

# Polar-Natural Distance and Curve Reconstruction

**Hyoung-Seok Kim\***

Dept. of Multimedia Engineering  
Dong-Eui University, Rep. of Korea

**Ho-Sook Kim**

Dept. of Mathematics and Computer Science  
KSA of KAIST, Rep. of Korea

## ABSTRACT

We propose a new distance measure between 2-dimensional points to provide a total order for an entire point set and to reflect the correct geometric meaning of the naturalness of the point ordering. In general, there is no total order for 2-dimensional point sets, so curve reconstruction algorithms do not solve the self-intersection problem because the distance used in the previous methods is the Euclidean distance. A natural distance based on Brownian motion was previously proposed to solve the self-intersection problem. However, the distance reflects the wrong geometric meaning of the naturalness. In this paper, we correct the disadvantage of the natural distance by introducing a polar-natural distance, and we also propose a new curve reconstruction algorithm that is based on the polar-natural distance. Our experiments show that the new distance adequately reflects the correct geometric meaning, so non-simple curve reconstruction can be solved.

**Key words:** Point Ordering, Curve Reconstruction, Polar-Natural Distance.

## 1. INTRODUCTION

The problems of reconstructing a shape from sample points appear in many scientific and engineering applications. Because of its practical importance, many algorithms have been proposed over the last two decades in the fields such as the reverse engineering of geometric models and image processing of medical images. Especially, curve reconstruction plays an important role in the shape reconstruction problems because most of boundaries of shapes are represented by the set of curves.

Curve reconstruction is the problem of computing a piecewise linear approximation to a curve from a set of sample points. So curve reconstruction consists of two steps: the first step is how to order the given points and the second step is how to interpolate the ordered points [1]-[4]. The first step is called the *point ordering problem*. Many approaches have been suggested for the point ordering problem. Edelsbrunner et al. [5] defined the  $\alpha$ -shapes of point sets as the underlying space of a simplicial complex. Attali proposed the  $r$ -regular shape method, where the  $r$ -regular shapes are characterized by requiring that any circle passing the points on the boundary has radius greater than  $r$  [6]. These methods deal with only uniform sample points.

For the non-uniform sample points, Amenta et al. [7] proposed the first algorithm to reconstruct a curve from non-uniform sample points with guarantee. This algorithm uses the Voronoi diagram and the Delaunay triangulation of the sample points. Dey et al. [8] proposed the nearest neighbor approach based on the properties of Voronoi diagrams. These algorithms work only under the assumption that the sample points are dense and do not work for non-simple curve reconstruction because they reconstruct curves without the consideration of curve's orientation [9]-[12]. Kim et al. [3] proposed a point ordering approach with a natural distance to reflect the orientation. The natural distance is based on properties of Brownian motion so that the approach enables us to give 1-dimensional order to points on 2-dimensional space. However, the approach missed important geometric property: the right direction of Brownian motion.

In this paper, we introduce a new natural distance based on polar coordinates so that the distance enables our algorithm to reflect the direction properly and resolve the non-simple curve reconstruction.

## 2. POLAR-NATURAL DISTANCE

In this section, we review the properties of Brownian motion and the natural distance [3]. First of all, we will discuss naturalness for a direction-based distance. In order to do this,

\* Corresponding author, Email: [hskim@deu.ac.kr](mailto:hskim@deu.ac.kr)

Manuscript received Jan. 29, 2015; revised Apr. 13, 2015; accepted Apr. 20, 2015

we define the naturalness for a distance between points which is suitable to order the points on a 2D plane. For chosen consecutive points  $p_{i-1}$  and  $p_i$  which are already ordered, we define the ordering direction vector as vector  $\overrightarrow{p_{i-1}p_i}$ . For the ordering direction, what will be the next point  $p_{i+1}$ ?

Let  $P$  be the sample point sets. We define the candidate set  $C_{i+1}(r_i)$  for the next point  $p_{i+1}$  as

$$C_{i+1}(r_i) = \{p \in P \mid d(p, p_i) \leq r_i\}$$

as shown in Fig. 1, where  $d(p, q)$  is the Euclidean distance between  $p$  and  $q$ , and  $r_i$  is the radius of the ordering neighborhood. The radius  $r_i$  is computed by the average length of the consecutive points:

$$r_i = \frac{1}{i-1} \sum_{k=1}^{i-1} d(p_k, p_{k+1}).$$

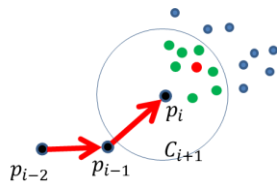


Fig. 1. The candidate set  $C_{i+1}$  for the next point  $p_{i+1}$ .

