Vision-Based Curve Reconstruction

Shu Ren Li

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

August 2007

Your file Votre référence
ISBN: 978-0-494-34632-7
Our file Notre référence
ISBN: 978-0-494-34632-7

ABSTRACT

Vision-Based Curve Reconstruction

Shu Ren Li

A typical curve reconstruction problem is to generate a continuous linear representation of a curve from a set of unorganized sampling points on the curve. These unorganized points should be joined by edges in the order in which they appear on the curve. There are many methods to reconstruct curves from existing point clouds. Most of the existing algorithms are designed based on concepts from computational geometry. The current algorithms have difficulties in reconstructing curves with sharp corners or noisy points and they depend on predefined parameters. The present thesis proposes a different way to reconstruct curves, that is, to reconstruct curves based on the experiments of human vision. The curves should be reconstructed in the same manner that human beings perceive them. In the present thesis, statistical experiments are conducted to construct a vision function. A software system, based on that vision function, is developed to simulate human vision for curve reconstruction. The experiments investigate the relationships between points and the relationship between points and curves, by using methods from Design of Experiments (DOE), ANOVA and the multivariate non-linear regression model. The errors between the predicted values using the regression model and the observed values from vision experiments follow normal distribution. The algorithm based on the constructed vision function uses the key factors as input to

identify curves for given points. Examples show that the curve reconstruction results using this new algorithm are advantageous by comparison the results with from existing algorithms.

Acknowledgements

First, I would like to thank my supervisor Dr. Yong Zeng. Dr. Zeng has offered me precise guidance and creative comments throughout my Master's study. I have benefited enormously from the discussions with Dr. Zeng, especially in research methodology, modes of thinking, and analytical strategies.

My gratitude also goes to Baiquan Yan, Shengji Yao, Bo Chen and other members in the Design Lab in CIISE at Concordia, who have helped me so much during my research.

My wife has shown her patience and understanding during my study. She has given me constant encouragement when I have encountered problems.

Table of Contents

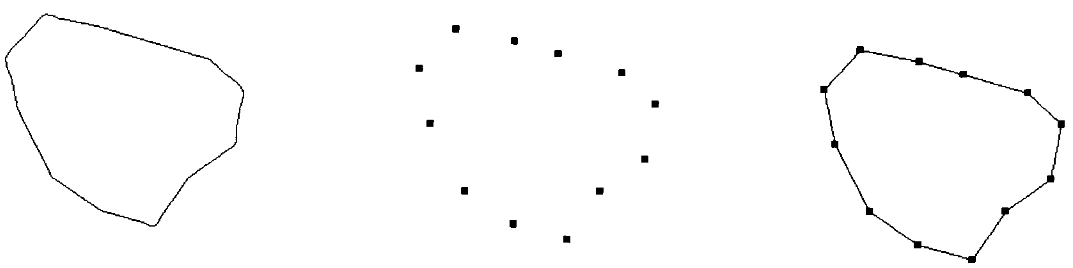
List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

The curve reconstruction problem is to reconstruct a set of unorganized sampling points into a continuous linear representation of a curve. This problem is illustrated in Figure 1.



(a) (b) (c)

Figure 1 Conventional curve reconstruction problem

Curve reconstruction has been used in a large number of applications such as reverse engineering, medical applications, and geotechnical exploration.

When a new product is developed, modifications are often made on a real-world prototype. In order to reflect the modification in an element file, the real-world prototype is often scanned with a laser scanner. A CAD model can then be derived from the scanned model of the real-world prototype. The procedures include scanning the prototype to take a set of sample points from the surface of the prototype, reconstructing curves on the sample points for each layer based, and reconstructing the surfaces based on the curves. Precisely reconstructed curves can improve the quality of the CAD model as is shown in Figure 2.

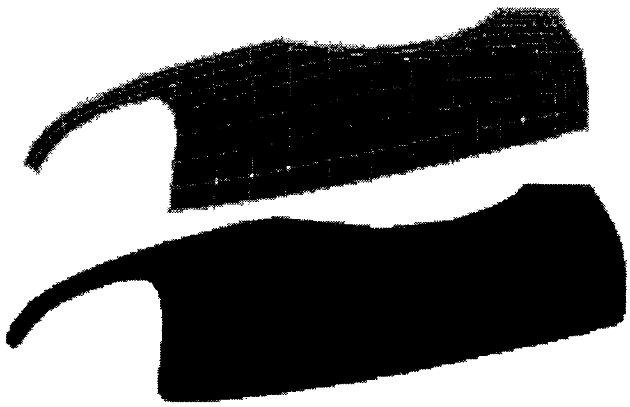


Figure 2 Application of curve reconstruction in CAD model (Pernot, 2002)

Another application area is biomedical engineering. Computertomographers are widely used in scanning the human body in layers, thereby gathering valuable information about the tissue structure. Both the layered view and the three-dimensional model are in great demand. These need to detect automatically, the boundaries between different tissue types in the layered scans. Using curve reconstruction techniques, this information can be processed into a collection of curves that represent the different tissue types in each layer. Figure 3 gives an example. For example, in computer tomography (CT) and magnetic resonance imaging (MRI), each slice or layer is a sample of the shape created by the intersection of the scanned object with a plane. One approach in reconstructing the surface is first to reconstruct the shapes in each layer and then connect adjacent layers appropriately.

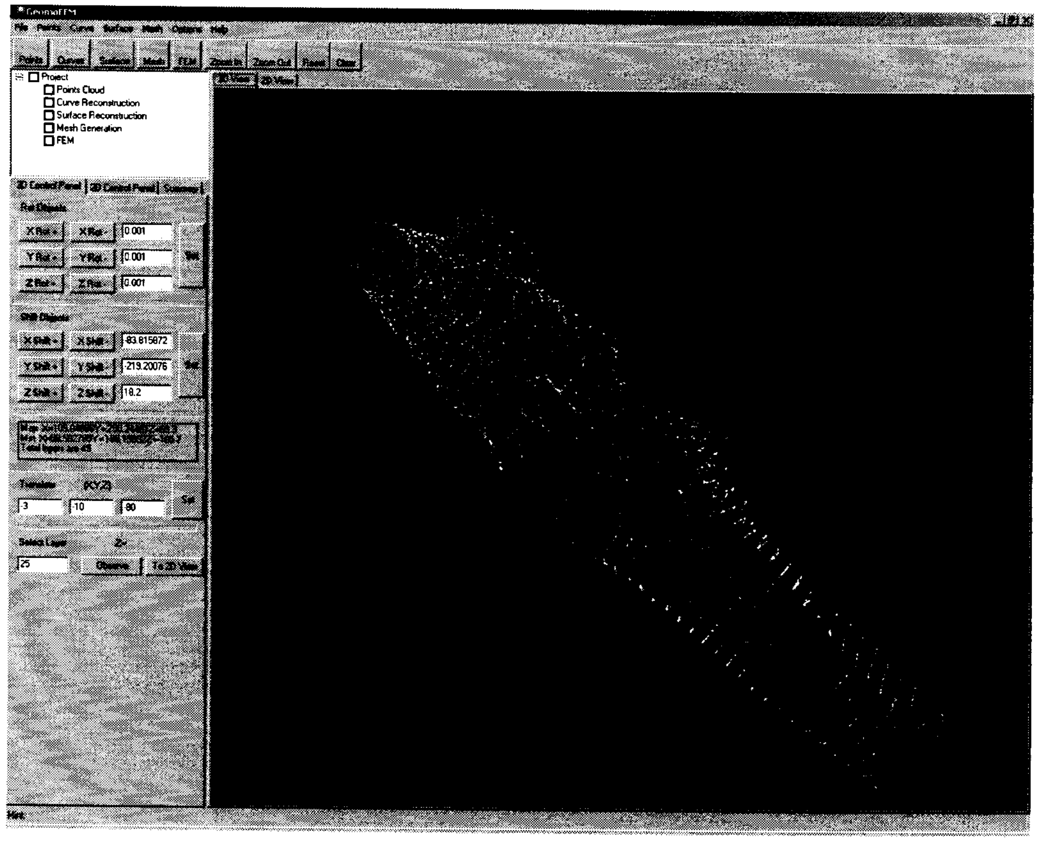


Figure 3 Application of curve reconstruction in biomedical engineering

In addition, geotechnical exploration also uses curve reconstruction techniques and surface reconstruction techniques to build the 3D geotechnical model. The methodology is similar to that used in the aforementioned applications (MRI and CT).

These applications require the curves to be reconstructed with precise, efficient, and robust algorithms. Because of the varieties in shapes and applications, many algorithms have been proposed over the last two decades. The target of the present thesis is general applications.

1.2 Objective

A typical curve reconstruction problem is to generate a continuous linear representation of a curve from a set of unorganized sampling points on the curve. These unorganized points should be joined by edges in the order in which they appear on the curve. Two sample points can be connected if and only if they are adjacent on the curve. In general, there are five types of curves: a) a simple smooth open curve, b) a simple smooth closed curve; c) a simple curve with sharp corners, d) a self-intersecting curve, and e) a combination of the four basic types of curves a) to d). A simple curve is a curve without self-intersection.

Obviously, using only these sample points, it is impossible to produce a linear representation of the curve unless some criteria are established beforehand. Considering the application domains of curve reconstruction algorithms, such as computer vision, biomedical imaging, and reverse engineering, the sampling points should be connected in a pattern that is natural to the human eye. The algorithm presented in the present thesis is based on two observations about the human visual system in the context of curve reconstruction: 1) two closest neighbors tend to be connected, and 2) sampling points tend to be connected into a smooth curve (Zeng *et al*, 2007).

The variety of reconstruction problems is huge. However, they can be classified into two categories. In the first category of problems, the original curve is known, researchers analyze the properties of the curve, and they propose a sampling condition. Then an algorithm is developed so that all samples can be correctly connected if the sampling condition is satisfied. In the other category of problems, the original curve is not known. The only information is a set of unorganized sampling points or a physical curve without an equation or any other mathematical representation from this curve using techniques such as CT or MRI as is the case in biomechanical engineering. Then researchers develop an algorithm to reconstruct this curve.

Many curve reconstruction algorithms have been designed in recent years. Most of those algorithms are based on concepts from computational geometry. People always compare the algorithms by the reconstruction results. The present thesis aims to design a new algorithm that reconstructs curves from points in the same way that human beings perceive them. To discover the effect of human vision on curve reconstruction, experiments were designed and conducted to identify how human vision recognizes curves from unorganized points.

1.3 Challenges

In the second category of problems introduced above, the original curve is not known. The only information is a set of unorganized sampling points. This category of reconstruction problems gives rise to two challenges: how to sample the curve and how to evaluate the result.

Suppose that a finite set of points on a plane curve is given. If the sample is large enough and is well distributed, then it is an easy task for human beings to perceive the shape of the curve. The human perception of curve shapes includes not only topological aspects, such as the identification of connected components and the differentiation between open and closed pieces, but also geometrical aspects, such as qualitative measures of curvature and winding.

A computer, on the contrary, has no such natural perception: the sample has no priori structure that can be exploited to provide a computational description of the curve. To the machine, the sample is merely a list of coordinates. A fundamental problem, which we would expect to solve using such computational descriptions, is exactly how to sort the points in an order compatible with the natural trace of the curve as perceived by a human. This order may be used to structure the sample into a polygonal approximation of the curve, thus reconstructing the curve from the sample. Reproducing the human perception of shapes from dot patterns (dense uniform samples of "fat" points on plane regions) is a classical problem in low-level computer vision, pattern recognition, and cluster analysis (Zahn 1971; Toussaint 1980). Much of the work in computational geometry was originally motivated by problems in these fields. Toussaint (1980) coined the term "computational morphology" for use in computational geometry for extracting shape information.

1.4 Contributions

The major contributions made in the present thesis can be summarized as follows:

1. Construction of a vision function based on experiments. The present thesis analyzes the experiment and calculates the regression model.

2. Development of a vision-based curve reconstruction algorithm.

1.5 Thesis organization

The rest of the present thesis is organized as follows:

Chapter 2 reviews the main algorithms for curve reconstruction and the basic concepts behind these algorithms, such as the Voronoi diagram, Delaunay triangulation, α-shape, medial axis and local feature size. The following algorithms are reviewed: ABE algorithm, NN-CRUST algorithm, Conservative-crust algorithm, GATHAN algorithm, and TSP algorithm.

Chapter 3 introduces a distance-based parameter-free algorithm. Also introduced are the criteria for the connectivity of points, the necessary and sufficient conditions for the sampling, and the platform for curve reconstruction algorithms design.

Chapter 4 explains the procedure of the modeling of human vision in the context of curve reconstruction. It includes factors analysis, experiments, and regression.

Chapter 5 gives a vision-based curve reconstruction algorithm. The comparison between the algorithm and current algorithms is discussed.

Chapter 6 summarizes the main research results based on the present thesis and points out future research directions.

The list of my publications completed during my graduate studies is provided after Chapter 6.

Chapter 2 Literature Review and Fundamental Concepts

As mentioned in Chapter 1, the curve reconstruction problems can be classified into two categories. In the first category of problems, the original curve is known; a sampling condition can be proposed by analyzing the properties of the curve. Then an algorithm is developed so that all samples can be correctly connected if the sampling condition is satisfied. In the other category of problems, the original curve is unknown. The only information is a set of unorganized sampling points.

For the first category of reconstruction problems, a curve can be sampled either uniformly or non-uniformly (Dey, 2004), which have been handled by researchers with different algorithms. A uniformly sampled simple closed curve can be reconstructed using the algorithms such as minimum spanning tree (Figueiredo *et al*, 1994), alpha shapes (Bernardini *et al*, 1997, Edelsbrunner *et al*, 1983), and r-regular shapes (Attali, 1997). For non-uniformly sampled simple closed curves, the crust and skeleton method has been developed by Amenta, Bern and Eppstein (Amenta *et al*, 1998), and refined by Gold (Gold, 1999). Dey and Kumar also proposed an algorithm to deal with this type of curve using the nearest neighbor, medial axis, and Delaunay triangulation (Dey *et al*, 1999). For simple open curves, a method using Voronoi and Delaunay disks of edges has been introduced by Dey, Mehlhorn and Ramos (Dey *et al*, 1999). To reconstruct curves with sharp corners, Dey and collaborators have recently developed another algorithm based on two predefined parameters: a threshold distance for filtering Delaunay triangles and an angle to adjust the smoothness and sharp corners in the reconstruction of curves (Dey *et al*, 2001). One of the main difficulties in using this algorithm is the selection of the predefined angle parameter. According to Dey and Wenger (Dey *et al*, 2001), the performance of the algorithm is "heavily dependent" on the value of the predefined angle. The shape of the curve to be reconstructed and the sampling properties of the curve usually determine this angle. In addition, methods for solving the Traveling Salesman Problem (TSP) were also used by Giesen (Giesen, 1999). Althaus and Melhlhorn (Althaus *et al*, 2000) also used travelling Salesman Problem (TSP). This method always generates a single curve due to the nature of the TSP algorithm.

All algorithms mentioned above have been proven correct for certain types of curves by using rigors mathematical procedure. The reconstruction results are homeomorphic to the original curve and correct. However, despite the successes of such algorithms that have already been developed for problems of the first group, it is still an open task to develop an algorithm that could deal with the second group problems, for it is not known much information about the topology of the curve and therefore any assumption about the curve cannot be made. The only information provided to a curve reconstruction algorithm is a set of sampling points and no knowledge about the topology of the curve is available. The algorithm developed for this type of problems will be able to distinguish and reconstruct multiple simple curves that may be open, closed, and/or with sharp corners. As such, some concepts that applied to problems of the first group, such as uniformly distributed or non-uniformly distributed sampling, cannot be applied directly to the second type of problems. As a result, most of the existing algorithms fit well for problems of the first group but not for the second category. On the contrary, an ideal curve reconstruction algorithm for problems of the second group should not make any assumption about the type of curves to be reconstructed and the characteristics of the sampling points.

The objective of the present thesis is to develop a curve reconstruction algorithm for the second category problem. Before a more detailed review is given for the existing curve reconstruction algorithms, concepts used in the curve reconstruction will be introduced.

2.1 Preliminaries

2.1.1 Voronoi diagram:

Voronoi diagram, named after Georgy Voronoi (Voronoi, 1907), is a decomposition of a metric space determined by distances to a specified discrete set of objects in the space, e.g., by a discrete set of points. Voronoi diagrams that are used in geophysics and meteorology to analyze spatially distributed data are also called Thiessen polygons after American meteorologist Alfred H. Thiessen. In the simplest and most common case, given a set of points $S \in R^2$, and the Voronoi diagram for S is the partition of the plane that associates a region V(p) with each point p from S in such a way that all points in V(p) are closer to p than any other points from S. A formal definition is given as follows (Dey, 2004).

Definition of Voronoi diagram (Dey, 2004): Given a point set $P \in R^d$ the Voronoi diagram V_P of P is a collection of Voronoi cell V_p, for each point $p \in P$, where

$$V_p = \{x \in R^d \mid \| x - p \| \leq \| x - q \| \; \forall q \in P\}$$

Figure 4 shows an example of 2D Voronoi diagram.

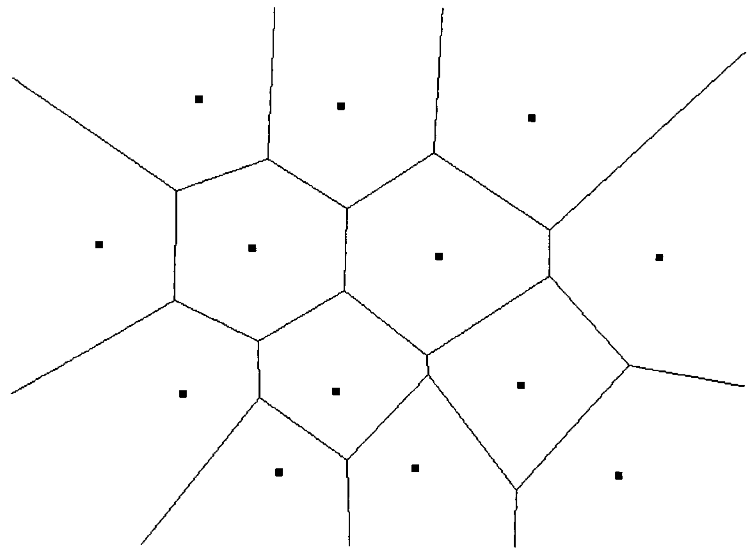


Figure 4 Example of Voronoi diagram

2.1.2 Delaunay triangulation:

Delaunay triangulation or Delone triangularization for a set P of points in the plane is a triangulation DT (P) such that no point in P is inside the circumcircle of any triangle in DT (P). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation. A formal definition is given as follows.

Definition of Delaunay Triangulation (Dey, 2004): The Delaunay triangulation of a point set $P \in R^d$ is the simplicial complex D_p so that a simplex with vertices $\{p_1, p_2, \ldots p_k\}$ is in D_p if and only if $\bigcap_{i=1,..k} V_{p_i} \neq \Phi$.

Figure 5 is an example of Delaunay triangulation. It is observed that the dual graph for a Voronoi diagram corresponds to the Delaunay triangulation for the same set of points S.

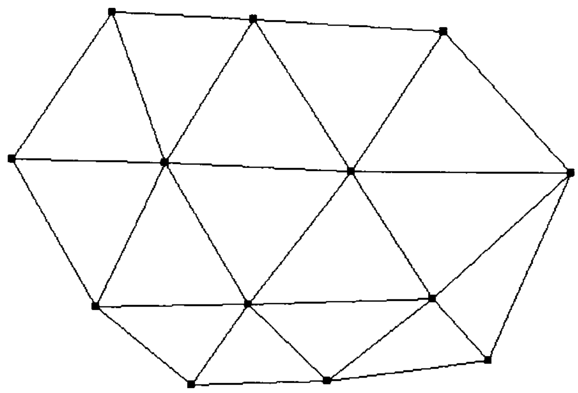


Figure 5 Example of Delaunay triangulation

2.1.3 α-shape:

Edelsbrunner *et al* (Edelsbrunner *et al*, 1983) define the α-shape of a point set $P \in R^d$ as the underlying space of a simplicial complex called α-complex. The α-complex of P is defined by all simplices with vertices in P that have an empty circumscribing disk of radius α.

Imagine that using one of these sphere-formed ice-cream spoons by carving out all parts of the ice-cream block that can be reached without bumping into chocolate pieces, thereby even carving out holes in the inside (e.g. parts not reachable by simply moving the spoon from the outside). In such a way, α-shape can be obtained as shown in Figure 6.

