

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/226264895>

# Ellipse-based principal component analysis for self-intersecting curve reconstruction from noisy point sets

Article in *The Visual Computer* · March 2011

DOI: 10.1007/s00371-010-0527-x · Source: DBLP

CITATIONS

8

READS

162

3 authors:



Oscar E. Ruiz

Universidad EAFIT

128 PUBLICATIONS 438 CITATIONS

[SEE PROFILE](#)



Carlos A Vanegas

University of California, Berkeley

20 PUBLICATIONS 787 CITATIONS

[SEE PROFILE](#)



Carlos A. Cadavid

Universidad EAFIT

42 PUBLICATIONS 96 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Modular Dynamics of Mechanisms [View project](#)



3D Mesh Reconstruction [View project](#)

# Ellipse-based Principal Component Analysis for Self-intersecting Curve Reconstruction from Noisy Point Sets

O. Ruiz · C. Vanegas · C. Cadavid

30 November 2009

**Abstract** Surface reconstruction from cross cuts usually requires curve reconstruction from planar noisy point samples. The output curves must form a possibly disconnected 1-manifold for the surface reconstruction to proceed. This article describes an implemented algorithm for the reconstruction of planar curves (1-manifolds) out of noisy point samples of a self-intersecting or nearly self-intersecting planar curve  $C$ .  $C : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}^2$  is self-intersecting if  $C(u) = C(v)$ ,  $u \neq v$ ,  $u, v \in (a, b)$  ( $C(u)$  is the self-intersection point). We consider only *transversal* self-intersections, i.e. those for which the tangents of the intersecting branches at the intersection point do not coincide ( $C'(u) \neq C'(v)$ ). In the presence of noise, curves which self-intersect cannot be distinguished from curves which near self-intersect. Existing algorithms for curve reconstruction out of either noisy point samples or pixel data, do not produce a (possibly disconnected) Piecewise Linear 1-manifold approaching the whole point sample. The algorithm implemented in this work uses Principal Component Analysis (PCA) with elliptic support regions near the self-intersections. The algorithm was successful in recovering contours out of noisy slice samples of a sur-

face. As a test for the correctness of the obtained curves in the slice levels, they were input to an algorithm of surface reconstruction, leading to a reconstructed surface which reproduces the topological and geometrical properties of the original object. The algorithm robustly reacts not only to statistical non-correlation at the self-intersections (non-manifold neighborhoods) but also to occasional high noise at the non-self-intersecting (1-manifold) neighborhoods.

**Keywords** self-intersecting Curve Reconstruction · Elliptic support region · Principal Component Analysis · Noisy Samples

## Glossary

PL: Piecewise Linear.

$C$ : Planar open or closed, possibly self-intersecting or nearly self-intersecting, curve.

$S = \{p_0, p_1, \dots, p_n\}$ : An unorganized noisy point sample of  $C$ .

$\epsilon$ : Stochastic component of the point sample.

$B(p, r)$ : The disk of radius  $r$  centered at point  $p$ .

$L(\lambda) = p + \lambda * \hat{v}$ : Parametric form of the straight line passing through  $p$ , directed by the unit vector  $\hat{v}$  with signed distance parameter  $\lambda$ .

$f_1, f_2$ : Foci of an ellipse in  $\mathbb{R}^2$ .

$E(f_1, f_2, \alpha)$ : Ellipse  $\{p \in \mathbb{R}^2 : d(p, f_1) + d(p, f_2) = 2\alpha\}$ .

---

C. Vanegas  
Department of Computer Science, Purdue University.  
West Lafayette, IN 47907-2066  
USA  
E-mail: cvanegas@cs.purdue.edu

$\rho_{X,Y}$ : Linear regression correlation coefficient between variables  $Y$  and  $X$ .

$[\rho, p, \hat{v}] = pca(S_E)$ : Principal Component Analysis of the point set  $S_E$ , rendering as a result the linear trend  $L(\lambda) = p + \lambda * \hat{v}$  with correlation coefficient  $\rho$ .

$Q$ : Queue whose elements are pairs  $[p, v]$  formed by a vector  $v$  anchored at point  $p$ .

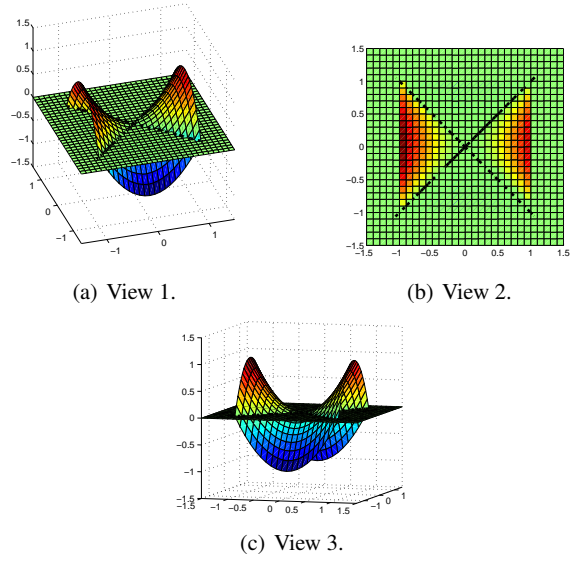
$PL\_Curve\_Set = \{c_1, c_2, \dots, c_m\}$ : Set of PL pairwise disjoint curves  $c_1, c_2, \dots, c_m$ .

## 1 Introduction

This paper discusses the implementation and results of an algorithm to reconstruct Piecewise Linear (PL) approximations for a possibly self-intersecting or nearly self-intersecting planar curve  $C$  sampled with a noisy point set.

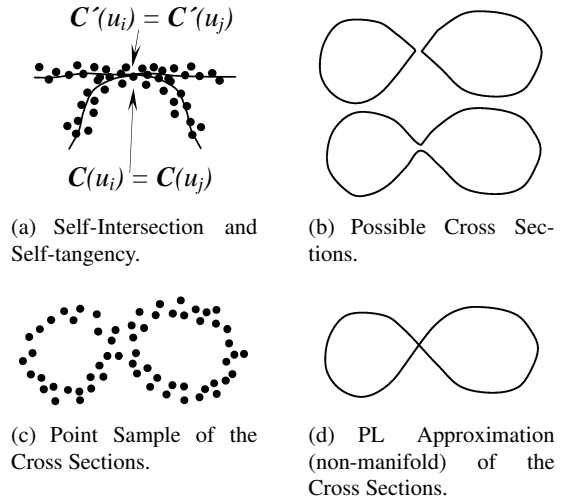
By  $C$  we mean a function  $C : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}^2$  that is continuously differentiable and regular (i.e.  $C'(u) \neq 0$  for all  $u \in [a, b]$ ).  $C$  will be said to be *self-intersecting* if there is a finite set  $\{u_1, \dots, u_n\} \subset (a, b)$  such that for each  $i$  there is a  $j \neq i$  such that  $C(u_i) = C(u_j)$  (the  $C(u_i)$ s are the self-intersection points). We consider only *transversal* self-intersections, i.e. those for which the tangents of the intersecting branches at the intersection point do not coincide ( $C'(u_i) \neq C'(u_j)$ ).

The cross cuts of a surface might be self-intersecting contours as shown in Figure 1. Figure 2(a) shows a non-transversal self-intersection with a sample. An  $\epsilon$ -near self-intersecting curve is one for which there exists a point sample with noise  $\epsilon$  being identical to the  $\epsilon$  sample of some self-intersecting curve. In the rest of the article we will simply refer to these as *nearly self-intersecting curves* (omitting the  $\epsilon$ ). The cross section of an object might have a configuration as in the upper or lower parts of Figure 2(b). A typical noise sample of such cross sections is the set of points  $S = \{p_0, p_1, \dots, p_n\}$  with  $p_i \in \mathbb{R}^2$  as in Figure 2(c). Notice that the curves might have any of the forms in Figure 2(b) or be actually self-intersecting as in Figure 2(d), and the point sample still looks as in Figure 2(c).



**Fig. 1** Cut of the Hyperbolic paraboloid  $z = x^2 - y^2$  with the plane  $z = 0$  forms contours which self-intersect at the saddle point  $(0, 0, 0)$ .

Notice that for curves as in Figure 2(d) there is no Nyquist-compliant sample, since the local characteristic dimension  $\delta$  is zero.



**Fig. 2** Ambiguous Noise Sample of Near self-intersecting Curves.

The input to the algorithm implemented in this work is presented in Figure 2(c). Either of the results in Figure 2(b) is acceptable for our algorithm, as a legal 1-manifold. Such contours may then be pro-

cessed for surface reconstruction as in [RCG<sup>+</sup>05] to get an approximation of a closed surface. This process is also conducted in the present work. Figure 2(d) shows a PL approximation of  $C$ , although it is obviously non-acceptable because it is a non-manifold. However, reaching it is an important achievement for any curve reconstruction algorithm, and can be easily corrected to obtain either situation in Figure 2(b).

