# Human Perception Faithful Curve Reconstruction Based on Persistent Homology and Principal Curve

Yu Chen, Hongwei Lin, and Yifan Xing
School of Mathematics Science, Zhejiang University
No. 866, Yuhangtang Road, Hangzhou, China
chenyu.math@zju.edu.cn, hwlin@zju.edu.cn, 22335069@zju.edu.cn

## Abstract

**Reconstructing curves that align with human visual perception from a noisy point cloud presents a significant challenge in the field of curve reconstruction. A specific problem involves reconstructing curves from a noisy point cloud sampled from multiple intersecting curves, ensuring that the reconstructed results align with the Gestalt principles and thus produce curves faithful to human perception. This task involves identifying all potential curves from a point cloud and reconstructing approximating curves, which is critical in applications such as trajectory reconstruction, path planning, and computer vision. In this study, we propose an automatic method that utilizes the topological understanding provided by persistent homology and the local principal curve method to separate and approximate the intersecting closed curves from point clouds, ultimately achieving successful human perception faithful curve reconstruction results using B-spline curves. This technique effectively addresses noisy data clouds and intersections, as demonstrated by experimental results.**

***Keywords:** Intersecting closed curves, Curve reconstruction, Principal curve, Persistent homology*

## 1. Introduction

Curve reconstruction is a fundamental problem that focuses on recovering a complete and accurate curve from a set of discrete points (a point cloud). The reconstructed curve is essential for representing shapes, modeling objects, and analyzing data across diverse fields. Given its wide-ranging applications, the problem of curve reconstruction has been extensively explored in numerous fields, establishing itself as a versatile and effective tool with applications across various disciplines. Its ability to recover complete and accurate curves from discrete data makes it an essential technique in many scientific and engineering domains [1, 37, 38]. However, there are a lot of complicated situations to deal with in curve reconstruction. An ongoing challenge in the realm of curve reconstruction involves aligning the reconstructed curves with human visual perception. According to cognitive psychology, human visual perception adheres to various theories, such as the Gestalt principles, which encompasses several principles guiding visual perception [26, 27, 43]. One essential principle, known as good continuation, posits that points forming straight lines or smooth curves, when connected, are perceived as belonging together, and these lines or curves are typically seen as connected in the smoothest way possible [16]. This principle aids in directing our eyes in a specific direction while connecting the points. Therefore, a curve that connects or fits the points in the smoothest manner, adhering to the Gestalt principle of good continuation, can be considered a curve faithful to human perception.

In the field of curve reconstruction, a common application of the good continuation principle involves reconstructing intersecting curves from point clouds, ensuring that the reconstructed curves adhere to the good continuation principle when crossing the intersection points. This process involves sequentially identifying each curve from the initial point cloud, with particular attention to accurately identifying points near the intersection based on the good continuation principle. Subsequently, each curve is reconstructed using an appropriate method. Since intersections are critical features of curves, their proper treatment is essential for ensuring that the reconstructed curves maintain correct topology. This is crucial in various applications, including trajectory reconstruction, path planning, and computer vision [25, 40, 41]. Moreover, in practice, sampling point clouds often contain noise, thereby complicating the reconstruction of intersecting curves. Consequently, in such scenarios, the curve reconstruction method should not only exhibit robustness to noise but also accurately address intersections in line with human visual perception.

In this paper, we present a method that utilizes persistent homology and principal curves to address the challenge of reconstructing closed curves aligned with human visual perception from noisy point clouds. To highlight the consistency of our approach with the good continuation principle

of visual perception, we will concentrate on the task of reconstructing intersecting closed curves from point clouds. It is important to note that our method can also be utilized to reconstruct closed curves without intersections, an area that has been previously investigated by various methods [31]. However, none of the prior methods have successfully automated the separation and reconstruction of intersecting closed curves, while our method is possible to obtain reconstructed curves that are consistent with the shape of the original sampled curves and are aligned with human visual perception in most cases.

Specifically, when presented with a noisy point cloud sampled from some intersecting closed curves, the propose method provides an automatic way to separate and reconstruct each curve from the point cloud, such that the reconstructed curves adhere to the good continuation principle of visual perception. This process entails initially providing a topological understanding of the sampling point cloud using persistent homology. Subsequently, the local principal curve method is employed to compute principal curves that approximate the original sampling curves. Finally, through the selection and smooth fitting of the derived principal curves, all curves can be successfully reconstructed. Fig. 1 shows a flowchart of the entire curve reconstruction procedure. In the following sections, we will see that the information provided by persistent homology allows for a sufficient topological understanding of sampling point clouds and the automatic determination of the parameters required for computing the principal curve, enabling the extraction of closed intersecting curves without additional human intervention. In conclusion, the contributions of this paper are as follows:

- Persistent homology is utilized to make topological understanding of point clouds.
- The selection of starting points and bandwidth parameter of local principal curves can be automated.
- Curves that align with the Gestalt good continuation principle can be reconstructed.

The remainder of this paper is organized as follows. First, we outline some related work on curve reconstruction, computational topology and principal curve in Section 2. Then we introduce certain concepts and properties concerning persistent homology and local principal curves as preliminary discussions in Section 3. The proposed curve reconstruction algorithm using persistent homology and local principal curve is introduced in Section 4. In Section 5, we show the experimental results and have a discussion. Finally, we conclude this study in Section 6.

## 2. Related Work

### 2.1. Curve reconstruction methods

Now we enumerate some work related to reconstructing curves from sampling point clouds that are noisy or have non-manifold structures. Many approaches are able to address noisy sampling cases for reconstruction. Levin [29] tackles this issue by employing moving least squares (MLS) techniques to interpolate and denoise the input points, thereby restoring the original smooth curve. Another method, FITCONNECT [33], is designed to handle noisy samples in curve reconstruction by fitting points with circular arcs to local neighborhoods, allowing for the reconstruction of noisy clusters and the denoising of the local fits to the estimated noise extent. Furthermore, STRETCH-DENOISE [32] enhances the FITCONNECT technique for denoising. It models the recovered curve connectivity separately from the high-frequency residuals and uses this information to shift point positions by minimizing angles between edges in the least-squares sense. However, none of the aforementioned methods can resolve the problem of reconstructing non-manifold curves, and thus are unsuitable for dealing with the reconstruction of intersecting curves.

Additionally, numerous methods are dedicated to reconstructing curves with self-intersections, or more generally, non-manifold curves. Lenz [28] proposes an algorithm for reconstructing curves with sharp corners and self-intersections, initiating the process with a seed edge between the two closest points and connecting edges by tracing along with a probe shape. However, this method is sensitive to noise and sampling conditions. PEEL [35] is another algorithm specifically designed for reconstructing curves with self-intersections. It begins by denoising the point cloud and reconstructing the curve with a vertex degree constraint of a maximum of three. Subsequently, post-processing is employed to identify and restore self-intersections by exploring potential intersections at vertices through one-ring Delaunay neighborhoods. However, self-intersections are not automatically identified, and the approximation of the curves may be coarse since the algorithm relies on two user-selected parameters. De Goes et al. [7] address this issue by formulating an optimal transport problem and minimizing the total cost to reconstruct non-manifold curves, which can also handle noisy point clouds. However, when it comes to the problem of reconstructing different intersecting curves, none of these methods are able to automatically separate and extract each curve.

More details on curve reconstruction are available in [31]

### 2.2. Computational topology

Computational topology [5, 10] is a new branch of mathematics that is dedicated to extracting the topological structure of data. Key research in this field encompasses topolog-