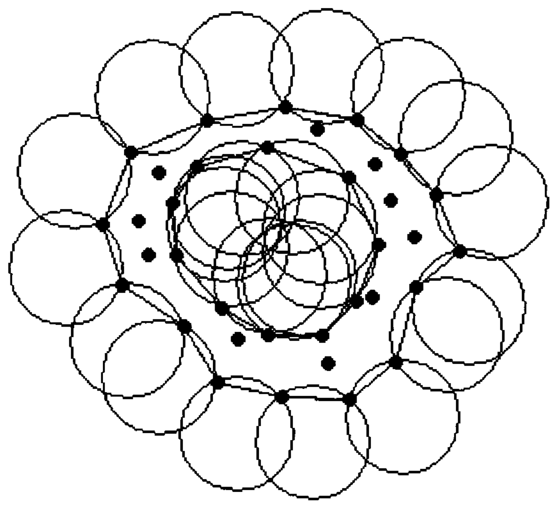
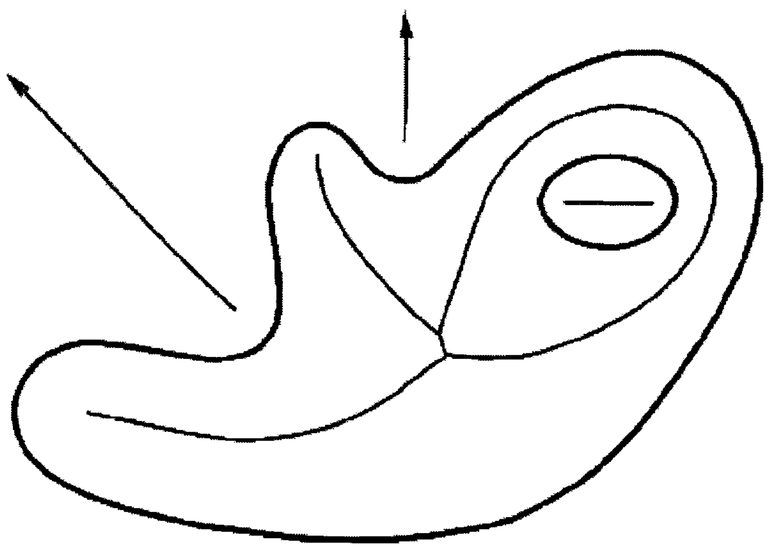


Figure 6 Example of α-shape

2.1.4 Medial axis and local feature size:

In many algorithms such as CRUST (Amenta *et al*, 1998), NN-CRUST (Dey and Kumar, 1999), Conservative-crust (Dey *et al*, 1999), and GATHAN (Dey *et al*, 2001), medial axis and local feature size are two key concepts by which sampling conditions are given.

Definition of Medial axis (Amenta *et al*, 1998)**:** The medial axis of a curve F is the closure of the set of points in the plane that has two or more closest points in F.

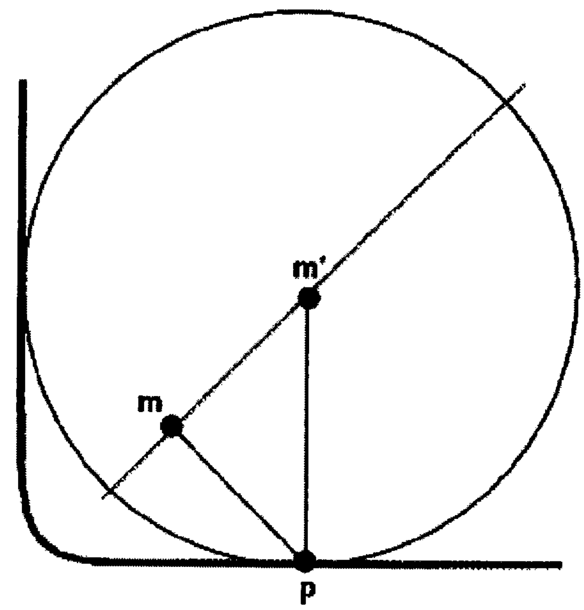


The light curves are the medial axis of the heavy curves.

Figure 7 Example of medial axis (Amenta *et al*, 1998)

Figure 4 shows the medial axis of a smooth curve. That includes components of the medial axis on either side of the curve, so that some components of the medial axis may extend to infinity. Since the medial axis is defined to be a closed set, it includes the centers of all empty osculating disks (the empty disks tangent to F with matching curvature), which are its limit points (Amenta *et al*, 1998). The medial axis can also be thought of as the Voronoi diagram generalized to an infinite set of input points.

Definition of local feature size (Amenta *et al*, 1998): The local feature size, LFS(p), of a point p ∈ F is the Euclidean distance from p to the closest point m on the medial axis.



$LFS(p)$ is the distance $d(p, m)$, *not* the perpendicular distance $d(p, m')$ to the center of the largest empty tangent ball at p.

Figure 8 Example of local feature size (Amenta *et al*, 1998)

Figure 4 shows an example of local feature size. It should be noted that this definition of local feature size depends on both the curvature at p and the proximity of nearby features because it uses the medial axis (Amenta *et al*, 1998).

2.2 Previous Work

In this section, the present thesis will review the existing curve reconstruction algorithms. They include Crust, NN-CRUST, Conservative Crust, CATHAN, and TSP.

2.2.1 ABE algorithm (CRUST)

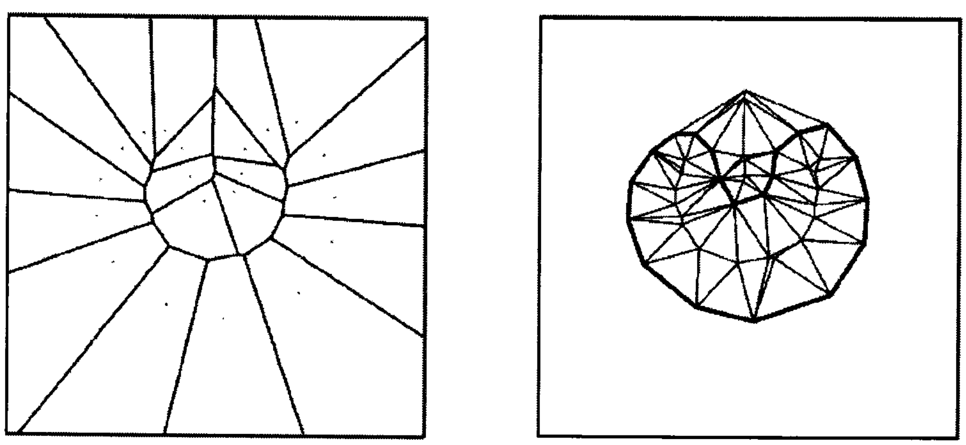
ABE algorithm (Amenta *et al*, 1998) is the first algorithm for curve reconstruction that applies Delaunay and Voronoi diagrams. The algorithm is mainly composed of three steps:

Step 1: Constructs the Voronoi diagram VD of the points in S,

Step 2: Constructs a set L≡S ∪ V, where V is the set of vertices of VD,

Step 3: Constructs the Delaunay triangulation DT of L and makes G the graph of all edges of DT that connect points in S.

It is proven that CRUST correctly reconstructs a collection of closed smooth curves Γ if the sample set S is an ε-sample for ε< 0:252 (Funke,2001).



A Voronoi diagram of a point set *S* and the Delaunay triangulation of *S* ∪ *V*, with the crust highlighted.

Figure 9 Example of ABE Algorithm (Amenta *et al*, 1998)

Figure 9 is an example of curve reconstruction using ABE algorithm. Figure 6 shows the detailed flow chart of the algorithm.

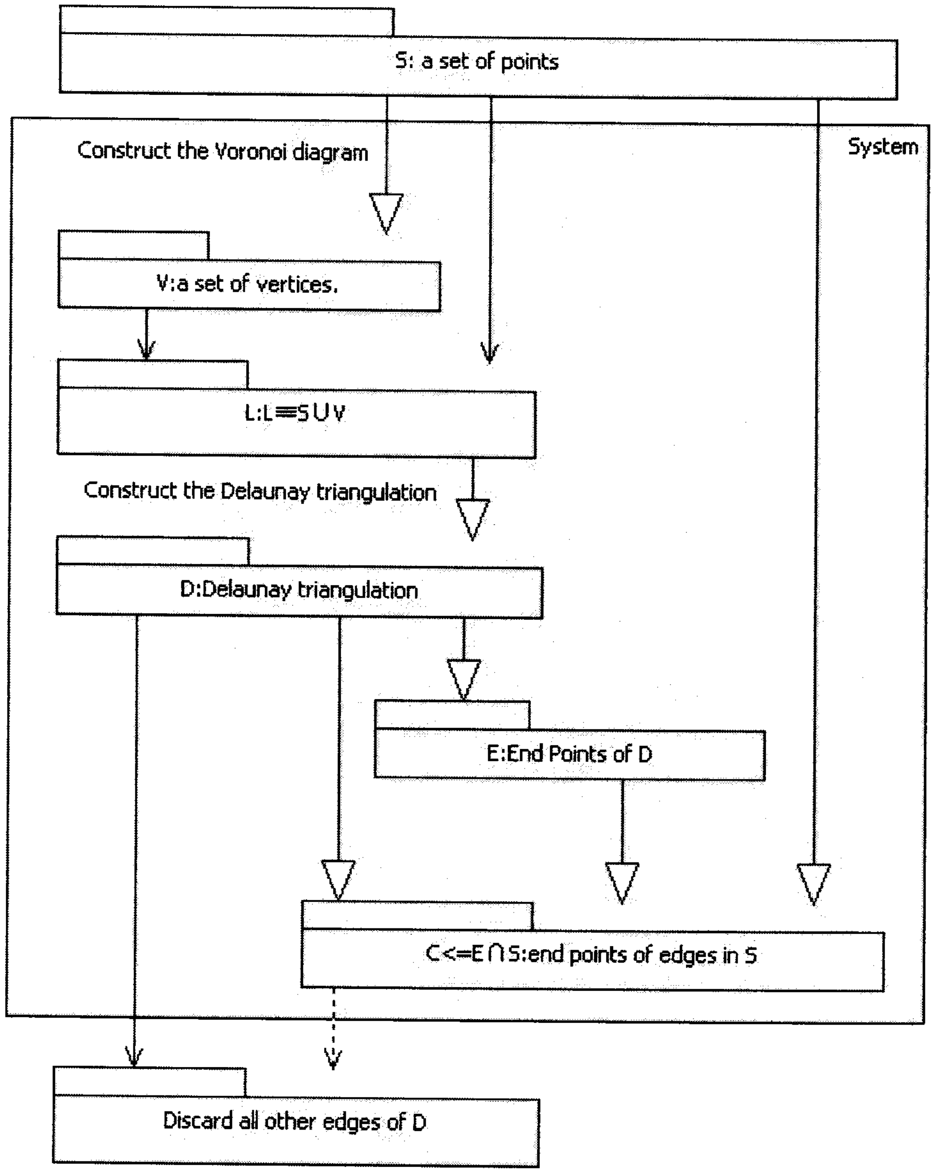


Figure 10 Flow chart of ABE algorithm

2.2.2 Reconstruction of smooth curves (NN-CRUST)

NN-CRUST (Dey and Kumar, 1999) computes crust in three steps:

Step 1: Compute the set of edges N that connect nearest neighbors in P.

Step 2: Let Pa be a point that is incident with only one edge E_e in N. Compute the shortest edge incident with Pa among all the edges that make an angle more than $\pi/2$ with E_e. Let D be the set of all such edges.

Step 3: Output $G = N \cup D$.

Both steps 1 and 2 can be performed on the edges of the Delaunay triangulation. An Example using NN-CRUST is shown in Figure 11.

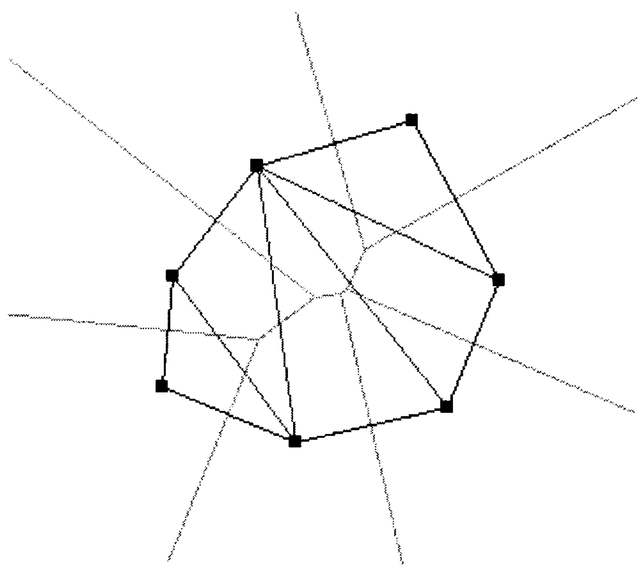


Figure 11 Example of NN-CRUST algorithm

This algorithm requires the sampling density $e \leq 1/3$. It can solve the problem of curve reconstruction that takes a set of sample points on a single smooth closed curve C. It cannot treat the sharp corner, intersection problems and endpoint problems. The algorithm does not need parameters and can be extended to three dimensions but it needs special care of endpoints.

All the reconstruction calculations are in the edges of the Delaunay triangulation. If one of the sample points is missing or some noise points are added, the Delaunay triangulation will have some wrong edges, which result in a wrong connection.

The algorithm requires that Delaunay triangulation and medial axial should be computed first. The computation of Delaunay triangulation needs O (nlogn). Then steps 1, 2, 3 all need O (n) in R^2. However, in step 2, there should be at least n iterations, which means that step 2 actually needs O (n^2). Therefore, the total time complexity should be O (n^2). Meanwhile, since Delaunay triangulation is needed, the space requirement in this algorithm is much larger than those algorithms without using Delaunay triangulation.

2.2.3 Reconstruction of curves with or without boundary (Conservative-crust)

The Conservative-crust algorithm (Dey *et al*, 1999) takes as input the point set P and a parameter ρ . It outputs a plane graph G with vertex set P and a curve that witnesses the correctness of the reconstruction. In details, the algorithm is described as follows.

Algorithm Conservative-Crust(P; ρ)

1. Compute the Delaunay triangulation D(P).

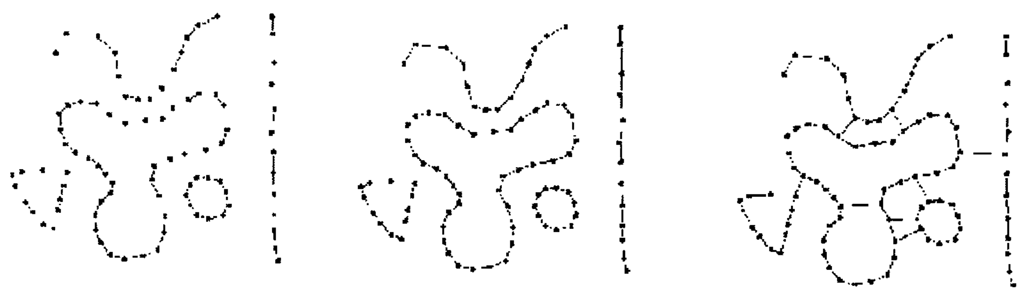
2. Extract the Gabriel graph G(P) $\subseteq$D(P).

3. Compute the graph $G_0 \subseteq$ G(P), where e $\in$ G(P) is in G_0 iff B(e,l(e)/ ρ) is a V-disk.

4. Refine G_0 into G by eliminating any $e \in G_0$ for which $X = B(e, l(e)/4\rho) \cap G_0$ contains a degree-0 vertex or a degree-1 vertex not connected to e within X. Repeat until no such edge remains.

5. Output G as defined above.

The algorithm computes the Delaunay triangulation, and then deletes some of invalid edges. The idea of algorithm is how to figure out the invalid edges. The algorithm use Gabriel graph, if B $(e, l(e)/\rho)$ is a V -disk. B $(e, l(e)/4\rho)$ without a degree-0 vertex or a degree-1 vertex to delete invalid edges. An example is shown in figure 8.



Effect of the parameter ρ: 1/2, 1, 2.5.

Figure 12 Example of Conservative-crust Algorithm (Dey *et al*, 1999)

The algorithm uses a global parameter ρ and can compute multiple curves. However, it cannot treat the sharp corners and intersection problems. Another problem with this algorithm is that if the global parameter ρ is too small, some of points cannot be connect; if ρ is too large, some parts of crust will be over-connected.

The accuracy of the algorithm heavily depends on the global parameter ρ.

The computational time of the algorithm is O(n log n). The algorithm needs to calculate order-1, order-2 and order-3 Voronoi diagrams, Delaunay triangles.

2.2.4 Reconstruction of curves with corners (GATHAN)

The algorithm GATHAN (Dey *et al*, 2001) is described in the following.

Gathan(P,α,ρ);

Compute the Voronoi diagram V_P ;

For each p $\in$ P do

Compute the pole and the normal line l_p.

Let E be the set of Delaunay edges incident to p satisfying the following Conditions:

A. normal to each e $\in$ E makes an acute angle less than α with l_p ,

B. h/l> ρ where l and h are the lengths of e and its dual Voronoi edge.

Keep only the smallest edges pq $\in$ E and ps $\in$ E on each side of l_p.

End for

Delete any edge that is not among the smallest two edges incident to a sample point.

End

The algorithm first computes all the Delaunay edges. Then, it computes the normal line l_P for each edge and pole for each point. Usually, the pole is farthest Voronoi vertices from the samples in their respective Voronoi cells if a Voronoi cell is bounded, otherwise the pole is estimated to be the average of the directions given by the two unbounded rays. Amenta and Bern (Amenta et al, 1998) observed that a Voronoi cell V_p for a sample p is elongated along the normal direction if the sampling condition (R) is satisfied. Therefore, the direction of the pole is that along which Voronoi cell is elongated.

This algorithm can compute non-smooth curve problems for curve reconstruction. It can compute multiple curves with shape corners and endpoints, but it cannot compute intersection problems and it is hard to extend to 3D.

This algorithm needs two parameters: α and ρ. α determines how sharp the corner can be where ρ determines how detailed the outline of the crust will be. Therefore, the algorithm is very sensitive to two parameters: α and ρ.

Obviously, some parts of crust may have wrong edges because the algorithm has to choose a compromised value. In some situations, if shape corner and endpoint are not very clear, the algorithm may compute wrong corner.

The computation of Delaunay triangulation needs O (nlogn). The algorithm computes Delaunay triangulation, the normal line and Voronoi edge.

2.2.5 TSP algorithm

Some researchers view curve reconstruction as a TSP problems (Althaus *et al*, 2000).

1. Compute the Delaunay triangulation of the given points. Delete parallel edges and compute the length of the remaining edges.

2. Create an L_p for this graph and add the degree constraints to the L_p.

3. Iterate solving the L_p and computing violated cutest constraints until either the current solution is integral and the graph is connected, or any violated constraint cannot be found.

Since TSP algorithm always generates a single curve, it is not suitable for general curve reconstruction problems. In addition, TSP algorithm in computational complexity theory is classified as NP-hard.

Table 1 is a comparison of four algorithms.

Name	**Scope**	**Advantage**	**Disadvantage**	**Complexity**
ABE	Multiple smooth curve	Simplicity	It cannot treat the sharp corner, and endpoint problems.	O(n log n)
NN-CRUST	Single smooth closed curve	Simplicity	It cannot treat the sharp corner, intersection problems and endpoint problems.	O(n log n)
Conservative-crust	Multiple smooth curve	The algorithm can compute multiple curves.	It cannot treat the sharp corner, intersection problems.	O(n log n)
GATHAN	Multiple nonsmooth curve	The algorithm can compute multiple curves	It cannot treat intersection problems.	O(n log n)

		with sharp corner		
TSP	Single smooth closed curve	The algorithm can compute single curves with sharp corner	It cannot treat multiple curves and intersection.	NP-hard

Table 1 Comparison of ABE, NN-CRUST, Conservative-crust and GATHAN

All the algorithms discussed above are based on the computation of Delaunay triangulation. The difference between the five algorithms lies in how they compute valid crust edges and delete invalid edges through different methods. If the algorithms use global parameters, they may lose local feature property. Moreover, it is not accurate to compute different curves by using the same global parameter.

Chapter 3
Distance-Based Parameter-Free Algorithm

3.1 Introduction

The Design Lab (CIISE, Concordia) developed a distance-based parameter-free algorithm for curve reconstruction (Zeng *et al*, 2007). This chapter will review this algorithm. The distance-based parameter-free algorithm, named DISCUR, is a simple, efficient, and parameter-free algorithm to reconstruct curves from unorganized sample points. The scope of this algorithm includes multiple simple curves that may be open, closed, and/ or with sharp corners. The algorithm is able to deal with the cases that the given information is only point cloud with or without knowing the sampling curves.

The algorithm originates from two observations made concerning the human visual system: 1) two closest neighbors tend to be connected, and 2) sampling points tend to be connected into a smooth curve. Based on these two observations, the neighborhood features of a curve and the statistical properties of a set of samples are studied. A general vision function is proposed to quantify the connectivity of a point with a curve segment. The necessary and sufficient sampling conditions for the algorithm to work correctly were included in two theorems, which define the precise scope of the algorithm.

