REPRESENTATION OF RATIONAL BÉZIER QUADRATICS USING GENETIC ALGORITHM, DIFFERENTIAL EVOLUTION AND PARTICLE SWARM OPTIMIZATION

ZAINOR RIDZUAN YAHYA

UNIVERSITI SAINS MALAYSIA

2013

REPRESENTATION OF RATIONAL BÉZIER QUADRATICS USING GENETIC ALGORITHM, DIFFERENTIAL EVOLUTION AND PARTICLE SWARM OPTIMIZATION

by

ZAINOR RIDZUAN YAHYA

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

July 2013

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisors Professor Dr. Abd. Rahni Mt. Piah and Associate Professor Dr. Ahmad Abd. Majid of Universiti Sains Malaysia, for all their guidance, support, patience, advice and suggestions to enable me to write this thesis and complete my Ph. D work, for which I am extremely grateful.

I am also indebeted to my field supervisor Professor Sarfraz of Kuwait University. It was particularly kind of him to allow me to refer to his wide expertise of this field. My special honour also goes to Universiti Sains Malaysia, especially to the Dean of the School of Mathematical Sciences and the Ministry of Higher Education of Malaysia for giving me an opportunity to study and providing me with the facilities and financial assistance to pursue my higher degree.

Last but certainly not least, I would like to express my love and appreciation to my parents and my wife for their understanding and great support at all the time and to all my other family members and friends who had either directly or indirectly help me to complete my study.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GA	Genetic Algorithm
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SE	Simulated Evolution
DE	Differential Evolution
HSA	Harmony Search Algorithm
ACO	Ant Colony Optimization
TS	Tabu Search
ANN	Artificial Neural Networks
SSE	Sum Square Error
ISE	Integral Square Error
MSE	Mean Square Error

PERWAKILAN KUADRATIK BÉZIER NISBAH MENGGUNAKAN ALGORITMA GENETIK, EVOLUSI PEMBEZAAN DAN PENGOPTIMUMAN SEKAWAN PARTIKEL

ABSTRAK

Perwakilan data adalah masalah yang mencabar dalam bidang seperti pembinaan semula fon, imej perubatan dan imej yang diimbas. Teknik matematik langsung pada kebiasaannya memberikan ralat terkecil tetapi kadang-kadangnya mengambil masa yang lebih lama untuk pengiraan. Sebagai alternatif, teknik aplikasi kecerdasan buatan digunakan secara meluas untuk masalah pengoptimuman dengan masa pengiraan yang lebih pendek. Disamping itu, penggunaan kaedah kecerdasan buatan untuk perwakilan data menjadi semakin popular akhir-akhir ini. Oleh itu, tesis ini didedikasikan untuk perwakilan lengkung dan permukaan. Tiga teknik pengkomputeran lembut iaitu Algoritma Genetik (AG), Evolusi Pembezaan (EP) dan Pengoptimuman Sekawan Partikel (PSP) digunakan untuk memanipulasi lengkung dan permukaan yang diingini. Teknik-teknik ini telah digunakan untuk mengoptimumkan titik kawalan dan pemberat dalam keterangan fungsi splin yang digunakan. Komponen pra-pemprosesan seperti pengesanan bucu dan pemparameteran panjang perentas juga diterangkan dalam tesis ini. Bagi setiap teknik pengkomputeran lembut yang dicadangkan, talaan parameter dilakukan sebagai satu kajian penting. Hasil tambah ralat kuasa dua (HTRKS) digunakan sebagai fungsi objektif. Maka, kajian ini juga adalah satu masalah peminimuman dengan nilai terbaik untuk titik kawalan dan pemberat diperolehi apabila nilai HTRKS adalah minimum. Kuadratik Bézier nisbah telah digunakan untuk perwakilan lengkung. Pembinaan semula permukaan telah dicapai dengan melanjutkan kuadratik Bézier nisbah kepada pasangannya, dwi-kuadratik Bézier nisbah. Kaedah lengkung dan permukaan yang kami cadangkan, dengan bantuan tambahan

daripada teknik pengkomputeran lembut telah dugunakan untuk pengvektoran bentuk serta objek 2D dan 3D. Algoritma yang terperinci telah dirangka untuk menukarkan model digital 2D dan 3D kepada bentuk pengvektoran. Pelbagai demonstrasi praktikal daripada pembinaan semula fon yang mudah kepada perwakilan imej perubatan telah dibuat untuk menyokong hasil kerja yang berjaya oleh algoritma yang telah direka. Satu kajian perbandingan skim yang dicadangkan dengan yang sedia ada telah dibuat sebagai satu bahagian yang perlu dalam kajian ini. Perbandingan ini mengandungi kedua-dua bentuk, penglihatan dan berangka.

REPRESENTATION OF RATIONAL BÉZIER QUADRATICS USING GENETIC ALGORITHM, DIFFERENTIAL EVOLUTION AND PARTICLE SWARM OPTIMIZATION

ABSTRACT

Data representation is a challenging problem in areas such as font reconstruction, medical image and scanned images. Direct mathematical techniques usually give smallest errors but sometime take a much longer time to compute. Alternatively, artificial intelligence techniques are widely used for optimization problem with shorter computation time. Besides, the usage of artificial technique for data representation is getting popular lately. Thus, this thesis is dedicated for the representation of curves and surfaces. Three soft computing techniques namely Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO) are utilized for the desired manipulation of curves and surfaces. These techniques have been used to optimize control points and weights in the description of spline functions used. Preprocessing components such as corner detection and chord length parameterization are also explained in this thesis. For each proposed soft computing technique, parameter tuning is done as an essential study. The sum of squares error (SSE) is used as an objective function. Therefore, this is also a minimization problem where the best values for control points and weights are found when SSE value is minimized. Rational Bézier quadratics have been utilized for the representation of curves. Reconstruction of surfaces is achieved by extending the rational Bézier quadratics to their rational Bézier bi-quadratic counterpart. Our proposed curve and surface methods with additional help from soft computing techniques have been utilized to vectorize the 2D and 3D shapes and objects. Detailed algorithms have been devised to convert the 2D and 3D digital models into vectorized form. Various practical demonstrations from

simple fonts reconstruction to medical images representation have been made to support the successful working of the devised algorithms. Comparative study of our proposed schemes with the existing ones has been made as an essential part of this study. The study includes both visual and numerical comparison.

