

Reconstruction of helices from their orthogonal projection [☆]



Frederic Cordier ^{a,b,*}, Mahmoud Melkemi ^a, Hyewon Seo ^b

^a LMIA, EA 3993, Université de Haute Alsace, 12 rue des Frères Lumière, 68093 Mulhouse, France

^b ICube, UMR 7357 CNRS, Université de Strasbourg, Pôle API, BP 10413, Bd Sébastien Brant, 67412 Illkirch cedex, France

ARTICLE INFO

Article history:

Received 26 August 2014

Received in revised form 11 April 2016

Accepted 12 April 2016

Available online 28 April 2016

Keywords:

Helix

3D reconstruction

ABSTRACT

We describe a method for modeling helices from planar curves. Given a polygonal curve in the (x, y) plane, the method computes a helix such that its orthogonal projection onto the (x, y) plane fits the polygonal curve. The helix curve is first sampled and the transformation matrix that best aligns points of the sampled helix to those of the polygonal curve is calculated. This transformation matrix is then used to estimate the parameters of the helix whose projection fits the polygonal curve.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Helices are three-dimensional curves whose curvature and torsion are constant. These curves are very common in nature and among human-made objects. Examples of objects having the shape of helix are hair curls, coiled phone cords, spring coils, tendrils of climbing plants, etc.

In this paper, we focus on the modeling of circular helices from their orthogonal projection. Our algorithm takes as input a polygonal curve in the plane $z = 0$ and generates a helix segment such that its orthogonal projection on the plane $z = 0$ best approximates the input polygonal curve. Our contribution is a method which is much more robust with respect to noise in the input data than the previous ones.

In this work, we aim at reconstructing a single helix from its orthogonal projection. There are many cases for which such method would be useful. The first application is related to computer vision; our algorithm could be used to reconstruct circular helices from an image taken with a camera. There are many examples of objects that may have the shape of circular helix: a photo of the architectural structure of a building, the image of the DNA structure taken with an electron microscope, etc. Another application of our algorithm could be for the reconstruction of helices from scan digitized images of engineering drawings. Engineering drawings are usually composed of primitives such as segments of lines, circles, ellipses and orthogonally projected helices. The digitized image would be first converted into a vector image using existing approach (Hilaire and Tombre, 2006). The helices could then be reconstructed from the curves that compose the vector image.

2. Related work

In this section, we review existing methods that have been developed for the modeling of circular helices. We also describe some of the methods in the domain of sketch-based modeling that could be used for the modeling of 3D curves.

[☆] This paper has been recommended for acceptance by Javier Sánchez-Reyes.

* Corresponding author. Tel.: +33 651819020.

E-mail address: frederic.cordier@gmail.com (F. Cordier).

2.1. Sketch-based modeling of wireframe models

One of the first approaches in sketch-based modeling is to let the user to draw the shape in wireframe. The input sketch is composed of a set of straight lines segments connected to each other and located in a plane; the goal is to compute the third coordinates of the segment extremities to create a 3D shape such that its projection matches the input sketch. The reconstruction is often solved with an optimization whose unknown variables are the third coordinates of the segment extremities. Different objective functions have been proposed such as minimizing the entropy of the angle distribution (Shoji et al., 2001) or minimizing the standard deviation of the segment lengths (Brown and Wang, 1996). The main drawback of these methods is that they cannot be used to reconstruct curvilinear shapes; they assume that the shape to reconstruct is rectilinear.

Cordier et al. (2013) have proposed a method to reconstruct curvilinear 3D shapes from a set of planar curves. The main limitation is that the reconstructed shape is assumed to be mirror-symmetric; their method cannot be extended to process non-symmetric shapes.

2.2. Approximating curves with piecewise helix curves

Several researchers have worked on the problem of fitting helices to 3-dimensional polygonal curves or 3-dimensional point clouds (Christopher et al., 1996; Nievergelt, 1997; Enkhbayar et al., 2008; Piuze et al., 2011; Goriely et al., 2009; Ghosh, 2010; Derouet-Jourdan et al., 2013). These algorithms are not directly applicable to the reconstruction of helices from orthogonal projection.

2.3. Reconstruction of helices from their orthogonal projection

Several researchers have worked on the problem of finding a helix whose orthogonal projection matches a polygonal curve provided by a user. In the work by Wither et al. (2007), the axis of the reconstructed helix is assumed to be parallel to the projection plane. Our method is able to handle the reconstruction of helices with arbitrary orientation.

Marchal et al. (2009) proposed a method to reconstruct trochoids using the derivative of the input curve with respect to its arc-length. Trochoids are special case of orthogonal projection of helices. One limitation of their method is that it is restricted to the reconstruction of x -axis aligned trochoids.

