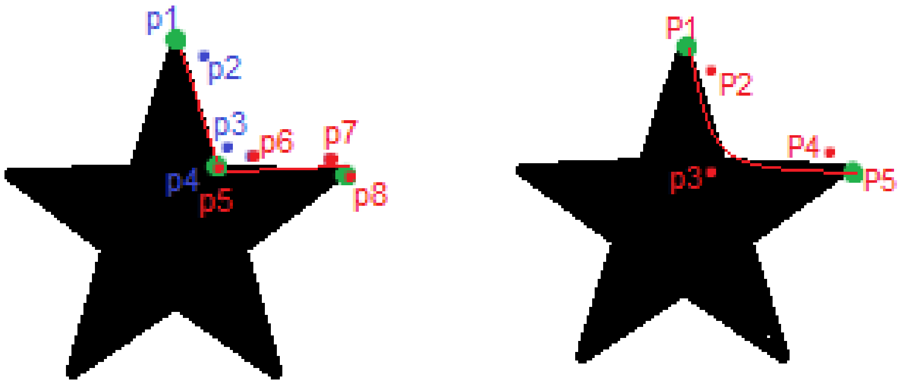
c. **End While**

- Optimizing curve :
  a. **For Each 2 curve sequence** shown in Figure (4.5)

Curve1: $P_{1,2,3,4}$ control points, **Curve2**: $P_{5,6,7,8}$. Use Bezier Curve of 4degree, and new curve consist of five control points $P_{\mathbf{n1}}$=p1, $P_{\mathbf{n2}}$=p2,

P_{n3}= (p3+p5) / 2, P_{n4}=p7, P_{n5}=p8. We use Eq(11) to calculate Bezier curve.

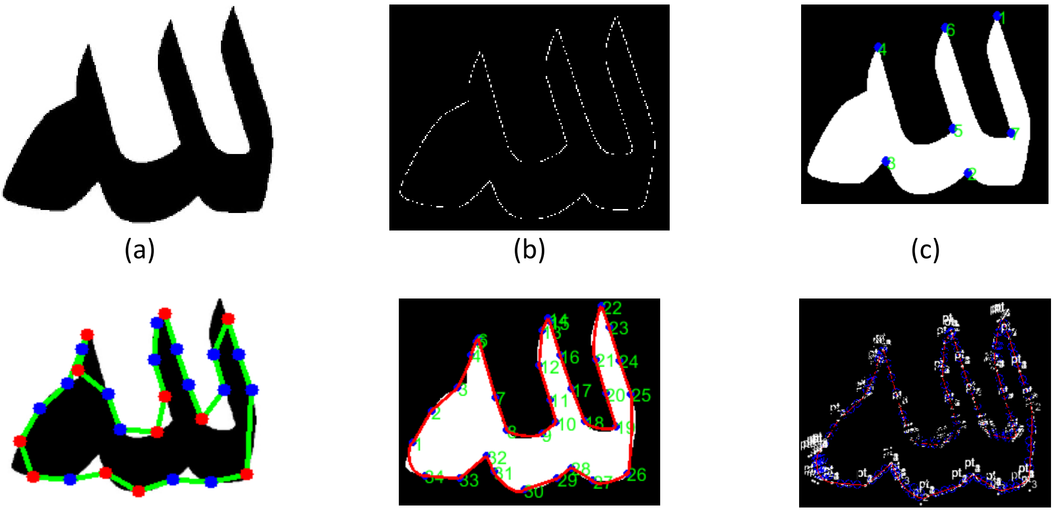
$$B = (1-t)^4P1 + 4 \times (1-t)^3tP2 + 4 \times (1-t)^2t^2P3 + 4 \times (1-t)t^3P4 + t^4P5, t \in [0,1] \qquad Eq(11)$$



(a) (b)

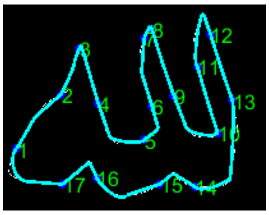
Figure (4.5): (a) Before optimizing curve, (b) After optimizing.

In Figure(4.6) shows the developed method steps, (b) shows the detected boundary of the image of (a), (c) shows the detected corner point for an Arabic word, (d) shows the active contour model into images.(e)Merging corner points and Active Contour model points to obtain set of significant points and ordering them. (f) Shows the fitted cubic Bezier over boundary. (g) Shows the merging curve for all segment of the curve.



(a) (b) (c)

(d) (e) (f)



(g)

Figure(4.6): (a) Digitized image of 'Lelah' Word. (b) Detected boundary. (c) Corner Points. (d) Active contour model (e) Merge corner detection and Snake. (f) Curve Fitting. (g) Optimizing curve.

4.3 Summary

To conclude, a new model has been developed as it has been shown in Figure (4.2) by merging the corner points with Snake model. We could get significant points within the curves that have been inaccessible before.

The developed model helped us to get outlines of different Arabic words with different level of complexities with high quality and acceptable performance. Moreover, it can be easily used.

Chapter 5
Evaluation

Chapter 5
Evaluation

In this chapter, we present the evaluation of our developed method, and discuss the results. This chapter covers the following aspects of the evaluation: a) a description of a set of images used in the experiments, b) the evaluation measures based on which performance was determined, c) an explanation of the experimental setup and procedures, and d) the resulting performance measurements.

5.1 Data Set

We collected 50 images from various sources, and others from web sites. The images can be characterized as follows:

- All of them are binary
- Different types: shapes, words
- Different sizes: between (533*634 to 287*199).

Most importantly, the images had different levels of complexity varying from simple, to moderate, up to high complexity depending on two primary factors: boundary concavities, and the number of corners. Shapes with more corners and deeper boundary concavities are considered highly complex, while shapes with less corners and shallower boundary concavities are considered simple.

5.2 Evaluation Measures

We adopted the following set of evaluation measures:

1. **Average Euclidian distance**: average of square error distances of all contour segments based on average of 3 points L,M,R in each segments, calculated using the following equation:

$$D(B,C) = \sqrt{(X_C - X_B)^2 + (Y_C - Y_B)^2} \quad \text{Eq (12)}$$

B: *bezier point* , C: *contour point*

2. **Max distance**: maximum average distance over the whole contour.
3. **Standard deviation**: Std of the average distances.

$$StD = \text{mean (Euclidian distance)} \qquad \text{Eq (13)}$$

4. **Fitting ratio**: number of curve points located on shape boundary divided by total number of curve points.

5.3 Experiment

The evaluation of our method consists of an experiment which we performed on 50 set of images. In the following subsections, we describe the experimental setup and procedure.

