2010

Curve reconstruction: Experimental comparison and certification

Chong Wang
University of Windsor

Curve Reconstruction:

Experimental Comparison and Certification

by

Chong Wang

A Thesis
Submitted to the Faculty of Graduate Studies
through Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2010

Your file Votre référence
ISBN: 978-0-494-62751-8
Our file Notre référence
ISBN: 978-0-494-62751-8

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

This thesis also incorporates the outcome of joint research under the supervision of Dr. Asish Mukhopadhyay. The collaboration is covered in Chapter 3 and Chapter 4 of the thesis. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of co-authors was primarily through the provision of some key ideas and constructive criticism.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes one original paper that has been previously published/submitted for publication in peer reviewed journals, as follows:

Thesis Chapter	Publication title/full citation	Publication status*
Chapter 3, 4	Certifying curve-reconstruction algorithms, The 26th European Workshop on Computational Geometry (EuroCG'10)	published

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

This thesis work deals with curve reconstruction problem, which contains two parts: experimental comparison and certification.

In the first part we establish the effectiveness of the simple RNG-based algorithm by experimental comparisons with two leading algorithms: the NN-crust and the Conservative-crust. By comparing the outputs of these three algorithms on different samples of increasing complexity, we demonstrate that the RNG-based algorithm performs as well or better.

Since there is no way to verify that a given sample from some unknown curve satisfies the sampling condition, in the second part of this thesis we propose a novel approach that bypasses this problem by certifying to the accuracy of the reconstruction. We smooth the polygonal output of a reconstruction algorithm and sample the smoothed curve. The closeness of the original sample set and resampled set is an indication of the accuracy of the curve-reconstruction algorithm.

Keywords: curve-reconstruction, comparison, certification, resampling.

Dedicated to my parents and I love you forever.

Acknowledgements

First and most important, I would like to take this opportunity to express my appreciation to my advisor Dr. Asish Mukhopadhyay. This thesis work could not have been possible to complete without his continuous support and insightful guidance. I am very thankful for his valuable advice and help in both research work and daily life of my master's degree, which is the most important experience in my life. I would also like to thank my committee members, Dr. Dan Wu and Dr. Myron Hlynka for spending their precious time reading my thesis and giving their valuable suggestions on this work.

I am also grateful for the support and advice from my colleague Md. Shaiful Alam. I can not finish my thesis work successfully without his help.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Research on the Curve Reconstruction problem has been conducted for almost 30 years. Given a point set which is from a smooth open and closed curve, the **Curve Reconstruction problem** is to compute the polygonal reconstruction graph, where this point set is the vertex set of the graph and each of the edges exactly connects adjacent sample points on the original curve.

Figure 1 displays a simple scenario for the curve reconstruction problem. A set of samples are given from an original curve and the polygonal graph is produced to describe the shape by connecting the samples (See Fig. 1).

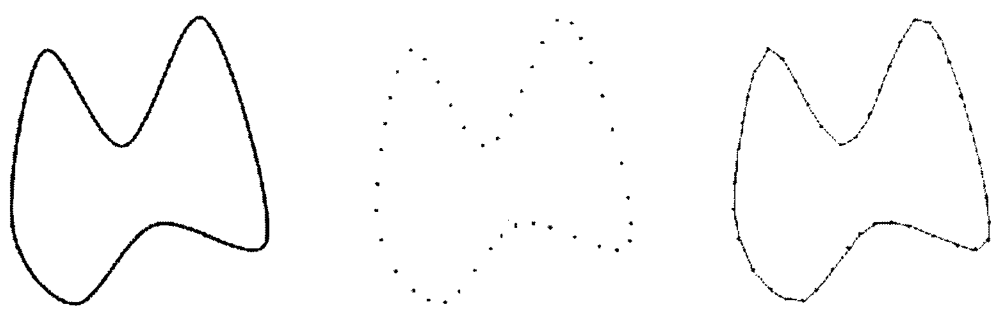


Figure 1: The original curve, sample points and the reconstructed curve

Due to its wide applications in image processing, pattern recognition and computer vision, it has received a lot of attention from researchers over the last decade. Another reason for this attention is that it provides a basis for solving the more difficult problem of surface reconstruction in three dimensions.

1.2 Classification

There are two main types of approaches for Two-Dimensional Curve Reconstruction:

- Voronoi Diagram & Delaunay Triangulation based approaches.
- Non-Voronoi Diagram & Delaunay Triangulation based approaches.

In this thesis, we put an emphasis on approaches based on the Voronoi Diagram & Delaunay Triangulation technique.

In recent years, with the fast development of 3D techniques, people have become more interested in developing Three-Dimensional Surface Reconstruction applications. Although Curve Reconstruction is restricted to two dimensions, the research in two dimensions establishes a theoretical basis for work on the surface reconstruction problem in three dimensions.

1.3 Application areas

Three typical applications of the Curve Reconstruction problem are shown below:

- Image processing.
- Pattern recognition.
- Computer vision.

More importantly, some applications of Three-Dimensional Surface Reconstruction are based on the theory of Two-Dimensional Curve Reconstruction.

There are two applications to 3D surfaces of the information obtained from a set of planar contours:

In Biology: biologists try to understand the shape of microscopic objects from serial sections through the object.

In Computer Aided Design (CAD): lofting techniques specify the geometry of an object by means of a set of contours.

1.4 Terminology

1.4.1 Delaunay Triangulation & Voronoi Diagram

In computational geometry, a Delaunay Triangulation (DT) of a sample set in two dimensions is a triangulation, where the circumcircle of any triangle does not contain any other point of the sample set (See Fig. 2).

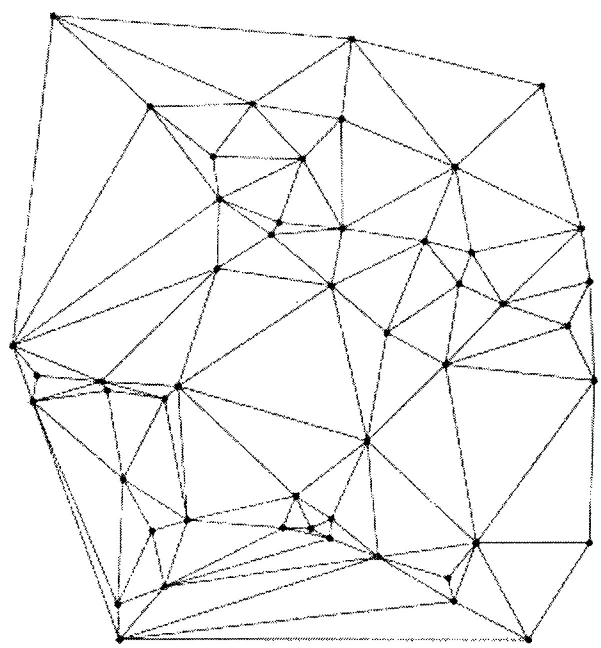


Figure 2: Delaunay Triangulation of a set of points

Voronoi Diagram (VD), as the dual graph of the Delaunay Triangulation, is also used in various areas. The Voronoi Diagram of n points in two dimensional space divides the plane into a set of convex regions. In each convex region, there is exactly one generating point. Every point in one region is closer to its generating point than to any other generating point in the plane (See Fig. 3).

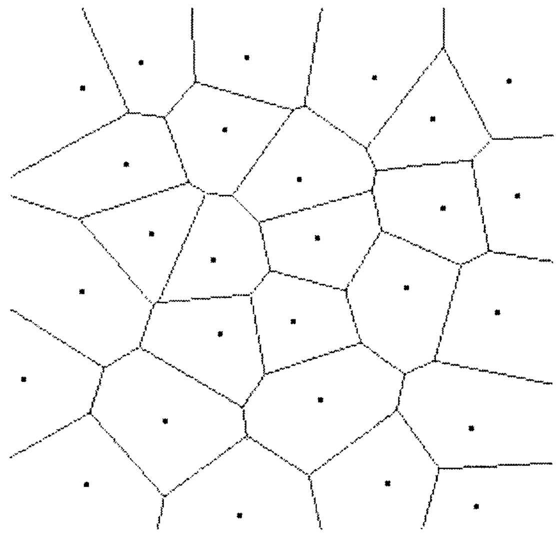


Figure 3: Voronoi Diagram on a set of points

Both the Delaunay Triangulation and Voronoi Diagram are very significant data structures in the Curve Reconstruction area. It is straightforward to obtain one from the other (See Fig. 4).

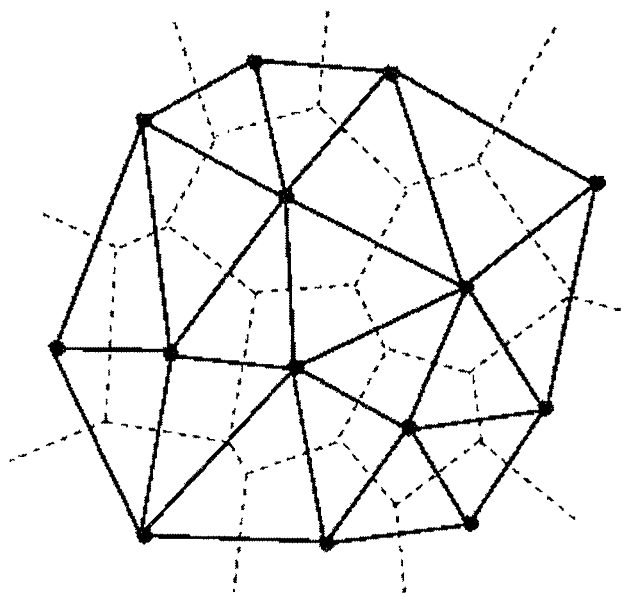


Figure 4: Transformation of DT and VD

From Delaunay Triangulation, connecting the centers of the circumcircles of all the triangles produces the corresponding Voronoi Diagram. From Voronoi Diagram, connecting the generating points of two convex regions that share a common edge produces the corresponding Delaunay Triangulation.

1.4.2 Medial Axis & Local Feature Size

The concept of the **medial axis**, introduced by Blum [8], is an important tool for Curve Reconstruction. It approximates the shape of a curve.

Amenta et al. [3] introduced the concept of the **local feature size**, which determines the sampling density in the neighbourhood of a sample point.

The **medial axis** of a smooth curve is the locus of the centers of the topological 1-disks which touch more than one point of the curve (See Fig. 5). Every point on the medial axis has two or more closest points on the curve.

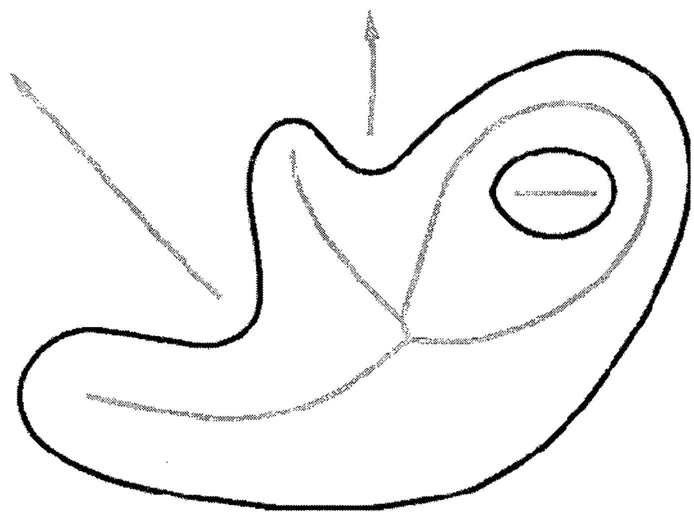


Figure 5: Medial axis of a curve

The **local feature size** of one point on the curve is the distance between this point and the nearest point on the medial axis (See Fig. 6).