Section 3.2 will introduce the criteria of the connectivity of points, followed by the distance-based parameter-free algorithm for curve reconstruction. The sufficient and necessary sampling conditions are explained in Section 3.4. The algorithm (DISCUR) is developed and implemented on the curve reconstruction platform designed by myself. This general platform will be briefly introduced in 3.5.

3.2 Criteria for the connectivity of points

3.2.1 Notations

A finite set of n points is denoted as $P = \{p_1, p_2, ..., p_n\}$. The quasi-distance between two point sets P and S can be defined as

$$d(P,S) = \min\{\|p_i - s_j\|; i = 1,...,k \; and \; j = 1,...,m\}. \quad (3.1)$$

A polyline is a continuous and piecewise linear curve. It can be denoted by $T = [y_1, y_2, ..., y_n]$. In a $< h_d, \sigma_d >$ *–distributed* polyline, $< h_d, \sigma_d >$ present

$$h_d = \frac{\sum_{i=1}^{m-1} \|y_i - y_{i+1}\|}{m-1} \quad (3.2)$$

$$\sigma_d = \sqrt{\frac{\sum_{i=1}^{m-1} (\|y_i - y_{i+1}\| - h_d)^2}{m-1}} \quad (3.3)$$

Symbols $< h_d, \sigma_d >$ are distance mean and standard deviation of a polyline.

3.2.2 Requirement for curve reconstruction algorithms

A curve reconstruction problem can be reduced to a combination of connecting points and connecting a point with a curve. Thus, the criteria for the connectivity should be investigated.

The major challenges in designing a curve reconstruction algorithm are: first, how to handle multiple features such as boundaries, sharp corner, intersection, and outlier points. Secondly, what criteria should be chosen for evaluating the correctness of reconstructed curves.

In this research, it has be assumed that the points should be connected in a pattern that is natural to the human perception. To illustrate, two general observations about human vision are given below:

- Nearness: the nearest neighbors tend to be connected.
- Smoothness: the connected curve tends to be as smooth as possible.

The nearness observation implies that two neighbors could be connected if their distance is small enough. The distance-based parameter-free algorithm (DISCUR) implies smoothness by considering only nearness under the assumption that if a curve can be reconstructed in the way that human beings perceive it, then this curve has the best smoothness among all its possible reconstructions.

By using the geometric features of samplings, the observation of nearness shows the connectivity of two points and the criterion of the connection.

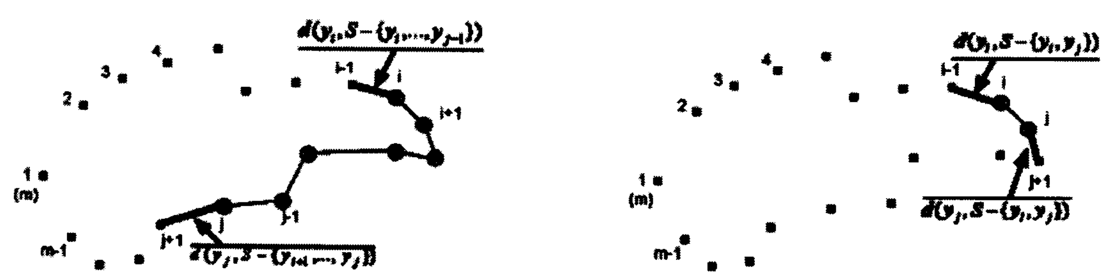
The proposed geometric features for a sample set of a simple smooth closed curve in R^n can be described as follows: for a given finite set S in R^n with $|S| =$ m - 1 > 2, there is an order of $y_1, y_2, \ldots, y_{m-1}$ for all points in S such that the closed polyline T = [$y_1, y_2, \ldots, y_m$, $y_m = y_1$] satisfies:

(NF1) (Neighborhood Feature 1): For all $z \in s$, if $w \in s$ is its nearest neighbor, then z and w are adjacent vertices on the polyline T.

(NF2) (Neighborhood Feature 2):

(a) For any i, j with $1 \le i < i+1 < j \le m-1$, if d($y_i$, S − {$y_i$,..., y_{j-1}})≤ d(y_j, S − {y_{i+1},..., y_j}) (or d(y_i, S − {y_i,..., y_{j-1}})≥ d(y_j, S − {y_{i+1},..., y_j})) then y_{i-1} (or resp. y_{j+1}) is the unique point in S so that d(y_i, y_{i-1}) = d(y_i, S − {y_i,..., y_{j-1}}) (or resp. d(y_j, y_{j-1}) = d(y_j, S − {y_i,..., y_{j-1}}). specially, if d(y_i, S − {y_i,..., y_{j-1}}) = d(y_j, S − {y_i,..., y_{j-1}}) then y_i and y_j are the nearest neighbor to each other.

(b) For any i; j with $1 \le i < i+1 = j \le m-1$, if d($y_i$, S − {$y_i$, y_j}) ≤ d(y_j, S − {y_i, y_j}) (or d(y_i, S − {y_i, y_j}) ≥ d(y_j, S − {y_i, y_j})), then y_{i-1} (or resp. y_{j+1}) is the unique point in S such that d(y_i, y_{i-1}) = d(y_i, S − {y_i, y_j}) (or resp. d(y_j, y_{j+1}) = d(y_j, S − {y_i, y_j})).



a) Neighborhood feature (NF2a)   b) Neighborhood feature (NF2b)

Figure 11 Illustration of neighborhood features (Zeng *et al*, 2007).

Figure 11 a) and b) illustrate neighborhood features (NF2), which implies the characteristics of a point adjacent to one of the boundaries of an open curve belonging to the closed polyline T.

The nearest neighbors tend to be connected by human vision, but some curves should be open. It also reflects another aspect of human vision. For example, in Figure 12, the reconstructed open curve is more acceptable than the closed one.

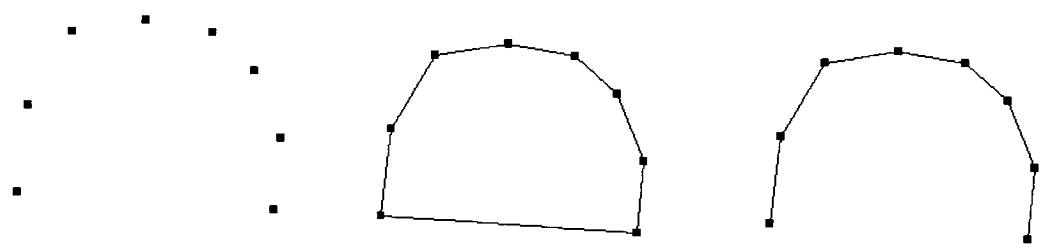


Figure 12 Closed and open curves

Thus, a criterion should be defined to determine when two nearest neighbors can be connected if the reconstructed curve should be natural to the human vision. Zeng *et al* (Zeng, *et al*, 2007) developed a dynamic vision function to define the range of connectivity for a point p to be connected to an existing curve segment T_q. A general form of this vision function can be written as follows:

$$E[p, T_q] = f(p, V) \tag{3.4}$$

where V is a vector that comprises statistical properties of the curve segment T_q. If $d(p, q) < E[p, T_q]$, then p and q can be connected. The function can be obtained through experiments or through observations. A distance-based parameter-free algorithm for curve reconstruction (Zeng *et al*, 2007) gives a formula based on observations about human vision.

$$E[p, T_q] = h_d \frac{h}{s}(1 + \frac{h_d}{\sigma_d})^{\frac{h_d}{\sigma_d}} \tag{3.5}$$

3.3 Distance-based parameter-free algorithm

Before introducing the distance-based parameter-free algorithm, the present thesis first introduce a simple distance-based algorithm. This simple algorithm implements the two observations without considering the vision function.

Algorithm: SimpleSmoothClosedCurve(Input:S;Output:T)

find the nearest neighbors $x_1, x_2 \in S$.

$T \Leftarrow [x_1, x_2], S \Leftarrow S - \{x_1, x_2\}$.

if d(x_1, S) < d(x_2, S) then

find $x_0 \in S$ such that $\|x_1 - x_0\| = \min\{\|x_1 - x\|;\ x \in S\}$

$T \Leftarrow [x_0|T]$, $S \Leftarrow S - \{x_0\}$

else

find $x_3 \in S$ such that $\|x_2 - x_3\| = \min\{\|x_2 - x\|;\ x \in S\}$

$T \Leftarrow [x_3|T]]$, $S \Leftarrow S - \{x_3\}$.

end if

$i \Leftarrow 3$

while T[1] $\neq$ T[i] do

if d(T[1], S$\cup$ {T[i]}) < d(T[i], S$\cup$ {T[1]}) then

find $x_0 \in S$ such that $\|T[1]-x_0\| = \min\{\|T[1]-x\|;\ x \in S\}$

T $\Leftarrow$ [x_0|T], S $\Leftarrow$ S − {x_0}.

else

find $x_{i+1} \in S$ such that $\|T[i]-x_{i+1}\| = \min\{\|T[i]-x\|;\ x \in S\}$

T $\Leftarrow$ [T|x_{i+1}], S $\Leftarrow$ S − {x_{i+1}}.

end if

$i \Leftarrow i+1$

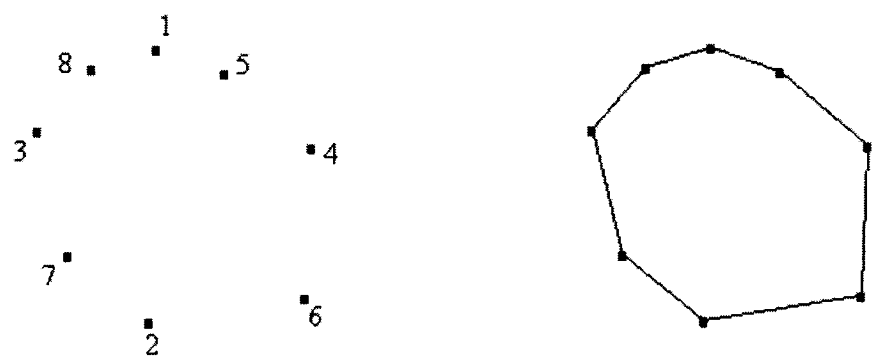
end while

Output T.

The basic idea of this algorithm can be explained as follows: first, find two different points $x_1, x_2 \in S$ that have the minimum distance in S. Second, find x_0, x_3 such that $\|x_1 - x_0\| = d(x_1.S-\{x_1,x_2\})$ and $\|x_2 - x_3\| = d(x_2.S-\{x_1,x_2\})$. If $\|x_1 - x_0\| < \|x_2 - x_3\|$, then x_0 is added to the head of T and the polyline is updated as T = [$x_0; x_1; x_2$].

Otherwise, x_3 is added to the tail of T and the polyline is updated as T = $[x_1; x_2; x_3]$. Third, from the new T (for convenience, the present thesis write T = $[x_1; x_2; x_3]$), find x_0 and x_4 such that $\|x_1 - x_0\| = d(x_1.S - \{x_1, x_2\})$ and $\|x_3 - x_4\| = d(x_3.S - \{x_2, x_3\})$. If $\|x_1 - x_0\| < \|x_3 - x_4\|$, then x_0 is added to the head of T and the polyline is updated as T = $[x_0; x_1; x_2; x_3]$. Otherwise, x_4 is added to the tail of T and the polyline is updated as T = $[x_1; x_2; x_3; x_4]$. Step by step, T can be expanded into a reconstructed closed polyline. Figure 11 shows a set of sample points and the reconstructed curve using this algorithm. The reconstruction of the closed polyline in Figure 13 is implemented according to the following procedures: [8; 1]→ [8; 1; 5] → [3; 8; 1; 5] → [3; 8; 1; 5; 4] → [7; 3; 8; 1; 5; 4] → [2; 7; 3; 8; 1; 5; 4] → [6; 2; 7; 3; 8; 1; 5; 4] → [6; 2; 7; 3; 8; 1; 5; 4; 6].

It is proven that any evenly sampled C^1-curve has the geometric features (NF1) and (NF2), and any curve that satisfies (NF1) and (NF2) can be reconstructed by Algorithm: SimpleSmoothClosedCurve [15].



a) Original sample points b) Reconstructed curve

Figure 13 Example of Distance-based algorithm

The algorithm DISCUR improves this algorithm by using the vision function $E[p,T_q] = h_d \frac{h}{s}(1+\frac{h_d}{\sigma_d})^{\frac{h_d}{\sigma_d}}$. Algorithm DISCUR has three main steps. Step 1 computes the Delaunay triangulation for the sample set S and do initialization. Step 2 processes all the Delaunay edges to determine which edges should be connected, which edges should be removed, and which edges should be retained for further procession. Step 3 processes the Delaunay edges retained in step 2 and completes the curve reconstruction.

Algorithm DISCUR (Sample Set: S)

1: Step 1 – Delaunay triangulation and initialization

2 Step 2 – Determining the connectivity of Delaunay edges

---

3 Step 3 – Updating the connectivity of Delaunay edges

4 Output the reconstructed curves

Algorithm DISCUR uses the vision function $E[p,T_q] = h_d \frac{h}{s}(1+\frac{h_d}{\sigma_d})^{\frac{h_d}{\sigma_d}}$ to check the connectivity between a point and a curve but not connect the point directly. By this way, all the Delaunay edges that meet the vision function will be connected and the Delaunay edges conflict with the vision function will be disconnected even if they are the shortest neighbors.

3.4 Necessary and sufficient condition for the sampling

Necessary and sufficient conditions for the algorithm DISCUR to work correctly are given in two theorems as follows (Zeng *et al*, 2007):

Theorem 1 Suppose that S is a set of sample points on a curve or a collection of curves. For every sample point $p \in S$, points $t_p^1, t_p^2 \in S$ are the two neighbors of p and $p \notin \{t_p^1, t_p^2\}$. Without loss of generality, it can be assumed that $r_p = d(p,t_p^1) = \max\{d(p,t_p^1), d(p,t_p^2)\}$. There exists an N_p, which is a subset of S, such that $N_p = \{q \in S : d(p,q) \le r_p \; and \; q \ne p, t_p^1, t_p^2\}$. The point p will be guaranteed to be connected to its neighbors t_p^1 and t_p^2 by Algorithm DISCUR, if and only if the following conditions are satisfied:

1) $\mathrm{r_p} < \max\{E[p,T_{t_p^1}], E[t_p^1,T_p]\}, \exists T_{t_p^1}, T_p$ (3.6)

2) $[|\mathrm{N_p}| = 0] \vee [\mathrm{d(p,q)} > \mathrm{r_q} \wedge q \neq t_p^2, \forall q \in N_p]$ (3.7)

where $E[p,T_{t_p^1}]$ and $E[t_p^1,T_p]$ are the vision function and $\mathrm{r_p} = d(q,t_q^1) = \max\{d(q,t_q^1), d(q,t_q^2)\}$

Theorem 2 Suppose that S is a set of sample points on a curve or collection of curves. For every boundary point $\mathrm{p} \in \mathrm{S}$, there exists a set $\mathrm{B_p}$, which is a subset of S, such that $\mathrm{B_p} = \{\mathrm{q} \in \mathrm{S} : [\mathrm{p,q}]$ is a Delaunay edge$\}$. The point p will not be connected to any point in $\mathrm{B_p}$ by Algorithm DISCUR, if the following tow conditions are satisfied:

1) All interior points are sampled according the Theorem 1

2) $d(p,q) \geq \max\{E[p,T_q], E[q,T_p]\}, \forall q \in B_p, T_q, T_p$

where $E[p,T_q]$ and $E[q,T_p]$ are the vision function.

3.5 Platform for curve reconstruction algorithm design

During the development and research of the curve reconstruction algorithm, I designed a platform for developing curve reconstruction algorithms. This platform implements almost all the major algorithms about curve reconstruction. The platform provides a toolkit to help researchers implement and evaluate new algorithms.

The function of the platform includes a graphic interface to show points and reconstruction result, Voronoi diagram, Delaunay triangulations, Circumcircle, multiple data I/O method, demonstration of curve reconstruction algorithms, sampling tool, statistic tools, and software interface for implementing new algorithms.

The interface of the platform is shown as below:

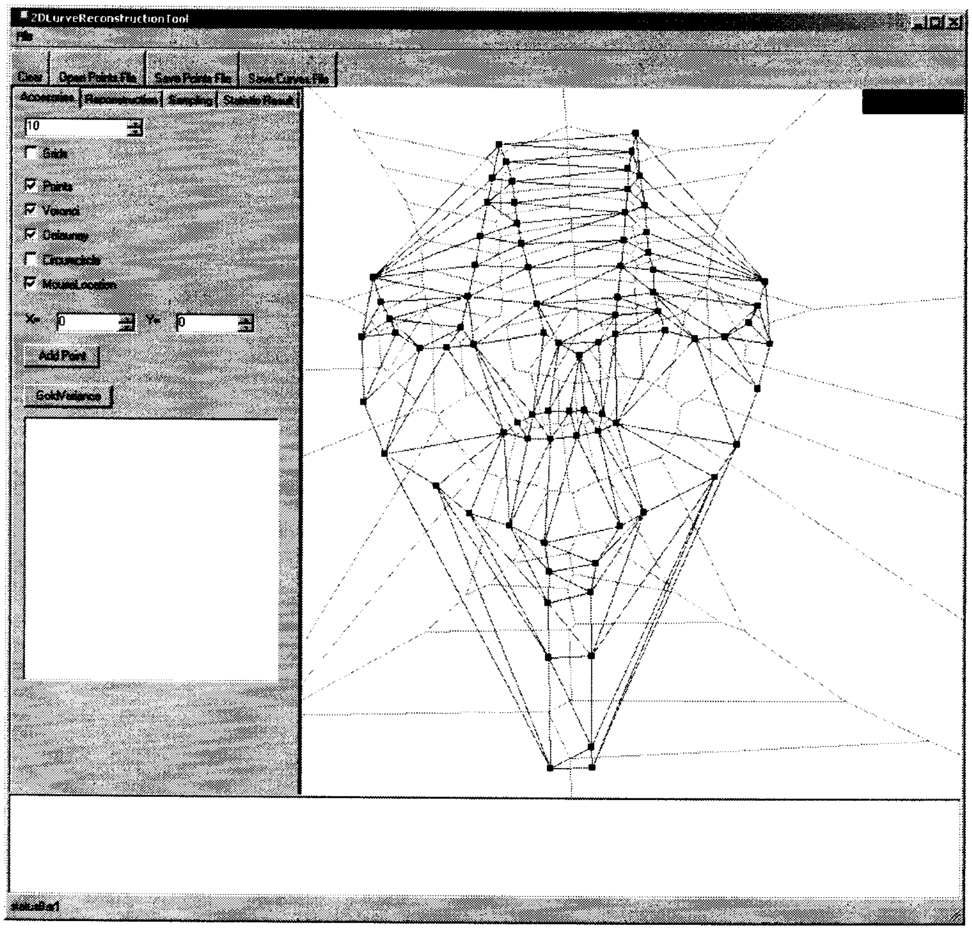
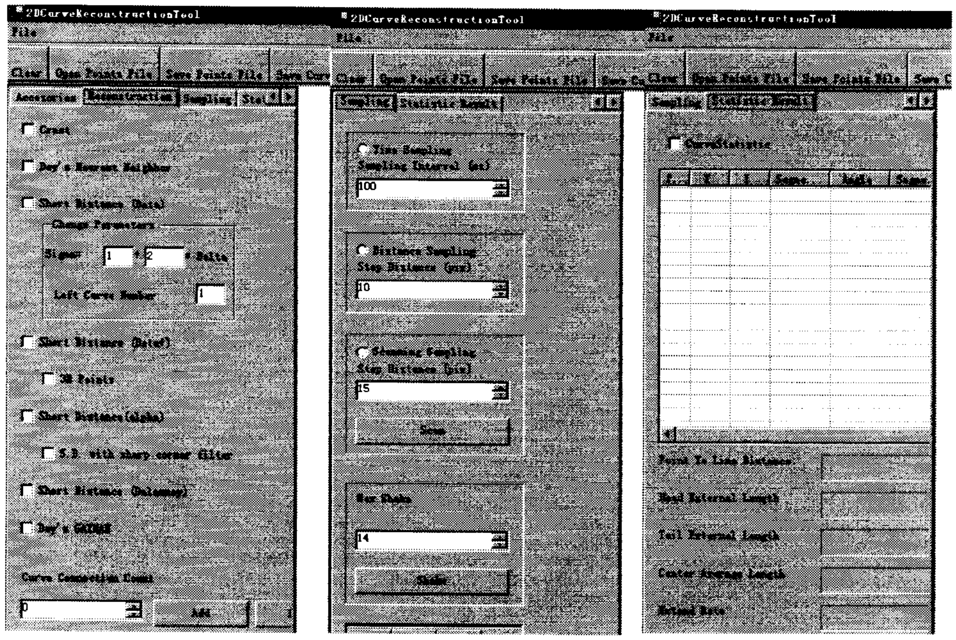
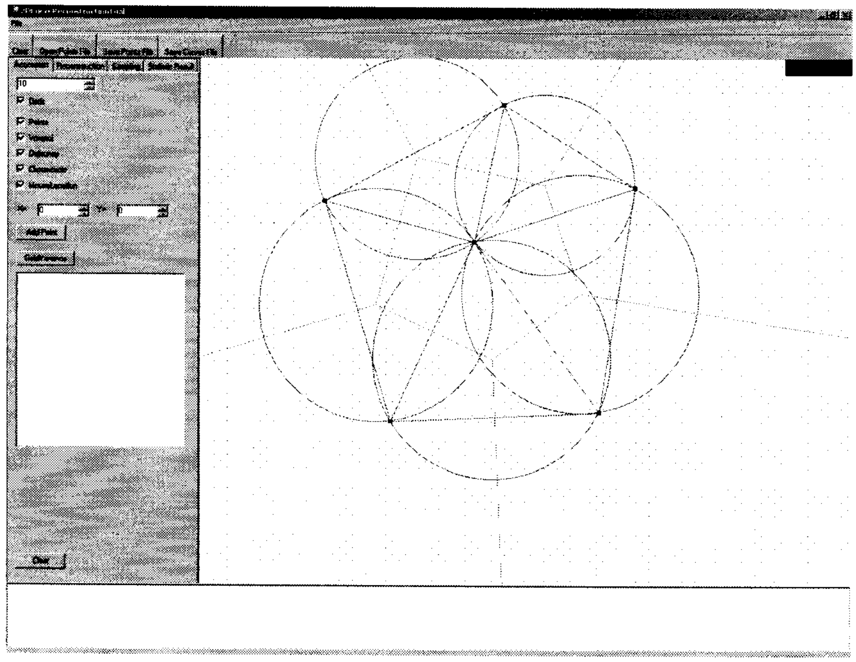