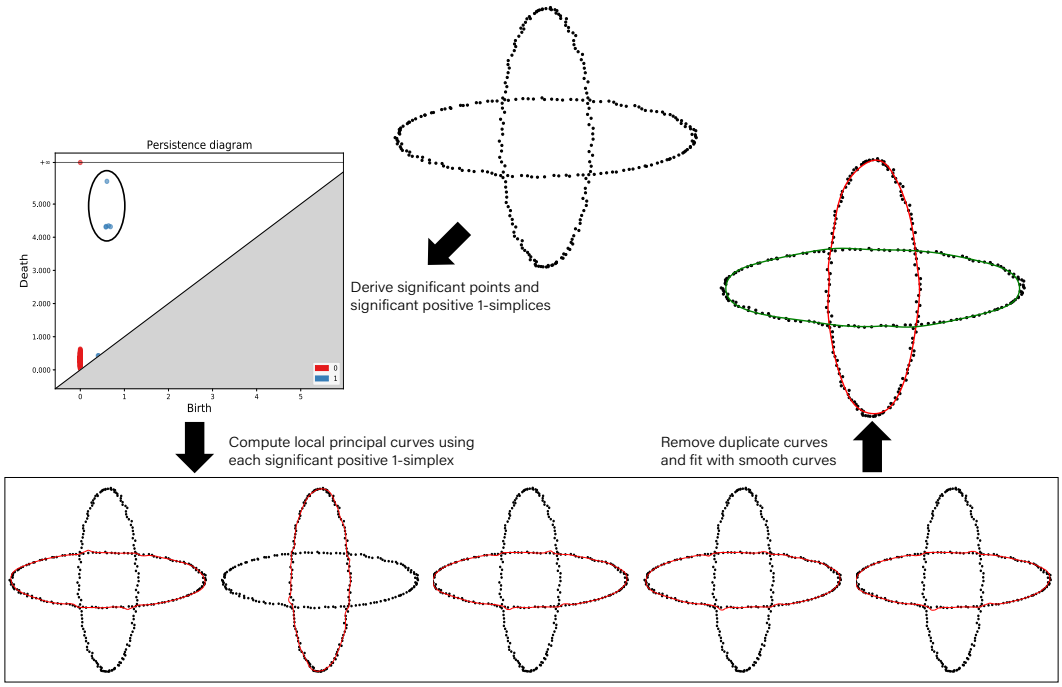



Figure 1. Flowchart of our curve reconstruction algorithm. The point cloud is sampled from two intersecting ellipses, and the number of significant points in 1-PD is 5 due to the new appearing loops. By considering all the significant positive 1-simplices, we can compute five local principal curves, while four of them represent the same ellipse. After removing duplicate curves, two ellipses can be successfully reconstructed.

ical data analysis (TDA) [42], an approach to comprehend the distributional characteristics of a dataset by computing its topology, which has been successfully employed to many fields, such as biomedicine [39], computer vision [2] and machine learning [36]. This approach typically involves constructing simplicial complexes and filtrations from the datasets, and employing pivotal tools such as persistent homology to analyze their topological structures. The persistent homology [10], serving as a fundamental tool for topological data analysis, acquires topological features within the point cloud through a filtration procedure, enabling the recognition of topological features within point cloud data [4], such as the number of connected components and loops. In particular, it provides a multiscale description of homology by constructing filtrations from the original point cloud. Consequently, when a point cloud is sampled from closed curves, persistent homology can offer extensive information about the sampling curves. In this paper, we employ persistent homology to offer a topological understanding of a sampling point cloud, which yields essential information for curve reconstruction.

### 2.3. Principal curve

The principal curve is first introduced by Hastie and Stuetzle [19], which can move along the "middle" of the data point, offering an one-dimensional nonlinear approximation of the data cloud [18, 19, 24, 34]. It effectively capturing the underlying data structure and providing a concise one-dimensional nonlinear summary of the data.. Furthermore, this concept has been extended to Riemannian manifolds [20, 22], and has found applications in many fields, such as image processing [23], traffic patterns analysis [12] and data clustering [6, 44]. However, parameter tuning can pose a challenge for principal curves [3]. Dealing with a point cloud representing more than two intersecting curves is also challenging. One possible approach to address this is the use of principal graphs [23], which provides an approximation of the data, including cases with self-intersected curves, but automatic separation of two or more intersected curves remains challenging. An alternative possible approach to address this problem is the local principal curve method [13, 14], a variant of the principal curve that relies on localized principal component analysis, thus paying more attention to localized analysis of the data clouds. This method facilitates the identification of closed curves as well as cases of local curve crossings. However, it necessitates the careful selection of several parameters, such as the original point of the curve and the bandwidth parameter. Without suitable parameter selections, it is difficult to generate principal curves that fit the shape of the point clouds.

## 3. Preliminaries

In this part, some key concepts about persistent homology and local principal curve used in this paper are presented.

### 3.1. Persistent homology

We begin by introducing some concepts and properties of persistent homology. An $n$-*simplex* is defined as the convex hull formed by $n+1$ affinely independent points $\{u_0, u_1, \ldots, u_n\}$ in the Euclidean space $\mathbb{R}^N$, denoted as $[u_0, u_1, \ldots, u_n]$. Geometrically, an $n$-simplex can be represented as a geometric model, such as vertex (0-simplex), line segment (1-simplex), triangle (2-simplex), and tetrahedron (3-simplex).

A *simplicial complex* $K$ is a collection of simplices that satisfies specific properties: every face of a simplex in $K$ is also in $K$, and the intersection of any two simplices in $K$ is a face of both of them [11]. The dimension of a simplicial complex $K$ is defined as the maximum dimension among all the simplices in $K$. Let $K$ be a simplicial complex and $n$ its dimension. An $n$-*chain* is a sum of $n$-simplices in $K$, denoted by $c = \sum a_i \sigma_i$, where $\sigma_i$ represents an $n$-simplex, and here $a_i$ is its coefficient belonging to $\mathbb{Z}_2$. With the addition operation, the $n$-chains form the $n$-chain group, denoted by $C_n(K)$. The *boundary* for a $n$-simplex $\sigma = [x_0, x_1, \ldots, x_n]$ is given by

$$\partial_n \sigma = \sum_{j=0}^{n} (-1)^j [x_0, \ldots, \hat{x}_j, \ldots, x_n],$$

where $\hat{x}_j$ denotes the omission of $x_j$. Let $Z_n(K) = \mathrm{Ker}(\partial_n)$ be the group of $n$-*cycle* and $B_n(K) = \mathrm{Im}(\partial_{n+1})$ the group of $n$-*boundary*. The $n$-*th homology group* of $K$ is defined as the quotient group $H_n(K) = Z_n(K)/B_n(K)$.

The *Vietoris–Rips (VR) Complex* [11] is a type of simplicial complex, whose construction is defined by following rules:

**Definition 1.** *(VR Complex) For any $\varepsilon > 0$, a finite subset $\{x_0, x_1, \ldots, x_n\} \subseteq P$ of the space $\mathbb{R}^N$ forms a simplex $[x_0, x_1, \ldots, x_n]$ if and only if the distance between any pair of points $x_i$ and $x_j$ satisfies $d(x_i, x_j) \leq \varepsilon$. The collection of all simplices generated by the point cloud $P$ that satisfy these conditions constitutes the* ***VR complex****.*

Given a point cloud $P$ and a series of parameters $0 = \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_m$, the nested VR complex sequence

$$VR(P, \varepsilon_0) \subset VR(P, \varepsilon_1) \subset \cdots \subset VR(P, \varepsilon_m)$$

is referred to as a *VR filtration* [11], and $\varepsilon$ is called the filtration parameter. To simplify notation, we will also denote $VR(P, \varepsilon)$ as $VR(\varepsilon)$. Fig. 2 gives an example of VR filtration.

**Definition 2.** *(Persistent Homology [11]) Suppose we have a filtration*

$$K_0 \subset K_1 \subset \cdots \subset K_m$$

*then for every $0 \leq i \leq j \leq m$ we have an inclusion map from $K_i$ to $K_j$ and therefore an induced homomorphism $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$ for each dimension $p$. The $p$-th* ***persistent homology groups*** *are defined as the images of the homomorphisms induced by inclusion:*

$$H_p^{i,j} = \mathrm{Im}\, f_p^{i,j},\ \forall 0 \leq i \leq j \leq m.$$

*The corresponding $p$-th* ***persistent Betti numbers*** *are the ranks of these groups: $\beta_p^{i,j} = \mathrm{rank}\, H_p^{i,j}$.*

The meaning of persistent Betti numbers is the number of "holes" exsisting during the considered period in the filtration. Thus, persistent homology is effective to detect topological structures of underlying point clouds. The standard method for computing the persistent homology of a filtration is the reduced matrix method [11]. Numerous software packages and libraries are now available for computing persistent homology, such as the GUDHI Python module [30].

One of the most common tools for visualizing persistent homology is the *persistence diagram* (PD) [11], depicted in Fig. 3. The set of points that records the birth time and death time of the $n$-cycles is denoted as the $n$-PD. Each point in an $n$-PD takes the form $(b_i, d_i)$, representing an $n$-cycle and capturing its *birth time* $b_i$ and *death time* $d_i$, and the value $|d_i - b_i|$ is termed as the *persistence* of this cycle. Typically, points within an $n$-PD can be classified into two classes based on their persistence. Points that exhibit larger persistence are called *significant points*, while others are called *noise points*. Various methods exist for identifying significant points from PDs, such as the delineation of confidence sets [15] and the clustering of these points [21]. Then, if the number of closed curves corresponding to the sampling point cloud is not known in advance, it can be regarded as the number of significant points in 1-PD.