5.3.1 Setup

We implemented our developed correspondence method using MATLAB R2016a. Experiments were executed on an Intel Core 5 Due machine running at 2.60 GHz clock speed with 6 GB RAM.

5.3.2 Procedure

The experiment procedure was composed of the following steps that were performed on each image in the image set. Although, all steps were performed by implementing MATLAB code, a human operator has to manually click button for each action.

Table (5.1): Experiment procedure

No.	Step	Outcome
1	Image input	-
2	Edge detection	Edge map
3	Corner detection	Set of corner points
4	Boundary detection by snake	Set of snake points
5	Merging corner points and snake points	A curve made of the combined set of the two sets above
6	Curve fitting	The curve after fitting to the shape's boundary
7	Optimizing curve	The curve above with reduced number of points

A number of notes are important to mention about some of the steps above:

- Step 2, edge detection, we used the canny edge detection, a lot of edge detection algorithms, such as Robert detector, Prewitt detector, Kirsch detector, Gauss-

Laplace detector and canny detector have been proposed. Among these algorithms, canny algorithm has been used widely in the field of image processing because of its good performance.

- Step 3, corner detection, an option is available to determine the number of desired corner points.
- Step 4, boundary detection, the user could choose between automatic snake initialization with the desired number of initial snake points, or chose to locate each point.

5.4 Results

This section presents the output results of performed the experiment described above on the set of images.

5.4.1 Simple shapes

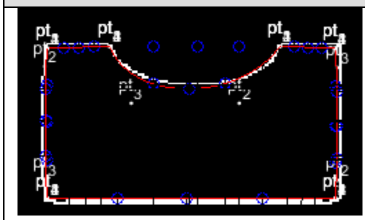
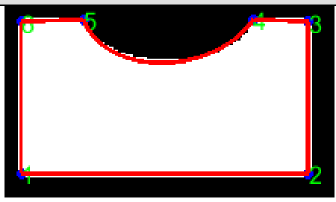
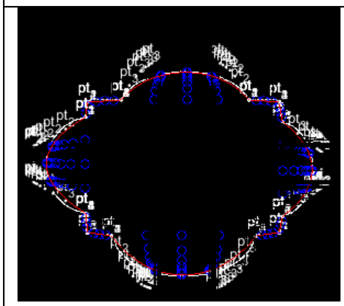
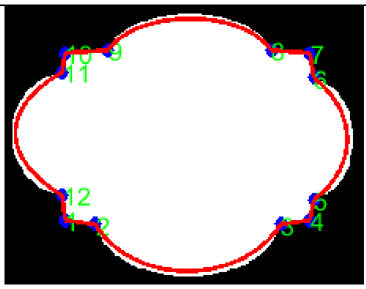
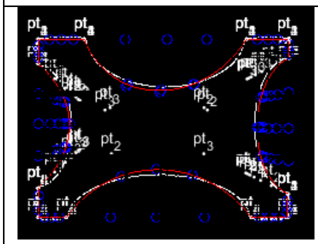
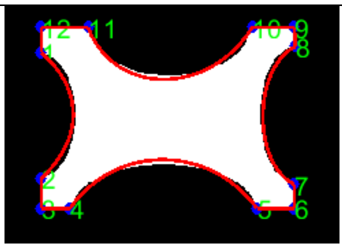
Fitting Bezier Curve	**Segmentation**	**Avg. Ec.Dis. & Avg. Error**
	 **6 Segment**	Avg. Ec. Dis: **0.42** St.D:**0.2** Fitting ratio:**100%** Time:**2.2** sec
	 12 **Segment**	Avg. Ec. Dis:**1.06** St.D:**0.55** Fitting ratio:**91.5**% Time:**6.3**
	 12 seqments	Avg. Ec. Dis:**1.84** St.D:**0.54** Fitting ratio:**90.26**% Time:**5.3**

Figure (5.1): Results for simple complexity.

Figure (5.1) shows that the performance of our method. The results show that we have achieved an average distance approximate to 1 with standard deviation less than one which shows a high level of accuracy.

5.4.2 Moderate complexity shapes

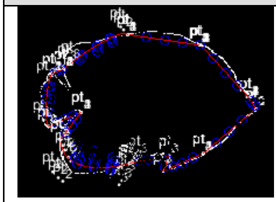
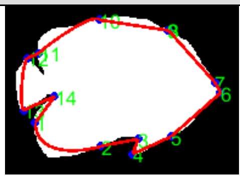
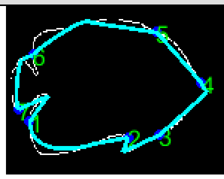
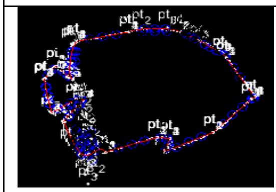
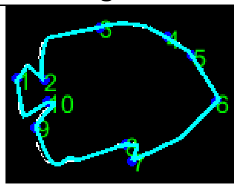
Fitting Bezier Curve	Segmentation	Optimizing curve	Avg. Dis. & Avg. Error
	 **14 Segments**	 **7 Segments**	Avg. Ec. Dis:**1.95** St.D:**0.83** Fitting ratio:**87%** Time:**5.8** sec.
	 **20 Segments**	 **10 Segments**	Avg. Ec. Dis:**0.94** St.D:**0.5099** Fitting ratio:**98.217**% Time:**8.2** sec.

Figure (5.2): Fish Shape for moderate complexity.

As it can be seen from Figure (5.2), our method shows a better performance than other methods (Masood & Sarfraz, 2009; Sarfraz & Khan, 2002; Ferdous A Sohel, Karmakar, Dooley, & Bennamoun, 2010), because the merging curves led to decreasing number of the segments.

The results show that we have achieved an average distance approximate to 1 with standard deviation less than one which shows a high level of accuracy.

Table (5.2) illustrates a comparison between the developed method and others. It presents the results for a range of different segment rates and both the visual and empirical results produced by the Enhance Bezier Curve (EBC), EBC-n models and our method.

Table (5.2): Comparison with others algorithms for Fish Shape.

Alghoritm	Segment=7		Segment=10	
	Max	Avg	Max	Avg
Bizer Curve (Yang et al., 2001)	6.3	4	3.5	1.3
Eenhanced Bizer Curve(Ferdous Ahmed Sohel et al., 2005)	5.5	3	3.2	0.9
EBC-n (Ferdous Ahmed Sohel et al., 2005)	5	2.3	2.9	0.7
This research	4.8	1.95	1.8	0.9

Table (5.2) showed comparison between our method and others, the performance of our method is higher performance comparison with others.