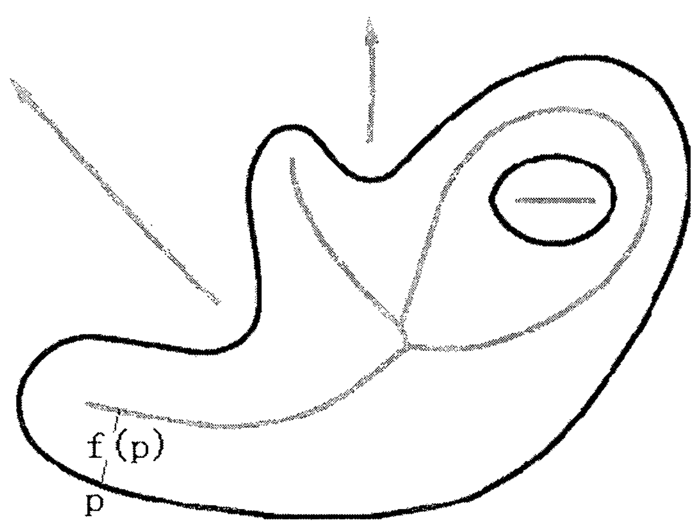


Figure 6: Local feature size f(p) at point p

1.4.3 Sampling conditions (ε-sample)

In order to guarantee the correctness of reconstruction, a sampling condition is necessary. Some algorithms for curve reconstruction require uniform sampling while some other can allow non-uniform sampling. Here we introduce the notion of an ϵ–sample which is non-uniform sampling.

An ϵ–sample is defined as follows: for any point p on the curve, the distance to its nearest sample point is at most $\epsilon * f(p)$, where f(p) is the local feature size of point p (See Fig. 7).

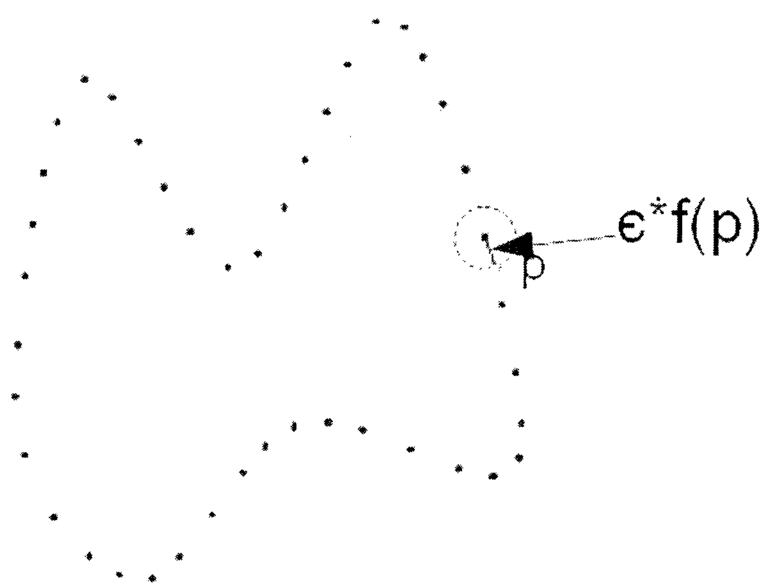


Figure 7: Sample points of Curve showing variable sampling density

Based on this non-uniform sampling, we need more samples at some parts with more details (like sharp corners, intersections etc.) and fewer samples at other parts with less detail.

1.4.4 Relative neighbourhood graph (RNG)

The Relative Neighbourhood Graph (RNG) is also an important data structure in the

Curve Reconstruction area. For a set of n distinct points on the plane, the edge pq, where p and q are from the point set, is defined as an RNG edge if and only if these two points p and q are relatively close (See Fig. 8).

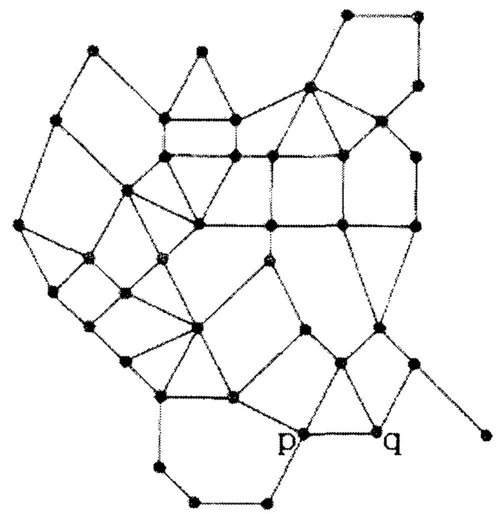


Figure 8: Relative neighbourhood graph of points

Lankford [32] defines two points p and q as being relatively close if $d(p, q) \leqslant \max[d(p, r), d(q, r)]$, where r is one of n points and $r \neq p, q$. In addition, it is shown that RNG is a subgraph of the Delaunay Triangulation on the same point set.

1.4.5 Gabriel graph

The Gabriel graph is another important data structure in the Curve Reconstruction area. For a set of points on the plane, the edge pq is defined as a Gabriel edge if and only if the diametral circle on this edge dose not contain any other points from the point set (See Fig. 9).

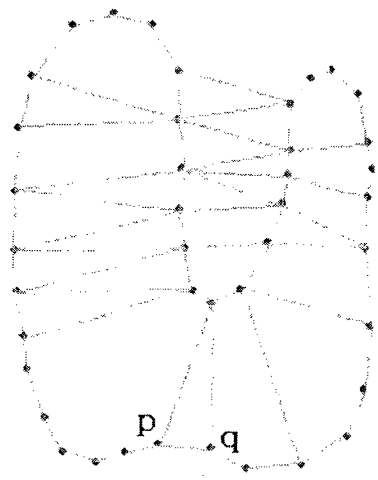
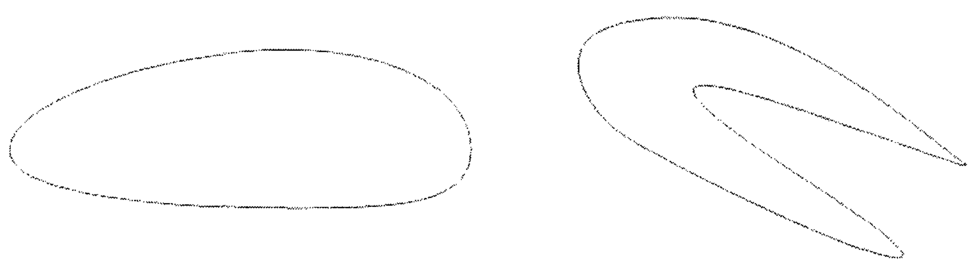


Figure 9: Gabriel graph of points

In addition, it is shown that the Gabriel graph is a subgraph of the Delaunay Triangulation and a supergraph of the RNG on the same point set.

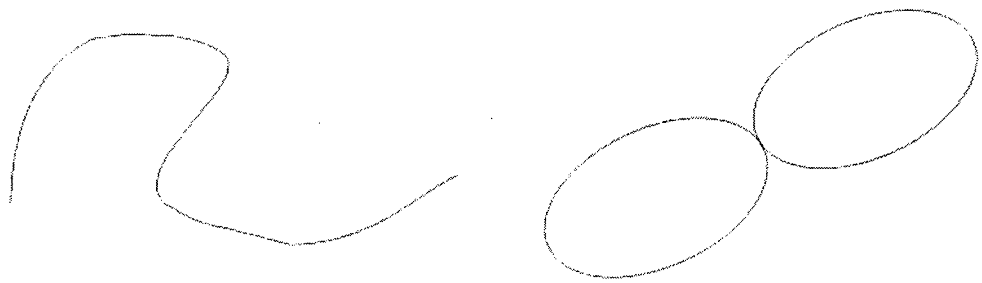
1.4.6 Types of curves

Different curve reconstruction algorithms could handle different types of curves. Here we show four kinds of typical curves (See Fig. 10).



(a) Smooth closed curve

(b) Curve with sharp corner



(c) Curve with endpoints (d) Curve with intersection

Figure 10: Types of curves

Smooth curves satisfy one condition: for any points on the curve, it has the same left and right tangent.

Curves with corners, also called non-smooth curves, satisfy one condition: the point in the corner has different left and right tangents. In this situation, the medial axis will go through the corners such that the areas near corners need an infinite number of points to satisfy the sampling density.

Curves with endpoints are called open curves.

Curves with intersections are also called self-intersecting curves.

Chapter 2

Literature Review

2.1 Voronoi Diagram & Delaunay Triangulation Based Approaches

Many approaches to the curve reconstruction problem compute the Voronoi diagram and Delaunay triangulation of the given point set as their first step since it is shown that the polygonal reconstruction is the subgraph of the Delaunay triangulation when point set is dense enough. Then some heuristic is used to make a local test of all the Delaunay edges and to check which edges are redundant.

In this thesis, we mainly discuss the Voronoi Diagram & Delaunay Triangulation based approaches, which have been sorted into 4 groups as follows: the first group handles smooth curves, the second group handles curves with corners, the third group handles curves with noisy samples, and the fourth group handles curves with multiple features.

2.1.1 Approaches for Smooth Curves

Smooth curves satisfy one condition: the left and right tangents of every point in the curve are the same.

Five papers in this subsection provide approaches for smooth curves. The first paper was written in 1997 and the last in 2006.

Amenta et al. [3] presented a method for obtaining the reconstruction of a smooth curve from a non-uniform sample. Earlier Kirkpatrick and Radke [31] defined the concept of a β–skeleton, where the β value must be found. Amenta et al. [3] do not identify any shortcomings of previous work, but they stated that they made the β–skeleton approach work by giving a proper value to the parameter β as long as the sample density is satisfied. In addition, they made two contributions: One is proposing an approach to find the crust graph. The other one is that they introduce the concept of local feature size such that the density in the neighbourhood of a sample point can be determined. With the introduction of a non-uniform sampling condition, three other algorithms by Dey and Kumar [13], Gold [21] and Dey et al. [14] were proposed to handle for smooth curves with different features.

Dey and Kumar [13] proposed an approach that is able to handle smooth closed curves in higher dimensions. The authors in this paper did not identify any shortcomings of Amenta et al. [3], but they improved the sampling density from 0.252 in Amenta et al. [3] to 1/3.

Gold [21] appears to be the first to present a one-step algorithm to find two graphs called the crust and the skeleton, which are used to deal with map input problems. The author refers to the approach by Gold et al. [22], which is able to obtain a skeleton from the Voronoi diagram. But it has two drawbacks: first, labelled points are needed; secondly, this approach is restricted to polygon map problems. Although Gold [20] directly generated topologically correct maps when the quality of scanned input is sufficiently high, labelled points are still required. Thereafter, he developed a new method to obtain the skeleton from unlabelled points, which is based on the approach proposed by Amenta et al. [3].

Dey et al. [14] refer to algorithm CRUST by Amenta et al. [3] and algorithm NN-CRUST by Dey and Kumar [13], but neither of them is able to handle an ϵ–sample from an open curve. Hence, they developed the algorithm CONSERVATIVE-CRUST (P, ρ), where ρ is a non-negative real parameter, that could deal with either closed curves or open curves as long as the sample is sufficiently dense.

Hiyoshi [23] refers to the work by O'Rourke et al. [41] and he identifies one shortcoming for their work: the method based on minimal spanning Voronoi tree (MSVT) is not able to handle the curve with multiple connected components. Then the author in this paper presents an approach called ZERO-ONE which is able to reconstruct a set of sample set from closed curves. It is claimed that every edge of polygonal reconstruction exactly connects adjacent points on the original curve. Under the assumption that the smooth closed curve consists of many separate pieces, the cases of circular arcs and line segments can be ignored. Therefore, the author derives a linear function to handle the minimization problem, which can be directly transferred to zero-one programming problem. On the other hand, because the running time of ZERO-ONE algorithm is quite high, another two heuristics, SCISSORS and PASTE, are proposed to reduce the complexity. As a result, SCISSORS and PASTE work well for large input cases and polygonal reconstruction is correctly produced as the sample density is sufficient. In addition, the author presents the comparison of outputs of ZERO-ONE, SCISSORS, PASTE, the crust and β-skeleton algorithms performed on Computational Geometry Algorithm Library. For the samples from letter U, both ZERO-ONE and PASTE are successful to obtain the reconstruction, but there are several errors in the crust, SCISSOR and β–skeleton algorithms. For the samples from letters ABC, all the algorithms could not output the correct polygonal reconstruction as the sampling density is not sufficient. It is claimed that the time complexity of either SCISSORS or PASTE algorithm is O(n log n) and ZERO-ONE needs the smallest sample density among all the proposed algorithms. Moreover, a slightly modified SCISSORS method is still able to handle open curves.

2.1.2 Approaches for Curves with Corners

Curves with corners, also called non-smooth curves, satisfy one condition: the left and right tangents of the points near or in the corners are different. In this situation, the medial axis will go through the corners, which means that the areas near corners need infinite points to satisfy the sample density. As a result, the approaches for smooth curves presented in section 2.2.1 could not work for curves with sharp corners. It has been shown that they not only fail theoretically but also in practice.

