Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 2239 – 2258
© TÜBİTAK
doi:10.3906/elk-1203-114

A novel dynamic bandwidth selection method for thinning noisy point clouds

Mehmet ÖZTÜRK,* Zeynep HASIRCI
Department of Electrical & Electronics Engineering, Faculty of Engineering, Karadeniz Technical University, Trabzon, Turkey

Received: 25.03.2012 • **Accepted:** 20.07.2012 • **Published Online:** 30.10.2013 • **Printed:** 25.11.2013

Abstract: In this paper, we propose a dynamic bandwidth selection method for thinning noisy point clouds into curves. Due to the nonhomogeneous distribution of noise or varying curvature of the data, the thinning procedure requires a dynamically adjusted bandwidth along the curve. On the other hand, the selected local region must be sorted along a suitable direction vector for local curve fitting purposes. The contribution of this paper to the field is 2-folded: first, a normalized eigenvalue analysis-based method is used to determine the best local bandwidth. The second task is getting a good regression line of the local region for ordering the data. In this step, a feature that is based on the Euclidean distances between points is used to determine the best regression line in the search space. Finally, a third-order parametric polynomial is fitted to the local data using the least squares method. We adapt and test our algorithm on the widely used k-nearest neighbor and constant radius methods. Our experiments show that the proposed methods automatically produce comparable results with these methods at their best parameter, which is empirically determined. Our methods also do not require any empirically determined parameter to adjust the best bandwidth or to select a good regression axis. Moreover, this method can be used with point sets that do not have a regression axis.

Key words: Curve reconstruction, eigenvalue analysis, point cloud processing, bandwidth selection, tangent estimation

1. Introduction

Curve reconstruction from point clouds plays an important role in pattern recognition, computer-aided design, image processing, and reverse engineering research areas. Due to the importance of its applications in the abovementioned areas, this subject has been focused on by many researchers in recent years.

A point cloud can be considered as a set of unorganized sampling points on a curve. That means we sample a curve with uniform or nonuniform intervals. Furthermore, these samples can contain noise. The problem is that we do not know the connections between those sample points. Hence, with the curve reconstruction, we try to connect adjacent points to each other in a most suitable way.

Previous studies are classified according to 4 fundamental properties of the curves. The first property is the sampling way of the curve, nonuniform or uniform, which is shown as in Figures 1a and 1b, respectively. Via uniform sampling, we take equally spaced samples along the curve. On the other hand, if we sample the curve nonuniformly, the distance between the samples could be different.

For the second property, a curve can be open or closed. If a curve has 2 distinct starting and end points, it is an open curve (Figure 2a). If the curve is closed, there are no starting or end points (Figure 2b).

The third property is the smoothness of the curve, which means a continuous curve (no gaps or dis-

*Correspondence: mehmetozturk@ktu.edu.tr

continuities) with no corners (no abrupt changes in the slope at any point, such as you would get from the intersection of 2 lines) (Figures 3a and 3b).

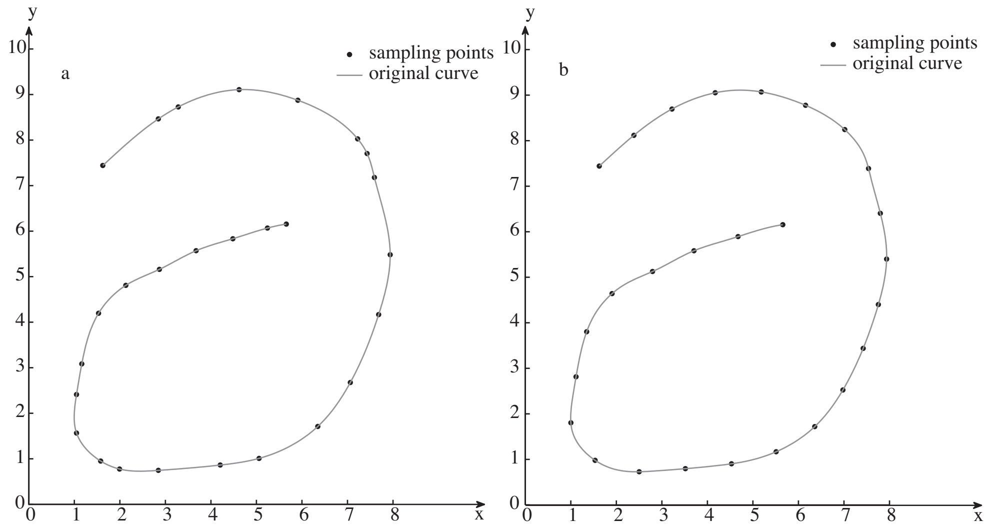

Figure 1. Examples of nonuniform (a) and uniform (b) sampling.

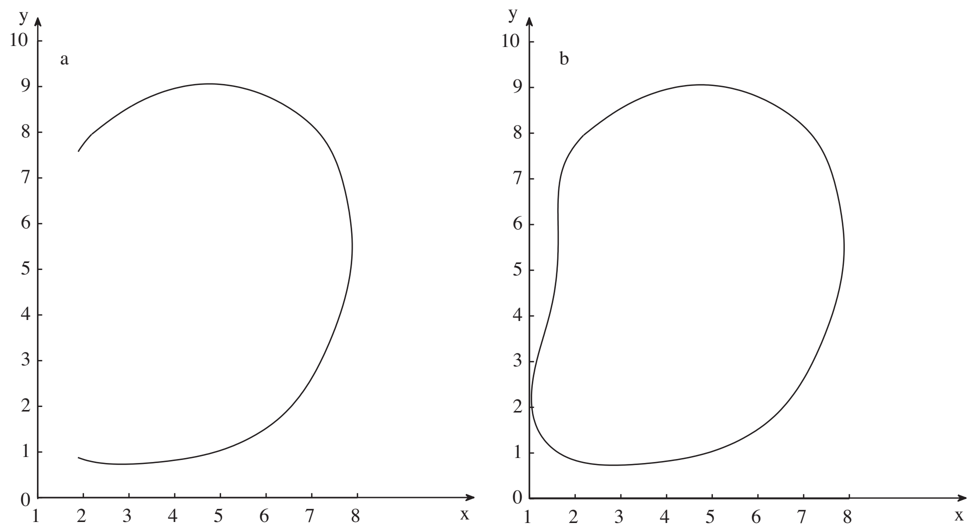

Figure 2. Examples of an open curve (a) and a closed curve (b).

For the fourth property, we can divide the curves into 2 groups as nonsimple or simple curves, with examples shown in Figures 4a and 4b, respectively. A simple curve can be described as a curve without any branches or self-intersections.

In the literature, previous studies dealing with point clouds can be divided into 2 main groups according to whether the point clouds contain noise or not.

1.1. Studies about point clouds without noise

There are a few algorithms on uniformly sampled curves. Edelsbrunner et al. [1] analyzed and generalized the convex hull of a finite set of points in the plane. They introduced the alpha-shapes concept. The alpha-shapes

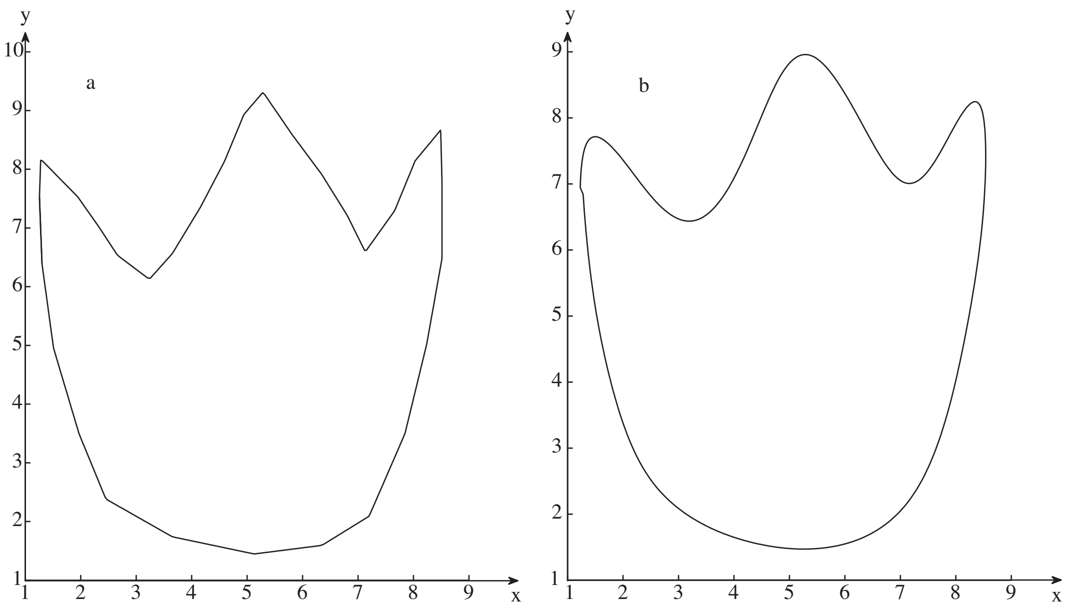

Figure 3. Examples of nonsmooth (a) and smooth (b) curves.

