



Efficient Reconstruction of Non-Simple Curves*

Yuan-di ZHAO¹, Jun-jie CAO^{1,2}, Zhi-xun SU¹, Zhi-yang LI¹

¹School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China

²State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024, China

E-mail: jjcao1231@gmail.com

Received ?; Revision accepted ?; Crosschecked ?

Abstract: In this paper, we present a novel algorithm to reconstruct curves with self-intersections and multiple parts from unorganized strip-shaped points, which may have different local shape scales and sampling densities. We first extract an initial curve, a graph composed of polylines, to model the different structures of the points. Then a least-squares optimization is used to improve the geometric approximation. The initial curve is extracted in three steps: anisotropic farthest point sampling with an adaptable sphere, graph construction followed by non-linear region identification, and edge refinement. Our algorithm produces faithful results for points sampled from non-simple curves without pre-segmenting them. Experiments on many simulated and real data demonstrate the efficiency of our method, and more faithful curves are reconstructed compared to other existing methods.

Key words: Reverse engineering, Strip-shaped points, Curve reconstruction, Anisotropic adaptive sampling

doi:? **Document code:** A **CLC number:** TP391.4

1 Introduction

Curve reconstruction from unorganized points is a fundamental problem in reverse engineering. A large body of classical algorithms have been proposed over the past decades (Cheng *et al.*, 2005; Dey and Kumar, 1999; Edelsbrunner and Mücke, 1994; Funke and Ramos, 2001; Gold and Snoeyink, 2001; Hiyoshi, 2006; Lee, 2000; Wang *et al.*, 2006; Yang *et al.*, 2005), mostly developed for thin points. We are concerned with algorithms designed for strip-shaped points (Lin *et al.*, 2005), which are useful in many applications, such as fingerprint identification and text recognition.

The existing methods can be roughly classified into three categories. The first category tries to thin the points as much as possible, then reconstructs the curve by traditional methods designed for thin points (Lee, 2000). The second category includes methods that the boundary curve at first, followed by extracting the middle curve of the boundary curve (Lin *et al.*, 2005; Song, 2010). The third category includes methods that are well-studied in statistics, called principal curves, which are smooth, one-dimensional curves that pass through the middle of a p-dimensional data set (Einbeck *et al.*, 2005; Liu and Jia, 2005; Verbeek *et al.*, 2002; Wang and Lee, 2006).

However, several problems exist with the aforementioned algorithms, especially when the data is concentrated around a highly curved path with multiple self-intersections. Moreover, most of these algorithms exhibit poor performance for reconstructing multiple curves simultaneously. Generally speaking, most of the

‡ Corresponding author

* Project (No. U0935004) supported by the National Natural Science Foundation of China-Guangdong Joint Fund, Project (No. 60873181) supported by the National Natural Science Foundation, the Fundamental Research Funds for the Central Universities. ©Zhejiang University and Springer-Verlag Berlin Heidelberg 2010

first two kinds of algorithms make no attempt to deal with point sets with self-intersections and multiple parts. Some recently proposed algorithms inherited from principal curves deal with these cases. Verbeek *et al.* (2002) proposed an incremental method to generate principal curves by fitting line segments. These segments fit the points locally contained in each Voronoi cells and are connected to form polygonal lines. However, this divide and conquer algorithm may generate excess self-intersections and lead to incorrect connections (see Figs. 8b and 8e). Einbeck *et al.* (2005) constructed principal curve by a series of local centers of mass, the so-called local principal curve (LPC). It is performed with an angle penalization to ensure LPCs pass each crossing straight, and initialized more than once to handle multiple crossings. However, the simple angle criterion is not suitable for complicated self-intersecting curves (see Figs. 8c and 8f). The alpha shapes is another popular technique in curve reconstruction (Albou *et al.*, 2008; Bernardini and Bajaj, 1997; De-Alarcón *et al.*, 2002; Krasnoshchekov and Polishchuk, 2008; Ohbuchi and Takei, 2003), and also can be exploited to extract the boundary of the strip-shaped points (Song, 2010). However, a proper value of the alpha is hard to choose for points with changing density, which is a common case in some applications, such as faded text recognition.

In this paper, we propose an effective curve reconstruction method for strip-shaped points with self-intersections, multiple parts, different local shape scales, and sampling densities. The most important problem is how to produce a proper graph to capture the topology of underlying geometry of those points. The classic 'top-down' and 'bottom-up' strategies may give rise to incorrect reconstruction in the self-intersecting regions, as shown in the Figs. 8b and 8e. To handle those intractable cases and obtain more faithful curves passing through those regions, we present a novel strategy consisted of node sampling and edge refinement. For describing the algorithm without confusion, we denote the sampled points by nodes in the rest of this paper. In the sampling step, we apply the anisotropic farthest point sampling technique with an adaptable sphere, which proceeds along the principal direction of the points. The anisotropic adaptive sampling reflects the intrinsic properties of the points, minimizes the possibilities of incorrect connections, and greatly increases the efficiency of the entire algorithm. By inheriting the neighborhood of the original points, a graph is constructed on those nodes. With the help of the adaptive sampling, the nodes of different non-linear regions, including self-intersecting regions and highly curved regions, are distinguished by a clustering operation. Next the graph is restructured by an edge refinement procedure, which consists of two successive operators. The contraction operator is applied to delete the loops, while the merging operator removes unnecessary branches and self-intersections. Finally, the pruned graph is optimized by a least-squares (LS) system to efficiently improve the geometric approximation of the reconstructed piecewise linear curves. The pipeline of our algorithm is shown in Fig. 1.