The sample of the curve  $C$  in Figure 2(c) must respect the Nyquist or Shannon ([Nyq28],[Nyq02], [Sha49], [Sha98]) criterion for digital sampling to be able to retain the topology of  $C$ . This means that the effective sampling interval  $\delta = \delta_n + \epsilon$  (nominal plus stochastic components) must be smaller than half of the minimal detail that the sampling is supposed to preserve.

### 1.1 1-Manifolds in $R^2$

$M$  is a 1-manifold in  $R^2$  if for each point  $p \in M$  there exists a  $\delta > 0$ , such that  $M \cap B(p, \delta)$  is homeomorphic to the real interval  $(0, 1)$ .  $M$  is said to be a 1-manifold with border if for each point  $p \in M$  there is a  $\delta > 0$ , such that  $M \cap B(p, \delta)$  is homeomorphic to either of the real intervals  $(0, 1)$  or  $[0, 1)$ . A set of mutually disjoint closed non-self-intersecting curves is a 1-manifold. A set of mutually disjoint non-self-intersecting curves with at least one of them open is a 1-manifold with border. Informally, a small neighborhood of a point at which a curve ceases to be a 1-manifold looks like three or more semi-arcs emanating from the point. For example, in Figure 2(d), a small neighborhood of the non-manifold point looks like four semi-arcs emanating from the point.

### 1.2 PL Reconstruction of Self-intersecting Curves Sampled with Noise

An  $n$ th order PL approximation of a curve  $C$  out of a noisy sample of it is a polygonal curve  $P = [q_0, q_1, \dots, q_n]$  which resembles the original curve  $C$  up to its  $n$ -th derivative. A by-product of the process

producing  $P$  is a parameterization of the contour so recovered, which is fundamental in downstream applications, such as surface reconstruction from cross cuts.

In  $P$  the concept of a sequence is central. Many algorithms for curve reconstruction fail to establish such a sequence when they approach the self-intersections of  $C$ , exactly because the concept of order is destroyed at such neighborhoods. Our algorithm is able to find the sequence of points forming  $P$ , even at the self-intersections. A post-processing is then used to break down  $P$  into manifold components.

In the present article the authors attack the problem of self-intersecting or nearly self-intersecting curves (which in the presence of noise are undistinguishable) by using a mutating elliptic support region for the PCA calculation. Informally speaking, near the self-intersections the support region for PCA becomes an ellipse, and far away it is circular. This variation makes the algorithm more robust when facing low correlation coefficients at the intersections.

This article is organized as follows: Section 2 presents a taxonomy of the existing approaches addressing the problem, including previous algorithms developed by the authors. Section 3 proposes improvements to existing algorithms to take into consideration self-intersecting curves, along with the necessary mathematical facts supporting such algorithms. Section 4 addresses the application of the methodology to non trivial topological cases and presents the results of surface reconstruction from planar slice samples. Section 5 concludes the article and discusses possible future work directions.

## 2 Literature Review

The reconstruction of a curve  $C$  out of a noiseless point sample is addressed by relatively abundant literature, relying mostly on graph synthesis techniques. However, since we are interested in Design and Manufacturing applications we must address noisy point samples.

The strategies for the reconstruction of  $C$  mainly found in the reviewed literature are: (1) Medial Axis

calculation, (2) Scalar Field calculation, with (2.1) Radial Basis functions, (2.2) Differential Equations (e.g. Level sets), (3) Statistical estimation by Principal Component Analysis, including (3.1) Noising / De-noising of the point set, (3.2) Straight Segment Synthesis, (3.3) Parametric Curve synthesis (Bezier, Spline, Nurbs), (4) Probabilistic estimation of Topological properties, (5) Probabilistic estimation of Geometrical properties.

Techniques transversal to many of the approaches mentioned above are: (a) minimization techniques, (b) graph theory, (c) probabilistic and statistical estimation, (d) Delaunay-Voronoi-based methods.

In the reviewed literature, the vast majority of the articles do not address the issue of (nearly) self-intersecting curve reconstruction out of point samples. The two references reporting such advance do not explain how their methods effectively deal with such a feature.

In the consulted literature there is a general absence of formal analysis for the computational complexity of the proposed algorithms. When present, such a discussion only addresses average cases and central *time* expenses, ignoring the *space* complexity of the collateral data structures and the *time* spent in building them.

## 2.1 Medial Axis

[Kég99] and [KK02] explore the recovery of a Principal Graph underlying a 2D point sample (e.g. a character meant to be pen strokes). The authors set up a numerical optimization algorithm that balances two competing criteria: (i) the inclusion in the graph of as many pixels as possible of the ones present in the stroke, and (ii) the minimization of the medial axis curvature. Since this algorithm aims at character recognition, its final result is not required to be a 1-manifold. Therefore, self-intersections are permitted (like in the “H” or “8” characters). In our case, the final result of the reconstruction must be a set of disjoint non-self-intersecting curves, and therefore one must meet higher requirements than the ones met by [KK02] and [Kég99]. The algorithm implemented in [Kég99] and [KK02] finds

an approximation to the medial axis of the black pixel region. The complexity of their algorithm is estimated by the authors in  $O(N)$ , where  $N$  is the number of black pixels in the image. This estimation must be carefully interpreted since it only takes into account a part of the process. For example, only finding a medial axis approximation has a minimal complexity of  $O(N^2)$ . In addition, the proposed strategy requires collateral data structures whose time and memory expenses significantly increase the cost of the whole process of curve reconstruction.

[TZCO09] discusses the synthesis of the skeleton of a 2-manifold sampled with *oriented points*. An *oriented point* contains the  $(x, y, z)$  coordinates and the vector normal to the surface at such point. The manifold is constructed with cylindrical branches meeting at joint neighborhoods. The point sample might be incomplete, according to the authors. The point sample is partitioned in quasi-planar sections. Each point subset of the partition must look like the section of a cylinder, according to the basic assumptions on the manifold being sampled. Each cross section of the cylinder has a *statistical* center. The sequence of such centers contributes to the skeleton. At the joints, there are no such skeletons. At such regions several processes are applied to obtain a line-like structure: smoothing, thinning, re-centering, joint collapsing and re-distribution of point samples. This process requires intensive user control. Notice that the skeleton is not a manifold in general. Therefore, the proposed algorithm is not required to neutralize the non-manifold neighborhoods. The authors do not discuss the complexity of their algorithm.

## 2.2 Scalar Fields

### 2.2.1 Radial Basis Functions

[PMG04] presents a method to establish a likelihood map or scalar function in  $R^2$  around a noisy point sample. The scalar function records a high value for  $(x, y)$  if it is close to a sampled 2D curve  $C$ . After a like-

likelihood map is calculated (similar to Figure 3(b)), a PL approximation is initialized in the form of a topological circle on  $R^2$ , and then allowed to drift to settle on the highest values of the likelihood map. The paper does not discuss self-intersecting curve point samples, disconnected curves, initial size or position of the circle, or computational complexity of the method.

[KZ04] presents a definition of an implicit surface over a noisy point cloud using weighted least squares based on a geometric proximity graph. Self-intersections are not addressed. In this reference, computing a Close Pair Shortest Paths (CPSP) table among  $N$  points in  $R^3$  is reported to consume  $O(N)$  computing time and  $O(N)$  storage space. Such predictions do not count the time in building the collateral (breadth-first, depth-first) data structures. No accounting is devoted to the administration of collateral data structures or pre-processing time.

### 2.2.2 Fitting of $y = f(x)$

[AK03] fits polynomials  $h(x)$  to a series of points  $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$  such that  $h(x_i) \in [y_i - \delta, y_i + \delta]$ . Important limitations of this reference are: (1) the majority of applications in applied computational geometry deal with curves in  $R^2$  which are not the graphs of *functions*. For example, a closed contour in  $R^2$  cannot be expressed as  $y = h(x)$ . (2) No applications are given in the article. (3) The authors do not discuss the complexity and scope of their solution.