If a measure  $\mu$  prefers to pick up the next point  $q \in C_{i+1}$  satisfying the following conditions, then we say that the measure has a naturalness and call it as a *natural distance*:

(C-1) Suppose  $p_i$  and  $q_1, q_2, \dots, q_m \in C_{i+1}$  are on the same line. Then the measure prefers to choose the closest one  $q_k$  to the previous point  $p_i$  among  $q_j$ 's :

$$d(p_i, q_k) = \min_j d(p_i, q_j).$$

(C-2) Suppose  $q_1, q_2, \dots, q_m \in C_{i+1}$  are on the same circle with center  $p_i$ , and the angle between  $\overrightarrow{p_{i-1}p_i}$  and  $\overrightarrow{p_i q_k}$  is strictly smaller than the angle between  $\overrightarrow{p_{i-1}p_i}$  and  $\overrightarrow{p_i q_j}$ . Then  $q_k$  is more preferable to  $p_j$ :

$$aug(p_i, q_k) = \min_j aug(p_i, q_j)$$

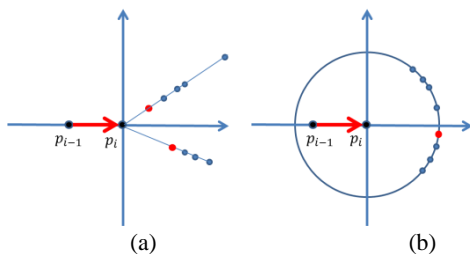


Fig. 2. Geometric conditions for naturalness of ordering: (a) (C-1) condition and (b) (C-2) condition

The condition (C-2) implies that, among the points with the same Euclidean distance from  $p_i$ , we want to choose the point which makes the arc, made by connecting the chosen

points, most smooth as shown in Fig. 2. To show the naturalness of distance, we adopt a simple property of one dimensional Brownian motion.

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 [17], [18], 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  $T$ , Brownian motion starting at 0 is a random variable normally distributed with mean 0 and variance  $T$ .

Roughly speaking, (P.B) means that the set of original points on the real line at time 0, which move randomly, become the set which is normally distributed with mean 0 and variance  $T$  after  $T$  time. Now, we will introduce the polar-natural distance function by understanding the right geometric meaning of the Brownian property.

We assume that  $p_{i-1}$  and  $p_i$  are already chosen and  $q$  is an arbitrary point in  $C_{i+1}$ . In order to determine the next point  $p_{i+1}$ , Kim et al. [3] defined the natural distance in the following manner. They applied a 2-dimension transformation so that  $p_{i-1}$  is on the negative  $x$ -axis and  $p_i$  is the origin of  $R^2$ . In order to help the geometric understanding, let  $R(p_{i-1}, p_i)$  be the ray with the initial point  $p_i$  and the directional vector  $\overrightarrow{p_{i-1}p_i}$  and  $L(q, p_{i-1}, p_i)$  be the perpendicular line to  $R(p_{i-1}, p_i)$  passing through the point  $q$  satisfying (C-1) as shown in Fig. 3. Therefore, the arbitrary point  $q$  can be represented by the new coordinates  $q = (q_t, q_s)$  as follows:

$$q_t = \|\overrightarrow{p_i q}\| \cos \theta_i(q), \quad q_s = \|\overrightarrow{p_i q}\| \sin \theta_i(q),$$

where

$$\cos \theta_i(q) = \frac{\overrightarrow{p_{i-1}p_i} \cdot \overrightarrow{p_i q}}{\|\overrightarrow{p_{i-1}p_i}\| \|\overrightarrow{p_i q}\|}$$

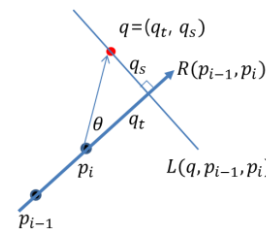


Fig. 3. New coordinates for point  $q$  [3]

The coordinate  $q_t$  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 coordinate  $q_s$  is the signed 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 think the pair  $(q_t, q_s)$  as the new coordinates of  $q$ . With the property (P.B) of Brownian motion, we further think of the  $q_t$ -axis as the time axis with the original point  $p_i$ , and  $q_s$ -axis as the axis representing the value of Brownian motion.