The closest work to ours is the one proposed by Chérin et al. (2014). Similarly to our method, their approach is aimed at finding a helix whose orthogonal projection matches a planar polygonal curve. Compared to their method, ours is much more robust with respect to noises in the input polygonal curve. We provide a detailed comparison of the two methods in the result section to support this claim.

3. Overview

Our method takes as input a 2D polygonal curve C in the (x, y) plane and generates a helix segment such that its orthogonal projection onto the (x, y) plane fits the polygonal curve. Note that the problem of fitting an orthogonally-projected helix segment to a polygonal curve is an underdetermined problem and therefore may not have a unique solution.

Instead of computing the fitting of the projected helix with the polygonal curve C , we simplify the problem by sampling the helix and compute the alignment of the points of the sampled helix with those of the polygonal curve. The problem comes down to estimating the transformation matrix that best aligns the points of the sampled helix with those of the polygonal curve. Using this optimal transformation matrix, we then estimate the parameters of the helix whose projection best fits the polygonal curve. This helix fitting method is explained in Section 4.

One important assumption of using this alignment method is that there is a one-to-one correspondence between the points of the sampled helix and those of the polygonal curve C . This implies that the helix and the polygonal curve C must be sampled such that this correspondence is valid. In section 5, we present an adaptive sampling method that satisfies this property. In section 6, we describe how this adaptive sampling is used in combination with the helix fitting.

The main contribution of the paper is to propose a new method for the fitting of an orthogonally-projected helix segment to a polygonal curve. Compared to the state-of-the-art methods, this new method is much more robust with respect to noise in the input polygonal curve. In section 7, we provide a detailed comparison of our method with the previous one.

4. Computing the helix parameters and the fitting error

Let C be a polygonal curve in the (x, y) plane. C is composed of n points $\{p_1, p_2, \dots, p_i, \dots, p_n\}$; each point p_i has coordinates $(x_{p,i}, y_{p,i})$. Let $H(t)$ be the parametric equation of a helix:

$$H(t) = \begin{bmatrix} r \cos(t) \\ pt \\ r \sin(t) \end{bmatrix}$$

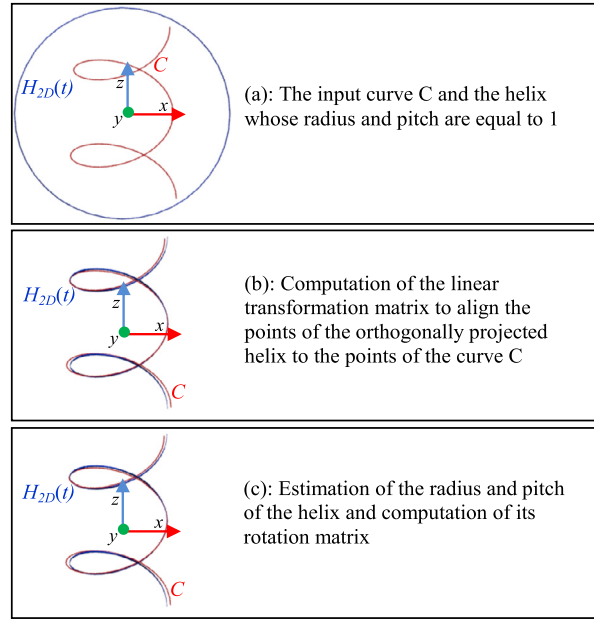


Fig. 1. Overview of the method.

The coefficients r and p are the radius and the pitch of the helix respectively. Similarly to polygonal curves which have two endpoints, we also define two endpoints of the helix segment with two different values of t . For the sake of simplicity, these values are $-\alpha$ and $+\alpha$; $H(-\alpha)$ and $H(\alpha)$ being the first and the second endpoints of the helix segment. Points that belong to the helix segment are those whose parameter t is in the interval $[-\alpha, +\alpha]$.

The problem is formulated as follows: given a polygonal curve C in the (x, y) plane, the goal is to estimate the parameters of the helix segment (radius r and pitch p) as well as its rotation matrix R such that the orthogonal projection of this helix segment onto the (x, y) plane fits the polygonal curve C .

