Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 450460, 17 pages
doi:10.1155/2010/450460

Research Article

Point Ordering with Natural Distance Based on Brownian Motion

Philsu Kim[1] and Hyoungseok Kim[2]

[1] *Department of Mathematics, Kyungpook National University, Daegu 702-701, South Korea*
[2] *Department of Multimedia Engineering, Dongeui University, Busan 614-714, South Korea*

Correspondence should be addressed to Hyoungseok Kim, hskim@deu.ac.kr

Received 7 August 2009; Revised 24 November 2009; Accepted 12 February 2010

Academic Editor: Gradimir V. Milovanović

We consider a curve reconstruction problem from unorganized point clouds with noise. In general, the result of curve reconstruction depends on how to select and order the representative points to resemble the shape of the clouds. We exploit a natural distance based on a property of one-dimensional Brownian motion to order sample points, which simultaneously reflect smoothness and nearness of points, so that our algorithm is able to reconstruct not only simple curves but also nonsimple curves. Numerous examples show that this algorithm is effective. The natural distance proposed in this paper is able to play an important role in a variety of fields of measuring the distance of points with considering direction.

1. Introduction

In many engineering fields such as the reverse engineering of geometric models and image processing of medical images, great attention has been shown to the problems of reconstructing a shape from sample points. The curve reconstruction problem we want to discuss in this article plays an especially important role in shape reconstruction problems (see, e.g., [1–4] and references therein) and has also been instrumental in the recent realizations of many algorithms.

Curve reconstruction problems can be regarded as the problem of fitting a curve or a family $\mathcal{F}$ of curves to an irregularly spaced set $\mathcal{P}$ of unorganized data points. Most existing techniques for solving the problem can be divided into two categories depending on whether the sample points of $\mathcal{P}$ are distributed exactly on the family $\mathcal{F}$ of curves without noise or are distributed around $\mathcal{F}$ with some noise (possibly a point cloud around $\mathcal{F}$). The techniques in the first category can also be classified depending on situations where the curves are either closed or open, either smooth or nonsmooth, and the sample points are distributed either uniformly or non-uniformly and so on. In any case, the techniques in the first category are well developed by many researchers (e.g., see [5–10] and the references therein) and most of these techniques give mathematical guarantees. There are a few algorithms working for uniformly sampled curves such as alpha shape [9], r-regular shape [6], and EMST [11]. There are other algorithms that work for nonuniformly sampled curves such as Crust [5], Nearest Neighbor [8], Conservative Crust [12], Traveling Salesman Path [13], and Gathan [14, 15]. The algorithms provide a sampling condition with respect to geometric properties, which may result in a sample set not natural to human vision.

Recently, there are two algorithms which use two important properties of human vision: proximity and smoothness [16, 17]. Although the sample points are non-uniformly distributed, they can well reconstruct open simple and nonsmooth curves because they use several local features such as the mean of the distances and the mean of the angles of already-obtained edges. The creative vision function of VICUR [16] is a mathematical representation with experimental foundation. They considered only proximity and smoothness of the sample points on the Delaunay triangulation. That is, the candidates for the next point are limited to only points adjacent to the last point although the other points may produce smoother reconstructions. Thus, this is not able to reconstruct nonsimple curves and solve the curve reconstruction problem with noise.

Even though the second category, handling data points with noise, is closer to practical situations, relatively little research has been carried out on the second category (e.g., see [10, 18–20] and the references therein). Fang and Gossard [10] used a method based on spring energy minimization to approximate an unorganized point set with a curve, which requires a good initial guess to find the solution. Taubin and Ronfard [20] reconstructed a planar curve from unorganized points using an implicit simplicial curve, which is defined by a planar triangular mesh and the values at the vertices of the mesh. Lee [19] proposed an algorithm based on a modified least-squares method and Euclidean minimum spanning tree. This method is not appropriate for curves with self-intersections. Most curve reconstruction algorithms mentioned above are based on the Euclidean distance to compute the proximity and the adjacency without consideration for smoothness, so they do not solve nonsimple curve reconstruction.

Assume that the set $\mathcal{P}$ of data points we want to handle fits into the second category. More precisely, let $\mathcal{P} = \{q_k = (x_k, y_k) \in \mathbb{R}^2 : k = 1, 2, \ldots, n\}$. As indicated by Dedieu and Favardin in [7], one of the main steps for solving the curve reconstruction problem is to give an order on a subset of $\mathcal{P}$ to fit a curve interpolating the points in the subset. If we denote the ordered points as p_1, p_2, p_3, and so on, and we assume that $p_1, \ldots, p_i$ are already ordered, then the central problem is to find a way of reasonably selecting a next successive point p_{i+1} from a subset $\mathcal{C}_i \subset \mathcal{P} \setminus \{p_1, \ldots, p_i\}$, where we assume that the known facts are only the coordinates of points. For the case of the set $\mathcal{P}$ in the first category, it is sufficient to estimate the angle between the lines for three consequence points, or estimate the Euclidean distance between nested points [7]. However, a separate usage of these two factors (angle and Euclidean distance) is not sufficient in the presence of noise.

What we try to do in this paper is to expound a new persuasive measure (called a *natural distance*) giving a smoothness and closeness between points, and applicable to order the points in the second category by combining naturally two facts: angle and Euclidean distance. The natural distance to pick up the next consecutive point p_{i+1} from $\mathcal{C}_i$ is defined by

$$f_i(q) = t_i(q) + K \frac{s_i(q)}{\sqrt{t_i(q)}}, \quad q \in \mathcal{C}_i, \tag{1.1}$$

where

$$t_i(q) = |\overrightarrow{p_i q}| \cos\theta, \qquad s_i(q) = |\overrightarrow{p_i q}| \sin\theta, \qquad \theta = \theta(q) = \cos^{-1}\left(\frac{\overrightarrow{p_{i-1}p_i} \cdot \overrightarrow{p_i q}}{|\overrightarrow{p_{i-1}p_i}||\overrightarrow{p_i q}|}\right). \tag{1.2}$$

Hereafter, we call $t_i(q)$ a *time distance*, $s_i(q)/\sqrt{t_i(q)}$ a *standardized probability distance,* and K a *subjective weight*. By taking into account simple properties of Brownian motion, we will interpret a naturalness about the distance $f_i(q)$ ordering the points in a subset of $\mathcal{P}$. In particular, we will show that, the time distance $t_i(q)$ is a diffusing time measuring the closeness between points, while the standardized probabilistic distance $s_i(q)/\sqrt{t_i(q)}$ is a function of measuring the smoothness between points. Finally, we will show that, under an appropriate lower bound of the subjective weight K, the natural distance $f_i(q)$ provides a plausible answer in ordering the points of $\mathcal{P}$. Also we will provide remarkable results through several figures of numerical experiments.