### 2.2.3 Reconstruction by Differential Equations and Level Sets

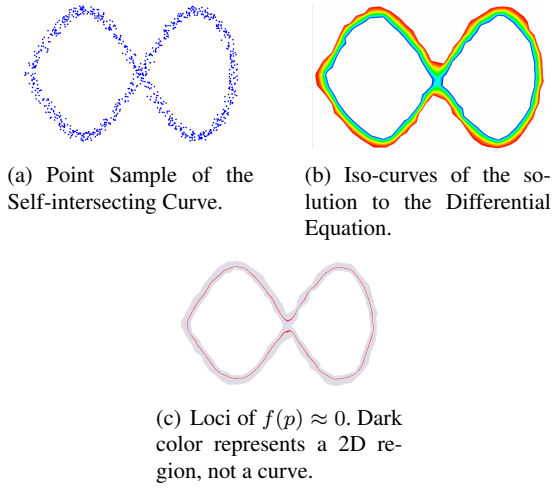
Consider a noisy point set  $S$  in  $R^2$  sampled on a regular curve  $C$ . There is abundant literature that seeks to recover an approximation to  $C$  as a solution of a differential equation stated on a domain  $\Omega$  which contains  $S$ . The goal of such methods is to synthesize an implicit function  $f : R^2 \rightarrow R$ , solution of a differential equation, and the sought curve  $C$  is the implicit curve  $f(x, y) = 0$ .

From the reviewed literature ([OF00], [OS88], [kZOMK00], [ZOF01], [LZJ<sup>+</sup>05]) we may conclude the following: (a) The definition of the  $\Omega$  region covering the set  $S$  is already an open problem. However, a convenient informal definition of  $\Omega$  would be the tape-shaped polygon covering  $S$  (see Figure 3). (b) To solve the differential equation it is essential to draw geometric information from the point sample itself. For example, Level Set methods require a vector field  $v : R^2 \rightarrow R^2$  normal or tangent to  $C$  at every point to be able to reconstruct  $C$ . As an effect, self-intersecting curves cannot be recovered in this manner, since in such curves the tangent / normal fields are undefined at the self-intersections. (c) Whichever solution  $f : R^2 \rightarrow R$  can be found by solving differential equations, it must be kept in mind that any curve  $C$  recovered as an iso-level set of  $f$  will be a closed one. This implies that open curves  $C$  cannot be recovered by differential equation methods, unless additional manipulations on the domain  $\Omega$  (not reported yet) are introduced. (d) Once a function  $f$  has been estimated by solving the differential equation, the value  $k$  representing the iso-curve  $f(x, y) = k$  that approaches  $C$  must be guessed (Figure 3(c)), and such an iso-curve might not be a 1-manifold. (e)  $f(x, y) = k$  might contain disconnected curves even if  $C$  was originally connected. (f) The computationally obtained solution to a differential equation  $f$  is a function whose domain is a grid of points. Passing from  $f$  to iso-curves  $f(x, y) = k$  clearly requires an additional process ([Blo94],[BF95],[Blo88]). For all these reasons, finding  $f$  as a solution to a differential equation still needs a significant effort before it can be considered as a reliable tool for building 1-manifolds out of the point set  $S$ .

## 2.3 Statistically Based Methods

### 2.3.1 Noising / De-noising of Point Set

[CFG<sup>+</sup>05] attack the problem of fitting PL curves to noisy point samples by computing a sequence of new point sets having less noise than the initial point set



**Fig. 3** Curve Reconstruction with Differential Equations ([RVC07], [UR08]).

and less points. This is done by searching thin rectangles normal to the local curve tangent for sample points and by driving the sample to the expected value of the curve. When the point set is sufficiently thin, the actual PL approximation to  $C$  is computed using a crust algorithm (in this case the NN Crust by [DK99]). The algorithm is guaranteed to converge if a sufficiently good point sample of  $C$  is available. We must point out that such a sample does not exist for self-intersecting curves, as the Nyquist criterion cannot be met. This algorithm is misled when the curve gets close to itself because the thinning of the point set about the likely curve locus has two or more attractors. [CFG<sup>+</sup>05] discusses the probabilities of the obtained curve being homeomorphic to the original one. They neither discuss computational complexity nor present application examples.

[MD07] presents an algorithm that takes a noise-free sample of a non-self-intersecting curve in  $R^2$ . The algorithm adds noise in the point sample, in the direction perpendicular to the originally sampled curve. The algorithm eliminates the noise by replacing the points falling in a circle by the center-point of a segment joining the most extreme points inside the circle. After the noise is removed, the point set is fed to a Relative Neighborhood Graph, derived from the Delaunay trian-

gulation. Shortcomings of this approach are: (1) Noise is added to a point set that is originally noise-free. (2) The original point set is filtered, with high frequencies removed. (3) The noise removal pre-processing costs  $O(N^3)$  computation time. (4) The article presents results for the enlargement of the point set but it does not do so for the curve reconstruction itself. (5) The article does not discuss complexity at any point.

### 2.3.2 Synthesis of Straight Segments

[Lee00] presents a least-squares algorithm to approximate a set of unorganized points with a simple 3D curve without self-intersections. This algorithm uses an Euclidean Minimum Spanning Tree (EMST). The algorithm performs thinning on the point cloud before calculating its PL approximation. No discussion of computational complexity is presented.

[VVK01] presents a PL approximation of a planar curve  $C$  whose sample  $S$  has noise. A set of straight segments is accommodated on the region defined by the sample, with each segment being locally tangent to  $C$ . A second part of the algorithm defines an order on the straight segments and threads the tail of one with the head of the next one. This approach cannot define how a self-intersecting curve is handled, since no straight segment and no *tail*, *head* or *next* segment can be defined at the self-intersection, due to the lack of correlation in the local point set. No discussion of computational complexity is presented.

### 2.3.3 Synthesis of Parametric Curves

[WPL06] fit B-Splines to a set of noisy point sets using curvature-based squared distance minimization. The control points  $q_i$  of a Spline curve  $P(t) = \sum_{i=1}^M B_i(t) * q_i$  are set such that the summation of square distances between  $P(t)$  and the point sample is minimized (Square Distance Minimization, SDM). Limitations of this algorithm are: (1) It specifically excludes self-intersecting curves. (2) A (time or space) complexity discussion is absent, (3) Both, the sampled curve and the recovered curve are required to be twice

differentiable. (4) Calculation of the distance from a point to a parametric curve is expensive since the latter normally has parametric degree larger or equal than three. This makes the SDM an expensive method.

[LYW05] presents an algorithm which seeks to fit B-Spline curves to a set of noisy points in  $R^2$  on which a Euclidean Minimal Spanning Tree (EMST) is calculated. The algorithm uses a band shaped support region to collect point subsets whose Principal Component Analysis trend determine the local tangents to the sought B-Spline curve. The authors claim that the algorithm handles sharp features and self-intersections of the curves. However, no clear algorithm is given to handle such situations. A fundamental flaw of such an algorithm is that a point sample of a 2D curves does not in general have the topology of a *tree* since cycles exist naturally in such sets. In addition, the algorithm uses a significant amount of heuristic constants that are not discussed in the paper and whose set values are not specified. The B-Splines are naturally smooth in spite of the fact that the point set is sampled from a non-smooth 2D curve. The paper does not discuss the computational complexity of the algorithm.

#### 2.4 Probability of Topological Properties of the Curve

[NSW08] proposes (without actual implementation or tests on data) an algorithm to probabilistically estimate topological properties of a manifold out of a noisy sample of it. [NSW08] specifically avoids the estimation of topological properties in the case of self-intersections. In this reference there is no comment on the computational expenses of the proposed algorithm.

#### 2.5 Probability of Geometrical Properties of the Curve

In [ULVH06] the local neighborhood of 3D curves in the space is sampled with noise  $\sigma$  and sampling density  $\rho$ . A local coordinate frame is associated with each neighborhood and the point sample is used to diagnose the curvature  $\kappa$  and torsion  $\tau$  of the 3D curve. The angle between the actual tangent vector and the

PCA-estimated tangent vector is found and plotted as a function of the radius  $r$  of the PCA support region. The value of the deviation angle varies with respect to the local curvature  $\kappa$ , noise  $\sigma$ , sampling density  $\rho$  and support region radius  $r$ . The article does not give a self-tuning algorithm (the radius is given as an absolute number), making the results unusable when the scale of the point set changes. The article does not discuss the computational complexity of the algorithm.

#### 2.6 Conclusion of Literature Review

The reviewed literature presents some salient features: (1) Implicit function calculations require large computational expenses. (2) Medial Axes methods produce inherently non-manifold constructs. (3) Curve synthesis as solution of Differential Equations is not adequate for the reasons given in section 2.2.3. (4) The fitting of higher degree parametric curves (Bezier, Spline, Nurbs) requires non-linear minimizations at every stage of the construction, which implies large computation costs.

We conclude that an explicit form of  $C$ , in Piecewise Linear form, is cheaper to determine and is acceptable for subsequent applications (in contrast with implicit forms). To find an explicit PL form of  $C$  we implement a method which is sensitive to the proximity of the self-intersection. In such a locality the point set to be fed to a PCA algorithm is the one included in ellipses rather than in disks. As it will be shown, such a variant allows the overcoming of the intersection neighborhood in a more robust manner.