CHAPTER 1

INTRODUCTION

Traditionally, the data fitting problem is solved using direct methods such as data linearization and least squares method. However, nonsystematic and big amount of data are difficult to be solved directly. Consequently, the soft computing methods are increasingly accepted as an option to solve this kind of problem. Soft computing technique does not only provide a good solution but is computationally efficient. This is important in the regeneration of data process that requires fast computation. This thesis reviews some of the soft computing methods used in this field.

Soft computing refers to a fusion of methodologies that mainly bring together neural networks, fuzzy logic and evolutionary algorithms (Dubois and Prade, 1998). There are many techniques in soft computing and they have been used to solve many problems in engineering. Some of the soft computing techniques are Genetic Algorithm (GA), Ant System, Simulated Annealing (SA), Simulated Evolution (SE), Particle Swarm Optimization (PSO), Tabu Search and Harmonic Search.

Curve fitting is the process of constructing a curve or mathematical function that has the best fit to a series of data points that are possibly subjected to constraints. Curve fitting can involve either interpolation where an exact fit to the data is required or smoothing in which a 'smooth' function is constructed that approximately fits the data. Fitted curves can be used as an aid for data visualization to infer values of a function where no data are available and to summarize relationships among two or more variables.

Over the last two decades, soft computing methods are gaining attention in the field of Computer Aided Geometric Design (CAGD). The study focused on data representation have also been carried out. Some study has given good results but did not show positive results. Among the findings that gave result was the study using Genetic Algorithm (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO). The previous method using DE was not carried out very well but studies that used GA and PSO gave a very good result. In this regard we would like to solve the problem of data representation using these three techinques. All the selected method certainly converges as described in the literature.

A Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions in optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Choosing the best parameter is important to ensure convergence.

Inspired by a proposed method by Jones et al. (1993), Huyer and Neumaier (1999) presented a global optimization algorithm based on multilevel coordinate search. It is guaranteed to converge if the function is continuous in the neighbourhood of a global minimizer. By starting a local search from certain good points, an improved convergence result is obtained. They discussed implementation details and gave some numerical results.

Particle Swarm Optimization (PSO) is an optimization technique proposed by Kennedy and Eberhart by means of particle swarm (Weise, 2009). PSO incorporated swarming behaviours that were observed in flocks of birds, school of fish, swarm of bees and even social behaviour, from where the idea was emerged. PSO is a population-based optimization tool which could be implemented and applied easily to solve various function optimization problems or problems that can be transformed to function optimization problems. To apply PSO successfully, one of the key issues is to find that how to map the problem solution into the PSO article which directly affect its feasibility and performance.

Storn and Price (1997) propose a new floating point encoded evolutionary algorithm for global optimization and called it Differential Evolution (DE) owing to a special kind of differential operator, which they invoke to create a new offspring from parent chromosomes instead of the classical crossover or mutation (Das et al., 2008). Easy methods of implementation and negligible parameter tuning made the DE algorithm quite popular. Donor vector selection is essential in DE as the vector ensures convergence.

Functional networks are a generalization of the standard neural networks in the sense that the weights are now replaced by neural functions which can exhibit, in general, a multivariate character (Iglesias et al., 2004). In addition, while working with functional networks they are able to connect different neuron outputs with convenience. Furthermore, different neurons can be associated with neural functions from different families of functions. As a consequence, the functional networks exhibit more flexibility than the standard neural networks (Castillo, 1998).

Artificial Immune System (AIS) is a model of the immune system that can be used by immunologists to explain, experiment and predict activities that would be difficult or impossible in "wet-lab" experiments (Garrett, 2005). This is also known as "computational immunology". AIS is also an abstraction of one or more immunological processes. Since these processes protect us on a daily basis from the ever-changing onslaught of biological and biochemical entities that seek to prosper at our expense. It is also a reason why they may be computationally useful.

In metallurgy and material sciences, annealing is a heat treatment of material with the goal of altering its properties such as hardness. Metal crystals have small defects, dislocations of ions which weaken the overall structure. By heating the metal, the energy of the ions and hence, their diffusion rate is increased. Then the dislocations can be destroyed and the structure of the crystal is reformed as the material cools down and approaches its equilibrium state. When annealing a metal, the initial temperature must not be too low and the cooling must be done sufficiently slow so as to avoid the system getting stuck in a meta-stable, non-crystalline state representing a local minimum of energy (Weise, 2009).

1.1 Curve Fitting using Soft Computing Techniques

In this section, some of the soft computing techniques used for curve fitting have been discussed.

1.1.1 Genetic Algorithm (GA)

Raza and Sarfraz (2001) investigated the use of GA to perform curve fitting. They use Akaike's information criterion as an objective function. They use the Arabic fonts "Ali" and "Pound" sign as examples. Their method determines an appropriate number and location of knots automatically and simultaneously. This research is one of the earliest soft computing technique used to solve the curve fitting problem. It can be seen that no parameter tuning had been done in this research.

Singh et al. (2003) proposed a method to approximate digital planar curves with line segments and circular arcs using GA. Breakpoints on the digital curve were located in such a manner that when line segments and circular arcs were appropriately fitted between all pairs of adjacent breakpoints, an approximating error (i.e. an integral square error) was minimized. They used four digital test curves (i.e. chromosome-shaped curve with 60 points, figure-of-eight curve with 45 points, leaf-shaped curve with 120 points and curve with four semicircles with 102 points) as their test cases.

