VISION BASED CURVE RECONSTRUCTION ALGORITHMS AND THEIR APPLICATION TO GRAPHICAL PASSWORD

THANH AN NGUYEN

A THESIS
IN
THE DEPARTMENT
OF
CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE (INFORMATION SYSTEMS SECURITY)
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

APRIL 2009

Your file Votre référence
ISBN: 978-0-494-63296-3
Our file Notre référence
ISBN: 978-0-494-63296-3

Abstract

Vision based curve reconstruction algorithms and their application to graphical password

Thanh An Nguyen

Curve reconstruction is the problem of approximating a curve or multiple curves from a point cloud. Curve reconstruction problem has received numerous attention over the last few decades due to its significant application in geometric modeling. In this thesis, based on the relationship between human vision and curve reconstruction, two Gestalt laws have been identified for the curve reconstruction: the law of proximity indicating that our vision tends to perceptually group near objects together and the law of continuation pointing out that objects following a consistent continuous direction are perceptually grouped together. Two algorithms have been proposed to implement these two laws in curve reconstruction. This first algorithm, DISCUR, connects points based on the law of proximity. The second algorithm, VICUR, considers both laws. The algorithms have been compared to the main curve reconstruction algorithms available in the literature.

Another contribution of this thesis is a new application of curve reconstruction in the field of cryptography. In the thesis, a new graphical password scheme is introduced. The proposed scheme requires users to create their secret by selecting individual points or by connecting points into curves from a given set of points. It is reasonable to assume that the users will connect points into curves that look natural to their vision so that they can recall easily. Consequently, the password may be a part of the reconstructed results of the human-vision based curve reconstruction algorithms and the attacker can use these results to crack the password. We present the application of curve reconstruction algorithm in the evaluation of our graphical password scheme.

Acknowledgments

In our daily lives, we must see that it is not happiness that makes us grateful, but the gratefulness that makes us happy.
Albert Clarke

I would like to take this opportunity to express my gratefulness to those who have been playing important roles in my life and those who have been helping me to complete my Masters study. First of all, I thank my supervisor, Dr.Yong Zeng. I would not be able to finish this thesis without his tremendous professional help and emotional support. During my last two years in the program, I have learned a great deal from him. Not only does he tirelessly provide many opportunities for me to learn and to grow but also does he encourage me to act, to dream big, to dare to make mistake and most importantly to have strong faith. I wholeheartedly thank him for his patience and kindness that he has shown to me. He is not only my professor but also my spiritual friend.

I thank all the people in the Design Lab: Guangqing He for all the discussions during my work on curve reconstruction project, Baiquan and Shuren Li for their helps in the beginning of the project, Min Wang, Yao Tang, Liu Wei, Da Yong, and all others for their company. I thank all the CIISE professors and staffs who have given me as well as other students a friendly academic environment.

I appreciate all my friends for always being by my side. I sincerely thank Quan for his selfless help whenever I need him.

I send all my love to my family (my grandmother, father, mother and brother) and relatives (my aunts and uncles). I believe whatever I have achieved today is always a partly result of my father's tremendous self-sacrifice, my mother's immense devotion and my grandmother's grand love.

I deeply appreciate all the events that have come to me, all the success and failure, all the moments of glory and the moments of humiliation, all the good and the bad, the truth and the delusion. I feel blessed for they came to me at the right time to show me the power of serenity. Lastly, I thank those who have been watching over me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

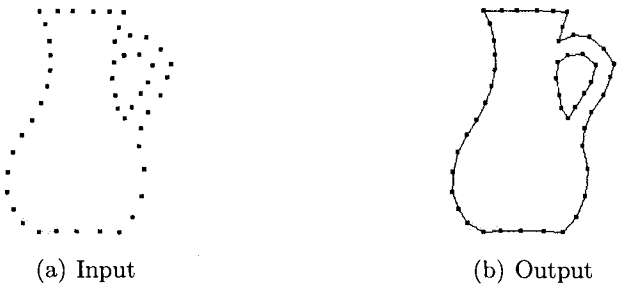
1.1 Objectives

This thesis has two main objectives. The first is to develop human-vision based curve reconstruction algorithms whereas the second is to design a new graphical password scheme based on the developed curve reconstruction algorithms. Though the first objective belongs to the field of computational geometry and the second objective belongs to computer security, these two objectives are closely related.

1.1.1 Objective 1: Human-vision based curve reconstruction

The main problem of curve reconstruction is to construct piecewise linear curves from a set of unorganized discrete points, as shown in Figure 1. Application of curve reconstruction can be found in three dimensional (3D) object modeling. Particularly, in medical field, constructing 3D objects from a set of planar contours is relatively popular. In the study of the structure of microscopic specimens, due to the monocular view of the microscope, the view of the specimens is limited to two dimensions. Therefore, the 3D structure of the specimens has to be constructed from a series of 2D images [FKU77]. Likewise, medical scanning devices such as Magnetic Resonance Imaging (MRI) scanner or Computed Tomography (CT) scanner can only generate 2D images of internal body parts. For diagnosis purpose, sometimes it is required of the physicians to view the 3D structure of the objects. In general, the process of modeling 3D object from contour lines requires three main steps.

In the first step, the contour lines of desired features are identified and extracted from 2D images as a set of discrete points as shown in Figure 2. In the second step, the curve reconstruction algorithm constructs the polygonal curves for each corresponding point set, as illustrated in Figure 3. A point set may include multiple curves with various features such as sharp corners or boundaries. Eventually, pairs of constructed curves in neighboring section are stacked together to generate the mesh over the contour lines as shown in Figure 4. The final structure of the reconstructed 3D object will be defined by creating the surface over these wire-frame contours.


(a) Input

(b) Output

Figure 1: A curve reconstruction problem.

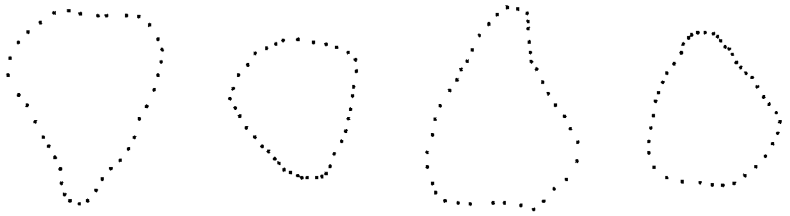


Figure 2: A set of contour points extracted from 2D images.

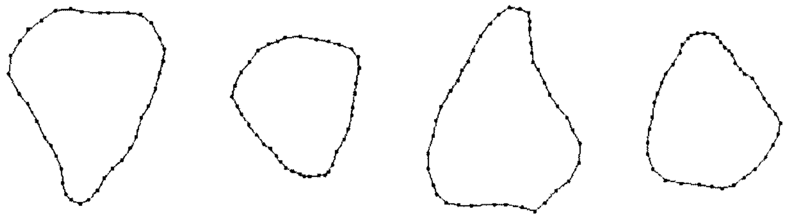


Figure 3: The contour points are connected to form curves.

A critical factor in the curve reconstruction problem is to define the criteria for connecting the unorganized points. One of such criteria is that the curves must be reconstructed in

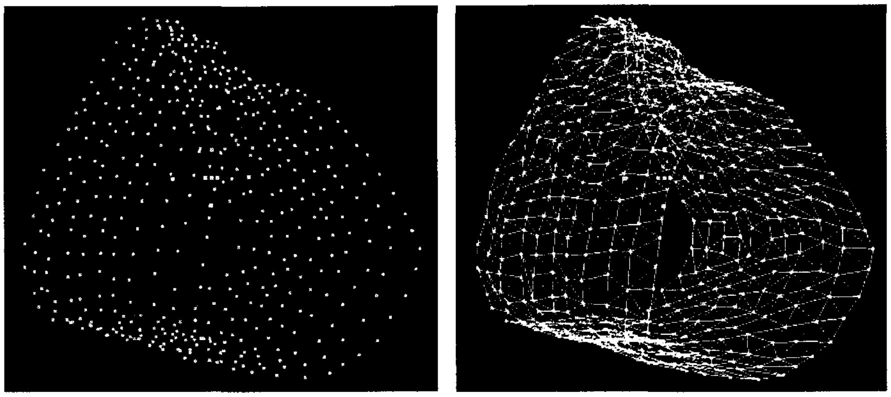


Figure 4: Constructed curves and surface in 3D.

the way that is natural to human perception. This thesis aims to develop a human-vision based curve reconstruction algorithm which connect points based on such criterion.

1.1.2 Objective 2: Graphical password

Graphical password is an authentication means alternative to textual password, biometric password, smart card, etc. In using graphical password, users can select or draw pictures to authenticate themselves to the system. A secure password must satisfy two basic requirements: 1) it must be memorable by its owners, 2) it must be difficult to be guessed by attackers. Textual password has been a popular authentication mean. However, with the increasing growth in computer hardware and software, many current textual password may not be able to satisfy the two requirements. It is well known that a strong textual password must be random and long enough. However this means the password will be hard to memorize by its owner. To solve this problem, researchers have proposed graphical password.

Graphical passwords are classified into two groups: picture based password and user-drawn based password [Tao]. In the picture based graphical password, users are asked to choose pictures as their secret. In general, the pictures are provided by the system but in some password schemes users can even upload their own pictures. The authentication is successful when users can prove that they know the secret (e.g. by correctly choosing the pre-selected images). In the user-drawn based graphical password, users are required to draw an image on a provided canvas. The authentication is successful when users can reproduce the same pattern on the canvas.

In order to make a graphical password memorable, the password must be intuitive to human perception. Therefore, this thesis aims to design a new graphical password scheme that helps users to create passwords intuitive to their vision.

A fundamental hypothesis for this research is that human-vision based curve reconstruction algorithm is an effective method to crack a graphical password based on unorganized points.

To facilitate subsequent discussions, we introduce some notations and definitions in the next section.

1.2 Notations and definitions

For a finite set of points $S = \{p_1, p_2, ..., p_m\}$ in $\mathbf{R}^n$, the Euclidean distance between two points p_i and p_j is denoted by $d(p_i, p_j) = \|p_i - p_j\|$. $|S|$ is the total number of points in the finite set P.

A polyline T is a continuous and piecewise linear curve and is denoted by $T = [q_1, q_2, ..., q_m]$, where $q_1, q_2, ..., q_m$ are vertices on the polyline and $q_i \neq q_{i+1}$ for all $i = 1, ..., m-1$. If $q_1 \neq q_m$, then T is an open curve; otherwise, T is a closed curve. For any closed curve T, $[q_1, q_2, ...q_m = q_1], [q_2, q_3, ..., q_{m-1}, q_1, q_2]$ and so on are considered the same. For any open polyline $T = [q_1, q_2, ..., q_m]$, a point p can be added to T by $[p|T_{q_1}]$ or $[T_{q_m}|p]$, respectively.

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; and interior point if there are two edges connected to it.

The distance mean of the polyline $T = [q_1, q_2, ..., q_m]$ and the standard deviation of distance are denoted as h_d and σ_d respectively where

$$h_d = \frac{\sum_{i=1}^{m-1} \|q_i - q_{i+1}\|}{m-1}, \tag{1}$$

$$\sigma_d = \sqrt{\frac{\sum_{i=1}^{m-1}(\|q_i - q_{i+1}\| - h_d)^2}{m-2}}. \qquad (2)$$

An open polyline $T = [q_1, q_2, ..., q_m]$ is called α-smooth if:

$$\pi - \alpha \leq \angle(q_{i-1}q_iq_{i+1}) \leq \pi + \alpha, \; i = 2, ..., m-1, \qquad (3)$$

where $\angle(q_{i-1}q_iq_{i+1})$ is the angle at vertex q_i in the halfspace defined as a clockwise rotation from the edge joining p and its nearest curve endpoint to the curve segment incident to the curve endpoint. Figure 5 shows the angles at vertices of a polyline T. Value α determines the smoothness of the curve. The smaller the value α is, the smoother the curve is.

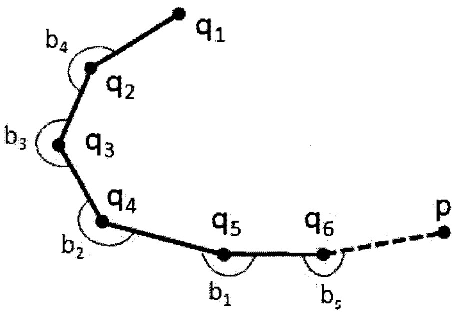


Figure 5: Angles at curve vertices

1.3 Literature review

1.3.1 Curve reconstruction

Because of its importance in various application domains, curve reconstruction has been attracting numerous research attention over the last three decades. In this thesis, we consider only simple curves that do not have intersections. Thus, the term curve(s) mentioned in the thesis is implicitly referred to simple curve(s). There are mainly two kinds of simple curves: open and closed curves. A curve without sharp corner is called smooth curve. In curve reconstruction problem, to construct the desired curves, the points in a point set must satisfy certain condition, called sampling condition. The process of generate points from a curve is called sampling. A curve can be sampled either uniformly or non-uniformly.

There are a few algorithms working for uniformly sampled curves such as alpha-shape [EKS83], r-regular shape [Att98] and EMST [dFdMG94]. There are other algorithms that work for non-uniformly sampled curves such as CRUST [ABE98, Gol99], Nearest Neighbor (NN or NN-CRUST) [DK99], Conservative Crust (CC) [DMR99], Traveling Salesman Path (TSP) [AM00, Gie99] and GATHAN [DW01, DW02]. Non-uniform sampling allows sparser sampling at less detailed section of the curve while uniform sampling unnecessarily requires dense sampling in areas where sparse sampling should be enough. Most of the existing algorithms for non-uniform sample requires the sampling points to be an ϵ-sample. A point set S is called an ϵ-sample of a curve T if any point p on T has a sample within distance $\gamma f(p)$ where γ is a constant factor and $f(p)$ is local feature size at p defined as minimum Euclidean distance from p to medial axis [Dey07].

The first provable curve reconstruction algorithm for simple close smooth curves is given by Amenta et al. named CRUST [ABE98]. The paper proves that for $\gamma \leq 0.252$ the polygonal reconstruction of a curve is the crust. The crust is constructed by computing Delaunay triangulation on the set of sampling points and Voronoi vertices, and choosing only Delaunay edges which have the endpoints belonging to the set of sampling points. Later, Dey-Kumar presented Nearest Neighbor algorithm [DK99] which is based on CRUST. The algorithm constructs simple close smooth curves by connecting each point to its nearest neighbor; then, for each point p that is incident to only one edge e, the algorithm connects p to its nearest neighbor in the other halfspace orthogonal to e. In fact, NN-CRUST is based on CRUST but with better sampling density. To deal with open curves, Dey et al. proposed Conservative Crust [DMR99]. The algorithm constructs the curve by computing Delaunay triangulation on the sampling points and choosing only Delaunay edge e which has an empty ball of Voronoi vertices centering at the midpoint of e with radius $\frac{l(e)}{k}$ where k is a parameter for the algorithm and $l(e)$ is the length of edge e. Then it filters all chosen edges e that have a large enough ball centering at the midpoint of e containing a zero degree or one degree vertices. Conservative Crust presents γ as a constant c multiplied by parameter k. Later, Funke and Ramos used Conservative Crust with different sampling condition near corner points to guarantee construction of open non-smooth curves [FR01]. In 1999, Dey and Wenger proposed an algorithm, named GATHAN, which can construct curves with sharp corners [DW01]. The algorithm has different sampling conditions for non-smooth regions and smooth regions of the curve. The sampling density near non-smooth regions is dependent on the angle of the sharp corner. Nevertheless, the authors does not guarantee the reconstruction result of their algorithm. In 2002, a guarantee version of GATHAN is introduced called GATHANG [DW02]. Another algorithm can reconstruct non-smooth curve is based on traveling salesman problem (TSP). TSP-based algorithm defines a modified cost function to set the sampling condition for every two adjacent sample points on a curve [AM00]. This condition results in a sparser sampling density compared to CRUST, NN-CRUST and CC [AMNS00]. According to [AMNS00], the TSP algorithm also works for the same sampling condition proposed in the algorithms CRUST, NN and CC [AMNS00]. However, TSP only handles close single curve. Recently, a parameter-free, distance-based algorithm DISCUR proposed in [ZNYL08] requires neither parameters in the algorithm nor parameters in the sampling condition. Algorithm DISCUR connects the sample points based on the observation that human eyes tends to group near points together given that they are close enough. This observation is called nearness property. Guaranteed reconstruction of DISCUR algorithm is based on two theorems. The first theorem provides sampling condition for sampling interior points while the second theorem ensures that boundaries are detected correctly. DISCUR is proved to correctly reconstruct non-smooth open curves. However, since DISCUR uses only nearness to quantify human visual perception, it requires very dense sampling near corner areas and it may not guarantee construction when there exists a sample p which has more than one neighbor with equal shortest distance to p. To address these problems, Nguyen and Zeng proposed VICUR which considers, in addition to the nearness property, a second observation that human eye tends to connect points to form a smooth curve, named smoothness property [NZ08]. The algorithm associates each property with a parameter to determine which property has stronger influence than the other during the connection process.

1.3.2 Graphical password

In this section, a literature review of graphical password will be given. As mentioned in Section 1.1.2, graphical password is categorized into user-drawn based and picture-based. The password scheme proposed in this thesis belongs to user-drawn based graphical password. Thus, the literature review of graphical password will solely focus on this category. There are many existing scheme in the literature. We hardly mention all of them, instead, we cover the initial user-drawn based graphical password scheme and some of its variants. The first user-drawn based password was proposed by Jermyn et al., named DAS [JMM+99]. In DAS, users create their passwords by drawing pictures on a $G \times G$ grid, G is a positive integer. To pass the authentication, users have to reproduce the same images in the same order that the images were drawn. Figure 6 shows an example of a DAS password drawn on a 4×4 grid. The password is encoded by recording the sequences of cells being passed by in the same order as the drawing is created. Each cell in the grid is mapped to coordinates (x,y) that belong to $[1 \dots G] \times [1 \dots G]$. A DAS password can contain several strokes. Each strokes are separated by a "pen-up" event denoted by $(G+1, G+1)$. For example, the encoding for the image in Figure 1 is: $(2,2)$, $(3,2)$, $(3,3)$, $(2,3)$, $(2,2)$, $(2,1)$, $(5,5)$. In this example, the pair $(5,5)$ is a "pen-up" event.