Four papers in this subsection provide approaches for curves with corners. The first paper was written in 1999 and the last in 2001.

Giesen [18] appears to be the first to present an approach to reconstruct a single curve with sharp corners. For most Voronoi diagram and Delaunay triangulation based algorithms, the sampling density is controlled by the parameter **ε** such that the distance from one point p on the original curve to its nearest sample point is at most ε*f(p), where f(p) is the local feature size of point p. It is shown that those approaches based on that sample condition are able to handle smooth open curves or smooth closed curves. However, their approaches have one common drawback: it may fail to deal with the curves with sharp corners. Because the corner points are on the medial axis, the parts close to the corners of curves need infinite samples. Therefore, Giesen proposes one method to overcome above drawback. His algorithm is derived from Travelling Salesman problem and the author uses two corollaries of Menger's theorem to further develop the local property to the global such that a shortest polygonal graph, which connects all the points by order, represents the curve reconstruction. In addition, the necessary and sufficient sample conditions are defined to guarantee that the travelling salesman path (TSP) algorithm is able to obtain the correct reconstruction.

Althaus and Melhorn [1] indicate that they use Giesen's theory [18] as a basis for their work. However, there are two shortcomings in Giesen's work as follows: first, the travelling salesman tour or path algorithm could only handle a uniformly distributed point set from smooth closed curves, smooth open curves or semi regular curves, secondly, the sample density is not explicitly defined for different types of curves. Hence, the authors in this paper further develop Giesen's work in three directions: first, different sampling densities are given respectively according to the types of curves. It is found that smooth curves have similar sampling condition to that in the works by Amenta et al. [3], Dey and Kumar [13], Gold [21] and Dey et al. [14].

In addition, the travelling salesman tour or path approach is successful to reconstruct the curves when it is a non-uniformly distributed sample set; secondly, when the sample density is sufficiently high, the computation time of travelling salesman tour or path is polynomial; thirdly, based on some powerful theorems and reasonable assumptions, the travelling salesman tour or path is given a guarantee for the correctness of curve reconstruction.

Although the above algorithm was proposed to work for curves with corners, it could only handle single, closed curves. Dey and Wenger [15] present an approach called GATHAN that is able to reconstruct a collection of curves with corners. Their approach is derived from NN-CRUST by Dey and Kumar [13]. However, the previous sample density condition does not satisfy with non-smooth curves since the parts close to the corners need infinite samples, so Dey and Kumar introduce the angle condition, ratio condition and topological condition at the same time such that the appearance of incorrect or redundant edges will be avoided. The authors state that GATHAN algorithm handles curves with sharp corners, boundary points and multiple components quite effectively in comparison with the other algorithms. In addition, all steps in their algorithm could be extended to three dimensions. However, the unsolved problem is that they can not provide a proof for the correctness of their algorithm.

Subsequently, it was found that the algorithm GATHAN by Dey and Wenger [15] may fail for some special cases. In 2001, Funke and Ramos [17] propose an approach that is able to obtain the polygonal reconstruction of a set of samples from the curves which have corners and endpoints. The authors indicate that they use the same ideas as Dey and Wenger [15], which is first detecting smooth places and then probing sharp corners or endpoints. However, they have different conditions that decide whether the corners are found. By modifying their approach slightly, they could also produce the correct reconstruction of the sample set from curves with multiple components as long as the sample density is sufficient. In addition, the authors state that, with the theoretical parameters for sampling conditions, their algorithm gives better output than both the CRUST and CONSERVATIVE-CRUST approaches.

2.1.3 Approaches for Curve Reconstruction from Noisy Samples

All of the approaches presented in the other subsections can only handle the curves from a noise-free sample, but no heuristic is proposed to effectively deal with curves when noisy samples occur. Noise usually comes from input sample set. For instance, scanning an image may produce some noisy points. Typically, there are two different types of noisy samples. However, researchers often make an assumption that all the noisy samples, which are discussed in this subsection, are uniformly distributed around every actual point.

Two papers in this subsection provide approaches for Curve Reconstruction from Noisy Samples. The first paper was written in 2003 and the last in 2007.

Cheng et al. [12] appear to be the first to propose an approach that is able to construct the polygonal reconstruction of smooth closed curves with multiple components from noisy samples. The authors refer to many previous approaches by

Amenta et al. [3], Dey and Kumar [13] and Dey et al. [14], and it is shown that none of all the existing algorithms could handle curves from the noisy samples. The authors in this paper claim that they use method coarse() to put the samples from a relatively smooth area together as a group. The main idea of this algorithm is generally based on three steps as follows: first, drawing a circle based on every sample point such that the neighbourhood coarse() decides a strip which is relatively narrow to the neighbourhood size, secondly, using function refined() to delete all the noisy samples in a certain neighbourhood which is defined from above, thirdly, taking any existed approach for curve reconstruction, like NN-CRUST, to perform on noisy-free samples obtained from step2, and then the polygonal reconstruction is produced. In addition, the authors provide a proof for the correctness of their algorithm.

Subsequently, Mukhopadhyay and Das [38] also present an approach which is able to handle curves from a noisy sample. The authors refer to RNG-based heuristic proposed by themselves in 2006, which works well for curves with multiple features. However, it has a shortcoming that the approach will fail when there exists noisy samples in the given sample set. The authors indicate that their algorithm is based on the RNG heuristic that they had presented earlier. First, the useful samples are extracted by a filtering way; secondly, RNG heuristic is used to perform on the noisy-free samples to obtain polygonal reconstruction. In addition, the authors present five groups of the normal sample set and its corresponding noisy sample set from closed curve, curve with endpoints, curves with sharp corners, nested curves and multiple curves respectively. By performing their CRWN algorithm on the noisy sample set, the filtered point set is correctly shown. However, they indicate that the correctness of the output of the CRWN algorithm has not been supported by theoretical guarantees.

2.1.4 Approaches for Curves with Multiple Features

Three papers in this subsection provide approaches for Curves with Multiple Features. The first paper was written in 2006 and the last in 2007. The approaches presented in subsection 2.1.1 and 2.1.2 can only handle curves with one or two features. Since 2006, researchers have started to find approaches for curves with multiple features.

Lenz [34] extends the NN-CRUST algorithm [13]. By improving the sampling density from 1/3 to 0.48, the modified algorithm is able to handle the sample set from smooth closed curves. In addition, the author further develops an approach such that it could deal with many types of curves, including open curves, closed curves, smooth curves, curves with sharp corners and curves with intersections. Given a particular figure from form $L(t) \rightarrow (\sin 4\pi t, \cos 6\pi t)$, the author presents the comparison of three outputs of the reconstructions based on three different numbers of randomly distributed samples. The author states that the number of randomly distributed samples affects the output of the reconstruction. However, their algorithm has one shortcoming: when a single incorrect edge occurs, many subsequent edges may fail. Moreover, it is shown that the bad results could not happen as long as the sampling density is sufficiently high.

Mukhopadhyay and Das [37] state that the approaches by Dey and Kumar [13] and Dey et al. [14] are able to handle simple curves only. And the approaches by Giesen [18] and Dey and Wenger [15] can only handle curves with corners. The authors claim that the shortcoming of previous work is that each existing approach can not deal with curves with multiple features. Hence, they propose a new curve reconstruction algorithm which is an RNG-based heuristic. The first step is to construct a Relative Neighbourhood Graph (RNG) on the input sample set, which is a subgraph of the Delaunay triangulation. They show that the RNG contains all edges joining adjacent points on the original curve when the sampling density is less than 1/5. The second step is to remove redundant edges by a heuristic. Because the approximation of the medial axis of a Voronoi diagram is often crossed by non-adjacent edges, the heuristic computes the maximum distance between one endpoint and the Voronoi vertex of its Voronoi polygon to estimate the distance between this point and the medial axis. The authors state that the algorithm complexity is O(n log n), and this algorithm is able to handle simple curves, nested curves, curves with sharp corners, a set of curves as well as the curves which have end points. In addition, they claim that the sampling density at a normal smooth curve should be lower than that at the sharp corners.

Zeng et al. [49] state that CRUST [3], NN-CRUST [13] and CONSERVATIVE-CRUST [14] need a parameter to control the sampling density. However, a single fixed sample density is hard to satisfy for all different features of the curve, like endpoints and corners. Hence, they present a parameter free, human visual system based algorithm called DISCUR. It is shown that their approach is able to handle a sample set from curves with multiple features, including open curves, closed curves and curves with endpoints or sharp corners. This approach has two main advantages. On one hand, the parameter free algorithm is easier to handle for those curves with many different features. On the other hand, this algorithm is more suitable for unknown curves. Based on the human visual system, adjacent points on the original curve could be always correctly connected.

Chapter 3

Experimental Comparison

Mukhpadhyay and Das [37] proposed a simple but effective algorithm for curve reconstruction by constructing a Relative Neighbourhood Graph on the sample and then pruning non-curve adjacent edges. In the first part of this thesis we establish the effectiveness of this algorithm by experimental comparisons with two leading algorithms: the NN-crust [13] and the Conservative-crust [14]. By comparing the outputs of these three algorithms on a variety of different samples of increasing complexity, we demonstrate that the simple RNG-based algorithm performs as well or better.

3.1 Three Main Algorithms

3.1.1 RNG-based Algorithm

The RNG-based algorithm [37] starts with the Delaunay triangulation on the sample points and retains only the RNG-edges. This is shown to be a supergraph of the graph obtained by joining curve-adjacent pairs. The redundant edges (that is, edges joining pairs of points that are not adjacent on the unknown curve) are removed by a simple and effective heuristic.

It goes as follows. We construct the Voronoi diagram on the given samples. For each RNG edge $p_i p_j$, we compute the maximum distance from p_i to the vertices of its Voronoi polygon. Let this distance be d_i; do the same for the point p_j, obtaining a distance d_j. We delete the edge $p_i p_j$ if its length is greater than the maximum of d_i and

d_j.

Here is a formal description of the algorithm.

Algorithm curveReconstruction

Input: A set of sample points $\boldsymbol{S}$ from an unknown smooth curve $\boldsymbol{C}$

Output: A polygonal reconstruction of $\boldsymbol{C}$

Step 1. Compute the Delaunay Triangulation, DT on $\boldsymbol{S}$.

Step 2. Extract the RNG from the DT.

Step 3. Compute the Voronoi Diagram, VD, as the dual of the DT obtained in Step 1.

Step 4. For each edge $p_i\, p_j$ of the RNG computed in Step 2 do:

Step 4.1 Compute the maximum distance d_i from p_i to the vertices of its Voronoi polygon.

Step 4.2 Compute the maximum distance d_j from p_j to the vertices of its Voronoi polygon.

Step 4.3 Set $d_{max} = \max (d_i, d_j)$.

Step 4.4 If $d_{max} <$ length $(p_i\, p_j)$, delete edge $p_i\, p_j$.

Step 5. Output the remaining set of edges.

3.1.2 CONSERVATIVE-CRUST Algorithm

This algorithm [14] also starts with the Delaunay triangulation, DT, and goes through the following three filtration steps. Let $\rho > 0$ be a real-valued parameter.

Step 1. From the DT obtain the Gabriel graph, GG.

Step 2. From the GG, remove an edge e if the ball, **B**(e, l(e)/ρ), contains a Voronoi vertex. Let G' be the output graph after this step.

Step 3. From G' remove an edge e if the ball, **B**(e, l(e)/4ρ), contains a sample point of degree 0 or a sample point of degree 1 whose incident edge is not connected to e. Let G be the output graph after this step.

3.1.3 NEAREST NEIGHBOUR-CRUST Algorithm

Like the previous two algorithms, this one [13] also starts with the Delaunay triangulation on the sample set and extracts the Nearest Neighbour graph from it. However, unlike the previous two algorithms it now adds edges to this graph to obtain a reconstruction.