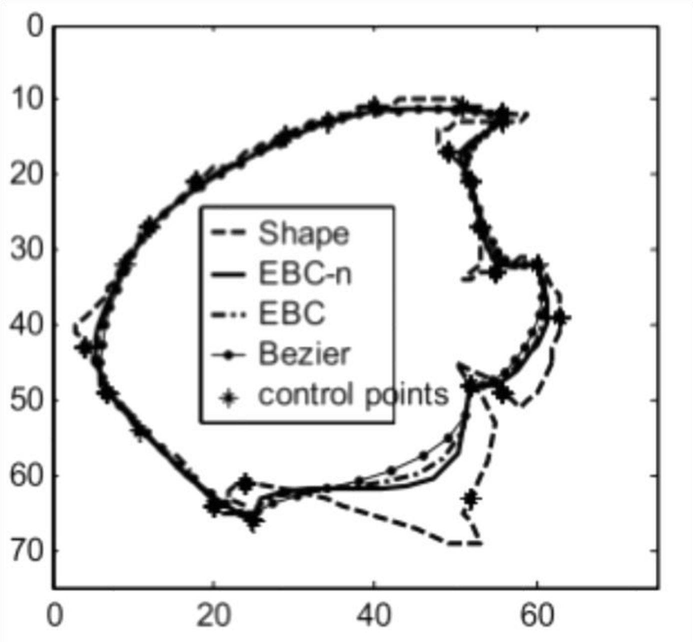


Figure (5.3): Experimental results for others method (Bizer Curve, Eenhanced Bizer Curve).

As can be seen from Figure (5.3), our method shows a better performance than other methods (Masood & Sarfraz, 2009; Sarfraz & Khan, 2002; Ferdous A Sohel, Karmakar, Dooley, & Bennamoun, 2010).

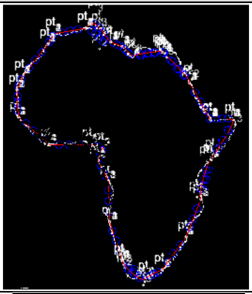
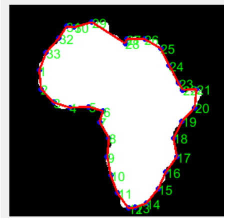
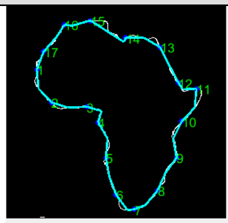
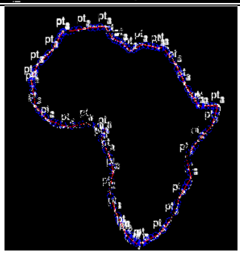
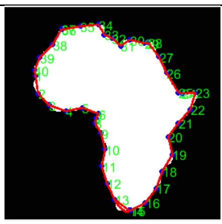
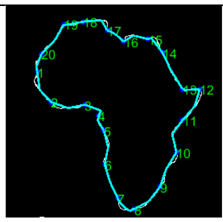
Fitting Bezier Curve	Segmentation	Optimizing curve	Avg. Dis. & Avg. Error
	 33 Segments	 17 Segments	Avg. Ec. Dis:**1.20** St.D:**0.57** Fitting ratio:**95.4**% Time:**9** sec Max=**2.4**
	 40 Segments	 20 Segments	Avg. Ec. Dis:**1.03** St.D:**0.74** Fitting ratio:**96.8%** Time: **10** sec. Max=**2**

Figure (5.4): Africa map for moderate complexity.

Figure (5.4) shows that, the performance of our method acceptable performance. The results show that we have achieved an average distance approximate to 1 with standard deviation less than one which shows a high level of accuracy with 14 segments, while with 12 segments we have achieved an average distance approximate to 1.91.

Table (5.3): comparison others algorithms for Africa map(Masood & Sarfraz, 2009)

Shape	Algorithm	No. of Seg2.	Max error	Avg. error	Computation times(s)
Africa Map 	(Masood & Sarfraz, 2009)	27	1.51	0.79	2.14
	(Sarfraz & Razzak, 2002)	63	1.45	1.09	3.04
	(Sarfraz & Khan, 2004)	23	2.38	1.81	3.37
	(Schneider, 1990)	35	1.57	0.87	2.75
	(Sarfraz & Razzak, 2003)	43	0.95	0.75	5.23
	Our Research	20	1.7	1.20	10
		17	1.9	1.03	9

Table (5.3) shows that we have used fewer segments but more efficient and accurate with other methods (Masood & Sarfraz, 2009; Sarfraz & Khan, 2002; Ferdous A Sohel, Karmakar, Dooley, & Bennamoun, 2010) which used more segment than our methods to get the approximate shape of Africa.

5.4.3 Highly complicated shapes

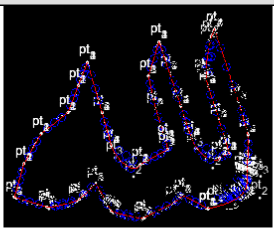
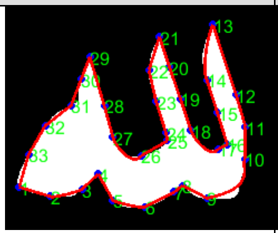
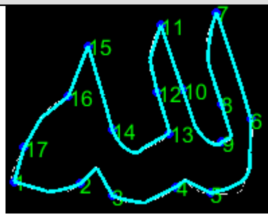
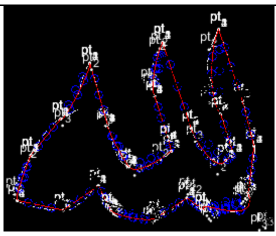
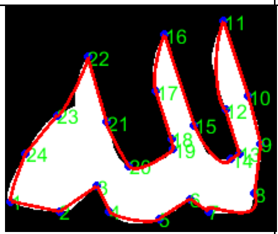
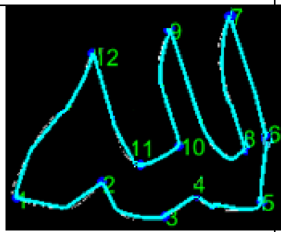
	Fitting Bezier Curve	Segmentation	Optimizing curve	Avg. Dis. & Avg. Error
(a)			 17 segments	Avg. Ec. Dis: **3.04** St.d: **2.95** Fitting ratio: **76.16%** Time: **22.2** sec. Max=**5.9**
(b)			 12 segments	Avg. Ec. Dis:**3.69** St.D:**3.2** Fitting ratio:**67.28%** Time:**16.5** Max =**3.8**

Figure (5.5): Lelah shape fitting proccess

Figure (5.5) shows that our developed method has an acceptable level of performance when compared with other methods (Masood & Sarfraz 2009; Sarfraz & Khan 2002; Ferdous & Bennamoun 2010). The results show in Figure (5.5) (a) that the new method has been achieved an average distance of 3 with standard deviation of 2.95, and the results show in Figure (5.5) (b) that the new method has been achieved an average distance of 3.6 with standard deviation of 3.2 which shows a slight low level of accuracy because a new method used fewer number of segments.