The authors of DAS scheme shows that DAS's memorable password space is larger than that of textual password.

In 2008, Van Oorschot et al. proposed a graphical password dictionary that constitutes: 1) mirror symmetrically drawn patterns denoted as Class $S1$, and 2) the number of components less than four denoted as Class $S2$, based on the assumption that these two classes contain passwords which are easy to memorize [vOT08]. The assumption is supported by a user study conducted by Tao [Tao]. The study was conducted with 167 subjects showing that when no password policy is applied, 72% of created passwords fall into Class $S2$, and 41% of created passwords fall into Class $S1$. In addition, the actual memorable password space, which corresponds to the combination of Class $S1$ and Class $S2$ with the password length equal to 12 on 5×5 grid, of the original DAS scheme is only 40 bits versus 58 bits of full space.

This result shows that attacker aiming at DAS may not lack of knowledge of password distribution as assume in [JMM$^+$99].

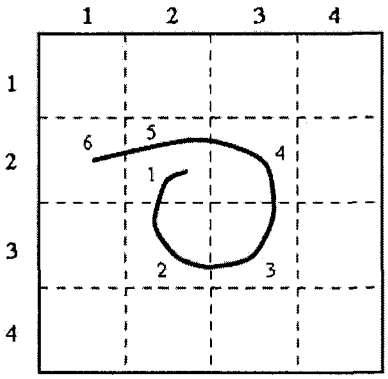


Figure 6: A DAS example password on 4x4 grid [JMM$^+$99].

In [TvO04], the authors claim that given the current user choice of password in a 5 × 5 DAS grid, increasing the grid size will increase the size of memorable password space. To minimize the negative impact of the increase in grid size on usability, the authors proposes a selection grid technique in which users select the drawing region; the region will be zoomed in and users can proceed to draw the password on the chosen region as they do in 5 × 5 DAS scheme. An example of grid selection DAS is illustrated in Figure 7.

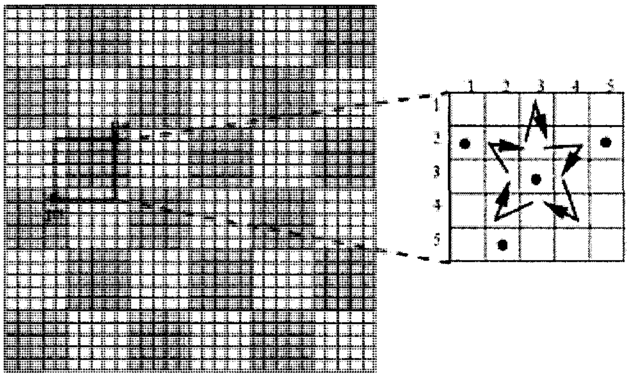


Figure 7: Grid selection [TvO04].

In the DAS scheme, the center of the grid is more likely to be chosen as the location to create passwords. This common choice makes DAS password highly predictable or susceptible to graphical dictionary attack. To address this problem Chalkias et al. [CAS06] proposed multi-grid DAS. Multi-grid DAS divides the grid into unequal-sized cells as shown in Figure 8. Based on user study of 30 participants from non-technical and technical background, the result shows the advantage of multi-grid DAS over the original DAS in a decrease in grid-centered password and an increase in the numbers of users who can memorize the location of their passwords. However, the percentage of ordering errors, which is errors occurring when the password is not drawed in the same order as initially created, stays the same and even increases in non-technical user group.

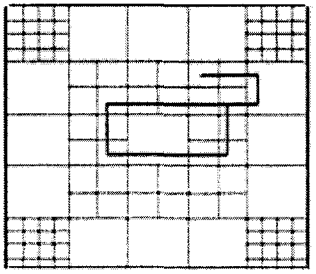


Figure 8: A multigrid DAS example [CAS06].

Later, in 2007, Dunphy et al. introduced a background image to DAS, called BDAS [DNO08]. Based on the study conducted on the total of 67 participants, the authors claims that in BDAS, users tend to create more complex passwords: the password is longer, numbers of components are greater and symmetric and centering drawings is reduced whereas the recall success rate is comparable to DAS. However, the negative impacts of the background image on the password choice is not explored in the paper although it is believed that the background image does provide attackers more information of password distribution and password patterns.

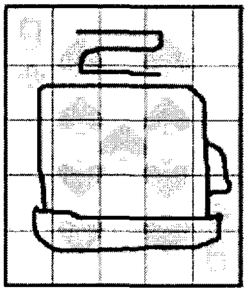


Figure 9: A BDAS password example [DY07].

Tao proposed another variant of DAS, called Pass-Go [Tao], illustrated in Figure 10.

Pass-Go requires the drawing to pass corner of the cells instead of passing the area of the cells. Therefore, the scheme allows users to create diagonal lines as well as provides users greater number of turns. For instance, starting at one point, user can go up, down, right, left, up-left, up-right, down-left, down-right to create a line; in DAS, users can only go up, down, left or right. Pass-Go has the smallest dictionary 3.3 times larger than text based password containing 7 alphanumeric character (including A-Z, a-z, 0-9). Compared to DAS, Pass-Go is claimed to be better resistant to symmetric dictionary attack.

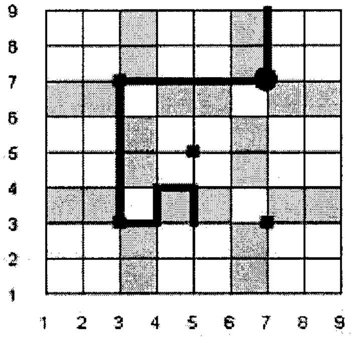


Figure 10: A Pass-Go password example [Tao].

1.4 Research contribution

The main contributions of this present thesis are listed as follows:

1. The necessary and sufficient sampling conditions are proposed and proved for a new curve reconstruction algorithm - DISCUR.

2. A new curve reconstruction algorithm, VICUR is proposed based on human vision.

3. A new graphical password scheme is proposed by taking a multidisciplinary approach combining visual perception, computational geometry and computer security.

1.5 Thesis organization

The thesis is organized as follows: Chapter 2 gives readers an overview of the relationships between vision, curve reconstruction and graphical password. The two current vision-based curve reconstruction algorithms DISCUR and VICUR are presented in Chapter 3 and 4, respectively. Chapter 5 proposes a new graphical password scheme and evaluates the memorable password space using vision-based curve reconstruction algorithm. Eventually, conclusions and future works are presented in Chapter 6 .

Chapter 2

Human-vision based curve reconstruction algorithm and graphical password: A research framework

Curve reconstruction belongs to the area of computational geometry whereas password design belongs to computer cryptography. Although they appear to be two distant research problems, this chapter shows how these two problems are related and introduces the basic concepts underlying this present research.

2.1 Logical connection between curve reconstruction and user-drawn based graphical password

Curve reconstruction deals with how to connect points so that the original curve can be reconstructed from unorganized points as shown in Figure 11. Most existing algorithms in the literature address the problem from the geometric point of view [Dey07]. Motivated by the fact that human can visualize a curve from a set of points, we have developed a set of vision based curve reconstruction algorithms [ZNYL08] [NZ08]. Those algorithms

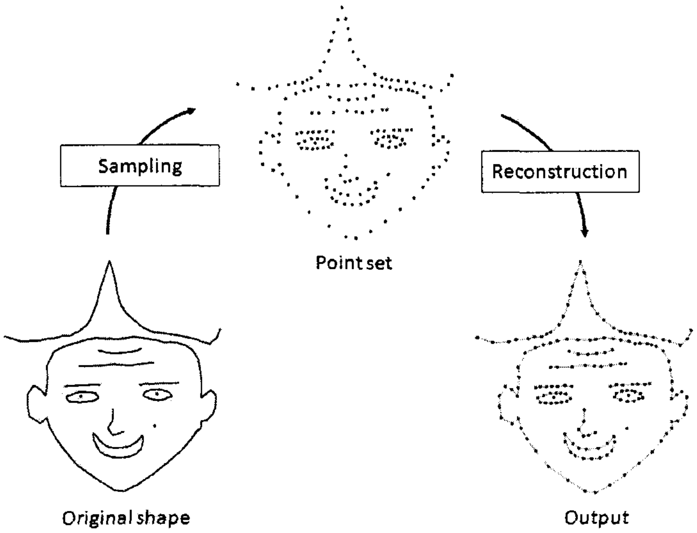


Figure 11: Overview of curve reconstruction problem.

intend to simulate human vision in the context of curve reconstruction problem.

The user-drawn based graphical password schemes let users create a password by drawing on a given canvas. Obviously, the number of possible drawings is huge. Hence, theoretically the chance for an attacker to select the correct password would be very low. The most efficient way for the attacker to select the correct password is to firstly try the drawings that users are likely to draw. For example, in the DAS scheme, users tend to create symmetric passwords, which reflect about the central horizonal and vertical axes [vOT08]. Figure 12a) and c) respectively shows a user's password and the highest probability reflection axes that will be used by an attacker as the basic knowledge of password distribution. This knowledge is used to simulate the way how users will create passwords. Similarly, in our proposed password scheme, users are given a set of points from which they can create the drawings by connecting any two points as shown in Figure 13. It is reasonable to assume that in this scheme users will draw the password intuitive to their eyes. Thus, the created password may be the subset of reconstruction result of human-vision based curve reconstruction algorithms.

The rest of this chapter will introduce Gestalt laws of human visual perception, followed by a brief analysis of the relations between Gestalt laws and curve reconstruction and between human-vision based curve reconstruction and graphical password.

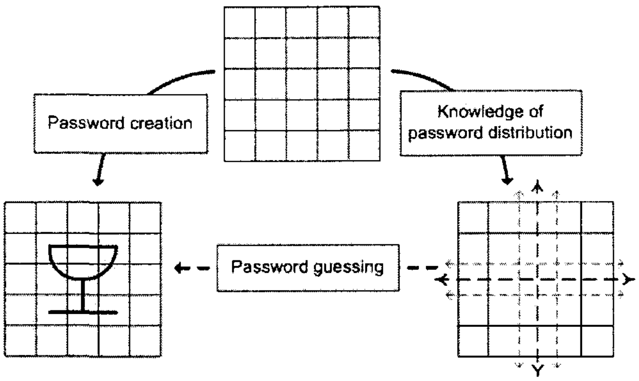


Figure 12: Cracking DAS scheme: many users' passwords are positioned in the center of the grid and have components symmetric about the central horizontal and vertical axes. Attackers can use this knowledge to crack the password.

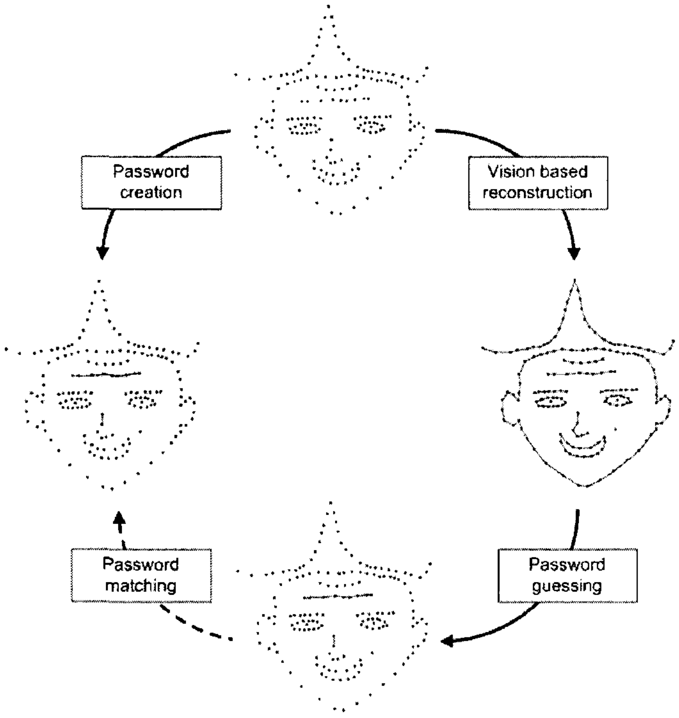


Figure 13: Cracking proposed graphical password scheme: it is assumed that users are likely to create drawings that look natural to their vision. Such drawings may be a subset of vision based curve reconstruction result, which can be used by attackers to crack the password.

2.2 Introduction to Gestalt Law

Gestalt in German means *shape* or *form*. The principles of Gestalt theory are known to be proposed by Max Wertheimer in 1912, then further developed and promoted by his colleagues Kohler and Koffka. Gestalt concept initially emerged in Ehrenfels's 1890 paper "On Gestalt Qualities" [KW07] as apposed to atomism. Atomism believes that our mind perceives the whole as summation of the parts whereas Ehrenfels believed the whole is summation of the parts plus Gestalt Qualities. Wertheimer, on the other hand, proposed the idea of Gestalt theory in which he stated that the whole is even different from the summation of its parts, the whole has an inherent structure of itself named *Gestalten* in which the parts are mutually related with each other and their properties are determined by the structural law of the Gestalten. In an attempt to find such structural laws, Max introduced five laws of perceptual organization.

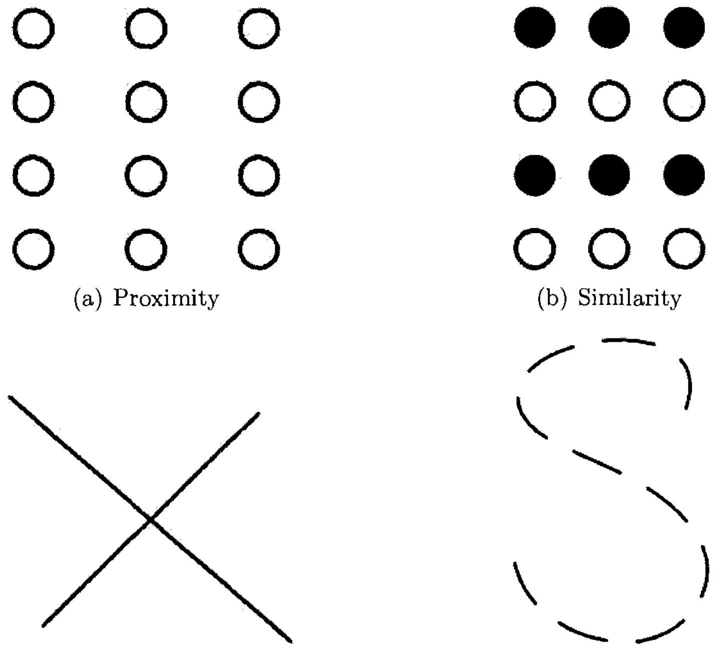
1. Law of proximity or nearness: our vision tends to perceptually group near objects together. The law of proximity is illustrated in Figure 14(a), we see three columns instead of four rows because the distance between the circles in each column is closer than the distance between the circles in each row.

2. Law of similarity: our vision tends to group together objects similar in features. In Figure 14(b), the black circles are perceptually grouped into one set and the white circles are perceptually grouped into another set.

3. Law of continuity: objects following a consistent continuous direction are perceptually grouped together. Figure 14(c) shows the law of continuity, in the picture we perceive two smooth lines cross each other instead of four line segments touching at one vertex or two V curves touching at their sharp corners.

4. Law of closure: our vision tends to perceive a whole to maintain the balance and harmony of the structure. Figure 14(d) shows the law of closure, we perceptually complete the gap between the lines to perceive the complete shape S.

5. Law of common fate: our vision tends to group objects that move in the same motion.



(a) Proximity

(b) Similarity

(c) Continuity

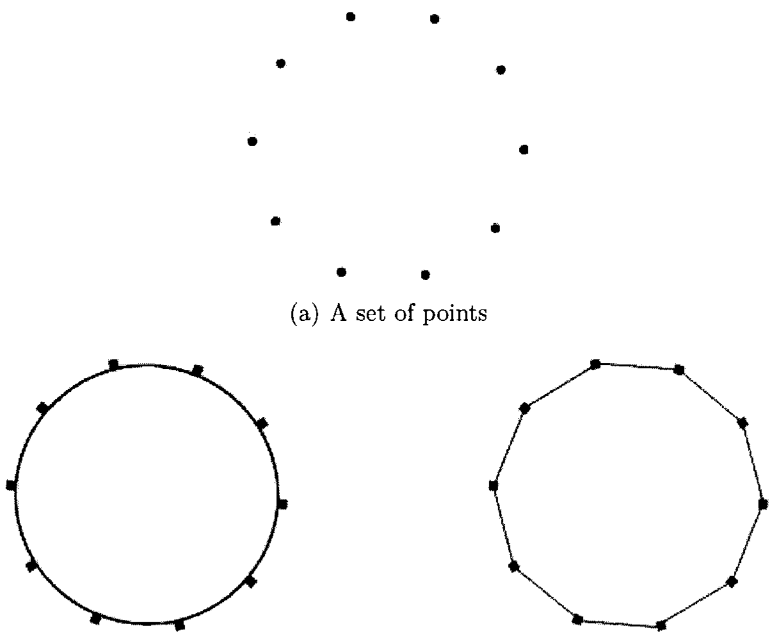
(d) Closure

Figure 14: Gestalt laws of perceptual organization.

These laws are all rooted in the law of Pragnanz which states that human mind tends to group the parts to a simple formation. The Gestalt laws of perceptual organization had an enormous impact on the field of perception at Wertheimer's time and continues to leave its trace in modern perceptual research. However, it should be noted that Gestalt theory is not merely the theory of perception. Rather, the study of perception is used to demonstrate the Gestalt theory.

2.3 Relationship between Gestalt Law and vision-based curve reconstruction

According to the Gestalt law of closure, human tend to form objects that are incomplete to form an entire structure. As illustrated in Figure 15(a), we can perceive a round shape out of a collection of separate points. This property of human perception motivates us to develop a vision-based curve reconstruction algorithm which can connect points into curves that are similar to the curves perceptually constructed by human vision. Thus, the result of the vision-based reconstruction from the points in Figure 15(a) is a polygon as shown in Figure 15(c), which is similar to the curve perceived by our mind shown in Figure 15(b).