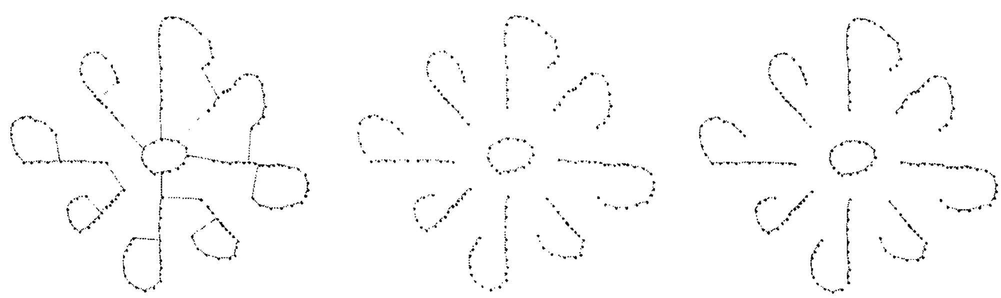
For this reason it is able to show that under a sampling condition ($\epsilon \leqslant 1/3$) the reconstruction has exactly the edge joining curve-adjacent points.

Step 1. Extract from the DT the nearest neighbour graph NN.

Step 2. Loop over the sample points in arbitrary order; if a sample point p is of degree 1, then consider the half-space, ***H***, defined by a line through p orthogonal to the edge e incident on p that does not contain e. Choose a shortest edge incident on p that lies in ***H*** and add it to NN.

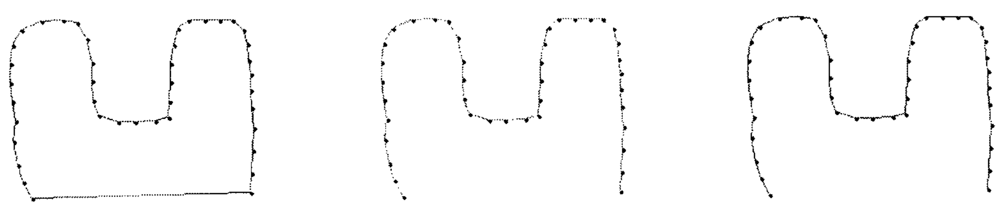
3.2 Experimental Results

As we did not have access to the source code for RNG-based, Conservative-Crust and Nearest Neighbour-Crust algorithms, and we were only interested in comparing the outputs of the three algorithms, we did brute-force implementations of these algorithms in Java programming language. We ran these three algorithms on 10 samples from different types of curves, such as simple curves, curves with sharp corners, curves with end points, nested curves and collection of curves (See Figs. 11-20).



(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 11: Sample 1, comparison of polygonal reconstructions

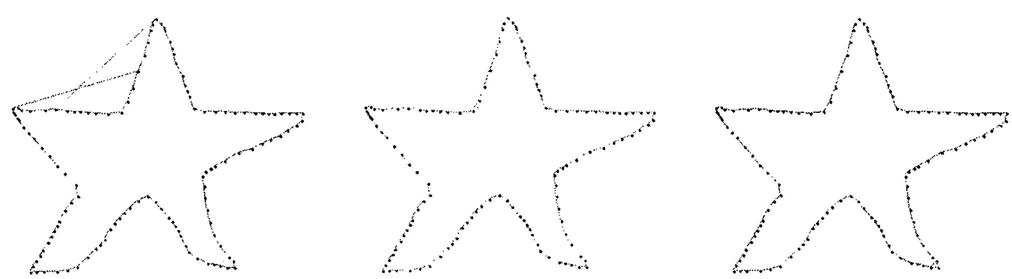


(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 12: Sample 2, comparison of polygonal reconstructions

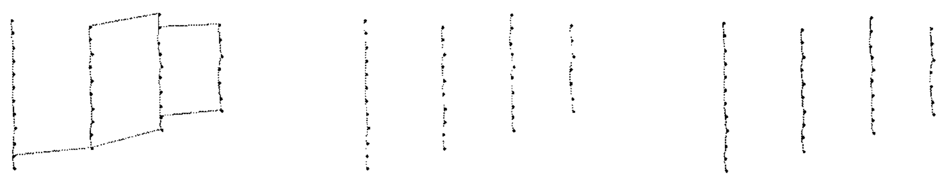
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 13: Sample 3, comparison of polygonal reconstructions



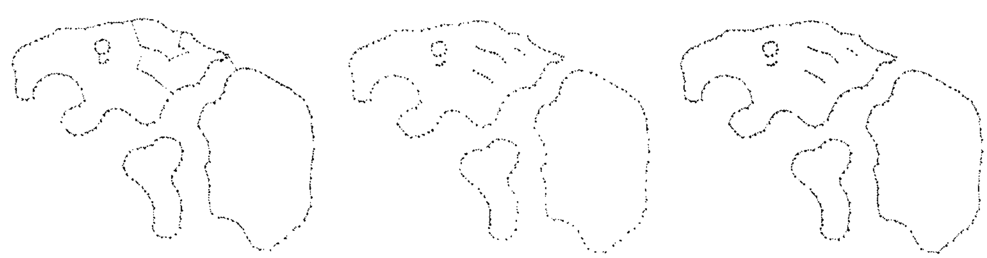
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 14: Sample 4, comparison of polygonal reconstructions



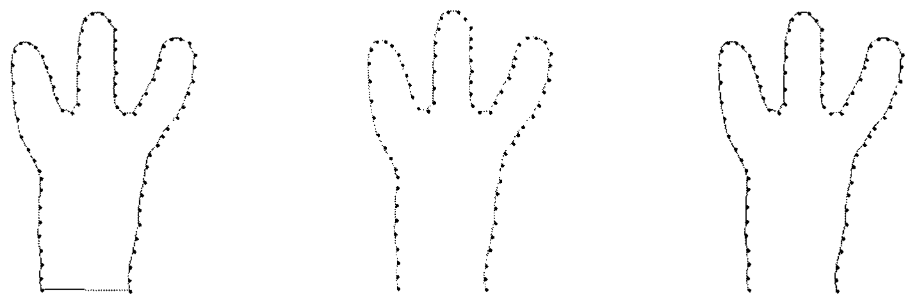
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 15: Sample 5, comparison of polygonal reconstructions



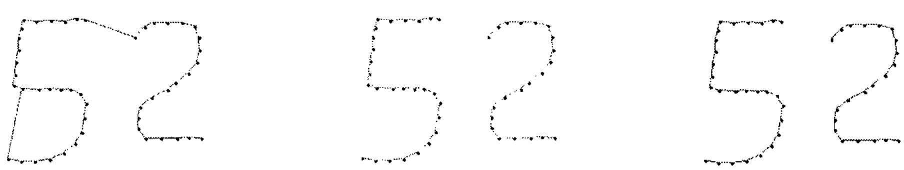
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 16: Sample 6, comparison of polygonal reconstructions



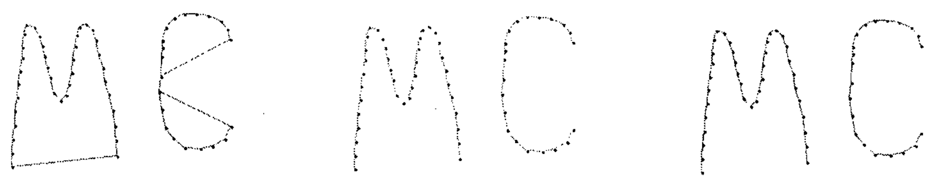
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 17: Sample 7, comparison of polygonal reconstructions



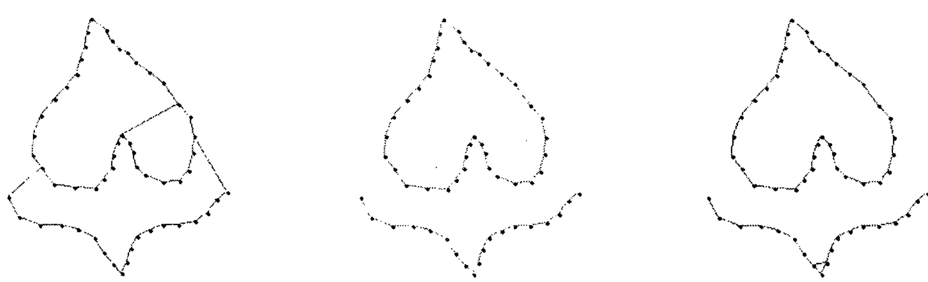
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 18: Sample 8, comparison of polygonal reconstructions



(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 19: Sample 9, comparison of polygonal reconstructions



(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 20: Sample 10, comparison of polygonal reconstructions

The samples above show clearly the superiority of the simple RNG-based heuristic. The Nearest Neighbour-Crust algorithm seems to perform the worst. This is somewhat intriguing as it comes with an iff-guarantee for the reconstruction provided we have an ϵ–sample with $\epsilon \leqslant 1/3$. We can observe that the NN-Crust algorithm always produces bad reconstructions for open curves; the RNG method, like Conservative-Crust algorithm, on the other hand does not come with an iff-guarantee and requires an ϵ–sample with $\epsilon \leqslant 1/5$. It is simple and works very well for the samples from different types of curves; The Conservative-Crust algorithm is troublesome to use. We had to fine-tune the parameter ρ to obtain an output to match the one produced by the RNG method. Particularly in the case of curves of varying features corresponding to their different parts, different parameters should be used for those different parts for the ideal sampling of the curves. Hence, the parameter is not always easy to be determined.

Chapter 4

Certification for Reconstruction

All the proposed algorithms can guarantee the correctness of the reconstruction provided the sample satisfies the sampling condition. But in practice we are just given a sample from some unknown curve and we can't verify a given sample is an ϵ–sample, therefore we are not sure whether or not the reconstruction produced by some algorithm is good.

In the second part of this thesis we propose a novel approach that bypasses this problem by certifying to the accuracy of the reconstruction.

4.1 Certification Algorithm

Let A be any curve-reconstruction algorithm. Our certification algorithm has the following four steps.

Algorithm **CERTIFICATION**

Step1. Run a reconstruction algorithm A on the sample S.

Step2. Smooth the resulting polygonal reconstruction into a set of curves C.

Step3. Resample the curves in C so that we have a sample point from each segment of a curve in C, which corresponds to an edge of the polygonal reconstruction.

Let S' be the resampled point set.

Step4. Match the point sets S and S'.

The closeness of the match in the last step is an indication of the accuracy of the curve-reconstruction algorithm.

4.1.1 Polygonal reconstruction

There are a number of reconstruction algorithms [3, 13, 14, 21, 37] that take an ϵ–sample as input and produce a provably correct reconstruction under suitable restrictions on the parameter ϵ. In this thesis, we use the RNG-based algorithm [37], the Nearest Neighbour-Crust algorithm [13] and Conservative-Crust algorithm [14] as $\mathcal{A}$.

4.1.2 Smooth the reconstruction

Let us assume that the polygonal reconstruction, P, consists of chains and cycles of varying sizes and isolated vertices. We smooth P, based on ideas suggested in [14].

The direction of the tangent to the smooth curve that passes through a vertex, p, of degree 2 is set to the direction of the tangent at p to the circumcircle, defined by p and its two neighbours. We fix the smooth curve piecewise for each edge pq as follows:

1. If both p and q are of degree 1, we retain this edge as part of our smooth curve.
2. If p is of degree 1 and q is of degree 2, then the piece of the smooth curve for this part is part of the circumcircle that is used to define the tangent at q (See Fig. 21).

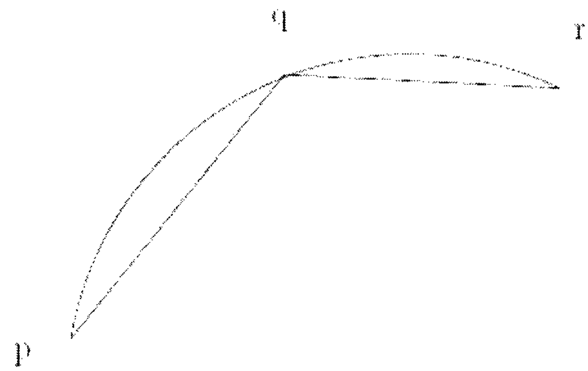


Figure 21: p is of degree 1 and q is of degree 2

3. If both p and q are of degree 2, we do this. Let up and qv be incident on p and q respectively. Two cases arise:

   - If u and v are on opposite sides of the supporting line of pq (See Fig. 22), the smooth curve through pq consists of 4 sub-pieces that are joined together to form a single piece. In the subpiece px, the circle section satisfies the tangent constraint at p and the tangent at x (1/4 location of pq) is parallel to pq. Similarly, in the subpiece qy, the circle section satisfies the tangent constraint at q and the tangent at y (3/4 location of pq) is parallel to pq. For the two middle subpieces, w is the midpoint of xy. zc_2 is the perpendicular bisector of xw, while sc_3 is the perpendicular bisector of wy.