Figure 13 Interface 1 of the platform for curve algorithm design

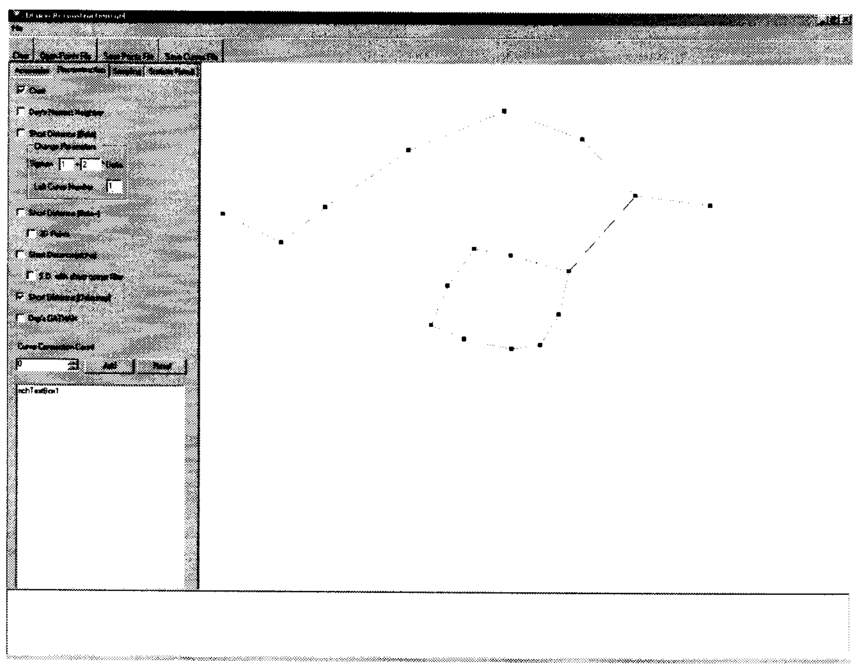
This interface provides general geometric calculation for the curve reconstruction algorithm. It includes points input, grid display, mouse location, Voronoi diagram, Delaunay triangulations, and Circumcircle in Figure 13. Inside the platform, it also provides the software interface for these calculations. When researchers design or test a new algorithm, they do not have to implement these algorithms. They can use the result directly. It can also load a picture as background for gathering points or comparing algorithm results.



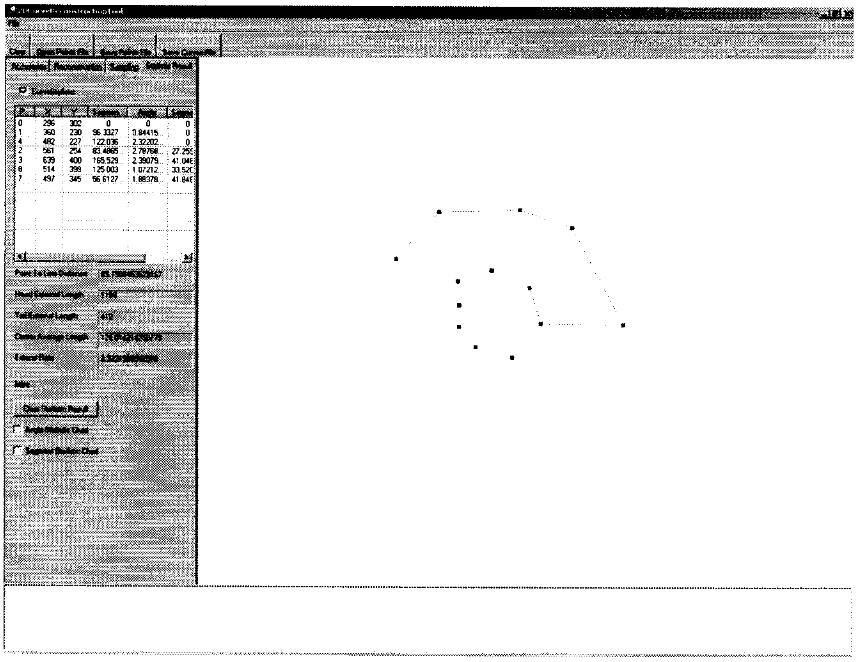
(a)



(b)



(c)



(d)

Figure 14 Interface 2 of the platform for curve algorithm design

These interfaces in Figure 14 (a) are designed for algorithm testing, sampling and statistical information collecting. The first panel provides almost all the current curve reconstruction algorithms; the second panel gives some simple sampling tools and the last one can measure the segment length and do some statistical calculation as is shown in Figure 14 (d). All the functions have been put into a software library for the future use. Researchers can use all the functions in the user interface. A geometric class library is also built. It helps the researcher implement the new algorithm by providing many data structures, classes, statistical functions and graphic output interface.

Chapter 4

Design of Vision Function for Curve Reconstruction

As was indicated in chapter 3, algorithm DISCUR uses the vision function $E[p, T_q] = h_d \frac{h}{s}(1+\frac{h_d}{\sigma_d})^{\frac{h_d}{\sigma_d}}$ in determining the connectivity between two points. The form of this vision function affects greatly the algorithm performance. Although this function works well, it does not consider the smoothness of curve explicitly. As a result, the sampling for sharp corners and two close curves would need extra effort (Zeng *et al*, 2007). This chapter aims to redesign this vision function based on experiments.

In understanding human vision, Gepshtein and Nanks did the research in size perception of vision and haptics (Gepshtein, Banks, 2003). Some other researchers studied the conscious experience and brain stimulation, like Baars, Ramsey (Baars, Ramsey, and Laureys, 2003) and Libet (Libet, 1966). Levin presented a geometric method for measuring and characterizing the perceptions of an observer of continuum of stimuli (Levin, 2000.). It is assumed from the previous work that there must be some relations between human vision and curve reconstruction. The present research attempts to explore the possibility to design experimentally the vision function $E[p, T_q]$. This chapter will present a framework of this experiment that the author tested himself. A more systematic experiment will be redesigned based on this exploration and is the future work of this research.

The rest of this chapter is organized as follows: the challenges of constructing the vision function are introduced in Section 4.1. The experimental framework for designing the vision function is given in Section 4.2. The last section summarizes this chapter.

4.1 Background of experiment

To design the vision function with the help of human vision, it is essential to understand the human perception (Morrone, Burr, 1988). The human visual system is a very complex system, which involves color, size, background, etc (Lubin, 1997). Since this chapter aims to design the vision function for curve reconstruction, the experimental framework will be developed in this specific context (Chellappa, Wilson, Sirohey, 1995).

The vision function should be a simple formula that can be computed efficiently. From the point of view of curve reconstruction algorithm, the inputs of the vision function must be a point and a curve. The output is the connectivity value that presents the relationship between the point and the curve. The two major challenges to find the vision function are summarized as below:

- The vision function must consider all the major factors that affect the characteristics of the potential curve to be reconstructed.
- The selection of regression model for the quantification of vision function.

To find vision function, all the potential factors in the context of curve reconstruction are firstly enumerated. Then experiments are designed and implemented to exclude the minor factors and to keep the major ones. Finally, the experiment of data of the major factors is used to construct the vision function and to evaluate the correctness of the function.

4.2 Factor analysis

Before designing the experiments, as many factors as possible should be considered. In looking into the problem, the following relevant factors are apparent: distance between points, angles at each curve node, curve extension (the maximum distance that a point could have from a curve for the point to be connectable to the curve), sharp corner, curve-to-curve distance, point size, point color, background color, shape of curve, curve length, etc. As the present thesis focuses only on the geometric factors in the context of curve reconstruction, only point size, point distance, and angles are considered. Intuitively, since when people reconstruct curves from points, the absolute distances and angles are not essential. Instead, only the mean and variance of these variables are more important, which implies the smoothness, homogeneous and continuousness of a curve. The present thesis uses experiments and regression to identify these factors, based on which a new algorithm is designed and implemented. The validity of these assumptions is evaluated according to the results of the algorithm in chapter 5.

If the experiments consider all the factors, the sample size for the experiment would be huge. Suppose that the 10 factors of points' distance, curve angle, curve extension, sharp corner, curve-to-curve distance, point size, point color, background color, shape of curve and curve length are considered and each factor has 4 levels, the total samples will be 1048576 (4^{10}). To finish the experiments with the speed of 1 second per sample, it would take 2 years without a break. For this consideration, experiments are done by starting with the simplest factors. If some factors do not affect the curve reconstruction result, they will be eliminated and then factors that are more complex are added to the experiments. Such procedure is repeated and only the most significant factors are kept. Finally, a multivariable nonlinear regression model is applied to identify the relationship between those major factors and connectivity.

The procedure of designing experiments for constructing the vision function is as follows:

Firstly, the simplest situation is considered, that is, a straight line is constructed by points. Through these experiments, the relationships among point size, point absolute distance, point's distance variance, curve extension and the interactions between these factors are analyzed by using analysis of variance (ANOVA). The objective of this experiment is to identify the major factors of straight line reconstruction.

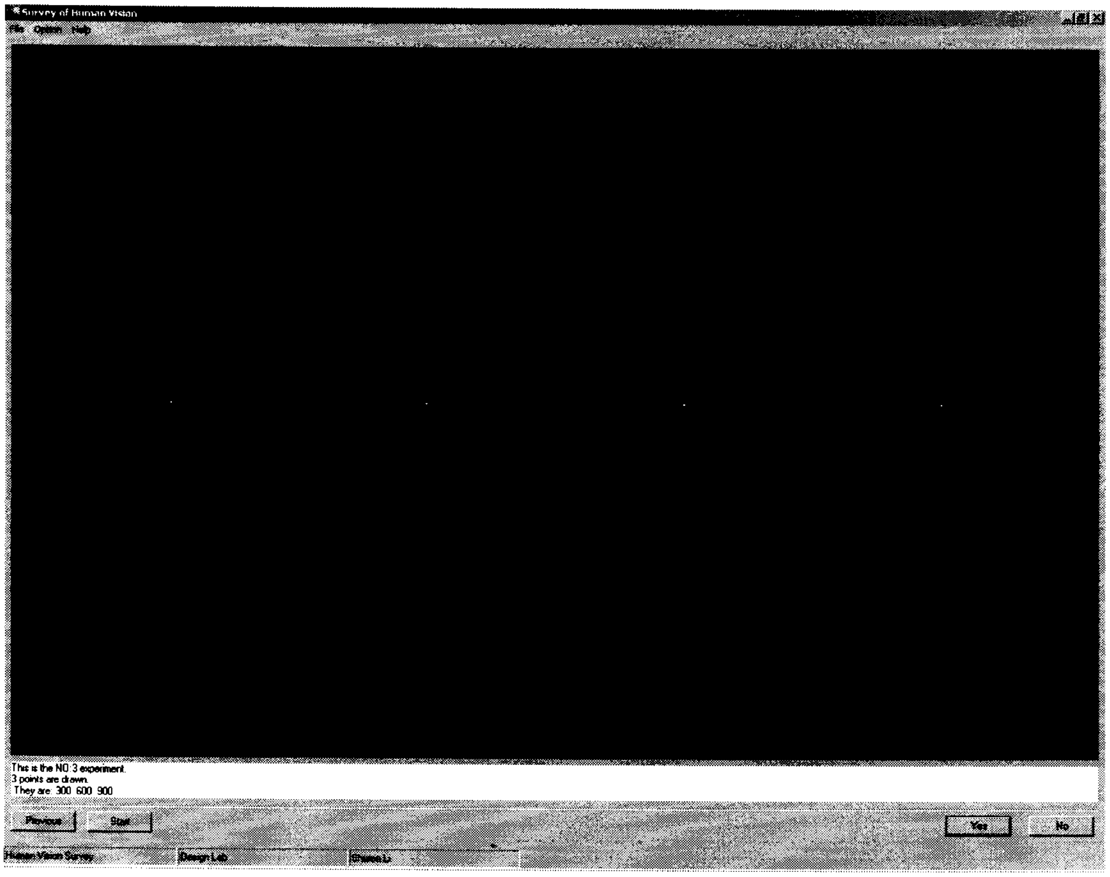
Secondly, keeping the major factors from step 1, the present thesis adds the curve angle to generate experiment of data. The objective of this experiment is to identify the major factors of curves with angle. Based on the result of these experiments, a multivariable nonlinear regression model can be applied to construct the vision function.

4.3 Experiments of straight line

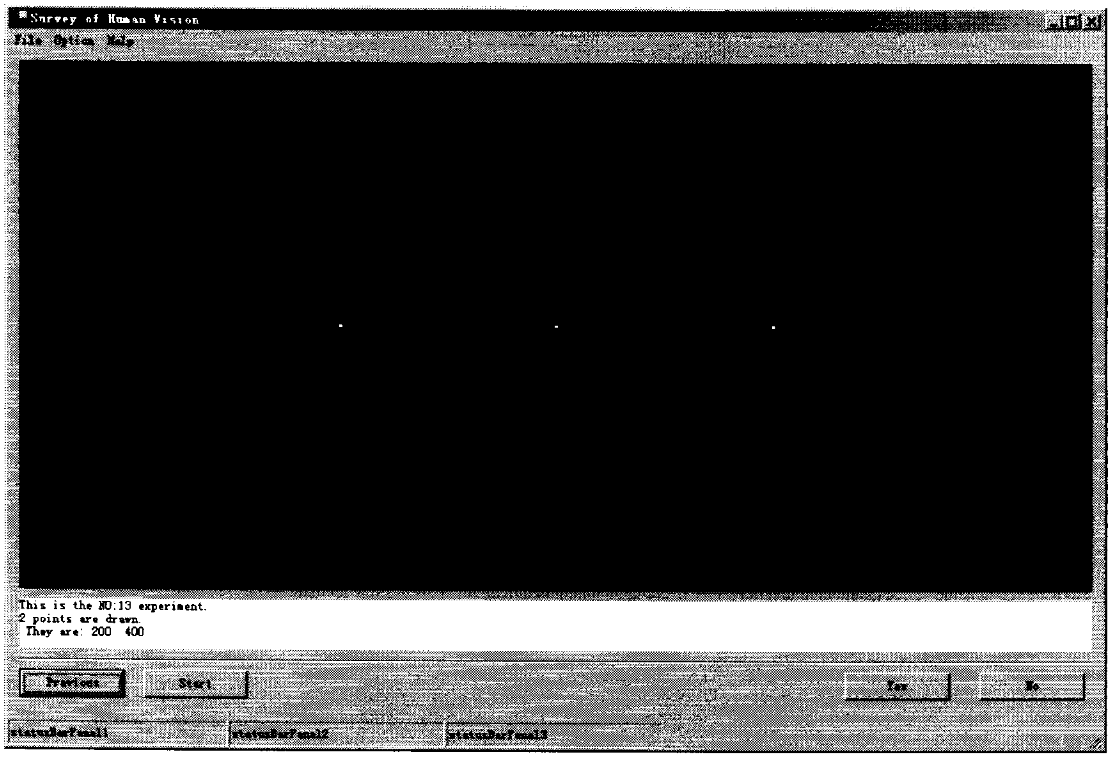
In this step, two experiments are designed. The objective of the first experiment is to exclude the influence of the point size. The point size should be considered because the sample points cloud is drawn on the screen. The point can be drawn in a small dot or a big disk. In the context of curve reconstruction algorithm, the point size should not be considered. Thus, before doing the experiment related to the curve reconstruction, an experiment should be done to make sure that effect of the point size in the experiments can be eliminated. The second experiment is related to the mean of point distance, standard deviation of point distance, extension rate (the scale rate of point to curve distance by the mean of point's distance of curve), and point count. This experiment can find out the significant factors in the straight line.

4.3.1 Experiment 1

The objective of this experiment is to find out the effect of point size on the vision function. The input is the horizontally distributed points of various sizes. The output is the connection of the points. A software system is developed for this experiment, the interface of which is shown as below:


Survey of Human Vision
This is the NO:3 experiment.
3 points are drawn.
They are: 300 600 900
Previous
Start
Yes
No
Human Vision Survey
Design Lab

(a)



(b)

Figure 15 Survey of simple curve's extension

The point cloud is created with the following factors: point size, point's absolute distance and the point count. The point size distributes in 2 pixels, 3 pixels and 4 pixels. The distance between two points distributes in 200 pixels, 300 pixels and 400 pixels. The point count distributes in 2, 3 and 4. The points are generated and sorted randomly. The sample data is listed in the table below:

Index	Location				Size	Distance	Count
	1	2	3	4			
20	0	400	800	0	2	400	3
27	0	400	800	1200	4	400	4
17	0	300	600	0	4	300	3
18	0	300	600	900	4	300	4
24	0	400	800	1200	3	400	4

23	0	400	800	0	3	400	3
10	0	300	0	0	2	300	2
11	0	300	600	0	2	300	3
7	0	200	0	0	4	200	2
3	0	200	400	600	2	200	4
19	0	400	0	0	2	400	2
14	0	300	600	0	3	300	3
21	0	400	800	1200	2	400	4
1	0	200	0	0	2	200	2
4	0	200	0	0	3	200	2
16	0	300	0	0	4	300	2
15	0	300	600	900	3	300	4
12	0	300	600	900	2	300	4
25	0	400	0	0	4	400	2
26	0	400	800	0	4	400	3
9	0	200	400	600	4	200	4
13	0	300	0	0	3	300	2
6	0	200	400	600	3	200	4
22	0	400	0	0	3	400	2
2	0	200	400	0	2	200	3
8	0	200	400	0	4	200	3
5	0	200	400	0	3	200	3

Table 2 Sample data from experiment 1

There are a total of 27 trials in experiment. The experiment is repeated 9 times on IBM PC under the common working environment. The decision about the connectivity is made based on the points drawn on screen. The time spent on each decision is recorded to identify outliers. If an experiment takes too much time, it should be redone.

One of the experimental results is listed below.

Index	Time	Answer
20	19	1
27	15	1
17	11	1
18	11	1
24	22	0
23	12	0

10	10	0
11	8	1
7	22	0
3	9	1
19	14	0
14	10	0
21	9	1
1	14	1
4	12	1
16	8	1
15	8	0
12	9	1
25	30	0
26	63	1
9	8	1
13	5	1
6	12	0
22	15	1
2	15	0
8	14	1
5	7	0

Table 3 Sample result of experiment 1

As the experiment is repeated 9 times, the sum of answer is in 0 to 9. The sample data and sum of result was analyzed by using ANOVAN in MATLAB (R2006A).

The MATLAB analysis result is listed below.

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Points size	0.889	2	0.4444	2	0.1975
Points distance	146	2	73	328.5	0
Points counts	27.556	2	13.7778	62	0
Points size*Points distance	1.111	4	0.2778	1.25	0.364
Points size*Points counts	0.889	4	0.2222	1	0.4609
Points distance*Points counts	12.444	4	3.1111	14	0.0011
Error	1.778	8	0.2222		
Total	190.667	26			

Constrained (Type III) sums of squares.

Table 4 Analysis of variance of experiment 1

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Points size	3868.7	2	1934.33	4.36	0.0524
Points distance	4010.9	2	2005.44	4.52	0.0486
Points counts	3504.2	2	1752.11	3.95	0.0641
Points size*Points distance	4130.4	4	1032.61	2.33	0.1437
Points size*Points counts	2609.1	4	652.28	1.47	0.2972
Points distance*Points counts	5134.9	4	1283.72	2.89	0.0939
Error	3548.4	8	443.56		
Total	26806.7	26			

Constrained (Type III) sums of squares.