For each $H_n(K)$, suppose its basis is $\{[r_1], \cdots, [r_m]\}$, then each $r_i, i = 1, \cdots, m$ is called a *representative $n$-cycle* of the homology class $[r_i]$. The representative $n$-cycle of the persistent pair $(b, d)$ in persistent homology, referred to as a *persistent $n$-cycle* [9], is a specific representative $n$-cycle in the filtration $K^b \subset K^{b+1} \subset \cdots \subset K^d$. For instance, a persistent 1-cycle corresponding to $(b, d)$ emerges at $K^b$ and becomes a boundary at $K^d$.

By definition, persistent $n$-cycle may not be unique for a given persistence pair. However, these persistent $n$-cycles have a common positive $n$-simplex that gives birth to them, hence each point in 1-PD will correspond to a positive 1-simplex.

**Definition 3.** *(Positive Simplex [11]) Consider a filtration process that adds one simplex each time. We refer to a sim-*

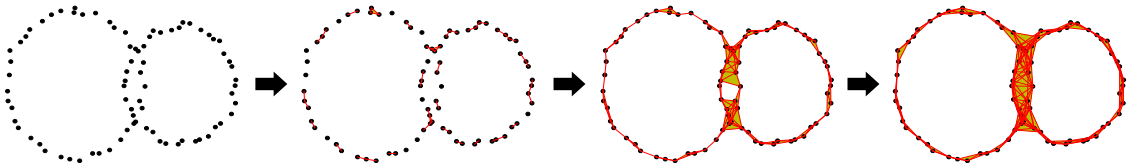


Figure 2. An example of VR filtration. As the filtration parameter increases, more and more simplices appear.

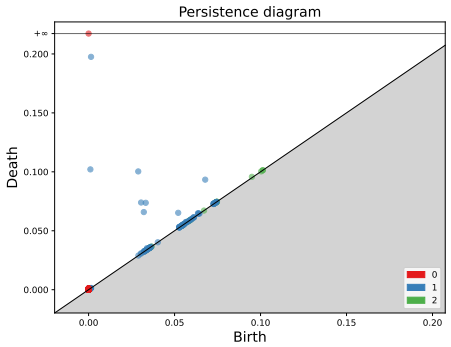



Figure 3. An example of persistence diagram.

*plex $\sigma$ as **positive** if its addition creates a new persistent cycle, thus giving birth to a new homology class. Otherwise, we regard it as **negative**.*

A positive 1-simplex corresponding to a significant point in 1-PD is called a *significant positive 1-simplex*.

### 3.2. Local principal curve

Now we present certain concepts and details about local principal curves. Let $P = \{p_1, \cdots, p_n\}$ represent a $d$-dimensional point set in $\mathbb{R}^d$, where $p_i = (p_{i1}, \cdots, p_{id})$. Let $H$ denotes a bandwidth matrix and $K_H(\cdot)$ signifies a $d$-dimensional kernel function (usually the Gaussion kernel function). Typically, $H$ is set as $H = \{h^2 \cdot I : h > 0\}$, with $I$ representing the $d$-dimensional identity matrix, and $h$ referred to as the bandwidth parameter [13]. For $x \in \mathbb{R}^d$, the *local center of mass* around $x$ is given by [13] as:

$$\mu^x = \frac{\sum_{i=1}^n K_H(p_i - x)p_i}{\sum_{i=1}^n K_H(p_i - x)}$$

Then, let $\Sigma^x = (\sigma^x_{jk})$ represent the local covariance matrix of $x$, where the $(j,k)$-th entry $(1 \le j,k \le d)$ is given by

$$\sigma^x_{jk} = \sum_{i=1}^n w_i(p_{ij} - \mu^x_j)(p_{ik} - \mu^x_k)$$

with weights $w_i = K_H(p_i - x)/\sum_{i=1}^n K_H(p_i - x)$.

**Definition 4.** *Let $\vec{\gamma}^x$ be the first eigenvector of $\Sigma^x$. Then, $\vec{\gamma}^x$ constitutes the first column of the loadings matrix $\Gamma^x$ from the eigen decomposition $(\Gamma^x)^T\Sigma^x\Gamma^x = \Lambda^x$, where $\Lambda^x = \mathrm{diag}(\lambda^x_1, \ldots, \lambda^x_d)$ is a diagonal matrix containing the ordered eigenvalues of $\Sigma^x$, with $\lambda^x_1 \ge \cdots \ge \lambda^x_d$. We refer to $\vec{\gamma}^x$ as the **first local principal component** at $x$, and the process of finding $\vec{\gamma}^x$ as **local principal component analysis** [13].*

The key ingredients in computing a local principal curve are the selection of the starting points and the bandwidth parameter. In our work, we will tackle these issues by utilizing persistent homology, which provides an automated approach for parameter selection.

## 4. Perception Faithful Curve Reconstruction

In this section, we present the human perception faithful curve reconstruction algorithm, focusing on the separation and reconstruction of intersecting closed curves. For the sampling point clouds of the original closed curves, it is assumed that each curve is densely sampled with a similar density, and that noise exists in the sampling points.

### 4.1. Topological understanding of the point cloud

By constructing the VR filtration from a given point cloud and computing its persistent homology, we can obtain the 1-PD, from which a wealth of important information about the underlying point cloud can be derived. For instance, through this process, we can identify all the positive 1-simplices of the persistent 1-cycles, along with their respective positions and lengths.

In cases of intersecting closed curves, particularly planar intersecting closed curves, new loops will emerge due to the presence of an even number of intersection points. As a result, the region enclosed by original closed curves is segmented into several parts, with each part having a persistent 1-cycle representing its boundary. For instance, Fig. 1 demonstrates an example where the region enclosed by two ellipses is segmented into five parts, and their boundary cycles form five points with larger persistence in the 1-PD. Generally, if the region enclosed by the original closed curves is separated into $k$ parts, the 1-PD will contain $k$ points with persistence larger than other noise points, although there may be differences between the persistence of these $k$ points too. In the subsequent discussion, we denote the $k$ points in the 1-PD corresponding to these $k$ persistent 1-cycles generated by all intersecting closed curves as *significant points*, and the corresponding positive 1-simplices

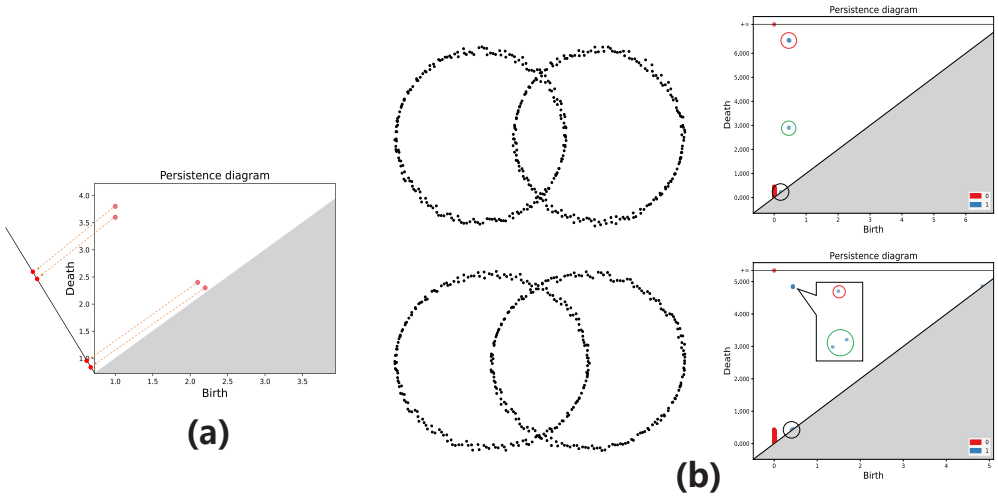



Figure 4. (a) An illustration of projecting points in PD onto the line $y = -x$. (b) Illustrations of clustering to identify significant points. On the left are the point clouds used as examples, on the right are the results of clustering into three classes (each colored circle denotes a class).

as *significant positive 1-simplices*. In general, $k$ is larger than the number of original sampling closed curves. But as we will see later, by extracting all significant positive 1-simplices and then employing the next steps of our curve reconstruction algorithm on each of them, it is feasible to automatically separate and reconstruct all original sampling closed curves in the correct quantity.

Initially, we assert that for each curve, there will be at least one significant positive 1-simplex located on the sampling points of this curve.

**Proposition 1.** *Given a point cloud $P$ sampled from some intersecting closed curves, then by constructing VR filtration from $P$ and computing persistent homology, for each closed curve, at least one significant positive 1-simplex will be located on the sampling points of this curve.*

The proof is available in Supplementary Information.