(a) A set of points

(b) How human sees

(c) Correct reconstruction result of a vision-based algorithm

Figure 15: Human perception and vision-based curve reconstruction algorithm.

In the context of curve reconstruction, we observed that the law of proximity and the law of continuity are relevant. Therefore, our algorithm is developed based on two criteria: 1) nearest points should be connected, and 2) points should be connected to form a smooth curve.

A good curve reconstruction should be able to correctly construct closed curves, open curves, curve with sharp corners and multiple curves. To achieve this objective, we introduce a function called connectivity function to determine when a sample point p should be connected to a curve T. The connectivity function can be denoted by

$$E[p,T] = f(p,V) \tag{4}$$

where V is a vector that includes statistical properties of the curve segment T, such as the distance mean, distance standard deviation, angle mean, and angle standard deviation. The function $f(p, V)$ can be obtained through experiments or through observations.

However, in some cases, conflict may arise as depicted in Figure 16. According to criterion number one, point q_4 should be connected to point q_5 because point q_5 is closer to point q_4 than point q_6, but according to criteria number two, point q_4 should be connected to point q_6 because the connection between q_4 and q_6 will result in a smoother curve. To solve this problem, we propose two solutions. The first solution is to avoid such a conflict by considering only nearness property. This solution implies that any points connected based on nearness property also forms a smooth curve. The second solution considers both criteria and introduces additional parameters to evaluate which criteria should be followed when conflict occurs. The two solutions are implemented in DISCUR and VICUR algorithms, respectively. More details about these two algorithms will be given in Chapter 3 and Chapter 4 of the present thesis.

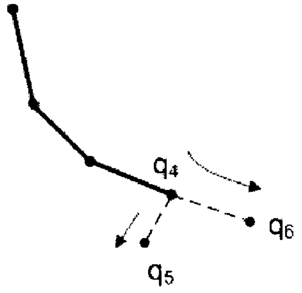


Figure 16: A case of conflict.

2.4 Relationship between human-vision based curve reconstruction algorithms and graphical password

In creating a password, users try to satisfy the following requirements explicitly or implicitly:

1. The password has to be easily to remember.

2. The password needs to be easy to input.

3. The password must be difficult to be cracked.

To satisfy the first requirement, the graphical password should be natural to human vision since the major advantage of the graphical password is its intuitiveness, which is advantageous for human memory. To satisfy the second requirement, the length of the password must not be too long. To satisfy the third requirement, the password either has to be random enough so that it is hard to be guessed or the password space has to be large enough so that it will be computationally infeasible to select the correct password for an attacker.

From the first requirement, it is reasonable to assume that users will select the passwords that look natural or meaningful to their vision. Consequently, the attacker can reconstruct the entire curves by using human-vision based curve reconstruction algorithms and the users' passwords may be parts of the reconstructed curves.

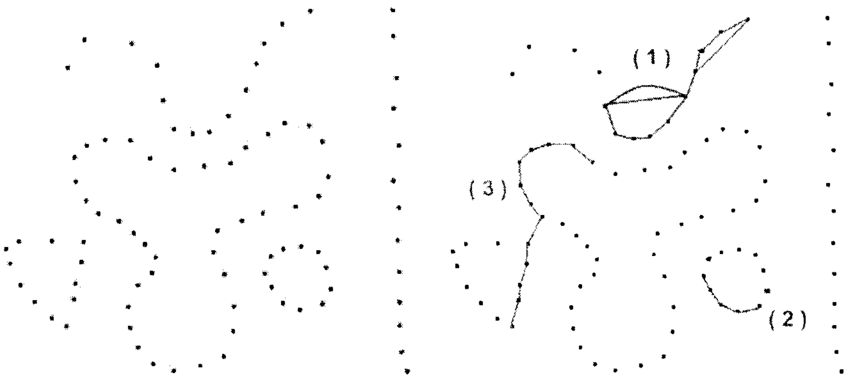
Assumingly, from the attacker's point of view, there are three main ways to crack the proposed password:

1. Perform an exhaustive search.

2. Reconstruct the curves on the whole point set and take the reconstructed curves as the foundation for password guessing.

3. Choose a subset of points from the point set, reconstruct the curves on the subset and take the reconstructed curves as the foundation for password guessing.

If the password space is relatively large, the first approach is unrealistic. Both the second and the third approach can be conducted by using vision-based curve reconstruction algorithms. The difference is the curve reconstruction algorithms construct curves on the whole point set in the second approach while it constructs curves on a subset of the point set in the third approach. Figure 2.4b) shows different password drawings. The first approach can help attackers to find the drawing (1), which consists of multiple edges or intersecting curves; the second approach can help attackers to find the drawing (2), which is the subset of the curve reconstruction result and the third approach can produce the drawing (3), which is the subset of the reconstruction result on the subset of the point set.



(a) A set of points (b) Graphical password

Figure 17: An example of a graphical password

2.5 Summary

This chapter discusses the relationships between visual perception, curve reconstruction, and graphical password. Visual perception is the foundation of curve reconstruction and graphical password in that a curve should be reconstructed from unorganized points in a way that is natural to human vision and a graphic password created from unorganized points should also look natural to the owner's visual perception. Human vision based curve reconstruction will be used to evaluate the security property of the proposed graphical password scheme. Based on this research framework, the following three chapters will address the following three issues. First, how to reconstruct the curves based only on nearness property. Second, how to reconstruct the curves by considering the nearness and the smoothness properties of human visual perception, third, how to evaluate the proposed graphical password scheme based on the principles of human vision and the capacity of vision based curve reconstruction algorithms.

Chapter 3

DISCUR algorithm: simulation of nearness property of human vision in the context of curve reconstruction

DISCUR algorithm reconstructs curves using Gestalt perceptual law of proximity which means that satisfying nearness property implies that smoothness property is also satisfied. DISCUR is guaranteed to reconstruct curves correctly from unorganized points

3.1 Simulation of nearness property

Based on the vision function defined in Equation 4, the following two rules can be used to determine the connectivity of two potentially connectable samples:

Rule 1: point-curve connectivity. For a curve $T = [q_1, q_2, ..., q_i], i > 1$, which is partially reconstructed from a sample set S, suppose that there exists a sample point $p \in S$ that is the nearest neighbor to $q = q_i$ (or q_1). If $d(p, q) < E[p, T_q]$, then p and q can be connected.

Rule 2: curve-curve connectivity. For two curves $T^1 = [q_1, q_2, ..., q_i], T^2 = [p_1, p_2, ..., p_j]$, $i, j > 1$, which is partially reconstructed from a sample set S, if q_1 (or q_i) and T^2 or p_1 (or p_j) and T^1 can be connected by Rule 1, then these two curves can be connected.

In the following, a concrete form of Equation 4, which considers only distance in the equation, is given as follows [ZNYL08]:

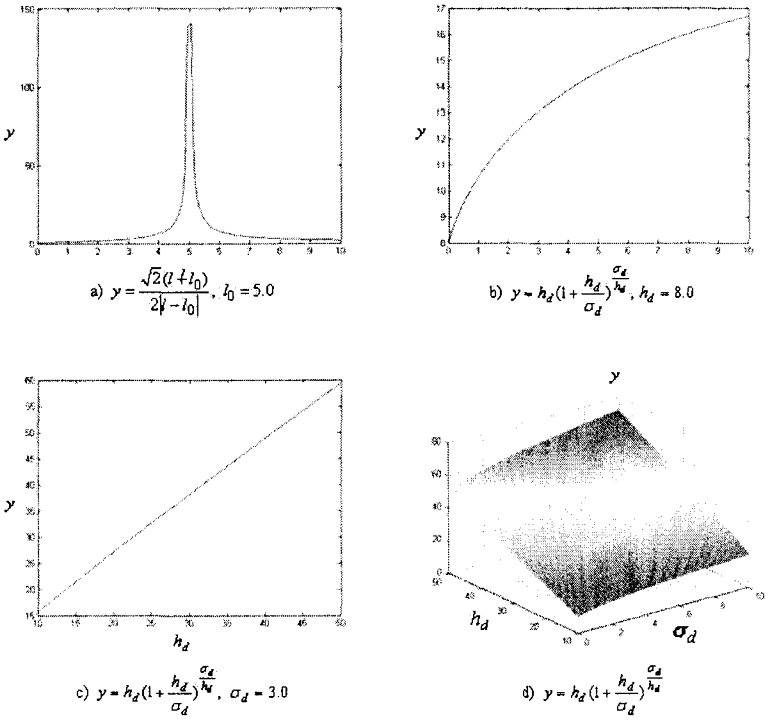
$$E[p,T_q] = h_d \frac{h}{s} (1 + \frac{h_d}{\sigma_d})^{\frac{\sigma_d}{h_d}} \tag{5}$$

where $h = \frac{l+l_0}{2}$, $s = \frac{|l-l_0|}{\sqrt{2}}$, $l_0 = d(q_{i-1}, q_i)$ (or $l_0 = d(q_2, q_1)$), and $l = d(p, q_i)$ (or $l = d(p, q_1)$). σ_d and h_d are defined in Equation 1 and 2, respectively.

Equation 5 shows that the connectivity between the sample p and the curve T_q depends on two relations: the relation of p to T_q, defined by h_d and σ_d, and the relation of p to its nearest segment $q_{i-1}q_i$ (or q_1q_2), defined by h and s.

First, let us examine the case when only one edge has been connected, i.e., $T_q = [q_1, q_2]$. Therefore, $h_d = l_0 = d(q_1, q_2)$, $\sigma_d = 0$. Now we want to connect q_3 to T_q, let $l = d(q_3, q_2) < d(q_3, q_1)$. In this case, Rule 1 is reduced to $l < l_0 \frac{h}{s}$. It is noted that if the difference between l and l_0 becomes larger, the ratio of $\frac{h}{s}$ becomes lower, which means that the probability of the connection becomes lower. On the other hand, when the difference between l and l_0 is smaller, the probability of connection is higher.

Secondly, let us examine when many edges have been connected, i.e., $T_q = [q_1, q_2, ...q_i]$. In this case, Rule 1 identifies two factors that affect the connectivity: the reconstructed part of the curve and the edge nearest to the sample to be connected. The former factor exerts global requirement on the new edge, the later factor a local requirement. These two requirements imply that a new edge should not bring about abrupt change to the already constructed part in terms of length and that the new edge should be compatible with its neighbor.



a) $y = \frac{\sqrt{2}(l+l_0)}{2|l-l_0|}$, $l_0 = 5.0$

b) $y = h_d(1+\frac{h_d}{\sigma_d})^{\frac{\sigma_d}{h_d}}$, $h_d = 8.0$

c) $y = h_d(1+\frac{h_d}{\sigma_d})^{\frac{\sigma_d}{h_d}}$, $\sigma_d = 3.0$

d) $y = h_d(1+\frac{h_d}{\sigma_d})^{\frac{\sigma_d}{h_d}}$

Figure 18: Graphic illustration of point-curve connectivity.

Figure 18a) shows that it becomes more probable for the point p to be connected to the curve T_q as l approaches l_0. In the extreme case, if $s = 0$, then $l = l_0$ and $\frac{h}{s}$ becomes infinity. The point p should be added into the curve T_q. Figure 18b) illustrates that the connectivity between p and T_q increases as σ_d of T_q becomes larger. In the case where $\sigma_d \to 0$, $(1 + \frac{h_d}{\sigma_d})^{\frac{\sigma_d}{h_d}} = \lim(1 + \frac{h_d}{\sigma_d})^{\frac{\sigma_d}{h_d}} = 1$. As a result, the criterion is reduced to $l < h_d\frac{h}{s} = \frac{\sqrt{2}l_0(l+l_0)}{2|l-l_0|}$ and only the boundary segment of T_q will have an effect on the connectivity. Intuitively, the value of σ_d indicates how evenly the curve T_q is sampled. The more unevenly the curve is sampled, the further a connectable sample can be away from the boundary q of the curve T_q. Figure 18c) presents the third case where a greater h_d, resulted from fewer sampling points in the curve, will enhance the probability of p being connected to T_q. Figure 18d) gives the combined effect of h_d and σ_d on the connectivity between p and T_q.

3.2 Algorithm

From the rules in Section 3.1, an algorithm named DISCUR is developed to reconstruct multiple simple curves that may be open, close, and with or without sharp corners. The major steps of the algorithm is given in Figure 19.

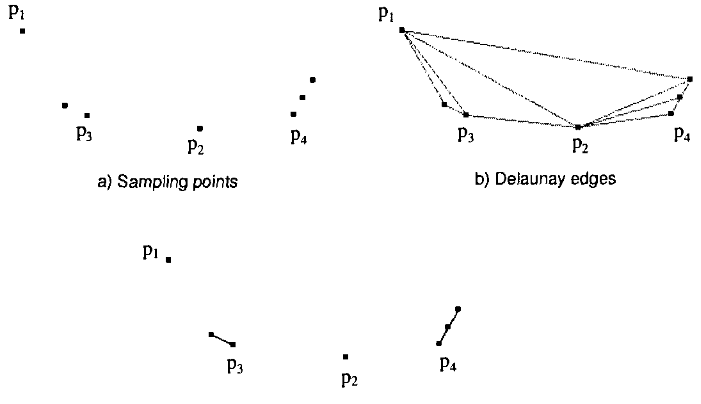
```
Algorithm $DISCUR(SampleSet : S)$
  Step 1 - Delaunay triangulation and initialization
  Step 2 - Determining the connectivity of Delaunay edges
  Step 3 - Updating the connectivity of Delaunay edges
  Output the reconstructed curves
```

Figure 19: Main steps of DISCUR.

DISCUR takes a set of sampling points as input and reconstructs the curve in three main steps. Step 1 computes the Delaunay triangulation for the sample set S and initializes the connectivity properties of sample points and Delaunay edges. 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 processing. Step 3 processes the Delaunay edges retained in Step 2 and completes the curve reconstruction.

In the first step, the algorithm computes Delaunay triangulation, marks all these Delaunay edges as 0 and initializes the degree for each sampling point to 0. Variable $mark[e]$ for a Delaunay edge e has two possible values: 0 and 1. If the edge e is not yet processed for connectivity then 0 is assigned to $mark[e]$ in Step 2 and to 1 in Step 3; however, if e is found to be the shortest edge but cannot be connected because of its connectivity value E defined by Equation 5 then $mark[e] = 1$ in Step 2 and $mark[e] = 0$ in Step 3. Variable $degree[p]$ for a sample p is used to track the number of shortest Delaunay edges that are adjacent to p. Only two nearest neighbors to p should be considered for connection to p, which makes 2 the maximum degrees of a sample. As soon as $degree[p]$ is equal to 2, it is not necessary to check other points for connection to p.

It should be noted that there are cases when $degree[p] = 2$ and p is still a free point. Figure 20 illustrates such case. Figure 20b) shows that there is Delaunay edge between points p_1 and p_2 but p_1 and p_2 are not connected to each other even they are free points as shown in Figure 20c). The reason is that *degree*[p_2] is already 2 because p_3 and p_4 should have been connected to p_2 if their connectivity value E were greater than their distances. This yields a reconstruction acceptable to human perception as shown in Figure 20c). Pseudocode of the first step is given in Figure 21.



a) Sampling points

b) Delaunay edges

c) Reconstruction result

Figure 20: Meaning of variable degree.

```
Step 1 Delaunay triangulation and initialization
Compute the Delaunay triangulation of $S$
Let $D_e$ be the set of Delaunay edge
for all $e \in D_e$ do
    $mark[e] \leftarrow 0$
end for
for all $p \in S$ do
    $degree[p] \leftarrow 0$
end for
```

Figure 21: Step 1 of DISCUR.

The second step determines the connectivity of each shortest Delaunay edge. The pseudocode is shown in Figure 22. As the shortest Delaunay edge e is found, the degree for each vertex incident to the edge e will increase by 1 (line 2). When the *degree* values of both vertices of the edge e are 0, they are connected directly and the Delaunay edge is removed (line 3 and 4). Otherwise, the connectivity value at each vertex should be computed (line 6 to 12). Two vertices should be connected if either Rule 1 or Rule 2 is satisfied (line 13 and 14). If two vertices cannot be connected at this step, the edge is marked as 1 so that it can be considered again in Step 3 (line 16). The reason for this is that the connectivity value E may change as the curve may extend. As mentioned in Step 1, when two nearest neighbors to a sample point p are found, other neighbors should not be considered. Therefore, all other adjacent Delaunay edges to p will be removed (line 19 to 25).

```
Step 2 Determining the connectivity of Delaunay edges
 1: for each shortest Delaunay edge $e = [p_i, p_j] \in D_e$ and $mark[e] = 0$ do
 2:   $degree[p_i] \leftarrow degree[p_i] + 1$, $degree[p_j] \leftarrow degree[p_j] + 1$
 3:   if both $p_i$ and $p_j$ are free points then
 4:     Connect $p_i$ and $p_j$, $D_e \leftarrow D_e - \{e\}$
 5:   else
 6:     for each $p_1 \in \{p_i, p_j\}$ do
 7:       $p_2 \leftarrow p_i + p_j - p_1$
 8:       $E[p_2, T_{p_1}] \leftarrow 0$
 9:       if $p_1$ is an endpoint of a curve $T_{p_1}$ then
10:         Compute the connectivity value $E[p_2, T_{p_1}]$
11:       end if
12:     end for
13:     if $d(p_i, p_j) < \max(E[p_i, T_{p_j}], E[p_j, T_{p_i}])$ then
14:       Connect $p_i$ and $p_j$, $D_e \leftarrow D_e - \{e\}$
15:     else
16:       $mark[e] \leftarrow 1$
17:     end if
18:   end if
19:   for each $p \in \{p_i, p_j\}$ do
20:     if $degree[p] = 2$ then
21:       for all $e' \in D_e$ incident to $p$ and $mark[e'] = 0$ do
22:         $D_e \leftarrow D_e - \{e'\}$
23:       end for
24:     end if
25:   end for
26: end for
```

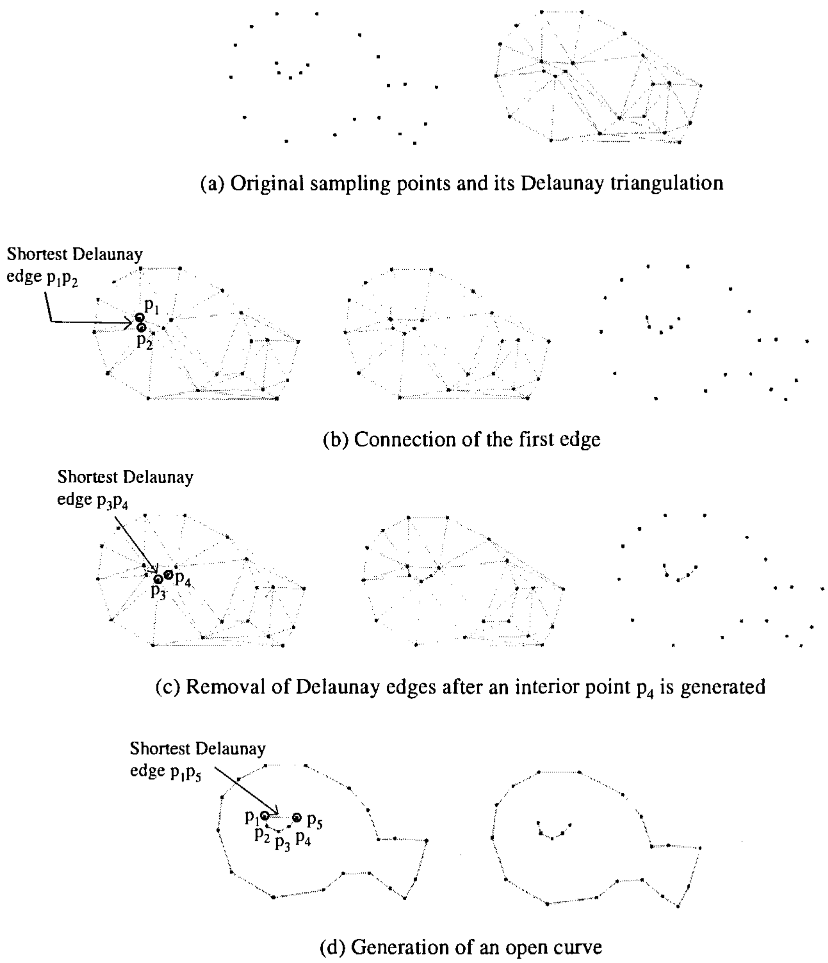
Figure 22: Step 2 of DISCUR.