2. Natural Distance

In this section, we will discuss a naturalness for the distance $f_i(q)$ defined in (1.1). In order to do this, we must first define a naturalness for a measure between points applicable to order the points. For chosen consecutive points p_{i-1} and p_i which are already ordered, let $\mathcal{C}_i$ be a subset of $\mathcal{P}$ such that for any $q \in \mathcal{C}_i$ the angle between $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_i q}$ is acute. If a measure μ prefers to pick up the next point $p_{i+1} \in \mathcal{C}_i$ satisfying the following conditions, then we say that the measure has a naturalness and call it a *natural distance*.

(C.1) Suppose that p_i and $q_1, q_2, \ldots, q_m \in \mathcal{C}_i$ are on the same straight line. Then the measure prefers the closest one to the previous point p_i among q_j, $j = 1, 2, \ldots, m$.

(C.2) Suppose that $q_1, q_2, \ldots, q_m \in \mathcal{C}_i$ are on the same circle with center p_i and the angle between $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_i q_k}$, $1 \le k \le m$, is the smallest angle among those between $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_i q_j}$ for all $j = 1, 2, \ldots, m$,. Then the measure prefers q_k among q_j, $j = 1, \ldots, m$ as the next point of p_i.

Two conditions (C.1) and (C.2) are similar to the first and second criterions of [7], respectively. To show a naturalness of $f_i(q)$, we will use a simple property of one-dimensional Brownian motion introduced in the following subsection.

An easier way to understand Brownian motion is to think of it as a continuous version of the random walk process. As documented in an enormous amount of literature on stochastic processes such as [21, 22], Brownian motion has several properties which explain the irregular motion observed in nature; for example, in two-dimensional space the motion of pollen grains suspended in liquid which does not flow. Among those, we use only the following property throughout this paper.

(P.B) For given time T, $B(T, \cdot)$ starting at 0 is a *random variable* normally distributed with mean 0 and variance T, where $B(T, \cdot), T \ge 0$ is the Brownian motion process.

Roughly speaking, (P.B) means that the set of original points on the real line at time 0, which move randomly, becomes the set which is normally distributed with mean 0 and variance T after T time.

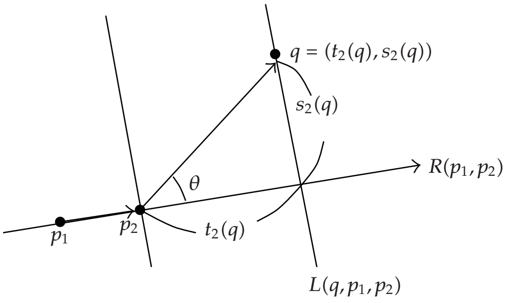


Figure 1: New coordinates for a scattered point q with $\theta > 0$ when $i = 2$.

2.1. A Motivation of the Distance $f_i(q)$

Before proving the naturalness for $f_i(q)$, we first explain a motivation of the derivation of $f_i(q)$. To do this, we begin with the following interpretation for the set $\mathcal{C}_i$. For two ordered points p_{i-1} and p_i, ($i \geq 2$), which were previously chosen, let $R(p_{i-1}, p_i)$ be the ray with initial point p_i and directional vector $\overrightarrow{p_{i-1}p_i}$ and let $L(q, p_{i-1}, p_i)$ be the perpendicular line to $R(p_{i-1}, p_i)$ passing through the point $q \in \mathcal{C}_{i+1}$ satisfying (C.1). Also define the signed angle θ ($-\pi/2 < \theta < \pi/2$) between $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_iq}$ by

$$\cos\theta = \frac{\langle \overrightarrow{p_{i-1}p_i}, \overrightarrow{p_iq} \rangle}{|\overrightarrow{p_{i-1}p_i}||\overrightarrow{p_iq}|}, \tag{2.1}$$

where the sign of θ is positive if q is located on the left-hand side of $R(p_{i-1}, p_i)$ and is negative on the right-hand side. Here, $\langle \cdot, \cdot \rangle$ denotes the inner product in $\mathbf{R}^2$. Define two functions $t_i : \mathcal{C}_{i+1} \to \mathbf{R}^+$ and $s_i : \mathcal{C}_{i+1} \to \mathbf{R}$ by

$$t_i(q) = |\overrightarrow{p_iq}| \cos\theta, \qquad s_i(q) = |\overrightarrow{p_iq}| \sin\theta. \tag{2.2}$$

The function $t_i(q)$ is the Euclidean distance between p_i and the intersection of $R(p_{i-1}, p_i)$ and $L(q, p_{i-1}, p_i)$, and the function $|s_i(q)|$ is the Euclidean distance between q and the intersection of $R(p_{i-1}, p_i)$ and $L(q, p_{i-1}, p_i)$. Then we can consider the pair $(t_i(q), s_i(q))$ as the new coordinates representation of q (see Figure 1).

With the property (P.B) of Brownian motion, we further think of the t_i-axis as the time axis with the original point p_i, and s_i-axis as the axis representing the value of Brownian motion. Therefore, if $t_i(q) = t > 0$, $s_i(q)$ is considered as a sample point of the random variable which has the normal distribution with mean 0 and variance t as shown in Figure 2, then we consider the point q as one of the normally distributed points after t time, which is randomly diffused from original point.

With this setting, we want to find a function F providing the distances between p_i and candidate points of p_{i+1}, which prefer to choose a point $q \in \mathcal{C}_{i+1}$ satisfying the conditions (C.1) and (C.2) where F has the minimum value. Then what is suitable for variables of F? We suggest two factors as variables of F.

We adopt the function t_i, called ith *time distance*, as its first variable. The condition (C.1) in the previous section requires that F should be strictly increasing for t_i. This implies that we prefer to choose the point taking the smaller diffusion time to p_{i+1}.