Table (5.4) illustrates the comparison our method with others. It presents the results for (**D**) accurate distortion measurement and for "Lelah" word, it shows that our developed method has an acceptable level of performance when compared with others methods.

Table (5.4): Compare others algorithms for Lelah Word(Ferdous A Sohel et al., 2010).

Shape	**Arabic character**		
Algorithm	**D_{max}**	**D_{ms}**	**D_n, %**
(Ferdous A Sohel et al., 2010)	1.12	0.80	0.61
(Cinque, Levialdi, & Malizia, 1998)	21.1	1.35	0.95
(Sarfraz & Khan, 2002)	1.3	0.95	0.65
(Yang et al., 2001)	1.35	1.2	0.95
(SHDAIFAT I., GRIGAT R., 2003)	1.4	1.4	0.66
Our Research	1.1	1.1	0.24

$$Dn = \frac{no.of\ pixels\ mismatched\ in\ the\ approximated\ shape}{no.of\ pixels\ in\ the\ original\ shape} \quad \text{Eq (14)}$$

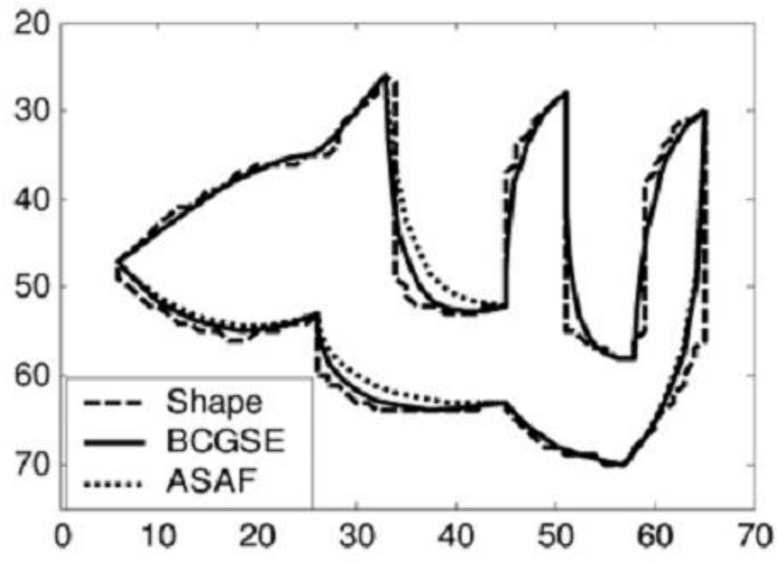


Figure (5.6): Lelah fitting others method(Sohel 2005 and Sarfraz 2002**).**

Figure (5.6) shows the real boundary of the shape using other methods. This explicitly reveals the problem of accuracy and efficiency.

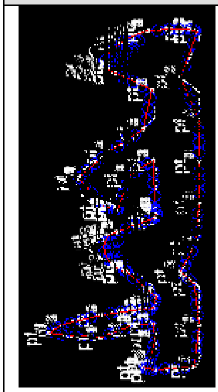
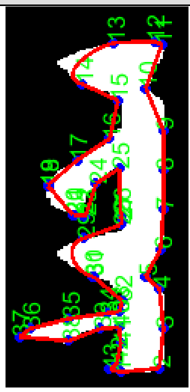
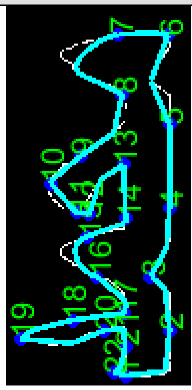
Fitting Bezier Curve	Segmentation	Optimizing curve	Avg. Dis. & Avg. Error
		 22 segments	Avg. Ec. Dis:**1** St.D:**2.2** Fitting ratio:**76.2%** Time: **23.5** sec. Max = **3.2**

Figure (5.7): Mohammad Shape fitting Process.

Figure (5.7) shows that the new method has been achieved moderate accuracy despite of the complexity of the word. The results show that we have achieved an average distance of 1 with standard deviation of 2.2 which shows a slight low level of accuracy.

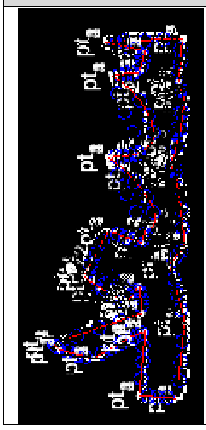
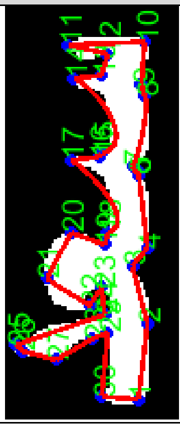
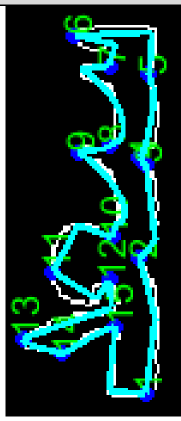
Fitting Bezier Curve	Segmentation	Optimizing curve	Avg. Dis. & Avg. Error
			Avg. Ec. Dis:1.19 St.D:0.70 Fitting ratio:87% Time: 24.2 sec. Max = 2.8

Figure (5.8): Saad Shape fitting process.

Figure (5.8) shows that we achieved moderate accuracy despite of the complexity of the word. The results show that we have achieved an average distance of 1 with standard deviation of 0.70 which shows a slight low level of accuracy.

5.5 Summary:

To sum up, our developed method shows a higher level of efficiency and effectiveness when compared to other methods (Masood & Sarfraz, 2009; Sarfraz & Khan, 2002; Ferdous A Sohel, Karmakar, Dooley, & Bennamoun, 2010). To indicate the effectiveness of our developed method, it has been obvious to us that it achieved a fewer number of corner points; it also has shown a contracted size of space when storing and a high level of accuracy.

Chapter 6
Conclusion and Future Work

Chapter 6

Conclusion and Future Work

6.1 Conclusion

A new method has been developed for outlining shapes using Bezier curves with minimal set of curve points. A distinguishing characteristic of our method is that it combines the active contour method (snake) with corner detection to achieve an initial set of points that is as close to the shape's boundaries as possible.

Our developed method is built upon the following steps:

- Boundary Extraction.
- Corner Detection (Using corner detection technique and Snake point).
- Segment Fitting: the corner points separate sets of data-points into segments.