The third step reconsiders Delaunay edges retained in Step 2 whose values of *mark* were assigned to 1. For each edge $e = [p_i, p_j]$, if $d(p_i, p_j) \geq \max(E[p_i, T_{p_j}], E[p_j, T_{p_i}])$, e is marked as 0 and will be excluded from consideration in the for loop starting at line 1. This edge $e = [p_i, p_j]$ can only be reconsidered when any curve incident to p_i or p_j is updated (line 8 to 14). If $d(p_i, p_j) < \max(E[p_i, T_{p_j}], E[p_j, T_{p_i}])$, then p_i and p_j will be connected. As a result, the curve is extended, which changes the connectivity value E for some unconnected Delaunay edges. Thus, as long as the curve is extended, any Delaunay edge incident to the endpoints of this curve should be checked for connection (line 6 to 16). This step terminates when all Delaunay edges have been examined for connection and no more connection can be made. The pseudocode of Step 3 is given in Figure 23.

```
Step 3 Updating the connectivity of Delaunay edges
 1: for each Delaunay edge $e = [p_i, p_j] \in D_e$ and $mark[e] = 1$ do
 2:   if $d(p_i, p_j) < \max(E[p_i, T_{p_j}], E[p_j, T_{p_i}])$ then
 3:     Connect $p_i$ and $p_j$
 4:     $D_e \leftarrow D_e - \{e\}$
 5:     $T^1 \leftarrow [T_{p_i} | T_{p_j}]$
 6:     repeat
 7:       $T^2 \leftarrow \emptyset$
 8:       for each Delaunay edge $e = [p_m, p_n]$ incident to an endpoint of $T^1$ do
 9:         if $d(p_m, p_n) < \max(E[p_m, T_{p_n}], E[p_n, T_{p_m}])$ then
10:           Connect $p_m$ and $p_n$
11:           $D_e \leftarrow D_e - \{e\}$
12:           $T^2 \leftarrow [T_{p_m} | T_{p_n}]$
13:         end if
14:       end for
15:       $T^1 \leftarrow T^2$
16:     until $T^1 = \emptyset$
17:   else
18:     $mark[e] = 0$
19:   end if
20: end for
```

Figure 23: Step 3 of DISCUR.

The procedures included in this algorithm is illustrated in Figure 24, where images without Delaunay triangulation are included to improve the visibility of reconstructed curves. Figure 24a) shows the input of this algorithm, which is a set of sampling points; the corresponding Delaunay triangulation is also given. In Figure 24b), $[p_1, p_2]$ is the shortest Delaunay edge and both p_1 and p_2 are free points; the first edge $[p_1, p_2]$ is then connected and removed from D_e. Figure 24c) shows an intermediate step where $[p_3, p_4]$ is the current shortest Delaunay edge and the sample p_4 is an interior point. After the edge $[p_3, p_4]$ is connected, all the Delaunay edges connected to p_4 are removed from D_e. In Figure 24d), the current shortest Delaunay edge is $[p_1, p_5]$ and both p_1 and p_5 are endpoints. Obviously, these two points has a shorter distance than many connected edges. The edge $[p_1, p_5]$ is still a Delaunay edge because it was marked as 1 when it could not be connected earlier



(a) Original sampling points and its Delaunay triangulation

(b) Connection of the first edge

(c) Removal of Delaunay edges after an interior point p_4 is generated

(d) Generation of an open curve

Figure 24: Example of the reconstruction process by using DISCUR.

due to the connectivity value between these two points. At this stage, $[p_1, p_5]$ is the only Delaunay edge left, and $T_{p_1} = T_{p_5} = [p_1, p_2, p_3, p_4, p_5]$. The coordinates of those five points are $p_1(114, 131), p_2(119, 151), p_3(143, 162), p_4(164, 151), p_5(180, 134)$, respectively. In terms of Equation 5, $E[p_1, T_{p_5}] = 44.274, E[p_5, T_{p_1}] = 40.375$. Since $d(p_1, p_5) = 66.07 > 44.27$, points p_1 and p_5 should not be connected and the Delaunay edge $[p_1, p_5]$ is removed from the set D_e and no more Delaunay edge exists. The curve reconstruction process ends.

Figure 25 gives an example of reconstruction of curves with multiple features by using this present algorithm.

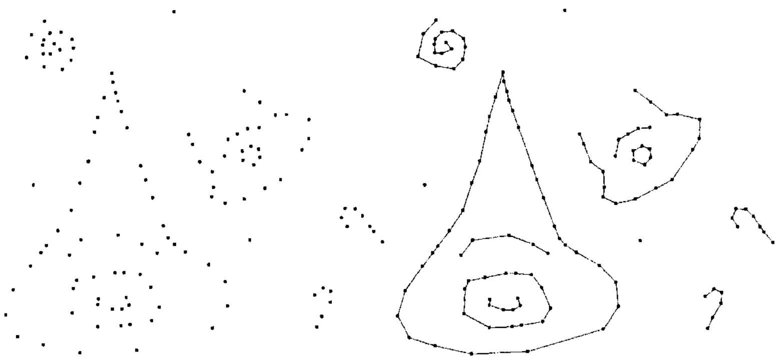


Figure 25: Reconstruction of curves with multiple features.

3.3 Necessary and sufficient sampling conditions

Theorem 3.3.1 provides necessary and sufficient conditions for sampling the interior points of a curve for DISCUR to work correctly. This deals with Case 1 and 2 as shown in Figure 26. Theorem 3.3.2 provides necessary and sufficient conditions for the sampling boundary points.

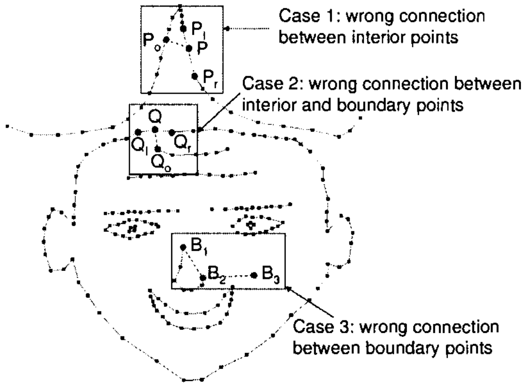

Case 1: wrong connection between interior points
P_o
P_l
P
P_r
Case 2: wrong connection between interior and boundary points
Q
Q_l
Q_r
Q_o
B_1
B_2
B_3
Case 3: wrong connection between boundary points

Figure 26: Overview of sampling conditions for DISCUR.

Theorem 3.3.1 *Suppose that S is a set of sample points on a curve or a collection of curves T. 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, assume that $r_p = d(p, t_p^1) = max\{d(p, t_p^1), d(p, t_p^2)\}$ and $C[p, t_p^1] = E[p\prime, T_{p\prime}] = max\{E[p, T_{t_p^1}], E[t_p^1, T_p]\}$. Furthermore, let N_p be a subset of S, such that $N_p = \{q \in S : d(p, q) \leq r_p$ and $q \neq p, t_p^1, t_p^2\}$. The point p will be connected to its neighbors t_p^1 and t_p^2 by Algorithm DISCUR, if and only if the following conditions are* *satisfied:*

$$r_p < C[p, t_p^1], \exists T_{p\prime} \subseteq T, (p\prime = p \vee p\prime = t_p^1), \textit{ and the connectivities}$$
$$\textit{of all the segments in } T_{p\prime} \textit{ do not depend on the connection of } [p, t_p^1]. \tag{6}$$

$$[|N_p| = 0] \vee [d(p, q) > r_q \wedge q \neq t_q^2, \forall q \in N_p]. \tag{7}$$

where $E[p, T_{t_p^1}]$ *and* $E[t_p^1, T_p]$ *are the vision function defined in the form of Equation 5 and* $r_q = d(q, t_q^1) = max\{d(q, t_q^1), d(q, t_q^2)\}$.

The following will provide a mathematical proof of Theorem 3.3.1. The proof of this theorem includes two parts. First, the sufficient condition will be proven by showing that the algorithm will reconstruct the curve correctly when both conditions in (6) and (7) are satisfied. Secondly, the necessary condition will be proven by showing that the algorithm will not guarantee the correct connection when any of the condition in (6) and (7) is not satisfied.

[Proof] Proof of sufficient condition.

Case 1: consider the conditions $r_p < max\{E[p, T_{t_p^1}], E[t_p^1, T_p]\}, \exists T_{t_p^1}, T_p$ and $|N_p| = 0$. For any point $p \in S$, there is no other point q other than its neighbors t_p^1 and t_p^2 such that $d(p, q) \leq r_p$ because $|N_p| = 0$. Thus $[p, t_p^2]$ and $[p, t_p^1]$ must be the shortest and the second shortest edges incident to point p. Hence, for the shortest edge $e = [p_i, p_j] \in D_e$ found in Algorithm DISCUR, p_j must be one of the two neighbors of p_i. There are following three possibilities:

1. Both p_i and p_j are free points. In this case, p_i and p_j will be connected directly.

2. One of p_i and p_j is a free point. Without loss of generality, let us assume that p_j is a free point. Let $p_i = p$. If $p_j = t_p^2$, then $d(p_i, p_j) < d(p_i, t_p^1)$. In this case, the edge $[p_i, p_j]$ would have to be considered before the edge $[p_i, t_p^1]$ when p_i was still a free point. Hence, p_j must be t_p^1. By (6) we have $r_p < max\{E[p, T_{t_p^1}], E[t_p^1, T_p]\}, \exists T_{t_p^1}, T_p$. Therefore, p_i and p_j will be connected by the algorithm DISCUR.

3. Both p_i and p_j are end points. Suppose that t_i and t_j are respectively the other

neighbors of p_i and p_j. If $d(p_i, p_j) < d(p_i, t_i)$ and $d(p_i, p_j) < d(p_j, t_j)$, then the edge $[p_i, p_j]$ would have to be considered before the edges $[p_i, t_i]$ and $[p_j, t_j]$ when both p_i and p_j were still free points. Hence, $d(p_i, p_j)$ must be greater than or equal to one of $d(p_i, t_i)$ and $d(p_j, t_j)$. Assume that $d(p_i, p_j) \geq d(p_i, t_i)$. Hence, $p_i = p$ and $p_j = t_p^1$. By (6) we have $r_p < max\{E[p, T_{t_p^1}], E[t_p^1, T_p]\}, \exists T_{t_p^1}, T_p$. Therefore, p_i and p_j will be connected by the algorithm DISCUR.

Case 2: consider the condition $r_p < max\{E[p, T_{t_p^1}], E[t_p^1, T_p]\}, \exists T_{t_p^1}, T_p$ and $|N_p| \neq 0$ but $d(p, q) > r_q \wedge q \neq t_q^2, \forall q \in N_p$.

Without loss of generality, suppose that $w \in N_p$ is the sample closest to p. In this situation, w is not a boundary point ($w \neq t_w^2$) and $r_w < max\{E[w, T_{t_w^1}], E[t_w^1, T_w]\}, \exists T_{t_w^1}, T_w$ because (6) applies to any sample point. Moreover, $p \notin N_w$ since $d(p, w) > r_w$. So, w will be connected with its own neighbors, which do not include p. As a result, w becomes an interior point and $degree(p) = 2$. According to the algorithm DISCUR, Delaunay edge $[w, p]$ will be removed from D_e. This process applies to all the samples in N_p, which will make $|N_p|$ to be 0. Therefore, p can be correctly connected with its neighbors in terms of Case 1.

In summary, if the conditions in (6) and (7) are met, the curves will be correctly reconstructed. This proves the sufficient condition.

Proof of Necessary condition.

Case 1: suppose that the condition given in (6) is not met.

In this case, there exists at least one sample point $p \in S$ such that $r_p \geq max\{E[p, T_{t_p^1}], E[t_p^1, T_p]\}, \forall T_{t_p^1}, T_p$. According to Algorithm DISCUR, p and t_p^1 will not be connected, which is not correct.

Case 2: suppose that the condition given in (6)is met; but $|N_p| \neq 0$ and there exists a boundary point $w \in N_p$ such that $d(p, w) > r_w$.

Without loss of generality, assume that w is the closest to p, among all the boundary points in N_p. Since w is a boundary point, $t_w^2 = w$, $d(w, t_w^2) = 0$ and $r_w = d(w, t_w^1)$. However, according to Algorithm DISCUR, $[w, t_w^1]$ and $[p, w]$ are respectively the shortest and the second shortest edges incident to w. It can be assumed that no other points are closer to

w than p. Since $d(p,w) < r_p$, the degrees of p and w will be increased by 1. As a result, one of the two neighbors of the point p will not be connectable to p.

Case 3: suppose that the condition given in (6) is met, but $|N_p| \neq 0 \wedge d(p,q) \leq r_q \wedge q \neq t_q^2, \forall q \in N_p$.

In this case, there exists at least one non-boundary sample point p such that a non-boundary point $q \in N_p$ will make $d(p,q) \leq r_q$ and $d(p,q) \leq r_p$. Under this circumstance, there exist following possible scenarios:

1. If $d(p,q) < r_q$ and $d(p,q) < r_p$

   - Both p and q are free points. As a result of Algorithm DISCUR, p and q will be connected, which makes at least one of p's(and q's) own neighbors not connectable to p (and q).

   - Neither p nor q is a free point, $[p,q]$ is the shortest Delaunay edge incident to both p and q. According to Algorithm DISCUR, the degrees of p and q will be increased to 2. As a result, all other Delaunay edges incident to p and q will be removed from D_e, which makes p's(and q's) second neighbor not connectable to p (and q).

   - Without loss of generality, suppose that p is free but q is not. According to Algorithm DISCUR, the degrees of p and q will be increased by 1 since $d(p,q) < r_q$ and $d(p,q) < r_p$. As a result, $degree(q) = 2$ and all other Delaunay edges incident to q will be removed from D_e. This makes q's second neighbor not connectable to q.

2. If $d(p,q) = r_p$ or $d(p,q) = r_q$

   Edges $[p,q]$ and $[p,t_p^1]$ (or $[q,t_q^1]$) are the second shortest incident to p (or q). By (6) we have $d(p,q) = r_p < max\{E[p,T_{t_p^1}], E[t_p^1,T_p]\}, \exists T_{t_p^1}, T_p$ (or $d(p,q) = r_q <$

$max\{E[q, T_{t_q^1}], E[t_q^1, T_q]\}$, $\exists T_{t_q^1}, T_q)$. Algorithm DISCUR will choose arbitrarily between $[p, q]$ and $[p, t_p^1]$ (or $[q, t_q^1]$) for connection. If $[p, q]$ is chosen to be connected, then at least one of p's (and q's) own neighbors will not be connected to p (and q).

In summary, if the conditions (6) or (7) can not be satisfied, at least one sample point cannot be guaranteed for the connection to its own neighbor. This proves the necessary condition.

Theorem 3.3.2 *Suppose that S is a set of sample points on a curve or a collection of curves. For every boundary point $p \in S$, there exists a set B_p, which is a subset of S, such that $B_p = \{q \in S : [p, q]$ is a Delaunay edge$\}$. The point p will guarantee not to be connected to any point in B_p by Algorithm DISCUR, if and only if the following two conditions are satisfied:*

1. *All interior points are sampled according to Theorem 3.3.1*
2. $d(p, q) \geq max\{E[p, T_q], E[q, T_p]\}, \forall q \in B_p, T_p, T_q.$

where $E[p, T_q]$ and $E[q, T_p]$ are the vision function defined in the form of Equation 5.

This theorem is self-evident. If both conditions (1) and (2) are satisfied, the curve is constructed correctly (as proved in Theorem 3.3.1) and boundary points are not connected (from condition (2)). If condition (1) is not satisfied, the curve does not guarantee a correct reconstruction as proved in theorem 3.3.1. If condition (2) is not satisfied, boundary points p and q are wrongly connected.