Our ellipse-PCA algorithm will be tested on slice samples of  $C^2$  2-manifolds. As shown in Figure 1, this is one of the many possible scenarios where self-intersecting or nearly self-intersecting curves are present. As such cases are compounded by noise (present in all industrial sensors), we consider that efforts in this area are useful for the computer-aided design and manufacturing communities.



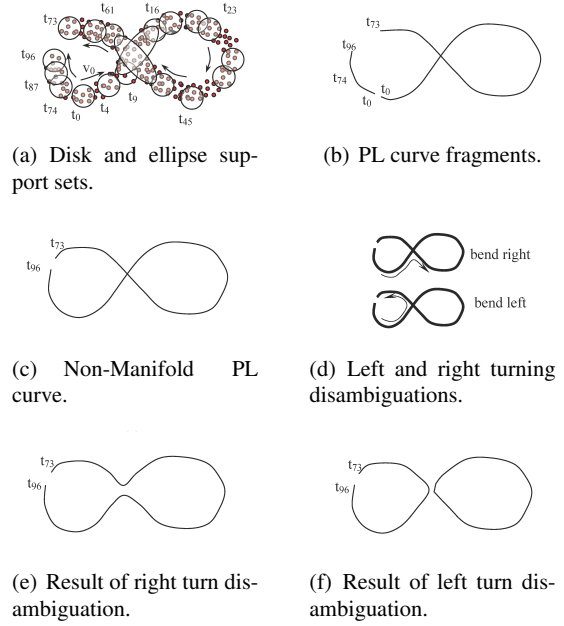
### 3 Methodology

The algorithm implemented in this article is based on statistical approximation of the local tangent of a curve  $C$  sampled with a noisy point set. The algorithms based on Principal Component Analysis do not perform well in curve self-intersection regions because the linear trend is lost there. To avoid this effect, we used a directional (elliptic) support region for the PCA algorithm. The ellipse becomes sharper as the linear trend of the point sample degenerates (for example at self-intersections). The elliptical support region has major axis in the direction of the last reliable vector tangent to the curve. This region excludes point samples in the direction perpendicular to such a tangent, thus ignoring the confusing trends at the self-intersection. In this manner the algorithm overcomes self-intersection regions and continues with the PL approximation of  $C$ .

Figures 4a-c display an intuitive functioning of the algorithm implemented, applied to the point sample of an open, self-intersecting curve  $C$  (Figure 4-a). To simplify the drawing, the point sample is suppressed from some figures, showing only the pursued PL approximation for  $C$ . Figure 4(a) shows that the algorithm starts in the neighborhood  $t_0$ , with the direction  $v_0$ , with a circular PCA-support region. The local approximation travels following the neighborhoods  $t_4, \dots, t_9, \dots, t_{73}$  (subscripts only indicate a *supposed* number of iteration). At the self-intersection, the PCA-support region becomes elliptical, so the algorithm is capable of crossing this zone, whose correlation coefficient  $\rho$  is fundamentally low if a circular support region had been used. The algorithm proceeds until it reaches  $t_{73}$ , where it finds a dead end, meaning that a connected subset of  $C$  has been found. The algorithm then revisits  $t_0$  with direction  $-v_0$ , rendering the sequence  $t_0, \dots, t_{74}, \dots, t_{87}, \dots, t_{96}$  (Figure 4(b)).

The post-processing part of the algorithm is fully known in computational geometry: it merges the PL approximations  $[t_0, \dots, t_{73}]$  and  $[t_0, \dots, t_{96}]$  into  $L = [t_{73}, \dots, t_0, \dots, t_{96}]$ , and then splits the self-intersecting PL curves determined by  $L$  as per the decisions in Fig-

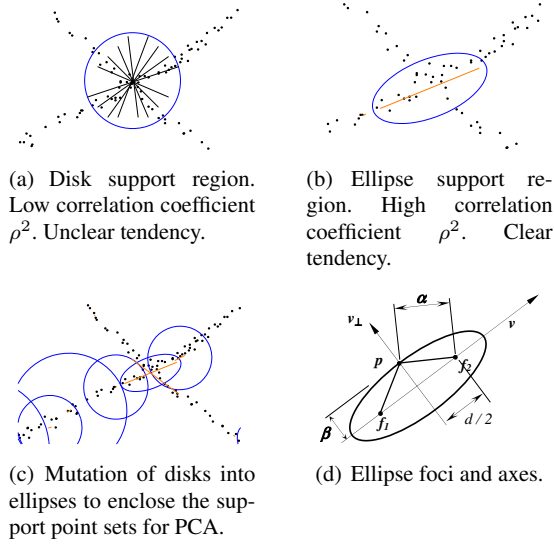
ure 4(d), resulting in the 1-manifolds in Figures 4(e) or 4(f). Either, right or left splitting produces topologically correct and geometrically different results. Since the (noisy, self-intersecting) sample does not allow for a canonical choice of either splitting (left or right) such a decision in the post-processing stage is one of mere convention.



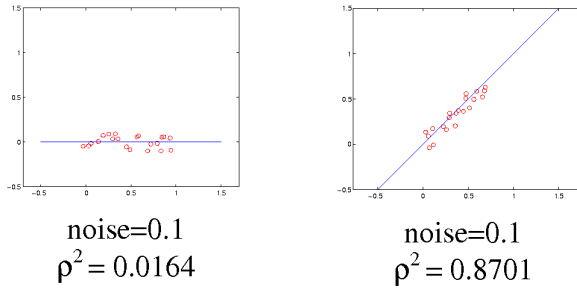
**Fig. 4** Execution of ellipse-based PCA algorithm.

#### 3.1 Measure of Goodness-of-fit for Principal Component Analysis

If a curve  $C$  has local curvature radius  $r$  and is point-sampled, there are upper and lower bounds in the length of a linear segment  $\overline{ab}$  which statistically approaches  $C$  at the point  $C(u)$ . If  $\overline{ab}$  is too long, the segment will not correctly approach the curve  $C$  at  $C(u)$ . If  $\overline{ab}$  is too small, only few sample points will be available to fit the segment  $\overline{ab}$ . In both cases, goodness-of-fit of the linear approximation is degraded. If  $C$  is planar, the Principal Component Analysis may be evaluated by using the linear regression correlation coefficient  $\rho = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y}$  with  $\rho \in [-1, 1]$ .  $|\rho| \approx 1.0$  and  $|\rho| \approx 0.0$  are associated with good and



**Fig. 5** Points and ellipses in a self-intersection region.



**Fig. 6** Dependency of Correlation Coefficients with slope  $m$ .

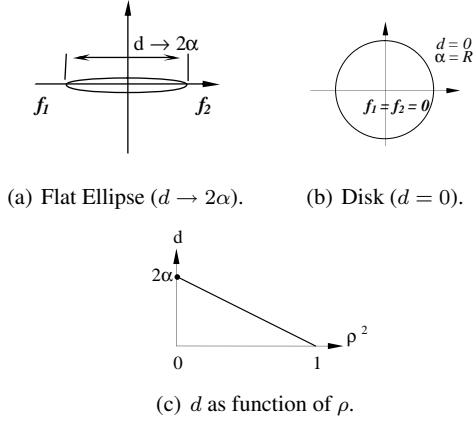
poor linear correlation, respectively. Caution must be exercised because linear regression parameters ( $m, b$  in  $y = mx + b$ ) are dependent on the particular coordinate system. Figure 3 shows that for the same level of noise the correlation coefficient  $\rho^2$  between  $x$  and  $y$  varies as  $m$  does. If we wish to compare *linearity* of point sets using the  $\rho^2$  values we should first achieve a fixed  $m$  (for example  $m = 1$ ) by rotating the point set and only then calculate its  $\rho^2$  value. In this manner we compare  $m$ -constant  $\rho^2$  values of different point sets. On the other hand, an advantage of using the correlation coefficient to grade the linear regression is that the  $\rho^2$  value is bounded ( $\rho^2 \leq 1$ ). Our approach is to use the linear regression with its correlation coefficient by following these steps: (a) Calculate a tentative linear

trend  $y = m_0.x + b_0$  for the local point set. (b) Rotate the local point set to get a slope of  $45^\circ$  ( $m \approx 1$ ). (c) Calculate  $\rho$  with the standard linear regression formulas. (d) Rotate back the measured  $m$  if needed. A number of iterations (limited to 10) is used to improve the value  $\rho$  by varying the shape of the ellipse enclosing the local point set. In this manner, we use the advantages of 2D linear regressions and neutralize its dependence on  $m$ . This heuristic worked correctly, as discussed in the Results section.