Hence, the initial assertion holds true, indicating that each sampling curve possesses at least one significant positive 1-simplex within its sampling point cloud. When the number of significant positive 1-simplices exceeds the number of original sampling curves, more than one positive 1-simplices will be located on a single sampling curve, resulting in the generation of reconstructed curves that represent the same original sampling curve. In such cases, a filtering condition can be introduced to remove redundant curves, which will introduced in section 4.3. Therefore, all the original sampling curves can be reconstructed by simply performing our curve reconstruction algorithm on all significant positive 1-simplices. In practice, we usually consider these positive simplices in descending order of the persistence of the corresponding persistent 1-cycles.

The remaining problem pertains to identify all the significant points in the 1-PD. In this paper, we will utilize the method of clustering. Initially, according to the assumption that each curve is densely sampled with a similar density, these significant positive 1-simplices should be close in length, hence the birth time of all significant points in 1-PD should be close too. Thus, we can remove points with birth times greater than a certain threshold, as these points cannot be significant in this context. This threshold can be automatically selected by relying on the birth time $b_1$ of the point $(b_1, d_1)$ in 1-PD with the largest persistence, for instance, by setting the threshold as $2 \cdot b_1$.

Subsequently, all the remaining points in 1-PD are projected onto the line $y = -x$, hence projected points farther from the origin have greater persistence (see Fig. 4 (a) for an illustration). By denoting the number of points in the 1-PD as $m$, we add an additional $m$ points $(0, 0)$ to the projected point set, thereby expanding it to $2m$ points. Then we can identify significant points in 1-PD by clustering the projection points. In cases of intersecting closed curves, new loops may form with relatively small persistence compared to points with the largest persistence. However, these loops possess relatively large persistence compared to noise points, and as such, should not be classified as noise points. For instance, in the upper row of Fig. 4 (b), the middle loop formed by intersections is smaller than the other two, corresponding to a point with moderate persistence in 1-PD. Since the above scenario is common in cases of intersecting curves, to prevent genuinely significant points, such as those corresponding to small loops, from being misclassified as noise, and to robustly handle noise points in PD, we choose to cluster the $2m$ points into three classes by employing the $k$-means method [17]. Then we consider the points in the 1-PD before projection corresponding to the two classes far from the origin (indicating larger persis-

tence) as significant points, and the remaining class close to the origin as noise points. This method effectively addresses cases like the upper row in Fig. 4 (b). Furthermore, for cases where the significant points are all relatively close together, this method can also identify the correct significant points. This is due to the addition of a large number of points to the origin, ensuring that truly noise points are consistently clustered into one class, thereby rendering all the points in the remaining classes as significant points. The lower row in Fig. 4 (b) provides an example where three points have proximate persistence, and clustering the points in 1-PD into three classes yields correct significant points. Thus, by clustering we can effectively and generally determine the number of genuinely significant points in most cases.

Moreover, all significant positive 1-simplices can also be further extracted from the VR filtration. Therefore, by computing persistent homology and extracting significant points in 1-PD, it is possible to determine the position and length of all significant positive 1-simplices within the filtration. Subsequently, applying this information to the next reconstruction step to compute local principal curves, while also eliminating curves that represent duplicates of the same sampling curve, enables the successful reconstruction of all the original sampling closed curves. The above process can be summarized as Algorithm 1.

**Algorithm 1** Deriving Persistent Homology Information

**Input:** A point cloud $P$.

1: Construct the VR filtration from $P$.
2: Calculate persistent homology and the 1-PD. Find the birth time $b_1$ of the point with largest persistence, and remove points with birth time larger than $2b_1$.
3: Cluster the remaining points in 1-PD, and derive the significant points.
4: Identify the position and length of all significant positive 1-simplices.
5: Sort these positive 1-simplices in descending order based on the persistence of corresponding persistent 1-cycles.

**Return:** The sorted significant positive 1-simplices along with their position and length.

### 4.2. Selecting starting point and bandwidth to compute local principal curves

In practice, the sampling point clouds often contain noise. To address this issue, we utilize the local principal curve, a form of piecewise linear curve, to initially approximate closed curves from the noisy sampling point clouds. But as mentioned earlier, the algorithm requires manual selection of two parameters: the starting point $x_0$ of the local principal curve and the bandwidth $h$. In other words, to detect a single closed curve from a point cloud, we must identify a point $x_0$ sampled from this curve and select a suitable parameter $h$, which introduces a practical challenge. In the original approach for computing local principal curves [13], the starting points are often randomly selected, and the bandwidth needs to be determined manually, leading to increased difficulties in practical application. However, since we have gained a topological understanding of point clouds, based on the outcomes of Algorithm 1, we have determined the position and length of each positive 1-simplex. Given that every original closed curve can be represented by a persistent 1-cycle, and we have identified the positive 1-simplices of these cycles, we designate one vertex of the corresponding positive 1-simplex as the starting point for each local principal curve and set the bandwidth equal to the length of this positive 1-simplex. Consequently, Algorithm 1 enables the automatic selection of parameters for computing local principal curves.

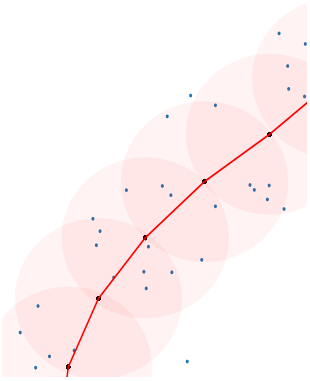


Figure 5. A localized illustration of a local principal curve. The blue points represent the point cloud, while the centers of the circles (the black points) depict the vertices of the local principal curve, and the radius represents the bandwidth.

Therefore, to reconstruct the closed curves from given point cloud, for each closed curve, we will first compute a local principal curve to approximate it. Fig. 5 gives a localized illustration of a local principal curve, and its original computing method can be described as follows [13]:

- Select a suitable bandwidth $h$ and a starting point $x(0)$, set $x = x(0)$ initially.
- Calculate the local center of mass $\mu^x$ around $x$ and perform a local principal component analysis.
- Find the new value $x$ by following the first local principal component $\vec{\gamma}^x$ starting at $\mu^x$, and update $x$ as $\mu^x + h \cdot \vec{\gamma}^x$.
- Repeat above two steps until the stopping condition is satisfied.

It is worth noting that this method can handle local crossings between curves, making it suitable for addressing our

problem. Specifically, when calculating the unit direction vector $\vec{\gamma}^x$ under the assumption that the intersections exist, its computation can be modified by adding an angle penalization to ensure that the local principal curve continues in the same direction at the intersection as the last obtained $\vec{\gamma}^x$ [13]. Specifically, let $\vec{\gamma}^x_{(i-1)}$ and $\vec{\gamma}^x_{(i)}$ denote the unit direction vectors at the $(i-1)$-th and $i$-th steps, respectively. We define

$$a^x_{(i)} := \left|\vec{\gamma}^x_{(i-1)} \cdot \vec{\gamma}^x_{(i)}\right|^k$$

It is typical to set $k = 2$ [13]. Then, using angle penalization, the modified $\vec{\gamma}^x_{(i)}$ can be computed as

$$\vec{\gamma}^x_{(i)} := a^x_{(i)} \cdot \vec{\gamma}^x_{(i)} + \left(1 - a^x_{(i)}\right) \cdot \vec{\gamma}^x_{(i-1)} \tag{1}$$

This method of computing $\vec{\gamma}^x$ takes into account previous directions of the local principal curves, thereby aligning with the good continuation principle. Then the algorithm of computing local principal curve can be described as Algorithm 2.

**Algorithm 2** Computing Local Principal Curve

**Input:** A suitable bandwidth $h$ and a starting point $x(0)$.

1: Set $x = x(0)$.
2: Calculate the local center of mass $\mu^x$ around $x$.
3: Perform a principal component analysis locally at $x$.
4: At the $i$-th step, update $\vec{\gamma}^x_{(i)}$ at $x$ using Equation 1, then update $x$ as $\mu^x + h \cdot \vec{\gamma}^x_{(i)}$.
5: Repeat steps 2 through 4 until the stopping condition is satisfied.

**Return:** The local principal curve represented by a sequence of points.

Since we only consider closed curves here, we set the stopping condition as $d(x_0, x) < h$ when the iteration has been repeated more than once. Note that the Algorithm 2 obtains a local principal curve represented by a sequence of points by advancing a fixed step size at a time, in practice, it is convenient to set the step size equal to the bandwidth parameter $h$ [13, 14].

Now, we provide a discussion regarding the rationale behind our parameter selection, based on certain properties of local principal curves. Einbeck and Zayed have presented some asymptotic properties for localized principal components and local principal curves [14], and we will offer additional theoretical discussions related to our work, particularly focusing on the connections between the local principal curves obtained by the proposed algorithm and the persistent 1-cycles.