The method links these points (snake + corner) into a compound Bezier curve, and iteratively improves the fitting of the curve over the actual boundaries of the shape.

In this thesis, the developed method has come to good results in respect to efficiency and accuracy, such as:

- It achieved 86% of accuracy when measured to true shape boundary.
- It also shows an increase number of segments to the shape after being treated by our method.

Finally, an efficient method has been implemented to enhance word and shape outlining capability based on combining Active Contour Model and Corner detection is more efficient and accuracy.

6.2 Future work

For future research work, the following suggested ideas can enhance the system:

1 Applying on shapes with opening and multi-segments words.

2 Suggesting a time saving method on images capturing.

3 The effect of applying a mixture of methods to obtain efficiency and effectiveness in capturing of images.

References

References

Altarawneh, N. M., Luo, S., Regan, B., & Sun, C. (2015). A Modified Distance Regularized Level Set Model for Liver Segmentation. *2015*, *6*(1), 1–11.

Altarawneh, N. M., Luo, S., Regan, B., Sun, C., & Jia, F. (2014). Global Threshold and Region - Based Active Contour Model For Accurate, *5*(3), 1–11. https://doi.org/http://dx.doi.org/10.5121/sipij.2014.5301

Berger, M., N, I. L. C. R. I., Cede, B. P. V. X., & Mohr, R. (1990). Towards Autonomy in Active Contour Models, 847–851.

Cheng, J., & Liu, C. (2008). Image segmentation with GVF snake and corner detection. *Proceedings - International Conference on Computer Science and Software Engineering, CSSE 2008*, *1*, 1017–1020. https://doi.org/10.1109/CSSE.2008.1156

Cinque, L., Levialdi, S., & Malizia, A. (1998). Shape description using cubic polynomial Bezier curves. *Pattern Recognition Letters*, *19*(9), 821–828. https://doi.org/10.1016/S0167-8655(98)00069-5

Globacnika, T., Žalikb, B., Globačnik, T., & Žalik, B. (2010). An efficient raster font compression for embedded systems. *Pattern Recognition*, *43*(12), 4137–4147. https://doi.org/doi:10.1016/j.patcog.2010.07.018

Groch, W. (2011). *Computer vision*. https://doi.org/10.1111/j.1475-1313.2011.00834.x

Haralambous, Y. (2007). *Fonts & Encodings*. *Production* (First Edit). © 2007 O'Reilly Media, Inc. All rights reserved. Retrieved from https://books.google.co.jp/books?id=qrElYgVLDwYC

Harris, C. (1987). Determination of Ego-Motion from Matched Points. *Alvey Vision Conference*. https://doi.org/10.5244/C.1.26

Harris, C., & Stephens, M. (1988). A Combined Corner and Edge Detector. *Procedings of the Alvey Vision Conference 1988*, 147–151. https://doi.org/10.5244/C.2.23

Karow, P. (1994). *Font Technology*. © Springer-Verlag Berlin Heidelberg 1994.

https://doi.org/10.1007/978-3-642-78505-4

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: active contour model. *International Journal on Computer Vision*.

Khan, K., & Lobiyal, D. K. (2017). Bèzier curves based on Lupaş (p , q) -analogue of Bernstein functions in CAGD. *Journal of Computational and Applied Mathematics*, *317*, 458–477. https://doi.org/10.1016/j.cam.2016.12.016

Kitchen, L. (1993). Early Jump-Out Corner Detectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(8), 823–828. https://doi.org/10.1109/34.236246

Liljeback, P., Pettersen, K. Y., Stavdahl, O., & Gravdahl, J. T. (2012). A control framework for snake robot locomotion based on shape control points interconnected by Bezier curves. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3111–3118. https://doi.org/10.1109/IROS.2012.6386117

Lu, L., & Xiang, X. (2016). Note on multi-degree reduction of B ´ ezier curves via modified. *Journal of Computational and Applied Mathematics*. https://doi.org/10.1016/j.cam.2016.11.001

Masood, A., & Sarfraz, M. (2009). Capturing outlines of 2D objects with B??zier cubic approximation. *Image and Vision Computing*, *27*(6), 704–712. https://doi.org/10.1016/j.imavis.2008.07.012

On, O. (2014). Bézier Curves to recognize Multi - Font Arabic, University of Nizwa , Oman. *The International Arab Conference on Information Technology*, 105–112.

Sarfraz, M., & Khan, M. A. (2002). Automatic outline capture of Arabic fonts. *Information Sciences*, *140*(3–4), 269–281. https://doi.org/10.1016/S0020-0255(01)00176-1

Sarfraz, M., & Khan, M. A. (2004). An automatic algorithm for approximating boundary of bitmap characters, *20*, 1327–1336. https://doi.org/10.1016/j.future.2004.05.024

Sarfraz, M., & Raza, S. A. (2001). Capturing outline of fonts using genetic algorithm and splines. *Proceedings of the International Conference on Information Visualisation*, *2001–Janua*(1), 738–743. https://doi.org/10.1109/IV.2001.942138

Sarfraz, M., & Razzak, F. A. (2003). A Web based system to capture outlines of Arabic fonts. *Information Sciences*, *150*(3–4), 177–193. https://doi.org/10.1016/S0020-0255(02)00376-6

Sarfraz, M., & Razzak, M. F. A. (2002). An algorithm for automatic capturing of the font outlines, *26*, 795–804.

Schneider, P. J. (1990). An Algorithm for Automatically Fitting Digitized Curves. In A. S. Glassner (Ed.), *Graphics Gems* (pp. 612–626). Academic Press.

SHDAIFAT I., GRIGAT R., L. D. (2003). ACTIVE SHAPE LIP MODELING. *Proc. IEEE Int.*, 875–878. Retrieved from http://ieeexplore.ieee.org./document/1246820/

Smith, S., & Brady, J. M. (1997). {SUSAN}: A New Approach to Low Level Image Processing. *International Journal of Computer Vision*, *23*, 45–78.