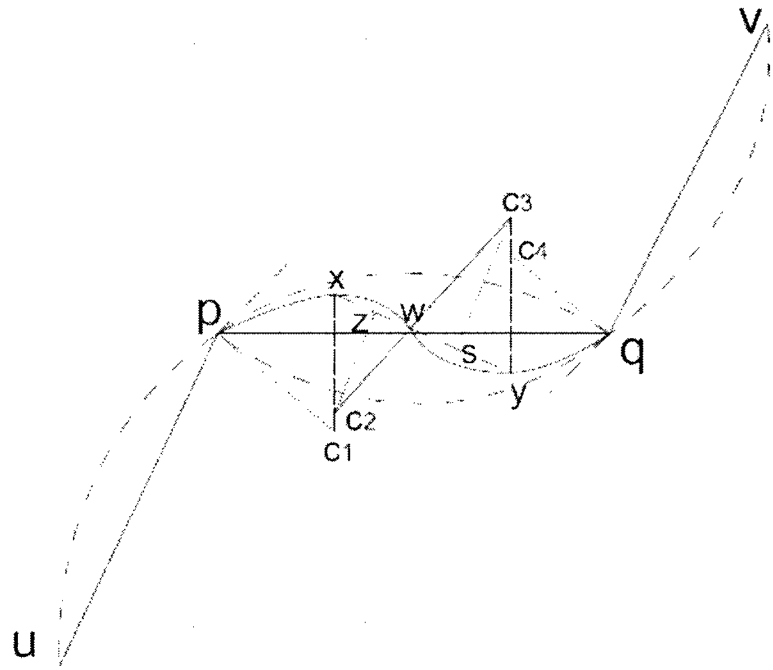


Figure 22: p and q are of degree 2 and the neighbors are on opposite sides

- If u and v are on the same side of the supporting line of pq (See Fig. 23), the smooth curve through pq consists of 2 sub-pieces that are joined together to form a single piece. Kp is the angle bisector of $\angle$mpq and Kq is the angle bisector of $\angle$nqp. The tangent b at K is perpendicular to line d. pc_1 is perpendicular to um and qc_2 is perpendicular to vn.

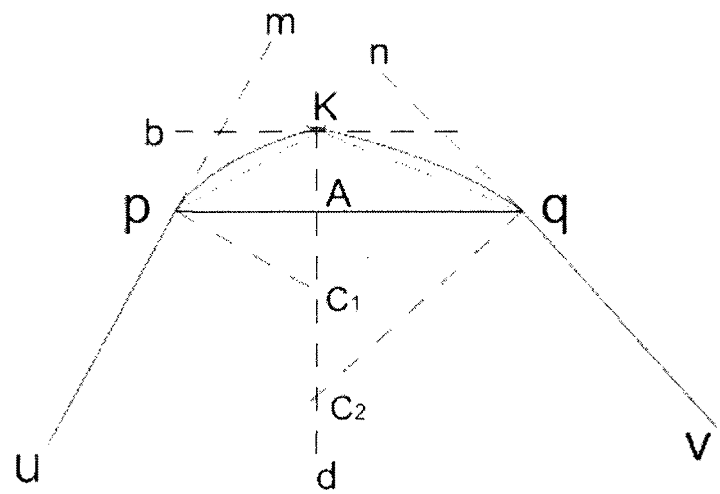


Figure 23: p and q are of degree 2 and the neighbors are on the same side

4.1.3 Sampling the smoothened curve

We sample the smooth curves in ***C*** constructed in the last section by choosing a random point from each section of a curve in ***C*** that corresponds to an edge in the polygonal reconstruction.

4.1.4 Matching the two samples

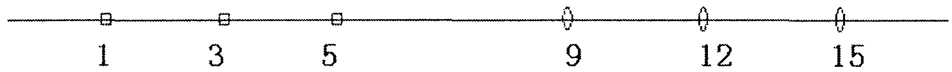
We discuss **Hausdorff distance** for quantifying the "distance" between the original sample ***S*** and the sample obtained from ***C***. We contend that the accuracy of the reconstruction algorithm ***A*** is reflected by this "distance".

The Hausdorff distance between two non-empty subsets X and Y of a metric space (M, d) is defined thus:

$$D_H(X, Y) = \max \{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \}$$

We demonstrate the Hausdorff distance between two sets by three examples.

Example 1 : $X = \{1, 3, 5\}$, $Y = \{9, 12, 15\}$.

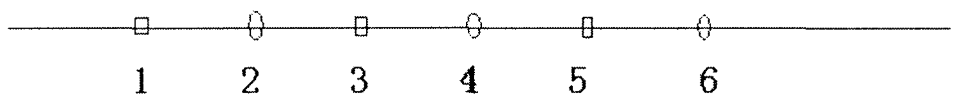


$\sup_{x \in X} \inf_{y \in Y} d(x, y) = \sup_{x \in X} d(x, Y) = \sup_{x \in X} \{8, 6, 4\} = 8$

$\sup_{y \in Y} \inf_{x \in X} d(x, y) = \sup_{y \in Y} d(X, y) = \sup_{y \in Y} \{4, 7, 10\} = 10$

hence, $D_H(X, Y) = \max \{8, 10\} = 10$

Example 2 : $X = \{1, 3, 5\}$, $Y = \{2, 4, 6\}$.

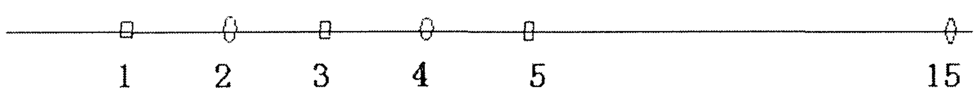


$\sup_{x \in X} \inf_{y \in Y} d(x, y) = \sup_{x \in X} d(x, Y) = \sup_{x \in X} \{1, 1, 1\} = 1$

$\sup_{y \in Y} \inf_{x \in X} d(x, y) = \sup_{y \in Y} d(X, y) = \sup_{y \in Y} \{1, 1, 1\} = 1$

hence, $D_H(X, Y) = \max \{1, 1\} = 1$

Example 3 : $X = \{1, 3, 5\}$, $Y = \{2, 4, 15\}$.



$\sup_{x \in X} \inf_{y \in Y} d(x, y) = \sup_{x \in X} d(x, Y) = \sup_{x \in X} \{1, 1, 1\} = 1$

$\sup_{y \in Y} \inf_{x \in X} d(x, y) = \sup_{y \in Y} d(X, y) = \sup_{y \in Y} \{1, 1, 10\} = 10$

hence, $D_H(X, Y) = \max \{1, 10\} = 10$

4.2 Experimental Results

4.2.1 Part Ⅰ

In this part, we ran our certification algorithm on the outputs of the RNG-reconstructions prior to and after removal of the non-curve adjacent edges. For each non-curve adjacent edge, we retain it as the smooth part through that polygonal edge and we choose a random point from that straight edge as a resampled point.

To test our RNG-algorithm [37], we measured $D_H(X, Y)$ for 10 different samples by screen coordinates (See Figs. 24-33). The entries in the second and third columns of Table 1, are respectively the values of $D_H(X, Y)$ prior to and after the removal of non-curve adjacent edges.

Sample	Without Remove-edges	Remove-edges
S1	18.00	13.99
S2	32.38	11.24
S3	27.31	10.62
S4	42.30	11.06
S5	35.60	10.95
S6	20.30	13.97
S7	39.82	11.72
S8	39.61	12.32
S9	29.15	16.28
S10	119.05	13.09

Table 1: Comparing Hausdorff distance between without/with Remove-edges

Since the reconstruction can be construed as being poor prior to the removal of the non-curve adjacent edges, the entries in the third column of Table 1 are smaller than those in the second.