Therefore, if  $q_t > 0$ ,  $q_s$  is considered as a sample point of the random variable which has the normal distribution with mean 0 and variance  $q_t$  as shown in Fig. 4, then we consider the point  $q$  as one of the normally distributed points after  $q_t$  time, which is randomly diffused from the original point.

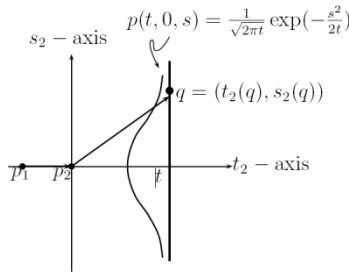


Fig. 4. The distribution of Brownian motion [3]

The value of distribution cannot directly be applied to the natural distance because the distribution changes according to the position of  $q$ . So we transform the distribution into the standard normal distribution so that the random variable  $q_s$  is transformed to the standard random variable  $\frac{q_s}{\sqrt{q_t}}$ . It is well known that the larger the value of the standard random variable is, the smaller the probability is. Therefore, the problem of determining the next point  $p_{i+1}$  is transformed into that of computing the minimum of the standard random variables. Kim et al. [3] defined a natural distance by the following manner:

$$nd(p_i, q) = q_t + K \frac{q_s}{\sqrt{q_t}}$$

where  $K$  is a constant and plays a role of a subjective weight.

It is well known that the natural distance of Kim et al. [3] marked a turning point in the research on point ordering problem. Most of the distances used in the previous point ordering research cannot give a total order to the set of points on  $R^2$ , whereas the natural distance can do that. So we could solve the problem of reconstructing curves having self-intersections. However, there are obvious shortcomings in the natural distance. The first is that the random variable  $\frac{q_s}{\sqrt{q_t}}$  used in the natural distance cannot reflect the right geometric meaning of the conditions (C-1) and (C-2). The second is that the result of point ordering is very sensitive to the value of subjective weight. The condition (C-1) shows that all points on the same line with  $p_i$  have the same value  $q_s$ , the condition (C-2) shows that all points on the same circle with center  $p_i$  have the same value  $q_t$ . So the random variable  $\frac{q_s}{\sqrt{q_t}}$  cannot resemble the right geometric meaning of conditions. In this paper, we introduce a new natural distance which may give a total order to two-dimensional points and reflect the right geometric meaning of conditions (C-1) and (C-2). First of all, we define the polar-natural coordinates as  $(r_i(q), \theta_i(q))$ , where  $r_i(q) = \sqrt{q_t^2 + q_s^2}$  and  $\theta_i(q) = \tan^{-1} \frac{q_s}{q_t}$ . By using this coordinates and understanding the geometric meaning as shown in Fig. 5, we define the polar-natural distance as

$$pd(p_i, q) = r_i(q) + K \frac{\tan \theta_i(q)}{\sqrt{r_i(q)}}$$

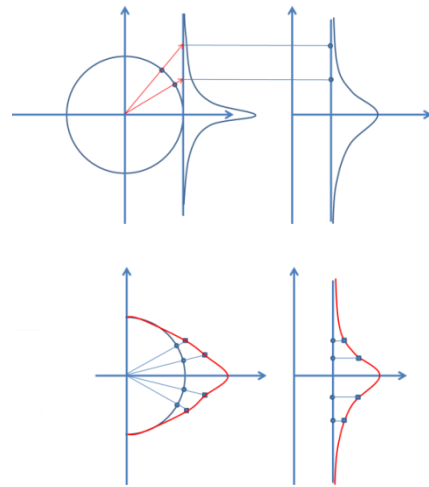


Fig. 5. The new distribution of Brownian motion based on the polar-natural coordinates