The driving idea of our approach is to sample the helix segment and the curve C and to compute the matching between the two sets of points obtained from the sampling. Formally, the optimization problem to compute the helix parameters (radius r and pitch p) and the rotation matrix R is as follows:

$$\min_{r,p,R} \|M_{r,p}R - M_C\|_F^2 \quad \text{subject to } R^T R = I \quad (1)$$

$M_{r,p}$ is the set of points of the helix whose radius and pitch are r and p respectively. M_C is the set of points of the curve C . $R^T R = I$ is the orthonormality constraints for the rotation matrix. This approach which is based on the sampling of the helix and the input curve C implies two assumptions:

- The length of the helix segment (i.e. parameter α of the two endpoints of the segment) is defined. The choice of the value of α is discussed in Section 4.4. For the moment, we assume the value of α to be fixed.
- The helix segment and the curve C should be sampled such that a one-to-one correspondence exists between points of the helix and those of the curve C . This adaptive sampling is described in Section 5. In the remaining of Section 4, we assume that the one-to-one correspondence between the two point sets of the helix and the input curve C is valid.

The optimization problem (1) is difficult to solve because of the orthonormality constraints. We propose a method to compute an approximate solution (i.e. not exact solution) of the optimization problem; our approach consists of decomposing this problem into three steps that require solving simple linear systems.

In the first step, we compute a set of points by uniformly sampling the helix segment whose radius and pitch are equal to 1 (see Fig. 1(a)). Then we compute the optimal linear transformation matrix L that aligns the set of points of the sampled helix to those of the polygonal curve C (see Fig. 1(b)). Since the transformation matrix L is not constrained to be a matrix with orthonormal columns, it includes scale and shear components in addition to rotation. In consequence, the helix after transformation with L may not be any more a helix. In order to preserve the helix properties, L should be equivalent to a rotation transformation only, that is, L must have orthonormal columns.

In the second step, we modify the helix parameters (radius and pitch) such that the linear transformation matrix L becomes close to a matrix with orthonormal columns (see Fig. 1(c)). The idea is that the scale component of L can be set to one by modifying the helix parameters (radius and pitch). For instance, modifying the scale along the axis of the helix is actually equivalent to modifying its pitch.

Once the helix parameters have been estimated, the third step is to estimate the optimal rotation R matrix such that the points of the helix match those of the polygonal curve P , that is, we compute the optimal orthonormal matrix R that minimizes the root mean squared deviation between the points of the two curves (see Fig. 1(c)). This is a common problem known as Procrustes problem.

4.1. Computing the optimal transformation matrix L of the helix whose radius and pitch are equal to 1

Let $H_U(t)$ be the helix whose radius and pitch are equal to 1:

$$H_U(t) = \begin{bmatrix} \cos(t) \\ t \\ \sin(t) \end{bmatrix}$$

We compute a uniform sampling of the curve $H_U(t)$ with respect to the parameter t ; this is done by computing $H_U(t)$, for a set of n values of t equally distributed in the interval $[-\alpha, +\alpha]$, n being the number of points of the polygonal curve C . This uniform sampling is the set of values $\{\alpha_0, \alpha_1, \dots, \alpha_i, \dots, \alpha_{n-1}\}$ with $\alpha_i = \frac{2\alpha}{n-1}i - \alpha$ and i ranging from 0 to $n-1$.

The result of this sampling is a set of n points $\{H_U(\alpha_0), H_U(\alpha_1), \dots, H_U(\alpha_{n-1})\}$ which are equal to $\{H_U(-\alpha), H_U(\frac{2\alpha}{n-1} - \alpha), \dots, H_U(\alpha)\}$. The choice of α which defines the two endpoints of the helix segment is discussed in section 4.4. For the moment, we assume α to be a fixed value.

Without loss of generality, we assume that the centroid of the sets of points of the curve C and the sampled helix coincide with the origin of the coordinate system. This is done by subtracting from the point coordinates the coordinates of the respective centroid.

Given these two sets of paired points of the curve C and the sampled helix, we compute a linear transformation matrix that best aligns the coordinates of the sampled helix to those of C , i.e. the optimal linear transformation matrix L that minimizes the root mean squared deviation between the two paired sets of points. Formally, L is calculated such that the objective function (2) is minimal:

$$\|M_{U,\alpha}L - M_C\|_F^2 \quad (2)$$

$\|\cdot\|_F$ denotes the Frobenius matrix norm; $M_{U,\alpha}$ and M_C are the matrices representing the coordinates of the sampled helix and C respectively.

$$M_{U,\alpha} = \begin{bmatrix} \cos(-\alpha) & -\alpha & \sin(-\alpha) \\ \cos(\frac{2\alpha}{n-1} - \alpha) & \frac{2\alpha}{n-1} - \alpha & \sin(\frac{2\alpha}{n-1} - \alpha) \\ \vdots & \vdots & \vdots \\ \cos(\alpha) & \alpha & \sin(\alpha) \end{bmatrix} \quad M_C = \begin{bmatrix} x_{P,1} & y_{P,1} \\ x_{P,2} & y_{P,2} \\ \vdots & \vdots \\ x_{P,n} & y_{P,n} \end{bmatrix}$$

The linear transformation matrix L which is solution to the above optimization is:

$$L = (M_{U,\alpha}^T M_{U,\alpha})^{-1} M_{U,\alpha}^T M_C$$

Note that the dimension of L is 3 rows by 2 columns:

$$L = \begin{bmatrix} l_{1,1} & l_{1,2} \\ l_{2,1} & l_{2,2} \\ l_{3,1} & l_{3,2} \end{bmatrix}$$

This is because L is calculated with the 3-dimensional points of the helix and the 2-dimensional points of the polygonal curve C .