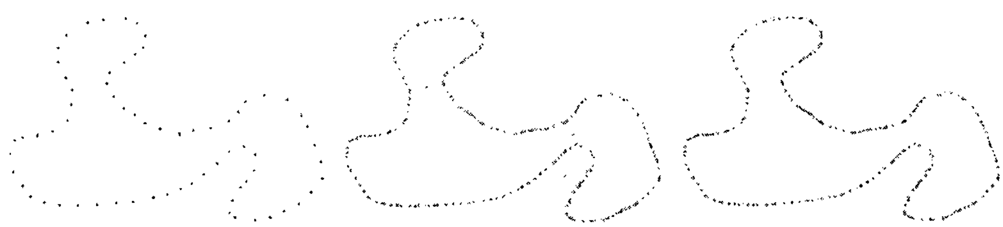


Figure 24: Sample 11, without Remove-edges, with Remove-edges

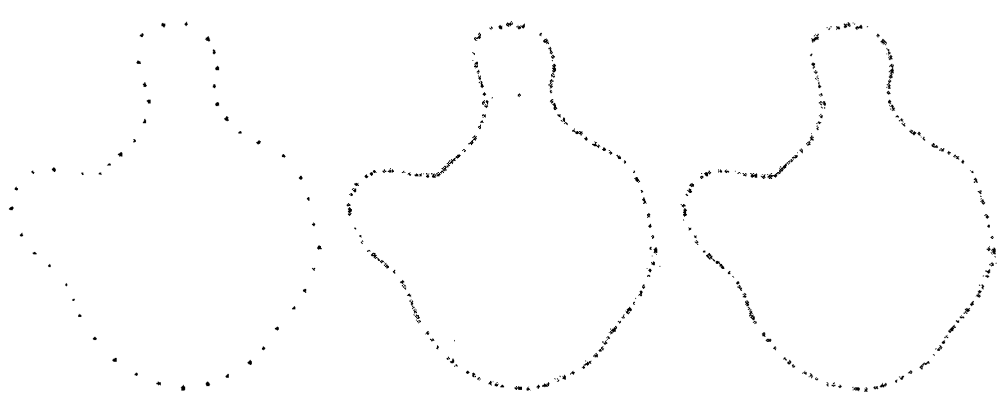


Figure 25: Sample 12, without Remove-edges, with Remove-edges

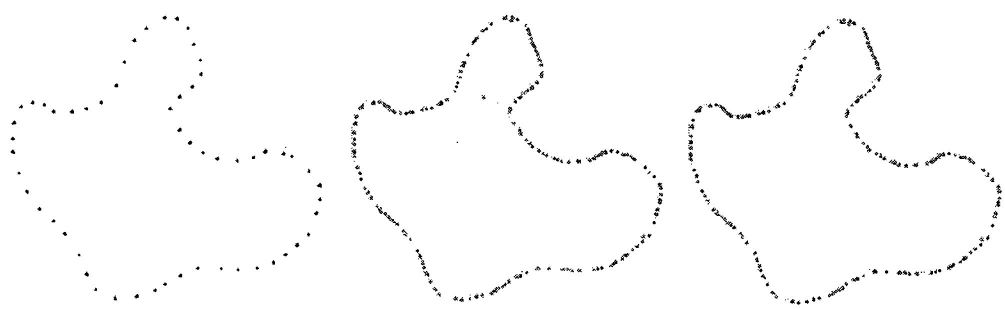


Figure 26: Sample 13, without Remove-edges, with Remove-edges

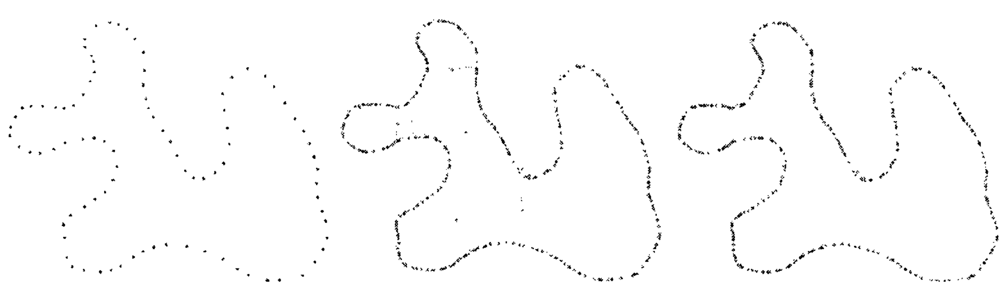


Figure 27: Sample 14, without Remove-edges, with Remove-edges

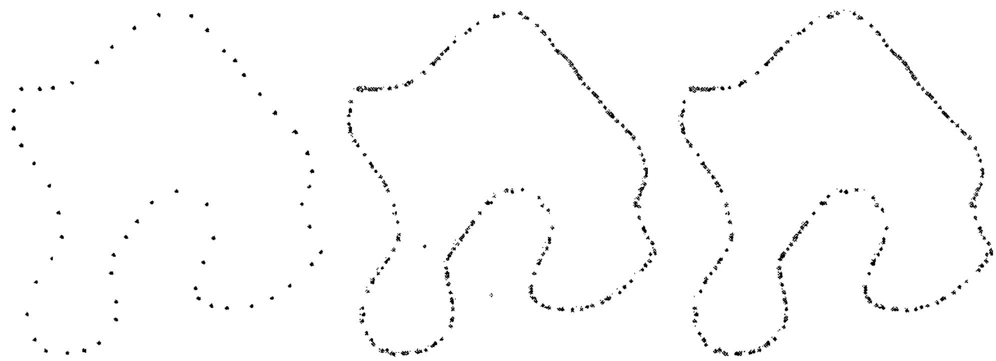


Figure 28: Sample 15, without Remove-edges, with Remove-edges

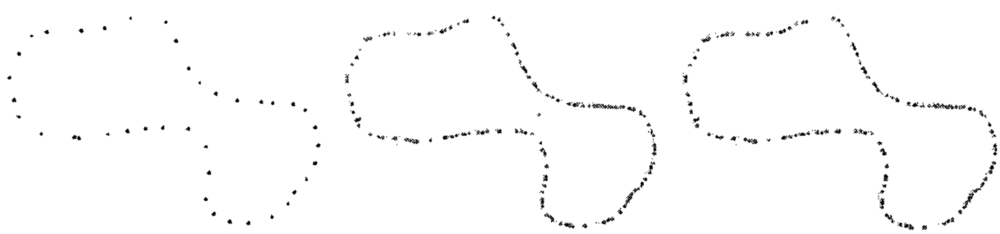


Figure 29: Sample 16, without Remove-edges, with Remove-edges

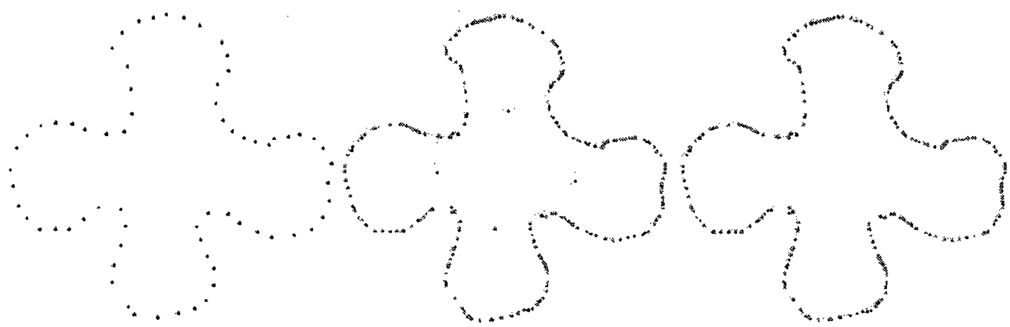


Figure 30: Sample 17, without Remove-edges, with Remove-edges

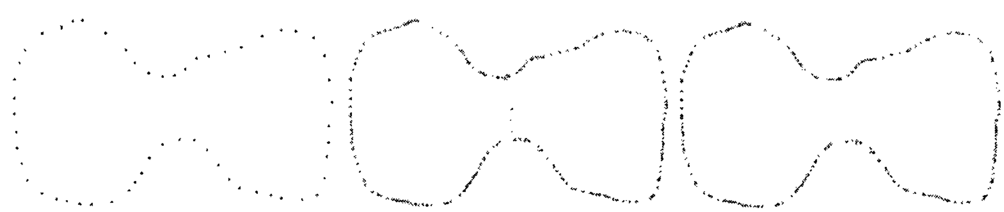


Figure 31: Sample 18, without Remove-edges, with Remove-edges

Figure 32: Sample 19, without Remove-edges, with Remove-edges

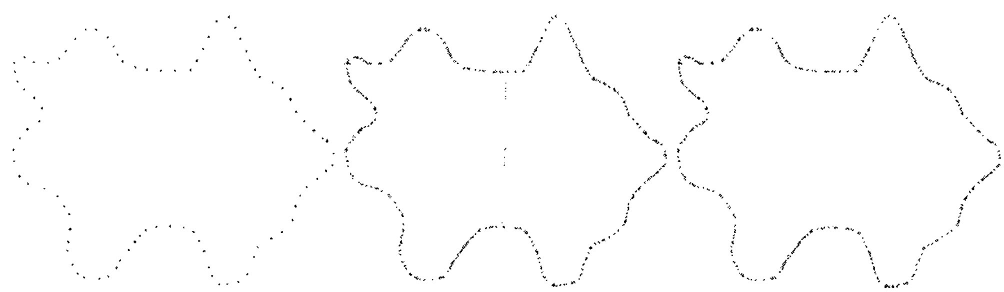
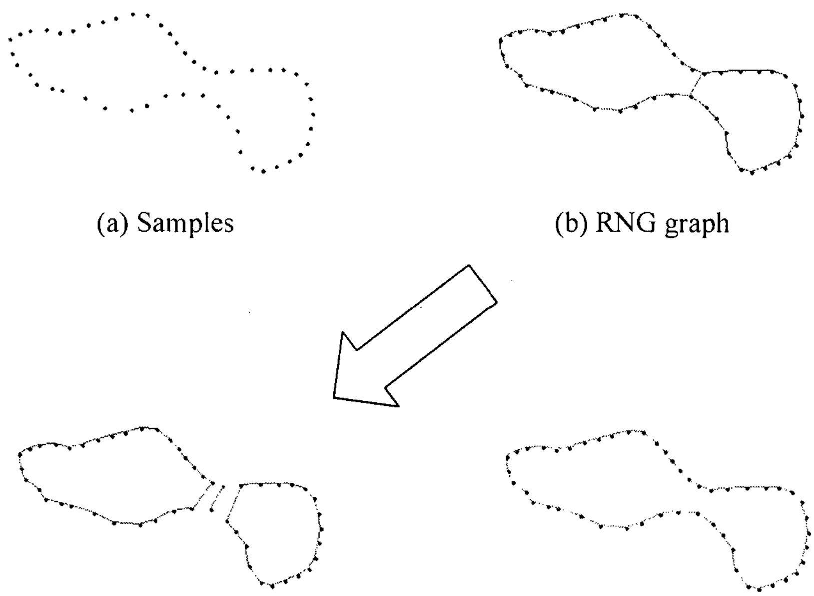


Figure 33: Sample 20, without Remove-edges, with Remove-edges

4.2.2 Part Ⅱ

We have also used our RNG-based reconstruction to simulate some bad reconstructions. Instead of removing the non-curve adjacent edges that appear before the clean-up stage, we make use of these to break up the polygonal graph into cycles and isolated edges (See Fig. 34).



(a) Samples (b) RNG graph

(c) Break-up (d) Remove-edges

Figure 34: Simulating a bad reconstruction

We ran our Certification Algorithm to compare this "poor" reconstruction with the "good" reconstruction that we get after removing the non-curve adjacent edges.

We compared $D_H(X, Y)$ for 10 different samples (See Figs. 35-44). The results are shown in the Table 2. The entries in the second and third columns of Table 2, are respectively the values of $D_H(X, Y)$ with Break-Up operation and after the removal of non-curve adjacent edges.

Sample	Break-up	Remove-edges
S1	22.83	11.68
S2	27.74	11.53
S3	74.98	10.95
S4	48.37	16.41
S5	42.58	12.84
S6	28.03	12.73
S7	32.40	13.54
S8	25.74	10.32
S9	31.83	11.89
S10	31.36	13.66

Table 2: Comparing Hausdorff distance between Break-up and Remove edges

Again we see that the reconstruction with the Break-Up operation is worse than that after the removal of the non-curve adjacent edges as the entries in the third column are smaller than those in the second.