### 3. POLAR-NATURAL DISTANCE BASED POINT ORDERING ALGORITHM

In this section, we will describe a practical algorithm for ordering points in an unorganized point cloud. Assume that there are arbitrarily scattered  $n$  points, called sample points, in  $R^2$ . Denote the set of all sample points by  $P$ . The point ordering problem aims to choose a subset  $SP$  of  $P$ , where  $SP$  is a well-ordered set satisfying the natural conditions (C-1) and (C-2) mentioned in Section 2. In the above theoretical section, we have assumed that the first point  $p_1$  and the second point  $p_2$  are known. This easily enabled us to initialize the algorithm at the first point with an initial direction  $\vec{p_1 p_2}$ . In general, the result of the point ordering problem is dependent on the initial direction. But, in practice,  $p_1$  and  $p_2$  are not known. Moreover, the well-ordered subset  $SP$  may have several connected components, and may be open or closed according to the distribution of the data points. So we need to analyze the distribution of the sample points in order to find out the property of the subset  $SP$ . In general, the sample points of the unorganized point cloud for curve reconstruction may cluster in several groups and the points of each group are densely located near meaningful trajectories. So, the shape of each cluster looks like the union of bands. The objective of our algorithm is to remove outliers in each cluster and find out a well-ordered subset that plays a role of skeletons of clusters. Our algorithm consists of three major steps: data clustering, determination of the initial direction, and local point ordering.

#### 3.1 Data Clustering

Most partitioning methods cluster objects based on the Euclidean distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes. We cannot apply such clustering methods directly to our problem because the shape of

the input point set  $P$  is unknown. In this paper, we adopt the DBSCAN clustering algorithm based on the density, which is useful to filter out outliers and discover clusters of an arbitrary shape. The general idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius  $cr$ , called *clustering radius* has to contain at least a minimum number of points. By exploiting DBSCAN algorithm, we may obtain  $nc$  clusters  $CL^1, CL^2, \dots, CL^{nc}$ , in  $P$  of a band-like shape. For each cluster  $CL^j$ , we choose a point which may well represent the character of the shape of the cluster as the initial point, denoted by  $cp^j$ .

### 3.2 Determination of Initial Direction

Now, we will try to determine the initial direction of the natural distance at the initial point  $cp^j$  for each cluster  $CL^j$ . For the sake of convenience, we set the initial point as  $p_1$ . In order to analyze the distribution of points near  $p_1$ , we compute the neighborhood  $N(p_1, r)$  of  $p_1$  with a given radius  $r$ , called *ordering radius*;  $N(p_1, r) = \{p \in CL^j | pd(p_1, p) \leq r\}$ , where  $pd(p, q)$  is the Polar-Natural distance between two points  $p$  and  $q$ , which is defined in Section 2. And then we apply the PCA (Principal Components Analysis) to the neighborhood because the principal component analysis of a set of points gives us the mean, an orthogonal frame, and the standard deviation of the neighbors. Let  $q_1, q_2, \dots, q_m$  be the points in the neighborhood  $N(p_1, r)$ . We compute the mean  $m$  of the neighbors such that  $m = \frac{1}{s} \sum_{j=1}^s q_j$  and then we construct the covariance matrix. We compute the eigenvector of the covariance matrix which corresponds to the maximal eigenvalue so that it plays a role of the major principal axis in the neighborhood. Let  $\vec{ID}$  be the unit vector on the principal axis. Then the neighborhood  $N(p_1, r)$  may be subdivided into two half discs:

$$N(p_1, r) = HD_-(p_1, r) \cup HD_+(p_1, r),$$

Where

$$HD_-(p_1, r) = \{p \in N(p_1, r) | \vec{p}_1 \vec{p} \cdot \vec{ID} \leq 0, p \neq p_1\},$$

$$HD_+(p_1, r) = \{p \in N(p_1, r) | \vec{p}_1 \vec{p} \cdot \vec{ID} > 0, p \neq p_1\}.$$

If  $HD_- = \emptyset$  or  $HD_+ = \emptyset$ , then the point  $p_1$  is one of the two end points of  $SP$ . Otherwise the point is a middle point of  $SP$  so that we should apply the following one-way ordering algorithm to both  $HD_+$  and  $HD_-$  with the direction vectors  $\vec{ID}$  and  $-\vec{ID}$ , respectively. Let  $SP_+$  and  $SP_-$  be the solutions of the one-way ordering problem with the initial directions  $\vec{ID}$  and  $-\vec{ID}$ , respectively. Then the solution  $SP$  of the original point ordering problem may be obtained by merging the subsets  $SP_+$  and  $SP_-$ , i.e.,  $SP = SP_+ \cup SP_-$ .