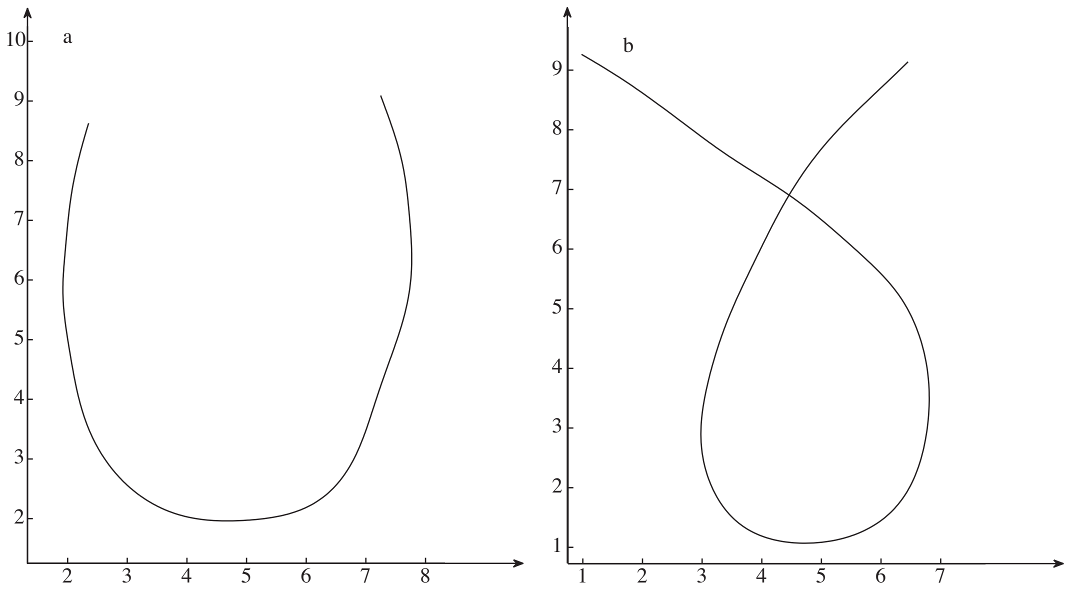

Figure 4. Examples of simple (a) and nonsimple (b) curves.

consist of straight-line graphs that are generalizations of the convex hull and these alpha-shapes seem to capture the notions of the fine shape and crude shape of point sets. They proved that the alpha-shapes are subgraphs of the closest point or furthest point Delaunay triangulation. The developed algorithm that constructs alpha-shapes has optimal complexity in O(n log n). Attali [2] dealt with the problem of reconstructing a surface from a set of scattered data points. A formulation of the reconstruction problem was mathematically defined as a particular mesh of the surface. This is called normalized mesh, which is included inside the Delaunay graph. For 2D points that are sampled from an r-regular shape, they proposed a criterion to select boundary faces inside the Delaunay graph and to provide the exact solution. Because their solution cannot be extended to 3D, they suggested some heuristics in order to complete the surface. Figueiredo and Gomes [3] utilized Euclidean minimum spanning trees (EMSTs) to reconstruct differentiable arcs from sufficiently dense samples. They also proved the correctness of this reconstruction with the help of combinatorial characterization of minimal spanning paths and a description of the local geometry of arcs inside tubular neighborhoods. For more general curves, they also presented simple heuristics.

Some researchers proposed different methods for uniformly sampled, simple closed curves [1–4]. Bernardini and Bajaj [4] used alpha-shapes for curve reconstruction problems. The alpha-shapes give a formal characterization of the reconstruction problem. They proved that the reconstructed alpha-shape is homeomorphic to the original object and represented it within a fixed error bound in the case of satisfaction of certain sampling requirements. They also described a simple automatic selection method for the alpha value. Moreover, they dealt with nonideal scans and fit a piecewise algebraic smooth surface to the data points.

There are several studies about nonuniformly sampled curves [5–10]. Amenta et al. [5] tried to solve the curve reconstruction problem by constructing a graph on a planar point set. The principle point of the proposal was that if a curve is smooth and sampled densely enough, the graph on the samples is the polygon generated from the curve. Hence, this graph grabs the shape of this point set. The sampling density requirement depends on the local feature size. If the curve has a lower local curvature, the point set can be sampled less densely. They suggested 2 different graphs. Dey and Kumar [6] presented an algorithm to reconstruct a smooth closed curve from point clouds without noise. Their algorithm is based on the framework introduced by Amenta et al. [5]. The algorithm has the properties of being simple, requiring a sampling density better than previously known, and having a straightforward adaptivity for higher dimensional curve reconstruction problems. Dey et al. [7] proposed an algorithm that constructs a polygonal reconstruction from any set of input points that is not a dense sample of the closed smooth curve. According to Althaus and Mehlhorn [8], the problem of curve reconstruction is a finite sample of unknown curves [8]. To solve this problem, one has to connect the points in the unknown curve in the order in which they lie. They extend the traveling salesman tour of the input, which solves the reconstruction problem under fairly weak assumptions along 3 dimensions. While they generalized the assumptions and gave an alternate proof, they also showed that the traveling salesman tour can be constructed in polynomial time in the context of curve reconstruction. Dey and Wenger [9] proposed a new algorithm that can deal with nonsmooth curves with multiple components. They showed the effectiveness of their algorithm by conducting experiments on much input data. This algorithm has an advantage of extendibility to 3 dimensions for surface reconstructions. Dey and Wenger [10] also presented an algorithm to reconstruct a simple, smooth, closed piecewise curve from a set of n sample points in O(n log n) time. The sampling of the points can be uniform or nonuniform. However, they proved the correctness of their algorithm assuming certain sampling conditions. The sampling conditions were based on the minimum angle made by tangents at any corner point.

In the literature, there are also some studies for nonuniformly sampled simple closed curves [5,6,11]. In the study by Gold [11], his purpose was to extract the topology from scanned maps. A similar study was done previously by Gold et al. extracting a skeleton from a Voronoi diagram [12]. However, this process requires vertex labeling and was only suitable for polygon maps. In order to extract the skeleton from unlabeled vertices, they used the crust algorithm proposed by Amenta et al. [5]. They managed to extract both the crust and the skeleton simultaneously by reducing the algorithm to a local test on the original Voronoi diagram. Finally, with the help of various cartographic applications, they illustrated their work's utility and also showed that their proposal is equivalent to the original algorithm.

For nonsmooth curves, there are few studies in the literature [9,10,13]. Funke and Ramos [13] proposed an algorithm to reconstruct curves that could contain corners and endpoints from a point cloud under a certain sampling condition. The output of their algorithm is a polygonal reconstruction of the curves. They proved that the reconstruction contains the edges of the correct reconstruction. They showed that for a collection of curves with similar sampling conditions, they could reconstruct exactly the correct version of the curve with a little adaptation of their algorithm. They also applied the algorithm in practice.

In their study, Zeng et al. [14] proposed a parameter-free algorithm that they called DISCUR to reconstruct curves from unorganized samples. This algorithm is for simple curves. The points can be sampled from curves that may be open/closed and with/without sharp corners. They took their inspiration from the human visual system to get a criterion for curve reconstruction: 1) the 2 closest neighbors tend to be connected and 2) the sampling points tend to be connected into a smooth curve. They proposed a statistical criterion to determine in which case 2 sample points should not be connected even if they are the nearest neighbors. To be able to use their proposed algorithm, they suggested a necessary and sufficient condition for the sampling of curves. The effectiveness of their algorithm was shown on a large number of examples. A curve reconstruction method from unorganized point clouds with noise was proposed by Kim and Kim [15]. Due to the importance of ordering in curve reconstruction, this method uses a property of a 1D Brownian motion: 'natural distance' to order the representative points. Their algorithm is able to reconstruct both simple and nonsimple curves with the help of their proposed ordering method, which reflects the smoothness and nearness of the points. The effectiveness of their algorithm was tested on numerous examples. However, the success of this method is sensitive to the initial point and initial direction.

Table 1 shows a classification of previously mentioned curve reconstruction algorithms and their properties. The strengths and deficiencies of curve reconstruction algorithms were summarized by Dey [16]. According to Dey, all of the presented methods in the table assume that the point sets must be sampled from nonself-intersecting curves.

Table 1. The requirements of some curve reconstruction algorithms [16].