Sofiyanti, N., Fitmawati, D. I., & Roza, A. A. (2015). *Stenochlaena Riauensis (Blechnaceae), A new fern species from riau, Indonesia*. *Bangladesh Journal of Plant Taxonomy* (2008th ed., Vol. 22). Springer Publishing Company, Incorporated ©2007. https://doi.org/10.1007/s13398-014-0173-7.2

Sohel, F. A., Karmakar, G. C., & Dooley, L. S. (2007). Bezier curve-based character descriptor considering shape information. *Proceedings - 6th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2007; 1st IEEE/ACIS International Workshop on E-Activity, IWEA 2007*, (Icis), 212–216. https://doi.org/10.1109/ICIS.2007.69

Sohel, F. A., Karmakar, G. C., Dooley, L. S., & Arkinstall, J. (2005). Enhanced bezier curve models incorporating local information. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *IV*, 253–256. https://doi.org/10.1109/ICASSP.2005.1415993

Sohel, F. A., Karmakar, G. C., Dooley, L. S., & Bennamoun, M. (2010). A Bezier curve-based generic shape encoder. *IET Image Processing*, *4*(2), 92. https://doi.org/10.1049/iet-ipr.2008.0128

Teh, C. H., & Chin, R. T. (1989). On the Detection of Dominant Points on Digital Curve. *Pami*, *11*(8), 859–872. https://doi.org/10.1109/34.31447

Wang, H., & Brady, M. (1995). Real-Time Corner Detection Algorithm for Motion Estimation. *IMage and Vision Computing*, *13*(9), 695–703. https://doi.org/10.1016/0262-8856(95)98864-p

Willis, A. (2009). An Algebraic Model for fast Corner Detection. *IEEE 12th International Conference on Computer Vision, IEEE. pp.*(Iccv), 2296–2302. Retrieved from http://ieeexplore.ieee.org/document/5459443/

Xu, C., & Prince, J. L. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, *7*(3), 359–369. https://doi.org/10.1109/83.661186

Yahya, F., Md Ali, J., Majid, A. A., & Ibrahim, A. (2006). An automatic generation of G1 curve fitting of arabic characters. *Proceedings - Computer Graphics, Imaging and Visualisation: Techniques and Applications, CGIV'06*, *2006*, 542–547. https://doi.org/10.1109/CGIV.2006.18

Yang, H. M., Lu, J. J., & Lee, H. J. (2001). A bezier curve-based approach to shape description for Chinese calligraphy characters. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, *2001–Janua*, 276–280. https://doi.org/10.1109/ICDAR.2001.953798

Zhang, J., Yu, J., & Lin, H. (2007). Capturing character contours from images of ancient chinese calligraphy. *Proceedings - 2nd Workshop on Digital Media and Its Application in Museum and Heritage, DMAMH 2007*, 36–41. https://doi.org/10.1109/DMAMH.2007.4414523

Zhang, K., Song, H., & Zhang, L. (2010). Active contours driven by local image fitting energy. *Pattern Recognition*, *43*(4), 1199–1206. https://doi.org/10.1016/j.patcog.2009.10.010