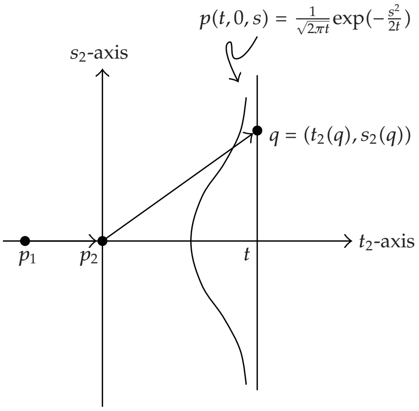


Figure 2: The distribution of Brownian motion after time $t = t_2(q)$.

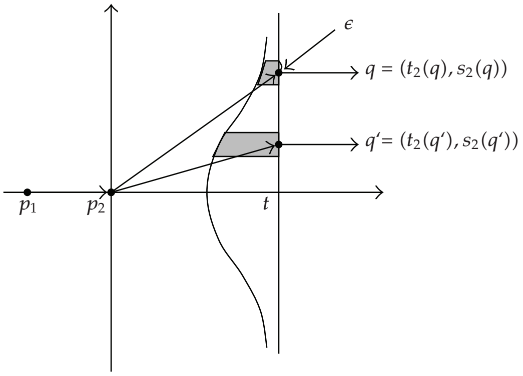


Figure 3: The probabilities of the ϵ-neighborhoods of two points $q,\ q'$ for $i = 2$.

Similarly thinking, one may take the function $|s_i|$ as the second variable of F. As shown in Figure 3, the probability density p satisfies that, for arbitrary $\epsilon > 0$ and two points $q, q' \in \mathcal{C}_{i+1}$ such that $t_i(q) = t_i(q') = t$ and $|s_i(q)| > |s_i(q')|$,

$$\int_{N(s_i(q),\epsilon)} p(t,0,s)ds < \int_{N(s_i(q'),\epsilon)} p(t,0,s)ds. \tag{2.3}$$

However, the function s_i is not enough as the second variable of F because it is not independent of the first variable t_i. Assume that $q, q' \in \mathcal{C}_{i+1}$ are the points having different reaching times, that is, $t_i(q) \neq t_i(q')$. Since $s_i(q)(s_i(q'))$ is interpreted as a sample point of the random variable with a normal distribution with variance $t_i(q)(t_i(q'), \text{resp.})$, we cannot compare the probability in the same way as above. For this reason, it is necessary to standardize random variables with normal distributions with the variance t_i to be the standardized random variable with normal distribution with the same mean and the variance 1. So we adopt the function $|s_i|/\sqrt{t_i}$ as the second variable of F. We call this function ith *standardized probability distance*. Easily speaking, if we ignore the effect of the ith time distance, we think that the original point p_i is more naturally attainable to q' than q whenever

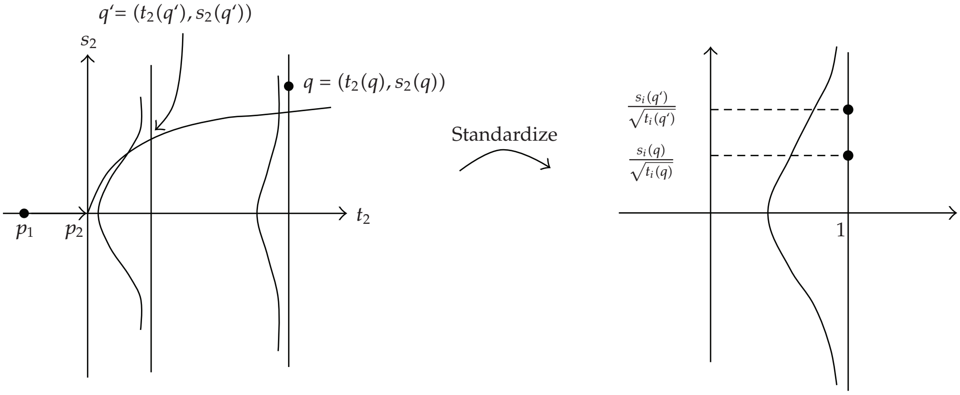

$q' = (t_2(q'), s_2(q'))$
s_2
$q = (t_2(q), s_2(q))$
t_2
p_1
p_2
Standardize
$\frac{s_i(q')}{\sqrt{t_i(q')}}$
$\frac{s_i(q)}{\sqrt{t_i(q)}}$
1

Figure 4: Standardizations of $s_i(q)$ and $s_i(q')$ when $i = 2$.

$|s_i(q')|/\sqrt{t_i(q')} > |s_i(q)|/\sqrt{t_i(q)}$ (see Figure 4). The condition (C.2) in the previous section requires that F should be strictly increasing for this second variable, though this is not a sufficient condition for (C.2).

Consequently, our problem is rewritten as the following, if F is assumed to be C^1.

> Choose $q \in \mathcal{C}_{i+1}$ satisfying the conditions (C.1) and (C.2) where the suitable (still undetermined) C^1-function
>
> $$F\left(t_i(q), \frac{|s_i(q)|}{\sqrt{t_i(q)}} \right) \tag{2.4}$$
>
> has the minimum value. Here the function $F(x, y) : \mathbf{R}^+ \times \mathbf{R}^+ \cup \{0\} \rightarrow \mathbf{R}^+$ strictly increases for both variables, that is, $F_x > 0$ and $F_y > 0$.

There are plenty of such increasing function F with two variables. Among all, the simplest one will be the linear function of the form

$$F(x, y) = x + Ky, \tag{2.5}$$

for a positive constant K. This is mostly caused by the convenience of calculation, but the function has several good features as the solution for the point ordering problem. The reader can check these, exploring some examples in the next section.

In this spirit, we define the ith *natural distance* f_i $(i \geq 2)$ by, for $q \in \mathcal{C}_{i+1}$ and a positive constant K,

$$f_i(q) = t_i(q) + K\frac{|s_i(q)|}{\sqrt{t_i(q)}}. \tag{2.6}$$

We call K the *subjective weight*, since it represents how much weight is given to the second factor.

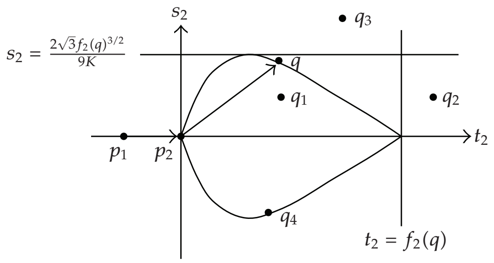


Figure 5: The level curve passing through a point q for $i = 2$.