3.4 Comparisons

In this section, DISCUR and existing algorithms, particularly CRUST [ABE98], NN-CRUST [DK99], and GATHAN [DW02], are made .

Table 1 [Dey07] and 2 show 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.

Table 1: Scope of curve reconstruction algorithms [Dey07]

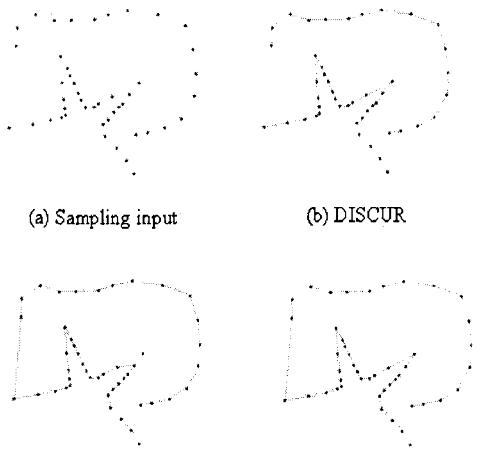
Algorithm	*Sampling*	*Smoothness*	*Boundary*	*Components*
CRUST [ABE98]	Non-uniform	Required	None	Multiple
NN-CRUST [DK99]	Non-uniform	Required	None	Multiple
TSP [AM00]	Non-uniform	Not required	Must be known	Single
CC [DMR99]	Non-uniform	Required	Any number	Multiple

Table 2: Scope of GATHAN and DISCUR

Algorithm	*Sampling*	*Smoothness*	*Boundary*	*Components*
GATHAN [DW01] [DW02]	Non-uniform	Not required Guaranteed	Any number No guarantee	Multiple No guarantee
DISCUR [ZNYL08]	Non-uniform	Not required Guaranteed	Any number Guaranteed	Multiple Guaranteed

3.4.1 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 have certain parameters as inputs.



(a) Sampling input

(b) DISCUR

(b) GATHAN angle = 10

(d) GATHAN angle = 23

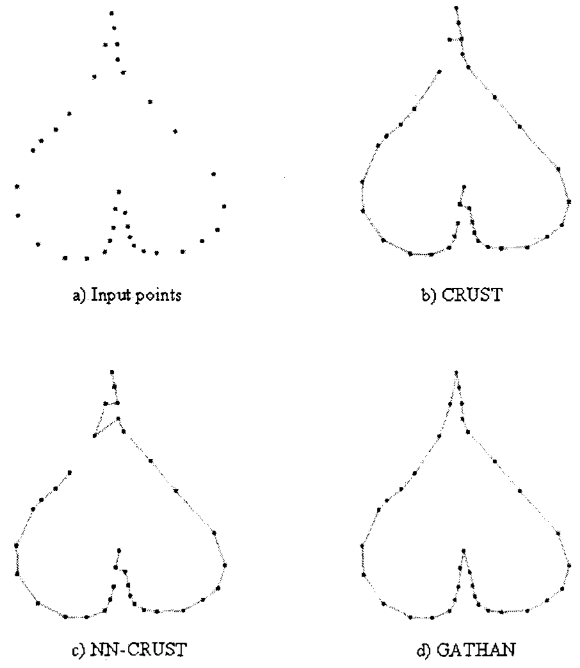
Figure 27: Reconstruction of GATHAN with different parameters.

Take the point cloud shown in Figure 27a) as an example. our algorithm DISCUR generates two components as shown in Figure 27b), which conforms to human perception. GATHAN, however, depends on parameters that in turn depend on the shape of the curve to be reconstructed. Figure 27c) and d) show some reconstruction results from GATHAN using minimum corner angle as a parameter. When the minimum corner angle is set to 10°, GATHAN would wrongly connect the curve as shown in Figure 27c). After the parameter value is set to 23°, GATHAN can reconstruct the sharp corner of the curve correctly.

3.4.2 Sharp corners

In the case involving sharp corners as shown in Figure 28, 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.

For the samples given in Figure 28, our parameter-free algorithm DISCUR does not obtain desired output since the sampling near the sharp corner violates the sampling condition. However, the problem can be corrected easily as in Figure 29b) by adding more points to the local area where the sampling connection is violated.


a) Input points

b) CRUST

c) NN-CRUST

d) GATHAN

Figure 28: Reconstruction of sharp corners.

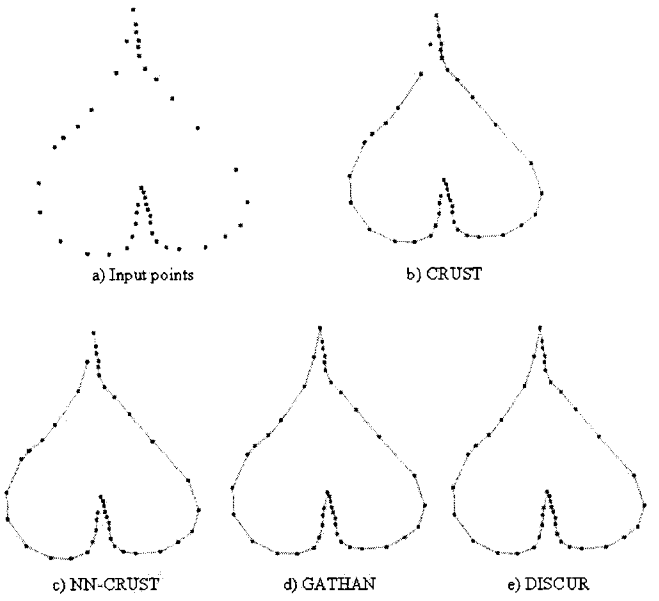


Figure 29: Reconstruction of sharp corner: change of sampling conditions.

3.4.3 Boundary and multiple components

In comparison with CRUST and NN-CRUST, DISCUR is able to detect the boundary points while NN-CRUST and CRUST wrongly connect them as shown in Figure 30.

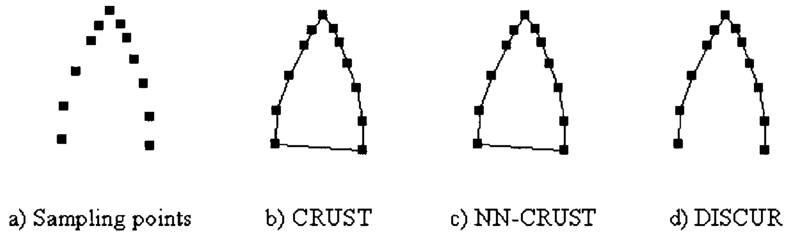
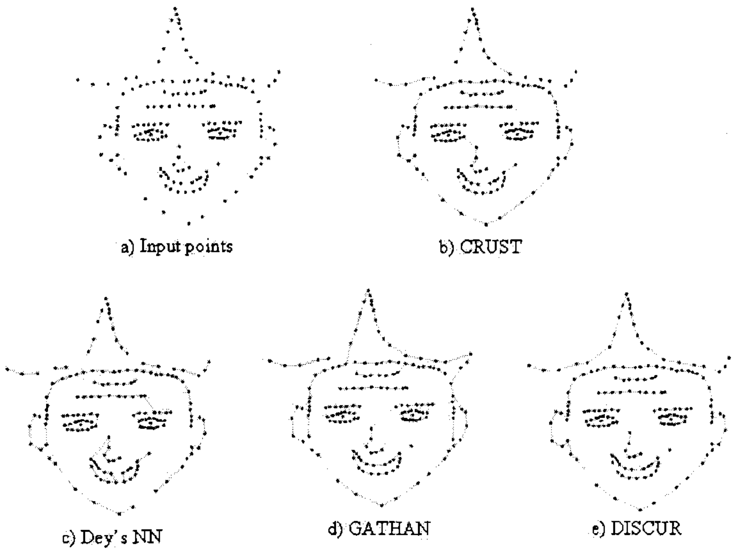


Figure 30: Reconstruction in the case of open curve.

3.4.4 Summary of comparison

Example in Figure 31 shows a more complex sample set, which includes multiple features such as uneven samplings, sharp corners, boundaries, and multiple components.



a) Input points

b) CRUST

c) Dey's NN

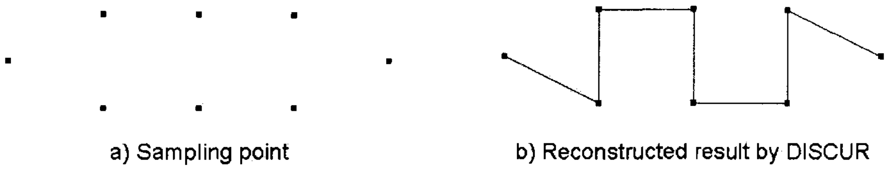
d) GATHAN

e) DISCUR

Figure 31: Reconstruction result from different algorithms.

3.5 Limitation

In developing Algorithm DISCUR, we have used only Gestalt law of proximity. The algorithm works correctly if sampling conditions in Theorem 3.3.1 are met. However, in some cases such as that given in Figure 32a), DISCUR reconstructs the curve as shown in Figure 32b), though Figure 33 is more visually acceptable. In order to correct the wrong connections by using DISCUR, more sampling points are needed to enforce the desired conditions. Alternatively, since the result in Figure 32b) obviously violates the smoothness property, the algorithm can be enhanced by adding a quantification of the smoothness observation based on angles between two edges to be connected. This research is addressed in VICUR algorithm.



a) Sampling point

b) Reconstructed result by DISCUR

Figure 32: An example of wrong connections.

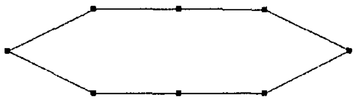


Figure 33: Desired result.

3.6 Summary

In this paper, a new algorithm is proposed to reconstruct multiple curves, which may be open, closed, and/or with sharp corners. This algorithm is parameter-free. The foundation of this algorithm originates from Gestalt law of nearness. To simulate this property, both the neighborhood features of a curve and the statistical properties of a set of samples are investigated. A general form of vision function $E[p, T_q]$ is proposed to determine the connectivity of a point to a curve segment. Then a concrete representation of $E[p, T_q]$ for the present algorithm DISCUR is given through observation.

Chapter 4

VICUR algorithm: simulation of nearness and smoothness property of human vision in the context of curve reconstruction

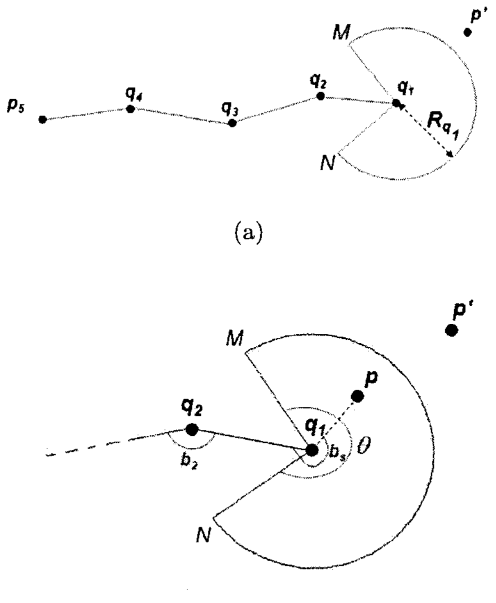
DISCUR algorithm satisfies the smoothness property in following the nearness property by enforcing a necessary and sufficient sampling condition. In relaxing the sampling condition near sharp corners and in solving the conflicts between nearness and smoothness properties, as was illustrated in Figure 32. In this chapter, we propose a new algorithm, VICUR.

4.1 Simulation of nearness and smoothness properties

4.1.1 Connectivity area

We observe that human eyes tend to connect a point to an existing curve when the point lies within a certain area determined by the characteristics of the curve. We name this area connectivity area, which is illustrated in Figure 34.

The connectivity area at an endpoint q_1, denoted as $A(q_1, R_{q_1})$, is a set of points having the probability of connection to q_1 greater than 0 and is defined as a sector of a circle centering at q_1 with center angle θ and radius $R_{q_1} = \beta \bar{d}$ where β is a parameter and $\bar{d}$ is the average distance of the α-smooth curve. All points within connectivity area are called candidate points. Points fall outside the bounded area are considered as outliers.



(a)

(b)

Figure 34: Connectivity area.

Figure 34 shows the connectivity area $A(q_1, R_{q_1})$ of a curve $T = [q_1, q_2, q_3, q_4, q_5]$, p is a candidate point, p' is an outlier with respect to q_1.

4.1.2 Connectivity function

When there are two or more than two sampling points in a connectivity area, all these samples are candidates to be considered for connection to the corresponding curve endpoint. We use the vision function $E[p, T_q] = f(p, V)$ to evaluate the possibility of the connectivity for each sample in the connectivity area. In this case, T_q is an α-smooth segment of the curve. We set $\alpha = 45°$ for all the experiments.

From observation and preliminary experiments [Li07][He08], we derived five factors which have the most impact on the construction process. These factors include candidate angle, length of candidate segment, average angle, average distance of the curve and standard deviation of the distance. Based on these elements, a concrete form of the function obtained through observation is given as follows

$$E[p, T_q] = (c(\frac{b_s}{\bar{b}} - 1)^2 + (1 - c)(\frac{d_s}{2(\bar{d} + \sigma)})^2 + 1)^{-1} \tag{8}$$

where b_s is the candidate angle, c is a user-defined parameter, $\bar{b}$ is the angle mean, d_s is the length of the candidate segment. We assume that if p has the highest value $E[p, T_q]$ among other candidate points, then p can be connected to q.

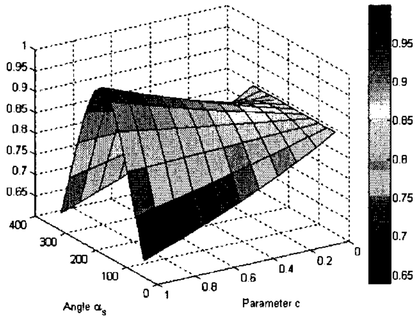


Figure 35: Relationship between candidate angle and parameter, $\bar{b} = 180°$ and $d_s = \bar{d}$

Figure 35 illustrates the relationship among candidate angle b_s, parameter c and the connectivity value. Given $d_s = \bar{d}$ and $\bar{b} = 180°$, for any parameter c, the connectivity value is the largest when the value of the candidate angle b_s reaches the value of the angle mean $\bar{b}$. The effect of the candidate angle b_s on the connectivity value increases, when parameter c approaches 1. When parameter c equal to 0, the connectivity value remains unchanged regardless of the value of the candidate angle because the connectivity value is determined by the candidate distance d_s only.

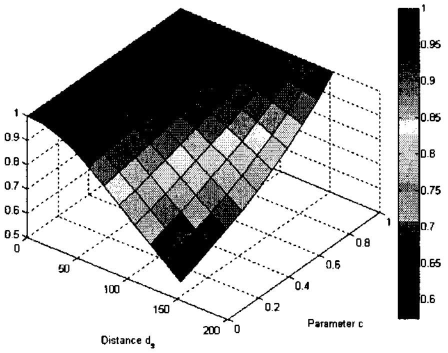


Figure 36: Relationship between candidate distance and parameter, $b_s = \bar{b} = 180°$.

Figure 36 shows the effect of the candidate distance d_s and the parameter c on the connectivity value. When parameter c approaches 1, the impact of candidate distance d_s on the connectivity value drops. When the candidate distance d_s approaches 0, which means that the candidate point is very near curve endpoint, the connectivity value rises substantially. When the parameter c approaches 0, the impact of candidate distance d_s on the connectivity value increases.

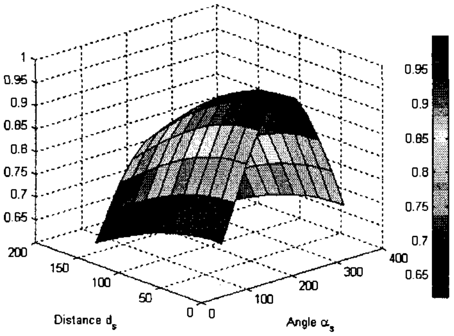


Figure 37: Relationship between candidate distance and candidate angle, $c = 0.8$.

In Figure 37, given the parameter $c = 0.8$, the largest connectivity value occurs when the candidate angle b_s reaches the angle mean $\bar{b}$ and the candidate distance d_s approaches 0. This means points that are very near to the constructed curve endpoint and also form the smoothest path with the constructed curve have the high possibility to connect to the curve. When the candidate angle b_s deviates from the angle mean $\bar{b}$, the connectivity value decreases. Similarly, when the candidate distance d_s is far from curve endpoint, the connectivity value decreases.

In summary, when connecting a point to a curve, two factors should be considered: distance from the point to the curve endpoint and the smoothness of the curve after the point is connected. If the nearest point to the curve endpoint also forms with the constructed curve the smoothest path, connection is easily determined. However, conflict arises when the nearest point does not form with the constructed curve a smooth path and when a point connects to the constructed curve resulting in the smoothest path is not the nearest point. To overcome this difficulty, a parameter c is introduced.

When parameter c approaches 0, the nearness property becomes more important than the smoothness property. In this case, the algorithm tries to connect the nearest neighbor rather than constructing a smooth curve. On the other hand, when parameter c approaches 1, the algorithm tries to maintain the smoothness of the curve.

4.1.3 Connectivity rules

Firstly, we continue to use Gestalt law of proximity, which indicates that human eyes tend to connect nearest points to form a curve. However, in some cases two closest neighbors are not necessarily two adjacent points on a curve. For example, in the case of sharp corner illustrated in Figure 38. Samples p_2 and p_3 are the nearest neighbors to each other but human eyes do not see them being adjacent on the curve. Therefore, an attempt to connect any two nearest free points may result in a wrong connection.

In this case, we observe that the shortest edge p_2p_3 forms with p_3p_5 (or p_3p_4) an angle smaller than $\angle(p1p3p5)$. Therefore, before connecting p_2 to p_3, it should be checked if there is an interior angle γ at p_3 (or p_2) formed by p_2p_3 and other incident edges such that γ is larger than other interior angle at p_3 (or p_2) formed by edges other than p_2p_3.