### 3.3 Local Point Ordering

Now, we are ready to explain the one-way point ordering algorithm with the first point  $p_1$  and the initial direction  $\vec{ID}$ . First of all, we have to select the candidates of the next point

$p_2$  in the neighborhood  $N(p_1, r)$ . It is well known that the candidate set  $C_2$  is the same as the half disc  $HD_+(p_1, r)$ . So, without loss of generality, we may put  $C_{i+1} = HD(p_i, r)$ . Next, we select the second point  $p_2$  in  $N(p_1, r)$  satisfying the following condition:  $f_1(p_1) = \min_{p \in HD_+(p_1)} f_1(p)$ , where the function  $f_1(p)$  is derived from the initial direction vector  $\vec{ID}$ . We have known that the above condition satisfies the properties of conditions (C.1) and (C.2). Then the point  $p_2$  is added to the well-ordered set so that  $SP_+ = \{p_1, p_2\}$ .

In order to find out the third point  $p_3$  we compute the candidate set  $C_3$  that is the half disc  $HD_+(p_2, r)$  with the direction vector  $\vec{D} = \vec{p}_1 \vec{p}_2$ . The third point  $p_3$  should satisfy the following condition:  $f_2(p_2) = \min_{p \in HD_+(p_2)} f_2(p)$ , where the function  $f_2(p)$  is derived from the direction vector  $\vec{D}$ . Then  $SP_+ = \{p_1, p_2, p_3\}$ . We apply the above process until the candidate set  $C_{i+1}$  is empty or the intersection of  $C_{i+1}$  and  $HD_-(p_1)$  is not empty. When the algorithm meets the former condition, it generates one of the two end points for an open curve, so it has to be applied to the other region of the cluster with the direction  $-\vec{ID}$ . If the latter is satisfied, then the new obtained point  $p_{n_1}$  is near the first point  $p_1$ . We can consider the well-ordered set is a closed piecewise curve so that there is no need to continue this one-way point ordering algorithm in the cluster. The solution of the one-way point ordering problem is  $SP_+ = \{p_1, p_2, \dots, p_{n_1}\}$ . By similar process, we may get  $SP_- = \{p_{-n_2}, p_{-n_2+1}, \dots, p_{-1}\}$  as the solution of the backward one-way ordering algorithm. Now, we may obtain the first connected well-ordered subset  $SP = SP_+ \cup SP_-$ . Therefore, the global point ordering problem can be solved by applying the above process to all of clusters in  $P$ . The outline of our natural point ordering algorithm is as follows.

#### Polar-Natural point ordering algorithm {

```

P, r are given;
CL1, CL2, ..., CLnc are the clusters of P ;
SP = ∅ ;
For each cluster CLj {
  p1 := cpj as the initial point ;
  Compute N(p1, r) and  $\vec{ID}$  ;
  Divide N(p1, r) into HD+(p1, r) and HD-(p1, r) ;
  SP+ := {p1} and  $\vec{D} = \vec{ID}$ ;
  C := HD+(p1, r) ;
  While ( C ≠ ∅ ) { //Forward one-way point ordering
    Define the polar-natural distance derived from  $\vec{D}$ ;
    Find out the next point pi+1 in C ;
    SP+ := SP+ ∪ {pi+1} ;
     $\vec{D} := \vec{p}_i \vec{p}_{i+1}$  ;
    C := HD+(pi+1, r) ;
    if ( C ∩ HD-(p1, r) ≠ ∅ ) {
      C := ∅ ;
      type := CLOSED;
    }
  }
  SP- := ∅ ;
  if ( type = OPEN ) {
     $\vec{D} := -\vec{ID}$  ;
    C := HD-(p1, r);
    While ( C ≠ ∅ ) { // Backward one-way point ordering