### 3.2 Circular vs. Elliptical Support Regions

Previous algorithms for curve reconstruction avoid addressing the topic of self-intersecting curves due to the effect shown in figure 5(a). At the self-intersection neighborhood, the identification of the local tangent becomes difficult and the curve reconstruction goes astray. This happens because the PCA analysis applied to an star-shaped point set will render lines in any direction of the  $R^2$  plane, accompanied of the correlation coefficient  $\rho$  being very low.

The algorithm reported in this article detects such low correlation regions and varies the shape of the support region for the PCA from round to elliptic (see Figure 5(b)), thus incrementing the correlation coefficient  $\rho$ . In our algorithm, ellipticity increases together with least square fitting error. The elliptic region has the major axis in the direction of the last reliable curve tangent  $v$  identified in the previous iteration (Figure 5(d)). This support region has the advantage of ignoring distracting points by having a tight span in the  $v_\perp$  direction, orthogonal to  $v$ . The ellipse is defined as the locus of points  $p$  such that  $|p - f_1| + |p - f_2| = 2\alpha$ . Our algorithm sets  $\alpha = 5 * \delta$ , where  $\delta$  is the effective sampling interval of the device. This value is set to approximately include 10 sampled points in each local Principal Component Analysis. Such a heuristic has worked in a stable manner in our implementations. Therefore  $\alpha$  is a known value that is fixed in the algorithm.



**Fig. 7** Heuristic rule which increases  $d$  (i.e. flattening the ellipse,  $d \rightarrow 2\alpha$ ) as the correlation coefficient deteriorates ( $\rho \rightarrow 0$ ).

### 3.3 Heuristic to overcome intersection places

In general, an ellipse in  $R^2$  can be specified by the position of its foci  $f_1$  and  $f_2$  (Figure 5(d)) and the length of its major semi-axis  $\alpha$  as  $E(f_1, f_2, \alpha) = \{p \in R^2 : |p - f_1| + |p - f_2| = 2\alpha\}$ . In any ellipse, if  $d$  is the distance between the foci, then  $\beta^2 = \alpha^2 - (\frac{d}{2})^2$ .

Our claim is that in a well defined curve point sample a PCA circular support region would work fine. If the point set deteriorates, the support region must become an ellipse. If the linear correlation of the point set is poor ( $\rho^2 = 0$ ), we would like to have a strongly elliptical support region (large  $d$ , Figure 7(a)). If the linear correlation is good ( $\rho^2 = 1$ ) a circular disk ( $d = 0$ , Figure 7(b)) would provide a convenient support region. We propose a decreasing function  $d = 2\alpha(1 - \rho^2)$ , as in Figure 7(c). Therefore,  $d$  ranges between 0 (good PCA correlation) and  $2\alpha$  (poor PCA correlation).

### 3.4 Reconstruction Algorithm

The implemented algorithm takes a point set as in Figure 4(a) and returns a set of PL curve fragments as in Figure 4(b). This result is the fundamental one, because joining the PL fragments of  $C$  in  $PL\_Curve\_Set$  in a manifold manner is a standard procedure.

The algorithm is based on the heuristic proposed in section 3.3, which gradually mutates a circular into an elliptical support region for the Principal Component Analysis. A simplified version of it is presented as Algorithm 1.

The algorithm contains three nested WHILE iterations. The invariant of the WHILE in line 8 is that a set of PL curve fragments ( $PL\_Curve\_Set$ ) has been synthesized and that there exist neighborhoods of unused points which have not been considered yet ( $Q \neq \emptyset$ ). The invariant of the WHILE in line 12 is that a local PL fragment ( $local\_C$ ) is being threaded as long as a well defined tangent is identified along it. The invariant of the innermost WHILE (line 18) indicates that in a particular neighborhood, a PCA based on circular support regions has produced a low  $\rho$  indicator. Therefore, the support region is gradually flattened until either  $\rho$  surpasses a threshold or a specified number of trials is reached. In the first case, the algorithm proceeds to the next (in the direction of  $v$ ) neighborhood. In the second case, the algorithm recognizes the fact that no clear tangent has been identified and declares the local curve as finished. This WHILE iteration implements the heuristic discussed in section 3.3.

The elements of the queue  $Q$  (line 5) have the form  $[p, v]$  where  $p$  is a point near  $C$  and  $v$  is a (unit) vector tangent to  $C$  near  $p$ . Notice that  $-v$  is also tangent to  $C$  at  $p$ . Each element of  $Q$  indicates a place and direction for traversing the sample  $S$  for the recovery of a portion of  $C$ . If the queue  $Q$  is empty, the algorithm finishes. The built-in function  $[\rho, p, v] = pca(S)$  is used in order to perform the Principal Component Analysis of the point set  $S$  giving as a result the trend  $v$ , the center of mass  $p$ , and the correlation coefficient  $\rho$ .

The algorithm builds each curve fragment or local curve  $local\_C$  as long as there is a clearly computable vector tangent to  $C$  at each point  $p$ . If the tangent is clear, a circular-supported PCA calculation is enough (lines 15,16) to determine it. Otherwise, an ellipse-supported PCA is attempted (lines 18-24). If the ellipse-based PCA manages to overcome the self-intersection, the algorithm continues (lines 26,27) com-

---

**Algorithm 1** PCA-based reconstruction algorithm using ellipses.

---

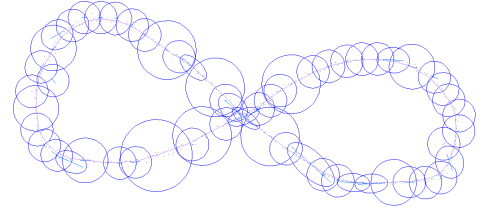
```

1: Comment:  $S$  is the sample of  $C$  with noise.
2: Comment:  $r$  is set to the sampling noise plus sampling distance.
3: Let  $p$  be any point in  $S$  such that  $S \cap B(p, r)$  presents a correlation coefficient  $\rho \approx 1$ .
4:  $[\rho, p_t, v_t] = \text{pca}(S \cap B(p, r))$ 
5:  $Q = \text{queue}([p_t, v_t])$ 
6:  $Q = \text{add}(Q, [p_t, -v_t])$ 
7:  $PL\_Curve\_Set = []$ 
8: while  $Q \neq \emptyset$  do
9:    $[p, v] = \text{discharge}(Q)$ 
10:   $local\_Cu = []$ 
11:   $\text{clear\_tangent} = \text{TRUE}$ 
12:  while  $\text{clear\_tangent}$  do
13:     $local\_C = [local\_C, p]$ 
14:     $p_t = p + \lambda * v$ 
15:     $local\_S = S \cap B(p_t, r)$ 
16:     $[\rho, p_t, v_t] = \text{pca}(local\_S)$ 
17:     $Num\_Trials = 1$ 
18:    while  $(Num\_Trials < Max\_Trials)$  and  $(\rho < Lower\_Bound)$  do
19:       $d = 2\alpha(1 - \rho^2)$ 
20:       $f1, f2 = p_t \pm \frac{d}{2} * \hat{v}$ 
21:       $local\_S = S \cap E(f1, f2, \alpha)$ 
22:       $[\rho, p_t, v_t] = \text{pca}(local\_S)$ 
23:       $Num\_Trials = Num\_Trials + 1$ 
24:    end while
25:    if  $(Num\_Trials < Max\_Trials)$  then
26:       $v = v_t$ 
27:       $p = p_t$ 
28:    else
29:       $\text{clear\_tangent} = \text{FALSE}$ 
30:      Let  $p$  be an unused point in  $S$  whose neighbor point set inside a disk  $B(p, r)$  has  $\rho \approx 1$ .
31:      if  $p$  is found then
32:         $[\rho, p_t, v_t] = \text{pca}(S \cap B(p, r))$ 
33:         $Q = \text{add}(Q, [p_t, v_t])$ 
34:         $Q = \text{add}(Q, [p_t, -v_t])$ 
35:      end if
36:    end if
37:  end while
38:   $PL\_Curve\_Set = [PL\_Curve\_Set, local\_C]$ 
39: end while
40: Comment:  $PL\_Curve\_Set$  is the set of PL fragments approximating  $C$ 

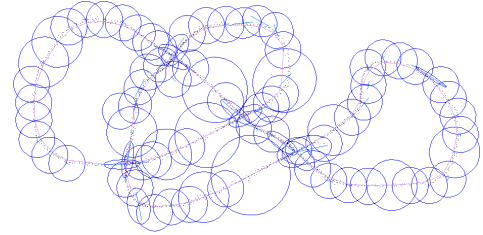
```

---

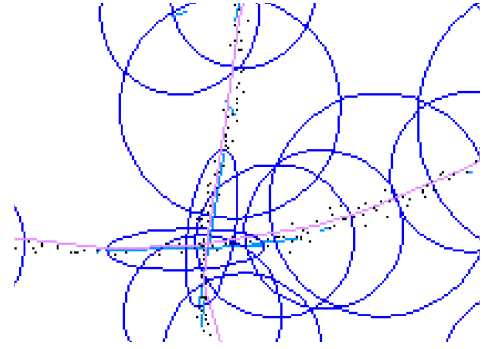
pleting the curve fragment  $local\_C$ . If the elliptic support regions at both extremes of  $local\_C$  lead the PCA to fail identifying a clear tangent vector, the algorithm stops processing the current fragment  $local\_C$  (line 29). In this situation, the algorithm seeks unused neighborhoods of the point set that may originate another fragment  $local\_C$  when taken as seed in later iterations (line 30). If such neighborhoods are found, they are input in the queue  $Q$  (lines 32-34).