2.2. *Naturalness of* $f_i(q)$

To prove the naturalness of $f_i(q)$, we need to make some technical assumptions. Choosing points on the same level curve is problematic, where the points have the same natural distance from p_i. If points lie on the same level curve with a different Euclidean distance from p_i, we choose the closest point from p_i to the Euclidean distance. Also, if two points lie on the same level curve with the same Euclidean distance from p_i, we choose the point lying on the right-hand side of the direction $\overrightarrow{p_{i-1}p_i}$.

The ith natural distance f_i has the following properties, which can be easily checked.

(P.1) If q_1 lies in the interior of the *level curve* $t_i + K(|s_i|/\sqrt{t_i}) = f_i(q)$, then $f_i(q) > f_i(q_1)$,

(P.2) If $f_i(q) < t_i(q_2)$, then $f_i(q) < f_i(q_2)$,

(P.3) If $2\sqrt{3}f_i(q)^{3/2}/9K < |s_i(q_3)|$, then $f_i(q) < f_i(q_3)$,

(P.4) If q_4 lies on the level curve $t_i + K(|s_i|/\sqrt{t_i}) = f_i(q)$, then $f_i(q) = f_i(q_4)$.

Some of these can be used to simplify numerical algorithms ordering the points. Here $q, q_j \in C_{i+1}$ for $j = 1, 2, 3, 4$ (see Figure 5). (P.1) implies that the level curves do not intersect each other in (t_i, s_i)-plane, so the natural distance $f_i(q)$ satisfies (C.1). (P.3) is derived by the fact that the maximum value of the level curve $t_i + K(|s_i|/\sqrt{t_i}) = f_i(q)$ is $2\sqrt{3}f_i(q)^{3/2}/9K$. In the following theorem, we will show that the natural distance $f_i(q)$ locally satisfies (C.2).

Theorem 2.1. *Assume that*

$$K > \left(\frac{27}{83 + 48\sqrt{3}}\right)^{1/4} \sqrt{r_0} \tag{2.7}$$

for some fixed positive r_0. Then the natural distance $f_i(q)$ satisfies the condition (C.2).

Proof. To show (C.2), it is enough to show that, for any fixed $c > 0$ and $0 < r < r_0$, the system of equations

$$\begin{gathered} x^2 + y^2 = r^2, \\ x + K\frac{y}{\sqrt{x}} = c \end{gathered} \tag{2.8}$$

has at most one root in the first quadrant $x > 0$, $y > 0$.

The second equation of (2.8) gives the equation $y^2 = x(x^2 - 2cx + c^2)/K^2$. Hence from the first equation of (2.8), we have

$$g(x) = K^2 r^2, \quad \text{where } g(x) = x\left(x^2 + \left(K^2 - 2c\right)x + c^2\right). \tag{2.9}$$

Thus, it suffices to prove that two curves $y = g(x)$ and $y = K^2 r^2$ meet at most one point in the first quadrant $x > 0, y > 0$. Note that, if the cubic function $g(x)$ has three zeros, all of them must be less than zero because $x^2 + (K^2 - 2c)x + c^2 = (x - c)^2 + K^2 x > 0$, and hence $g(x) > 0$ provided $x > 0$. In this case, the intersection of two curves $y = g(x)$ and $y = K^2 r^2$ on the first quadrant $x > 0, y > 0$ is at most one.

First, assume that $g(x) = 0$ has only one zero at $x = 0$. If the quadratic equation $g'(x) = 0$ does not have solution or has negative solutions, then $g(x)$ is a strictly increasing function for $x > 0$, and hence the system has at most one root. Therefore the only case that has to be considered is the case

$$K^2 < \left(2 - \sqrt{3}\right)c, \quad \text{or equivalently } c > \left(2 + \sqrt{3}\right)K^2. \tag{2.10}$$

Note that, in this case, the quadratic equation $g'(x) = 0$ has two distinct positive solutions

$$x_1 = \frac{2c - K^2 - X}{3}, \qquad x_2 = \frac{2c - K^2 + X}{3}, \tag{2.11}$$

where $X = \sqrt{c^2 - 4cK^2 + K^4}$, and the minimal extreme value of $g(x)$ is obtained at x_2. Using the relation $g'(x_2) = 0$, we can derive

$$\begin{aligned} g(x_2) &= -\frac{2}{9}x_2\left(K^4 - 4cK^2 + c^2\right) - \frac{1}{9}c^2\left(K^2 - 2c\right) \\ &= -\frac{2}{9}x_2 X^2 - \frac{1}{9}c^2\left(K^2 - 2c\right) \\ &= \frac{1}{27}\left(-2X^3 + 2c^3 + K^2\left(15c^2 - 12K^2 c + 2K^4\right)\right). \end{aligned} \tag{2.12}$$

From (2.10) it can be easily verified that X has the maximum value c. So $g(x_2)$ is bounded such as

$$g(x_2) \geq \frac{K^2}{27}\left(15c^2 - 12K^2 c + 2K^4\right) \geq \frac{83 + 48\sqrt{3}}{27}K^6. \tag{2.13}$$

The last inequality is derived from $c > (2 + \sqrt{3})K^2$. Hence if the inequality

$$r^2 K^2 \leq \frac{83 + 48\sqrt{3}}{27}K^6, \quad \text{or equivalently } K > \left(\frac{27}{83 + 48\sqrt{3}}\right)^{1/4}\sqrt{r}, \tag{2.14}$$

holds, then the proof is complete. □

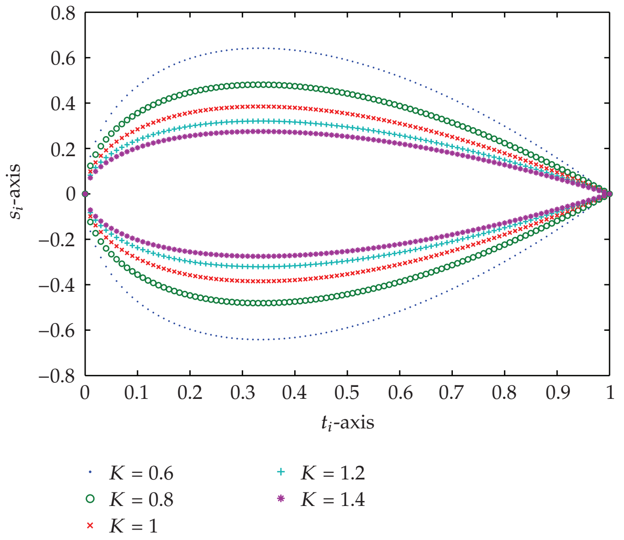


Figure 6: Level curves for various subjective weights K.