Algorithm	Sample	Smoothness	Boundary	Components
$\alpha-$Shape	Uniform	Required	None	Multiple
r-Regular shape	Uniform	Required	None	Multiple
EMST	Uniform	Required	Exactly 2	Single
Crust	Nonuniform	Required	None	Multiple
Nearest neighbor	Nonuniform	Required	None	Multiple
Traveling salesman	Nonuniform	Not required	Must be known	Single
Conservative crust	Nonuniform	Required	Any number	Multiple

1.2. Studies about point clouds with noise

The previous works about curve fitting to noisy point clouds are limited. Cheng et al. [17] used a noise model that assumes sampling the curve with a locally uniform distribution, followed by a uniform perturbation in the normal directions to reconstruct a collection of disjoint smooth closed curves from noisy samples. The true reconstruction probability approaches 1 as the sampling density increases. Lee [18] presented a 2-step algorithm: first, the noisy point samples were reduced to a thin curve using the moving least squares method, and he then approximated this thinned point set with simple curves without self-intersections. In his paper, he proposed a technique to improve the moving least squares method using the EMST region expansion strategy. In their paper, Taubin and Ronfard [19] proposed simplicial models, which are curves and surfaces defined by piecewise linear functions. They used simplicial models against parametric deformable models that require previous knowledge of the topological type of the data, a good initial guess for the data, and expensive computations to check for and prevent self-intersections while tracking deformations. However, simplicial models allow for local deformations, control of the topological type, and the prevention of self-intersections. They applied their method on the curve reconstruction problems of 2D point clouds. With the help of an adaptive space subdivision approach, they estimated the geometry, the number of connected components, and the topology of the data.

2. Motivation of this study

The fundamental basis of most of the practical curve reconstruction algorithms is local polynomial fitting or its variants. The success of these algorithms significantly depends on how to select and order the representative points. The previous works about ordering the noisy point clouds are limited [15]. The first step in the studies related to noisy point clouds is the thinning stage, which reduces the point cloud to a thin curve [18]. This stage requires the calculation of the dynamic bandwidth of the selected local region of the noisy point cloud. Bandwidth is the ratio of the number of selected points to the total number of points. On the other hand, studies about bandwidth selection on the local region of point clouds are under development. In addition, to the best of our knowledge, there has been no work conducted with determination of the best local regression axis to sort the locally selected points. By determining the sorting direction, if the point cloud is sampled from a curve that can be defined only with a regression axis, the curve reconstruction problem can be reduced to ordering the points according to this regression axis and defining this point cloud with a spline. Most of the commercial programs dealing with such problems require a user-defined regression axis or vector.

3. The proposal

In this paper, we will initially propose a dynamical bandwidth selection method that adjusts the bandwidth according to the local properties of the data. We have improved our previous work [20] by using a feature to be able to determine a good ordering direction for sorting the locally selected data and fitting a curve to estimate the correct position of a point. In our study, we will use noisy and unordered point clouds lying on a curve with no self-intersections. To create a connection table between the points, the EMST method is utilized. This method was proven to be able to exploit the underlying shape of an unordered point set under some conditions such as, for example, dense enough sampling [3]. However, if the point cloud is corrupted with noise, the EMST output does not yield a meaningful connection table. Thus, before using this method, the point cloud must be thinned into a curve-like shape. As mentioned above, thinning the point clouds requires some local data processing steps. To automatically determine the local region bounds, we propose a dynamic bandwidth selection method. The result of thinning the noisy points is a dense unordered point set that is suitable for using the EMST to reconstruct a connection table final curve.

3.1. Dynamic bandwidth selection

The nonuniformity in the distribution of the noise or a changing curvature along the point clouds, as shown in Figure 5a, requires a dynamic adjustment of the local region radius or bandwidth. While the standard deviation of the noise or the thickness of the point cloud is small, relatively fewer points will be sufficient for a good polynomial fitting. Thus, we need a smaller radius or short bandwidth. On the other hand, when the thickness increases with the standard deviation, which means that the disruptive effect of the noise is higher, then we will need more bandwidth or more points to tolerate the high noise effect. The bandwidth must also be inversely proportional to the degree of curvature, as shown in Figure 5b. While the curvature increases, the bandwidth must be getting smaller, because the local region linearity decreases rapidly. We exploit the property of local linearity in the curve through local eigenanalysis.

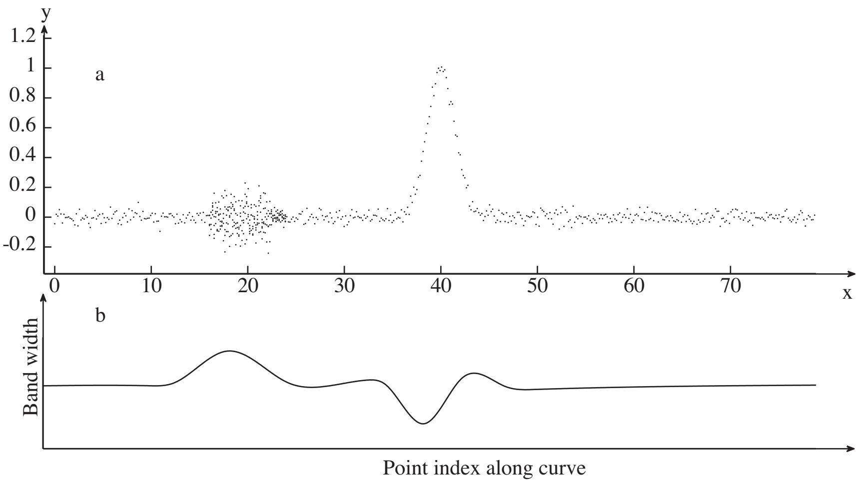

Figure 5. A point cloud with changing curvature and thickness (a) and its relatively optimal bandwidth (b) along the cloud.

3.2. Eigenanalysis

Two major problems that must be considered in the curve reconstruction process from point clouds are: 1) selection of the local region and 2) definition of the local regression line or ordering of the points. For the second problem, one technique to estimate the direction of the local tangent or local regression line at a given sample point on a point cloud is to look at the shape of a scatter matrix computed using points in its neighborhood. If the curve is smooth, it is reasonable to expect that the scatter matrix will be elongated and that its major axis, or principal eigenvector, will approximate the direction of the local tangent for some appropriate range of neighborhood sizes (Figure 6a). However, this estimation does not always give the correct tangent vector to sort the data. For some degenerate data, such as, for example, on a high curvature section, as shown in Figure 6b, we utilize another approach to be able to determine a suitable sorting direction. This approach will be explained later. On the other hand, aside from the eigenvectors, the eigenvalues can give valuable information about the elongation of the data. We must measure the elongation so that we are able to make a decision about local linearity. The key point in this decision is to select the neighbor points until we get a certain linearity value.

The eigenanalysis is based on the covariance matrix of the selected points. Covariance is a measure of the extent to which corresponding elements from 2 sets of data move in the same direction. The eigenvectors of this matrix give us this direction (Figures 6a and 6b).

Consider an $n \times m$ matrix D, in which n and m are the number of observations and the number of independent variables, respectively. The covariance matrix of D will be a $m \times m$ matrix. An eigenanalysis of this covariance matrix gives us m orthogonal vectors and m positive constants, which are named eigenvectors and eigenvalues, respectively. The eigenvalues will be positive because the covariance matrix is positive definite since negative variances are not defined. The eigenvector that corresponds to the maximum eigenvalue is the principal direction of the data contained in matrix D.

The eigenvalues also give us detailed information about the directionality of the data. Consider circular point data, as shown in Figure 7a. Due to the circular point data, which are not elongated in any direction, the orientation of the eigenvector that corresponds to the maximum eigenvalue is uncertain or will not make much sense. If the data have a distribution, as shown in Figures 7b and 7c, the eigenvector will have a unique direction.

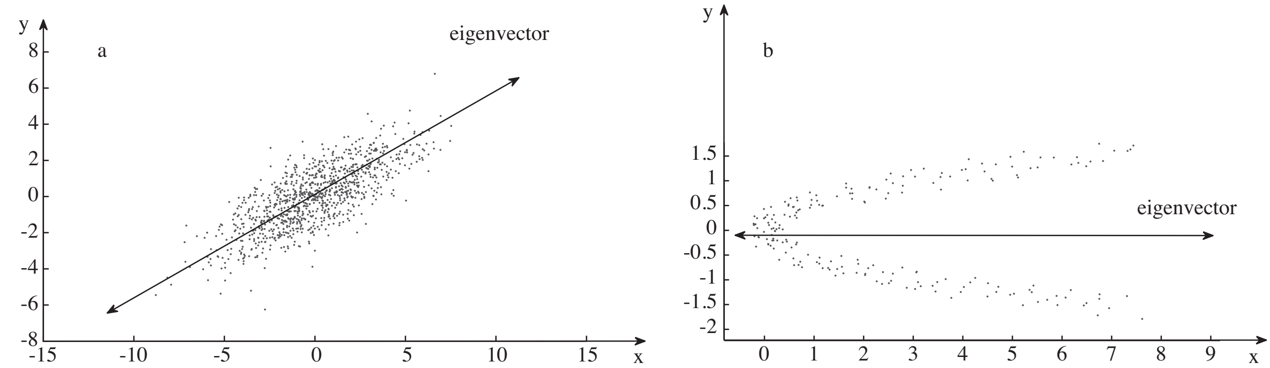

Figure 6. A smooth point cloud and its eigenvector (a) and a nonsmooth high curvature point cloud and its eigenvector (b).