4.2. Estimation of the radius and pitch of the helix

The linear transformation matrix L which is solution to the optimization problem (2) includes shear, scale and rotation components. The idea is to modify the helix coefficients (radius and pitch) in a way to make the transformation matrix close to a matrix with orthonormal columns. Let r and p be the radius and the pitch of the helix respectively. The matrix representing the coordinates of the helix with r and p is:

$$\begin{bmatrix} r \cos(-\alpha) & -p\alpha & r \sin(-\alpha) \\ r \cos(\frac{2\alpha}{n-1} - \alpha) & p(\frac{2\alpha}{n-1} - \alpha) & r \sin(\frac{2\alpha}{n-1} - \alpha) \\ \vdots & \vdots & \vdots \\ r \cos(\alpha) & p\alpha & r \sin(\alpha) \end{bmatrix} = M_{U,\alpha} M_{rp} \quad \text{with } M_{rp} = \begin{bmatrix} r & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & r \end{bmatrix}$$

The objective function (2) is rewritten as follows:

$$\| (M_{U,\alpha} M_{rp}) (M_{rp}^{-1} L) - M_C \|_F^2 \tag{3}$$

The goal is to compute the unknown parameters r and p such that the matrix $(M_{rp}^{-1} L)$ becomes close to a matrix with orthonormal columns. If $(M_{rp}^{-1} L)$ is a matrix with orthonormal columns, then $(M_{rp}^{-1} L)^T (M_{rp}^{-1} L)$ is equal to the identity matrix. It follows that the objective function to minimize is:

$$\| (M_{rp}^{-1} L)^T (M_{rp}^{-1} L) - I \|_F^2 \tag{4}$$

I is the identity matrix.

$$(M_{rp}^{-1} L)^T (M_{rp}^{-1} L) = \begin{bmatrix} \frac{l_{2,1}^2}{p^2} + \frac{l_{1,1}^2 + l_{3,1}^2}{r^2} & \frac{l_{2,1}l_{2,2}}{p^2} + \frac{l_{1,1}l_{1,2} + l_{3,1}l_{3,2}}{r^2} \\ \frac{l_{2,1}l_{2,2}}{p^2} + \frac{l_{1,1}l_{1,2} + l_{3,1}l_{3,2}}{r^2} & \frac{l_{2,2}^2}{p^2} + \frac{l_{1,2}^2 + l_{3,2}^2}{r^2} \end{bmatrix}$$

After simplification, the objective function (4) becomes:

$$\| AX - B \|_F^2 \tag{5}$$

with $A = \begin{bmatrix} l_{2,1}^2 & l_{1,1}^2 + l_{3,1}^2 \\ l_{2,1}l_{2,2} & l_{1,1}l_{1,2} + l_{3,1}l_{3,2} \\ l_{2,1}l_{2,2} & l_{1,1}l_{1,2} + l_{3,1}l_{3,2} \\ l_{2,2}^2 & l_{1,2}^2 + l_{3,2}^2 \end{bmatrix}$, $X = \begin{bmatrix} \frac{1}{p^2} \\ \frac{1}{r^2} \end{bmatrix}$ and $B = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$.

r and p are the unknown variables. The vector X which is solution of the optimization problem (6) is:

$$X = (A^T A)^{-1} A^T B \tag{6}$$

The helix parameters r and p are easily computed using the vector X which is solution of the optimization problem (6).

4.3. Estimation of the rotation matrix

Once the parameters r and p have been computed, the objective function (3) can be rewritten as follows:

$$\| (M_{U,\alpha} M_{rp}) Z - M_C \|_F^2 \quad \text{with } Z = (M_{rp}^{-1} L) \tag{7}$$

The matrix Z is supposed to be a rotation matrix to align the points of the helix whose parameters are r and p to the points of the polygonal curve P . However, solving the objective function (7) for computing r and p does not guarantee that Z which is equal to $(M_{rp}^{-1} L)$ is a matrix with orthonormal columns. The next step is therefore to estimate the optimal rotation matrix R that minimizes the root mean squared deviation between the points of the helix and those of the polygonal curve P . The objective function to compute R is:

$$\| (M_{U,\alpha} M_{rp}) R - M_C \|_F^2 \quad \text{subject to } R^T R = I \tag{8}$$