From Theorem 2.1, we know that points lying on the same level curve with the same Euclidean distance to the points should be located symmetrically with respect to the t_i-axis. Thus if there exist such points, then there are only two in $N(p_i, r_0)$.

Now, we need to make a practical remark concerned with Theorem 2.1. As seen in its proof, the lower bound for K is not obtained optimally, so we conjecture that this bound can be smaller in most cases. For practical usage of our method, we think that it is not inevitably necessary to give a consideration for the bound, since the purpose of our point ordering is to construct (or reconstruct) meaningful shapes (or the original shape) after connecting all ordered points. For this reason, in the following experiments, we are not concerned with the lower bound for K on using the natural point ordering method.

Before closing this section, we remark on the effect of the subjective weights K on the point orderings. It is obvious that, if K is very big, the ith natural distance f_i is more sensitive to magnitudes of the changes of the ith standardized probability distance $|s_i|/\sqrt{t_i}$ than those of the ith time distance t_i. This fact can be checked in Figure 6, which presents the level curves of (2.6) for several values of the subjective weight K. As K increases, the width between the line $s_i = 0$ and level curve becomes narrower.

3. Practical Algorithm

In this section, we will describe a practical algorithm for ordering points in an unorganized point cloud as well as a sample point set.

Let $\mathcal{P}$ be a set of arbitrarily scattered n points in $\mathbf{R}^2$. The *point ordering problem* aims to choose a family $\mathcal{S} = \{\widetilde{\mathcal{P}}_j\}$ of subsets of $\mathcal{P}$, where each subset $\widetilde{\mathcal{P}}_j$ is a well-ordered set satisfying the natural conditions (C.1) and (C.2). Assume that $p_1^j, p_2^j, \ldots, p_m^j$ are already ordered so that we may denote that $\widetilde{\mathcal{P}}_j = [p_1^j, p_2^j, \ldots, p_m^j]$ with the last point p_m^j and the direction $\overrightarrow{p_{m-1}^j p_m^j}$. We

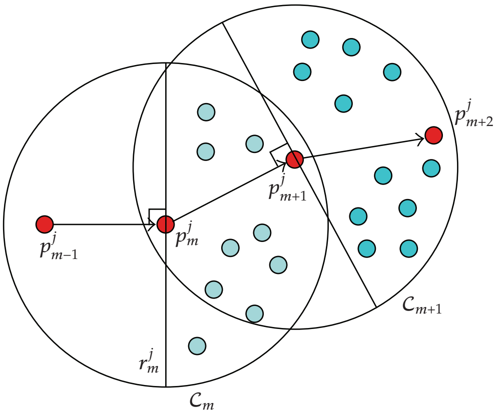


Figure 7: The candidate points of p^j_{m+1} are in the right half-disc $\mathcal{C}_m$ with center p^j_m and radius r^j_m.

define the connectivity t^j_m of an ordered subset as the mean of the lengths of edges of the ordered subset: $t^j_m = (1/(m-1))\sum_{l=1}^{m-1} d^j_l$, where d^j_l is the distance between p^j_l and p^j_{l+1}.

We explain a process for computing the set of candidates for the next point and choosing the next point among the candidates, called *NextPointFinder*. Let $N(p^j_m, r^j_m)$ be a neighborhood with center p^j_m and radius $r^j_m = at^j_m$ satisfying the following condition:

$$N\left(p^j_m, r^j_m\right) = \left\{q \in \mathcal{R} : d\left(p^j_m, q\right) \leq r^j_m\right\}, \tag{3.1}$$

where $\mathcal{R}$ is the remaining point set such that $\mathcal{R} = \mathcal{P} \setminus \bigcup_{l=1}^{j} \widetilde{\mathcal{P}}_l$ and a is a parameter. We define the candidate set $\mathcal{C}_m$ by

$$\mathcal{C}_m = \left\{q \in N\left(p^j_m, r^j_m\right) : \overrightarrow{p^j_{m-1}p^j_m} \cdot \overrightarrow{p^j_m q} > 0\right\} \tag{3.2}$$

as shown in Figure 7. We find out the point p^j_{m+1} among the elements of $\mathcal{C}_m$ such that $f_m(p_{m+1}) = \min_{q\in\mathcal{C}_m}\{f_m(q)\}$.

There is a main difference between our method and VICUR [16] in the process. Our method permits all of points $q \in \mathcal{C}_m$ as a candidate for p_{m+1} whether $[p_m, q]$ is a Delaunay edge or not, whereas VICUR considers only the points q, where $[p_m, q]$ is a Delaunay edge. We think that it plays a role of a key to solve a nonsimple curve reconstruction problem.

In order to deal with the first category problem, we assume that there is no noise in the point set $\mathcal{P}$.

NADIAS (NAtural DIstance based Algorithm for Sample points)

Step 1. Compute Delaunay triangulation DT($\mathcal{P}$).

Step 2. Let $\widetilde{\mathcal{P}}_j$ be the jth local solution generated by NextPointFinder. Let $e^j = [p^j_1, p^j_2]$ be the shortest Delaunay edge in $\mathcal{R}$. Apply NextPointFinder with the last point p^j_2 and the direction $\overrightarrow{p_1^j p_2^j}$ until the candidate set is empty. Let $\widetilde{\mathcal{P}}_{j_+}$ be the solution of this process. Apply NextPointFinder with the last point p_1^j and the direction $\overrightarrow{p_2^j p_1^j}$ until the candidate set is empty. Let $\widetilde{\mathcal{P}}_{j_-}$ be the solution of this process. Merge the two subsets $\widetilde{\mathcal{P}}_{j_+}$ and $\widetilde{\mathcal{P}}_{j_-}$ into $\widetilde{\mathcal{P}}_j$, that is, $\widetilde{\mathcal{P}}_j = \widetilde{\mathcal{P}}_{j_+} \cup \widetilde{\mathcal{P}}_{j_-}$. Update $\mathcal{S} = \mathcal{S} \cup \{\widetilde{\mathcal{P}}_j\}$. Update $\mathcal{R} = \mathcal{P} \setminus \bigcup_{l=1}^{j} \widetilde{\mathcal{P}}_l$.

Step 3. If $|\mathcal{R}| > 1$, then repeat Step 2.