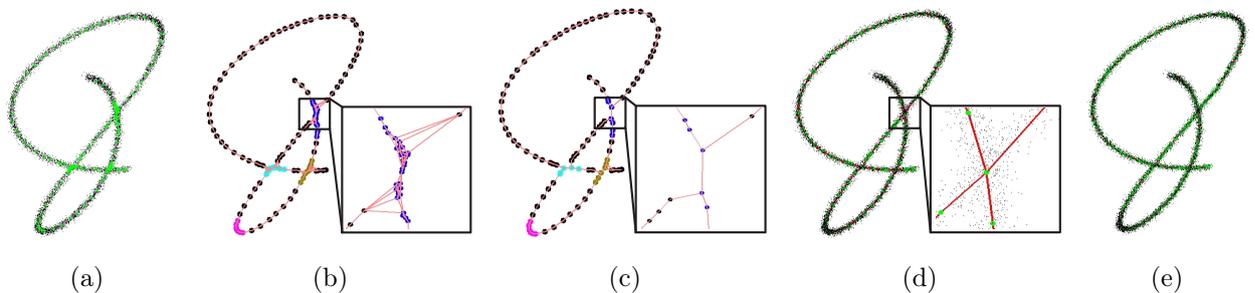


Fig. 1 The pipeline of our curve reconstruction algorithm. (a) Anisotropic adaptive sampling; (b) Non-linear region identification by clustering; (c) Edge refinement with contraction operator; (d) Edge refinement with merging operator; (e) Approximation improvement. The four node clusters corresponding to four non-linear regions in (b) and (c) are colored in pink, lightblue, Khaki and darkblue, respectively.

2 Related Works

2.1 Methods based on thinning points

Moving least-squares (MLS) technique is an efficient reconstruction method for thinning unorganized points (Alexa *et al.*, 2001; Levin, 1998). In order to choose an optimal weighting parameter for local least-squares fitting, Lee. (2000) estimated a correlation value which reflects the local thickness of the points. Once the point cloud became thin enough, he reconstructed the curve by ordering the points in the cloud after reducing the number of points if needed. However, applying MLS to points within a self-intersecting region, where the data does not process a linear structure, is not so meaningful (Tagliasacchi *et al.*, 2009). Hence, these approaches only work for point sets without self-intersecting regions.

2.2 Methods based on fitting boundary

Lin *et al.* (2005) presented a reconstruction algorithm based on computing an interval B-spline curve. Shape-based rectangle sequences were firstly constructed to cover the input points, then two pieces of boundary point sequences were computed. Finally, the boundary curves and middle curve were reconstructed using interval B-spline successively. Song. (2010) proposed a method that is good at removing outliers and handling a shape of changing density based on graph theory. He exploited an icluster tree computed from a minimal spanning tree (MST) on a graph to filter out the noises, confined a Voronoi tree for fitting the boundary of the input points, and thinned the internal region to obtain a middle curve. However, these boundary fitting methods make no attempt to attack points with self-intersections and multiple parts cases.

2.3 Methods based on principal curves

Hastie and Stuetzle (1989), who did the ground-breaking work on principal curves, proposed the concept of self-consistency, meaning that each point of the principal curve is the average overall points projected therein. A variety of other algorithms based on principal curves have been presented subsequently, which essentially differ in how the 'middle' of the distribution is found (Delicado, 2001; Kégl *et al.*, 2000; Tharpey and Flury, 1996; Tibshirani, 1992). Few of these algorithms can deal with curves with self-intersections or multiple parts. Verbeek *et al.* (2002) proposed an algorithm to reconstruct a self-intersecting curve by simply connecting line segments. However, it usually produces incorrect results when handling highly curved data. Moreover, redundant self-intersections may occur in their results (see Figs. 8b and 8e). Einbeck *et al.* (2005) utilized principal component analysis (PCA) to locally fit the principal curves. They introduced an angle penalization to distinguish the crossings and made the fitting process proceed along one direction. However, the reconstruction result is sensitive to the penalization threshold, especially in the multiple self-intersecting regions (see Figs. 8c and 8f). To reconstruct self-intersecting curves, Liu and Jia. (2005) presented a bottom-up strategy. A set of vertices was first initialized based on principal oriented points (POPs), and then a principal graph was constructed from these vertices through a two-layer process. One novelty of their algorithm is a post-processing designed for the self-intersections, which combines the two adjacent self-intersections when they are near enough. However, it may fail in more complicated situations, such as more than two adjacent self-intersections. Meanwhile, their method needs pre-segmentation when handling curves with multiple parts, because it is designed on the MST of a fully connected graph.

3 Methodology

To make our explanation clear, we first introduce some related notations. Suppose that P is a strip-shaped point set. We pay attention to fit a polygonal curve passing through P , which has the same topology and approximates the geometry of P . *Ave-dis* means the average of distances between each point and its nearest neighbor in P .

3.1 Anisotropic adaptive sampling

Constructing a graph by all the input points is unnecessary. Inspired by Cao *et al.* (2010), Liu and Jia. (2005), we sample the points first. Cao *et al.* (2010) utilized a farthest point sampling technique by a fixed-radius sampling sphere; yet, it is difficult to handle different local shape scales. If the sampling radius is too large, incorrect connections may be produced when the sampling sphere covers multiple close-by structures. And if the radius is too small, large-area dense connections may be generated even in branches in the next step, which may lead to redundant cycles after the edge contraction operation presented by Cao *et al.* (2010). These cycles are difficult to remove since we cannot distinguish whether they are formed according to the underlying topology or not.