Table 5 Analysis of variance of time spent in experiment 1

In this experiment and all other experiments, both survey result and the time spent are recorded and analyzed. Table 5 shows how these factors affect the time spent in the experiments. Table 4 is used to determine the significant factor.

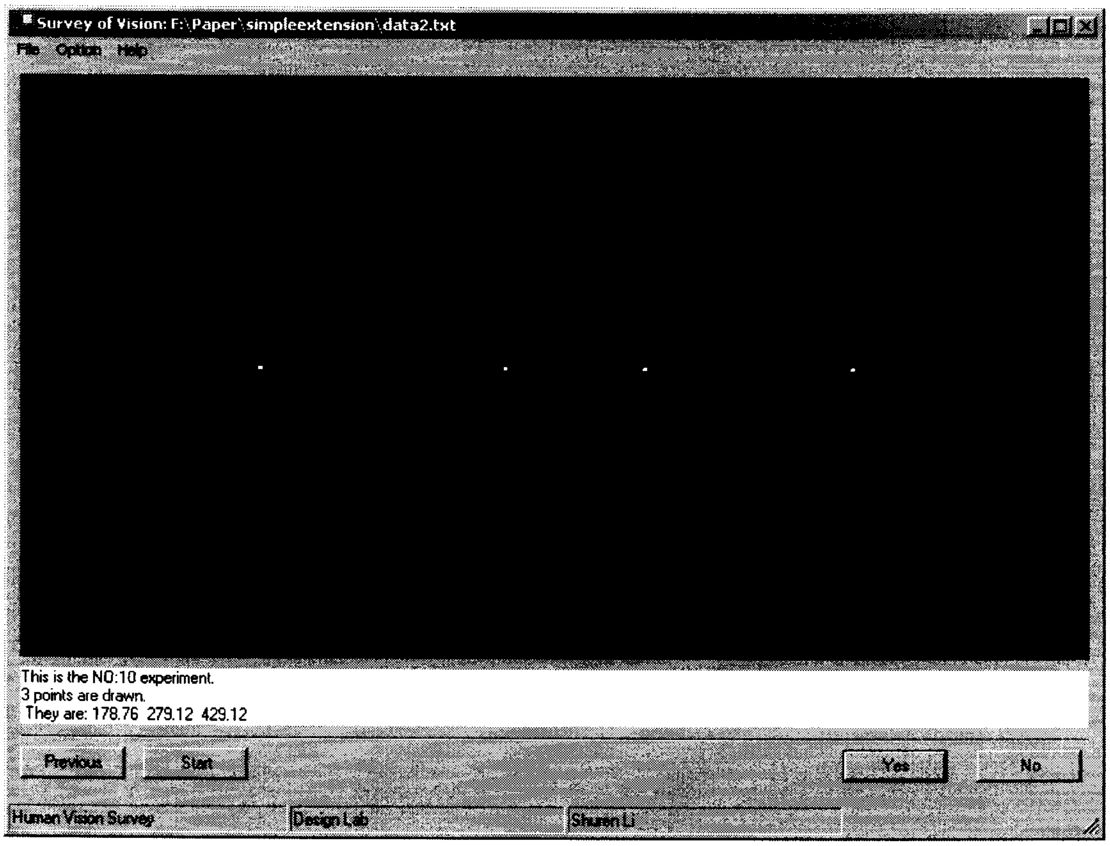
F value and Pr F have the same meanings as for multiple regressions. This is to be expected since analysis of variance is nothing more than the regression of the response on a set of indicators defined by the categorical predictor variable. The F Value or F ratio is the test statistic used to decide whether the sample means are within sampling variability of each other. This is the same thing as asking whether the model as a whole has statistically significant predictive capability in the regression framework. F is the ratio of the Model Mean Square to the Error Mean Square. Under the null hypothesis that the model has no predictive capability (all of the population means are equal, which means that the factor does not affect result), the F statistic follows an F distribution with p numerator degrees of freedom and n-p-1 denominator degrees of freedom. The null hypothesis is rejected if the F ratio is large. In this experiment and all other experiments, only the factors with large F value comparing to the others are significant factors.

From Table 4, the point's distance and point count play an important role in the experiment. The interaction between point's distance and point count also affect the decision. Therefore, the point's distance and point count are significant factors. Meanwhile, the point size does not affect the result significantly in the distribution of 2 pixels, 3 pixels and 4 pixels. Thus in the following experiments, the point size is not taken as a significant factor.

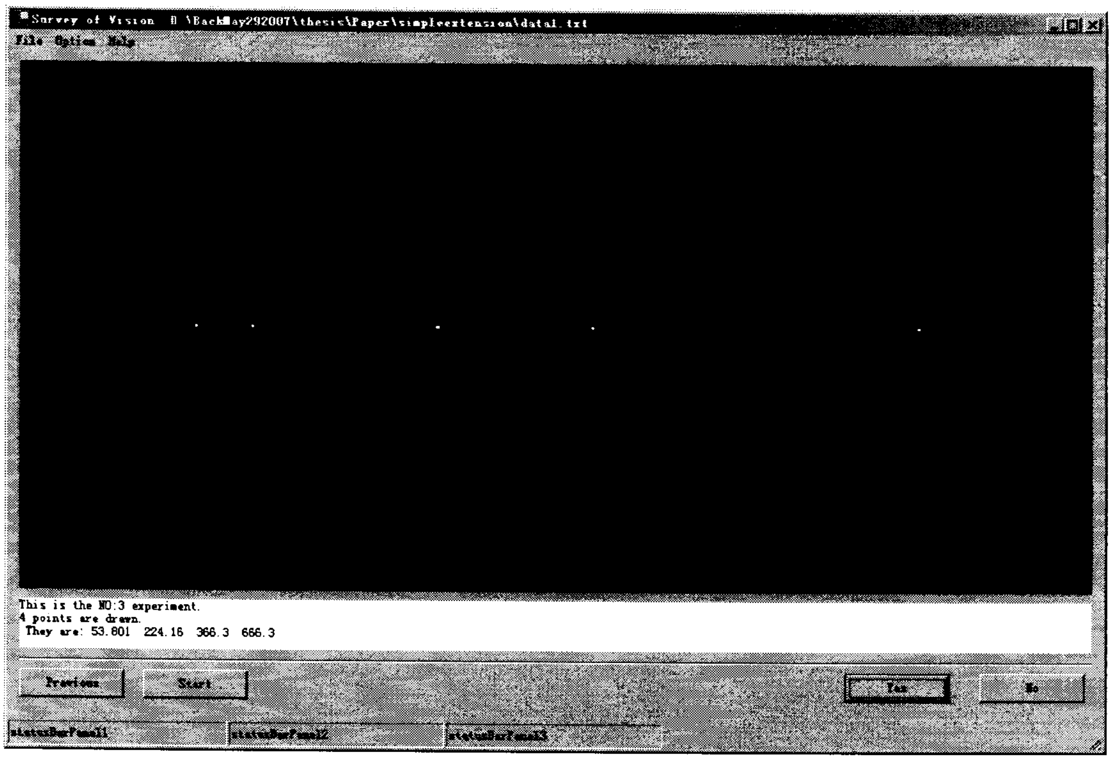
From Table 5, the time spent is related to 3 factors when a decision about the connectivity is made.

4.3.2 Experiment 2

In this experiment, the objective is to find out the significant factor in the straight line reconstruction. The input is a horizontally distributed point cloud. The output is the connectivity of those points. A software system is developed for this experiment, the interface which is shown below:


Survey of Vision: F:\Paper\simpleextension\data2.txt
File Option Help
This is the NO:10 experiment.
3 points are drawn.
They are: 178.76 279.12 429.12
Previous
Start
Yes
No
Human Vision Survey
Design Lab
Shuren Li

(a)



(b)

Figure 16 Experiment of the straight line reconstruction

The point cloud is created with the following factors: mean of point's distance, standard deviation of point distance, extension rate (the scale rate of point to curve distance by the mean of point's distance of curve), and point count. The mean of point's absolute distance distributes in 50 pixels, 100 pixels and 150 pixels. The point count distributes in 2, 3 and 4. The standard deviation of point distance distributes in 10%, 20%, and 40% around the absolute distance mean of points in normal distribution. Due to the central limit theorem, this experiment uses the normal distribution as a model of quantitative phenomena in the natural and behavioral sciences. Many psychological measurements and physical phenomena (like noise) can be approximated well by the normal distribution. The use of the normal model can be theoretically justified by assuming that many small, independent effects are additively contributing to each observation. The curve extension distributes in 1 times, 1.5 times and 2 times of absolute distance mean of points. The experimental data is generated and sorted randomly. A set of the sample experimental data is listed below:

Index	Location				Mean	Deviation	Extension	Count
	1	2	3	4				
9	53.062	96.444	143.14	243.14	50	0.1	2	4
62	153.03	314.47	614.47	0	150	0.1	2	3
57	152.23	276.83	437.62	587.62	150	0.1	1	4
81	53.801	224.16	366.3	666.3	150	0.4	2	4
71	167.81	341.51	641.51	0	150	0.2	2	3
48	55.673	154.64	210.21	310.21	100	0.4	1	4
51	77.632	147.5	284.53	434.53	100	0.4	1.5	4
42	97.971	145.27	245.83	395.83	100	0.2	1.5	4
5	47.729	94.469	169.47	0	50	0.1	1.5	3
18	47.072	97.901	155.56	255.56	50	0.2	2	4
22	75.187	150.19	0	0	50	0.4	1.5	2
75	108.87	242.77	321.47	471.47	150	0.4	1	4
60	138.63	295.27	458.94	683.94	150	0.1	1.5	4
25	39.146	139.15	0	0	50	0.4	2	2
50	120.01	199.32	349.32	0	100	0.4	1.5	3
58	167.13	392.13	0	0	150	0.1	1.5	2
8	49.677	92.457	192.46	0	50	0.1	2	3
15	35.94	82.195	127.49	202.49	50	0.2	1.5	4
34	86.994	286.99	0	0	100	0.1	2	2
4	40.686	115.69	0	0	50	0.1	1.5	2
27	43.281	104.11	172.75	272.75	50	0.4	2	4
30	105.53	206.37	322.14	422.14	100	0.1	1	4
31	96.692	246.69	0	0	100	0.1	1.5	2
10	48.539	98.539	0	0	50	0.2	1	2
35	93.95	179.06	379.06	0	100	0.1	2	3
79	172.15	472.15	0	0	150	0.4	2	2
68	125.51	338.34	563.34	0	150	0.2	1.5	3
41	111.97	214.91	364.91	0	100	0.2	1.5	3
47	135.68	298.82	398.82	0	100	0.4	1	3
39	51.708	137.82	209.99	309.99	100	0.2	1	4
77	156.15	303.69	528.69	0	150	0.4	1.5	3

3	47.069	104.76	155.46	205.46	50	0.1	1	4
23	50.883	94.6	169.6	0	50	0.4	1.5	3
59	173.28	344.03	569.03	0	150	0.1	1.5	3
20	56.538	123.8	173.8	0	50	0.4	1	3
37	82.231	182.23	0	0	100	0.2	1	2
52	90.059	290.06	0	0	100	0.4	2	2
63	130.68	266.38	428.06	728.06	150	0.1	2	4
36	105.59	202.81	289.88	489.88	100	0.1	2	4
61	133.89	433.89	0	0	150	0.1	2	2
74	154.53	272.93	422.93	0	150	0.4	1	3
6	50.517	99.413	148.02	223.02	50	0.1	1.5	4
16	67.513	167.51	0	0	50	0.2	2	2
28	94.297	194.3	0	0	100	0.1	1	2
21	63.588	124.68	194.72	244.72	50	0.4	1	4
69	152.4	274.29	443.36	668.36	150	0.2	1.5	4
1	44.368	94.368	0	0	50	0.1	1	2
26	68.245	114.8	214.8	0	50	0.4	2	3
32	107.95	200.1	350.1	0	100	0.1	1.5	3
49	130.03	280.03	0	0	100	0.4	1.5	2
54	112.5	320.12	431.7	631.7	100	0.4	2	4
29	85.014	184.51	284.51	0	100	0.1	1	3
43	82.474	282.47	0	0	100	0.2	2	2
64	149.81	299.81	0	0	150	0.2	1	2
46	85.965	185.96	0	0	100	0.4	1	2
38	80.27	178.84	278.84	0	100	0.2	1	3
7	46.332	146.33	0	0	50	0.1	2	2
17	57.532	108.18	208.18	0	50	0.2	2	3
78	15.145	134.5	299.45	524.45	150	0.4	1.5	4
66	164.46	290.85	463.41	613.41	150	0.2	1	4
67	144.99	369.99	0	0	150	0.2	1.5	2
11	52.481	101.71	151.71	0	50	0.2	1	3
40	106.59	256.59	0	0	100	0.2	1.5	2
12	67.382	133.6	189.87	239.87	50	0.2	1	4
45	76.835	188.44	293.23	493.23	100	0.2	2	4
70	200.46	500.46	0	0	150	0.2	2	2
53	94.007	143.67	343.67	0	100	0.4	2	3
33	87.369	194.04	280.11	430.11	100	0.1	1.5	4
14	41.924	87.311	162.31	0	50	0.2	1.5	3
76	164.91	389.91	0	0	150	0.4	1.5	2
19	94.737	144.74	0	0	50	0.4	1	2
13	50.918	125.92	0	0	50	0.2	1.5	2

73	138.31	288.31	0	0	150	0.4	1	2
24	54.534	124.47	198.79	273.79	50	0.4	1.5	4
72	153.16	298.4	474.53	774.53	150	0.2	2	4
56	153.7	282.17	432.17	0	150	0.1	1	3
65	165.73	356.66	506.66	0	150	0.2	1	3
80	160.75	308.51	608.51	0	150	0.4	2	3
2	45.925	97.758	147.76	0	50	0.1	1	3
44	94.69	188.14	388.14	0	100	0.2	2	3
55	128.66	278.66	0	0	150	0.1	1	2

Table 6 Data of experiment 2

There are a total of 81 trials in this experiment. The experiment is repeated 9 times on IBM PC under the common working environment. The sample data is regenerated for each experiment. The decision about the connectivity is made based on the points drawn on the screen. The time spent on each decision is recorded to identify outliers. If an experiment takes too much time, it should be redone. Some of experimental results are listed below.

Index	Time	Answer
9	21	0
62	13	0
57	14	1
81	22	0
71	10	0
48	84	0
51	35	1
42	41	1
5	18	1
18	42	0
22	11	1

Table 7 Result of experiment 2

As the experiment is repeated 9 times, the sum of answer is in 0 to 9. The sample data and sum of results are analyzed by using ANOVAN in MATLAB (R2006A), which is given below:

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Mean of Points distance	1.556	2	0.778	0.97	0.386
Standard deviation	4.519	2	2.259	2.82	0.0695
Points extension rate	394.963	2	197.481	246.57	0
Points count	25.407	2	12.704	15.86	0
Mean of Points distance*Standard deviation	2.815	4	0.704	0.88	0.4838
Mean of Points distance*Points extension rate	3.259	4	0.815	1.02	0.4079
Mean of Points distance*Points count	6.37	4	1.593	1.99	0.1113
Standard deviation*Points extension rate	59.407	4	14.852	18.54	0
Standard deviation*Points count	3.852	4	0.963	1.2	0.3221
Points extension rate*Points count	7.407	4	1.852	2.31	0.0711
Error	38.444	48	0.801		
Total	548	80			

Constrained (Type III) sums of squares.

Table 8 Analysis of variance of experiment 2

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Mean of Points distance	1479.7	2	739.9	0.35	0.7042
Standard deviation	11437.8	2	5718.9	2.73	0.0753
Points extension rate	197862.5	2	98931.3	47.24	0
Points count	13022.3	2	6511.2	3.11	0.0537
Mean of Points distance*Standard deviation	14590	4	3647.5	1.74	0.1563
Mean of Points distance*Points extension rate	18574.1	4	4643.5	2.22	0.0811
Mean of Points distance*Points count	7037.2	4	1759.3	0.84	0.5066
Standard deviation*Points extension rate	44499.6	4	11124.9	5.31	0.0013
Standard deviation*Points count	15420.5	4	3855.1	1.84	0.1364
Points extension rate*Points count	17891.8	4	4472.9	2.14	0.0907
Error	100520.1	48	2094.2		
Total	442335.7	80			

Constrained (Type III) sums of squares.

Table 9 Analysis of variance of time spent in experiment 2

This experiment applies the same rule as that for analyzing the significant factor, that is, the factors with large F value comparing to the others are significant factors.

From Table 8, the mean of point distance is not main factor because F value equals to 0.35 and the interaction between the mean of point distance and point count has a small F value. The standard deviation and point's extension rate affect the result very much. The interaction between the standard deviation and the point count is a significant factor. The interaction between the mean of point distance and point's extension rate is a significant factor, that is, there must be a relationship between them. The interaction can be understood as a part of the standard deviation of whole point cloud. Actually, the point's extension rate changes the standard deviation of whole points set. Therefore, considering the whole point cloud, the standard deviation and the point count are kept to continue the experiments.

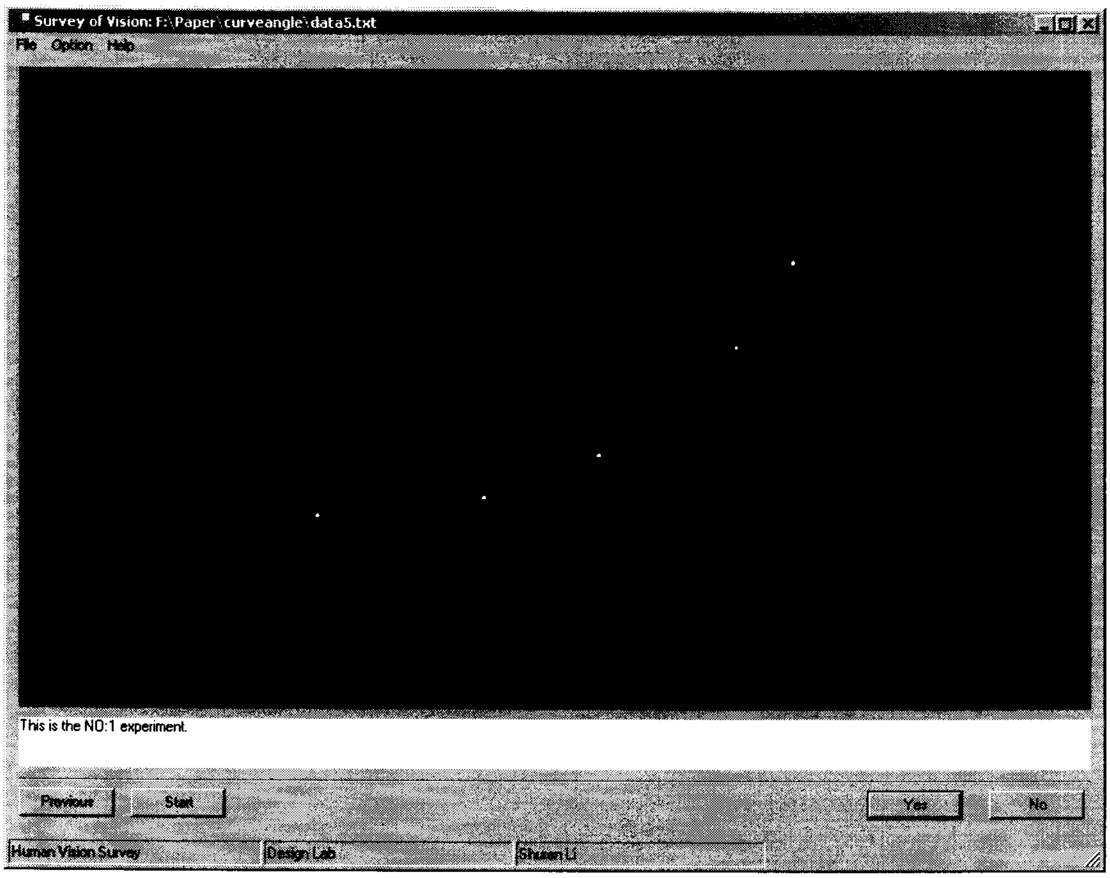
From Table 9, the point count does not affect time spent much but it affects the result significantly. In common sense, the major factor always affects time spent more, but in this experiment, it does not. It is an interesting phenomenon.