Step 4. If $|\mathcal{S}| > 1$, then apply the following bridging process to a pair of $\widetilde{\mathcal{P}}_i$ and $\widetilde{\mathcal{P}}_j$ of $\mathcal{S}$: for two ordered subsets $\widetilde{\mathcal{P}}_i = [p_1^i, p_2^i, \ldots, p_l^i]$ and $\widetilde{\mathcal{P}}_j = [p_1^j, p_2^j, \ldots, p_m^j]$, if the distance between p_1^i (or p_l^i) and p_1^j (or p_m^j) is less than the average of the connectivity t_l^i and t_m^j, then our algorithm connects these two ordered subsets.

Step 2 says that our algorithm follows a type of an incremental greedy approach: at each step it chooses the next point from among the candidate points that is best under the natural distance, including "close to the last point" and "in the direction of current travel". The bridging process is similar to the curve-curve connectivity of [16] and makes our method reconstruct curves with sharp corners.

The time complexity of our algorithm is $O(n \log n)$, where n is the number of the input points because the dominant process of our algorithm is to compute Delaunay triangulation of them and it runs in $O(n \log n)$. The main process of our algorithm is NextPointFinder which is repeatedly to find out a candidate set $\mathcal{C}_i$, and to select one point that has the minimum natural distance among the candidates. It takes $O(kM)$, where k is the number of the output points in $\widetilde{P}$ and M is a constant.

For the second category problem, we outline our algorithm as follows.

NADIAC (NAtural DIstance based Algorithm for Cloud points)

Step 1. Compute Delaunay triangulation $DT(\mathcal{P})$.

Step 2. Choose a point $p_1 \in \mathcal{P}$. Compute a neighborhood $N(p_1, r_c)$, where r_c is a constant radius. Apply Principal Component Analysis to $N(p_1, r_c)$ in order to obtain the initial direction $\overrightarrow{\mathrm{ID}}$ which plays a role of the major principal axis in the neighborhood.

Step 3. Apply NextPointFinder with the last point p_1 and the direction $\overrightarrow{\mathrm{ID}}$ until the candidate set is empty. Let $\widetilde{P_+}$ be a local solution generated by NextPointFinder.

Step 4. Apply NextPointFinder with the last point p_1 and the direction $\overrightarrow{-\mathrm{ID}}$ until the candidate set is empty. Let $\widetilde{P_-}$ be a local solution generated by NextPointFinder.

Step 5. $\widetilde{P} = \widetilde{P_+} \cup \widetilde{P_-}$.

4. Experimental Results

In this section, we tested several reconstruction problems where each problem is focused on the following facts: (1) nonsimple curve from sample points, (2) sharp corners from sample

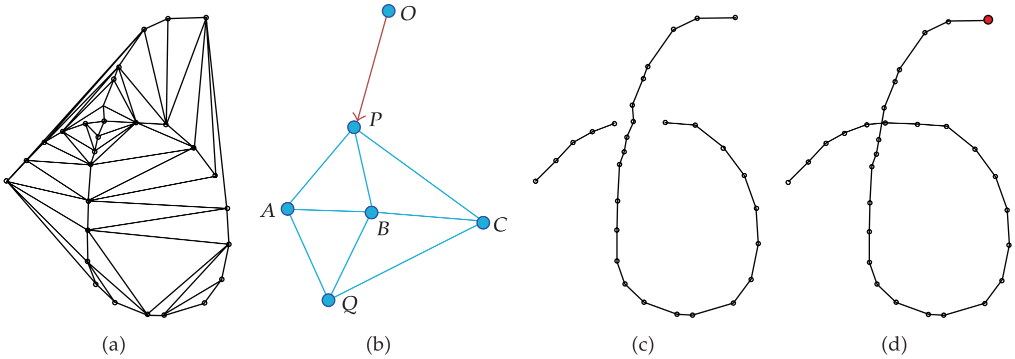


Figure 8: Nonsimple curve reconstruction: (a) Delaunay triangulation, (b) enlarged Delaunay triangulation at the dense area, (c) reconstructed curve generated by [16], and (d) reconstructed curve generated by our method.

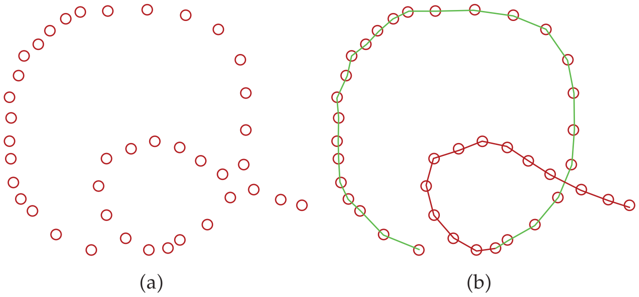


Figure 9: Nonsimple curve reconstruction: (a) sample points and (b) result by NADIAS.

points, (3) the difference between Natural distance and Euclidean distance, (4) the effect of the subjective weight K, and (5) the treatment of cloud sets.

Figure 8 shows that our method can reconstruct a nonsimple curve from sample points. Figure 8(a) is the Delaunay triangulation of the sample points. In order to explain the difference between VICUR [16] and our method, we zoom in the Delaunay triangulation as shown in 8(b). The adjacent points to P are A, B, and C, whereas Q is not adjacent to P. So VICUR does not permit Q as a candidate for the next point of P although the edge PQ is smoother to the already-ordered edge OP than the other Delaunay edges. On the other hand, our algorithm deals with all of four points as candidates for the next point of P since they belong to the neighborhood of P with radius $a \cdot t$, where t is its connectivity and $a = 2.5$ is set. Figure 8(d) shows a nonsimple curve reconstructed by our method. The red point in Figure 8(d) means the initial point of our algorithm.

Figure 9 shows another example for a nonsimple curve from sample points as shown in Figure 9(a). In this example, we choose the initial point as that having a minimum edge length between points. In Figure 9(b), the boundary point between red line and green line is the chosen initial point. Thus, our algorithm works in two opposite directions. Figures 8 and 9 are smooth nonsimple curves.