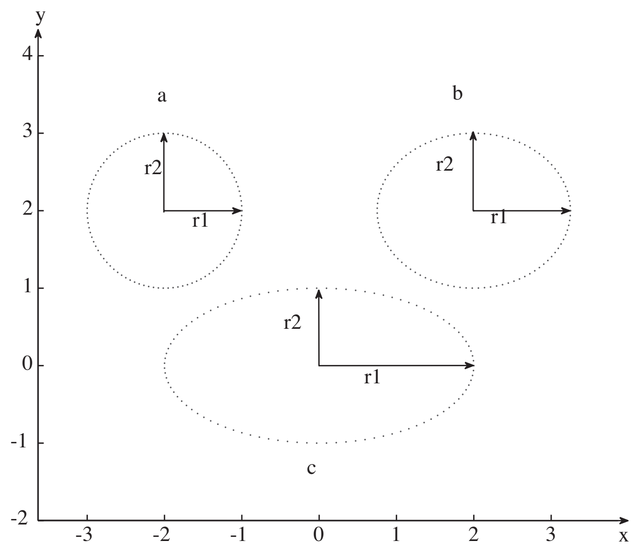

Figure 7. Data with no unique direction (a) and data that have directionality (b, c).

The maximum eigenvalue can be considered as a measure of the directionality of the data. Since these eigenvalues depend on the scale of the data, this parameter is necessary but not sufficient alone to determine the directionality. Thus, the rate between the eigenvalues is independent from the scale; a normalization step is used to extract this rate. If one considers the eigenvalues as the amplitudes of an $m-$dimensional vector, it is possible to make normalization as in Eq. (1):

$$\lambda_n = \lambda_n / \sqrt{\sum_n \lambda_n^2} \quad n = 1, 2, ..., m. \tag{1}$$

The maximum eigenvalue obtained with this normalization will be between $1/\sqrt{m}$ and 1. The minimum value of the maximum eigenvalue depends on the dimensions of the data. Our goal here is getting a scale between

0 and 1 that is independent from the dimension. To transform from $\left[\frac{1}{\sqrt{m}}\ 1\right]$ to [0 1], Eq. (2) is used. If the normalized maximum eigenvalue is represented with λ_{max}, its final form can be obtained as in Eq. (2).

$$\lambda_{\max} = (\lambda_{\max} - \frac{1}{\sqrt{m}})/(1 - \frac{1}{\sqrt{m}}) \tag{2}$$

Table 2 shows the eigenvalues before and after the normalization process of the data shown in Figures 7a, 7b, and 7c, respectively.

Table 2. Raw and normalized maximum eigenvalues of the data in Figures 7a, 7b, and 7c.

	Maximum eigenvalue	Normalized maximum eigenvalue
a ($r_1/r_2 = 1$)	0.7141	0.0238
b ($r_1/r_2 = 1.25$)	0.8471	0.4778
c ($r_1/r_2 = 2$)	0.9713	0.9018

The regression line obtained for a point cloud, as shown in Figure 8a, does not make much sense. The validity of the obtained regression line can be measured with the normalized maximum eigenvalue. Figures 8a, 8b, and 8c and Table 3 present how the selected points can be better represented with a line. The normalized maximum eigenvalue is much closer to 1. A value of 0.9 is selected as a threshold because at this threshold value, the data are elongated enough along a direction. This threshold value is used for the determination of the local neighbor radius or bandwidth.

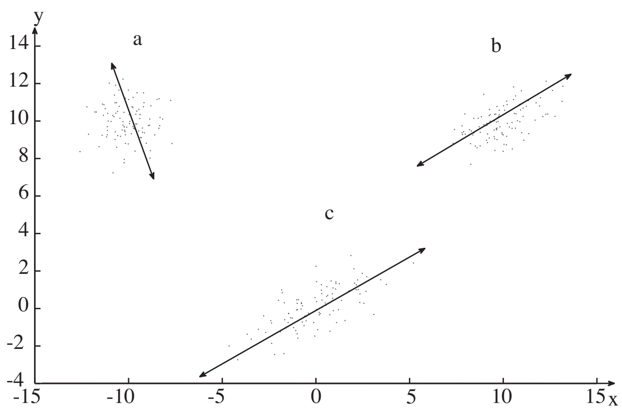

Figure 8. A point cloud with no unique direction (a) and a point cloud that has directionality (b, c).

Table 3. Raw and normalized maximum eigenvalues of the point cloud in Figures 8a, 8b, and 8c.

	Maximum eigenvalue	Normalized maximum eigenvalue
a	0.7338	0.0912
b	0.8775	0.5819
c	0.9681	0.8910

In this study, we were inspired by the k-nearest neighbor (KNN) algorithm to select the local points. As is known, in the KNN algorithm, the k value increases sequentially until we reach a certain value. However, in our method, we stop increasing the k value when the normalized maximum eigenvalue is greater than or equal to 0.9.

Figure 9 shows the point selection process with an approach of changing the radius of a circle step by step and taking the points inside that circle. The change in the normalized maximum eigenvalue can be seen in Figure 10. The top-left circle in Figure 10 indicates a situation where the selected points form a line because of an insufficient number of points (usually 2 or 3), and this situation is not a case of suitable bandwidth.

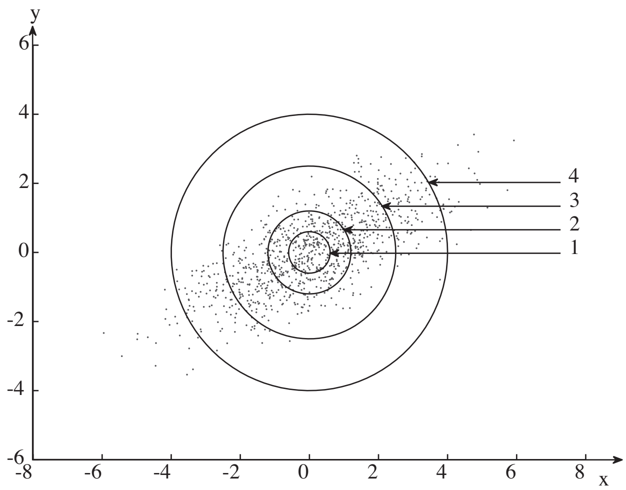

Figure 9. Some selection radii of the local region.

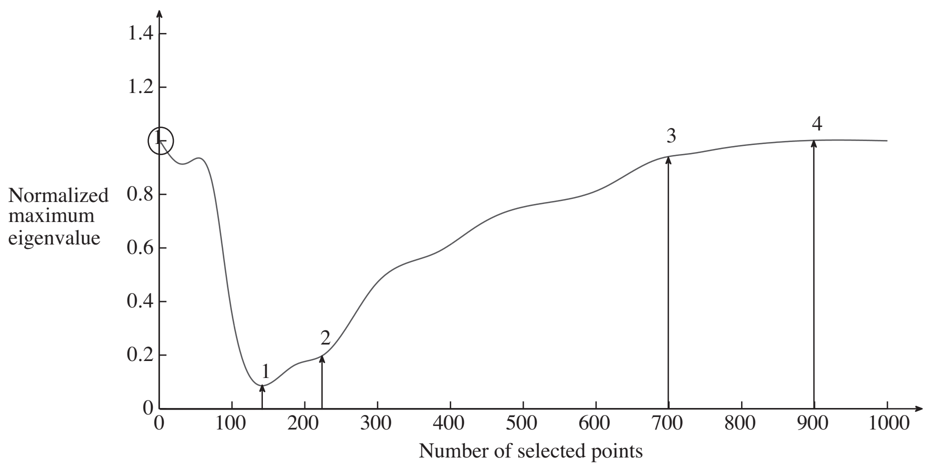

Figure 10. Variation of the normalized maximum eigenvalue with the local neighbor radius.