4.4 Experiment of curve with sharp corners

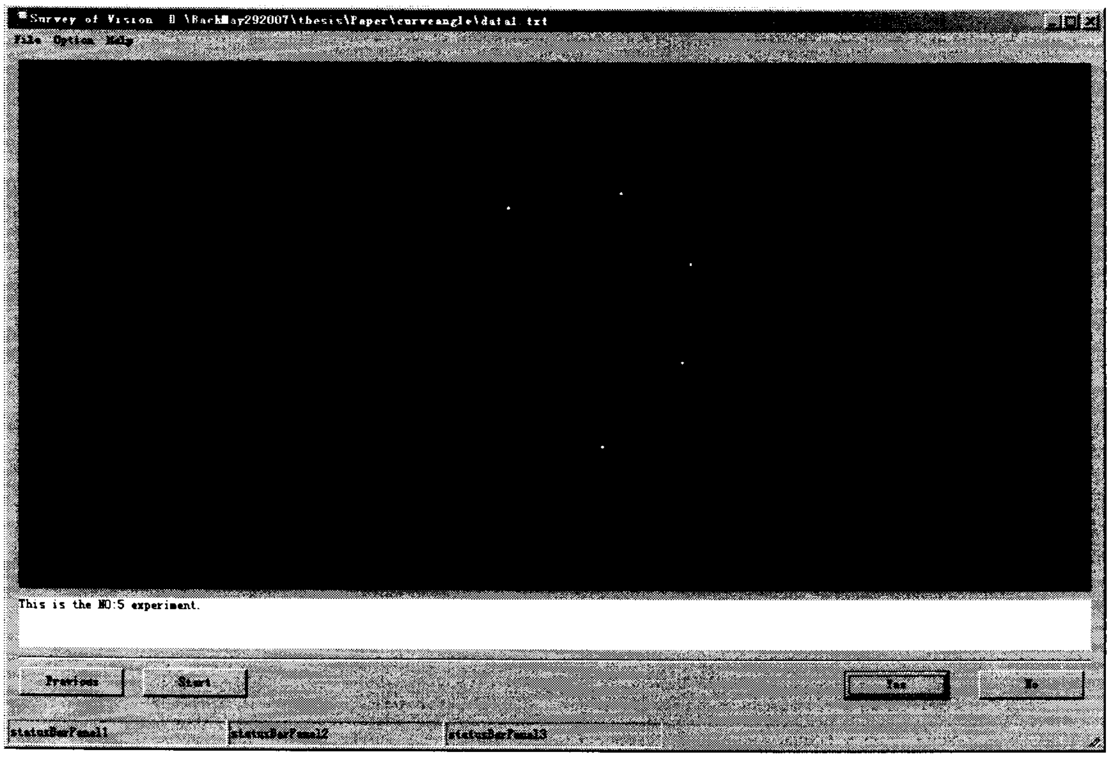
The objective of this step is to identify the major factors that affect the vision function for curves. The effect of the curve's angle will be studied. Based on experiment 2, this experiment considers the entire point cloud. Determining whether a point should or should not be connected to a curve uses the same logic of determining if the potential curve (extended point connected to the curve) is or is not a curve that can be reconstructed by vision function. In a global view, the extension rate changes the standard deviation of distance. Therefore, the extension rate is not considered anymore and the mean of points' angle, standard deviation of angle, standard deviation of distance and point count are the four factors in the experiment.

4.4.1 Experiment 3

In this experiment, the objective is to determine the significant factors in constructing the vision function for the reconstruction of curves with angles. The point cloud does not distribute in a straight line anymore. This experiment only investigates anti-clockwise turning curve because any curve can be decomposed into a set of anti-clockwise and clockwise turning curves and anti-clockwise curves can be easily transformed to clockwise turning curves . A software system is developed for this experiment, the interface of which is shown below:



(a)



(b)

Figure 17 Survey of curve with angle 1

The point cloud is shown on the interface of the software system. The order of points is shown below:

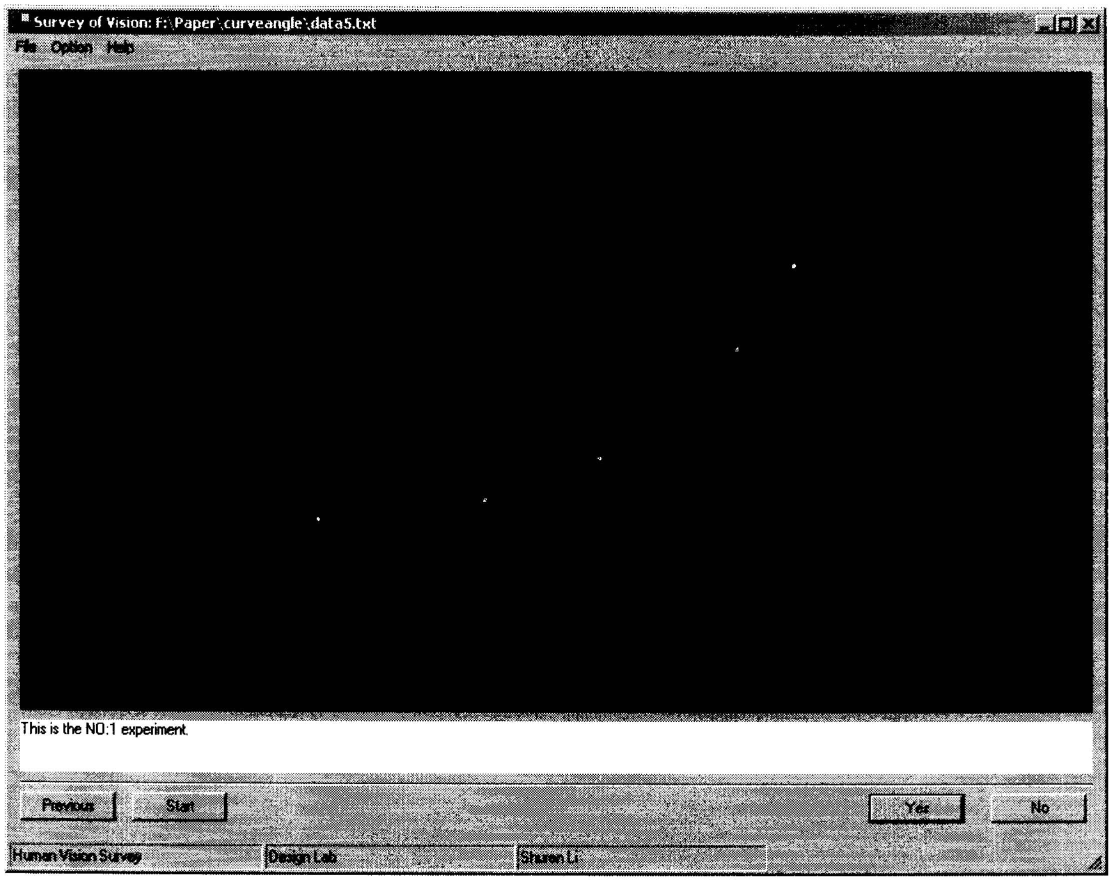


Figure 18 Survey of curve with angle 2

The point cloud is created with the following factors: the mean of curve angles, standard deviation of angles, standard deviation of distances and point count. The mean of curve angles distributes in 15 degree, 45 degree, 75 degree. The standard deviation of angle distributes in 6, 5, 4. The point count distributes in 2, 3 and 4. The absolute distance mean of points is 100 pix. The standard deviation of point distance distributes in 10%, 20%, and 40% around the absolute distance mean of points in normal distribution.

The experimental data is generated and sorted randomly. One set of the sample data is listed below:

Index	Location				Distance deviation	Count	Angle				Angle mean	Angle deviation
	1	2	3	4			1	2	3	4		
15	96.255	87.316	127.83	113.68	20	4	8.1145	15.237	34.893	50.194	15	5
79	61.644	54.714	0	0	30	2	74.859	149.79	0	0	75	4
81	128.17	65.512	178.69	171.27	30	4	75.226	147.87	218.94	288.43	75	4
66	103.61	96.154	103.55	108.35	10	4	79.384	157.12	232.45	312.73	75	5
14	79.249	82.209	117.56	0	20	3	18.251	33.787	48.701	0	15	5
33	106.22	89.404	91.353	104.36	20	4	46.053	85.08	134.57	187.56	45	6
73	92.031	108.56	0	0	10	2	71.681	150.39	0	0	75	4
41	89.163	98.07	46.404	0	20	3	38.188	80.634	126.45	0	45	5
78	98.403	114.75	130.7	125.35	20	4	82.981	158.88	228.61	304.34	75	4
8	130.88	143.34	103.78	0	30	3	20.9	45.007	66.014	0	15	6
6	116.41	110.02	95.107	117.4	20	4	18.863	32.816	33.621	58.436	15	6
42	73.741	70.816	131.59	87.786	20	4	35.204	87.961	133.29	180.59	45	5
40	98.685	115.23	0	0	20	2	44.7	84.683	0	0	45	5
45	59.964	80.328	116.9	115.95	30	4	47.398	89.491	131.79	181.7	45	5
21	110.81	101.72	120.84	103.65	10	4	21.113	41.023	67.618	72.322	15	4
29	101.07	99.095	115.2	0	10	3	43.41	88.048	133.3	0	45	6
43	102.68	98.819	0	0	30	2	47.482	88.158	0	0	45	5
60	85.766	112.58	132.19	114.4	20	4	73.841	138.41	205.79	274.2	75	6
65	105.08	85.202	103.1	0	10	3	73.598	140.19	209.51	0	75	5
77	118.58	100.42	89.213	0	20	3	76.615	153.05	221.37	0	75	4
62	126.78	109.67	55.223	0	30	3	82.887	147.72	229.09	0	75	6
30	94.336	87.895	80.358	88.51	10	4	46.2	93.615	135.33	179.55	45	6
31	153.15	116.62	0	0	20	2	35.709	90.806	0	0	45	6
76	69.134	77.654	0	0	20	2	78.265	155.29	0	0	75	4
34	120.14	94.356	0	0	30	2	47.046	83.852	0	0	45	6
48	99.819	95.886	103.32	84.26	10	4	42.32	89.416	134.9	176.63	45	4
70	74.397	27.941	0	0	30	2	78.853	146.24	0	0	75	5
20	94.233	94.557	86.314	0	10	3	18.186	36	51.958	0	15	4
13	117.7	94.965	0	0	20	2	19.648	34.708	0	0	15	5
27	60.032	122.84	120.47	91.122	30	4	12.902	26.22	38.616	54.46	15	4
47	96.842	101.95	109.15	0	10	3	42.735	88.74	130.75	0	45	4
7	128.07	121.31	0	0	30	2	12.115	24.383	0	0	15	6
67	75.025	71.43	0	0	20	2	80.011	156.5	0	0	75	5
4	118.88	65.349	0	0	20	2	14.787	27.98	0	0	15	6
26	148.28	102.87	99.536	0	30	3	9.6901	22.655	33.737	0	15	4
36	77.22	172.9	70.245	154.84	30	4	46.684	98.955	149.09	191.57	45	6
37	104.77	103.17	0	0	10	2	40.375	77.951	0	0	45	5
10	90.639	99.156	0	0	10	2	18.899	39.894	0	0	15	5
52	118.45	103.98	0	0	30	2	44.975	94.709	0	0	45	4
58	120.49	98.378	0	0	20	2	83.485	159.48	0	0	75	6
9	124.54	51.549	141.32	79.006	30	4	15.716	28.613	43.099	47.961	15	6
75	101.88	100.96	114.18	104.2	10	4	79.151	155.61	224.23	298.94	75	4
1	104.46	99.19	0	0	10	2	28.291	48.234	0	0	15	6
28	109.28	98.391	0	0	10	2	46.257	89.75	0	0	45	6
50	118.15	95.055	94.755	0	20	3	40.681	88.217	128	0	45	4
16	88.974	115.89	0	0	30	2	5.6004	19.52	0	0	15	5
55	104.75	104.09	0	0	10	2	83.709	151.17	0	0	75	6
74	93.071	101.88	87.724	0	10	3	68.726	134.01	204.35	0	75	4
3	91.483	104.18	106	88.264	10	4	14.523	28.963	26.17	37.719	15	6
64	94.866	108.49	0	0	10	2	76.5	147.49	0	0	75	5

46	111.26	117.18	0	0	10	2	47.683	88.079	0	0	45	4
19	87.221	94.996	0	0	10	2	16.533	33.712	0	0	15	4
22	79.501	131.65	0	0	20	2	21.803	35.823	0	0	15	4
32	143.25	77.412	100.04	0	20	3	42.692	75.978	117.74	0	45	6
35	156.75	82.984	98.615	0	30	3	37.147	89.895	138.38	0	45	6
72	50.191	121.65	139.28	62.323	30	4	76.529	150.58	230.87	312.26	75	5
2	93.731	93.258	105.77	0	10	3	15.346	34.788	54.047	0	15	6
49	104.59	104.28	0	0	20	2	45.885	92.029	0	0	45	4
53	114.84	116.06	85.24	0	30	3	41.952	85.392	130.22	0	45	4
38	103.78	91.972	105.86	0	10	3	42.648	86.453	133.37	0	45	5
59	91.4	24.023	86.017	0	20	3	72.856	132.13	208.12	0	75	6
17	118.72	37.629	103.87	0	30	3	14.262	22.549	31.161	0	15	5
57	98.135	88.837	99.681	120.6	10	4	80.435	159.12	244.1	328.67	75	6
63	90.525	89.33	86.149	148.21	30	4	82.078	154.01	227.7	293.43	75	6
80	97.014	117.07	116.48	0	30	3	75.531	145.61	227.19	0	75	4
18	153.92	68.07	120.38	121.77	30	4	22.723	33.408	44.255	49.411	15	5
24	82.262	68.44	118.91	122.04	20	4	13.576	21.006	37.968	55.776	15	4
12	94.478	111.69	80.517	95.984	10	4	15.32	20.386	40.364	56.494	15	5
5	89.146	67.717	104.4	0	20	3	18.724	40.129	52.992	0	15	6
56	96.102	102.93	116.91	0	10	3	77.408	158.88	243.24	0	75	6
44	131.79	83.905	65.448	0	30	3	45.803	89.964	140.77	0	45	5
25	91.193	82.395	0	0	30	2	10.093	37.171	0	0	15	4
11	91.506	101.61	110.57	0	10	3	14.51	32.978	42.956	0	15	5
23	32.875	92.58	127.82	0	20	3	14.172	35.854	55.575	0	15	4
51	73.685	132.82	99.399	100.48	20	4	41.988	87.244	133.98	184.5	45	4
54	6.7911	129.24	157.29	91.602	30	4	47.544	91.779	140.42	189.48	45	4
39	109.81	83.471	94.082	103.21	10	4	53.705	104.74	150.69	196.66	45	5
61	150.6	75.166	0	0	30	2	74.157	147.73	0	0	75	6
69	100.03	88.561	83.991	88.655	20	4	73.483	143.5	217.04	289.07	75	5
71	74.623	111.68	85.314	0	30	3	76.046	147	218.51	0	75	5
68	78.426	98.902	125.82	0	20	3	71.513	148.15	216.64	0	75	5

Table 10 Data of experiment 3

There are a total of 81 trials in the experiment. The experiment is repeated 15 times on IBM PC under the common working environment. The sample data is regenerated for each experiment. The decision about the connectivity is made based on the points drawn on screen. The time spent on each decision is recorded to identify outliers. If an experiment takes too much time, it should be redone. Some of experimental results are listed below.

Index	Time	Answer
15	21	0
79	13	0

81	14	1
66	22	0
14	10	0
33	84	0
73	35	1
41	41	1
78	18	1
8	42	0
6	11	1

Table 11 Result of experiment 3

As the experiment is repeated 15 times, the sum of answers is in 0 to 15. The sample data and sum result can be analyze by using ANOVAN in MATLAB (R2006A).

The matlab analysis result is listed below.

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Mean of Points Angle	1394.3	2	697.148	537.8	0
Standard deviation of angle	0.52	2	0.259	0.2	0.8194
Standard deviation of distance	153.56	2	76.778	59.23	0
Points count	305.41	2	152.704	117.8	0
Mean of Points Angle*Standard deviation of angle	0.74	4	0.185	0.14	0.9653
Mean of Points Angle*Standard deviation of distance	6.15	4	1.537	1.19	0.3291
Mean of Points Angle*Points count	527.41	4	131.852	101.71	0
Standard deviation of angle*Standard deviation of distance	1.7	4	0.426	0.33	0.8574
Standard deviation of angle*Points count	2.96	4	0.741	0.57	0.6846
Standard deviation of distance*Points count	18.59	4	4.648	3.59	0.0123
Error	62.22	48	1.296		
Total	2473.56	80			

Constrained (Type III) sums of squares.

Table 12 Analysis of variance of experiment 3

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Mean of Points Angle	2884814.8	2	1442407.4	2.43	0.0986
Standard deviation of angle	1302137.6	2	651068.8	1.1	0.3419
Standard deviation of distance	2041529	2	1020764.5	1.72	0.1898
Points count	419056.5	2	209528.2	0.35	0.7042
Mean of Points Angle*Standard deviation of angle	2792626.4	4	698156.6	1.18	0.3328
Mean of Points Angle*Standard deviation of distance	1568560.6	4	392140.1	0.66	0.622
Mean of Points Angle*Points count	833198.4	4	208299.6	0.35	0.8419
Standard deviation of angle*Standard deviation of distance	2748956	4	687239	1.16	0.3408
Standard deviation of angle*Points count	2628632.3	4	657158.1	1.11	0.3637
Standard deviation of distance*Points count	3320160.5	4	830040.1	1.4	0.2484
Error	28470892.7	48	593143.6		
Total	49010564.8	80			

Constrained (Type III) sums of squares.

Table 13 Analysis of variance of time spent 3

In this experiment the same rule is applied as that for analyzing the significant factor, that is, the factors with large F value comparing to the others are significant factors.

From Table 12, the means of curve angles, standard deviation of distances, point count are the role factors. The F value of the standard deviation of angle is small (0.2). It shows that the vision function is not sensitive to the small change of angle. From Table 13, the F value of point count is 0.35 but in Table 12 the F value of point count is 117.8. It is hard to explain why the point count does not affect the time of decision making but does affect the result of curve reconstruction.

4.5 Regression model

In this part, a vision function is derived from the data in experiment 3 by using the least square multiple variant non-linear regression method.

Nonlinear least squares regression extends linear least squares regression and can be used for a much larger and more general class of functions. Nonlinear regression model can almost simulate any function into a closed form. The greatest advantage of nonlinear least squares regression is that it can fit a more broad range of functions than many other techniques.

Standard multiple regression estimates the regression coefficients that can minimize the residual variance (sum of squared residuals) around the line of regression model. Least squares estimation is used in minimizing the sum of squared deviations of the values in experiment 3 from the predicted value simulated by the vision function.

The following function is used as vision function:

$$F = a_0 + a_1 * x + a_2 * y + a_3 * z + a_4 * w + a_5 * x * y + a_6 * x * z + a_7 * x * w + a_8 * y * z + a_9 * y * w + a_{10} * z * w + a_{11} * x * x + a_{12} * y * y + a_{13} * z * z + a_{14} * w * w$$

where F is the subject's decision from 0 to 15, x is the mean of angle, y is the standard deviation of angle, z is the standard deviation of distance, and w is the count of point. From a0 to a14 are the regression coefficients in the form of vector A. A value is found by using the least square multiple variant non-linear regression method.

A= [27.5648 0.5151 0.0509 -0.0611 -12.7037 -0.0009 0.0002 -0.0944 0.0028 0.1944 0.0167 -0.0043 -0.0556 -0.0044 2.2778]

Comparing the simulated data with the original data, the following chart is given.

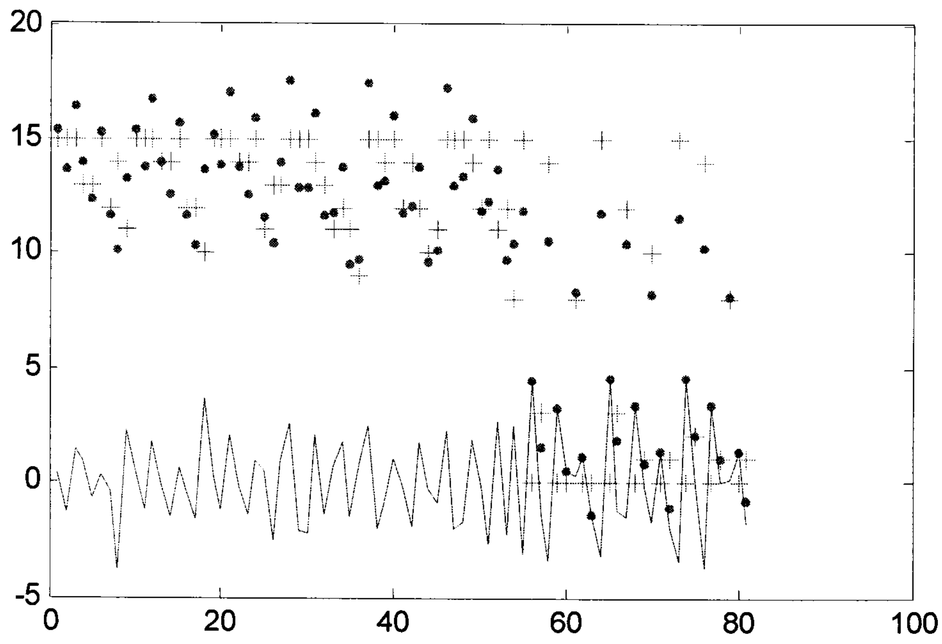


Figure 19 Comparison between the simulated data and original data

The cross point is the original data. The dot point is the simulated data. The curve is the error. In Figure 19 the residual is calculated and plotted.