Figure 10 shows the process of our algorithm, called NADIAS (NAtural DIstance based Algorithm for Sample points). Figure 10(a) is an example for sample points on a piecewise smooth curve. For the sample data, NADIAS finds out the shortest Delaunay edge

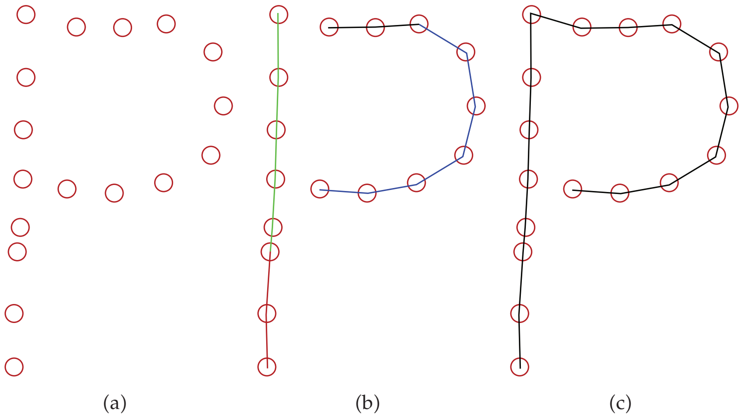


Figure 10: The process of NADIAS: (a) sample points, (b) NextPointFinder, and (c) bridging.

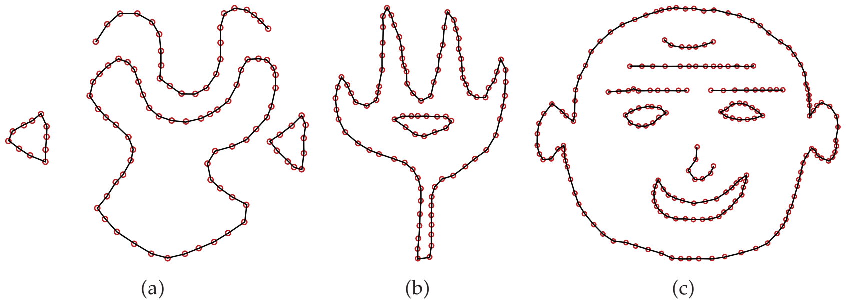
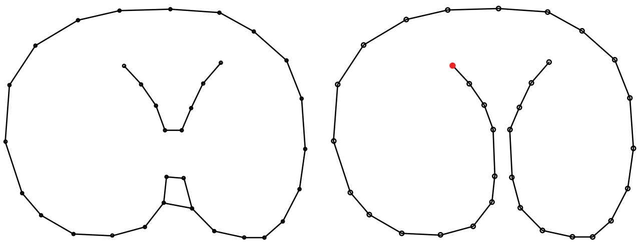


Figure 11: Examples with multicomponents.

and then applies NextPointFinder process two times with opposite directions. Figure 10(b) shows the result which has two components of smooth curves. The boundary of the red line and the green line is the initial point of the first component, and the boundary of the black line and the blue line is one of the second component. Figure 10(c) is the result after our bridging process. This figure contains one component because the minimum distance between the end points of two components is smaller than the given threshold distance. Figure 11 shows that NADIAS is able to well extract the several components whether or not each component is open and whether or not its shape is sharp.

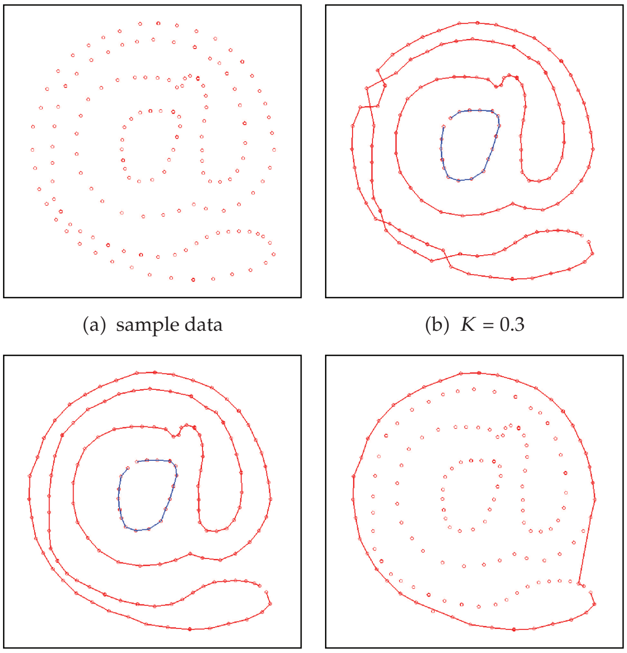
There are two different results of point ordering for the same set of sample points by using Euclidean distance and Natural distance in Figure 12. The ordered set with Euclidean distance in (a) does not contain all sample points, contrary to that with natural distance in (b). It implies that there is an example in which parameters can be chosen for the natural point ordering that will reconstruct the curve, while parameters chosen for Euclidean distance may not.

In Figure 13, the complicated examples for three different subjective weights are presented. For the set of scattered sample points in (a), the case $K = 0.5$ of (c) produces a meaningful closed curve of ‘@’ almost completely. One also finds the fact that, as K increases, our method produces smoother shapes. Figures 13(b) and 13(c) are generated by the first algorithm which regards the point data as sample points, so there are two curves. On the other hand, Figure 13(d) shows the result of the second algorithm. K is very large, so the



(a) Euclidean distance [5] (b) Natural distance

Figure 12: Point orderings based on the different distance measures.



(a) sample data (b) $K = 0.3$

(c) $K = 0.5$ (d) $K = 3.0$

Figure 13: A complicated example using the natural point ordering method.

smoothness is emphasized more than the proximity. Therefore the algorithm meets the stop condition before most of points are applied to the algorithm.

Figure 14 shows a more meaningful example of the natural point ordering problem. The sample data are generated by drawing a spiral and then adding three small circles to the spiral. One may understand that the major configuration of the dataset is a spiral. Our natural point ordering method can cut out the minor configurations such as three small circles from the spiral, so the ordered set is only on the spiral.

Figure 15 shows the results of NADIAC for point clouds. The cloud data of a spiral type are generated by random process and the clouds of the right example are generated by drawing a nonsimple curve in several times. Although the set of sample points is

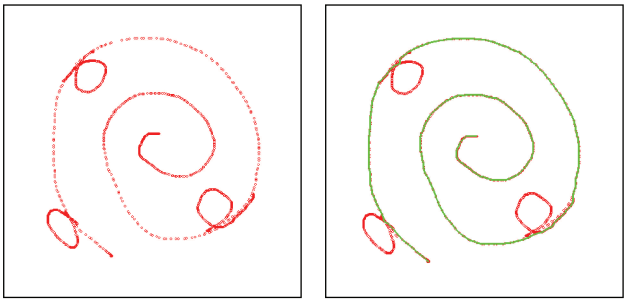

(a) (b)

Figure 14: Extraction of the major configuration: sample points (a) and the result (b) with $K = 0.8$.