3.3. Ordering the points

Ordering the point clouds is one of the major steps in the field of curve reconstruction. If the ordering is known, the point clouds could be modeled with a parametric approach for every dimension and a spline could also be fitted to the ordered point data set. A simple way to be able to order data for curve fitting purposes is to determine the regression vector. The common approach to obtain the sorting direction is an eigenvector analysis. This method gives the only direction in which the maximum variance can be obtained. However, the regression vector calculated from eigenvectors is not always the right vector to order the point cloud data. To obtain the best regression axis for sorting the locally selected data, samples from all of the possible regression vector spaces are tested for a feature that determines the suitability for ordering. This feature is the mean of the Euclidean distances of all of the successive points for that sample vector, and it is called the interpoint distance. The idea behind this feature is that it is expected from the point to be connected to one of the nearest points for the best regression axis [15]. In the overall case, for a good regression axis, the interpoint distance should be minimal among the other axes. In Figure 6b, a test point cloud can be seen. This point cloud is chosen because its eigenvector or maximum variance analysis gives a wrong regression axis. The true regression axis is the one that corresponds to the y axis in Figure 6b. It is clearly seen in Figure 11 that the interpoint distance feature can give us this regression direction.

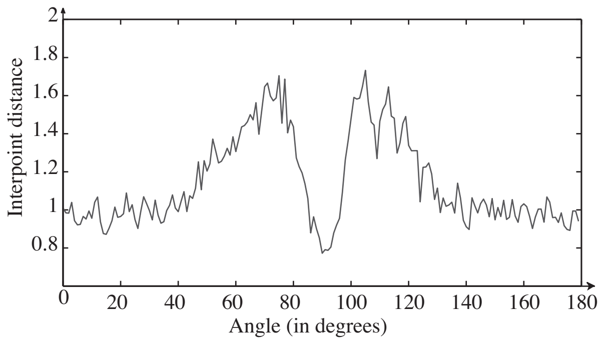

Figure 11. The interpoint distance values for the data shown in Figure 6b. It is clear from the figure that the best regression axis is the one that makes about a 90° angle from the x axis.

For 2D data in the xy plane, the regression vector space can be constituted by rotating a vector 180° around the z axis. In this study, a vector space was sampled uniformly in 2 dimensions with 1° intervals. On the other hand, for 3D data, one needs to sample a sphere uniformly. For this purpose, the golden section spiral method is utilized to generate a set of evenly distributed points on a unit sphere.

While sampling the vector space, for each sample, all of the data are orthogonally projected on this test vector. This process also gives the Euclidean distances of the projected points that are calculated according to a reference point on the test vector. Data ordering for this test vector is done by sorting these distances. Figure 12 shows this process, where the data points are orthogonally projected on a test regression line vector and sorted according to the distances $d_i, i = 1, 2, 3, 4$.

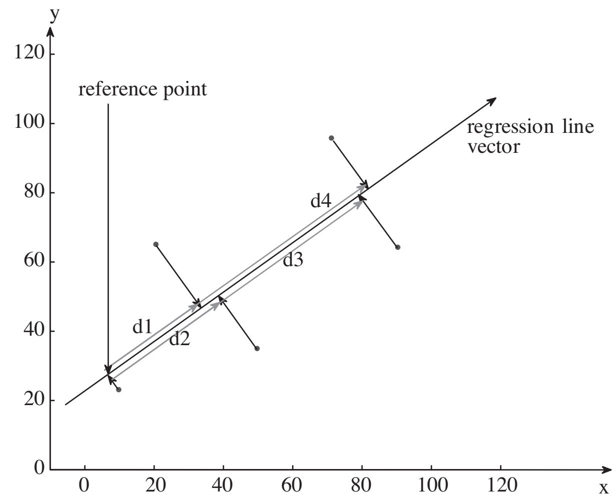

Figure 12. Perpendicular projections and the distances of the points on a regression line.

3.4. Postprocessing of the thinned points

We use the EMST algorithm to organize the points to form a connected set. The EMST algorithm is a minimum spanning tree of a set of n points in N-dimensional space, where the weight of the edge between each pair of points is the Euclidean distance between those 2 points. Simply, an EMST connects a set of points using lines in such a way that the total length of all the lines is minimized and any point can be reached from any other by following the lines.

After this process, we can order all the data using the connections from the EMST. The ordered data are now suitable for arc-length parameterization. This means that we can now determine the location of a point according to the distance from a reference (starting or ending) point. Hence, we utilize the arc-length parameterized least squares smoothing spline method to construct the final smooth curve for all of the data.

3.5. Method overview

In this section, the proposed method overview is given as shown in Figure 13. Figures 13a, 13b, 13c, and 13d explain some steps from the local region selection. The circles in Figures 13a, 13b, 13c, and 13d show the growing of the selected points around the processed point. Actually, the algorithm does not select the points inside the circle; however, this circle only gives information about the point selection strategy. This strategy is to select a nearly circular region. As we mentioned in Section 3.2, a normalized eigenvalue gives us insight about the directionality of the local region. If the data have normally distributed noise, the local region selection process will continue until reaching the point cloud thickness, as shown in Figure 13b, because this selection process corresponds to a circular region and the normalized eigenvalue remains close to zero. After determination of the point cloud thickness, the selected points will not form a circular shape anymore. Hence, the algorithm continues to expand the local region until the normalized eigenvalue reaches a threshold greater than 0.9, as shown in Figures 13c and 13d.

The next step after the selection of the local region is to determine the local regression axis. The purpose here is to get a sorted set of points from an unstructured set. The locally selected points will have a direction determined from its eigenvector. Generally, this direction is a good candidate for the regression axis, like in Figure 13e. However, because of selecting points locally, the eigenvector of some regions gives an incorrect regression axis, like in Figure 6b. Hence, we propose a feature to be able to determine the correct regression axis in every situation. This feature is called the interpoint distance and a detailed explanation is given in Section 3.3. A regression axis is simply a line, and for the determination of a line, a direction and a point that this line passes must be known. In the algorithm, this point is calculated from the mean of the data that are shown in Figure 13e as the intersection of the test lines. The direction of this test line is sampled from all of the possible regression axis spaces. Figure 13e shows only 3 examples of this test line. During the determination of the best regression axis, the algorithm orthogonally projects the selected points on each test line and sorts the points according to their positions on the line (Figure 13f). The interpoint distance is calculated after the sorting process and the line that has the minimum interpoint distance value is selected as the best regression axis.

After this step, the data are suitable for the least squares curve fitting process, as shown in Figure 13g. We use a third-order polynomial to approximate the data set.

All of the steps that are mentioned above are repeated for all of the points in the point cloud. This gives the thinned point cloud shown in Figure 13h.

The final step is to reconstruct a curve from the thinned points. With the help of the EMST algorithm, we organize the points to form a connected set. Since the points are thinned enough, the EMST will connect all