However, there is a case where the edge between two free points forms with other edge a largest angle but these two points should not be connected. The situation is illustrated in Figure 38 where $\angle(p_2p_3p_6)$ is the largest angle but p_2 should not be connected to p_3. This leads to another observation: there is a distance for which two points can be considered as 'a group'. Beyond this value, a point is seen as outlier or belongs to another group. Based on our tests, we set the value equal to the average distance of the shortest and second shortest Delaunay edges multiplied by a constant ϕ.

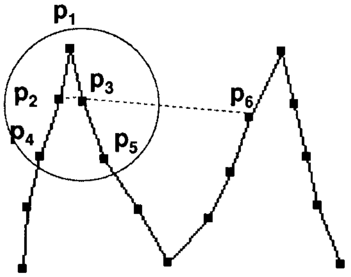


Figure 38: Connectivity rule for two free points.

Secondly, we apply Gestalt law of continuity in the algorithm. Law of continuity states that human eyes connect points into the smoothest path. This property of perception suggests that after a point is connected to a curve, it should not change the direction of the curve substantially. However, conflict between smoothness and nearness may arise as introduced in Chapter 2.3. To deal with the conflict, we add a weight to each property. In particular, in the connectivity function $E[p, T_q]$ introduced in Section 4.1.2, parameter c is a weight factor for nearness property whereas $(1-c)$ is a weight factor for smoothness property. The weight factor plays as a control to determine which property should be followed when conflict occurs.

We propose the following rules to determine the connection between two samples:

Rule 1: point-point connectivity. For any shortest edge $e = [q_1, q_2]$ where q_1 and q_2 are both free points. Let $B(q_1, r)$ be a ball centered at q_1 with radius $r = \frac{1}{2}\phi(q_1q_{k_1} + q_1q_{k_2})$ where $q_1q_{k_1}$ and $q_1q_{k_2}$ are the shortest and second shortest Delaunay edge to q_1. For all $q_i, q_j, q_t \in B$ and $q_i, q_j, q_t \neq q_2$, if $\angle(q_tq_1q_j) < \angle(q_iq_1q_2)$ where $\angle(q_tq_1q_j)$ and $\angle(q_iq_1q_2)$ are the interior angles at q_1, then q_1 and q_2 can be connected.

Rule 2: point-curve connectivity. For an α-smooth curve $T = [q_1, q_2, ..., q_m]$, $m > 1$, if there exists a sampling point $p \in A(q_1, R_{q_1})$ (or $A(q_m, R_{q_m})$) such that $E[p, T_{q_1}] > E[q_j, T_{q_1}]$ (or $E[p, T_{q_m}] > E[q_j, T_{q_m}]$) for all $q_j \in A(q_1, R_{q_1})$ (or $A(q_m, R_{q_m})$) and $q_j \neq p$, then p and q_1 (or q_m) can be connected.

Rule 3: curve-curve connectivity. For two α-smooth curves $T = [q_1, q_2, ..., q_m]$, $T' = [q'_1, q'_2, ..., q'_n]$,$n, m > 1$, if q_1 (or q_m) can connect to T' by the rule of point-curve connectivity and q'_1 (or q'_m) can connect to T by the rule of point-curve connectivity, then these two curves can be connected.

4.2 Algorithm

Algorithm VICUR contains two steps, as shown in Figure 39. Step one determines the connectivity for each Delaunay edge and step two updates the connectivity when necessary. The pseudocodes of the algorithm are given in Figure 40 and 41.

```
Algorithm VICUR(SampleSet: S)
Step 1 - Determining the connectivity of Delaunay edges.
Step 2 - Updating the connectivity of Delaunay edges.
```

Figure 39: Overview of VICUR algorithm.

```
Step 1 Determining the connectivity of Delaunay edges.
Compute the Delaunay triangulation of S
Let $D_e$ be a set of Delaunay edges
Let $W$ be a set of temporarily removed edges
for each Delaunay edge $e = [p_i, p_j] \in D_e$ do
  if $p_i$ and $p_j$ are both free points then
    Apply point-point connectivity rule
    if $p_i$ and $p_j$ cannot be connected then
      $R \leftarrow R \cup [p_i, p_j]$
    else
      Connect $p_i$ to $p_j$
    end if
  end if
  if $p_i$ is an endpoint of curve $T$ and $p_j$ is a free point then
    Consider only $\alpha$-smooth $T_{p_i}$ of curve $T$. Construct connectivity area $A(p_i, R_{p_i})$
    Let $Q = \{q : q \in A(p_i, R_{p_j}), p_i q \in (D_e \cup W)\}$
    if $(|Q| = 0)$ then
      $W \leftarrow W \cup [p_i, p_j]$
    end if
    if $(|Q| = 1)$ then
      Connect $p_i$ to $q$
    end if
    if $(|Q| > 1)$ then
      Compute the connectivity value $E(q, T_{p_i})$ for each $q \in Q$
      Choose the point $q$ having the largest corresponding connectivity value to connect to $p_i$
    end if
  end if
  if $p_i$ is an endpoint of curve $T$ and $p_j$ is an endpoint of curve $T'$ then
    Consider only the $\alpha$-smooth $T_{p_i}$ of curve $T$ and $T'_{p_j}$ of curve $T'$, respectively
    Construct connectivity area $A(p_i, R_{p_i}), A(p_j, R_{p_j})$
    Let $Q = \{q : q \in A(p_i, R_{p_i}), p_i q \in (D \cup R)\}$
    Let $Q' = \{q' : q' \in A(p_j, R_{p_j}), p_j q' \in (D \cup R)\}$
    if $(|Q| + |Q'| = 0)$ then
      $W \leftarrow W \cup [p_i, p_j]$
    end if
    if $(|Q| + |Q'| = 1)$ then
      Connect $p_i$ to $q$ (or connect $p_i$ to $q'$ )
    end if
    if $(|Q| + |Q'| > 1)$ then
      Compute connectivity value $E[q, T_{p_i}]$ for each $q \in Q$ and $E[q', T_{p_j}]$ for each $q' \in Q'$
      Choose the point with the largest connectivity value to connect to the corresponding
      endpoint
    end if
  end if
  $D_e \leftarrow D_e - \{e\}$
end for
```

Figure 40: Step 1 of VICUR.

The main idea of the algorithm above is that we first find the closest pair of sampling points. If two samples are free, apply rule 1 in Section 4.1.3. If one of the samples is not free or both of the samples are not free, construct connectivity area and apply rule 2 or rule 3. Currently, value β and ϕ are set at 1.849 based on observation.

```
Step 2 Updating the connectivity of Delaunay edges.
1: for each Delaunay edge $e = [p_i, p_j] \in M$ do
2:   $D_e \leftarrow D_e \cup \{e\}]$ $W \leftarrow W - \{e\}$
3: end for
4: for each Delaunay edge $e = [p_i, p_j] \in D_e$ do
5:   Apply line 5 to 43 in Step 1
6:   if ($p_i$ and $p_j$ are connected to form a new curve $T^1$) then
7:     repeat
8:       for each $e' \in (D_e \cup W)$ adjacent to $T^1$ do
9:         Apply line 5 to 43 in Step 1
10:      end for
11:    until ($T^1$ was not extended during line 8 to 10)
12:  end if
13: end for
```

Figure 41: Step 2 of VICUR.

4.3 Results and comparisons

4.3.1 Results

In this section, procedure of VICUR algorithm will be demonstrated. Figure 42 shows input sample and corresponding Delaunay triangulation. Figure 43 shows that three situations may occur in step 2 of the algorithm. The three situations are described as follows:

- Figure 43a) shows that p_1p_2 is the shortest Delaunay edge and both p_1 and p_2 are free points. The algorithm checks if connection between p_1 and p_2 may result in potentially wrong connection by drawing a ball $B(p_1, R_{p_1})$ as shown in Figure 43b) where $R_{p_1} = \frac{(p_1p_2+p_1p_5)}{2}\phi$. There are six samples in the ball $B : p_1, p_2, p_3, p_4, p_5, p_8$. We find that $\angle(p_4p_1p_5) > \angle(p_2p_1p_4)$ so p_1 and p_2 fails the checking test and is temporarily removed.

- Figure 43c) shows p_7p_8 is the shortest Delaunay edge where p_7 is an endpoint of $T_{p_7} = [p_7, p_6]$ and p_8 is endpoint of $T_{p_8} = [p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}]$. Construct connectivity area $A(p_7, R_{p_7})$ at p_7 for T_{p_7} and construct $A(p_8, R_{p_8})$ at p_8 for T'_{p_8} because $T'_{p_8} = [p_8, p_9, p_{10}]$ is an α-smooth segment of curve T_{p_8}. Figure 43d) shows $p_7 \in A(p_8, R_{p_8})$ and $p_8 \in A(p_7, R_{p_7})$. Thus, p_7 and p_8 is connected and p_7p_8 is removed from Delaunay set.

- p_5p_1 is the shortest Delaunay edge, p_1 is an endpoint of $T_{p_1} = [p_1, p_4, p_3, p_2]$ and p_5 is free point. We have $[p_1, p_4]$ which is an edge of the α-smooth curve $T_{p_1} = [p_1, p_4, p_3, p_2]$. Construct the connectivity area $A(pl, R_{p_1})$ as shown in Figure 43f). Vertex p_2 is the only sample in $A(p_1, R_{p_1})$ available for connection so p_1 and p_2 is connected and removed from Delaunay set.

Step 2 repeats until Delaunay set becomes empty.

Figure 44 shows step 3 of the algorithm. In step 3, all the temporarily removed edges in step 2 are reconsidered because during construction process, the curve may be updated which makes connectivity value and connectivity area changed. Figure 44 shows that all the temporarily removed edges are converted to Delaunay edges for reconsideration. Repeat step 2 on new Delaunay edge set. The reconstructed result is depicted in Figure 45.

To highlight the effectiveness of VICUR, Figure 46 shows an intuitive sample set including multiple single open and closed curves. Reconstructed curves from VICUR algorithm conforms to human vision. Also, constructed results from other algorithms are presented in Figure 47.

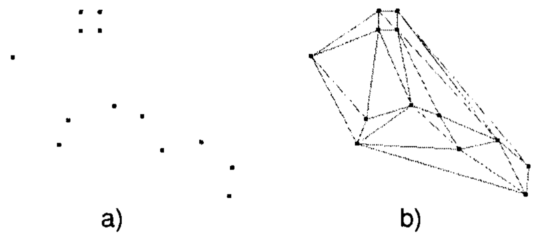


Figure 42: Input sample and Delaunay triangulation.

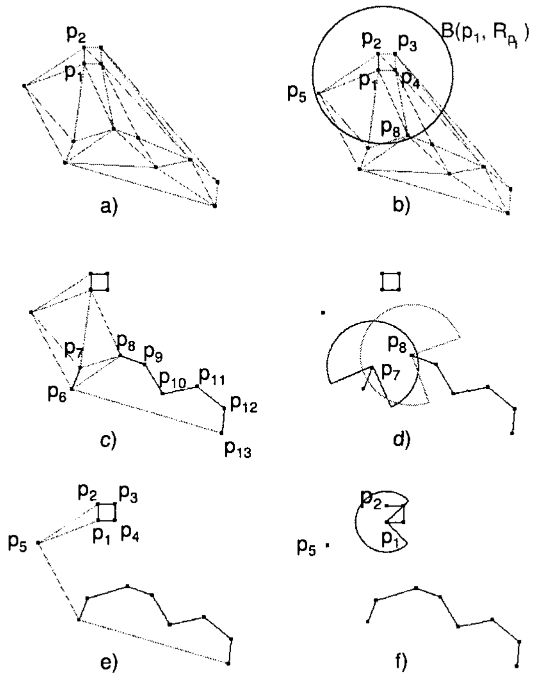


Figure 43: Construction in step two of VICUR algorithm, $\phi = \beta = 1.849$.

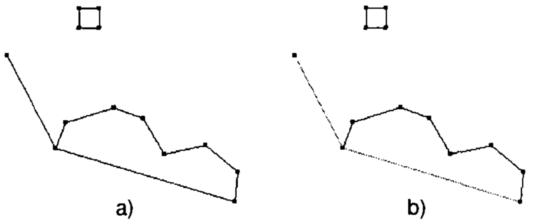


Figure 44: Step three of VICUR algorithm.

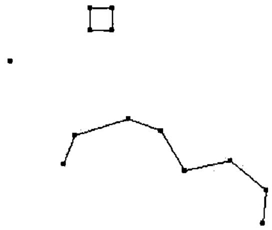
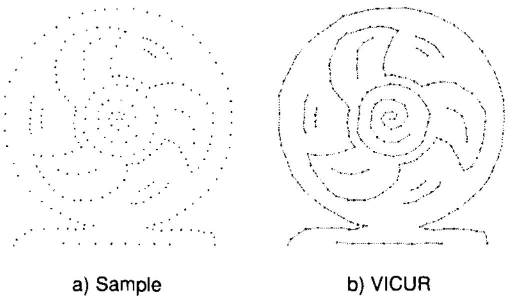


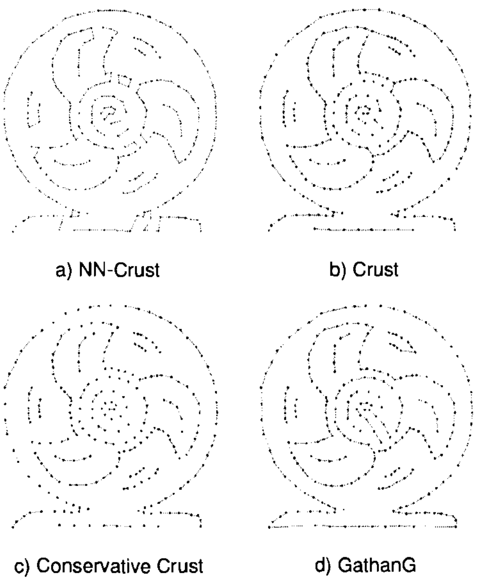
Figure 45: Final result.



a) Sample

b) VICUR

Figure 46: Reconstructed curves by VICUR.



a) NN-Crust

b) Crust

c) Conservative Crust

d) GathanG

Figure 47: Reconstructed curves by other algorithms.

4.3.2 Sampling condition

Most of the current algorithms use medial axis or local feature size to determine sampling condition. As a result, the sample sets for those algorithms are not intuitive. In contrast, VICUR algorithm aims to work for intuitive sampling condition, which implies that the samples are connected to form curves that look natural to human eyes. Intuitive sampling condition helps users to have better control over the sample set. Figure 48 shows an example where connections resulted from VICUR agrees with human perception.

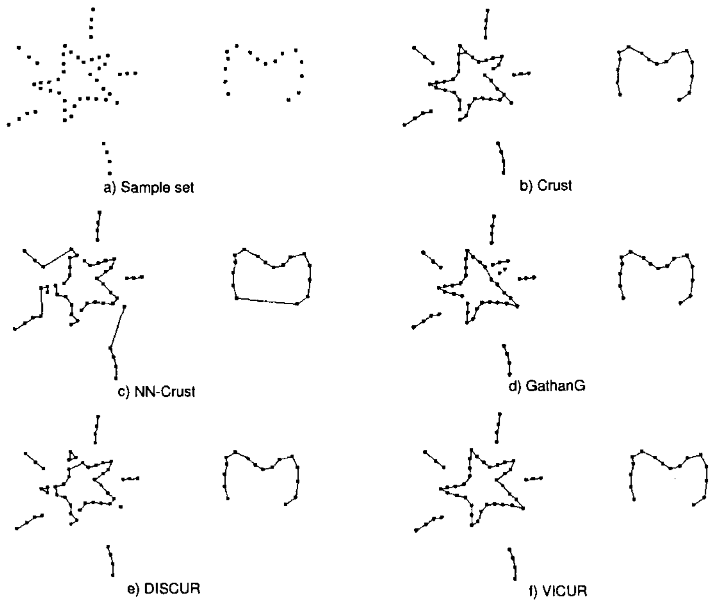


Figure 48: Result from VICUR is consistent with human visual system.

4.3.3 Boundary and sharp corner

Most of the current algorithms can construct closed smooth curves correctly. To reconstruct non-smooth curve, Giesen developed TSP but the algorithm can construct only single closed curves [Gie99]. Dey-Wenger introduced another algorithm named GATHAN which can detect corner point and endpoint well in practice but with no guarantees [DW01]. Later, they proposed GATHANG algorithm based on GATHAN [DW02]. GATHANG guarantees correct construction on closed curves but not on open curves. Figure 49 shows a situation where GATHANG fails.

The DISCUR algorithm presented in [ZNYL08] can also handle sharp corners but the sampling is very dense near the corner compared to our new algorithm as shown in Figure 50. Our algorithm correctly construct curve with the sample given in Figure 50a) while DISCUR needs a denser sampling, Figure 50d).

Funke and Ramos also proposed another algorithm that can construct open non-smooth curves with guarantees [FR01]. However, due to limited resources we did not do any experiment with their algorithm. The only comparison we did is the MPI data set taken from the article [FR01] as shown in Figure 51.

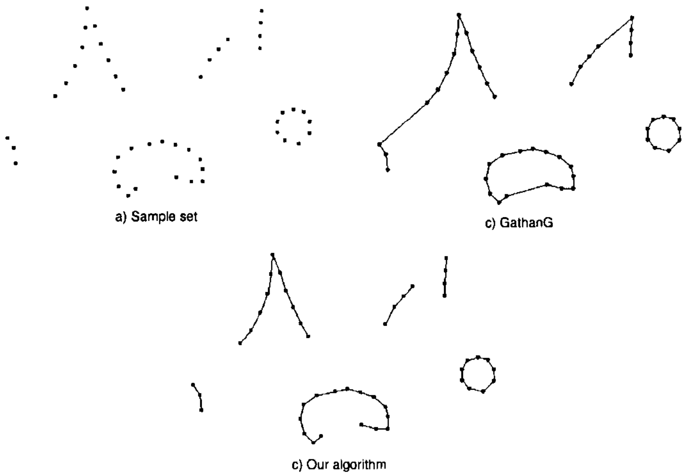


Figure 49: GATHAN fails to construct open curves.