The matrix containing the coordinates of the helix with parameters r , p and α is:

$$M_{U,\alpha} M_{rp} = \begin{bmatrix} r \cos(-\alpha) & -p\alpha & r \sin(-\alpha) \\ r \cos(\frac{2\alpha}{n-1} - \alpha) & p(\frac{2\alpha}{n-1} - \alpha) & r \sin(\frac{2\alpha}{n-1} - \alpha) \\ \vdots & \vdots & \vdots \\ r \cos(\alpha) & p\alpha & r \sin(\alpha) \end{bmatrix}$$

This is a common optimization problem known as orthogonal Procrustes problem in which the matrix R is constrained to be orthonormal (Schonemann, 1966). In our case, the Procrustes problem is unbalanced because the matrices $(M_U M_{rp})$ and M_C do not have same dimensions. Unlike orthogonal Procrustes problem which is solved using singular value decomposition, there is no closed-form analytical solution for the unbalanced Procrustes problems. Zhang and Du (2006) have proposed an iterative method to compute an approximation to the unbalanced Procrustes problem. Although iterative methods take longer time than those with closed-form solution, the computation time still remains very small because of the dimension of the matrix R which is 3 by 3. Note that unbalanced Procrustes problem may have multiple solutions. We refer to the work of Zhang and Du (2006) for discussion about multiple solutions.

The fitting error $Err(r, p, \alpha, R)$ which is the sum of the squares of the coordinate differences between corresponding points of the curve C and the projected helix is:

$$Err(r, p, \alpha, R) = \| (M_{U,\alpha} M_{rp}) R - M_C \|_F^2 \tag{9}$$

This fitting error is a measure to see how well the orthogonally projected helix fits the input curve.

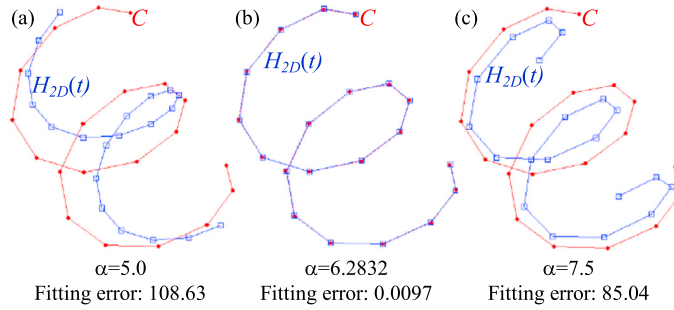


Fig. 2. The helix fitting with different values of α . The value $\alpha = 6.2832$ is chosen since it gives the lowest fitting error (b).

4.4. Estimation of the endpoint parameter of the helix segment

In sections 4.1, 4.2 and 4.3, we assumed that the parameter α of the two endpoints ($H(-\alpha)$ and $H(\alpha)$) of the helix segment to be fixed. Obviously, different values of α will give different helix parameters (radius r , pitch p and rotation matrix R) and different values of the fitting error (see Equation (9)). An example of helix fitting with different values of α is shown in Fig. 2.

Note that the endpoints $H(0)$ and $H(2\alpha)$ could be used instead of $H(-\alpha)$ and $H(\alpha)$. In fact, the helix with $H(0)$ and $H(2\alpha)$ can be made equivalent to the one with $H(-\alpha)$ and $H(\alpha)$ by applying a rotation of α about the z -axis.

We actually find the value of α such that the fitting error (Equation (9)) is the smallest. Suppose that the curve C with which we compute the helix fitting is sufficiently small; it follows that the length of the helix segment is small as well and we assume that α is in the interval $]0, 10\pi[$. We compute a set of α values equally distributed in the interval $]0, 10\pi[$. For each of these values α , we compute the corresponding Helix parameters (radius r , pitch p , rotation matrix R) and the fitting error; this step involves computing the optimal transformation matrix L (Section 4.1), computing the radius and pitch of the helix (Section 4.2) and the optimal rotation matrix as well as the fitting error (Section 4.3). We then select the value α for which fitting error is the smallest. The number of α values that we compute in the interval $]0, 10\pi[$ is about 500.

Any other method for unconstrained optimization and which does not require computing the derivative of the objective function such as (Lagarias et al., 1998) can also be used to compute the optimal value of α .

5. Adaptive sampling of the polygonal curve C

One essential assumption of our method is that there is a one-to-one correspondence between the points of the helix and those of the curve C . This implies that the helix and the curve must have same sampling. The main difficulty is how to sample the curve C such that the one-to-one correspondence is valid.