Appendix A

Image Set 1

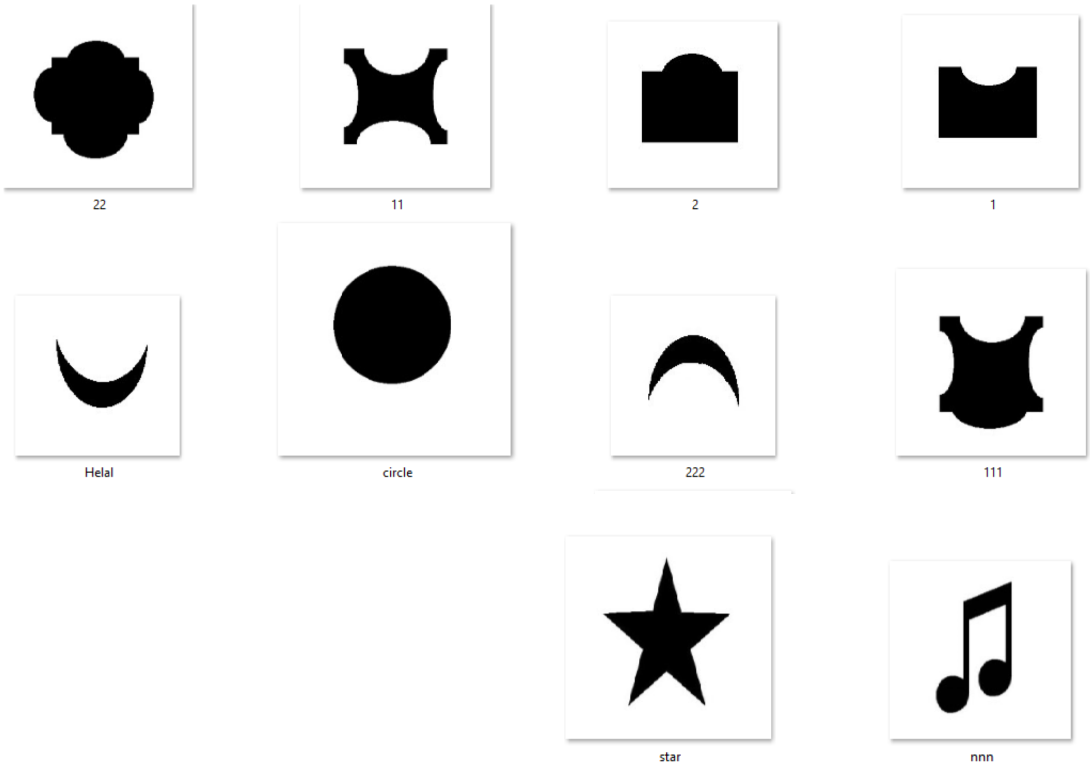


Image Set 2

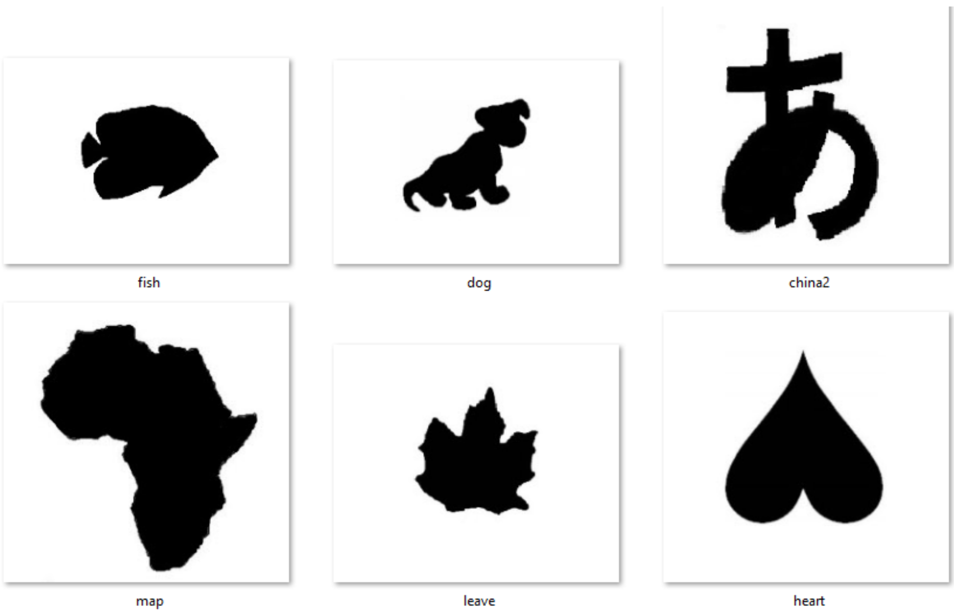


Image Set ٣

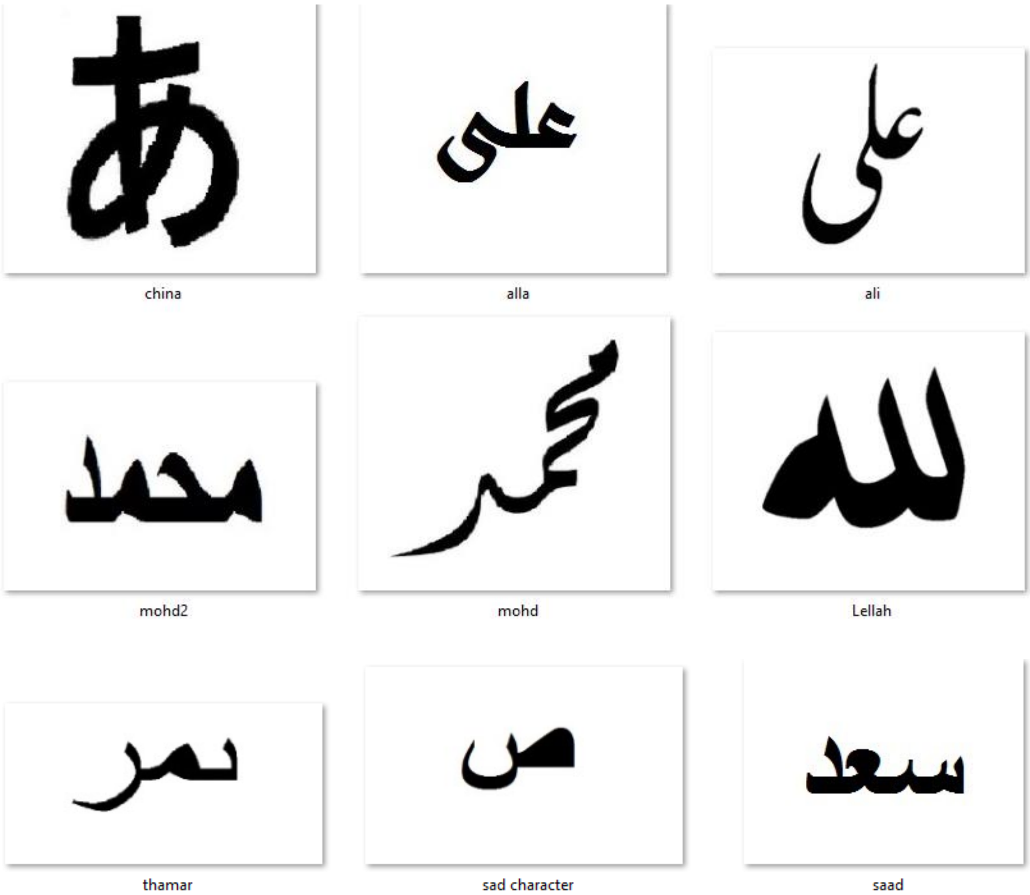
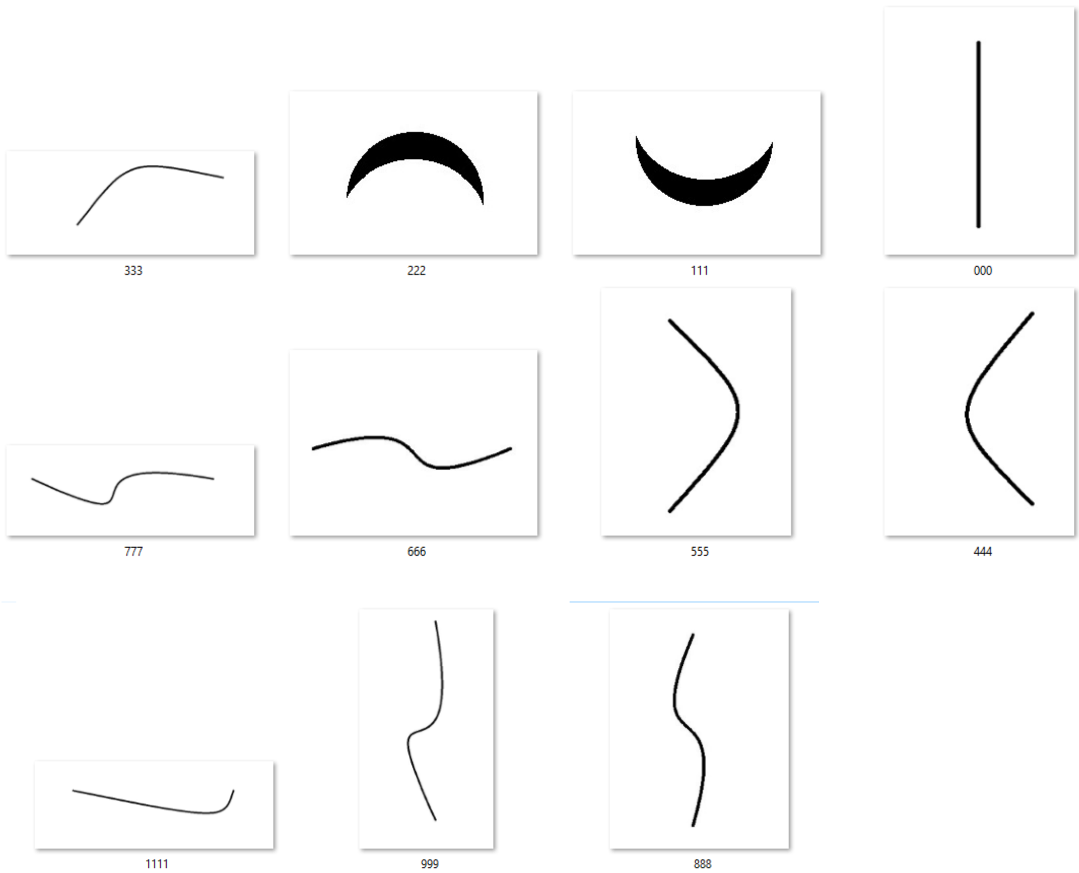
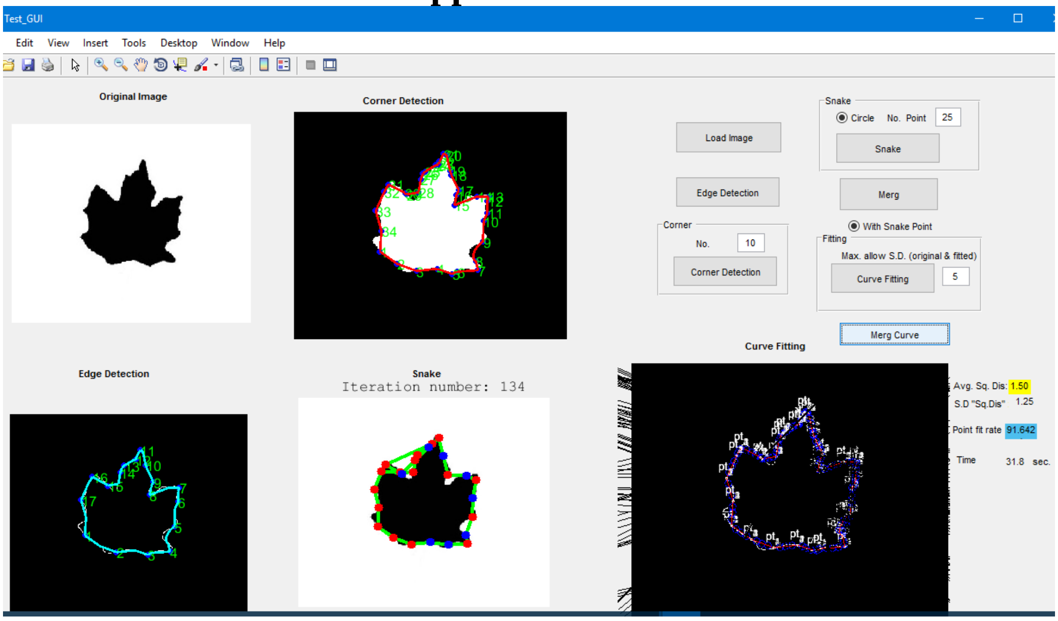