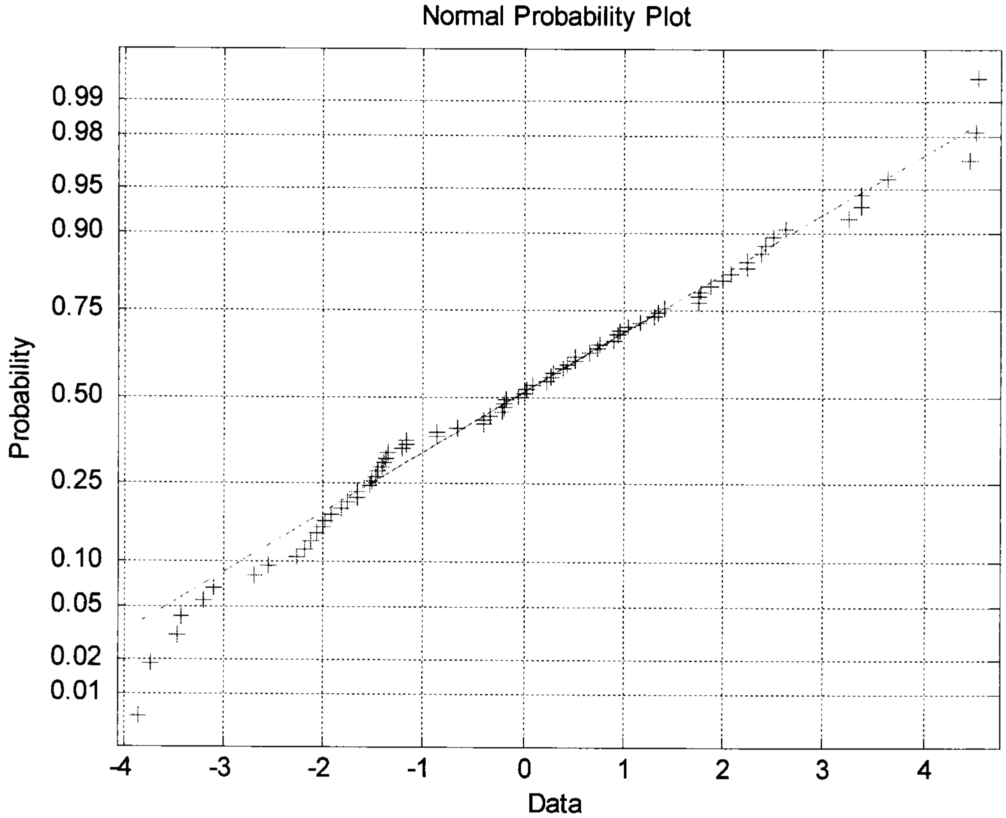


Figure 20 Residual of vision function

It shows that the error follows the normal distribution. It proves that the error is the background noise. The vision function is:

F=27.5648+0.5151*x+0.0509*y-0.0611*z-12.7037*w-0.0009*x*y+0.0002*x*z-0.0944*x*w+0.0028*y*z+0.1944*y*w+0.0167*z*w-0.0043*x*x-0.0556*y*y - 0.0044*z*z+ 2.2778*w*w;

where F is the connectivity decision from 0 to 15, x is the mean of angles, y is the standard deviation of angles, z is the standard deviation of distances, and w is the count of point.

Chapter 5
Vision-Based Curve Reconstruction: Algorithm and Comparison

5.1 Introduction

On the foundation of the distance-based parameter-free algorithm (DISCUR) in Chapter 3 and on the experimental results in Chapter 4, a vision-based curve reconstruction algorithm is proposed in this chapter. For a given set of point clouds, the proposed algorithm will connect the points into curves by using the vision function obtained from the previous chapter.

In Chapter 3, although Algorithm DISCUR is a simple, efficient, and parameter-free algorithm to reconstruct curves from unorganized sample points, algorithm DISCUR still can be improved in the following aspects.

Firstly, the algorithm DISCUR used only distance to quantify the two observations about the human visual system. The algorithm works correctly if sampling conditions in Theorems 1 and 2 (Chapter 3) are met. However, in some cases such as that given in Figure 21 a), if any two adjacent sampling points have the same distance, the curve cannot be reconstructed into a visually acceptable result.

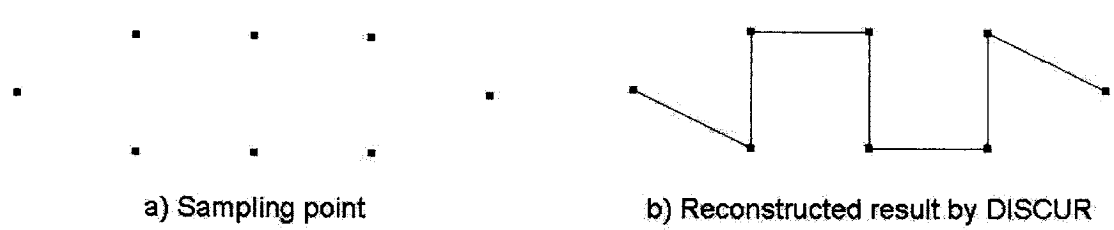


Figure 21 Example of wrong connections by DISCUR (Zeng *et al*, 2007)

In Figure 21 a), the wrong connections by using Algorithm DISCUR obviously violate the smoothness observation. The algorithm can be enhanced by adding a quantification of the smoothness observation based on angles between two edges to be connected. Vision function for curve reconstruction can be applied to deal with this problem.

Secondly, as mentioned in Chapter 3, algorithm DISCUR uses the vision function $\mathrm{E}[\mathrm{p},\mathrm{T_q}] = h_d \frac{h}{s}(1+\frac{h_d}{\sigma_d})^{\frac{h_d}{\sigma_d}}$ in determining the connectivity between two points. The formula $\mathrm{E}[\mathrm{p},\mathrm{T_q}] = h_d \frac{h}{s}(1+\frac{h_d}{\sigma_d})^{\frac{h_d}{\sigma_d}}$ is based on observations about human vision. Therefore, this formula could be replaced by vision function for curve reconstruction in Chapter 4, which is obtained experimentally.

A vision-based curve reconstruction algorithm is presented in this Chapter to improve algorithm DISCUR.

5.2 Vision-based curve reconstruction algorithm

To facilitate the description of the algorithm, a sample is called a free point if there is no edge connected to it; an end point or boundary if there is only one edge connected to it; an interior point if there are two edges connected to it (Zeng *et al*, 2007). The connectivity value of a curve and an edge is the result that is calculated by the vision function based on the curve and the edge.

vision-based curve reconstruction algorithm does not change the structure of algorithm DISCUR and only make some essential changes to make algorithm DISCUR compatible with the vision function in Chapter 4.

Two major modifications on algorithm DISCUR are given below:

First, vision-based curve reconstruction algorithm searches all the potential Delaunay edges and curves that connect to the end points of the nearest neighbor.

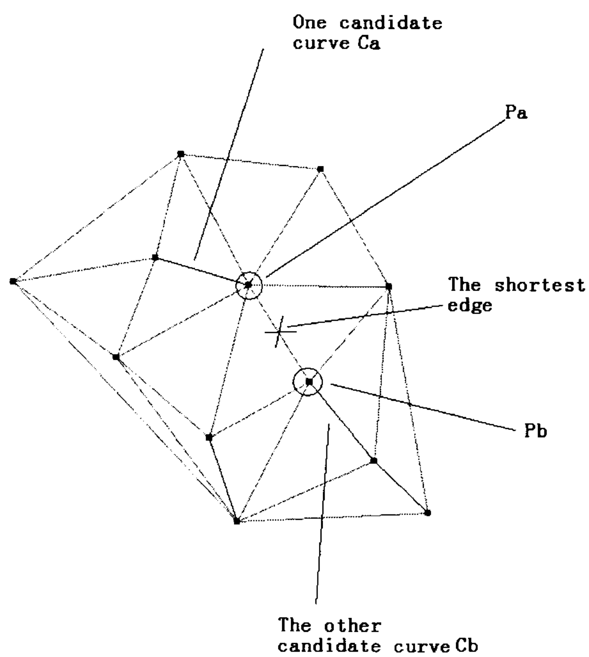


Figure 22 Candidate curves

Figure 22 shows how the algorithm searches for the nearest neighbor and connects them. In Figure 22, the algorithm searches the nearest neighbor because of observation 1) two closest neighbors tends to be connected. When the algorithm finds the two end points P_a and P_b of the nearest neighbor, the algorithm iterates all the Delaunay edges that connect to P_a and P_b, and find two curves C_a and C_b as candidate curves. For C_a, the algorithm find all the potential Delaunay edges that connected to P_a and are not a part of C_a, and then the algorithm calculates all the connectivity value of C_a and the potential Delaunay edges. The greatest one is marked as the connectivity value of C_a. The connectivity value of C_b is calculated similarly. Then the algorithm finds the greatest connectivity value of C_a and C_b. If the greatest connectivity value is big enough (greater than 7 or 8), the algorithm extends the curve with the correspond edge.

Secondly, while calculating the connectivity between a curve and an edge (or a curve and a point), the 2, 3 and 4 points on the curve that are closest to the point are calculated separately with the edge. All these 3 combinations have a connectivity value. Choose the greatest one as connectivity value of the curve.

The pseudocode of the algorithm is given as below:

```
Calculate the Delaunay triangulation.
WHILE Find the nearest neighbor.
   Find the two ends.
   IF both of the end points are not end point of a curve
      Connect the point pair.
   ELSE IF both of the end points are not interior points
     FOR each point in the end points
      IF the point is an end point of a curve
              FOR each edge that connect to the point
                      Calculate the connectivity value of the curve.
```

```
            ENDFOR
      END
      ENDFOR
      Find the edge with the largest connectivity value.
      IF the connectivity value is reasonable large
       Connect the edge.
      ELSE
       Disable the edge.
      ENDIF
    ELSE
      Disable the edge.
    ENDIF
  ENDWHILE
```

Figure 23 shows the connecting sequins of the vision-based curve reconstruction algorithm. Figure 23 (a) is the input points. Figure 23 (f) is the reconstructed curve.

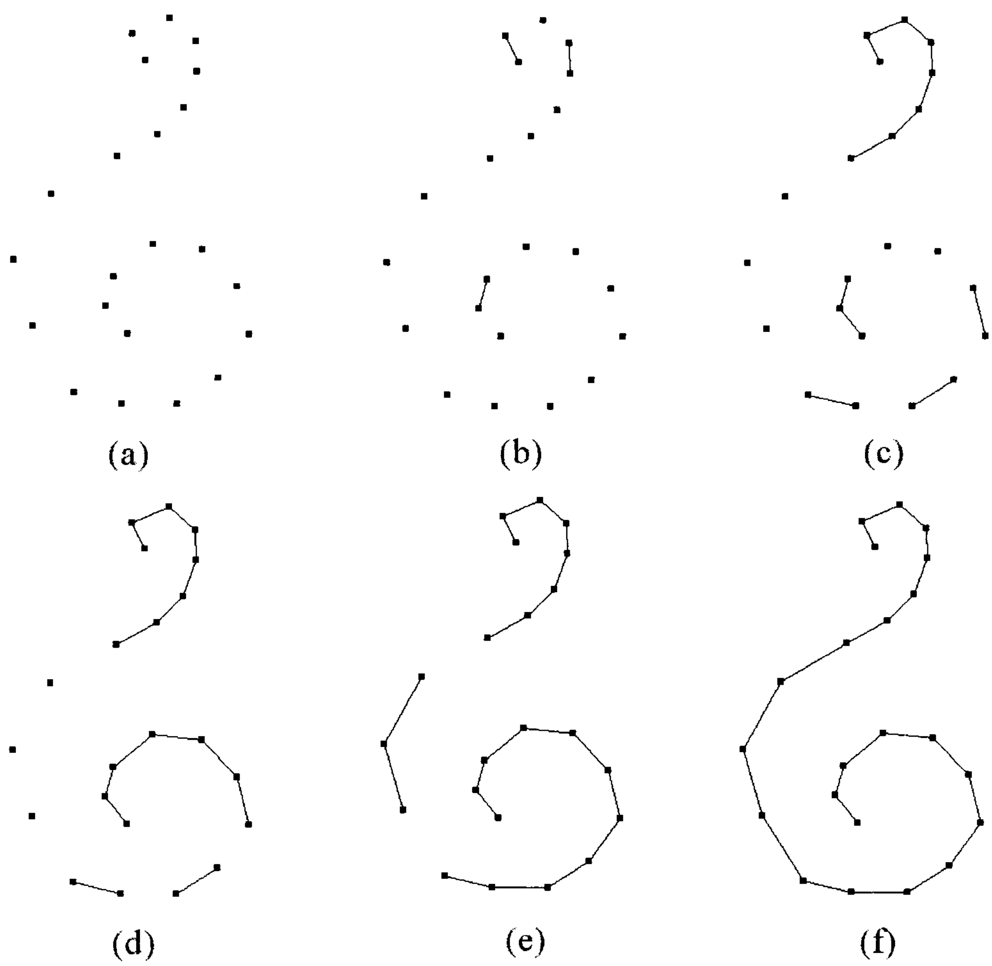


Figure 23 Connecting Sequins of the vision-based curve reconstruction

The major difference between the vision-based curve reconstruction algorithm and algorithm DISCUR is the treatment of the nearest neighbor.

The vision-based curve reconstruction algorithm first finds the end point of the nearest neighbor and all the edges that connect to the end points. Then the processing has three cases:

First, the end points in the nearest neighbor do not belong to any reconstructed curves.

This is the simplest situation. The points are connected immediately. Indeed, if the nearest neighbor is not a right construction, there must be some errors in the original points cloud and this violates the sampling condition. When a curve is sampled, there is no way the shortest distance of connected point pair is greater than the distance of non-connected point pair. This situation could be caused by noise point or some loss of point.

Second, one of the end points in the nearest neighbor belongs to a reconstructed curves.

As one of the end points is related to a reconstructed curve. Analysis should be made to investigate the relationship or effect of the curve. The algorithm finds all the edges that are connected to the end point. Thus, the algorithm tries to extend the curve to all the edges and find the most suitable edge. If the connectivity value of the edge is reasonable large, the algorithm connects the edge to the curve.

Third, both of the end points in the nearest neighbor belong to reconstructed curves.

This situation is about curve connection. There are a three of potential results. 1) The two curves could be connected. 2) The two curves are near but not connected. 3) The original curve intersects. The algorithm needs find all the edges that are connected to the two end points of the edge. Then the best two edges for each end point are found. Finally, if these two edges are the same one and the connectivity value is reasonably large, the two curves are connected. If they are not the same edge, then the algorithm find the edge has the greatest the connectivity value. If the connectivity value is reasonably large, the edges to the each curves are connected separately.

The computational time of the algorithm is O(n log n). First, the algorithm calculates Delaunay triangulation. The complexity is O(n log n). Secondly, the algorithm sorts the

Delaunay edges. The complexity is also O(n log n). In the third step, the algorithm iterates the related Delaunay edges and calculates the connectivity value, and the largest connectivity value. The complexity is O(n). So, the complexity of the algorithm is O(n log n).

5.3 Results

This part the result of the algorithm and comparison between the algorithm and other algorithms are given.

The input points cloud is shown in Figure 24. In Figure 25, Delaunay triangulations are constructed.

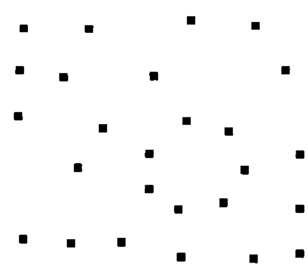


Figure 24 Original points

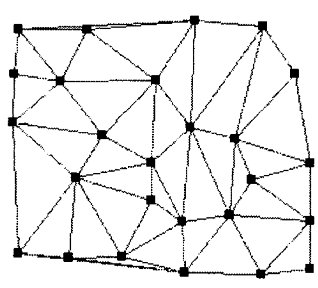


Figure 25 Delaunay triangulation

In Figure 26, the reconstruction result from our algorithm is shown.

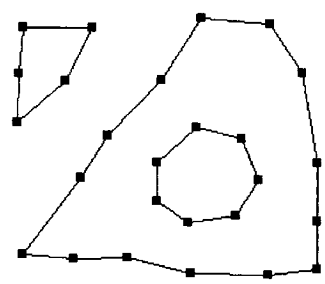


Figure 26 Reconstruction result

In Figure 27, the reconstruction result from ABE crust is shown.

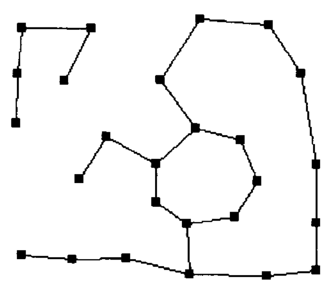


Figure 27 Reconstruction result from ABE crust

In Figure 28, the reconstruction result from Nearest Neighbor is shown.

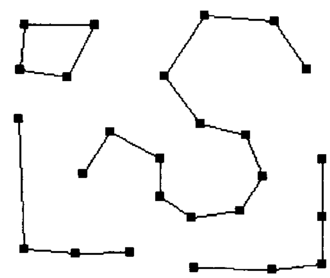


Figure 28 Reconstruction result from nearest neighbor

The comparisons with other algorithms show improvements made by our algorithm. By observing the differences, it is easy to find that vision-based algorithm has a more natural to human vision.

5.4 Comparison with existing algorithms

Most existing curve reconstruction algorithms, such as Crust (Amenta et al, 1998), NN Crust (Dey and Kumar, 1999), Conservative Crust (Dey et al, 1999), and Gathan (Dey et al, 2001), generate results homeomorphic to original curves provided their sampling conditions are satisfied. Most of those algorithms are O(n log n), which is comparable to the algorithm proposed in the present thesis. In this section, comparisons as regard to scope and accuracy will be made between vision-based curve reconstruction algorithm and existing algorithms, particularly Crust (Amenta et al, 1998), NN-CRUST (Dey and Kumar, 1999), and Gathan (Dey et al, 2001), which are viewed as effective ones in the literature.

Algorithm	*Sampling*	*Smoothness*	*Boundary*	*Components*
α-shape [3,4]	Uniform	Required	None	Multiple
γ-regular [5]	Uniform	Required	None	Multiple
EMST [2]	Uniform	Required	Exactly two	Single
Crust [6,7]	Non-uniform	Required	None	Multiple
NN [8]	Non-uniform	Required	None	Multiple
TSP [11,12]	Non-uniform	Not required	Must be known	Single
CC [9]	Non-uniform	Required	Any number	Multiple

Table 14 Scope of curve reconstruction algorithms (Dey, 2004)

Table 14 shows a comparison of most existing curve reconstruction algorithms as regard to their sampling condition, their ability to deal with sharp corners (smoothness of original curve), their capability to process open curves (curve with boundaries), and their ability to reconstruct multiple components. Examples will be given in the following to show the performances of these existing algorithms.

Sampling condition and parameters

Although many existing algorithms can successfully reconstruct curves from "dense enough" samples, they require the sampling conditions based on local feature size and pre defined parameters. The parameters depend heavily on the curves to be reconstructed. As a result, those algorithms may cause difficulties as the curves become complex. For example, multiple curve with different features. Different parameters should be adopted at different parts of the curve to achieve the desired result. More importantly, optimal parameter value should vary with curve features and thus make it more difficult to determine or balance. Figure 29 shows the results with different parameters have significant difference. The vision-based curve reconstruction algorithm can easily handle the difficulty as shown in Figure 30, which conforms to the human perception.