Sarfraz et al. (2010) presented an algorithm to capture outlines of bitmap characters. A cubic function with one shape parameter was used. GA was applied to find values of the shape parameter. Their procedure in capturing outlines consisted of the steps: boundary detection, detection of corner points, break points and fitting the curve. They used "Pound" and "Lambda" signs as their examples.

1.1.2 Simulated Annealing (SA)

Sarfraz et al. (2006) developed a non-deterministic evolutionary approach to approximate outlines of planar shapes. Non-uniform rational B-splines (NURBS) had been utilized as underlying approximating curves. SA was used to optimize the weight and knots simultaneously. The optimized NURBS model had been fitted over contour data of planar shapes for the ultimate and automatic outputs.

SA was also employed to optimize knots (Sarfraz and Riyazuddin, 2006) and weights (Sarfraz et al., 2006), separately. Then, Sarfraz (2008b) proposed a new technique to capture outlines of generic shapes. SA was used to optimize shape parameters in the generalized cubic spline. All of the research using SA was conducted by Sarfraz and his co-researchers. Their study gave quite good results and for their examples they used the symbols like "Pound", "Lambda" and "Plane".

1.1.3 Functional Networks (FN)

Iglesias and Galvez (2001) used functional networks to fit the given set of data from tensor product parametric surfaces. The performance of this scheme was illustrated for the case of Bézier surface. First, they built the simplest functional network representing such a surface and then they used it to determine the degree and coefficients of bivariate polynomial that fit given data. They calculated mean and root mean squared errors for different degrees of the approximating polynomial surface, which were used as their criterion of a good fitting.

Iglesias et al. (2004) extended their work to B-spline surface used in surface reconstruction. A careful analysis of errors mades it possible to determine the number of B-spline surface fitting control points that best fit the data points. This analysis included the use of two sets of data (training and testing data) to check overfitting which does not occur here.

Iglesias and Galvez (2008) introduced the RBS functional networks which is a new type of functional networks based on weighted B-spline basis functions that reproduces functional networks to curve fitting problem. This study did not seem to proceed further in this field. There was no latest study since 2008. The authors have moved on to study GA and PSO later which has been explained in the next section.

1.1.4 Particle Swarm Optimization (PSO)

Delint et al. (2009) discussed an alternative solution for curve fitting based on PSO by generating randomly the weight and control points of the NURBS curve, which are used to calculate NURBS points. Delint et al. (2010) then implemented PSO on NURBS curve approximation where the weights of the curve can be adjusted accordingly. The experiments were conducted on various parameterization methods for approximating the curves.

Cobo et al. (2007) combined least-squares and PSO in fitting 3D data points through free-form parametric curves. They used Bézier curve for their implementation. The sum of squared errors was used as fitness function. The small number of data points had been used in the research and simple examples were used. They also did not show the application of their proposed method.

1.1.5 Simulated Evolution (SE)

Sarfraz et al. (2005) presented an application of SE to the curve fitting problem using NURBS. They described a mapping scheme of the problem to SE followed by a proposed algorithm's outline with the result. The error and computation time were not reported. This research can be a good start on understanding SE to solve curve representation.

1.1.6 Multilevel Coordinate Search (MCS)

Sarfraz and Naelah (2009) proposed MCS for an outline capture of planar images. The overall technique had various phases including extracting outlines of images, detecting corner points from the detected outline and curve fitting. The idea of MCS was used to optimize shape parameters in the description of the generalized conic spline. The spline method produced optimal results for the approximate vectorization of the digital contour obtained from the generic shapes. This was the only one data fitting problem which is employed in Multilevel Coordinate Search. The error was not informed nevertheless the examples of data fitting gave a good visual result.

1.1.7 Differential Evolution (DE)

Priza et al. (2010) proposed an alternative solution for Bézier curve fitting using Differential Evolution (DE) algorithm. DE algorithm was conducted randomly to generate control points of a Bézier curve. These generated control points were used to calculate Bézier curve points. A fitness function of the DE algorithm was computed in order to search for minimum errors. Simply the data points had been used and there were no actual examples in this study. Although there were a good results shown but it is yet not relevant.

1.2 Surface Fitting using Soft Computing Techniques

Soft computing techniques used for surface fitting include genetic algorithm (GA), simulated annealing (SA), functional networks (FN), particle swarm optimization (PSO), artificial immune system (AIS) and hybrid techniques.

1.2.1 Genetic Algorithm (GA)

Gálvez et al. (2012) introduced a method for surface reconstruction from clouds of noisy 3D data points. Their method applied the GA paradigm iteratively to fit given cloud of data points by using strictly B-spline polynomial surfaces. GA was applied in two steps: determine parametric values of data points; followed by computing surface knot vectors. Fitting surface was calculated using least-squares method either by SVD (singular value decomposition) or LU methods. The method yielded very accurate results even for surfaces with singularities, concavities, complicated shapes or nonzero genus. Most of the results obtained were of a good value. It was carried out by researchers from University of Cantabria, Spain which involve Iglesias and Galvez. Even some said that Genetic Algorithm does not give good results but Galvez has proven GA is also able to give a very fine result in his research. Furthermore, he used such hard examples to set or fix for data fitting on it. The error given is very good.

1.2.2 Simulated Annealing (SA)

Wen et al. (2006) solved surface fitting problem by creating fair ship hull surface using NURBS. He developed an optimized NURBS ship hull fitting approach using SA as well as evaluated and analysed (in term of accuracy, fairness and speed of processing) according to their proposed approach. The quantity of data points and amount of runtime were not reported which shows that the whole research was not well explained.