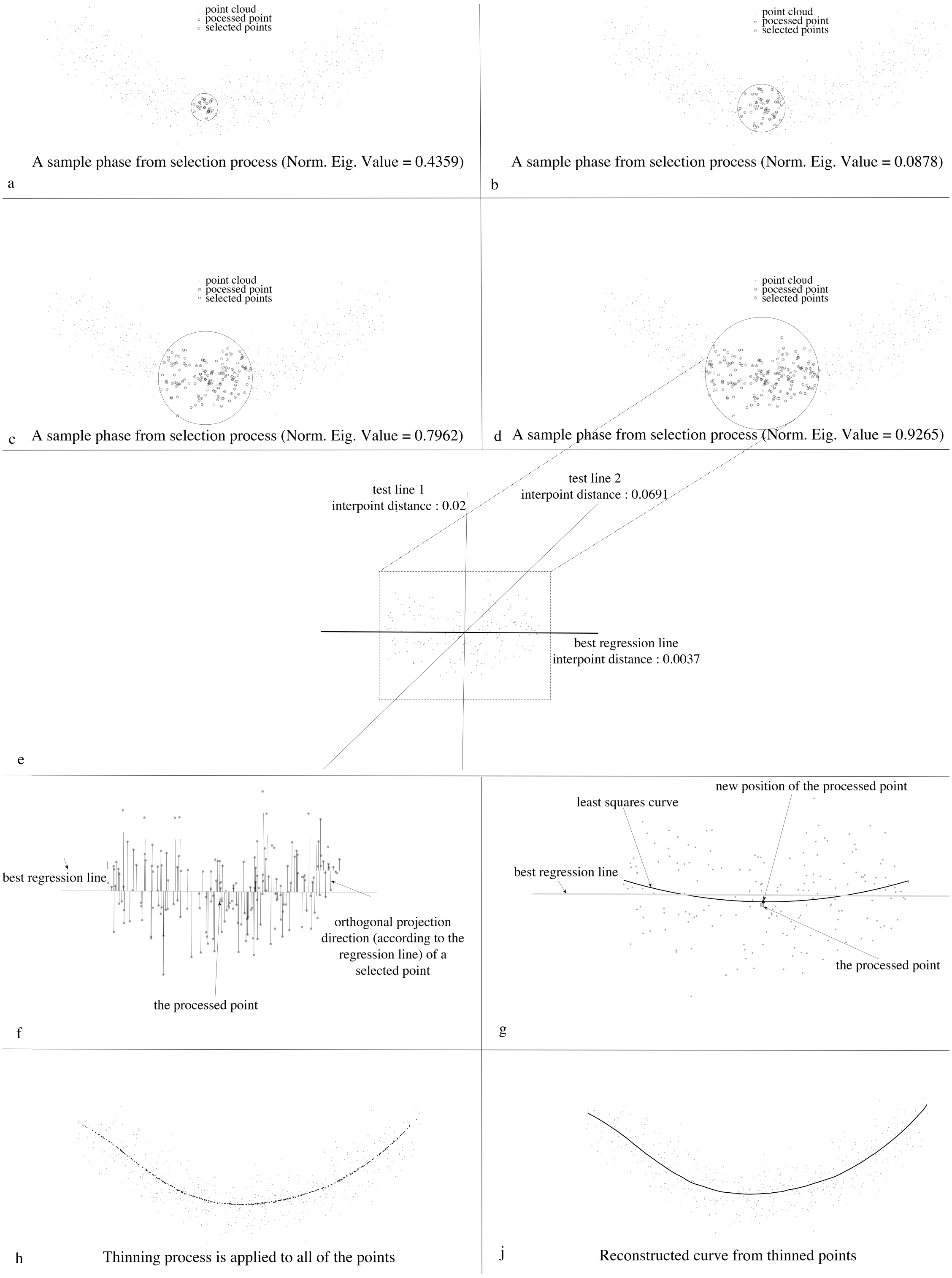

Figure 13. The proposed method steps: selection of the local region step by step (a, b, c, d); determination of the best sorting direction (e); sorting process of the selected points on the regression line (f); least squares, curve fitting, and repositioning of the processed point (thinning) (g); thinned version of the data (h); and reconstructed curve from the thinned data (j).

of the points consecutively. After pruning the unwanted branches, we obtain a connected set that is suitable for spline fitting or any operation for curve reconstruction. In Figure 13j, the result of the spline fitting is shown.

4. Experimental results

In this study, our first aim is to indicate the necessity of using dynamic bandwidth. For this reason, we conduct 2 different tests. The first regards the data that have a homogeneous noise distribution and the other one is about the data that have a nonhomogeneous noise content. We compare our methods with 2 heuristic approaches, constant radius and constant k for KNN, for bandwidth selection. In our tests, we determine the parameters of these methods by trial and error to get the best fit. We present the mean squared error (MSE) that is calculated from a real function as a quantitative result. We assume that we already know the regression axis in experiments 1, 5, and 6, and this axis corresponds to the x axis.

All of the data used in these experiments are generated synthetically with the corresponding formulas given below and a Gaussian-type zero-mean noise is added to every function. The standard deviation of the additive noise is also given.

The first experiment is shown below in Figure 14. The data of this experiment are generated using the following function: $t = \left(-\frac{3\pi}{2}, \frac{3\pi}{2}\right),\ x = t,\ y = \cos(2t)$.

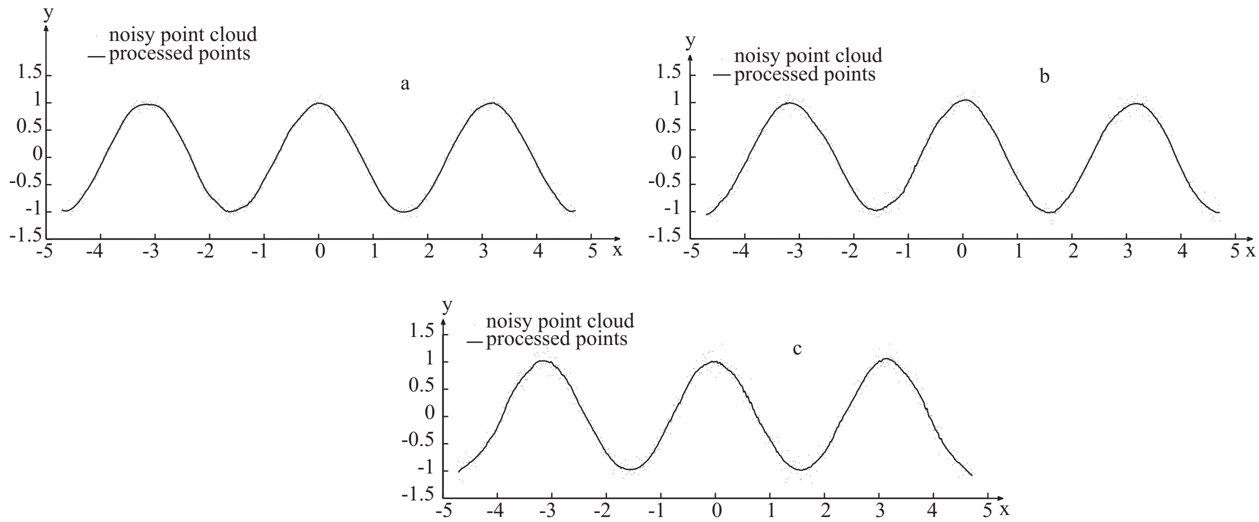

Figure 14. The results of the proposed method for the variable standard deviation of the noise: $\sigma = 0.0625$ (a), $\sigma = 0.1$ (b), and $\sigma = 0.125$ (c).

The results show that even though the noise increases gradually in Figures 14a, 14b, and 14c, the proposed method gives acceptable outputs, especially at the regions with a high curvature. The noise standard deviation and corresponding MSE are given in Table 4 for the data shown in Figures 14a, 14b, and 14c. The MSE is calculated by subtracting the fitted curve from the original data.

Table 4. The standard deviation of the additive noise and corresponding MSE of the experiment from Figure 14.

Data set (500 points)	Noise standard deviation	MSE
Figure 14a	0.0625	0.0040
Figure 14b	0.1000	0.0080
Figure 14c	0.1250	0.0110

The second experiment is conducted on a hand-made point cloud bearing a high curvature and variable point cloud thickness effects. The data are actually 2D, so one can assume an ordinary least squares technique to fit a suitable function for such 2D data. This is not possible because such data shown in Figure 15 cannot be formulated as a function of the x axis. Splines can be a solution if the data have suitable ordering, but in this case, the connections between the points are random. Hence, the most convenient way to fit a curve is when the data have to use local regression lines calculated in a suitable local region, like in this proposed method.

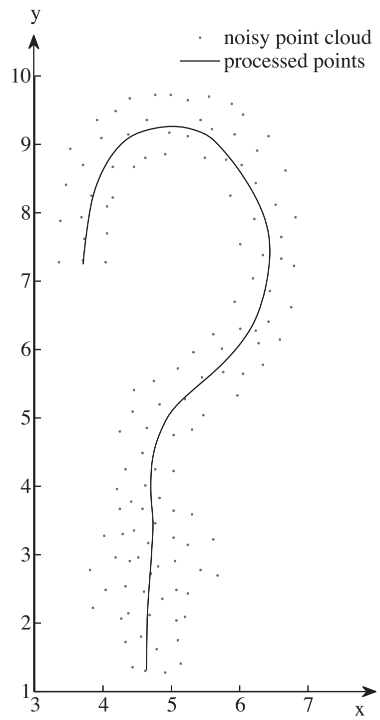

Figure 15. The result of the proposed method for the variable curvature and point cloud thickness.

The third experiment is shown in Figure 16. The function used to generate the data is:

$$t = \left(-\frac{3\pi}{2}, \frac{3\pi}{2}\right), \quad x = \cos(t), \quad y = \sin(t).$$

These data are actually in the form of a circle that has a unit length radius. The standard deviation of the additive noise changes linearly in [0, 0.05] with the angle. Thus, the bandwidth has to change accordingly with the angle. The result of the curve is almost a circle, as shown in Figure 16, due to the dynamically changed bandwidth of the proposed method.

Table 5 shows the quantitative results of the proposed method for the data shown in Figure 16. The MSE of the constant radius is lower than that of our proposed method. However, it is emphasized that the parameters of those methods are chosen so as to give the best results, while the proposed method uses the default value of 0.9. If one wants to get the best result using the proposed method, the threshold for the normalized maximum eigenvalue should be chosen as 0.95. As a result, we can achieve reasonable results without adjusting any parameters.

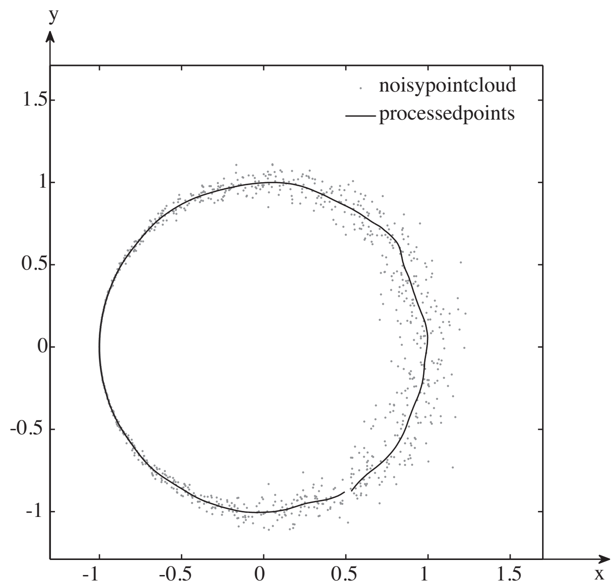

Figure 16. The result of the proposed method for the point cloud thickness and bandwidth.