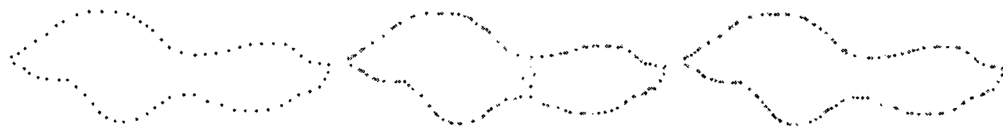


Figure 35: Sample 21, with Break-up, with Remove-edges

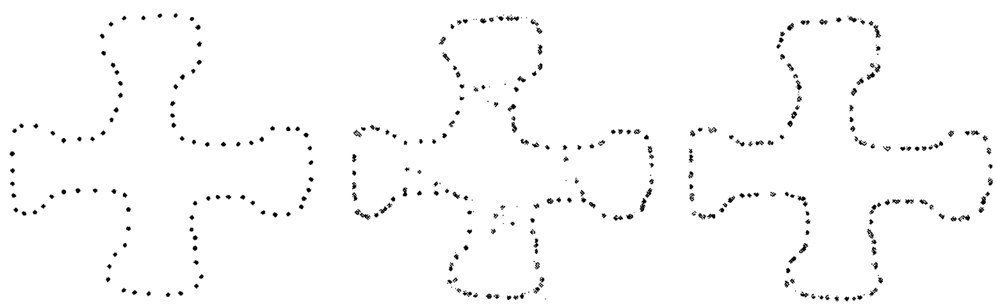


Figure 36: Sample 22, with Break-up, with Remove-edges

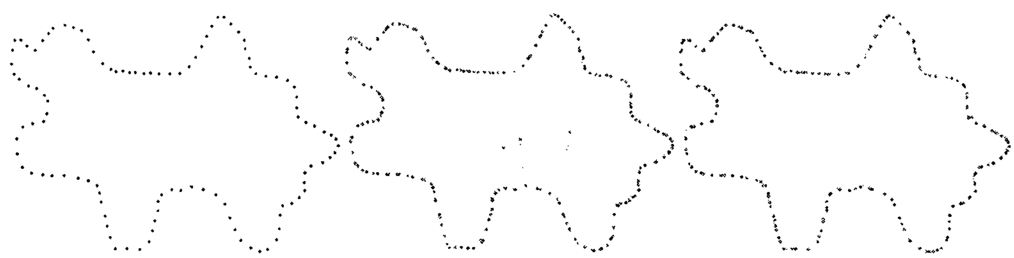


Figure 37: Sample 23, with Break-up, with Remove-edges

Figure 38: Sample 24, with Break-up, with Remove-edges

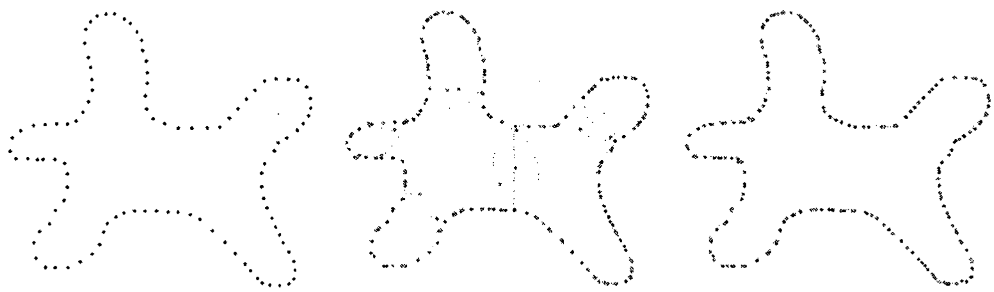


Figure 39: Sample 25, with Break-up, with Remove-edges

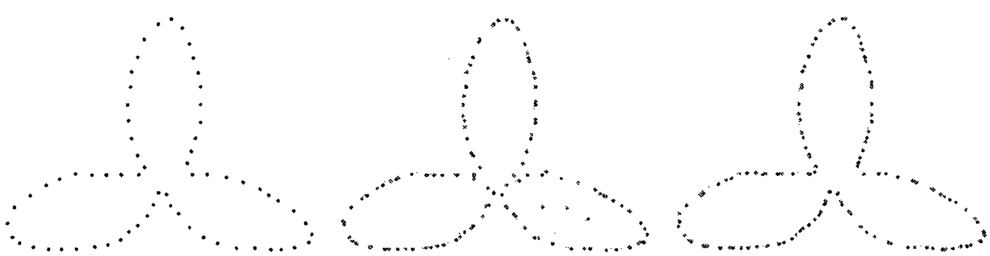


Figure 40: Sample 26, with Break-up, with Remove-edges

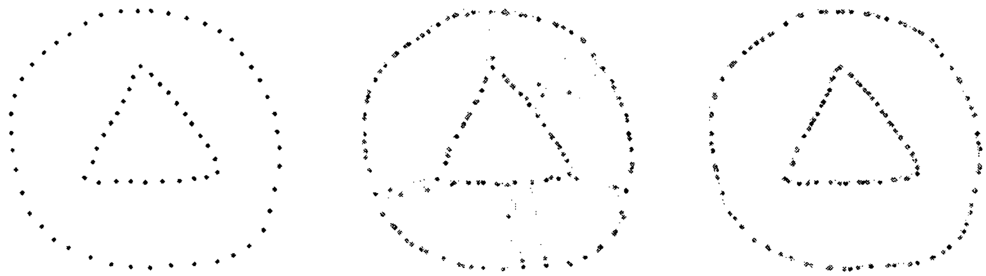


Figure 41: Sample 27, with Break-up, with Remove-edges

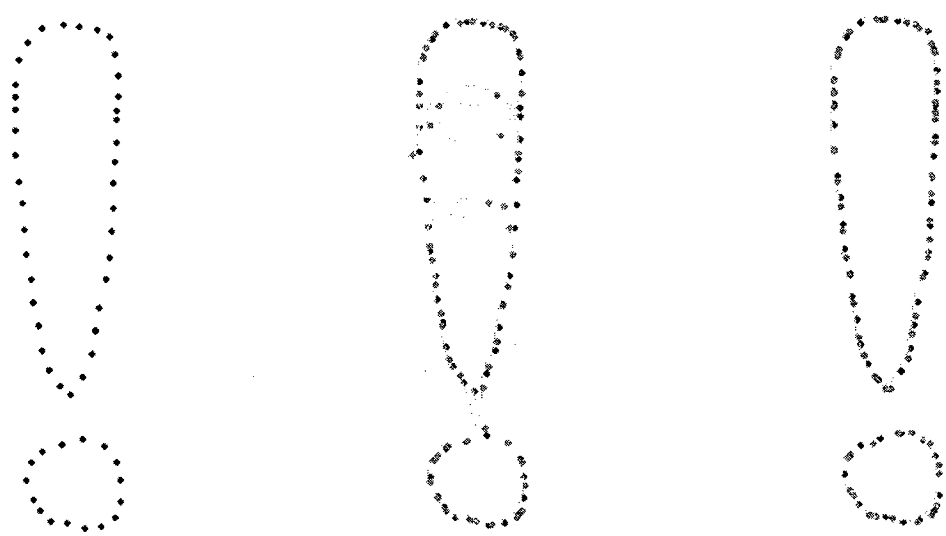


Figure 42: Sample 28, with Break-up, with Remove-edges

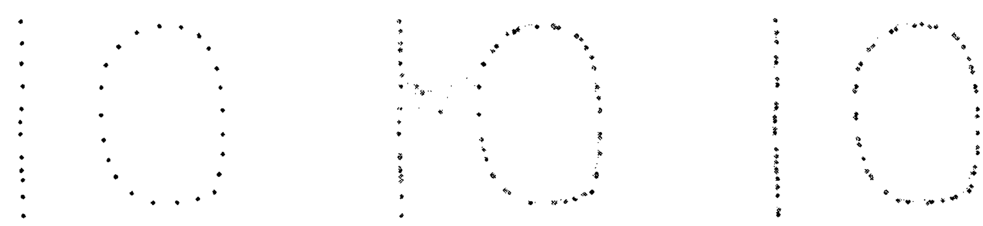


Figure 43: Sample 29, with Break-up, with Remove-edges

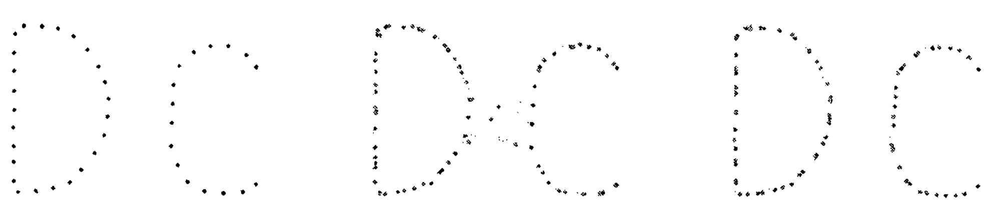


Figure 44: Sample 30, with Break-up, with Remove-edges

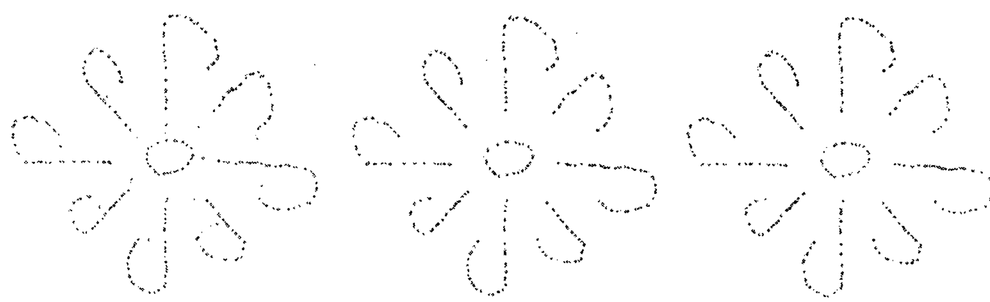
4.2.3 Part Ⅲ

Finally, and most importantly, we compared our RNG-based algorithm [37] with two leading algorithms: the Nearest Neighbour-Crust [13] and Conservative-Crust [14]. We ran our certification algorithm on the outputs of these three algorithms on 10 different samples, which have already been shown in the Chapter 3. When either endpoint of an edge of the polygonal reconstructions has a degree more than 2, we retain that edge as the smooth part and we choose a random point from it as a resampled point. The entries in the second, third and fourth columns of Table 3, are respectively the values of $D_H(X, Y)$ for the Nearest Neighbour-Crust, RNG-based algorithm and Conservative-Crust.

Sample	NN-Curst	RNG-based	Conservative-Crust
S1	30.50	12.27	12.87 (ρ=2.4)
S2	107.74	13.02	13.47 (ρ=2.4)
S3	48.95	13.49	13.45 (ρ=2.4)
S4	50.27	14.79	15.13 (ρ=3.4)
S5	49.25	15.63	15.75 (ρ=3.4)
S6	27.67	13.46	14.28 (ρ=3.4)
S7	58.60	13.15	13.53 (ρ=3.4)
S8	33.37	14.95	14.45 (ρ=2.8)
S9	92.99	15.60	15.37 (ρ=3.4)
S10	29.80	12.65	21.06 (ρ=4.7)

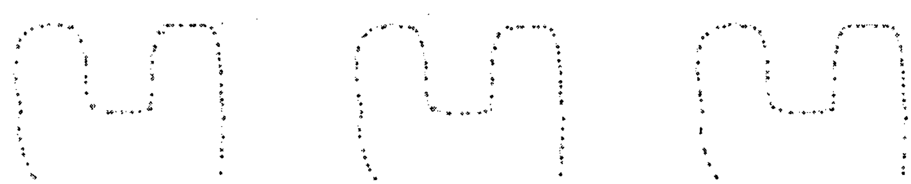
Table 3: Comparing Hausdorff distance among three algorithms

In the Conservative-Crust, we choose a suitable ρ value to produce the best reconstruction, which matches the one produced by the RNG method. We can claim that the outputs of RNG-based algorithm and Conservative-Crust are better than that of NN-Crust since the entries in the third and fourth columns are much smaller than those in the second. In addition, it shows very tiny differences between the entries of the third and fourth columns from S1 to S9. However, in S10, a non-curve adjacent edge at the bottom of the reconstruction in the Conservative-Crust leads to a bigger mismatch than that in the RNG-based algorithm. The figures below (See Figs. 45-54) appear to confirm the difference.



(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 45: Sample 1, comparison of smooth curves

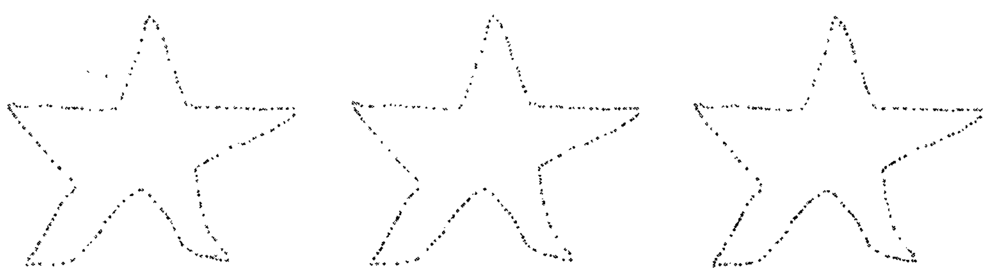


(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 46: Sample 2, comparison of smooth curves

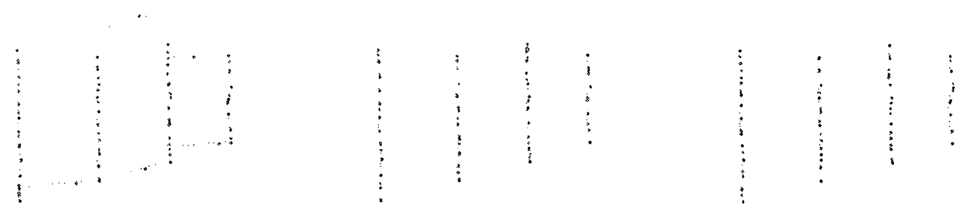
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 47: Sample 3, comparison of smooth curves



(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 48: Sample 4, comparison of smooth curves

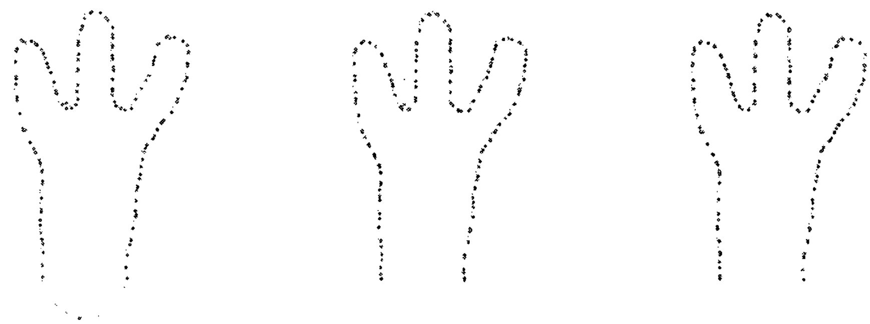


(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 49: Sample 5, comparison of smooth curves

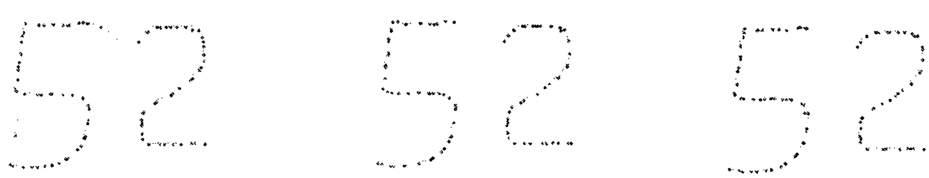
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 50: Sample 6, comparison of smooth curves



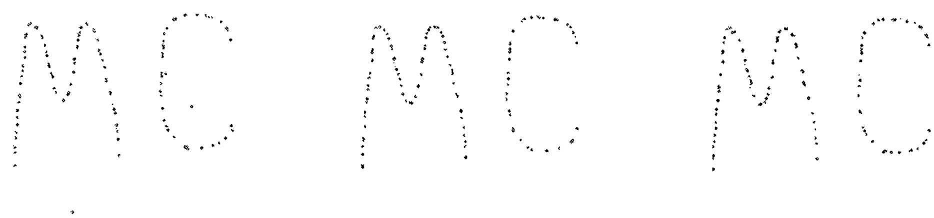
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 51: Sample 7, comparison of smooth curves



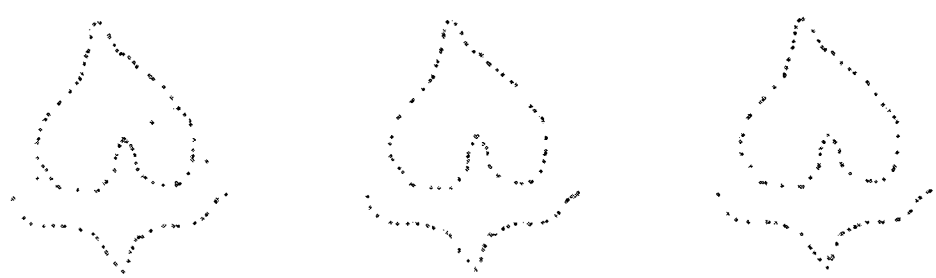
(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 52: Sample 8, comparison of smooth curves



(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 53: Sample 9, comparison of smooth curves



(a) NN-Crust (b) RNG (c) Conservative-Crust

Figure 54: Sample 10, comparison of smooth curves

Chapter 5

Conclusions and Future Work

Many algorithms for curve reconstruction have been proposed, some of which are only able to handle curves with particular features. In chapter 3 of this thesis, by comparing the polygonal reconstructions of RNG-based algorithm, NN-Crust and Conservative-Crust on a set of samples from different types of curves with multiple features, we establish the superiority of the simple RNG-based heuristic over the other two algorithms, where the RNG-based algorithm could reconstruct more types of curves with higher quality. However, the comparison is just based on visual inspection.