As mentioned in Section 4.1, the helix segment is uniformly sampled with respect to the curvilinear abscissa. The sampling of the curve C is a little bit more complicated. Unlike for the helix, the uniform sampling does not work for C . This is because we assume that the curve C is the orthogonal projection of a helix and the orthogonal projection is not isometry-invariant. In other words, the distances between the points are not preserved through the orthogonal projection; line segments of identical length project onto line segments of different length depending on their orientation with respect to the projection plane (see Fig. 3).

Our solution to this problem is to compute the sampling of the curve C using its arc length and the sampling of the projected helix segment that had been fitted to C . Our sampling algorithm works iteratively over the points of the sampled helix. Given a point $H_{2D}(t_j)$ of the projected sampled helix, we compute its arc length. The point of C corresponding to $H_{2D}(t_j)$ is then located along C using the arc length of $H_{2D}(t_j)$.

We first provide the analytical expressions of the arc length of the projected helix segment. Let $H(t)$ be a helix and R its rotation matrix. Without loss of generality, we assume that R includes only rotations about the x -axis. The rotations about the two other axes have no effect on the sampling.

$$H(t) = \begin{bmatrix} r \cos(t) \\ pt \\ r \sin(t) \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

θ , r and p are the rotation angle around the x -axis, the radius and the pitch of the helix respectively. The equation of the helix after rotation and projection onto the plane (x, y) is:

$$H_{2D}(t) = \begin{bmatrix} r \cos(t) \\ pt \cos(\theta) + r \sin(\theta) \sin(t) \\ 0 \end{bmatrix} \quad (10)$$

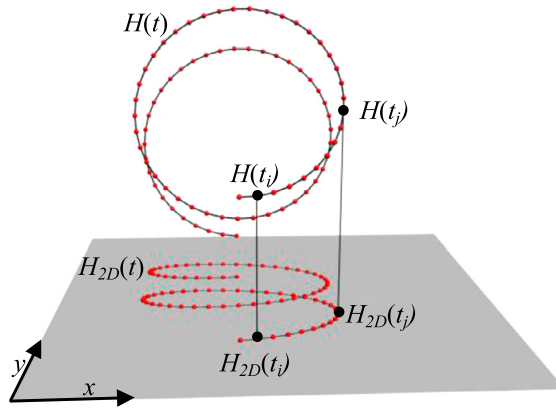


Fig. 3. The helix segment $H(t)$ with uniform sampling and its orthogonal projection $H_{2D}(t)$ onto the (x, y) plane. Unlike the helix segment, the sampling of the projected helix is not uniform; the inter-point distance (distance separating two red dots) at the point $H_{2D}(t_i)$ is larger than that of $H_{2D}(t_j)$. This is because the curve tangent at $H(t_i)$ is parallel to the plane whereas the one at $H(t_j)$ is perpendicular to the (x, y) plane. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

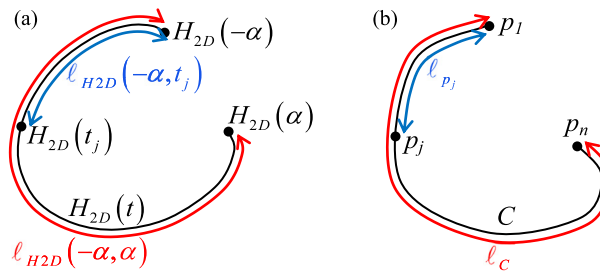


Fig. 4. The point p_j corresponding to $H_{2D}(t_j)$ is located along C such that its arc length is equal to ℓ_{p_j} (see Equation (11)).

The analytical expression of the arc-length $\ell_{H_{2D}}(t_i, t_j)$ of the projected helix segment whose endpoints are $H_{2D}(t_i)$ and $H_{2D}(t_j)$ is:

$$\ell_{H_{2D}}(t_i, t_j) = \int_{t_i}^{t_j} \left\| \frac{d}{dt} H_{2D}(t) \right\| dt = \int_{t_i}^{t_j} \sqrt{\left(\frac{d}{dt} (r \cos(t)) \right)^2 + \left(\frac{d}{dt} (pt \cos(\theta) + r \sin(\theta) \sin(t)) \right)^2} dt$$

Since the arc length expression is not in closed-form, we approximate its value using numerical methods.

Let us assume that a projected helix segment has been fitted to the curve C and that the radius r , the pitch p , the rotation angle θ and the two endpoints $H_{2D}(-\alpha)$ and $H_{2D}(\alpha)$ of the projected helix segment are known.