Adaptivity: We present an adaptive sampling operation based on two intuitive observations followed by all adaptive sampling algorithms (Ruiz *et al.*, 2007): (i) A small number of nodes are enough to depict linear regions (branches). (ii) Non-linear regions (self-intersecting regions and highly curved regions) need more nodes to be characterized faithfully. PCA techniques are well-known to identify linear regions. To identify such regions with different thickness, similar to Lee (2000), we increase the sampling radius r iteratively, initialized by a minimum radius r_{min} , until all points in the current sampling sphere are linear or the radius reaches a maximum radius r_{max} . The r_{max} is defined to avoid missing close-by self-intersections, and producing incorrect connections between two close-by regions when constructing a graph in the next step. When the sampling radius r reaches r_{max} in a non-linear region, we employ a sampling sphere with the minimum radius r_{min} . The adaptive sampling strategy overcomes the two shortcomings mentioned above: incorrect connections and additional cycles. The computation time and memory of our sampling method are also reduced compared with uniform sampling approaches.

Anisotropism: Instead of adaptive sampling isotropically, sampling along the direction of current branch may further decrease the number of nodes while maintaining the correct geometry. Therefore, during each sampling step, we first assign the current node as the origin, and establish a local coordinate system based on the three local principal axes of the points in the current sampling sphere. Then we measure the distances between the current node and the points in the sampling sphere by:

$$d = r_1 + 0.1 * 1/(1 + r_2) + 0.1 * 1/(1 + r_3), \quad (1)$$

where r_1, r_2, r_3 are the Euclidean distances between the point and its projections on the first, second and third local principal axis, respectively. Finally, we prescribe the farthest point as the next sphere center.

Algorithm: We define a distance set D to control the sampling process. In order to measure the linearity, we first compute the 3×3 covariance matrix:

$$C = \begin{bmatrix} \mathbf{p}_1 - \bar{\mathbf{p}} \\ \dots \\ \mathbf{p}_k - \bar{\mathbf{p}} \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{p}_1 - \bar{\mathbf{p}} \\ \dots \\ \mathbf{p}_k - \bar{\mathbf{p}} \end{bmatrix} \quad (2)$$

for the subset S of P included in the sampling sphere, where \mathbf{p}_i ($i = 1, \dots, k$) are the points in sampling sphere, and $\bar{\mathbf{p}}$ is their barycenter. Then a value $l(S)$ is defined to evaluate the linear trend of the points in S with the eigenvalues λ_i ($i = 1, 2, 3$) of C as:

$$l(S) = \frac{\lambda_2}{\lambda_0 + \lambda_1 + \lambda_2}, \quad (3)$$

where assuming $\lambda_0 \leq \lambda_1 \leq \lambda_2$. λ_i measures the variation of S along the direction of the corresponding eigenvector \mathbf{v}_i (Pauly *et al.*, 2002). Thus, we can say the local curve is linearly extending with the direction \mathbf{v}_2 if $l(S)$ is large enough (bigger than 0.9 for all our experiments). The increasing process of radius breaks when $r > r_{max}$, which means the current points locate in non-linear regions, and we set $r = r_{min}$. D consists of the distances computed by Eq. (1) for every point in P , which are initialized with zero for all points. Each element of D is the minimum distance between the point and the nodes. The algorithm is summarized as follows:

- 1) Choose an initial point in P randomly;
- 2) Determine the local radius r of the sampling sphere as described above;
- 3) Assign the barycenter of the points in the sphere as the node;
- 4) Update a subset of D with their new distances, only for the points in the sphere; and,
- 5) Proceed with the point with the largest $d \in D$ as the next sphere center. Go to 2).

Fig. 2 briefly demonstrates the adaptive sampling process of our algorithm, where the sampling spheres and nodes in the same step are marked with the same color. The current sampling sphere is drawn in a solid line. The bold point and the symbol 'x' are the center of the sphere and the sampled node, respectively. We see that the sampling spheres are different in size according to the local thickness, and the farthest point is assigned as the next sphere center. Note that, in Fig. 2b, we assign the farthest point along the principal direction as the next sphere center, and the farthest point with Euclidean distance is not that one. Applying this anisotropic sampling method, fewer nodes are sampled to preserve the geometry and topology of the input points. Fig. 3 shows that our anisotropic adaptive sampling operation is insensitive to r_{min} . The default value of r_{min} is $10 * Ave-dis$.

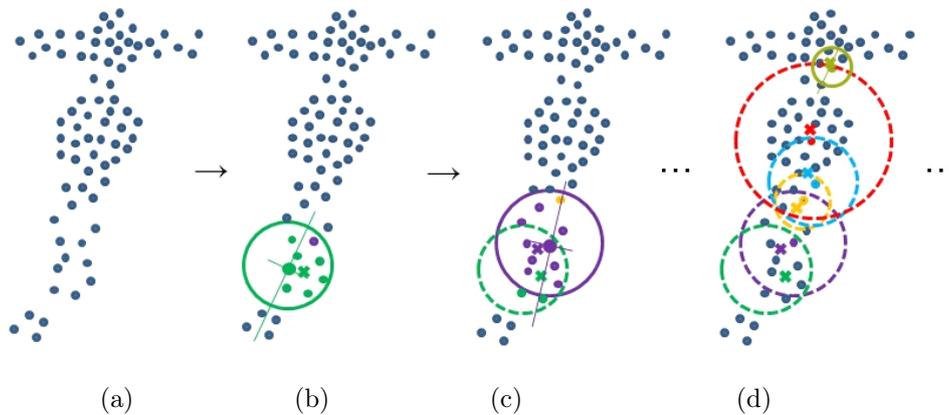


Fig. 2 The anisotropic adaptive sampling process. (a) Input point cloud; (b) The 1st-time sampling; (c) The 2nd-time sampling; (d) The n th-time sampling.