Table 5. Comparison of the KNN and constant radius methods with the proposed method. The best parameters used in this experiment are also given, as well as the MSE of the experiment from Figure 16.

Used method	Parameter	MSE
KNN	K = 110 (empirically determined)	0.0398
Constant radius	R = 0.5 (empirically determined)	0.0397
Proposed method	Thr = 0.9 (default)	0.0400
	Thr = 0.95 (empirically etermined)	0.0398

The fourth experiment is shown in Figure 17. These are 3D data, generated with the following function, and the additive noise has a value of sigma 0.05:

$$t = [-1, 1], \quad x = t, \quad y = t, z = t^3.$$

In this experiment we apply our algorithm to higher dimensional data. In the experiment shown in Figure 17, we suppose that we do not know the regression axis to sort the data. Hence, we use the proposed regression line determination technique.

The last 2 experiments are conducted for comparison purposes of our method with the KNN and constant radius methods on different data sets. The first data set, which is used in experiment 5, includes high curvature sections only. On the other hand, the second data set (used in experiment 6) has both high curvature and variable point cloud thickness regions. The x axis is used as a regression line in these experiments. Hence, we can compute the MSE by subtracting the fitted curve from the original data.

In experiment 5 (Figure 18), we use 500 points in [0, 2] for the parameter p. Figure 18a shows the result of the proposed method, while Figures 18b and 18c are for KNN and constant radius, respectively. The used function is as below. The MSE results of this experiment are summarized in Table 6.

$p =$ [0, 2]

$x = p$

$y = 0.7(0.5 \sin(4p - 4) + e^{-4(4p-4)^2}), \sigma = 0.3$ *(noise standard deviation)*

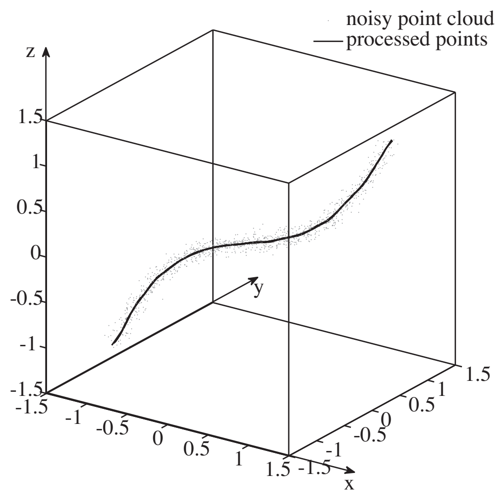

Figure 17. The result of the proposed method for the 3D point cloud data.

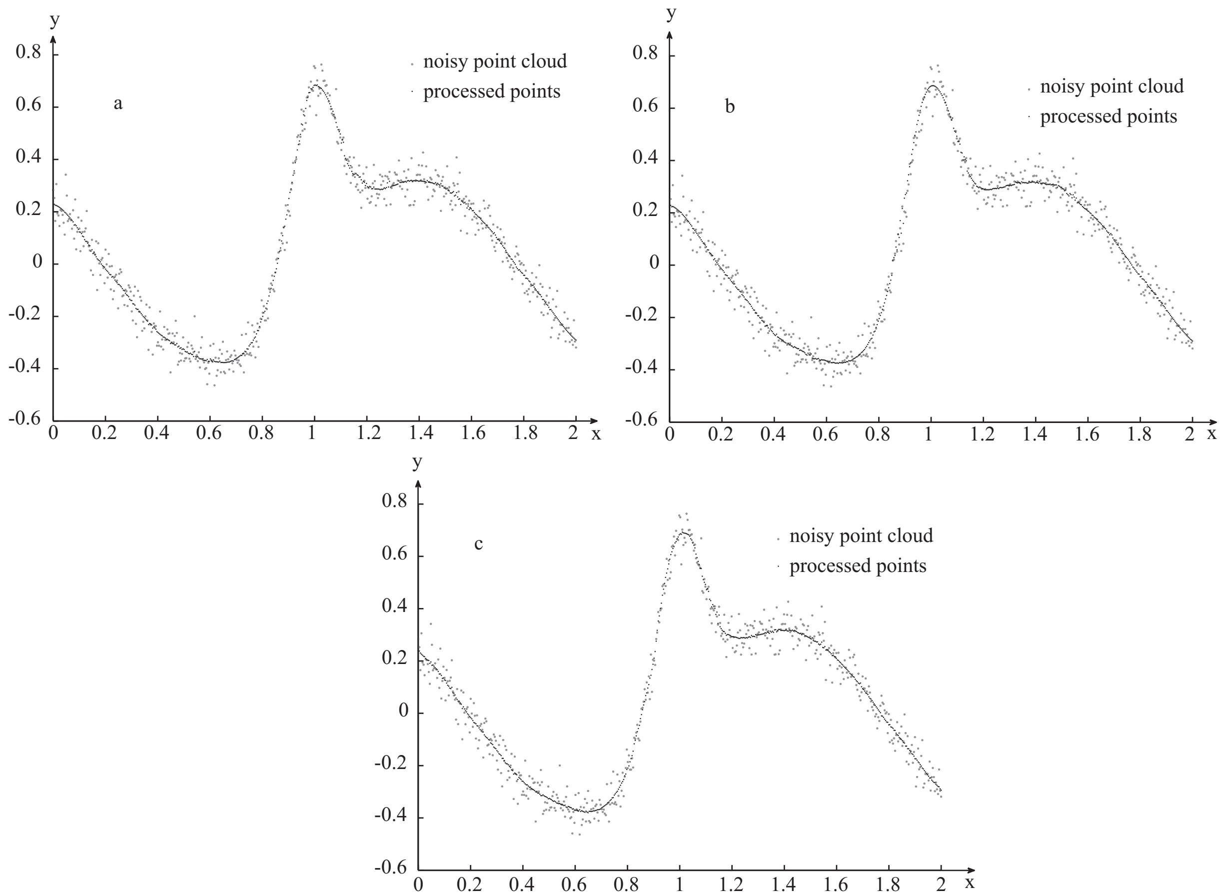

Figure 18. The comparison of the proposed method (a), KNN (b), and the constant radius method (c).

In experiment 6 (Figure 19), we use 700 points in [–2, 2] for the parameter p. Figure 19a shows the result of the proposed method, while Figures 19b and 19c are for KNN and constant radius, respectively. The used function is as below:

$x = p, \quad y = e^{-(x/0.1)^2}, p \in [-2\ 2],\ \sigma\ = 0.05$ *(noise standard deviation).*

Table 6. MSE of the 3 methods for the data in experiment 5. With a threshold value of 0.9, we achieve good results.

Method	Parameters	MSE
The proposed method	Thr = 0.9 (default)	0.00062070
KNN	K = 65 (empirically determined)	0.00063856
Constant radius	R = 0.22 (empirically determined)	0.00060584

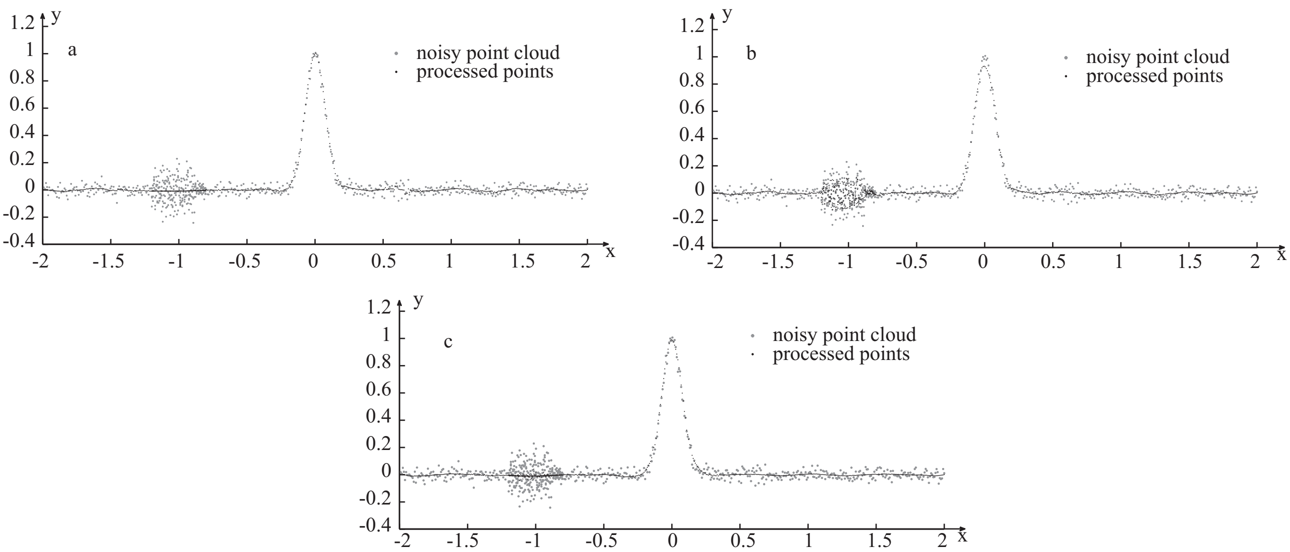

Figure 19. The comparison of the proposed method (a), KNN (b), and the constant radius method (c).