The adaptive sampling of C is computed iteratively over the n points of the sampled projected helix. Given a point $H_{2D}(t_j)$ of the projected helix, we compute its arc length $\ell_{H_{2D}}(-\alpha, t_j)$ (See Fig. 4.). The point p_j of C and which corresponds to $H_{2D}(t_j)$ is located along C such that its arc length ℓ_{p_j} is equal to:

$$\ell_{p_j} = \ell_{H_{2D}}(-\alpha, t_j) \frac{\ell_C}{\ell_{H_{2D}}(-\alpha, \alpha)} \tag{11}$$

$\ell_{H_{2D}}(-\alpha, \alpha)$ and ℓ_C are the arc lengths of the projected helix and the curve C respectively.

6. Combining the adaptive sampling with the helix fitting

As mentioned before, the adaptive sampling of the curve C is possible only if the projected helix has been initially fitted to C and if the helix parameters (r, p, α and R) have been computed. In fact, our adaptive sampling method (Section 5) and helix fitting (Section 4) are used alternately to compute successive approximations of the helix parameters. We first generate the curve C_{samp} by uniformly sampling the curve C (line 4 in the pseudo code shown in Fig. 6). The helix parameters are then estimated by fitting a helix to C_{samp} (line 9 in pseudo code). These helix parameters are then used to compute a new sampling of C_{samp} (line 10 in pseudo code). This process of helix fitting and curve sampling is iterated until the difference between two successive values of the fitting error fall below a threshold ε or increases. The fitting error is calculated with Equation (9) (see Section 4.3). The pseudo code of the method is given in Fig. 5.


```

1: Compute_Adaptive_Sampling_and_Helix_Matching(C)
2: Input: C
3: Output: r, p,  $\alpha$ , R, T
4:  $C_{\text{samp}} \leftarrow \text{Uniform\_Sampling\_Curve}(C)$ 
5:  $T \leftarrow \text{Compute\_Translation\_Component}(C_{\text{samp}})$ 
6:  $\text{err} \leftarrow +\infty$ ;
7: do
8:    $\text{prevErr} \leftarrow \text{err}$ ;
9:    $r, p, \alpha, R, \text{err} \leftarrow \text{Helix\_Fitting}(C_{\text{samp}})$ ;
10:   $C_{\text{samp}} \leftarrow \text{Adaptive\_Sampling\_Curve}(C, r, p, \alpha, R)$ ;
11: while ( $\text{prevErr} - \text{err} > \varepsilon$ )
12: return r, p,  $\alpha$ , R and T

```

Fig. 5. Pseudo code of the helix fitting and adaptive sampling.

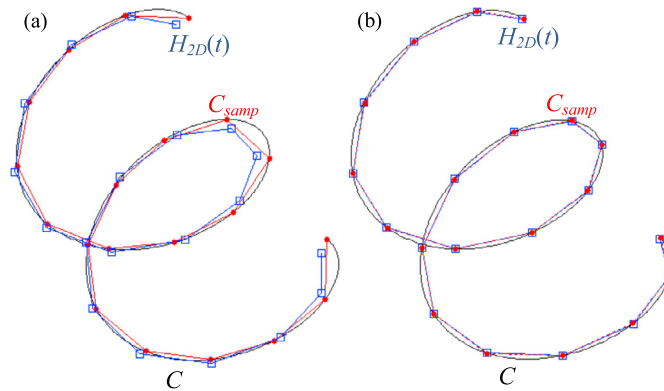


Fig. 6. In (a), the curve C_{samp} (shown in red) is obtained by uniformly sampling the curve C (shown in black). The fitting error between C_{samp} and the projected helix $H_{2D}(t)$ (shown in blue) is 1.83. In (b), after two iterations of helix fitting and adaptive sampling, the points of C_{samp} and the projected helix have almost same coordinates. The fitting error is reduced to 0.0097. The fitting errors are calculated with Equation (9). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In the pseudo code shown in Fig. 5, the function `Helix_Fitting` computes an estimation of the helix parameters and the fitting error as described in Section 4. The function `Adaptive_Sampling_Curve` computes the adaptive sampling of the polygonal curve C as described in Section 5.

The algorithm stops when the difference between the fitting error at the previous iteration and the one at the current iteration falls below the threshold ε or when the fitting error increases over the iterations. An example of convergence is shown in Fig. 6.

7. Results

Our algorithms have been implemented using the Matlab scripting language. We have applied our reconstruction method on curves with different shapes and lengths; Figs. 12 to 16 show the results of the reconstruction.

7.1. Comparison with the method by Chérin et al. (2014)

We have made a comparison with the previous method proposed by Chérin et al. (2014). The goal of their method is the same as ours: fitting an orthogonally-projected helix segment to a polygonal planar curve. However, the major difference is that the fitting is computed locally in the Chérin's method. In our method, the fitting is computed over the entire polygonal curve C .