3.2 Non-linear region identification by clustering

Non-linear region identification is important to capture the essential topology of the input points. However, most of the traditional methods (Cao *et al.*, 2010; Einbeck *et al.*, 2005; Lee, 2000; Lin *et al.*, 2005; Song, 2010) do not provide a special treatment on them, which may lead to incorrect curves in the self-intersecting regions (see Figs. 8b and 8e). Liu and Jia. (2005) proposed a post-processing to rectify the topology by combining two adjacent self-intersections when they are near enough. However, it will be confused if the model has more than two adjacent self-intersecting regions. Shapira *et al.* (2008) identified self-intersections by PCA analysis with a fixed-neighborhood, so it is difficult to distinguish different local shape characteristics.

We present a non-linear region identification algorithm based on constructing a weighted graph with the adaptive sampled node set Q , followed by distinguishing them with node clusters. Each node $\mathbf{q}_i \in Q$, which is sampled using a sphere with radius r_i , represents a set of associated points $P_i \subset P$, such that $P = \bigcup_i P_i$ and $P_i \cap P_j = \phi, \forall i \neq j$. Actually, the adaptive sampling process in Section 3.1 distinguishes the linear regions and non-linear regions of input points, according to the different radii determined by the

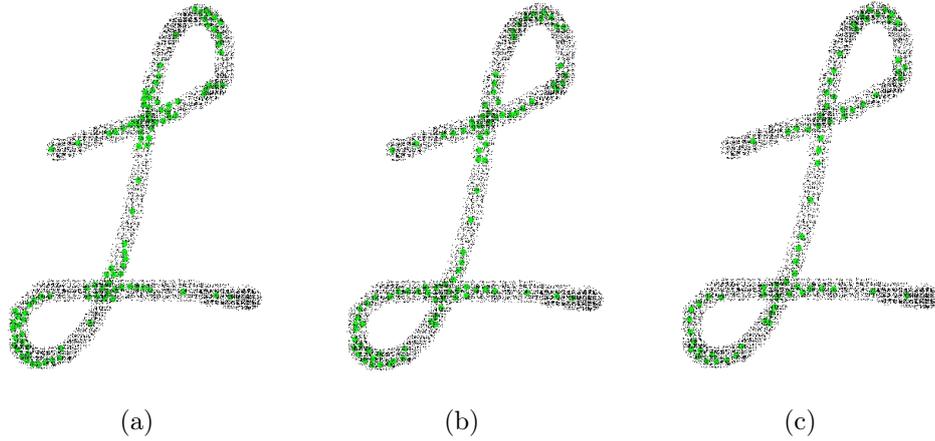


Fig. 3 The robustness of our anisotropic adaptive sampling method. (a) $r_{min} = 8 * Ave-dis$; (b) $r_{min} = 10 * Ave-dis$; (c) $r_{min} = 12 * Ave-dis$.

dynamic PCA analysis. If $r_i = r_{min}$, q_i most likely locates in non-linear regions. Otherwise, q_i is most likely in linear regions. Identifying non-linear regions by clustering directly on Q is parameter-sensitive since incorrect neighborhoods covering close-by regions are prone to generate due to the sparseness of Q , which may lead to incorrect clusters consequently. Therefore, we first construct a graph H of Q as follows: if the k nearest neighbors of the associated points of two nodes share common points, an edge between these two nodes is constructed. By examining all such pairs of nodes in Q , the graph H is constructed, respecting the shape of original points P . Then the weights are assigned to all edges of H by the following rules:

- 1) The weight for an edge is set with zero, if the length of the edge $d < 1.5 * r_{min}$;
- 2) Otherwise, the weight is directly set with the length of the edge.

With this weighted graph H and a randomly selected root node, the shortest-distances between the root and the nodes are computed and sorted. We cluster the nodes with the same shortest-distances into one group, and note them by one color, which implies that they may be sampled from the same non-linear region. Figs. 4a–4c are the non-linear region identification results under the different sampling radii in Figs. 3a–3c.

3.3 Edge refinement

Given the graph H with identified non-linear regions, we improve the edge contraction method proposed by Cao *et al.* (2010) to produce a 1D curve. To generate the curve passing through non-linear regions more faithfully, we define two successive operators: contraction and merging, to collapse unnecessary edges.

The contraction operator is designed for building a 1D curve with no triangles in H . It contracts the two endpoints of an edge by its midpoint, and then the points associated with the two endpoints are assigned to the newly created node. In addition, if one of the two endpoints is colored, the new node is assigned the same color. The contraction operator is forbidden, however, to apply if the two endpoints have different colors. In order to preserve the shape of H and make the final nodes uniformly distribute, we iteratively collapse the edge with shortest length and minimal degree, where the degree of an edge is defined as the minimal degree of its two endpoints.

The merging operator is applied to each colored group one by one, after the contraction operation. If one color group has more than two nodes whose degrees are more than two, merge them by a new one. Since we just merge them in one color group, this operator avoids merging different self-intersections, which may occur when using the method of Liu and Jia. (2005). We show the different functions of the two operators in Fig. 5b and Fig. 5c. The graph becomes a 1D curve after completing the two operators.