The data that are used in this experiment (experiment 6) have 2 difficulties encountered with the point clouds, which are nonhomogeneous distributions of noise and high curvature. Hence, constant bandwidth methods cannot achieve good results for this type of data, even if empirically adjusting its parameters as shown in Table 7.

Table 7. MSE of the 3 methods for the data in experiment 6. With a threshold value of 0.9, we achieve good results.

Method	Parameters	MSE
The proposed method	Thr = 0.9 (default)	0.000073331
KNN	K = 32 (empirically determined)	0.001300000
Constant radius	R = 0.291 (empirically determined)	0.000104080

5. Discussion and future work

Because of difficulties encountered with point clouds, such as variable curvature, noise, and point cloud thickness, a static bandwidth or local region radius is not an appropriate tool for the selection of a local region. In this study, an eigenvalue analysis-based simple method is proposed for dynamic bandwidth selection during curve reconstruction from noisy point clouds. The performance of the proposed method is tested on some synthetically generated data, which consists of the problems stated above. The results show that changing the bandwidth along the point cloud by using the normalized eigenvalue analysis method produces more accurate reconstructions than the static bandwidth methods. In our tests, we determine the parameters of other methods by trial and error to get the best fit. However, we use a default value of 0.9 for our proposed method to compare results. As a result, we get comparably good outcomes to the others. On the other hand, without adjusting any parameters, we can automatically get optimal results in the curve fitting process. It is a well-known fact in curve fitting problems that one cannot get a smaller error than the standard deviation of the additive noise. In Tables 6 and 7, the proposed method's MSE value shows that if one multiplies the MSE with the number of samples, the standard deviation of the additive noise can be attained.

Another important problem that affects the reconstruction quality regarding the fitting of curves from noisy point clouds is that we have to know the best local regression axis. In some of our examples, we suppose that we know the true regression line. However, most of the point cloud data come with no order or any regression axis to sort the data. We propose a simple feature called the interpoint distance to select a good regression axis, unlike those in [17,18,20]. In our proposed method, we use least squares polynomial fitting to estimate the true location of the current point, while the authors in [17] used the center of the rectangle for the same purpose. However, the method proposed in [18] needs a dimension reduction to a 2D space by calculating a best fit plane if it is necessary. After that, it also uses the least squares method for estimating the true location of the current point. The regression axis used in [18] is selected from one of the plane axes. However, our method does not need any dimension reductions. Another advantage of our method is the use of a novel local regression determination technique. It provides a more robust polynomial fitting and estimation of the true location of the current point. Our regression axis determination method itself can also be a solution for curve reconstruction problems under some conditions. The interpoint distance feature can also be used to replace the local processing step, which requires heavy calculations by partitioning the data into subregions that can be ordered with an individual regression axis. We are currently working on this subject.

A difficulty that we faced during this study was the selection strategy of the points around the neighborhood of the processed point. In [17], a rectangular selection region was used and the region growing was finalized with a criterion depending on the aspect ratio of the rectangle. The authors in [18] used a weighting parameter H for each local regression. As mentioned in that study, it is necessary to measure the thickness of the point cloud locally using some polygonal structures or the curvature analysis of the point set. Our proposed method, based on eigenvalue analysis, could be used to estimate the thickness without any extra methods. In our study, initially, we used a circular selection region. However, this method could yield some problems in high curvature regions, which were not examined in [17] or [18]. Hence, we used the EMST connection table to grow the local region, like in [18]. However, additionally, to provide a circular selection, we defined an extra connection table such that every point will be connected to its neighbor points that have a certain distance obtained from the EMST table. The breadth-first search method was utilized to search in a graph made using a combination of both connection tables (EMST connection table and extra connection table). We are also studying to find more efficient growing strategies.

There are 2 assumptions about the point clouds that are used in this work. The first is that the point clouds vary smoothly or the curvature must not be high. High curvature point clouds require more dense sampling points or less noise contribution for successful results. The second assumption about the point clouds assumes that there will not be any branches or self-intersections on the point clouds. For future work, the proposed method will be enhanced with a robust regression line estimator; thus, the problem of branching point clouds could be overcome.

The proposed method is independent from the dimension because of the use of Euclidean geometry in this work. In other words, the point clouds can be 2D, 3D, or of higher dimensions.

References

[1] H. Edelsbrunner, D.G. Kirkpatrick, R. Seidel, On the Shape of a Set of Points in the Plane, University of British Columbia, Vancouver, BC, Canada, 1981.

[2] D. Attali, "R-regular shape reconstruction from unorganized points", Proceedings of the 13th Annual ACM Symposium on Computational Geometry, pp. 248–253, 1997.

[3] L.H.D. Figueiredo, J.D.M. Gomes, "Computational morphology of curves", Visual Computer, Vol. 11, pp. 105–112, 1995.

[4] F. Bernardini, C.L. Bajaj, "Sampling and reconstructing manifolds using alpha-shapes", Proceedings of the 9th Canadian Conference on Computational Geometry, 1997.

[5] N. Amenta, M. Bern, D. Eppstein, "The crust and the beta-skeleton: combinatorial curve reconstruction", Graphical Models and Image Processing, Vol. 60/2, pp. 125–135, 1998.

[6] T.K. Dey, P. Kumar, "A simple provable algorithm for curve reconstruction", Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 893–894, 1999.

[7] T.K. Dey, K. Mehlhorn, E.A. Ramos, "Curve reconstruction: connecting dots with good reason", Proceedings of the 15th ACM Symposium on Computational Geometry, Vol. 15, pp. 229–244, 1999.

[8] E. Althaus, K. Mehlhorn, "TSP-based curve reconstruction in polynomial time", Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 686–695, 2000.

[9] T.K. Dey, R. Wenger, "Reconstruction curves with sharp corners", Proceedings of the 16th Annual Symposium on Computational Geometry, pp. 233–241, 2000.

[10] T.K. Dey, R. Wenger, "Fast reconstruction of curves with sharp corners", International Journal of Computational Geometry & Applications, Vol. 12, pp. 353–400, 2002.

[11] C. Gold, "Crust and anti-crust: a one-step boundary and skeleton extraction algorithm", Proceedings of the 15th Annual Symposium on Computational Geometry, pp. 189–196, 1999.

[12] C.M. Gold, J. Nantel, W. Yang, "Outside-in: an alternative approach to forest map digitizing", International Journal of Geographical Information Systems, Vol. 10, pp. 291–310, 1996.

[13] S. Funke, E.A. Ramos, "Reconstructing a collection of curves with corners and endpoints", Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithm, pp. 344–353, 2001.

[14] Y. Zeng, T.A. Nguyen, B. Yan, S. Li, "A distance-based parameter free algorithm for curve reconstruction", Computer-Aided Design, Vol. 40, pp. 210–222, 2008.

[15] P. Kim, H. Kim, "Point ordering with natural distance based on Brownian motion", Mathematical Problems in Engineering, Vol. 2010, Article ID 450460, 2010.

[16] T.K. Dey, "Curve and surface reconstruction", in Handbook of Discrete and Computational Geometry, 2nd ed., CRC Press, Boca Raton, FL, USA, 2004.

[17] S.W. Cheng, S. Funke, M. Golin, P. Kumar, S.H. Poon, E. Ramos, "Curve reconstruction from noisy samples" Computational Geometry: Theory and Applications, Vol. 31, pp. 63–100, 2005.

[18] I.K. Lee, "Curve reconstruction from unorganized points," Computer Aided Geometric Design, Vol. 17, pp. 161–177, 2000.

[19] G. Taubin, R. Ronfard, "Implicit simplicial models for adaptive curve reconstruction", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, pp. 321–325, 1996.

[20] Z. Hasirci, M. Ozturk, "An eigenvalue analysis based bandwidth selection method for curve reconstruction from noisy point clouds", 34th International Conference on Telecommunications and Signal Processing, pp. 478–482, 2011.