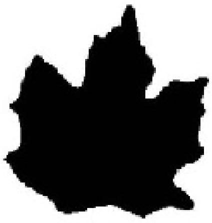
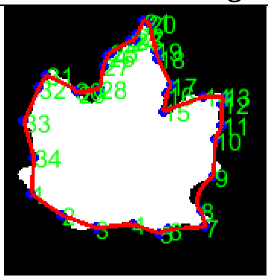
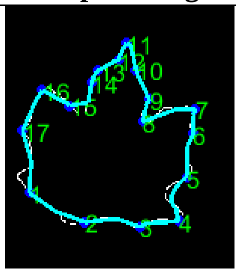
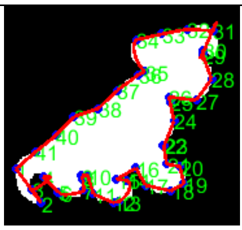
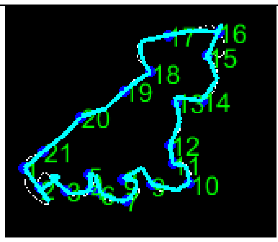
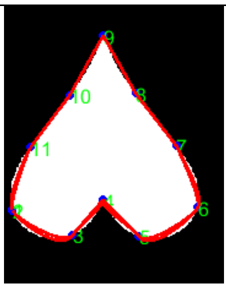
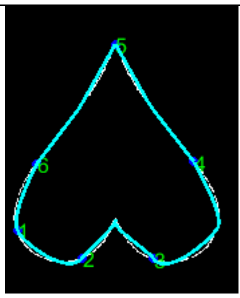
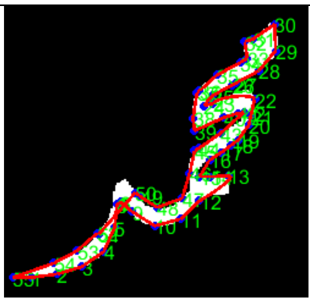
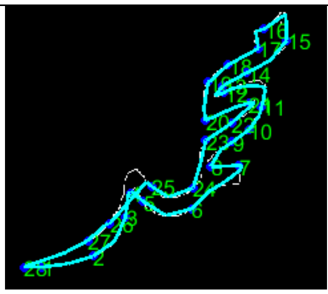
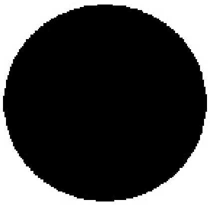
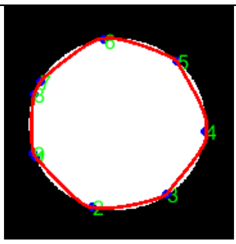
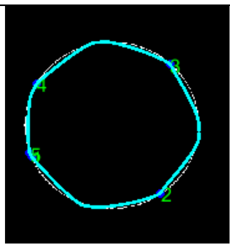
Image Set ٤

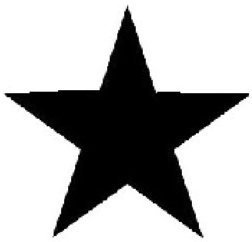
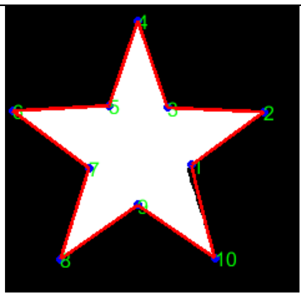
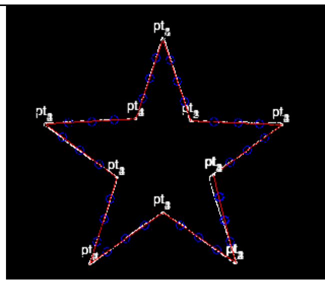
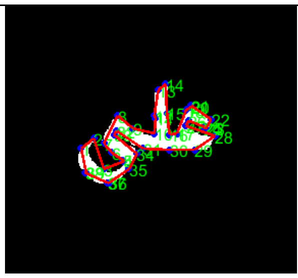
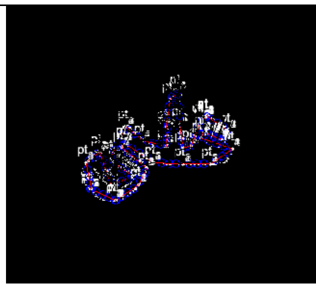


Application GUI



Test Set

Image	Original Image	Curve fitting	Optimizing Curve
leave			
dog			
heart			
mohd			
Circle			

star			
Ala			

Appendix B