Although most algorithms guarantee the quality of the reconstruction when sampling condition is satisfied, we are not sure whether or not the given sample set is an ϵ–sample since the local feature size of any point on the curve can not be computed. In chapter 4, therefore, we propose a novel approach that bypasses this problem by certifying to the accuracy of the reconstruction. By smoothing the polygonal output of a reconstruction algorithm, we obtain a good approximation to the original curve. Then we sample the smoothened curve and compute the Hausdorff distance between the original sample and resampled set. We argue that the closeness of the match is an indication of the accuracy of the curve reconstruction algorithm. The experimental results show that the Hausdorff distance metric works quite well. Actually, the value of the Hausdorff distance depends on the longest redundant edge of the polygonal reconstruction. Hence, a longer redundant edge leads to a bigger mismatch for the Hausdorff distance. Finally and most importantly, we ran our certification algorithm on the polygonal outputs of the same samples that we have shown at the end of chapter 3. This approach further verifies the same conclusion that RNG-based algorithm is better than the other two algorithms. On one hand, NN-Crust always causes the biggest mismatch among these three algorithms. On the other hand, although we find that RNG-based algorithm and Conservative-Crust usually produce very close values of Hausdorff distance, for the Conservative-Crust, we have to fine-tune the parameter to obtain an output to match the one produced by the RNG method.

In addition, we are convinced the certification algorithm can also be extended to three dimensions. This is an avenue worth exploring. Moreover, with the requirement of massive data in the advanced technologies, the whole data can not be held in the limited memory. It might be interesting to study the curve reconstruction problem in the streaming model.

Bibliography

[1] ALTHAUS, E. AND MEHLHORN, K. 2000. TSP-Based Curve Reconstruction in Polynomial Time. *Proc. 11th SLAM Symp. On Discrete Algorithm*, 686-695.

[2] ALTHAUS, E., MEHLHORN, K., NAHER, S., AND SCHIRRA, S. 1999. Experiments on curve reconstruction. unpublished.

[3] AMENTA, N., BERN, M., AND EPPSTEAIN, D. 1997. The crust and the β–skeleton: combinatorial curve reconstruction. *Manuscript, (1997). To appear in Graphical Model and Image Processing.*

[4] ATTALI, D. 1997. R-regular shape reconstruction from unorganized points. *Proc. 13th Ann. Sympos. Comput. Geom.*, 248-253.

[5] BENTLEY, J. 1992. Fast Algorithm for Geometric Traveling Salesman Problems. *In ORSA Jornal on Computing*, 4, 4: 387-411.

[6] BENTLEY, J.L., WEIDE, B.W. AND YAO, A.C. 1980. Optimal expected-time algorithms for closest point problems. *ACM Trans. Math. Software 6, 563-580.*

[7] BERNARDINI, F. AND BAJAJ, C.L. 1997. Sampling and reconstructing manifolds using α-shapes. *Proc. 9th Canadian Conf. Comput. Geom.*, 193-198.

[8] BLUM, H. 1967. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form MIT Press,* 362-380.

[9] BOLAND, J.W., BRIGHAM, R.C., AND DUTTON, R.D. 1987. The difference between a neighborhood graph and a wheel. *Congressus Numerantium, 58, 151-156, 1987.*

[10] BRANDT, J. AND ALGAZI, V.R. 1992. Continuous skeleton computation by Voronoi diagram. *Computer Vision, Graphical and Image Processing, 55,* 329-338.

[11] CALLAGHAN, J.F. 1975. An alternative definition for neighbourhood of a point. *IEEE Trans. Comput.* C-24, 1121-1125.

[12] CHENG, S.W., FUNKE, S., GOLIN, M., KUMAR, P., POON, S.H., AND RAMOS, E. 2003. Curve Reconstruction from Noisy Samples, *Proc. ACM Symposium on Computational Geometry, 2003, 302-311.*

[13] DEY, T.K. AND KUMAR, P. 1999. A simple provable algorithm for curve reconstruction. *Proc. 10th. SIAM Sympos. Discr. Algorithms*, 893-894.

[14] DEY, T.K., MEHLHORN, K., AND RAMOS, E.A. 1999. Curve reconstruction: connecting dots with good reason, *Proc. 15th ACM Symposium Computational Geometry*, 197-206.

[15] DEY, T.K. AND WENGER, R. 2000. Reconstructing curves with sharp corners. *Proc. 15th Ann. ACM Sympos. Comput. Geom.*, 233-241.

[16] EDELSBRUNNER, H., KIRKPATRICK, D., AND SEIDEL, R. 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*; 29(4): 71-8.

[17] FUNKE, S. AND RAMOS, E.A. 2001. Reconstructing a collection of curves with corners and endpoints. *In Proceeding of the twelfth annual ACM-SLAM symposium on Discrete algorithms*, 344-353. *Society for Industrial and Applied Mathematics.*

[18] GIESEN, J. 1999. Curve reconstruction, the TSP, and Menger's theorem on length. *In Proceedings of the 15th Annual ACM Symposium on Computational Geometry (SCG'99)*, 207-216.

[19] GIESEN, J. 1999. Regularity Considerations in Curve Reconstruction with Delaunay Complex. unpuplished, 1999.

[20] GOLD, C.M. 1998. The Quad-Arc data structure. *In: Poiker, T.K. and Chrisman, N.R. (eds.). Proceedings, 8^{th} International Symposium on Spatial Data Handling: Vancouver, BC*, 713-724.

[21] GOLD, C.M. 1999. Crust and anti-crust: a one step boundary and skeleton extraction algorithm. *In 15th ACM Symposium on Computational Geometry*, 189-196.

[22] GOLD, C.M., J. NANTEL AND W. YANG. 1996. Outside-in: an alternative approach to forest map digitizing. *International Journal of Geographical Information Systems, v. 10, no. 3,* 291-310.

[23] HIYOSHI, H. 2006. Closed Curve Reconstruction from Unorganized Sample Points. *Proceeding of the 3rd International Symposium on Voronoi Diagram in Science and Engineering*, 351-356.

[24] HUANG, N.F. 1990. A divide-and-conquer algorithm for constructing relative neighborhood graph. *BIT Numerical Mathematics, Springer Netherlands, 30*(2).

[25] JAROMCZYK, J.W. AND KOWALUK, M. 1987. Linear time algorithm to construct the relative neighborhood graph from the Delaunay triangulation in two dimensions with the metric Lp. *Technical report No. 129-88, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0027.*

[26] JAROMCZYK, J.W. AND TOUSSAINT, G.T. 1992. Relative neighbourhood graphs and their relatives. *Proc. IEEE*, 80(9): 1502-1517.

[27] JARVIS, R.A. 1978. Shared near neighbour maximal spanning trees for cluster analysis. *Proc. 4th Int. Joint Conf. on Pattern Recognition*, 308-313, Kyoto, Japan.

[28] KATAJAINEN, J. 1988. The region approach for computing relative neighbourhood graphs in the Lp metric. *Computing* 40, 147-161.

[29] KATAJAINEN, J. AND NEVALAINEN, O. 1986. Computer relative neighbourhood graphs in the plane, *Pattern Recognition, 19*(3), 221-228.

[30] KATAJAINEN, J., NEVALAINEN, O., AND TEUHOLA, J. 1987. A linear expected-time algorithm for computing planar relative neighbourhood graph. *Inform. Process. Lett*, 25, 77-86.

[31] KIRKPATRICK, D.G.. AND RADKE, J.D. 1988. A framework for computational morphology. *Computational Geometry, 217-248.*

[32] LANKFORD, P.M. 1969. Regionalization: theory and alternative algorithms, *Geographical Analysis*, 1.

[33] LENZ, T. 2005. Reconstructing collections of arbitrary curves. *In SCG '05: Proceedings of the 21st annual symposium on Computational geometry*. 366-367. http://compgeom.poly.edu/acmvideos/socg05video/index.html.

[34] LENZ, T. 2006. How to sample and reconstruct curves with unusual features. *22nd European Workshop Computational Geometry*, 273-276.

[35] LEWIS, B.A. AND ROBINSON, J.S. 1978. Triangulation of planar regions with applications. *Comput. J.* 21, 324-332.

[36] MEDEK, V. 1981. On the boundary of a finite set of points in the plane. *Computer Graphics and Image Processing, 15, 93-99.*

[37] MUKHOPADHYAY, A. AND DAS, A. 2006. An RNG-based heuristic for curve reconstruction, *Peoceedings of the International Symposium on Voronoi*

Diagram, 246-251.

[38] MUKHOPADHYAY, A. AND DAS, A. 2007. Curve reconstruction in the presence of noise. *Computer Graphics, Imaging and Visualisation, 151-156.*

[39] OKABE, K.S.A., BOOTS, B., AND CHIU, S.N. 2000. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. *John Wiley & Sons, Chichester, second edition.*

[40] OLARIU, S. 1989. A simple linear-time algorithm for computing the rng and mst of unimodal polygons. *Inform. Process. Lett*, 31, 243-248.

[41] O'Rourke, H.B.J AND WASHINGTON, R. 1987. Connect-the-dots: a new heuristic. *Computer Vision, Graphics, and Image Processing*, 39, 258-266.

[42] SHAMOS, M.I. AND HOEY, D. 1975. Closest point problems. *Proc. 16th Ann Symp. On the Foundations of Computer Science, IEEE.* 151-162.

[43] SUPOWIT, K.J. 1983. The relative neighbourhood graph, with an application to minimum spanning trees. *J. Assoc. Comput. Mach*, 30, 428-448.

[44] TOUSSAINT, G.T. 1980. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12, 261-268.

[45] TOUSSAINT, G.T. AND MENARD, R. 1980. Fast algorithms for computing the planar relative neighbourhood graph. *In Proc. 5th Symp. Operations Research*, 1980, 425-428.

[46] URQUHART, R.B. 1980. Algorithms for computation of relative neighbourhood graph. *Electronics Letter.*

[47] VELTKAMP, R. C. 1992. The γ-neighborhood graph. *Computational Geometry: Theory and Applications 1:4,* 227-246.

[48] YAO, A.C. 1982. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput. 11, 721-736.*

[49] ZENG, Y., NGUYEN, T.A., YAN, B., AND LI, S. 2007. A distance-based parameter free algorithm for curve reconstruction. *Computer-Aided Design*, 65-68.

[50] ZHU, D., ZENG, Y., AND RONSKY, J. 2004. Geometric feature-based algorithms for curve reconstruction. *Technical report, Concordia institute for Information systems Engineering, Concordia University; February 2004.*

Vita Auctoris

Name : Chong Wang

Year of Birth : 1984

Place of Birth : Wuhan, Hubei, China

Education :

- Wuhan University of Technology, Wuhan, China

  2003-2006 B.Sc.

- University of Windsor, Windsor, Ontario

  2006-2008 B.Sc.

- University of Windsor, Windsor, Ontario

  2008-2010 M.Sc.