1.2.3 Functional Networks (FN)

Functional networks are used to fit a given set of data of a tensor product parametric surface (Iglesias and Galvez, 2001). The performance of this method has been illustrated for the case of Bézier surface. First, they built the simplest functional network representing such a surface and then they used it to determine the degree and coefficients of the bivariate polynomial surface that fits the given data better. They calculated the mean and the root mean squared errors for different degrees of the approximating polynomial surface which were used as their criterion of a good fitting. In addition, FN provided a procedure to describe parametric tensor product surface in terms of families of chosen basis functions. This new approach was very general and can be applied not only to Bézier but also to any other interesting family of tensor product surfaces.

Iglesias et al. (2004) also used B-spline surface for surface reconstruction. A careful analysis of the errors made it possible to determine the number of B-spline surface fitting control points that best fit data points. This analysis also includes the use of two sets of data (training and testing) to check for overfitting. Even though the quantity of data points was stated, but there is no runtime reported. However the error obtained was quite reliable.

1.2.4 Particle Swarm Optimization (PSO)

Galvez et al. (2008) used PSO for Bézier surface reconstruction. They presented a simple but illustrative example to discuss performance of their proposed method. Sum of squared errors was used as fitness function. Galvez and Iglesias (2012) applied PSO approach to reconstruct a NURBS surface of certain order from a given set of 3D data points. Their surface reconstruction technique consists of two main tasks, surface parameterization and surface fitting. Their method allows users to obtain relevant surface data (such as parametric values of data points, knot vectors, control points and their weights) without a requirement for pre-/postprocessing. It yielded very good results even in the presence of problematic features, such as multi-branches, high-genus surfaces and real world scanned objects. The outcome given was very good and latest. Moreover, the examples used in the researches were very relevant to the problem.

1.2.5 Artificial Immune System (AIS)

Ulker and Isler (2007) presented a surface fitting problem solution using AIS based on B-splines. Their method can determine appropriate number and locations of knots automatically and simultaneously. They compared their proposed method to GA. All the given data points, runtime and error were good. For the examples the test function was just merely used.

1.2.6 Hybrid Metaheuristic

Fadni and Shamsuddin (2008) proposed a method for surface reconstruction based on hybrid soft computing techniques of Kohonen Network and Particle Swarm Optimization (PSO). Kohonen network learns sample data through mapping grid that can grow. The implementation was executed by generating Kohonen mapping framework of data subsequent to the learning process. Consequently, the learned and well represented data became the input for surface fitting procedure. Their algorithm are applied on different types of curves and surfaces to observe its ability in reconstructing objects while preserving its original shape.

Artificial intelligence techniques considered by Galvez et al. (2007) were for the curve/-surface parameterization where the use of genetic algorithm was proposed and for functional constraints problems, where functional networks scheme was applied. Both approaches were combined with least squares approximation method in order to yield a suitable method for Bézier surface fitting.

1.3 Thesis Objectives

The first objective of this research is to use soft computing techniques in solving curve and surface fitting problems. Soft computing has become a common techniques that is used to solve the problem in engineering and optimization cases. However, in the past two decades soft computing method is gaining attention in the field of CAGD. For that reason, we have used this method specifically to solve curve and surface representation.

The second objective is to assess the ability of soft computing techniques in reconstructing curves and surface for CAGD. Curve that we have used is rational quadratic Bézier. Soft computing techniques optimize parameter control point and weight which consist in the rational quadratic equation. As the two piecewise rational quadratic used, the smoothness between these two curves has been taken into consideration. This work is extended to two surfaces. Four patches of rational quadratic have also been used and as always the control points and weight has been optimized. This means that all type of curves in CAGD can use soft computing techniques.

The third objective is to generate alternative technique of solving fitting problem that are computationally efficient. Apart from direct technique, these methods can be used for solving data fitting problem. This shows that all optimization methods in soft computing can also be applied. We hoped that we can develop a technique that is robust and fast.

1.4 Thesis Outline

This thesis has been organized as follows: Chapter 2 presents the pre-processing stage in solving data fitting problem using soft computing techniques. It contains general algorithm for data representation, corner detection and chord length parameterization. Some example for corner detection algorithm has also been presented.

Then, the curve and surface scheme that is used for data representation has been discussed in Chapter 3. The objective function that has been used in the optimization process is also explained in this chapter. Chapter 4 which deals with Genetic Algorithm procedure and its usage for data fitting. GA has been explained in details and some simple examples have been used to understand GA better. GA has been then used for data fitting and parameter tuning has also been done.

Then, in Chapter 5, we have explained Differential Evolution in details and DE has been used for data representation. The parameter tuning for curve fitting has also been done. The third method which is Particle Swarm Optimization has then been presented in Chapter 6.

The comparative study which is based on these three methods has been done in Chapter 7. The comparison includes visual and numerical comparison. This chapter compares which one is the best method among the proposed three soft computing techniques. The comparison has also been made with the existing techniques. Finally , the conclusion has been given in Chapter 8.

CHAPTER 2

GENERAL ALGORITHM, CORNER DETECTION AND CHORD LENGTH PARAMETERIZATION

This chapter discusses general algorithm and pre-processing stage in solving curve and surface approximation. This section is organized as follows: General algorithm for data approximation is explained in Section 2.1. In Section 2.2, corner detection method is explained. Corner detection is important to detect the edge of the boundary. As parameterization is important for data representation, Section 2.3 explains chord length parameterization.

2.1 General Algorithm for Data Representation

The understanding of the general algorithm gives a brief overview of how the data representation process is done. Our curve representation method starts with the reading of the 2D data points. For curve representation, sets of 3D data point have been read using Matlab. If the raw data is an image, the boundary is extracted using Matlab image processing toolbox. Note that we have used Matlab R2012b for implementation. After the boundary has been extracted, corner detection has been done on the extracted boundary. We have used SAM06 method for corner detection. The details of the algorithm have been presented in Section 2.2. Then, for each segmented boundary, curve fitting has been done using soft computing techniques. The general algorithm is given in Figure 2.1.