(a) One self-intersection.



(b) Four self-intersections.



(c) Detail of self-intersection.

**Fig. 8** Ellipse-PCA Processing Self-intersecting Curves

### 3.5 Complexity of the Algorithm

Let us assume that the number of points in the sample is  $N$ . In Algorithm 1, either one of lines 21 or 22 contains instructions whose worst case cost is  $O(N)$ . Since such

instructions are inside threefold nested WHILE loops whose worst case complexity is  $O(N)$  each, we conclude that the *worst case* complexity for such an algorithm is  $O(N^4)$ . It is important to observe that in our approach no additional memory or time resources are spent in building or maintaining collateral data structures or in pre-processing the data.

In this regard, the literature reviewed is uniformly incomplete in that run-time complexities are given without reporting resources devoted to (a) collateral data space and (b) pre-processing. Since our evaluation  $O(N^4)$  is a worst-case estimate and specifically rules out the need of expenses (a) and (b) above, it is not comparable with other evaluations which concentrate on *expected* cases and neglect to take into account the expenses caused by (a) and (b) (see section Literature Review and its Conclusions).

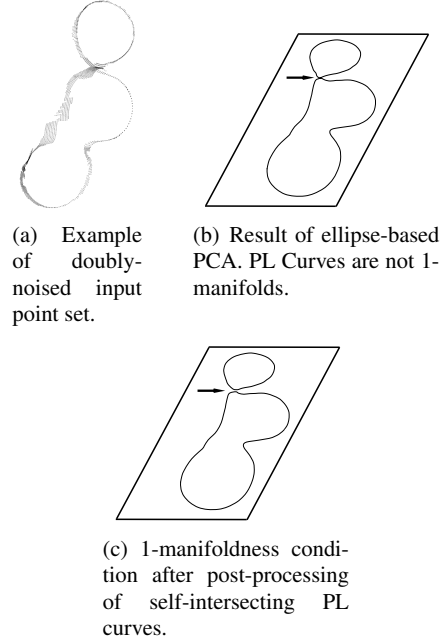
#### Integration of PL Fragments.

This part of the algorithm is well known in computational geometry and it is not dominant in terms of complexity, as compared with Algorithm 1. The statement of this post-processing is as follows. Given an unordered set of PL curve fragments  $PL\_Curve\_Set = \{c_1, c_2, \dots, c_m\}$  that approximate the point set  $S$  (Figure 4(b)), two steps are required: (1) the joining of  $c_i$  and  $c_j$  when their endpoints are closer than a distance  $\delta_s$  (Figure 4(c)), and (2) the splitting of the paths resulting from (1) to avoid self-intersections, by using the decision criteria in Figure 4(d). The final result appears in Figures 4(e) and 4(f). The processes (i) and (ii) considered together have complexity  $O(N^2)$ .

## 4 Results

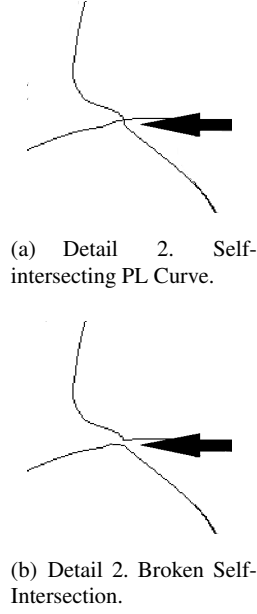
Figure 8 shows the functioning of the proposed algorithm, applied to a closed curve with self-intersections. It can be seen that the circular support regions ( $|f_1 - f_2| \rightarrow 0$ ) at manifold neighborhoods become flattened ellipses ( $|f_1 - f_2| \rightarrow 2\alpha$ ) at non-manifold neighborhoods.

Outliers (points sampled with unusually large sampling noise) do not participate in the execution. The algorithm is robust in this sense, since it flattens the ellipse as a response to the inclusion of such outliers in the PCA. As a result, such points are expeditiously ignored. The whole algorithm stops when most of the points (near 100%) have been considered in at least one ellipse or disk. The tests run provide strong evidence that this stopping criterion does not affect the efficacy of the algorithm.



**Fig. 9** Hand data set. Noisy point set (a) along with its processing (b) and post-processing into a disconnected 1-manifold (c).

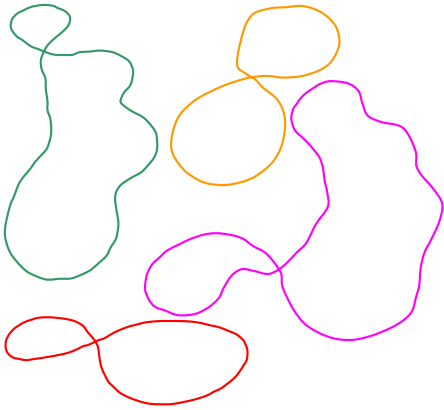
In surface reconstruction from slice samples it is not uncommon to have one or more (usually non-consecutive) missing slice samples. In such a case, it is appealing to replace the missing slice sample  $i$  by the projection of the point data from slices  $i - 1$  and  $i + 1$  onto the plane corresponding to it. An example of such a projected point set is depicted in Figure 9(a). It must be pointed out that such a point set presents the additional difficulty of having noise stemming from the point projection, besides the basic sampling noise. Figure 9(b) presents the result of the application of Algorithm 1 to such a point set.



**Fig. 10** Detail of broken self-intersection of Figure 9.

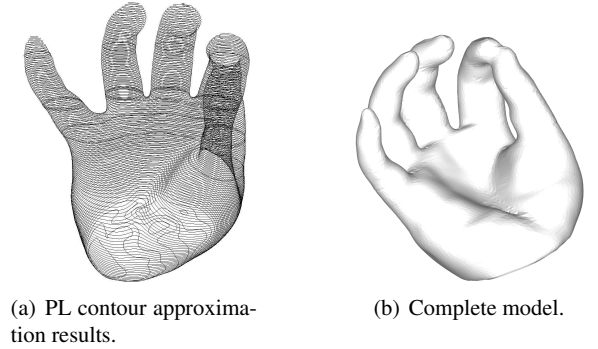
A standard algorithm for separation of non-manifold curves into manifold ones produces the separated contours (Jordan curves in  $R^2$ ) in Figure 9(c).

Figures 10(a) and 10(b) present a zoom on particular details of Figures 9(b) and 9(c), respectively. Figure 10(a) presents a neighborhood of self-intersecting PL curves obtained with Algorithm 1. Such neighborhood with the self-intersection removed is shown in Figure 10(b). Additional results of self-intersecting cross cuts of the *Hand* data set are displayed in Figure 11.



**Fig. 11** Additional Examples of Self-intersecting contours in the *Hand* data set.

Algorithm 1 was tested on the *Hand* data set, made of slice noisy point samples of an object. The result of applying Algorithm 1 to all slices of such a data set is displayed in Figure 12(a). The slices containing self-intersections are the darker ones. The PL contours belonging to the slices were then fed to well known algorithms ([RCG<sup>+</sup>05] or [Gei93]) to reconstruct the surface. Figure 12(b) presents the surface for the *Hand* point set including the whole set of cross sections.