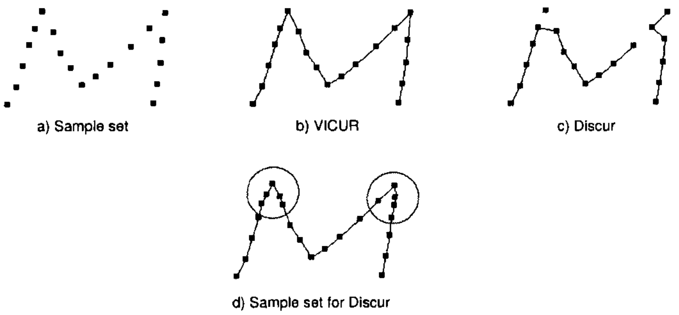
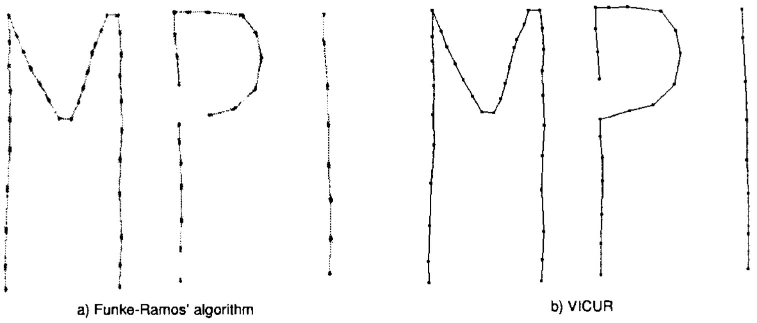


Figure 50: DISCUR requires dense sampling around corner point.

4.4 Limitation

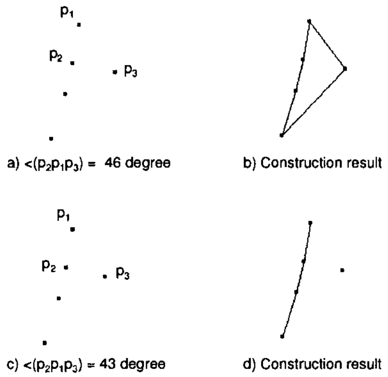
Despite the fact that VICUR can handle well many examples, we are aware of the limit of our algorithm. Firstly, VICUR is sensitive to vertex position. Figure 52 illustrates such a situation where human eyes hardly realize a difference between 46° and 43°. Construction result of Figure 52a) and Figure 52c) should be similar. The algorithm, on the other hand, detects a significant difference. If θ is set at 270°, 46° is considered within connectivity area boundary where 43° is out of the range. As a result, sample p_3 cannot be connected to p_1, causing construction result Figure 52b) and Figure 52d) to be different.

Secondly, although checking between two free points prior to connection helps avoid wrong connection in case of sharp corner, sometimes it may create a problem as shown in Figure 53b). In this case, VICUR detects $\angle(p_1p_2p_3)$ as a sharp corner with p_2 as a


a) Funke-Ramos' algorithm

b) VICUR

Figure 51: Result of MPI data set from Funke-Ramos's article.



a) <(p2p1p3) = 46 degree

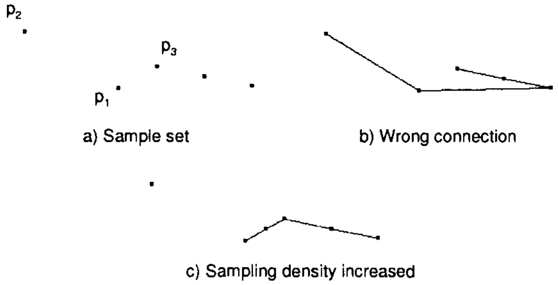
b) Construction result

c) <(p2p1p3) = 43 degree

d) Construction result

Figure 52: VICUR is sensitive to vertex position.

corner point. Consequently, the algorithm does not connect p_1, p_3 and the construction result becomes unnatural to human vision. However, this problem can be fixed easily by increasing the sampling density, as is shown in Figure 53.



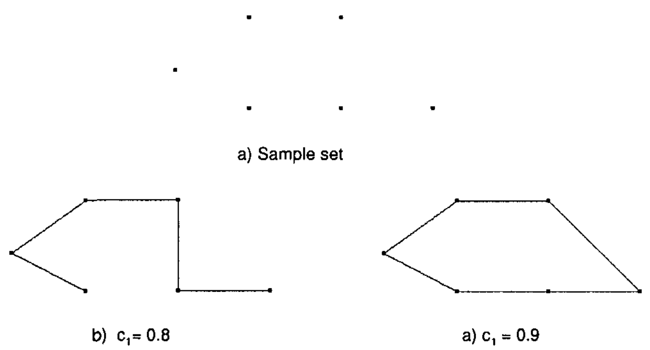
a) Sample set

b) Wrong connection

c) Sampling density increased

Figure 53: Testing for potentially wrong connection results in wrong construction.

Additionally, the parameter c needs to be adjusted to produce desired result. An example is illustrated in Figure 54.



a) Sample set

b) c_1= 0.8

a) c_1 = 0.9

Figure 54: Different parameters yields different results.

4.5 Summary

We proposed a new algorithm for curve reconstruction named VICUR. Foundation of VICUR algorithm is established from two laws of Gestalt theory of perceptual organization: law of proximity and law of continuity. VICUR can construct open, non-smooth curve and the result is agreeable with human perception. The algorithm is developed based on data obtained from observation. Motivation for our algorithm is not only to provide a new approach to curve reconstruction problem but also to attempt to quantify some properties of human visual perception.

Chapter 5

A new graphical password scheme

Most of user-drawn based passwords in the literature require users not only to memorize the drawing but also the information about how the drawing is created (i.e. exact starting cell, ending cell, pen-up event). Studies show that users can remember a picture more easily than they remember the process how the picture is created [TvO04]. Thus, including the drawing process in the password may increase the password space but also decrease the usability of the scheme. Motivated by the curve reconstruction algorithm introduced in Chapter 3 and Chapter 4, a new kind of graphical password scheme is proposed to solve the aforementioned problem in this chapter. This proposed password scheme is based on the hypothesis that a user would create a password that is natural to his or her vision.

5.1 Introduction to password design

In this section, we introduce a new user-drawn-based graphical password scheme that does not require users to remember the order of the stroke. In this scheme, users are required to create a drawing on a given set of points by selecting individual points or by connecting any two points.

To facilitate descriptions in the next sections, we introduce some terminology as follows:

- n - the total number of given points, $n > 0$
- v_i - the number of points that are chosen as password points, $i \geq 0$. These points

are also called isolated points or vertices because they are parts of the password but do not connect to any edge.

- e - the number of edges that form the password.
- L - length of a password and is defined as the sum of the number of edges and isolated points in a password.
- L_{max} - the maximum password's length beyond which the possibility of the password being created is zero.

5.2 A new graphical password scheme

The password authentication system is divided into two parts: 1) the enrollment process or registration process in which user creates his or her secret and 2) the authentication process in which user authenticates his or her identity to the system. In the next section, we will give further details in each process. In particular, we will discuss how the point set is generated, how a password is created, and how a password is encoded.

5.2.1 Point cloud generation

In our password scheme, the point set is predefined to the users. Thus, in the enrollment process, users can either choose a point set from a set of collection point file or the system contains only one point file.

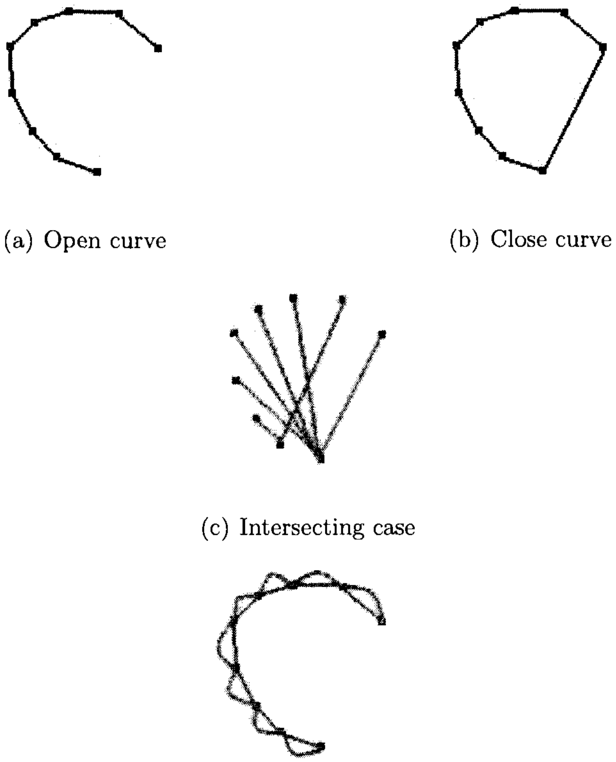
The point set needs to have sufficient large number of points so that it can yield a large number of possible combination, which is not easily exhausted by attackers. A sufficient large number of points also reduce the possibility of guessing the right password.

Another issue is that the points have to be organized in a way that it includes certain patterns appearing meaningful to human vision so that users can choose passwords that are easy to memorize. The advantage of our password scheme is that by organizing the points in such ways, the scheme actually helps users to recall the password. In DAS or Pass-Go scheme, the canvas is a grid which can be viewed as a set of points where the distance between two points is uniform whereas in our case the distance between two points can be non-uniform. The uniformity in DAS and Pass-Go does not assist users in recalling the shape of their password. Our scheme, on the other hand, with the points positioned in a way that are natural to human vision, may help users to recall the drawings of their password.

5.2.2 Password generation rules

The password is created by connecting any two points in a given point set to form a curve and any two points can be connected more than one time. A password can be a curve or multiple curves- open or closed curves as shown in Figure 55(a) and Figure 55(b), or a password can contain intersecting curves as shown in Figure 55(c), or can have multiple edges between two points as shown in Figure 55(d). Individual point can also be chosen as part of a password. This variety allows a greater password space than DAS which only allow connection between neighboring cells and Pass-Go which allow connection between two nearest points. Although it appears that users have plenty of options to generate password, the scheme does impose three restrictions in the password creation. The first restriction is that points that are on a curve cannot be selected as a password component. The second restriction is that an edge cannot connect a vertex to itself. Finally, the third restriction is that an isolated point selected as a part of a password cannot be selected again. Figure 56 shows the two invalid cases. The rules for password generation are summarized as follows:

- Any two points can be connected to form an edge.
- Any two points can have more than one edge.
- Loop on a single point is not allowed.
- Points that are adjacent to an edge can not be chosen as a password component.
- A point cannot be chosen more than once as an isolated password point.



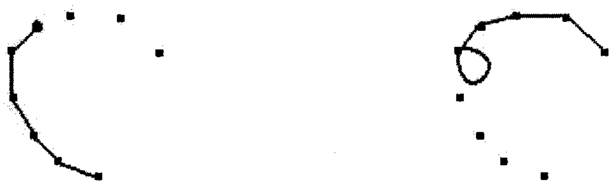
(a) Open curve

(b) Close curve

(c) Intersecting case

(d) Multiple edges

Figure 55: Examples of graphical passwords.



(a) Point selection

(b) Curve with loop

Figure 56: Invalid cases.

5.2.3 Password encoding

In our scheme, each point in the point file is identified by its coordinate (x, y). The whole password will be encoded as a multiset which consists of several subsets. Each subset contains one coordinates (x, y) of the point if the password is an isolated point; if the password is a curve, each subset contains the coordinates (x, y) of the endpoints of the edge. For instance, the password in Figure 57 is presented as:

$\{\{(304, 205), (447, 201)\}, \{(313, 159), (428, 159)\}\}$ where $A = (304, 205)$, $B = (447, 201)$, $C = (313, 159)$, $D = (428, 159)$.

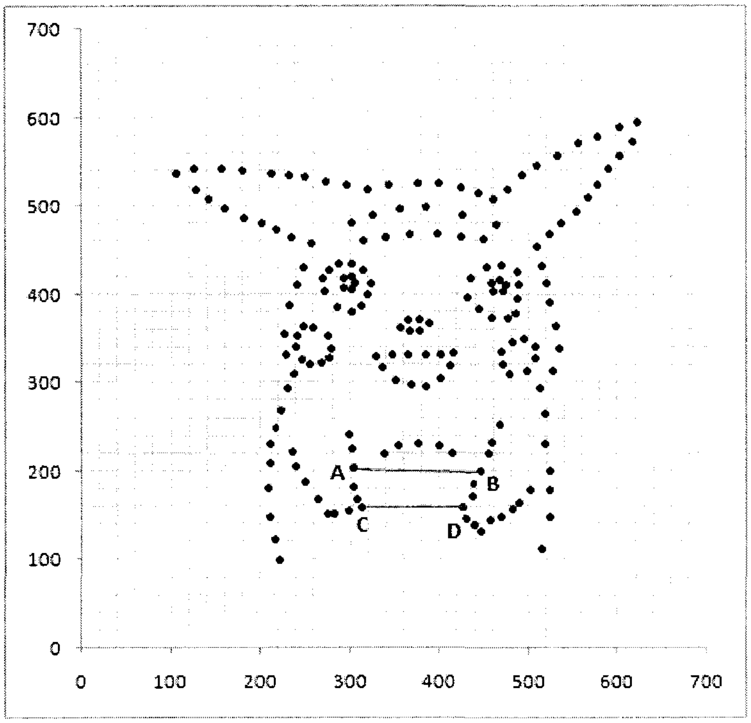


Figure 57: Password encoding.

5.3 Password space: Evaluation of the proposed password scheme

5.3.1 Full password space

The password space is computed as follows:

$$\text{Password space} = log_2(\Gamma_{n,L_{max}}) \tag{9}$$

where n is the a number of points in the point set, $\Gamma_{n,L_{max}}$ is the total number of possible drawings consisting of $1, 2, ...$ up to L_{max} password components whose vertices are derived from the n points. A password component can be an edge or an isolated points. $\Gamma_{n,L_{max}}$ is computed as:

$$\Gamma_{n,L_{max}} = N_{n,1} + N_{n,2} + ... + N_{n,L_{max}} = \sum_{L=1}^{L_{max}} N_{n,L} \tag{10}$$

$N_{n,L}$ is the total number of graphs of e edges with v_i isolated vertices, $e + v_i = L$.

$$N_{n,L} = \begin{cases} \sum_{v_i=0}^{L-1} G(n, v_i, e) & \text{if } 0 \leq v_i < L \\ \binom{n}{L} & \text{if } v_i = L \end{cases} \tag{11}$$

When $v_i = L$, all L components are isolated points, then $N_{n,L}$ is found by simply selecting L points from n points. The number of ways to choose such L components is $\binom{n}{L}$. When $v_i < L$, the L components contain e edges and v_i isolated points ($L = e + v_i$). In this case, $N_{n,L}$ is the total number of graphs consisting of (1 edges, $L - 1$ isolated vertices), (2 edges, $L - 2$ isolated vertices), ..., and (L edges, 0 isolated vertices). The graph contains e edges and v_i isolated vertices, whose vertices are selected from n points denoted by $G(n, e, v_i)$. There are $\binom{n}{v_e+v_i}$ such selections where v_e is the number of vertices adjacent to edges; the sum $v_e + v_i$ is the total number of vertices of the graph.

Let v_e denoted the number of vertices on e edges. We notice that for a graph containing e edges and v_i isolated vertices, the value v_e can be ranged from $\bar{v} = \lceil \frac{1+\sqrt{1+8e}}{2} \rceil$ to $2e$.

Therefore:

$$G(n,e,v_i) = \sum_{v_e=\bar{v}}^{2e} \left[\binom{n}{v_e+v_i} \binom{v_i+v_e}{v_e} g(e,v_e) \right] \tag{12}$$

The number of graphs consisting of e edges and v_i isolated vertices is equal to the number of graphs consisting of e edges, v_e vertices without isolated vertices denoted by $g(e, v_e)$. In addition, we need to count the number of ways to select v_e from $v_e + v_i$. We have $\binom{v_i+v_e}{v_e}$ ways.

We observe that $g(e, v_e)$ contains simple graphs and graphs with multiple edges. Hence:

$$g(e,v_e) = g_s(e,v_e) + g_m(e,v_e) \tag{13}$$

where $g_s(e, v_e)$ is the number of simple graphs consisting of e edges and v_e vertices, $g_m(e, v_e)$ is the number of graphs with multiple edges consisting of e edges and v_e vertices. The function $g_s(e, v_e)$ is calculated as follows:

$$g_s(e,v_e) = \begin{cases} \frac{1}{e}\left[\binom{v_e}{2} - e + 1\right] g_s(e-1,v_e) + \frac{1}{e}v_e(v_e-1)g_s(e-1,v_e-1) \\ \qquad +\frac{1}{e}\binom{v_e}{2}g_s(e-1,v_e-2) \text{ if } ((\lceil\frac{1+\sqrt{1+8e}}{2}\rceil \leq v_e < 2e) \\ \\ \prod_{i=0}^{v_e/2-1} (v_e-1-2i) \text{ if } (v_e = 2e) \end{cases} \tag{14}$$

$g_s(e, v_e) = 0$ if $\lceil\frac{1+\sqrt{1+8e}}{2}\rceil > v_e$ or $v_e > 2e$. A graph with multiple edges can be considered as a simple graph with extra edges added on existing edges. $g_m(e, v_e)$ having e edges can be computed by finding the number of simple graphs having m edges ($m < e$) and then add the remaining $(e - m)$ edges to the existing m edges to form multiple edges. Adding remaining $(e-m)$ edges to existing m edges is similar to choosing $(e-m)$ "places" (repetition is allowed) from the m-edge simple graph. Thus, there are $\binom{m+(e-m)-1}{e-m} = \binom{e-1}{e-m}$ ways to pick. Hence:

$$g_m(e,v_e) = \sum_{m=1}^{e-1} \left[\binom{e-1}{e-m} g_s(m,v_m) \right] \tag{15}$$

Table 3: Full password space at length L_{max} or less, given a set of n points to choose from.

	Password length L_{max}								
	1	2	3	4	5	6	7	8	9
n=100	12.3	23.6	34.32	44.62	54.6	64.31	73.8	83.1	92.22
n=75	11.48	21.95	31.84	41.31	50.46	59.35	68.01	76.48	84.77
n=50	10.32	19.63	28.35	36.66	44.64	52.36	59.85	67.15	74.28
n=25	8.34	15.68	22.41	28.72	34.7	40.42	45.9	51.19	56.31

Table 4: Comparison of password space between textual password, DAS-5 × 5 grid scheme and proposed password scheme.