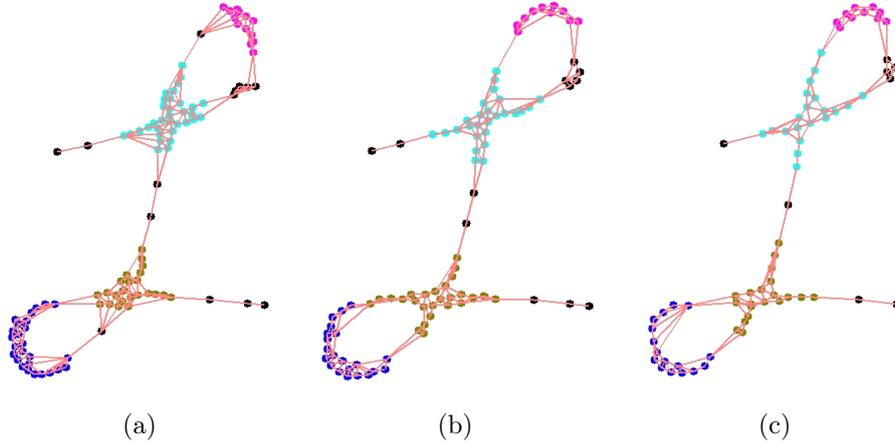


Fig. 4 The robustness of our non-linear region identification method. (a), (b), (c) are non-linear regions identified with $r_{min} = 8 * Ave-dis$, $r_{min} = 10 * Ave-dis$ and $r_{min} = 12 * Ave-dis$, respectively. The number of clusters remains at four (colored by pink, lightblue, Khaki, and dark blue respectively), which is robust to different radii.

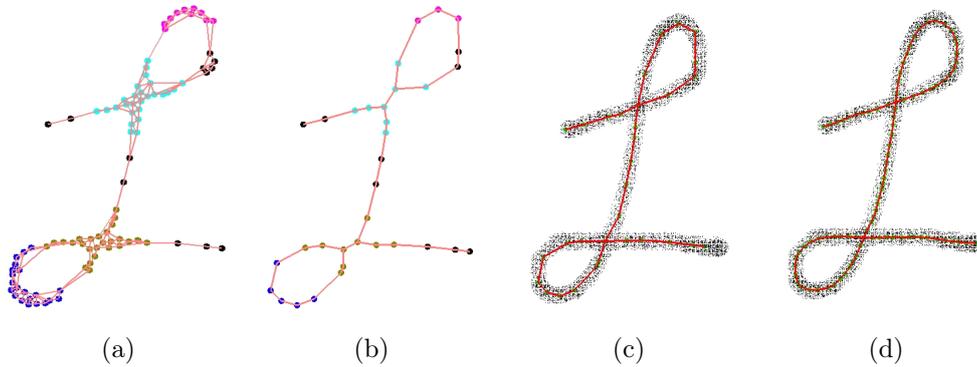


Fig. 5 Edge refinement and approximation improvement guided by the cluster groups. (a) Graph H with four clusters; (b) Result after applying the contraction operator; (c) Result after applying the merging operator; (d) Result after approximation improvement.

The edge refinement process destroys all triangles in the graph H , but may generate some imperfections occasionally, such as short circles. These imperfections can be easily detected and restructured by the classic operators on graphs. For instance, Kégl and Krzyzak. (2002) provided a collection of restructuring operations to improve the structural quality of the curve.

3.4 Approximation improvement

Generally speaking, the topology of the reconstructed curve after completing the above mentioned steps is correct, while the geometry is not always perfect. For example, as shown in Fig. 5c, some long edges of highly curved regions do not approximate the original points faithfully. Besides the topological consistency, the geometric approximation is another important factor to evaluate the reconstructed curve (Einbeck *et al.*, 2005; Liu and Jia, 2005; Singh *et al.*, 2000). As presented in Singh *et al.* (2000), the quality of the reconstructed curve depends not only upon the essential topology, but on the number of nodes as well. Therefore, we first adjust the number of the nodes, then improve the geometric approximation by LS curve fitting.

Since H is a simple piecewise linear curve, and the number of nodes can not be defined in advance,

we adjust the node set by equalizing the edge length of the graph H , such as splitting the relatively longer edges and merging the relatively short ones. Every original point is associated with one of the nodes which is the nearest. Inspired by the least-squares method for mesh processing (Sorkine and Cohen-Or, 2002), we optimize the graph H by minimizing the following LS system:

$$E(Q) = \sum \| \mathbf{q}_i - \mathbf{q}'_i \|^2 + w \| LQ \|^2, \quad (4)$$

where L is the Laplacian matrix of the graph H , providing the regularization term. In order to keep curve adhere to the original points, we exploit the approximation term used in the classic Locally Optimal Projection (LOP) operator (Lipman et al., 2007). \mathbf{q}'_i is the L_1 median of the associated points of node \mathbf{q}_i . The reason for computing L_1 median is its robustness to outliers. Penalty coefficient w is a constant value, which balances between the approximation and smoothness of the curve skeleton. Fig. 5d demonstrates the effectiveness of this fitting step.

4 Results and discussions

We have implemented our algorithm in Matlab and have conducted experiments on an Intel PC with Pentium D CPU 2.80GHz and 1 GB RAM. In all tests, we use the following thresholds: $r_{min} = 10 * Ave-dis$, $r_{max} = 6 * r_{min}$, $k = 5$, $w = 1$ by default. Various kinds of simulated and real data are tested, such as points with self-intersections, multiple parts, different local shape scales, and sampling densities. These problems are common but intractable in many applications, which can demonstrate the capabilities of our algorithm. We also compare our method with the traditional methods. Table 1 shows the runtime for all presented models, which indicates that our algorithm can produce good results with satisfied running time.

4.1 A gallery of our results

Fig. 6a shows the result of a point set with different local shape scales. Our algorithm overcomes the difficulty that the data has different local widths and reconstructs a desirable curve. Fig. 6b demonstrates an example of a point set with different sampling densities. The reconstructed curve is symmetric although the left points are much less than the right ones. In addition, the weights for Laplacian smoothing are designed carefully in order to preserve the features as shown in Fig. 6b.

Real data, including fingerprint, Chinese and English text images, are also tested. The results are shown in Figs. 6c–6e. Since our algorithm can provide satisfied results for these images, it may be well applied in fingerprint identification and text recognition fields.