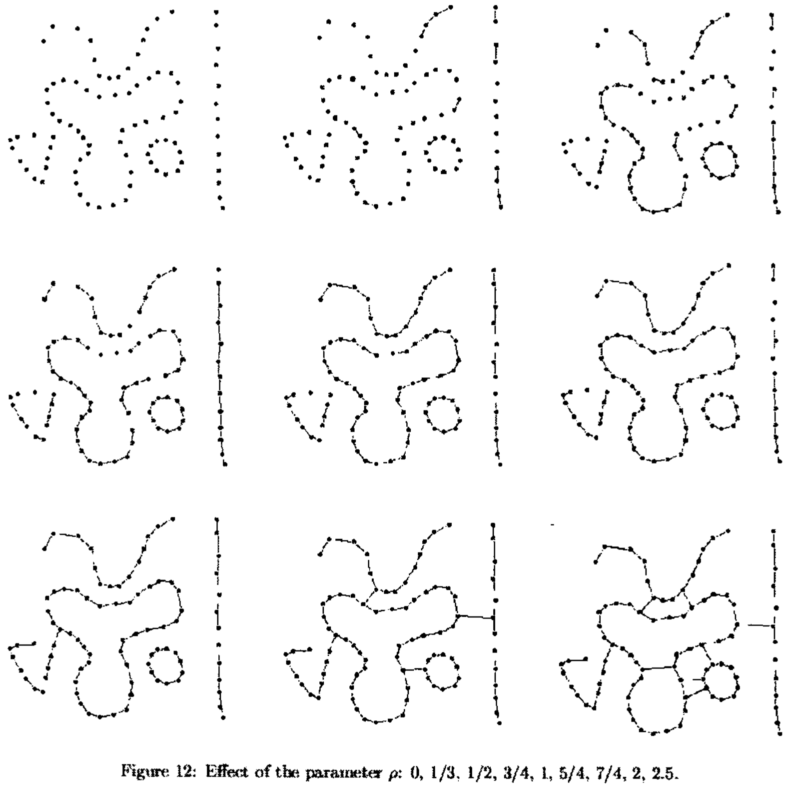


Figure 29 Sample of conservative-crust algorithm (Dey et al, 1999)

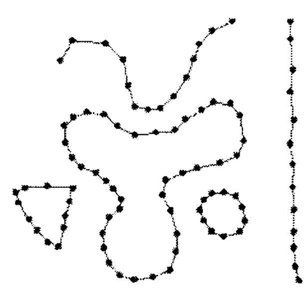
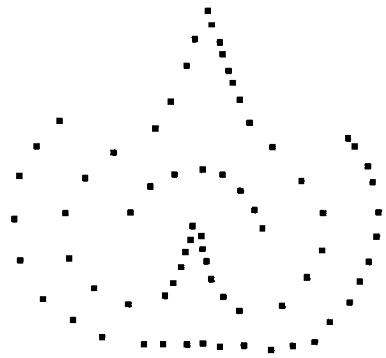


Figure 30 Result of vision-based curve reconstruction algorithm

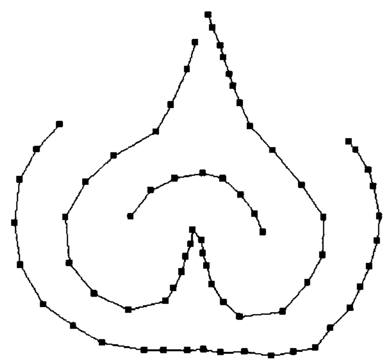
As a parameter-free algorithm, the vision-based curve reconstruction algorithm brings two major advantages. First, without parameter, reconstruction of multiple curves with multiple features is made an automatic process. There is no need of multiple parameters for 2 various parts with complex features. Secondly, the vision-based curve reconstruction algorithm can be used when original curve is not known. In this case, the sampling points will be connected as the human eye naturally perceive them. This makes it more useful for engineering applications than those dependent on known curve features.

Sharp corners

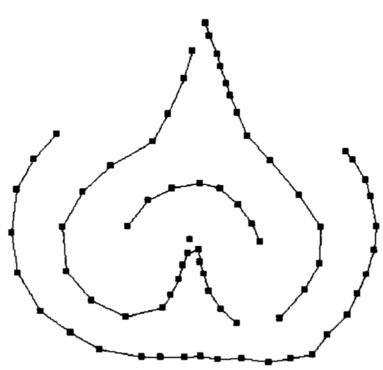
In the case involving sharp corners as shown in Figure 31, it is very difficult for CRUST and NN-CRUST to achieve a reconstruction close to the original curve. With correctly chosen parameters, GATHAN can successfully handle curves with sharp corners in some cases. Example in Figure 31 shows a complex sample set, which includes multiple features such as uneven samplings, sharp corners, boundaries, and multiple components. For the samples given in Figure 31, the vision-based curve reconstruction algorithm can obtain desired output if adding more points to the local area where the sampling connection is violated. As can be seen, CRUST and NN-CRUST still experience problems.



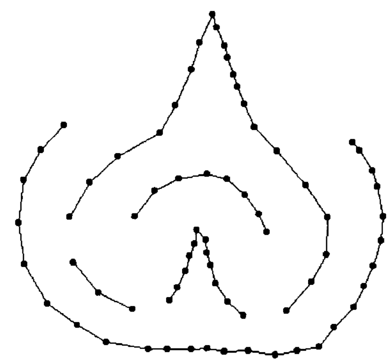
(a) Input point



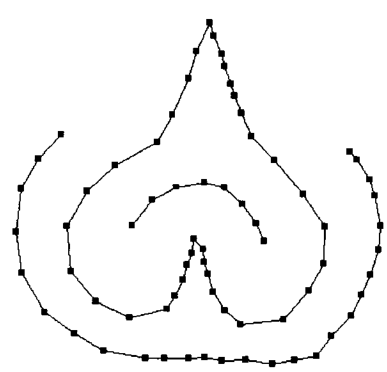
(b) Crust



(c) Nearest Neighbor



(d) GATHAN



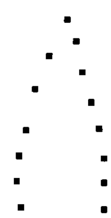
(e) Vision-based curve reconstruction

Figure 31 Comparison 1

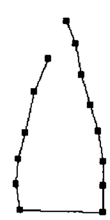
Figure 31 illustrates the effectiveness of CRUST, NN-CRUST, GATHAN. Obviously, vision-based curve reconstruction algorithm is the only algorithm that works for such a complex case.

Boundary and multiple components

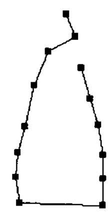
In comparison with CRUST and NN-CRUST, the vision-based curve reconstruction algorithm is able to detect the boundary points while NN-CRUST and CRUST wrongly connect them as shown in Figure 32. Given a sufficiently dense sample of a closed smooth curve, CRUST and NN-CURST successfully reconstruct the original curve but when the original is an open curve the algorithms do not guarantee the reconstruction in Figure 33 and Figure 34.



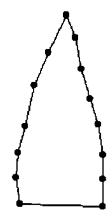
(a) Sampling points



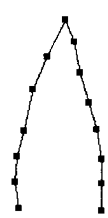
(b) CRUST



(c) NN-CRUST

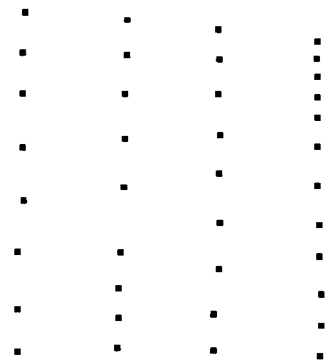


(d) GATHAN

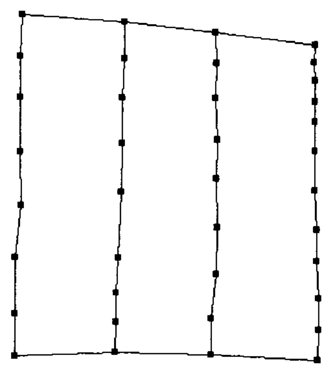


(e) Vision-based curve reconstruction algorithm

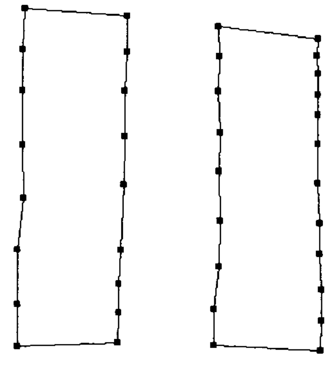
Figure 32 Reconstruction in the case of open curve 1



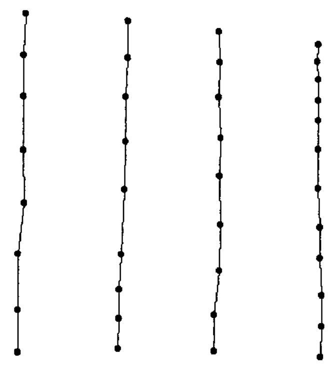
(a) Input points



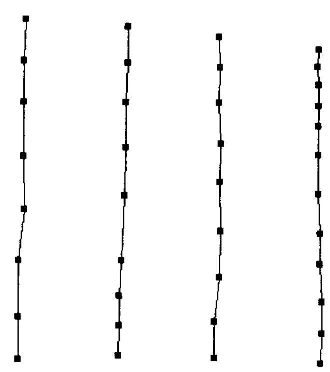
(b) Crust



(c) NN-CRUST

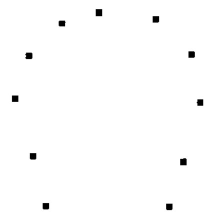


(d) GATHAN

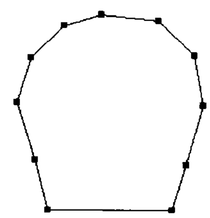


(e) Vision-based curve reconstruction algorithm

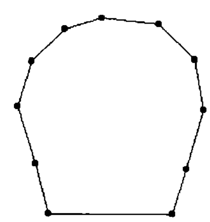
Figure 33 Reconstruction in the case of open curve 2



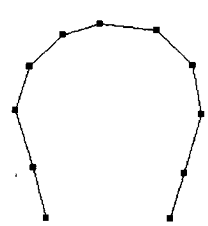
(a) Input points



(b) CRUST and NN-CRUST



(c) GATHAN



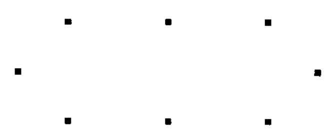
(d) Vision-based curve reconstruction algorithm

Figure 34 Reconstruction in the case of open curve 3

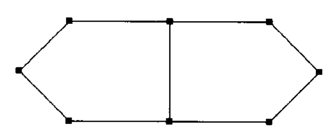
In some cases GATHAN algorithm can not detect the boundary endpoints correctly either. An example is given in Figure 34. It wrongly connects boundary points.

Two adjacent sampling points with the same distance

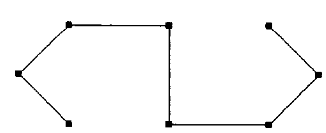
This problem originally comes from distance-based parameter-free algorithm (DISCUR) (Zeng et al, 2007) in Figure 35. Actually, many existed algorithms also have such kind of problem.



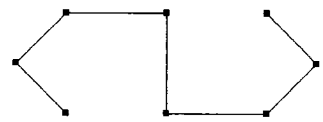
(a) Input points



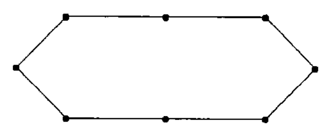
(b) CRUST



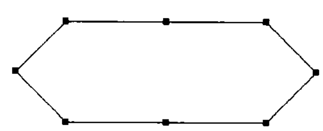
(c) NN-CRUST



(d) Distance-based parameter-free algorithm



(e) GATHAN



(f) Vision-based curve reconstruction algorithm

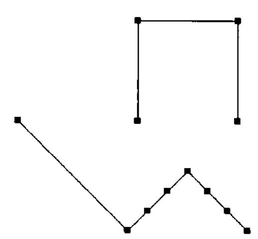
Figure 35 Two adjacent sampling points with the same distance

From Figure 35, NN-CRUST and distance-based parameter-free algorithm (DISCUR) cannot handle this problem because they are both distance oriented. GATHAN and the vision-based curve reconstruction algorithm can reconstructed the input points well because they both take angle in to account.

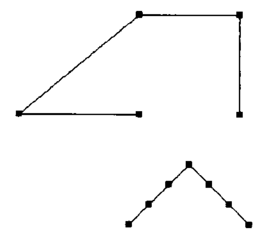
Noisy points

Existing algorithms, particularly Crust (Amenta et al, 1998), NN-CRUST (Dey and Kumar, 1999), and Gathan (Dey et al, 2001) cannot detect the noisy points in Figure 36.

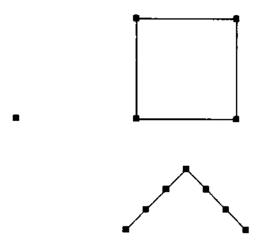
(a) Input points



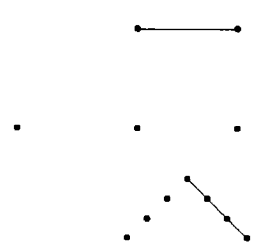
(b) CRUST



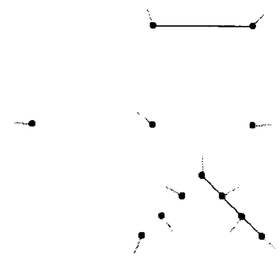
(c) NN-CRUST



(d) Vision-based curve reconstruction algorithm



(e) GATHAN



(f) Norm line of GATHAN

Figure 36 Reconstruction in the case of noisy point

From Figure 36, GATHAN is very sensitive to the noisy point because the noisy point changes the direction of normal lines of GATHAN in Figure 36 (f). The vision-based curve reconstruction algorithm can exclude the noisy point because this algorithm can perceive point cloud as naturally as the human eye.

5.5 Conclusion

Vision-based curve reconstruction algorithm has some advantage and disadvantage by comparison with other algorithms.

- This algorithm is good at dealing with points with outliers or noisy points.
- This algorithm can handle the sharp corners unless the connection between two nearest neighbor points is not in the edges of sharp corner.
- This algorithm cannot treat intersection problems.

Chapter 6
Conclusions and Future Work

The present thesis aims to design a new algorithm that reconstructs curves from points in the same way that human beings perceive them. To discover the effect of human vision on curve reconstruction, experiments were designed and conducted to identify how human vision recognizes curves from unorganized points. The present thesis has found a regression model by using methods from Design of Experiments (DOE), ANOVA and the multivariate non-linear regression model. To verify the vision function, the vision-based curve reconstruction algorithm has been designed.

The present thesis proposed a vision-based curve reconstruction algorithm based on a vision function constructed from experiments. The major contributions made in the present thesis can be summarized as follows:

1. Construction of a vision function based on experiments.

During the development and research of the curve reconstruction algorithm, I designed a platform for developing curve reconstruction algorithms. This platform implements almost all the major algorithms about the curve reconstruction. The platform provides a toolkit to help researchers implement and evaluate new algorithms.

To find vision function, all the potential factors in the context of curve reconstruction are firstly enumerated. Then experiments are designed and implemented to exclude the minor factors and to keep the major ones. Finally, the experiment of data of the major factors is used to construct the vision function and to evaluate the correctness of the function. Based on the experiments, a vision function is derived from the data by using the least square multiple variant non-linear regression method.

2. Development of a vision-based curve reconstruction algorithm.

The present thesis used the vision function, which is derived from experiments, to implement the vision based curve reconstruction. In doing that, the present thesis makes some essential changes to make algorithm DISCUR compatible with the vision function. The proposed vision-based curve reconstruction algorithm is more effective in dealing with curves with sharp corners.

In the future, this research should be focused on recognition of sharp corners by human vision, and curve reconstruction by human vision combining with human experiences and knowledge. The work will help solve sharp corner problems. Moreover, more research should be done with more experiments to study the relationship between two observations of human vision and balance the priority of two observations of human vision.

Publications

Publications in Refereed Journals

1. Y. Zeng, T. A. Nguyen, B. Yan, **S. Li** (2007), A distance-based parameter-free algorithm for curve reconstruction, *Computer Aided Design* (Under the second round of review)

Publications in Refereed Conferences

1. J. Xiao, **S. Li**, L. Yan, A. Ben Hamza, Y. Zeng (2006), Robust Statistical Quality Improvement for Soybean Machines, *Integrated Design and Process Technology, IDPT-2006*, SAN DIEGO , the United States of America, June 25-30, 2006.

References

Althaus, E. and Mehlhorn, K., 2000. TSP-based curve reconstruction in polynomial time, in Symposium on Discrete Algorithms, 686-695.

Althaus, E., Mehlhorn, K., Naher, S. and Schirra, S., 2000. Experiments on curve reconstruction, in Proc. 2nd Workshop Algorithm Eng. Exper., 103-114.

Amenta, N., Bern, M. and Eppstein, D., 1998. The crust and the beta-skeleton: combinatorial curve reconstruction, in Graphical Models and Image Processing,125-135.

Attali, D., 1997. r-regular shape reconstruction from unorganized points, in Proc. 13th Annual ACM Symposium on Computational Geometry, 248-253.

Aurenhammer, F., 1991. Voronoi diagrams - A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys, 23(3):345-405.

Baars, B. J., Ramsey, T. Z., and Laureys, S., 2003. Brain, conscious experience and the observing self. TRENDS in Neurosciences Vol.26 No.12 December 2003

Bernardini, F. and Bajaj, C., 1997. Sampling and reconstructing manifolds using a-shapes, in Proc. 9th Canadian Conference on Computational Geometry, 193-198.

Bowyer, A., 1981. Computing Dirichlet tessellations, The Computer Journal 24(2):162-166.

Chellappa, R., Wilson, C. L., Sirohey, S., 1995. Human and machine recognition of faces: a survey. Proceedings of the IEEE, 83(5), 705-741.

Delaunay, B. , 1934. Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikhi Estestvennykh Nauk, 7:793-800.

Dey, T. K. and Wenger, R., 19 (2001). Reconstructing curves with sharp corners, Computational Geometry Theory & Applications, 89-99.

Dey, T. K., Kumar,P., 1999. A simple provable algorithm for curve reconstruction, in Proc.10th Annual ACM-SIAM Symposium on Discrete Algorithms, 893-894.

Dey, T. K., Mehlhorn, K. and Ramos, E., 1999. Curve reconstruction: connecting dots with good reason, in Proc. 15th Annual ACM Symposium on Computational Geometry, 197- 206.

Dey, T. K., Wenger, R., 2001. Reconstructing curves with sharp corners, Computational Geometry Theory & Applications, 19, 89-99.

Dey, T. K., 2004. Handbook of Discrete and Computational Geometry, Goodman and O' Rourke eds., CRC press, 2nd edition,30:1.

Dirichlet, G. L., 1850. Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. Journal für die Reine und Angewandte Mathematik, 40:209-227.

Edelsbrunner, H., Kirkpatrick, D. and Seidel, R., 1983. On the shape of a set of points in the plane, IEEE Transactions on Information Theory, 71-78.

Edelsbrunner, H., 1998. Shape reconstruction with Delaunay complex, LNCS 1380,LATIN_98: Theoretical Informatics, 119-132.

Figueiredo, L. and Gomes, J., 1994. Computational morphology of curves, Visual Computer, 11. 105-112.

Funke, S. 2001. Combinatorial Curve Reconstruction and the Efficient Exact Implementation of Geometric Algorithms, pp.7.

Gepshtein, S., Banks, M., 2003. Viewing Geometry Determines How Vision and Haptics Combine in Size Perception. Current Biology, Vol. 13, 483–488, March 18, 2003

Giesen, J., 1999. Curve reconstruction, the TSP, and Menger_s theorem on length, in Proc. 15th Annual ACM Symposium on Computational Geometry, 207- 216.

Gold, C., 1999. Crust and anti-crust: a one-step boundary and skeleton extraction algorithm, in Proc. 15th Annual ACM Symposium on Computational Geometry,189-196.

Levin, D.N., 2000. A Differential Geometric Description of the Relationships among Perceptions. Journal of Mathematical Psychology archive Volume 44 , Issue 2, June.

Libet, B., 1966. Brain stimulation and the threshold of conscious experience. In J. C. Eccles (Ed.), Brain and conscious experience. New York: Springer Verlag.

Lubin, J., 1997. A human vision system model for objective picture qualitymeasurements. Broadcasting Convention, 498-503.

Mehlhorn, T.K., K., Ramos, E.,1999. Curve reconstruction: connecting dots with good reason, in Proc. 15th Annual ACM Symposium on Computational Geometry, 197- 206.

Morrone, M. C., Burr., D. C., 1988. Proceedings of the Royal Society of London. Series B, Biological Sciences, Vol. 235, No. 1280 (Dec. 22, 1988), pp.221-245.

Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N., 2002. Spatial Tessellations - Concepts and Applications of Voronoi diagrams. 2nd edition. John Wiley.

Pernot J-P., Guillet S., Léon J. C., Giannini F., Catalano C. E., Falcidieno B., 2002. Design and Styling Features using Surface Deformation, 4th International Conference on Integrated Design and Manufacturing in Mechanical Engineering (IDMME2002), Institut Français de Mécanique Avancée (IFMA), 14-16 may 2002.

Voronoi, G., 1907. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Journal für die Reine und Angewandte Mathematik, 133:97-178.

Watson, D. F., 1981. Computing the n-dimensional tessellation with application to Voronoi polytopes, The Computer Journal, Heyden & Sons Ltd., 1981,Vol 2, Num 24, pp.167-172.

Wikipedia, http://en.wikipedia.org/wiki/Voronoi_diagram.

Zhu, D., Zeng, Y. and Ronsky, J., February, 2004. Geometric feature-based algorithms for curvere construction, Technical Report, Concordia Institute for Information Systems Engineering, Concordia University.

Zeng, Y., Nguyen, T. A., Yan, B., Li, S., 2007. A distance-based parameter-free algorithm for curve reconstruction, Computer Aided Design (Submitted).