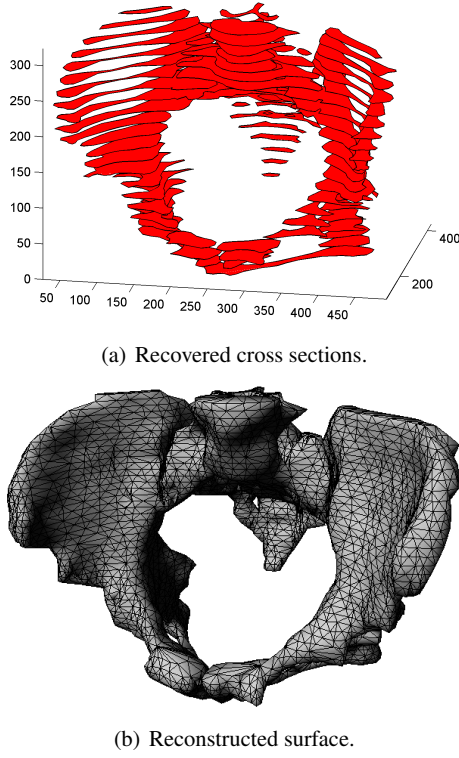


**Fig. 12** Algorithm results for the *Hand* data set. Rendered surfaces.

#### 4.1 Data Set 2. Pelvis

To illustrate here the robustness of the proposed method, a near self-intersecting contour set was extracted from the *Pelvis* data set (Figure 13) and added with noise levels  $[1\delta_n, 2\delta_n, 3\delta_n, 4\delta_n, 5\delta_n, 6\delta_n]$  ( $\delta_n$  is the nominal sampling interval). The algorithm was then run using such point sets (see Figure 14). The ellipse sequences of our algorithm are displayed in the left column, while the recovered contours (before splitting) appear in the right column.

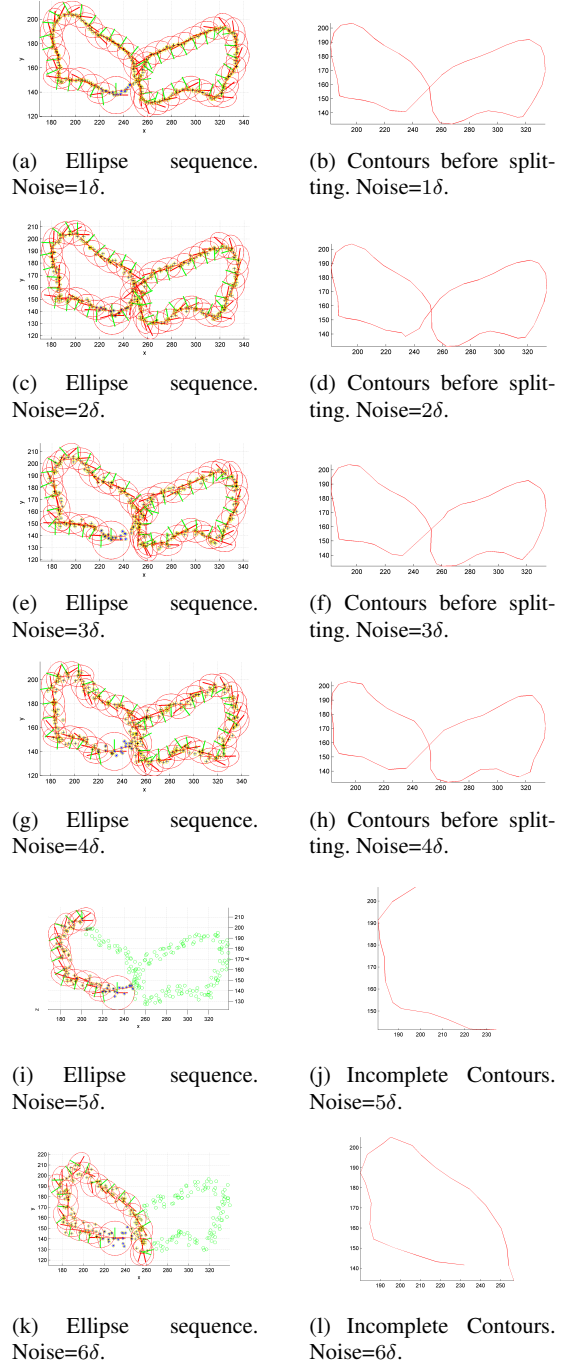
Notice that the algorithm is able to fit *one* PL curve the whole point set at once for noise levels  $[1\delta_n, \dots, 4\delta_n]$ , showing similar performance for such cases. For noise levels  $5\delta_n$  or  $6\delta_n$  the algorithm fits several PL curves to the point set, which must be integrated as in Figures 4(b) and 4(c). Such actions are discussed in the section “Integration of PL Fragments.”



**Fig. 13** Reconstructed Contours and Surfaces for the *Pelvis* data set.

#### 4.2 Data Set 3. Skull

The Skull data set consists of 64 slices. Each slice contains nested and/or disconnected contours. Some levels have contours which are nearly self-intersecting, as seen in Figure 15. A particular slice of such a data set contains a contour as the one shown in Figure 15(a). Figures 15(b), 15(c) and 15(d) show point samples of the contour with sampling noise of  $1\delta$ ,  $3\delta$  and  $6\delta$ , respectively. It is evident that the point samples, even for low noise, reflect a near-self-intersecting curve. Likewise, since the mentioned contours contain very fine geometric detail, the frequency content of them is quite high. As a consequence of the Nyquist principle, the minimal sampling distance needed to recover such contours is also very small (half of the size of the smallest geometric feature to be captured). This circumstance immediately reflects on the tightness of the sample, noise and the progression of the ellipse evolution, being

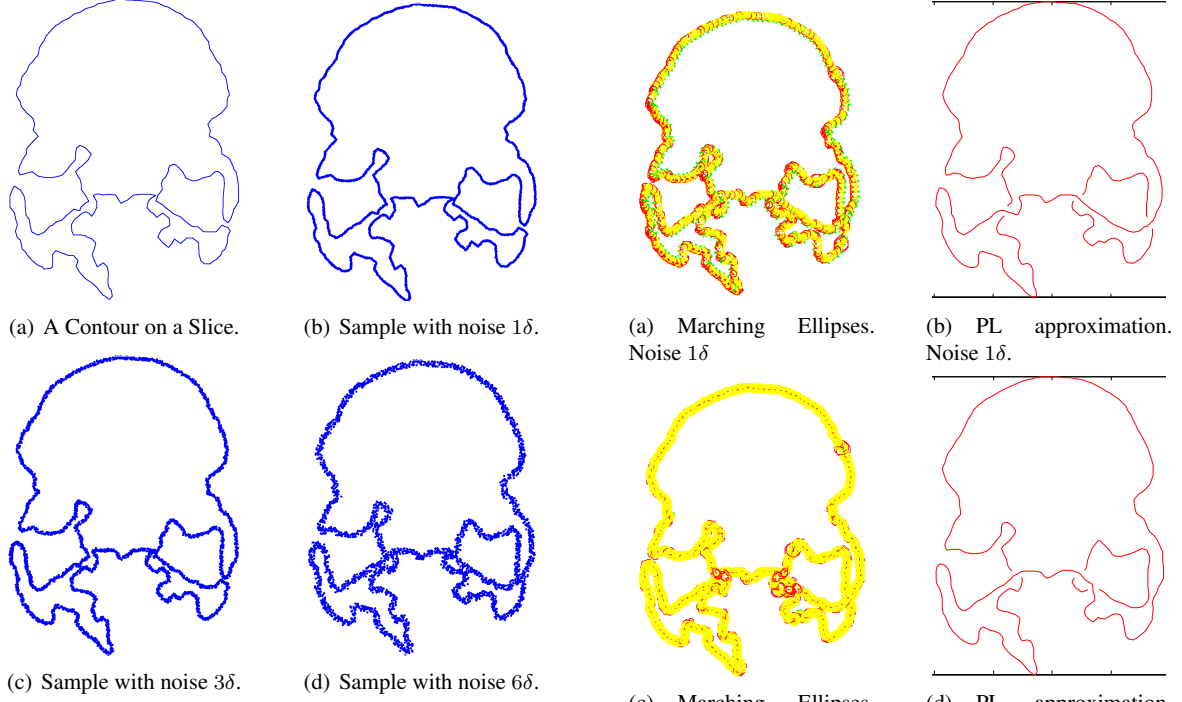


**Fig. 14** Algorithm Performance. Slice data from *Pelvis* data set.

all of them very different as compared with the *Pelvis* data set.

The algorithm 1 is run using the data sets of Figures 15(b), 15(c) and 15(d). The evolution of the ellipse algorithm for each noise level is displayed in Figures 16(a), 16(c) and 16(e), respectively. The inherent diffi-



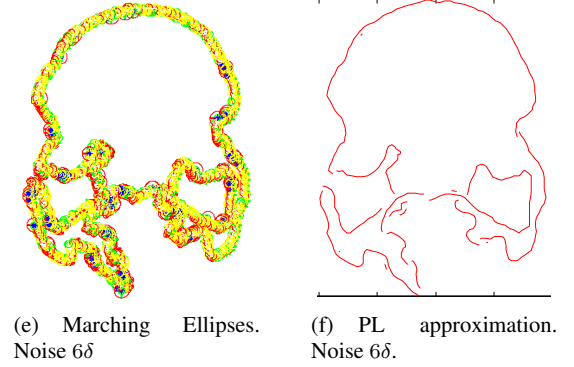


**Fig. 15** Noisy samples of a contour in the *Skull* data set.

culty in the contour processed produces a much tighter sequence of ellipses than the ones recorded in Figure 14 (*Pelvis* data set). Figures 16(b), 16(d) and 16(f) illustrate the result of the execution of Algorithm 1. The results of the recovery of individual PL approximations of  $C$  from the random noisy point sets are satisfactory for the noise levels  $1\delta$  and  $3\delta$  but fail for noise level  $6\delta$ .