For surface representation, first we got 3D data points from a function. Then we have determined the control points p_{00}, p_{02}, p_{04}, p_{20}, p_{22}, p_{24}, p_{40}, p_{42} and p_{44} based on the edge and middle location of the data points(p_{00}, p_{04}, p_{20}, p_{24}, p_{40}, and p_{44} are the endpoint and the rest are middle points). Our soft computing scheme has optimized control points p_{01}, p_{10}, p_{11},

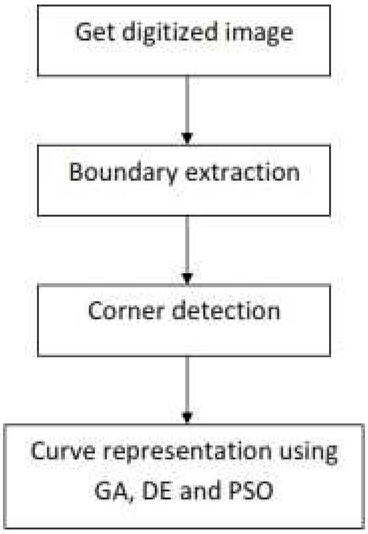

Figure 2.1: General curve representation algorithm

p_{12}, p_{14}, p_{21}, p_{41} and the only weight w_{11}. We have optimized the parameter that gives the smallest error (see Figure 2.2).

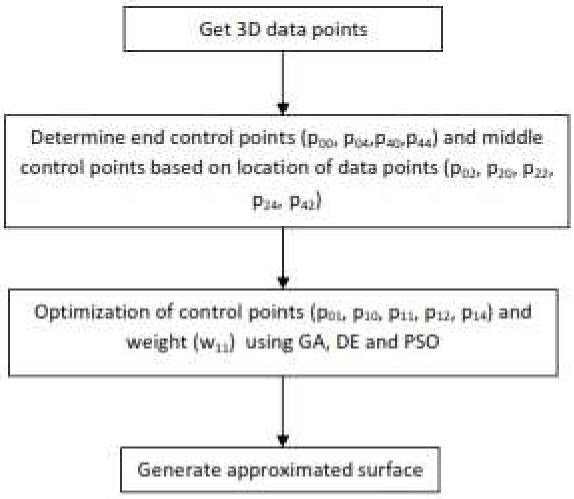

Figure 2.2: General surface representation algorithm

2.2 Corner Detection

In this algorithm, the detection of corner points is based on calculation of distances from the straight line joining two contour points on two sides of that corner. This algorithm is known as SAM06 explained in Sarfraz (2008a).The algorithm is robust, simple to implement, efficient and performs well on noisy shapes as well. The algorithm is divided into two passes. Corner point candidates are detected in the first pass and corner point superfluous candidates are discarded in the second pass.

2.2.1 First Pass

Any contour point, P_j is a corner point candidate if it satisfies two conditions. First, P_j (located between two contour points P_i and P_k) is at maximum perpendicular distance from the straight line joining these two contour points. Second, the maximum perpendicular distance is greater than the given threshold value D. For the contour point P_i where $1 \leq i \leq n$ and n is the number of contour points in a closed loop, the contour point P_k is given as:

$$P_k = \begin{cases} P_{i+L}, & \text{if } (i+L) \leq n, \\ P_{i+L-n}, & \text{otherwise,} \end{cases} \tag{2.1}$$

where L is a length parameter whose default value is 14. The perpendicular distance of all contour points between P_i and P_k is calculated from the straight line joining these contour points. Point P_j is the point with maximum perpendicular distance as shown in Figure 2.3. P_j is selected as a candidate corner point if its perpendicular distance (d_j) from the straight line is greater than the parameter D and the distance d_j is assigned to P_j . The perpendicular distance d_j of point $P_j(x,y)$ from the straight line joining the point $P_i(x,y)$ and $P_k(x,y)$ can be calculated as (Sarfraz, 2008a):

$$d_j = \begin{cases} \left|P_{j,x} - P_{i,x}\right|, & \text{if } m_x = 0, \\ \frac{\left|P_{j,y} - mP_{j,x} + mP_{i,x} - P_{i,y}\right|}{\sqrt{m^2 - 1}}, & \text{otherwise,} \end{cases} \tag{2.2}$$

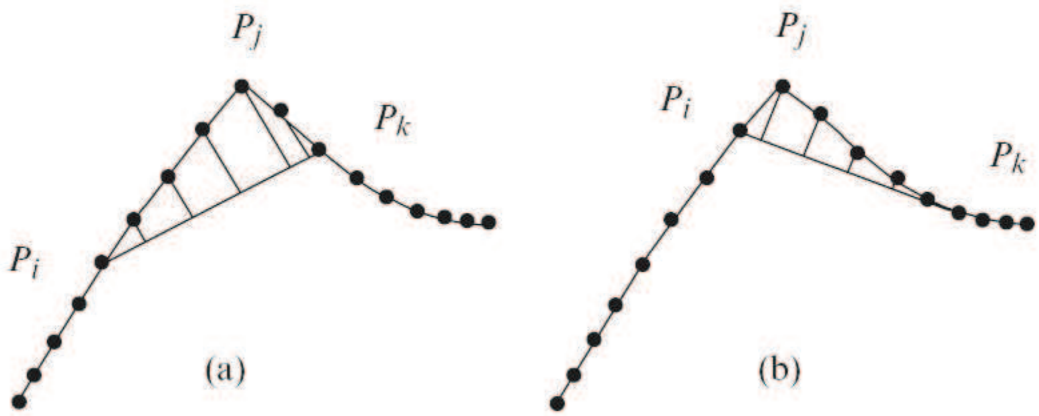

Figure 2.3: The contour point P_j at maximum perpendicular distance from the straight line P_iP_k.

where

$$m = \frac{m_y}{m_x} = \frac{P_{k,y} - P_{i,y}}{P_{k,x} - P_{i,x}}. \tag{2.3}$$

The next candidate corner point is detected for a new straight line by incrementing both i and k. The process continues for $i = 1$ to n. For one straight line, there can be only one candidate corner point or no candidate corner point at all. More than one straight line may respond to the same point P_j as shown in Figure 2.3(a) and 2.3(b). In this case, the higher value of d_j is assigned to P_j.