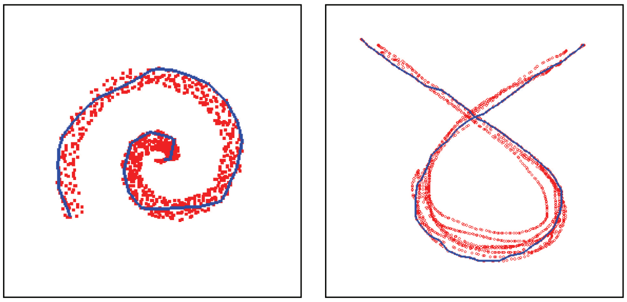

(a) (b)

Figure 15: Natural point orderings for point clouds: a point cloud generated by random process (a), a point cloud generated by drawing with mouse (b).

unorganized, our natural point ordering algorithm can extract the ordered subset from the point cloud, which plays a role of representative in their local neighborhood.

5. Conclusion

We have presented a new method for constructing curves from unorganized point clouds with noise. In general, the result of curve reconstruction depends on how to select and order the representative points to resemble the shape of the clouds. In this paper, in contradiction to the previous curve reconstruction algorithms based on Euclidean distance, we exploit a natural distance to reflect orientation to the ordering of sample points, so that our algorithm is able to reconstruct not only simple curves but also nonsimple curves. Moreover, for unorganized point clouds, this method efficiently extracts the skeletons of the clouds by cutting out the outliers, even though the result by our method is sensitive to the initial point and the initial direction. The method may be extended in two ways. The first thing is to extend the method to the surface reconstruction problem in three-dimensional space. We think that one should use the properties of two-dimensional Brownian motion in order to get a natural distance similar to that in this paper. Second, it is possible to improve our method which can control the subjective weight K to obtain the wanted shape. We conjecture that this can be fulfilled if we weaken the condition (C.1).

Acknowledgments

The authors would like to express their sincere gratitude to reviewers for many helpful comments and valuable suggestions on the first draft of this paper. This work was supported by Korea Research Foundation Grant KRF-2003-041-C00039.

References

[1] J.-D. Boissonnat and F. Cazals, "Smooth surface reconstruction via natural neighbour interpolation of distance functions," *Computational Geometry: Theory and Applications*, vol. 22, no. 1–3, pp. 185–203, 2002.

[2] J.-D. Boissonnat and B. Geiger, "Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation," in *Biomedical Image Processing and Biomedical Visualization*, Proceedings SPIE, pp. 964–975, San Jose, Calif, USA, February 1993.

[3] T. K. Dey, *Curve and Surface Reconstruction*, Cambridge University Press, Cambridge, UK, 2007.

[4] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proceedings of the 19th Annual ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '92)*, no. 2, pp. 71–78, Chicago, Ill, USA, July 1992.

[5] N. Amenta, M. Bern, and D. Eppstein, "The crust and the β-skeleton: combinatorial curve reconstruction," *Graphical Models and Image Processing*, vol. 60, no. 2, pp. 125–135, 1998.

[6] D. Attali, "r-regular shape reconstruction from unorganized points," in *Proceedings of 13th Annual ACM Symposium on Computational Geometry*, pp. 248–253, Nice, France, 1997.

[7] J.-P. Dedieu and C. Favardin, "Algorithms for ordering unorganized points along parametrized curves," *Numerical Algorithms*, vol. 6, no. 1-2, pp. 169–200, 1994.

[8] T. K. Dey and T. K. Kumar, "A simple provable algorithm for curve reconstruction," in *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 893–894, Baltimore, Md, USA, January 1999.

[9] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–559, 1983.

[10] L. Fang and D. C. Gossard, "Fitting 3D curves to unorganized data points using deformable curves," in *Proceedings of the 10th International Conference on Computer Graphics (CG '92)*, pp. 535–543, Tokyo, Japan, 1992.

[11] L. H. de Figueiredo and J. de Miranda Gomes, "Computational morphology of curves," *The Visual Computer*, vol. 11, no. 2, pp. 105–112, 1994.

[12] T. K. Dey, K. Mehlhorn, and E. A. Ramos, "Curve reconstruction: connecting dots with good reason," in *Proceedings of the 15th Annual Symposium on Computational Geometry (Miami Beach, FL, 1999)*, pp. 197–206, ACM, New York, NY, USA.

[13] E. Althaus and K. Mehlhorn, "TSP-based curve reconstruction in polynomial time," in *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 2000)*, pp. 686–695, ACM, New York, NY, USA.

[14] T. K. Dey and R. Wenger, "Reconstructing curves with sharp corners," *Computational Geometry: Theory and Applications*, vol. 19, no. 2-3, pp. 89–99, 2001.

[15] T. K. Dey and R. Wenger, "Fast reconstruction of curves with sharp corners," *International Journal of Computational Geometry & Applications*, vol. 12, no. 5, pp. 353–400, 2002.

[16] T. A. Nguyen and Y. Zeng, "VICUR: a human-vision-based algorithm for curve reconstruction," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 6, pp. 824–834, 2008.

[17] Y. Zeng, T. A. Nguyen, B. Yan, and S. Li, "A distance-based parameter free algorithm for curve reconstruction," *Computer Aided Design*, vol. 40, no. 2, pp. 210–222, 2008.

[18] S.-W. Cheng, S. Funke, M. Golin, P. Kumar, S.-H. Poon, and E. Ramos, "Curve reconstruction from noisy samples," *Computational Geometry: Theory and Applications*, vol. 31, no. 1-2, pp. 63–100, 2005.

[19] I.-K. Lee, "Curve reconstruction from unorganized points," *Computer Aided Geometric Design*, vol. 17, no. 2, pp. 161–177, 2000.

[20] G. Taubin and R. Ronfard, “Implicit simplicial models for adaptive curve reconstruction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 321–325, 1996.

[21] B. ∅ksendal, *Stochastic Differential Equations: An Introduction with Applications*, Universitext, Springer, Berlin, Germany, 5th edition, 1998.

[22] S. M. Ross, *Stochastic Processes*, Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics, John Wiley & Sons, New York, NY, USA, 1983.