We first illustrate a property of the local center of mass $\mu^x$ around a point $x$. That is, the Euclidean distance $d(\mu^x, x)$ is less than the bandwidth parameter $h$ if the kernel function is selected as Gaussion kernel function [13].

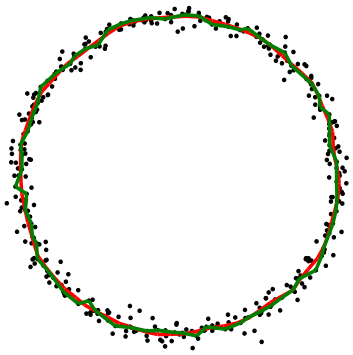


Figure 6. Comparison of local principal curve (red) and representative cycle (green). The representative cycle shows more obvious turns.

Since the local center of mass focuses on considering points around $x$, by properties of the Gaussian kernel function we can approximately analyze $\mu^x$ by considering only the points whose distance to $x$ is less than $h$.

**Proposition 2.** *If the kernel function $K_H(x)$ is selected as $n$-dimensional Gaussion kernel function, the following inequality holds: $d(\mu^x, x) < h, x \in \mathbb{R}^n$.*

The proof is available in Supplementary Information.

The following theory asserts that the local principal curves obtained from the proposed algorithm can serve as good approximations of representative cycles of the original curves, which can have a simpler shape, especially in the case of large noise. See Fig. 6 for an example.

**Theorem 1.** *Suppose $P$ is a point cloud sampling from a closed curve, $v_0$ represents one end of the positive 1-simplex $v_0v_m$ of the persistent 1-cycles of this curve, and $h$ represents the length of $v_0v_m$. Assuming the local principal curve $L$ with a starting point $x_0 = v_0$ and bandwidth $h$ is denoted by the sequence of vertices $L := x_0x_1 \cdots x_nx_0$, then there exists a persistent 1-cycle $C := v_0v_1 \cdots v_mv_0$ such that for every $x_i \in L$, there exists at least one $v_j \in C$ satisfying $d(x_i, v_j) < h$. Therefore, $L$ can serve as an approximation of $C$.*

The proof is available in Supplementary Information.

*Remark* 1. This theory still holds if $P$ is sampled from more than one curves (including situations of intersecting curves), since for each curve we can compute a local principal curve and derive the corresponding persistent 1-cycle.

If we introduce a parameter $\alpha$ to restrict the extent to which the sampling points deviate from the original curve, then, based on the bandwidth parameter $h$ determined by the positive 1-simplex as well as $\alpha$, we can provide an upper-bound estimate of the distance from the obtained local principal curve to the original curve.

**Theorem 2.** *Let $S$ represent the original sampling curve and $P$ denote the sampling point cloud. Considering the projection $\tilde{P} = \{\tilde{p}\}_{p\in P}$ of $P$ onto the true curve $S$, we assume the existence of $\alpha > 0$ such that*

$$\|\tilde{p} - p\| \leq \alpha, \forall p \in P.$$

*Let $L$ be the resulting local principal curve with bandwidth $h$. Then*

$$\sup_{x\in L} \inf_{v\in S} d(x, v) < h + \alpha.$$

*Thus, the distance from the local principal curve to the original curve is controlled by $h$ and $\alpha$.*

The proof is available in Supplementary Information.

Hence, if the sampling point cloud has a small deviation from the curve and the sampling is dense enough, then the generated local principal curve will have a smaller deviation from the true curve.

### 4.3. Removing duplicate curves

As we previously mentioned that different positive 1-simplices may produce duplicate local principal curves, thus requiring a filtering condition to exclude these duplicates. Here we filter these duplicate curves by considering the overlap points in the certain neighborhoods of the curves, and call the filtering condition as the *overlap condition*.

**Definition 5.** *(Overlap Condition) Let $L_1$ and $L_2$ be two local principal curves, with $h_1$ and $h_2$ as their bandwidth parameters respectively. Denote $P_1$ as the points in the sampling point cloud $P$ satisfying a distance to $L_1$ less than $h_1$, and $P_2$ as the points in the sampling point cloud $P$ satisfying a distance to $L_2$ less than $h_2$. Denote the number of elements in a set by "#", if*

$$\#(P_1 \cap P_2)/\#P_1 > 0.5 \text{ or } \#(P_1 \cap P_2)/\#P_2 > 0.5$$

*we say that $L_1$ satisfies the overlap condition with $L_2$.*

We assume there is no overlap between different sampling curves, thus checking if the number of overlapping points exceeds half is sufficient to determine whether they represent the same sampling curve.

In practice, we can establish a set wherein each new principal curve can be compared with the existing principal curves in this set. This process allows us to determine if the overlap condition is satisfied. If a curve satisfies the overlap condition with another existing one, it is discarded, and a new round of computation begins with the next significant positive 1-simplex. Otherwise, the curve is added to the set. Ultimately, the set contains all the principal curves representing the original sampled curves without duplication.

### 4.4. Fitting with smooth curves

To derive a better quality reconstruction curve, since the output of Algorithm 2 is a sequence of points representing vertices of the obtained local principal curve, we can directly fit these vertices with a smooth curve, usually a B-spline curve. This can remain the shape of local principal curve that approximates the sampling closed curve, meanwhile improved smoothness of resulting curve. Furthermore, various techniques can be employed to enhance the fitness of the B-spline curve and thus improve the quality of the reconstructed curve. One such technique is the least-squares progressive-iterative approximation (LSPIA) method [8] for optimizing the derived B-spline curve. Suppose a local principal curve $L$ with bandwidth $h$ and a fitted B-spline curve $S$ have been derived, given that there could be additional sampling curves apart from the one represented by $L$, we exclusively focus on the subset $P'$ of the original sampling point cloud $P$ that is contained within a local neighborhood of $L$:

$$P' = \{p \in P | \exists x \in L \text{ with } d(x, p) \leq h\}$$

This implies that $P'$ only comprises points near the original sampling curve corresponding to $L$, excluding points sampled from other curves. Subsequently, the LSPIA method can be utilized to optimize $S$ based on $P'$, thereby deriving a smooth curve, which consequently leads to an improved reconstruction of the point cloud. By employing this method for all initially fitted B-spline curves, the quality of all reconstructed curves can be enhanced.

The whole separation and reconstruction process can be described as the following Algorithm 3.

**Algorithm 3** Human Perception Faithful Closed Curve Reconstruction Algorithm

**Input:** A point cloud $P$ sampled from some closed curves.

1: Initialize $S$ as an empty set.
2: Compute persistent homology of $P$ to derive the sorted significant positive 1-simplices using Algorithm 1.
3: **For** each significant positive 1-simplex $e$ **do**
4: Denote the two vertices of $e$ as $x_0$ and $x_1$, and denote the length of $e$ as $h$.
5: Use Algorithm 2 to extract the local principal curve $L$ by setting the original point as $x_0$ and bandwidth as $h$.
6: **If** $L$ does not satisfy the overlap condition with all the existing curves in $S$ **then**
7: Fit $L$ with a smooth curve and add it to $S$.
8: **End if**
9: **End for**

**Return:** The set $S$ containing all curves representing original sampling curves.

## 5. Experiments and Discussions

In this section, comparisons with previous methods are first provided. Then, experimental results of the proposed method, along with some of its applications, are presented. Finally, we will give some discussions.

### 5.1. Influence of starting points and bandwidth

As previously mentioned, the selection of starting point and bandwidth significantly influences the quality of a local principal curve. Thus we initially compare the proposed method to the original algorithm [13], which computes local principal curves using randomly selected starting points and manually set bandwidth parameters. It is clear that randomly selecting the starting point is unsuitable for dealing with the challenge of reconstructing intersecting closed curves, due to the possibility of the random point falling on the sampling point cloud of the same curve multiple times. In scenarios involving a large number of curves, numerous random repetitions of the selection may be necessary to ensure that at least one point was selected in the sampling point cloud for each curve. Furthermore, concerning the selection of the bandwidth parameter, a parameter that is excessively large will yield a local principal curve that inadequately or wrongly approximates the point cloud, while one that is too small will lead to an inaccurate local principal component analysis, resulting in a "locally sparse" point cloud that prematurely stops the computation of the principal curve. Therefore, manually selecting the bandwidth parameter may involve tedious attempts to gradually increase it from a very small value.

Fig. 7 illustrates the results of comparing randomly selected starting points and unsuitable bandwidth parameters with our method, where we randomly select starting point for each sampling curve, and select bandwidth parameters less than half the length of the positive simplex (Fig. 7 (a)) and greater than twice the length of the positive simplex (Fig. 7 (b)), respectively. We can see that a bandwidth that is too small can result in an erroneous local principal component analysis and impede the curve searching process. Conversely, an excessively large bandwidth can lead to an inaccurate approximation of the point cloud and mishandling of intersections. In comparison, our method (Fig. 7 (c)) succeeds by automatically selecting appropriate starting points and bandwidth values, which is a significant advantage.