4.2 Comparison with thinning-based method

We also examine some simple data in our experiments. The data in Fig. 7 is adapted from Lee (2000), including 2D and 3D strip-shaped points. The results indicate that our algorithm can generate simple open curves, as well as the method proposed by Lee (2000). However, his method needs time-consuming thinning operation first, which is not easy to control, especially for the data with different local shape scales.

4.3 Comparison with principal curve-based method on self-intersecting data

Fig. 8 demonstrates the comparison results with the methods in Verbeek et al. (2002) and Einbeck et al. (2005) on some complicated data with self-intersections. It is obvious that the curves reconstructed by our method are much better. Although we have carefully tuned the parameters in their methods, the results of method in Verbeek et al. (2002) have incorrect connections and excess self-intersections as shown in the close-up views (Figs. 8b and 8e). The results of method Einbeck et al. (2005) fail in capturing the correct topology (Fig. 8f), due to the result is sensitive to the local weight influence factors and angle criterion, which are both difficult to control. In addition, users must try many times to obtain a good result when using their method, because they randomly choose the initial point but may not pass through the whole places.

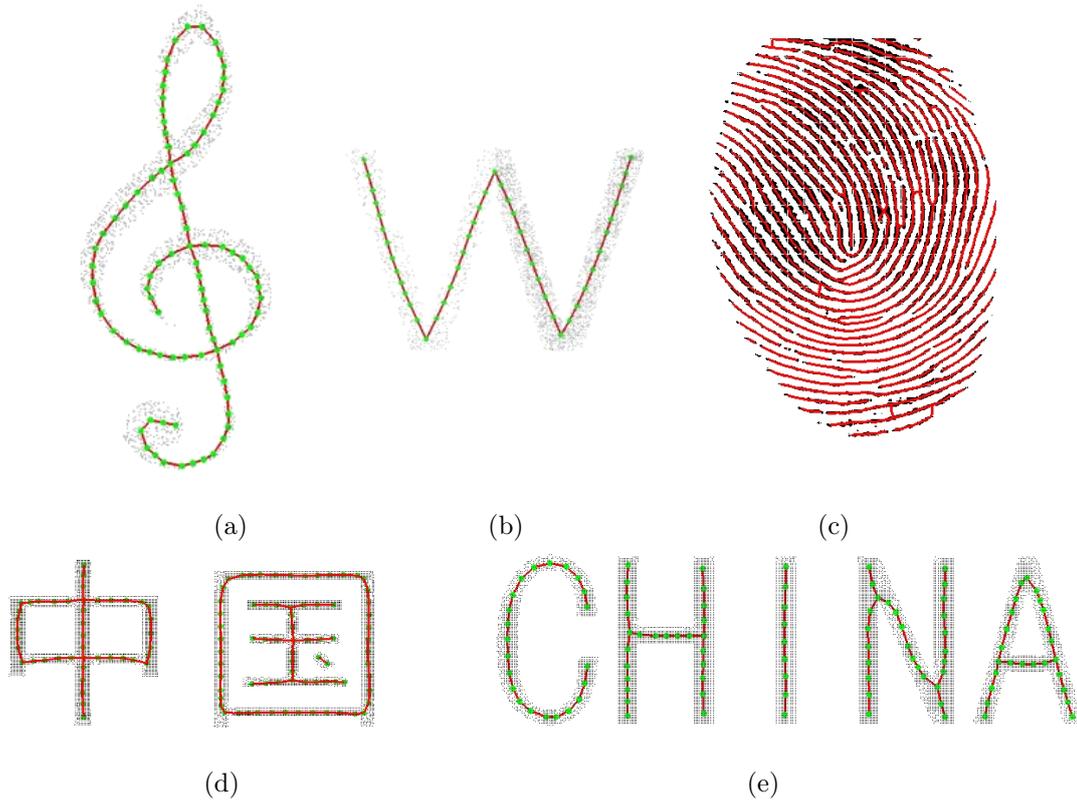


Fig. 6 More results on different kinds of data. (a) Multi-scales data; (b) Non-uniform data; (c) Fingerprint image; (d) Chinese text image; (e) English text image.

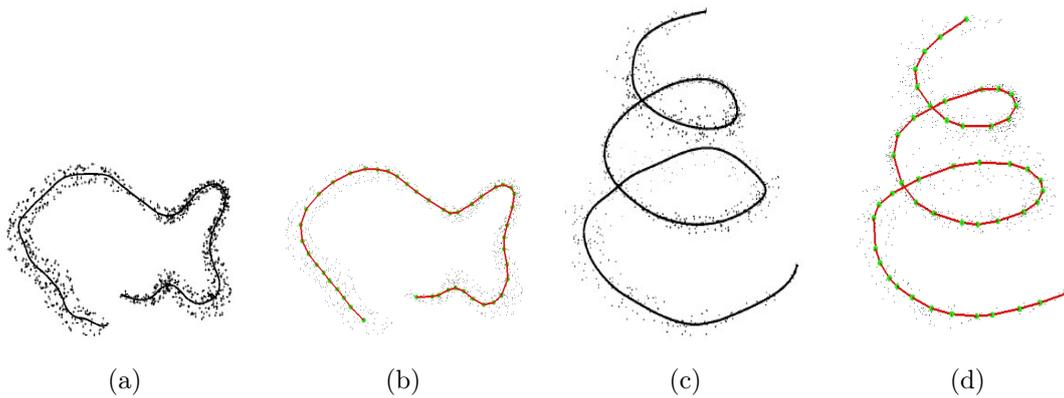


Fig. 7 Comparison with method proposed by Lee. (2000). (a) Result1 by Lee; (b) Result1 by our algorithm; (c) Result2 by Lee; (d) Result2 by our algorithm.