2.2.2 Second Pass

Sometimes the corners to be detected are not the sharp angle points and we may detect superfluous candidate corner points in the first pass, as shown in Figure 2.4. These superfluous points have been discarded in the second pass. The candidate corner point is superfluous if any other candidate with higher value of d_j is in the range R. The default value of parameter R is equal to parameter L. Therefore for any candidate point to be selected as a corner point, it must have its highest value of d_j among the R number of points on both sides. Three different points have been detected as candidate corner points in Figure 2.4(a), 2.4(b) and 2.4(c). P_j of Figure

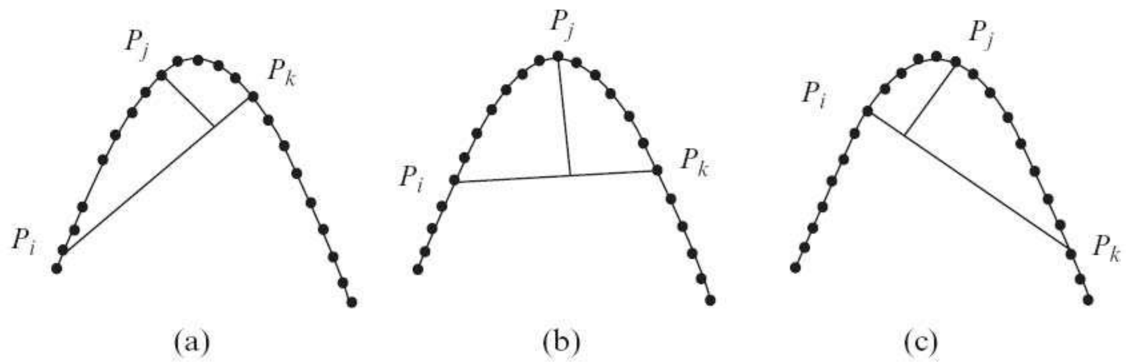

Figure 2.4: Superfluous candidate corner point P_j in (a) and (c).

2.4(a) and 2.4(c) are discarded as P_j of Figure 2.4(b) has higher d_j , which is in the range R.

2.2.3 Parameters

The algorithm needs three external parameters L, D and R, as given above. The length of the straight line P_iP_k is fixed as per length parameter L throughout the corner detection process. Thus, the straight line always joins the two contour points, L points apart. The default value of L is 14. This parameter takes care of object scaling and resolution. The default value assigned to L suits the size of all test shapes demonstrated in Section 2.2.1. Corners are the high curvature points which are recognized by their local sharpness and opening angle. It used the distance parameter D as a substitute for the sharpness and opening angle, to check their validity as a corner point.

Any point whose distance from the straight line P_iP_k goes beyond parameter D can be selected as a valid corner point. The default value of D is 2.6. This is an important parameter to control false selection of corners due to noise and other irregularities in a curve. Higher values of D may miss some valid corners and lower values may hit the wrong corners as well. For noisy shapes, accurate corners can be detected by adjusting this parameter (see Figure 2.3(a)). Sometimes the local sharpness of a corner is not high enough, but a global view of shape identifies it as a valid corner (Figure 2.4). Such corners are also detected successfully with this method at the cost of some additional invalid (superfluous) corners. These invalid corners have been removed in the second pass by fixing the domination range R. Only the most dominant corner (with highest d_j) in the range R is selected as a valid corner and all others have been discarded. The default value of R is equal to L but it must be given lower value to enable detection of closely located corners.

2.2.4 Demonstration

In this section SAM06 corner detection method has been used on several images, given in Figure 2.5.

2.3 Chord Length Parameterization

If an interpolating curve follows very closely to the data polygon, the length of the curve segment between two adjacent data points would be very close to the length of the chord of these two data points. The length of the interpolating curve would also be very close to the total length of the data polygon (Shene, 2010). In Figure 2.6, each curve segment of an interpolating polynomial is very close to the length of its supporting chord, and the length of the curve is close to the length of the data polygon. Therefore, if the domain is subdivided according to the distribution of the chord lengths, the parameters will be an approximation of the arc-length parameterization.

Suppose the data points are D_0, D_1, ..., D_n. The length between D_{i-1} and D_i is $|D_i - D_{i-1}|$, and the length of the data polygon is the sum of the lengths of these chords:

$$L = \sum_{i=1}^{n} |D_i - D_{i-1}|$$

Therefore, the ratio of the chord length from data point D_0 to data point D_k, denoted as L_k, over