### 5.2. Comparison with other methods

We also compare the proposed method with several state-of-the-art curve reconstruction approaches. Given that the intersection positions between different curves are locally non-manifold, we choose to compare our method with three state-of-the-art methods: method in [28], PEEL method [35] and optimal transport method [7], all of them are able to handle non-manifold cases. The comparison results are illustrated in Fig. 8. In Fig. 8 (a), unwanted edges appear since noise causes the algorithm to incorrectly process points near intersections. In Fig. 8 (b), the algorithm generates unwanted edges around the intersection and fails to reconstruct some edges. In Fig. 8 (c), the algorithm also creates unwanted edges around the intersection, and a gap is mistakenly formed. Importantly, none of them is capable of directly separating a single closed curve from the original point cloud, let alone fitting them separately with a smooth curve. They can only extract a general skeleton (such as a graph) of the entire point cloud. In contrast, our method successfully separates and reconstructs each closed curve by utilizing an ordered sequence of points to represent them (Fig. 8 (d)), which constitutes a significant advantage of the proposed approach.

### 5.3. Experimental results

We start by demonstrating the robustness of our method to noise. Fig. 9 display the results of the proposed method as noise is gradually increased within the same point cloud. Evidently, the proposed method can successfully extract each curve even in the presence of large noise, aligning with human visual perception. Therefore, our method exhibits robustness to the effects of noise.

Then we present several experimental results, reconstructed from point clouds by fitting with B-spline curves and optimizing by LSPIA method. In Fig. 10 (a–c), results of some combined geometric shapes are depicted, and Fig. 10 (d–f) show examples of reconstructing closed curves in cartoon designs. It is evident that our method can reconstruct both intersecting and non-intersecting closed curves. Moreover, the proposed method can also extract self-intersecting curves. In Fig. 10 (g), an example is shown where one curve with two self-intersecting points forms three loops due to self-intersecting. Our method correctly reconstructs this curve while handling the self-intersecting points, resulting in a representation that conforms to human visual perception.

Finally, we show some results of our approach in real applications. First, we present some reconstruction examples of patterns. These patterns are commonly found in the domain of graphic design, which are composed of closed base shapes. By sampling the pixels representing these shapes from images, we can obtain the corresponding point clouds. Subsequently, the proposed method is utilized to compute reconstructed curves representing these base shapes, with the outcomes depicted in Fig. 10 (h). Furthermore, we present several results of reconstructing trajectories [1]. In Fig. 11 (a), two closed trajectories for hiking around the West Lake (a famous natural attraction in Hangzhou, China)

[1] https://www.2bulu.com/track

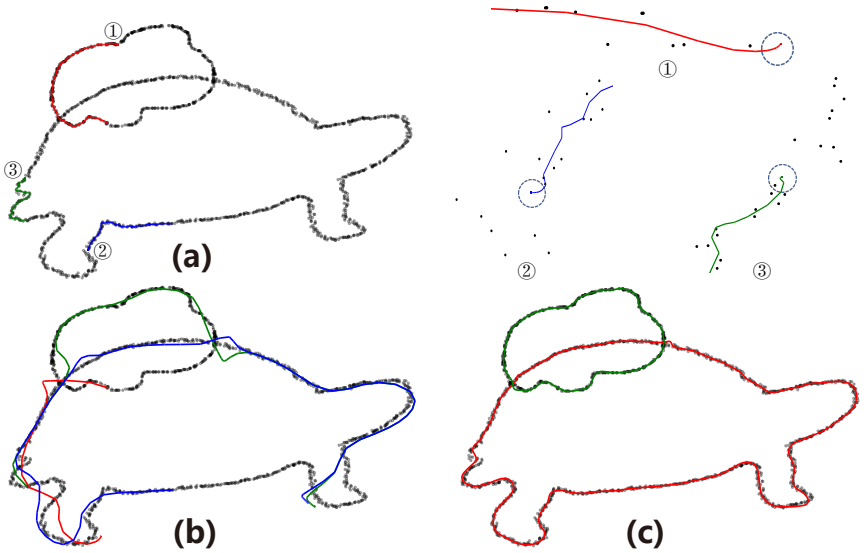



Figure 7. Comparison with method using randomly selected starting points and manually set bandwidth parameters. (a) Bandwidth is less than half the length of the positive simplex. The computation of principal curves prematurely stops (see the blue circles), since too small bandwidth will lead to inaccurate local principal component analysis. (b) Bandwidth is larger than twice the length of the positive simplex. The results fail to approximate the point cloud and handle intersecting points correctly. (c) Our experimental results.

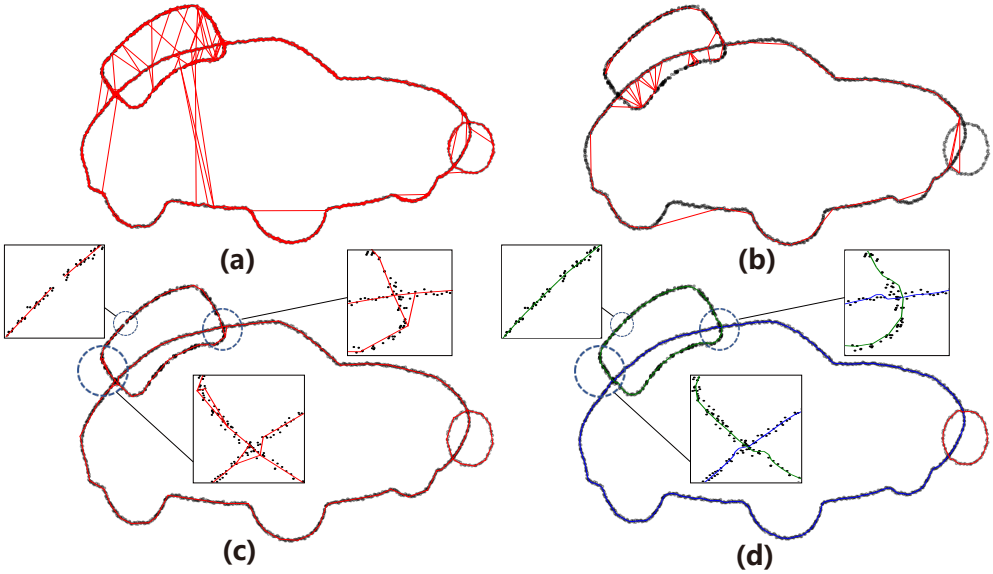



Figure 8. Comparison with other state-of-the-art approaches. (a) Results of method in [28]. Unwanted edges appear due to noise points near intersections. (b) Results of PEEL method [35]. The algorithm generates unwanted edges around the intersection and fails to reconstruct some edges. (c) Results of optimal transport method [7]. Unwanted edges around the intersection appear, and a gap is mistakenly formed, as shown in the blue circles. (d) Our experimental results, where different closed curves are drawn in different colors.

are depicted. The blue dots on the figure represent public GPS location data with minor noise, collected by the hikers at intervals during the hike. Consequently, reconstructing the curves from this point set will allow for an approximation of the hiking routes, and may help to discover some paths that are not marked on the map. Using the proposed algorithm, we conducted the reconstruction and obtained two heart-shape like closed curves. Similarly, in Fig. 11 (b) there are three intersecting closed trajectories, and three closed curves are successfully separated and reconstructed, again demonstrating the effectiveness of our method.

In the following we will discuss the quality of the generated principal curves. One index to assess the suitability of the bandwidth parameter $h$ is the *self-coverage* [13]. It means the fraction of all data points which are found in a

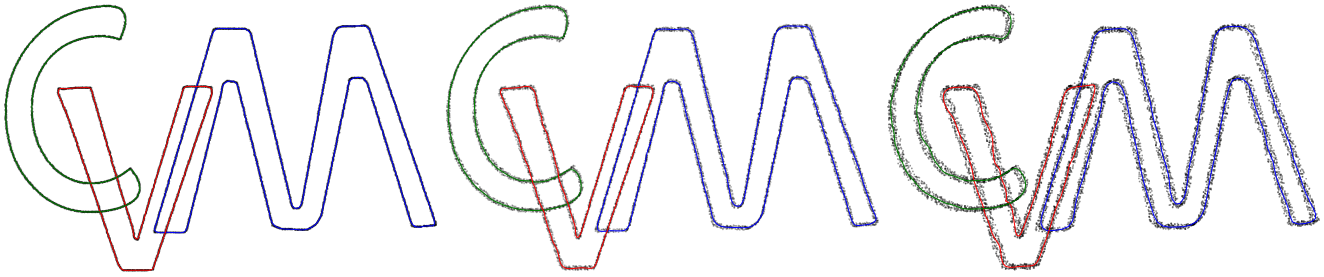


Figure 9. Illustration of the robustness of our method to noise. As the noise is gradually increased from left to right, our method successfully reconstructs all the curves.