	Password length L								
	1	2	3	4	5	6	7	8	9
95-ASCII characters	6.57	13.15	19.72	26.29	32.86	39.43	46.00	52.57	59.14
DAS-5	5	10	14	19	24	29	33	38	43
New scheme (n=25)	8.34	15.68	22.41	28.72	34.7	40.42	45.9	51.19	56.31
New scheme (n=50)	10.32	19.63	28.35	36.66	44.64	52.36	59.85	67.15	74.28

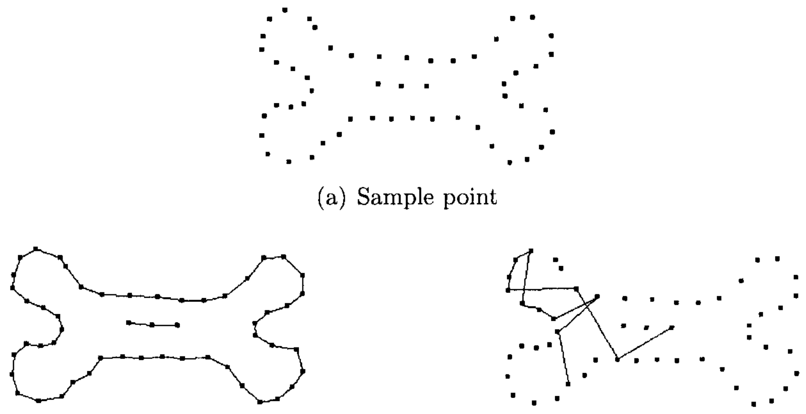
Table 3 shows the full password space computed as $log_2(number\ of\ passwords)$ of the passwords having length less than or equal to L_{max}.

Table 4 shows the comparison of password space between 95-printable ascii character textual passwords, 5 × 5 DAS scheme and the proposed password scheme ($n = 50$). We notice that at length 7, the number of the proposed passwords are already larger than the number of nine-character text passwords. However, this number only represent an ideal case where users' passwords are distributed uniformly. In reality, users' passwords may cluster into much smaller space, which can be exhausted easily by attackers.

5.3.2 Memorable password space and vision-based curve reconstruction algorithm

In practice, the full password space does not reflect the strength of a password scheme as real world users' passwords may belong to a smaller subset of the full password space. This password subset may be small enough to be exhausted by attacker's available resources. In the literature, researchers refer to this subset of full password space as memorable password space. In an attempt to compute the memorable password space and based on the assumption that users will choose a secret that is easy to remember, it is reasonable to deduce that in our proposed password scheme users will connect the given dots into a drawing that looks natural to their vision so that they can easily memorize their secret. Figure 58(b) shows a natural connection of the set of points given in Figure 58(a). By simply looking at the point in Figure 58(a), users can visualize the drawing in Figure 58(b). In contrast, the drawing in Figure 58(c) is not intuitive to human vision, as a result, it is expected that it takes more effort for human to memorize the drawing. Therefore, when users construct a secret, we expect that in general, they will try to connect points into a pattern that looks meaningful so that they can recall it easily.

The memorable password space, thus, will include the drawings that look intuitive to human vision.



(a) Sample point

(b) Connection that is natural to human vision

(c) Connection that is not natural to human vision

Figure 58: Natural and unnatural drawings.

To crack this password scheme, instead of exhausting all possible connections, an effective approach would be to examine the patterns that look natural or intuitive to human vision before trying other connections. As a result, the attacker may use vision based curve reconstruction to construct all possible patterns or curves. From psychology studies, the number of components that a person can memorize ranges from five to nine in which seven is the most common number of components that a person can memorize. Based on this result, we use seven as the maximum number of components comprising a password. A component can be an edge or an isolated point. Thus, a two-component password can consist of two edges, or one edge and one isolated vertex, or simply two isolated vertices.

We analyze the number of memorable drawing on n points where the points are organized in a way that is visualized as a closed curve. The number of memorable drawing of L components is equal to the number of ways to construct a graph forming from e edges and v_i isolated points where e edges belongs to the construction result of the vision based curve reconstruction algorithm.

Let ψ_a be the number of passwords consisting of passwords having seven components and the components are parts of the curve reconstruction result. If the passwords contain edge components, these edges are either distinctly separate or consecutive. For example, seven components consists of three edges and four points then either all three edges are adjacent or none of them are adjacent. The memorable password space ς_a is calculated as the logarithm base two of ψ_a. We compute the ς_a as follows:

$$\varsigma_a = log_2(\psi_a) \tag{16}$$

$$\psi_a = \binom{n}{L} + \sum_{e=1}^{L}\left[n\binom{n-e-1}{L-e}\right] + \sum_{e=1}^{L}\left\{\left[\binom{n-e}{e} + \binom{n-e-1}{e-1}\right]\binom{n-2e}{L-e}\right\} \tag{17}$$

Equation 17 is derived following the process below: We analyze a point cloud containing n points sampled from a simple closed curve. Hence, there are n points and n edges to select as password components. We calculate the memorable password space in three cases:

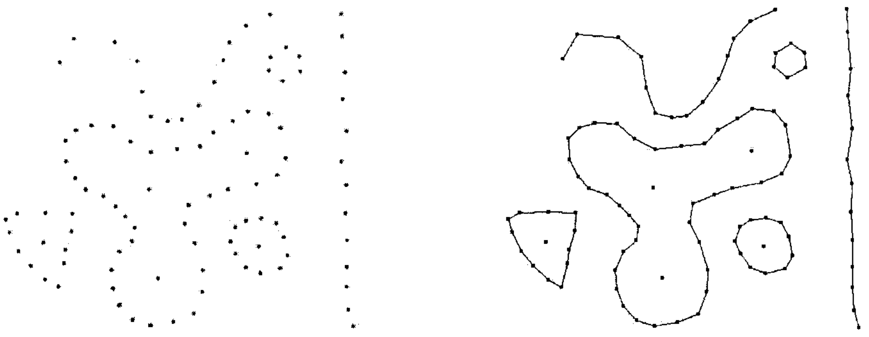
1. First, we consider the case where all the components are isolated points; in this case $e = 0$. There are such $\binom{n}{L}$ ways for such case.

2. Second, we consider the case where all the edges of the passwords are consecutive. All these edges form a single curve. To find the number of ways of choosing this curve, we need to count only the number of ways of choosing first edge of the curve. There are n ways to choose such an edge from n edges. The remaining password components are isolated points $v_i = L - e$. Because all the e edge components are adjacent, the number of points incident to the edges is $e + 1$ and the number of points left is $n - e - 1$. The number of ways to choose v_i points is $\binom{n-e-1}{L-e}$. The edge components can be $1, 2, ...$ up to L. The sum shows the total number of all possible number of edge component in this case.

Table 5: Comparison of memorable password space between DAS-5 × 5 grid scheme and proposed password scheme where password length $L_{max} = 7$.

		New password scheme		
	DAS-5	n=50	n=75	n=100
Memorable space	33.6	33.1	37.6	40.7

3. Finally, we consider the case where no edges are adjacent. each edge occupies two places (itself and the position to its right). Thus, in choosing e edges, e places are ignored. There are $n-e$ left to choose e edges from and there are $\binom{n-e}{e}$ such choices. Moreover, since the curve is closed, we need to add $\binom{n-e-1}{e-1}$ [Mar01]. The number of points left to choose $v_i = L-e$ points is $n-2e$ because each edge has two endpoints and no edge is adjacent to each other so the number of points incident to the edges is $2e$. This gives $\binom{n-2e}{L-e}$ ways to choose v_i points. The edge components can be $1, 2, ...$ up to L. The sum shows the total number of all possible number of edge component in this case.


(a) Sample point n=100

(b) Reconstruction result, n=100

Figure 59: Reconstruction result of a vision-based curve reconstruction algorithm.

Based on the above equation, we have Table 5 comparing our memorable password space with 5 × 5 DAS grid scheme. When $n = 75$, the memorable password space of the proposed scheme is much larger than the DAS. To make any conclusion about the effectiveness of the scheme, more analyses need to done to determine what value of n should be chosen so that the scheme can yield a sufficiently large memorable password space and still remains user-friendly.

5.4 Summary and discussions

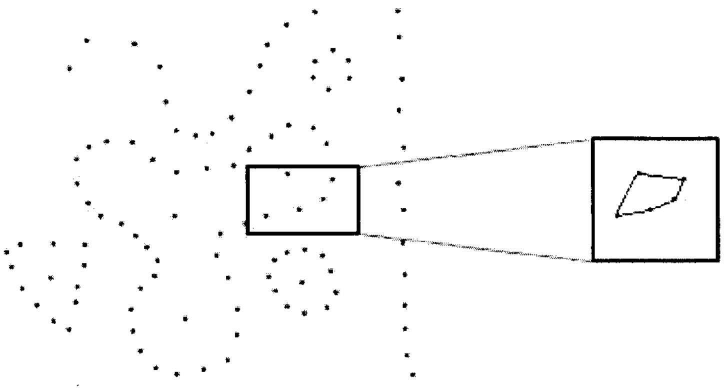


Figure 60: Choosing a subset of points from a point set.

In this chapter, we explore a new user-drawn based graphical password design which allows users to choose any point or connect any two points from a given point cloud to create a password. As mentioned in Chapter 2, there are three main methods that an attacker can use to crack the proposed password scheme: 1) try all possible connections, 2) choose, from the whole point set, only connections that are natural to human vision, and 3) choose, from a part of the point set, only connections that are natural to human vision. So far, the analysis of the memorable password space is based on Approach number 2). Approach number 3) can be done by taking a subset of points, as is shown in Figure 60, from the point set and applying the human-vision based curve reconstruction algorithm. There are several issues implied in this approach such as how to extract the points from the point cloud or which part of the point cloud should be taken. This problem can be address in the future research work.

Chapter 6

Conclusions and future research directions

6.1 Conclusions

In summary, our main contributions are: 1) a mathematical proof for the necessary and sufficient sampling condition for the distance based curve reconstruction algorithm DISCUR, 2) a new vision based curve reconstruction algorithm, VICUR, which considers both nearness and smoothness properties of human visual perception, and 3) a new graphical password scheme motivated by vision based curve reconstruction algorithms.

Firstly, the necessary and sufficient sampling condition for the parameter-free curve reconstruction algorithm DISCUR is introduced with two theorems. The first theorem determines the sampling for the interior points whereas the second theorem determines the sampling for the boundary points. The sufficient sampling condition implies that DISCUR guarantees the correct reconstruction result when the point cloud satisfies the sampling condition, while the necessary sampling condition implies that when DISCUR can construct the correct result from a point cloud, the point cloud certainly satisfies the sampling condition.

Secondly, VICUR is presented to tackle the limitation inherent in DISCUR. VICUR uses both Gestalt law of proximity and law of continuity as criteria to construct curves. In VICUR, a concept of $\alpha-$smooth curve is introduced to determine the smoothness of a curve and the concept of connectivity area is introduced to determine the boundary of the proximity. Furthermore, as apposed to DISCUR which is parameter-free, VICUR algorithm contains parameters to control the impact of nearness and smoothness properties during reconstruction process. Under many circumstances, VICUR can produce the correct reconstruction result. however, most of the factors in VICUR algorithm were merely derived from observations; thus, VICUR does not guarantee correct reconstruction.

The third contribution is to propose a new user-drawn based graphical password scheme. In this scheme, user creates the password by connecting points from an unorganized given point set provided by the authentication system. User authenticates to the system by recreating the drawing. This scheme does not require users to memorize how the password was created. Instead, users need to memorize only the final drawing. The analysis was conducted by applying vision based curve reconstruction algorithm to evaluate the memorable password space of the proposed password scheme. The result shows that when the number of points $n = 75$ the proposed scheme is larger than the 5×5 DAS scheme, given that the maximum length of a password is seven.

6.2 Future research directions

Vision-based curve reconstruction algorithm can be viewed as our preliminary attempt in quantifying Gestalt properties of human visual perception. Future research will continue to study the mechanism of human visual processes in the context of curve reconstruction. Moreover, we will consider extending our vision based curve reconstruction algorithm from 2D to 3D.

Regarding the proposed graphical password, a comprehensive security analysis should be done and several issues can be taken into account such as: what the optimal number of points should be in a point set and how the points should be distributed. As too many points will slow down the process of user authentication, which will diminish the usability of the system, the number of points should be chosen so that the system remains to be user-friendly and, at the same time, maintains its security level. In addition, the points should be arranged in such a way that does not provide attacker with knowledge about the distribution of users' passwords. Finally, some problems related to implementation should be studied (e.g. How the point set will be stored in the system).

Bibliography

[ABE98] Nina Amenta, Marshall Bern, and David Eppstein. The crust and the beta-skeleton: Combinatorial curve reconstruction. *Graphical models and image processing: GMIP*, 60(2):125–135, 1998.

[AM00] Ernst Althaus and Kurt Mehlhorn. Tsp-based curve reconstruction in polynomial time. In *Proc. ACM-SIAM Sympos. Discrete Algorithms*, pages 686–695. ACM Press, 2000.

[AMNS00] Ernst Althaus, Kurt Mehlhorn, Stefan Naher, and Stefan Schirra. Experiments on curve reconstruction. In *In Proc. 2nd Workshop Algorithm Eng. Exper*, pages 103–114, 2000.

[Att98] Dominique Attali. r-regular shape reconstruction from unorganized points. *Comput.Geom.Theory Appl.*, 10(4):239–247, 1998.

[CAS06] Konstantinos Chalkias, Anastasios Alexiadis, and George Stephanides. A multi-grid graphical password scheme. In *6th International Conference on Artificial Intelligence and Digital Communications*, Thessaloniki, Greece, 2006.

[Dey07] Tamal Krishna Dey. *Curve and surface reconstruction : algorithms with mathematical analysis.* Cambridge University Press, Cambridge ; New York, 2007.

[dFdMG94] Luiz Henrique de Figueiredo and Jonas de Miranda Gomes. Computational morphology of curves. *The Visual Computer*, 11(2):105–112, 1994.

[DK99] Tamal Krishna Dey and Piyush Kumar. A simple provable algorithm for curve reconstruction. In *In Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 893–894, 1999.

[DMR99] Tamal K. Dey, Kurt Mehlhorn, and Edgar A. Ramos. Curve reconstruction: Connecting dots with good reason. *In Proc.15th Annu.ACM Sympos.Comput.Geom*, 15:229–244, 1999.

[DNO08] Paul Dunphy, James Nicholson, and Patrick Olivier. Securing passfaces for description. In *SOUPS '08: Proceedings of the 4th symposium on Usable privacy and security*, pages 24–35, New York, NY, USA, 2008. ACM.

[DW01] Tamal Krishna Dey and RephaelKrishna Wenger. Reconstructing curves with sharp corners. *Comput.Geom.Theory & Appl*, 19:89–99, 2001.

[DW02] Tamal Krishna Dey and Rephael Wenger. Fast reconstruction of curves with sharp corners. *Int.J.Comput.Geometry Appl.*, 12(5):353–400, 2002.

[DY07] Paul Dunphy and Jeff Yan. Do background images improve draw a secret graphical passwords? In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 36–47, New York, NY, USA, 2007. ACM.

[EKS83] Herbert Edelsbrunner, David Galer Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *Information Theory, IEEE Transactions on*, 29(4):551–559, 1983.

[FKU77] Henry Fuchs, Zvi Meir Kedem, and Samuel Parker Uselton. Optimal surface reconstruction from planar contours. *Commun.ACM*, 20(10):693–702, 1977.

[FR01] Stefan Funke and Edgar Arturo Ramos. Reconstructing a collection of curves with corners and endpoints. In *Proc. 12th Annu. ACM-SIAM Sympos. Discrete Alg*, pages 344–353, 2001.

[Gie99] Joachim Giesen. Curve reconstruction, the traveling salesman problem and menger's theorem on length. In *SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*, pages 207–216, New York, NY, USA, 1999. ACM.

[Gol99] Christopher Gold. Crust and anti-crust: a one-step boundary and skeleton extraction algorithm. In *In Proceedings of the ACM Conference on Computational Geometry*, pages 189–196, 1999.

[He08] Guang Qing He. Quantification of two gestalt laws using curve resconstruction, 2008. Thesis (M.A.Sc.)–Concordia Institute for Information Systems Engineering, 2008.

[JMM+99] Ian Jermyn, Alain Mayer, Fabian Monrose, Michael Kendrik Reiter, and Aviel David Rubin. The design and analysis of graphical passwords. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, pages 1–1, Berkeley, CA, USA, 1999. USENIX Association.

[KW07] D. Brett King and Michael Wertheimer. *Max Wertheimer and Gestalt Theory.* Transaction, USA, 2007.

[Li07] Shu Ren Li. Vision-based curve reconstruction, 2007. Thesis (M.A.Sc.)–Dept. of Electrical and Computer Engineering, Concordia University, 2007.

[Mar01] George E. Martin. *Counting: The art of enumerative combinatorics.* Springer, 2001.

[NZ08] Thanh An Nguyen and Yong Zeng. Vicur: A human-vision-based algorithm for curve reconstruction. *Robot.Comput.-Integr.Manuf.*, 24(6):824–834, 2008.

[Tao] Hai Tao. Pass-go, a new graphical password scheme, 2006. Thesis (M.A.Sc.)–University of Ottawa, 2006.

[TvO04] Julie Thorpe and Paul C van Oorschot. Towards secure design choices for implementing graphical passwords. In *Computer Security Applications Conference, 2004. 20th Annual*, pages 50–60, 2004.

[vOT08] Paul C van Oorschot and Julie Thorpe. On predictive models and user-drawn graphical passwords. *ACM Trans.Inf.Syst.Secur.*, 10(4):1–33, 2008.

[ZNYL08] Yong Zeng, Thanh An Nguyen, Baiquan Yan, and Shuren Li. A distance-based parameter free algorithm for curve reconstruction. *Comput.Aided Des.*, 40(2):210–222, 2008.