```

```

Define the polar-natural distance derived from  $\bar{D}$ ;
Find out the next point  $p_{i+1}$  in  $C$  ;
 $SP_- := SP_- \cup \{p_{i+1}\}$ ;
 $C := HD_-(p_{i+1}, r)$ ;
}
 $SP_j := SP_+ \cup SP_-$ ;
}
 $SP := SP \cup SP_j$  ;
}
}
    
```

### 3.4 Time Complexity

The three major steps of this algorithm are to partition the point set  $P$  into several clusters, to find out the neighbor points in  $N(p, r)$ , and to select one point who has the minimum natural distance among them. The computational complexity of DBSCAN algorithm is  $O(n \log n)$ , where  $n$  is the number of the input cloud points. It is well-known that the algorithm is sensitive to the user-defined parameters. The second process of this algorithm runs in  $O(n)$  time. The process is to find out a neighborhood  $N(p, r)$  of a given point  $p$  and to apply the principal component analysis to the neighborhood in order to derive the initial direction. The third process is repeatedly to find out a candidate set  $C_{i+1}$ , and to select one point who has the minimum natural distance among the candidates. It takes  $O(kM)$ , where  $k$  is the number of the output points in  $SP$  and  $M = \max_{j=1}^n |CL^j|$ . Our algorithm is output-sensitive. However, in the worst case, the total time complexity of this algorithm may be  $O(n^2)$ , where the most of candidate sets have only one point as neighbors.

## 4. EXPERIMENTS

Our experimental results demonstrate the superiority of our point ordering algorithm based on the polar-natural distance. First of all, we compare the results of curve reconstruction for a sampled point set which are obtained by the natural distance and by the polar-natural distance.

The point set is obtained by sampling from points on two circles with different radius. Figure 6 presents two piecewise linear curves obtained by the natural point ordering method (a) and the polar-natural ordering method (b). These figures show quite different curves. The curve generated by our distance is more natural than that by the natural distance in [3]. Moreover, the polar-natural point ordering method can be used to reconstruct a curve for the sample points set containing the noise points not on the curve as well as those on it.

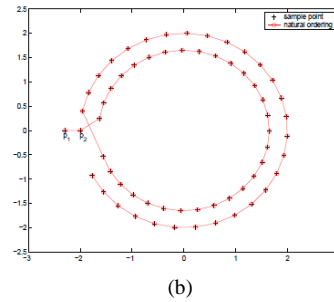
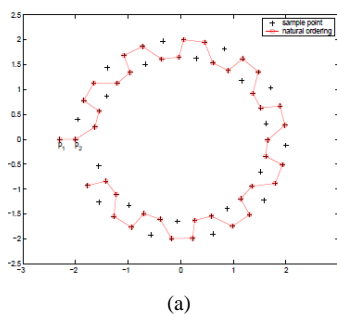


Fig. 6. The results of curve reconstruction for a sampled point set: the natural distance and the polar-natural distance

Fig. 7 shows the results of curve reconstruction for a point cloud set. The cloud point set was obtained by mouse dragging-and-dropping so that the global figure of the set has several self-intersections. Fig. 7(a) does not overcome the self-intersection problem, whereas Fig. 7(b) well reflects the global configuration of the data set.

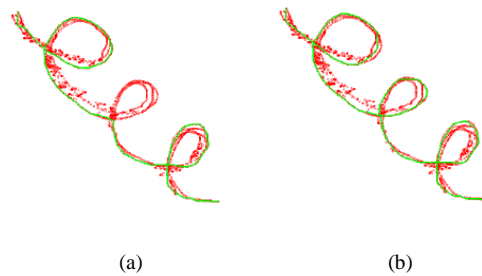


Fig. 7. The results of curve reconstruction for a point cloud set: (a) the natural distance and (b) the polar-natural distance

Fig. 8 shows the process of our polar-natural point ordering algorithm. Fig. 8(a) is the initial point set. The set was obtained by mouse dragging-and-dropping so that the global figure of the set is the union of two ‘C’-type point sets. Fig. 8(b) shows the middle state which has two curves. Each curve has two components  $SP_+$  and  $SP_-$ . The final result (c) has two curves. Since our algorithm has a special key that is the polar-natural distance it can solve the self-intersection problem. Therefore our algorithm can be applied to non-simple curve reconstruction.