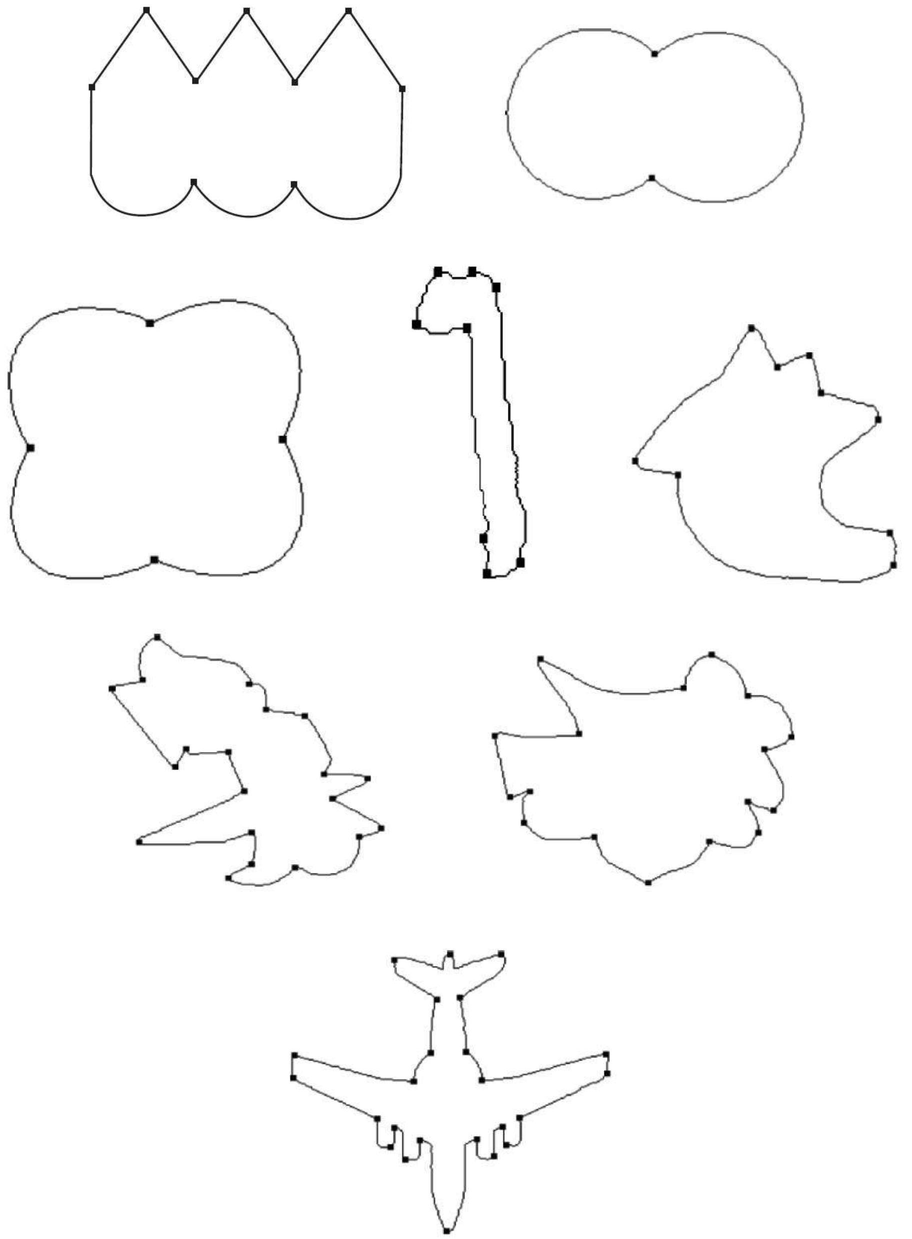

Figure 2.5: Corner detection examples using SAM06

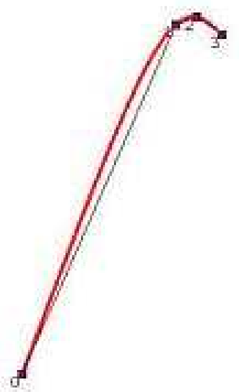

Figure 2.6: Chord length parameterization

the length of the data polygon is

$$L_k = \frac{\sum_{i=1}^{k} |D_i - D_{i-1}|}{L}$$

If we prefer to have an arc-length parameterization of the interpolating curve, the domain has to be divided according to the ratio L_k. More precisely, if the domain is $[0,1]$, then parameter t_k should be located at the value of L_k:

$$\begin{aligned}
t_0 &= 0 \\
t_k &= \frac{1}{L}\left(\sum_{i=1}^{k} |D_i - D_{i-1}|\right) \quad k = 1,2,...,n-1 \\
t_n &= 1
\end{aligned}$$

where L is the length of the data polygon. In this way, the parameters divide the domain into the ratio of the chord lengths.

As an example, suppose we have four data points ($n = 3$): $D_0 =< 0,0 >$, $D_1 =< 1,2 >$,

$D_2 =< 3,4 >$ and $D_3 =< 4,0 >$. The length of each chord is

$$|D_1 - D_0| = \sqrt{5} = 2.236,$$
$$|D_2 - D_1| = 2\sqrt{2} = 2.828,$$
$$|D_3 - D_2| = \sqrt{17} = 4.123$$

and the total length is

$$L = \sqrt{5} + 2\sqrt{2} + \sqrt{17} = 9.8176$$

Finally, we have the corresponding parameters:

$$\begin{aligned}
t_0 &= 0, \\
t_1 &= \frac{|D_1 - D_0|}{L} = 0.2434, \\
t_2 &= \frac{|D_1 - D_0| + |D_2 - D_1|}{L} = 0.5512, \\
t_3 &= 1.
\end{aligned}$$

Figures 2.7a and 2.7b show the data points and the parameter distributions of the uniformly spaced and chord length methods.

The chord length method is widely used and usually performs well. Since it is known that polynomial curves cannot be parameterized to have unit speed (i.e., arc-length parameterization), the chord length can only be an approximation. Sometimes, a longer chord may cause its curve segment to have a projection bigger than necessary. In Figure 2.8, the black and blue curves both interpolate 7 data points. As you can see, both curves have very similar shape, except for the last segment, and the one computed with chord length method wiggles a little.