Notice that the individual PL curves are not exactly manifolds because they are self-intersecting. Moreover, they are still fragmented. Therefore, the individual PL curves are still to be appended together as in Figures 4(b) and 4(c), and as discussed in section “Integration of PL Fragments”. Next, the self-intersecting PL curves must be split at the self-intersections as shown in Figure 4(d).

Figure 17(a) displays the Skull contour set as obtained by the iterated application of the algorithm discussed in the present article. Then, a surface reconstruction algorithm from parallel planar contours ([RCG<sup>+</sup>05] or [Gei93]) was executed rendering the surface shown in Figure 17(b).

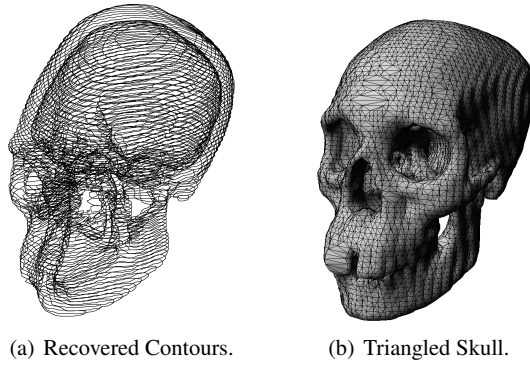


**Fig. 16** Results. *Skull* data set.

## 5 Conclusions and Future Work

This article has presented an algorithm (together with its testing scenario) for the reconstruction of a planar curve  $C$  out of a noisy sample of it. The algorithm has the following characteristics: (1) It constructs a Piecewise Linear approximation of  $C$ . (2) It is able to recover self-intersecting or near self-intersecting curves rendering a decomposition of them into disjoint 1-manifolds. (3) It performs local Principal Component Analysis using support regions whose form mutates from circular disks, in neighborhoods where there are no self-





**Fig. 17** Contours and reconstructed surface for the *Skull* data set.

intersections, to flat ellipses near the self-intersections. (4) It does not require collateral data structures or pre-processing, and its worst-case complexity is  $O(N^4)$  where  $N$  is the number of points sampled on  $C$ .

We consider this *worst* case complexity as non-comparable with the complexity reported by some authors addressing the same problem, since they estimate *expected* cases and fail to account for the *computing time* and *space* spent in the collateral data structures and pre-processing present in their algorithms. It must be pointed out that the vast majority of the literature reviewed does not address computational expenses of their proposed algorithms.

Future work in the topic of curve reconstruction includes the reconstruction of non-planar curves, and the lowering of complexity of reconstruction with implicit forms of higher degree (Spline, Bezier, NURBs).

## References

- [AK03] Sanjeev Arora and Subhash Khot. Fitting algebraic curves to noisy data. *J. Comput. Syst. Sci.*, 67(2):325–340, 2003.
- [BF95] Jules Bloomenthal and Keith Ferguson. Polygonization of non-manifold implicit surfaces. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 309–316, New York, NY, USA, 1995. ACM.
- [Blo88] J. Bloomenthal. Polygonization of implicit surfaces. *Comput. Aided Geom. Des.*, 5(4):341–355, 1988.
- [Blo94] Jules Bloomenthal. An implicit surface polygonizer. In *Graphics Gems IV*, pages 324–349. Academic Press, 1994.
- [CFG<sup>+</sup>05] Siu-Wing Cheng, Stefan Funke, Mordecai Golin, Piyush Kumar, Sheung-Hung Poon, and Edgar Ramos. Curve reconstruction from noisy samples. *Comput. Geom. Theory Appl.*, 31(1-2):63–100, 2005.
- [DK99] Tamal K. Dey and Piyush Kumar. A simple provable algorithm for curve reconstruction. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 893–894, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.
- [Gei93] B. Geiger. Three-dimensional modeling of human organs and its application to diagnosis and surgical planning. Technical Report RR-2105, 1993.
- [Kég99] B. Kégl. *Principal Curves: Learning, Design, And Applications*. PhD thesis, Concordia University, Montreal, Canada, 1999.
- [KK02] B. Kégl and A. Krzyzak. Piecewise linear skeletonization using principal curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(1):59–74, January 2002.
- [KZ04] Jan Klein and Gabriel Zachmann. Point cloud surfaces using geometric proximity graphs. *Computers and Graphics*, 28(6):839–850, 2004.
- [kZOMK00] Hong kai Zhao, Stanley Oshery, Barry Merrimany, and Myungjoo Kangy. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, 80:295–319, 2000.
- [Lee00] I.K. Lee. Curve reconstruction from unorganized points. *Computer Aided Geometric Design*, 17(2):161–177, 2000.
- [LYW05] Yang Liu, Huaiping Yang, and Wenping Wang. Reconstructing b-spline curves from point clouds—a tangential flow approach using least squares minimization. In *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005*, pages 4–12, Washington, DC, USA, 2005. IEEE Computer Society.
- [LZJ<sup>+</sup>05] DanFeng Lu, HongKai Zhao, Ming Jiang, ShuLin Zhou, and Tie Zhou. A surface reconstruction method for highly noisy point clouds. In *Third International Workshop on Variational, Geometric and Level Set Methods in Computer Vision, VLSM*, pages 283–294, 2005.
- [MD07] Asish Mukhopadhyay and Augustus Das. Curve reconstruction in the presence of noise. In *CGIV '07: Proceedings of the Computer Graphics, Imaging and Visualisation*, pages 177–182, Washington, DC, USA, 2007. IEEE Computer Society.

- 
- [NSW08] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete Computational Geometry*, 39(1):419–441, 2008.
- [Nyq28] Harry Nyquist. Certain topics in telegraph transmission theory. *Bell System Technical Journal*, 47, 1928.
- [Nyq02] H. Nyquist. Certain topics in telegraph transmission theory. *Reprint as classic paper in: Proc. IEEE*, 90(2):617–644, 2002.
- [OF00] S. Osher and R. Fedkiw. Level set methods: An overview and some recent results. Technical report, University of California Los Angeles, Stanford University, 2000.
- [OS88] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
- [PMG04] M. Pauly, N. J. Mitra, and L. Guibas. Uncertainty and variability in point cloud surface data. In M. Alexa and S. Rusinkiewicz, editors, *Symposium on Point-Based Graphics*, pages 77–84. Eurographics, June 2004.
- [RCG<sup>+</sup>05] O. Ruiz, C. Cadavid, M. Granados, S. Peña, and E. Vásquez. 2D shape similarity as a complement for Voronoi–Delone methods in shape reconstruction. *Elsevier J. on Computers and Graphics*, 29(1):81–94, February 2005. ISSN: 0097-8493.
- [RVC07] O. Ruiz, C. Vanegas, and C. Cadavid. Principal component and voronoi skeleton alternatives for curve reconstruction from noisy point sets. *Journal of Engineering Design*, 18(5):437 – 457, October 2007. ISSN: 1466-1837 (electronic) 0954-4828 (paper).
- [Sha49] C. E. Shannon. Communication in presence of noise. *Proc. Institute of Radio Engineers*, 37(1):10–21, 1949.
- [Sha98] C. E. Shannon. Communication in presence of noise. *Reprint as classic paper in: Proc. IEEE*, 86(2):447–457, 1998.
- [TZCO09] Andrea Tagliasacchi, Hao Zhang, and Daniel Cohen-Or. Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph.*, 28(3):1–9, 2009.
- [ULVH06] Ranjith Unnikrishnan, Jean-François Lalonde, Nicolas Vandapel, and Martial Hebert. Scale selection for the analysis of point-sampled curves. In *3DPVT*, pages 1026–1033, 2006.
- [UR08] D. Uribe and O. Ruiz. 2d curve reconstruction with heat transfer differential equations. Technical report, EAFIT University, CAD CAM CAE Laboratory, Nov. 2008.
- [VVK01] J. J. Verbeek, N. Vlassis, and B. Kröse. A soft k-segments algorithm for principal curves. In *Proc. Int. Conf. on Artificial Neural Networks*, pages 450–456, Vienna, Austria, August 2001.
- [WPL06] Wenping Wang, Helmut Pottmann, and Yang Liu. Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graph.*, 25(2):214–238, 2006.
- [ZOF01] Hong-Kai Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01)*, page 194, Washington, DC, USA, 2001. IEEE Computer Society.