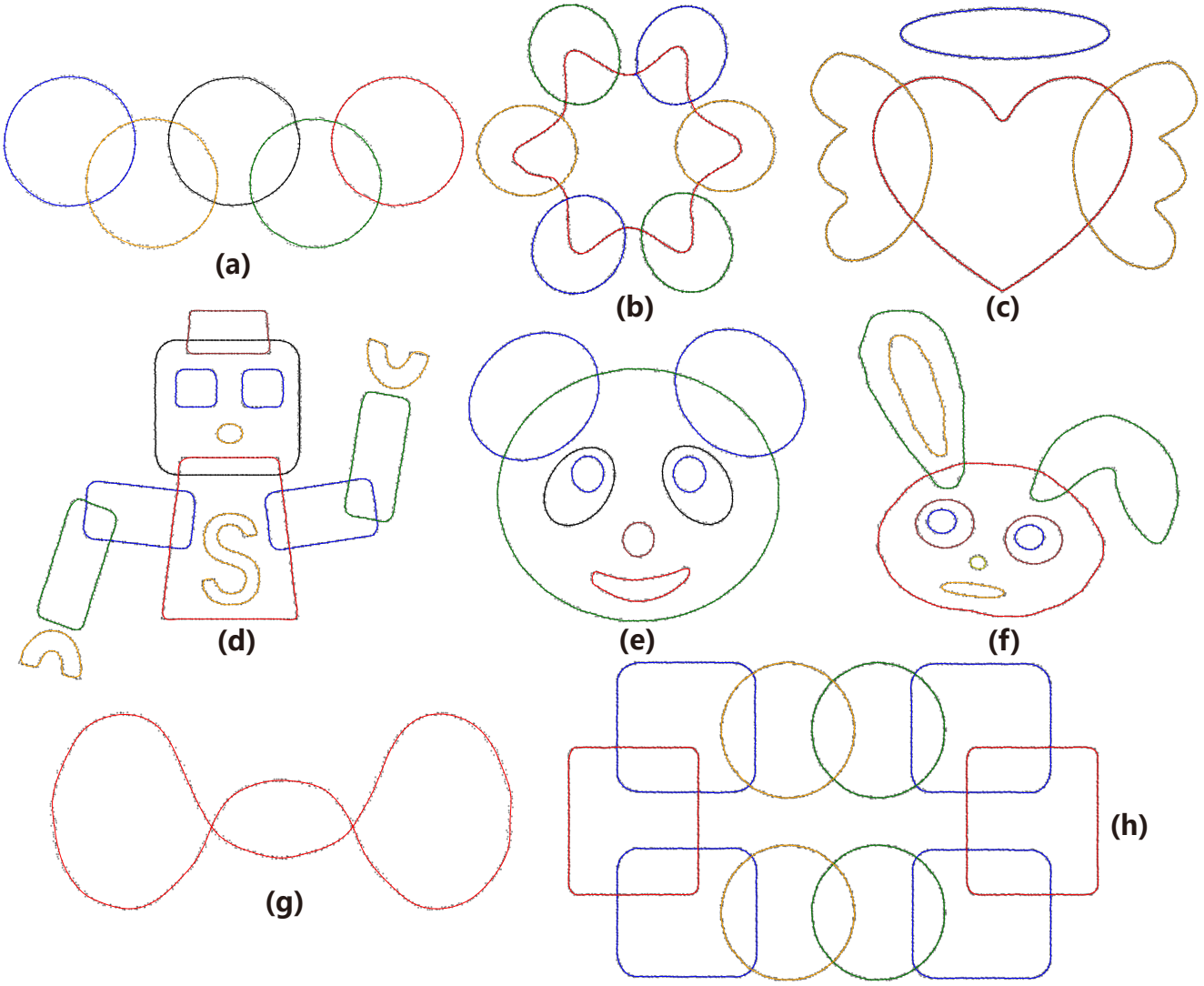
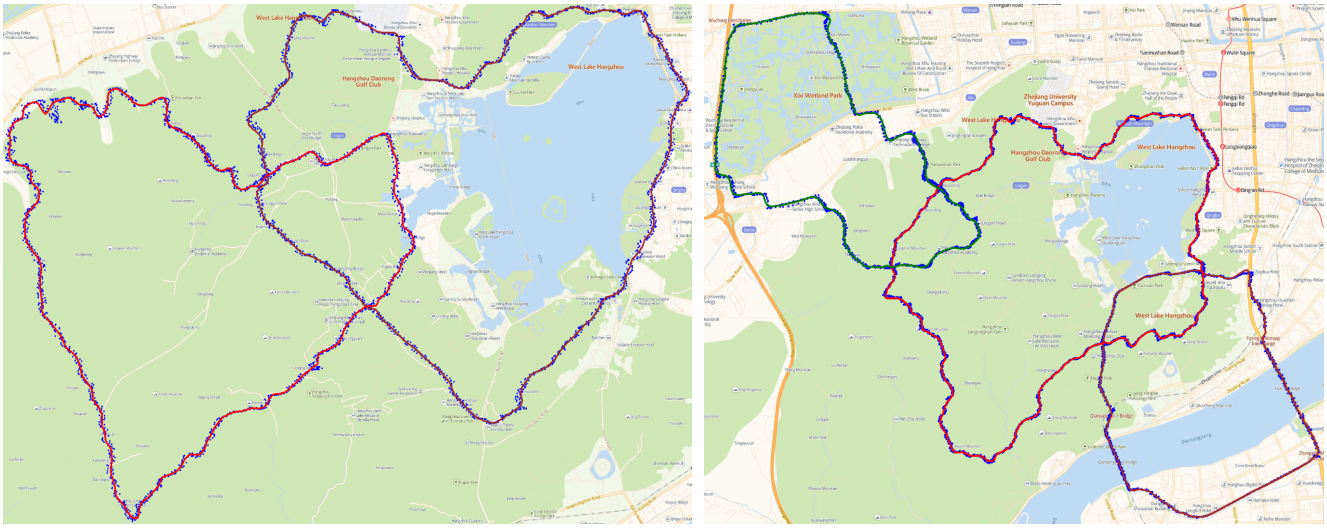



Figure 10. Some of our experimental results. Different curves are plotted in different colors.

certain neighborhood of the principal curve, defined as

$$C_L(h) = \#\{p \in P | \exists x \in L \text{ with } \|x - p\| \leq h\}/n$$

where $P$ is the given point cloud, $n$ is the number of points in $P$, and $L$ is the local principal curve. It is a monotonically increasing function of $h$ and will finally reach the value 1. Therefore, on the condition that the shape of the point cloud can be approximated, the closer the self-coverage is to 1, the better the quality of the principal curve is considered to be. In our problem of approximating intersecting closed



(a) (b)

Figure 11. The blue dots are public GPS location data with minor noise, the colored curves are reconstruction results of hiking trajectories using our method. (a) Two intersecting trajectories. (b) Three intersecting trajectories.

Table 1. Self-coverage of the resulting principal curves

| Point Clouds in Fig. 10 | | | |
|---|---|---|---|
| Fig. 10 (a) | 0.999 | Fig. 10 (e) | 1.000 |
| Fig. 10 (b) | 1.000 | Fig. 10 (f) | 1.000 |
| Fig. 10 (c) | 0.996 | Fig. 10 (g) | 1.000 |
| Fig. 10 (d) | 1.000 | Fig. 10 (h) | 1.000 |
| Point Clouds in Fig. 11 | | | |
| Fig. 11 (a) | 0.972 | Fig. 11 (b) | 0.998 |

curves, we can compute the self-coverage of all obtained local principal curves. Table. 1 shows the self-coverage of some experimental results, which proves that the bandwidth parameter given by our method is appropriate.

### 5.4. Limitations

There are some limitations of the proposed method. First, the topological understanding through persistence diagrams remains a technique under active research and development. At the current stage of the methodology, when dealing with certain types of noisy point sets or intersecting conditions, the topological understanding may yield an incorrect count of significant points, indicating a limitation of the current topological understanding method. Nonetheless, we anticipate that further exploration of persistent homology will yield additional methods for achieving a more accurate topological understanding of point clouds based on persistence diagrams in the future. Additionally, owing to the underlying principle of the local principal algorithm, it is more suitable for fitting smooth curves [12–14]. Consequently, for curves with sharp corners, this algorithm may not perform optimally. In fact, as shown in Fig. 11, the resulting curves fail to retain some sharp corners depicted in the original point clouds, although they generally provide good approximations of entire point clouds. Moreover, the proposed method cannot automatically reconstruct open curves yet.