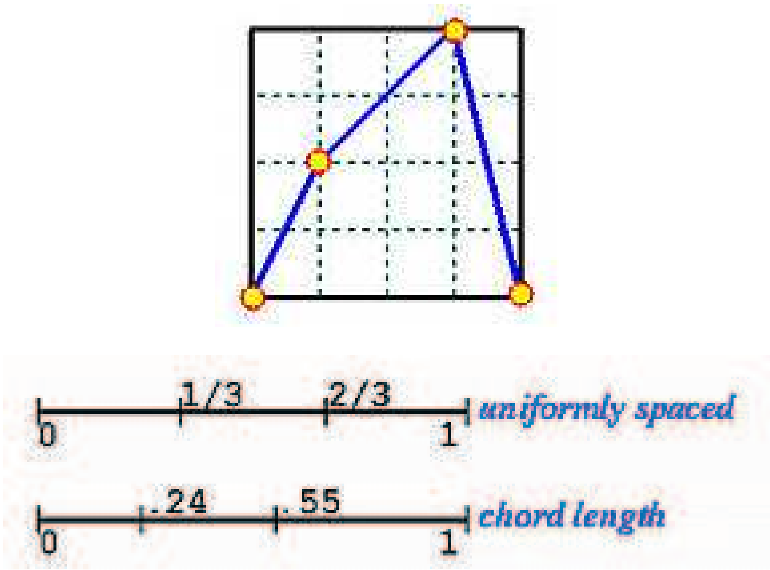

Figure 2.7: A uniformly spaced vs chord length parameterization

The last curve segments are very different and the curve using the chord length method has a large bulge and twists away from the black curve produced by the uniformly spaced method. This is a commonly seen problem with the chord length method.

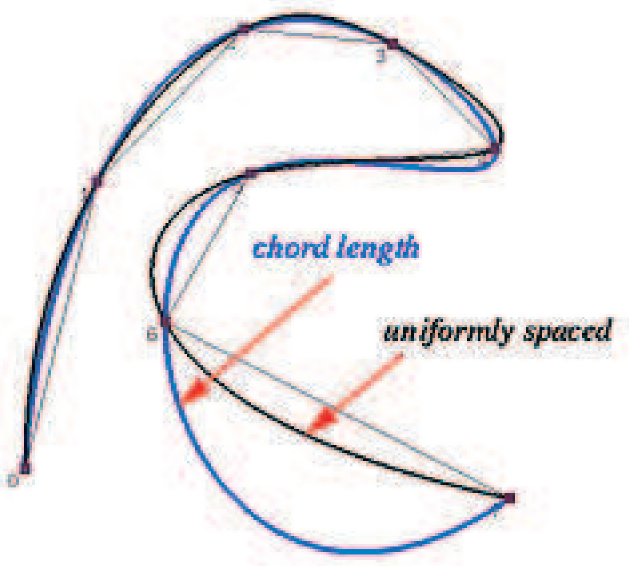

Figure 2.8: Uniform spaced vs chord length parameterization curves

CHAPTER 3

DATA FITTING WITH RATIONAL QUADRATIC BÉZIER

This chapter discusses the type of curve and surface that we want to use for data representation. It also discusses how the data points are fitted using the presented curve and surface scheme. This section is organized as follows: in Section 3.1, conics curve is presented. This type of curve is used for curve representation. The minimization of sum square error function is also explained in the same section. Blending bi-quadratic rational Bézier surface is explained in Section 3.2. Finally, the objective function of optimizing surface fitting will be presented in the same section.

3.1 Conic Curve

A standard form of conics is given by Yang (2003) and Farin (1989):

$$r(t) = \frac{B_0^2(t)b_0 + wB_1^2(t)b_1 + B_2^2(t)b_2}{B_0^2(t) + wB_1^2(t) + B_2^2(t)}, t \in [0,1] \tag{3.1}$$

where $B_i^2 = \binom{2}{i} t^i (1-t)^{2-i}$ are basis functions and $b_i (i = 0,1,2)$ are control points and w is the middle weight. Here we have listed some useful properties of conics:

1. For $0 < w < 1$, we obtain an ellipse; $w = 1$, a parabola; and $w > 1$, a hyperbola.
2. The straight line segments $[b_0, b_1]$ and $[b_1, b_2]$ are tangents directions to r at $r(0) = b_0$ and $r(1) = b_2$, respectively.
3. When $w \geq 0$, the curve segment (3.1) lies in the convex hull of the control polygon.
4. The point $s = r(1/2)$ of a conic segment in its standard form is called the shoulder point.

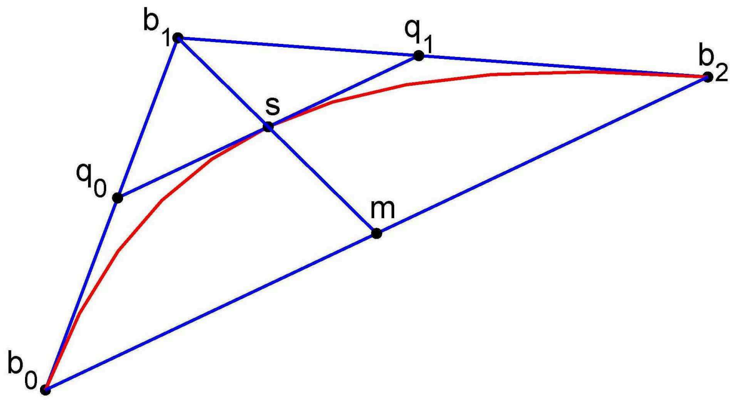

Figure 3.1: A rational quadratic curve with shoulder point **s** and tangent through q_0 and q_1

It can be computed from

$$s = \frac{1}{2}q_0 + \frac{1}{2}q_1,$$

where

$$q_0 = \frac{b_0 + wb_1}{1+w},\ q_1 = \frac{wb_1 + b_2}{1+w},$$

are the characteristic points. The tangent is spanned by q_0 and q_1. Note that shoulder tangent is parallel to $[b_0, b_2]$; see Figure 3.1. As a consequence,

$$w = \frac{||s-m||}{||b_1 - s||}$$

where m is the midpoint of b_0 and b_2.

5. The curvature κ of r at the endpoints is given by:

$$\kappa(0) = \frac{\tau}{w^2\rho^2},\ \kappa(1) = \frac{\tau}{w^2\lambda^2} \tag{3.2}$$

where τ denotes the area of the triangle formed by the control polygon; i. e. $\tau = \frac{1}{2}det(b_1 - b_0, b_2 - b_1)$, $\rho = ||b_1 - b_0||$ and $\lambda = ||b_2 - b_1||$. Note that κ denotes the signed curvature, since τ may be positive or negative.