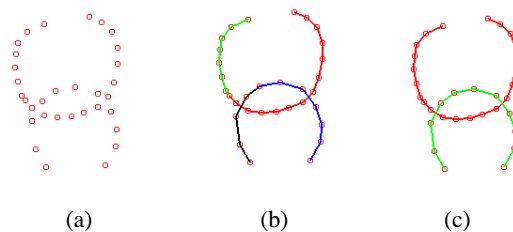


Fig. 8. The process of our point ordering algorithm: (a) point set (initial state), (b) each curve has two components (middle state), and (c) two curves (final state)

## 5. CONCLUSION

In this paper, we proposed an improved curve reconstruction algorithm by exploiting a new distance. We already introduced a natural distance in [3] which is based on a useful property of Brownian motion. The distance enables us to provide a totally order for the set of arbitrarily scattered points. However, the work of [3] missed the geometric meaning of the naturalness conditions (C-1) and (C-2). In this paper we proposed the polar-natural distance that is based on the polar coordinates. The totally ordered set generated by the polar-natural distance is more natural than that by the natural distance. By connecting these ordered points, we can obtain desired or meaningful curves. Also the polar-natural point ordering method can be used in the reconstruction of smooth curves, even though the data set contains noise points. The several experimental results show the superiority of our point ordering algorithm based on the polar-natural distance.

## REFERENCES

- [1] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface Reconstruction from unorganized points," Proc. SIGGRAPH'92, 1992, pp. 71-78.
- [2] J. D. Boissonnat and B. Geiger, "Three dimensional reconstruction of complex shapes based on the Delaunay triangulation," Proc. of Biomedical Image Process, Biomed. Visualization, 1993, pp. 964-975.
- [3] Philsu Kim and Hyoung-Seok Kim, "Point ordering with the natural distance based on Brownian motion," Mathematical problems in engineering, vol. 2010, article ID 450460, 2010, p. 17.
- [4] J. D. Boissonnat and F. Cazals, "Smooth surface reconstruction via natural neighbor interpolation of distance functions," Comp. Geom. Theo. Appl., vol. 22, 2002, pp. 185-203.
- [5] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," IEEE Trans. Information Theory, 1983, pp. 551-559.
- [6] D. Attali, "r-regular shape reconstruction from unorganized points," Proc. of 13th Ann. Sympos. Comput. Geom., 1997, pp. 248-253.
- [7] N. Amenta, M. Bern, and D. Eppstein, "The crust and the beta-skeleton: combinatorial curve reconstruction," Graphical Models, vol. 60, 1998, pp. 125-135.
- [8] T. K. Dey and T. K. Kumar, "A simple provable algorithm for curve reconstruction," Proc. 10th. ACM-SIAM Symp. Discr. Algorithms, 1999, pp. 893-894.
- [9] L. Fang and D. C. Gossard, "Fitting 3D curves to unorganized data points using deformable curves," Proc. of CG International'92, 1992, pp. 535-543.
- [10] J. P. Dedieu and C. H. Favardin, "Algorithms for ordering unorganized points along parametrized curves," Numerical Algorithms, vol. 6, 1994, pp. 160-200.
- [11] G. Taubin and R. Ronfard, "Implicit simplicial models for adaptive curve reconstruction," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, 1996, pp. 321-325.
- [12] Thanh An Nguyen and Yong Zeng, "VICUR: A human-vision-based algorithm for curve reconstruction," Robotics and Computer Integrated Manufacturing, vol. 24, 2008, pp. 824-834.



### Hyoung-Seok Kim

He received the M.S. and Ph.D. degrees in department of Mathematics from KAIST (Korea Advanced Institute of Science and Technology) in 1992 and 1998, respectively. During 1998-1999, he worked at ETRI as a post-Doctorial researcher. He is a professor in dept. of

Multimedia Engineering at Dong-Eui University from 1999. His interesting research fields are computer graphics, computer game engine, and applied mathematics.



### Ho-Sook Kim

She received the M.S. and Ph.D. degrees in department of Computer Engineering from Ewha Womans University in 1999 and 2005, respectively. She was an associate professor in division of Advance Computer Information at Dong-Eui Institute of Technology from 2001 to

2009. She is a teacher in dept. of mathematics and computer science at Korea Science Academy of KAIST from 2010. Her interesting research fields are database, data mining, GIS, and mobile technology.