Our anisotropic adaptive sampling method controlled by local PCA analysis easily solves these problems, since PCA analysis with proper radius has the ability of distinguishing each self-intersecting region.

5 Conclusions

In this paper, we propose an effective and efficient curve reconstruction method for non-simple strip-shaped points. Compared with traditional methods, our algorithm can deal with complicated data with

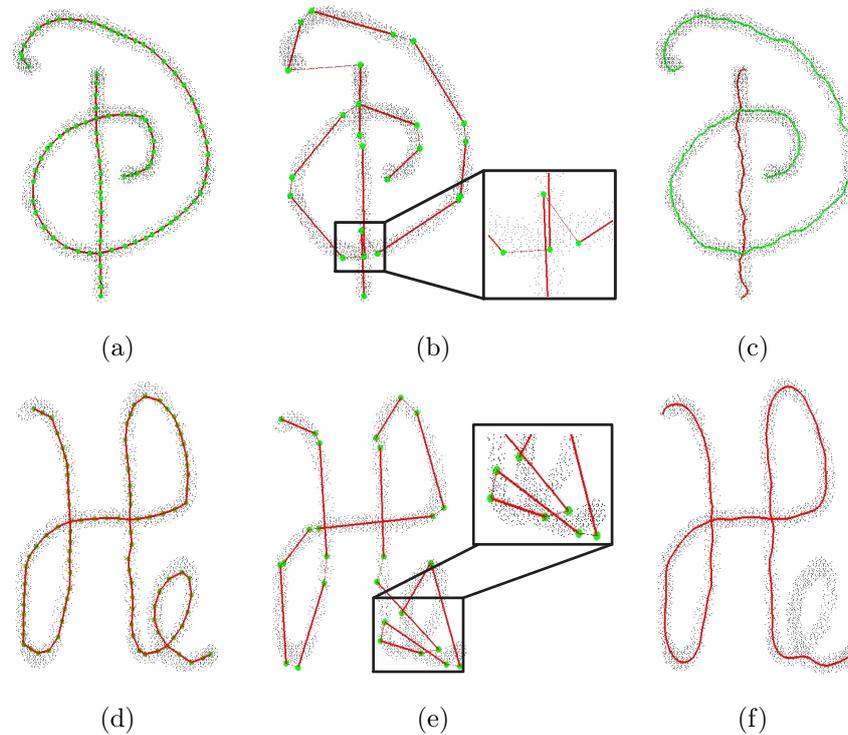


Fig. 8 Comparison with methods in Verbeek *et al.* (2002) and Einbeck *et al.* (2005). (a, d) Results of our method; (b, e) Results of Verbeek *et al.* (2002); (c, f) Results of Einbeck *et al.* (2005)

self-intersections, multiple parts, different local shape scales, and sampling densities. Not only simulated, but also real, data, such as fingerprint and text images, we tested in our experiments. The results demonstrate that our algorithm can generate faithful curves with high quality that capture the original shapes.

The main problem with our algorithm is that it may produce undesirable lines between objects, due to improper neighbor size and outliers, which is a common difficulty in point cloud processing. In addition, the proposed local linear measure does not effectively distinguish the self-intersecting regions and highly curved regions, which leads to excess computation slightly. We will try to overcome these shortcomings in future.

References

- Albou, L.P., Schwarz, B., Poch, O., Wurtz, J.M., Moras, D., 2008. Defining and characterizing protein surface using alpha shapes. *Proteins*, **76**:1-12. [doi:10.1002/prot.22301]
- Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T., 2001. Point set surfaces. Proceedings of the Conference on Visualization, p.21-28.
- Bernardini, F., Bajaj, C.L., 1997. Sampling and reconstructing manifolds using α -shapes. Proceedings of the 9th Canadian Conference on Computational Geometry, p.193-198.
- Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., Su, Z., 2010. Point cloud skeletons via Laplacian-based contraction. Proceedings of the International Conference on Shape Modeling and Applications, p.187-197. [doi:10.1109/SMI.2010.25]
- Cheng, S., Funke, S., Golin, M., Kumar, P., Poon, S., Ramos, E., 2005. Curve reconstruction from noisy samples. *Comput. Geom.*, **31**:63-100. [doi:10.1016/j.comgeo.2004.07.004]
- De-Alarcón, P., Pascual-Montano, A., Gupta, A., Carazo, J., 2002. Modeling shape and topology of low-resolution density maps of biological macromolecules. *Biophys.*, **83**(2):619-632.
- Delicado, P., 2001. Another look at principal curves and surfaces. *J. Multivariate Anal.*, **77**:84-116. [doi:10.1006/jmva.2000.1917]
- Dey, T.K., Kumar, P., 1999. A simple provable algorithm for curve reconstruction. Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, p.893-894.