## 6. Conclusions

In this paper, we propose a human perception faithful curve reconstruction approach that adheres to the good continuation principle of human visual perception. Specifically, it can automatically separate and reconstruct intersecting closed curves from noisy point clouds. Initially, we compute the persistent homology of the point cloud to gain a topological understanding and to identify relevant topological information, particularly focusing on the significant positive 1-simplices. Subsequently, the local principal curves are calculated to approximate the original curves, with parameters selection according to the results of the persistent homology. Finally, we fit each local principal curve with a smooth curve to obtain the final reconstructed curves of the original sampling curves. This method combines the advantages of both persistent homology and principal curves, making it effective for handling noisy point clouds and intersection cases, thereby producing faithfully reconstructed results aligned with human perception. The effectiveness of the proposed method is also validated through experiments.

In our future work, we plan to explore additional methods of topological understanding and curve approximating for reconstructing closed curves to overcome current limitations. Furthermore, we will research human perception faithful methods for open curve reconstruction based on

principal curves, with the choice of its parameters potentially informed by other dimensions of persistent homology.

## Acknowledgments

This work is supported in part by National Natural Science Foundation of China under Grant No. 62272406, and the Leading Goose R & D Program of Zhejiang under Grant No. 2024C01103.

## References

[1] A. Banerjee, F. Galassi, E. Zacur, G. L. De Maria, R. P. Choudhury, and V. Grau. Point-cloud method for automated 3d coronary tree reconstruction from multiple non-simultaneous angiographic projections. *IEEE transactions on medical imaging*, 39(4):1278–1290, 2019. 1

[2] A. Bernstein, E. Burnaev, M. Sharaev, E. Kondrateva, and O. Kachan. Topological data analysis in computer vision. In *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, pages 673–679. SPIE, 2020. 3

[3] G. Biau and A. Fischer. Parameter selection for principal curves. *IEEE Transactions on Information Theory*, 58(3):1924–1939, 2011. 3

[4] G. Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 23:289–368, 2014. 3

[5] G. E. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, 2009. 2

[6] I. Cleju, P. Fränti, and X. Wu. Clustering based on principal curve. In *Image Analysis: 14th Scandinavian Conference, SCIA 2005, Joensuu, Finland, June 19-22, 2005. Proceedings 14*, pages 872–881. Springer, 2005. 3

[7] F. De Goes, D. Cohen-Steiner, P. Alliez, and M. Desbrun. An optimal transport approach to robust reconstruction and simplification of 2d shapes. In *Computer Graphics Forum*, volume 30, pages 1593–1602. Wiley Online Library, 2011. 2, 10, 11

[8] C. Deng and H. Lin. Progressive and iterative approximation for least squares b-spline curve and surface fitting. *Computer-Aided Design*, 47:32–44, 2014. 9

[9] T. K. Dey, T. Hou, and S. Mandal. Persistent 1-cycles: Definition, computation, and its application. In *Computational Topology in Image Context: 7th International Workshop, CTIC 2019, Málaga, Spain, January 24-25, 2019, Proceedings 7*, pages 123–136. Springer, 2019. 4

[10] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & computational geometry*, 28:511–533, 2002. 2, 3

[11] H. Edelsbrunner and J. L. Harer. *Computational topology: an introduction*. American Mathematical Society, 2022. 4

[12] J. Einbeck and J. Dwyer. Using principal curves to analyse traffic patterns on freeways. *Transportmetrica*, 7(3):229–246, 2011. 3, 13

[13] J. Einbeck, G. Tutz, and L. Evers. Local principal curves. *Statistics and Computing*, 15:301–313, 2005. 3, 5, 7, 8, 10, 11, 13

[14] J. Einbeck and M. A. Zayed. Some asymptotics for localized principal components and curves. *Communications in Statistics-Theory and Methods*, 43(8):1736–1749, 2014. 3, 8, 13

[15] B. T. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh. Confidence sets for persistence diagrams. 2014. 4

[16] E. B. Goldstein. *Sensation and perception*. Wadsworth/Thomson Learning, 1989. 1

[17] G. Hamerly and C. Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 600–607, 2002. 6

[18] T. Hastie. Principal curves and surfaces. 1984. 3

[19] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American statistical association*, 84(406):502–516, 1989. 3

[20] S. Hauberg. Principal curves on riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1915–1921, 2015. 3

[21] Y. He, J. Yan, and H. Lin. Robust reconstruction of closed parametric curves by topological understanding with persistent homology. *Computer-Aided Design*, 165:103611, 2023. 4

[22] S. Kang and H.-S. Oh. Probabilistic principal curves on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3

[23] B. Kégl and A. Krzyzak. Piecewise linear skeletonization using principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):59–74, 2002. 3

[24] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger. Learning and design of principal curves. *IEEE transactions on pattern analysis and machine intelligence*, 22(3):281–297, 2000. 3

[25] A. Keler, J. M. Krisp, and L. Ding. Detecting vehicle traffic patterns in urban environments using taxi trajectory intersection points. *Geo-spatial information science*, 20(4):333–344, 2017. 1

[26] K. Koffka. *Principles of Gestalt psychology*, volume 44. Routledge, 2013. 1

[27] W. Köhler. Gestalt psychology. *Psychologische Forschung*, 31(1):XVIII–XXX, 1967. 1

[28] T. Lenz. How to sample and reconstruct curves with unusual features. In *EWCG: Proc. of the 22nd European Workshop on Computational Geometry*, pages 29–32, 2006. 2, 10, 11

[29] D. Levin. The approximation power of moving least-squares. *Mathematics of computation*, 67(224):1517–1531, 1998. 2

[30] C. Maria, J.-D. Boissonnat, M. Glisse, and M. Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *Mathematical Software–ICMS 2014: 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings 4*, pages 167–174. Springer, 2014. 4

[31] S. Ohrhallinger, J. Peethambaran, A. D. Parakkat, T. K. Dey, and R. Muthuganapathy. 2d points curve reconstruction survey and benchmark. In *Computer Graphics Forum*, volume 40, pages 611–632. Wiley Online Library, 2021. 2

[32] S. Ohrhallinger and M. Wimmer. Stretchdenoise: Parametric curve reconstruction with guarantees by separating connectivity from residual uncertainty of samples. In *Proceedings of Pacific Graphics 2018*. H. Fu, A. Ghosh, and J. Kopf (Guest Editors), 2018. 2

[33] S. Ohrhallinger and M. Wimmer. Fitconnect: Connecting noisy 2d samples by fitted neighbourhoods. In *Computer Graphics Forum*, volume 38, pages 126–137. Wiley Online Library, 2019. 2

[34] U. Ozertem and D. Erdogmus. Locally defined principal curves and surfaces. *The Journal of Machine Learning Research*, 12:1249–1286, 2011. 3

[35] A. D. Parakkat, S. Methirumangalath, and R. Muthuganapathy. Peeling the longest: A simple generalized curve reconstruction algorithm. *Computers & Graphics*, 74:191–201, 2018. 2, 10, 11

[36] C. S. Pun, S. X. Lee, and K. Xia. Persistent-homology-based machine learning: a survey and a comparative study. *Artificial Intelligence Review*, 55(7):5169–5213, 2022. 3

[37] M. Sarfraz, M. Ishaq, and M. Z. Hussain. Shape designing of engineering images using rational spline interpolation. *Advances in Materials Science and Engineering*, 2015(1):260587, 2015. 1

[38] I. Singh, S. Mbako, M. Singh, I. Benskrane, and R. Merzouki. Curve-based approach for shape reconstruction and planning of a mobile-continuum manipulator in structured environment. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1914–1919. IEEE, 2021. 1

[39] Y. Skaf and R. Laubenbacher. Topological data analysis in biomedicine: A review. *Journal of Biomedical Informatics*, 130:104082, 2022. 3

[40] R. Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022. 1

[41] J. Wang, X. Rui, X. Song, X. Tan, C. Wang, and V. Raghavan. A novel approach for generating routable road maps from vehicle gps traces. *International Journal of Geographical Information Science*, 29(1):69–91, 2015. 1

[42] L. Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5(1):501–532, 2018. 3

[43] M. Wertheimer. Laws of organization in perceptual forms. *Psycologische Forschung*, 4, 1923. 1

[44] R. Wu, B. Wang, and A. Xu. Functional data clustering using principal curve methods. *Communications in Statistics-Theory and Methods*, 51(20):7264–7283, 2022. 3