Table 1 Time statistics of the proposed algorithm (seconds)

Model	#Points	Sampling	Clustering	Refinement	Fitting	Total
Fig. 1	10000	0.526	0.090	0.036	0.057	0.710
Fig. 3, 4, 5	7883	0.405	0.057	0.023	0.042	0.527
Fig. 6a	2170	0.140	0.033	0.017	0.037	0.228
Fig. 6b	2998	0.137	0.032	0.013	0.020	0.202
Fig. 6c	43876	5.592	2.937	0.574	7.463	16.566
Fig. 6d	7917	0.444	0.070	0.018	0.084	0.616
Fig. 6e	9939	0.568	0.085	0.022	0.079	0.754
Fig. 7b	1000	0.079	0.018	0.008	0.024	0.129
Fig. 7d	1000	0.073	0.016	0.002	0.023	0.115
Fig. 8a	4403	0.225	0.046	0.021	0.042	0.334
Fig. 8d	3571	0.215	0.048	0.021	0.058	0.342

- Edelsbrummer, H., Mücke, E.P., 1994. Three-dimensional alpha shapes. *ACM Trans. Graph.*, **13**:43-72. [doi:10.1145/174462.156635]
- Einbeck, J., Tutz, G., Evers, L., 2005. Local principal curves. *Statistics and Computing*, **15**(4):301-313. [doi:10.1007/s11222-005-4073-8]
- Funke, S., Ramos, E.A., 2001. Reconstructing a collection of curves with corners and endpoints. Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, p.344-353.
- Gold, C.M., Snoeyink, J., 2001. A one-step crust and skeleton extraction algorithm. *Algorithmica*, **30**:144-163. [doi:10.1007/s00453-001-0014-x]
- Hastie, T., Stuetzle, W., 1989. Principal curves. *J. of the American Statistical Association.*, **84**(406):502-516.
- Hiyoshi, H., 2006. Closed curve reconstruction from unorganized sample points. Proceedings of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering, p.122-131.
- Kégl, B., Krzyzak, A., Linder, T., Zeger, K., 2000. Learning and design of principal curves. *IEEE Trans. Pattern Analysis and Machine Intelligence.*, **22**:281-297. [doi:10.1109/34.841759]
- Kégl, B., Krzyzak, A., 2002. Piecewise linear skeletonization using principal curves. *IEEE Trans. Pattern Analysis and Machine Intelligence.*, **24**(1):59-74. [doi:10.1109/34.982884]
- Krasnoshchekov, D.N., Polishchuk, V., 2008. Robust curve reconstruction with k-order alpha-shapes. Proceedings of the International Conference on Shape Modeling and Applications, p.279-280. [doi:10.1109/SMI.2008.4548006]
- Lee, I.K., 2000. Curve reconstruction from unorganized points. *Computer Aided Geometry Design*, **17**(2):161-177. [doi:10.1016/s0167-8396(99)00044-8]
- Levin, D., 1998. The approximation power of moving least squares. *Math. Comput.*, **67**:1517-1531. [doi:10.1090/s0025-5718-98-00974-0]
- Lin, H., Chen, W., Wang, G., 2005. Curve reconstruction based on an interval B-spline curve. *Vis. Comput.*, **21**(6):418-427. [doi:10.1007/s00371-005-0304-4]
- Lipman, Y., Cohen-Or, D., Levin, D., Tal-Ezer, H., 2007. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.*, **26**(3):22. [doi:10.1145/1276377.1276405]
- Liu, X., Jia, Y., 2005. A bottom-up algorithm for finding principal curves with applications to image skeletonization. *Pattern Recognition*, **38**:1079-1085. [doi:10.1016/j.patcog.2004.11.016]
- Ohbuchi, R., Takei, T., 2003. Shape-similarity comparison of 3D models using alpha shapes. Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, p.293-302. [doi:10.1109/PCCGA.2003.1238271]
- Pauly, M., Gross, M., Kobbelt, L.P., 2002. Efficient simplification of point-sampled surfaces. Proceedings of the Conference on Visualization, p.163-170. [doi:10.1109/VISUAL.2002.1183771]
- Ruiz, O., Vanegas, C., Cadavid, C., 2007. Principal component and Voronoi skeleton alternatives for curve reconstruction from noisy point sets. *J. of Engineering Design.*, **18**(5):437-458. [doi:10.1080/09544820701403771]
- Shapira, L., Shamir, A., Cohen-Or, D., 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.*, **24**(4):249-259. [doi:10.1007/s00371-007-0197-5]
- Singh, R., Cherkassky, V., Papanikolopoulos, N.P., 2000. Self-Organizing maps for the skeletonization of sparse shapes. *IEEE Trans. Neural Networks.*, **11**(1):241-248. [doi:10.1109/72.822527]

- Song, Y., 2010. Boundary fitting for 2D curve reconstruction. *Vis. Comput.*, **26**(3):187-204. [doi:10.1007/s00371-009-0395-4]
- Sorkine, O., Cohen-Or, D., 2004. Least-squares meshes. Proceedings of the International Conference on Shape Modeling and Applications, p.191-199. [doi:10.1109/SMI.2004.1314506]
- Tagliasacchi, A., Zhang, H., Cohen-Or, D., 2009. Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph.*, **28**(3):71:1-71:9. [doi:10.1145/1531326.1531377]
- Tharpey, T., Flury, B., 1996. Self-consistency: A fundamental concept in statistics. *Statistical Science*, **11**:229-243. [doi:10.1214/ss/1032280215]
- Tibshirani, R., 1992. Principal curves revisited. *Statistics and Computing*, **2**:183-190. [doi:10.1007/BF01889678]
- Verbeek, J.J., Vlassis, N., Kröse, B., 2002. A k-segments algorithm for finding principal curves. *Pattern Recognition Letters*, **23**(8):1009-1017. [doi:10.1016/s0167-8655(02)00032-6]
- Wang, H., Lee, C.M., 2006. Automatic parameter selection for a k-segments algorithm for computing principal curves. *Pattern Recognition Letters* **27**(10):1141-1150. [doi:10.1016/j.patrec.2005.12.005]
- Wang, W., Pottmann, H., Liu, Y., 2006. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graph.*, **25**(2):214-238. [doi:10.1145/1138450.1138453]
- Yang, Z., Deng, J., Chen, F., 2005. Fitting unorganized point clouds with active implicit B-spline curves. *Vis. Comput.*, **21**(8-10):831-839. [doi:10.1007/s00371-005-0340-0]