The Chérin's method consists of two steps: they first find p_c , the point of maximum curvature of the planar curve C and compute its tangent T_{pc} . Next they identify the orientation and translation of the helix using the tangent T_{pc} and the point of maximum curvature p_c . The last step is to estimate the other helix parameters (radius and pitch) by fitting a helix with fixed orientation and fixed translation to the part of curve C in the neighborhood of p_c . The helix is found such that it best fits the input curve C and it passes through p_c and its axis is the parallel to T_{pc} .

The main drawback of the Chérin's method is its sensitivity to changes in the curve shape; slight modification in the curve coordinates at the point of maximum curvature causes changes in the orientation of the curve tangent T_{pc} and therefore generates large errors in the estimation of the helix orientation. In contrary, our method computes the helix parameters using an optimization over the entire input curve C . The advantage of our method over Chérin's method is illustrated in Figs. 7 and 8.

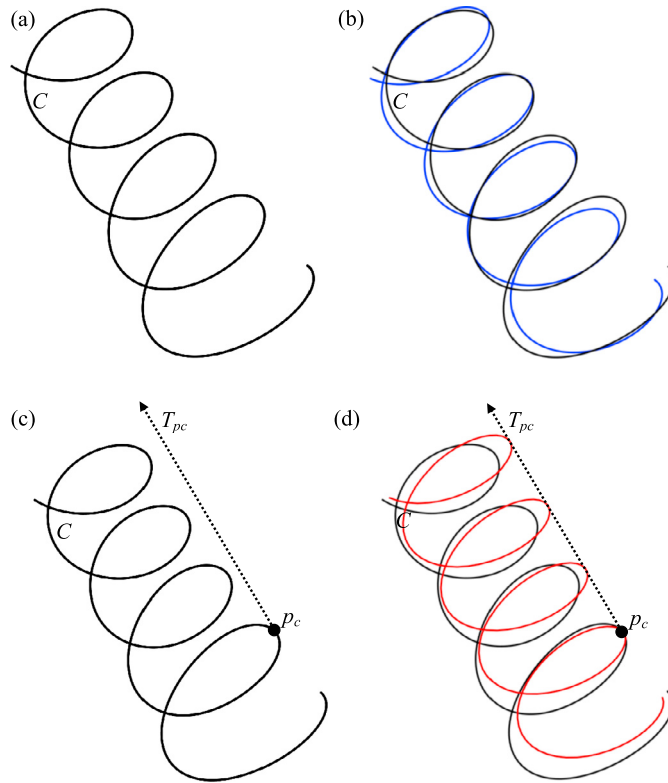


Fig. 7. The input polygonal curve is shown in (a). The helix produced by our method is shown in (b). In (c), the first step of Chérin et al. (2014) consisting of computing the point of maximum curvature p_c and its tangent T_{pc} . In (c), the helix produced by Chérin et al. (2014). This helix is constrained to pass through p_c and its tangent T_{pc} to be parallel to that of C ; this misalignment is due to the wrong orientation of the tangent T_{pc} at the p_c .

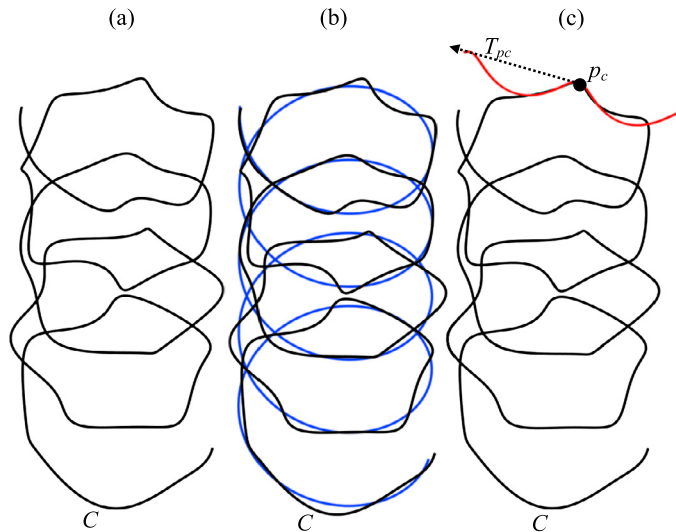


Fig. 8. The input polygonal curve C is shown in (a). The helix produced by our method is shown in (b). As shown in (c), the Chérin et al. (2014) method fails to compute the helix that fits the entire input curve C . This is because the computation of the tangent T_{pc} at the point of maximum curvature p_c does not capture the global orientation of the curve C .

7.2. Robustness with respect to noise

In order to demonstrate the robustness of our reconstruction method with respect to noise, we have conducted a set of experiments with a planar curve C which has been generated by orthogonally projecting a helix segment and by modifying

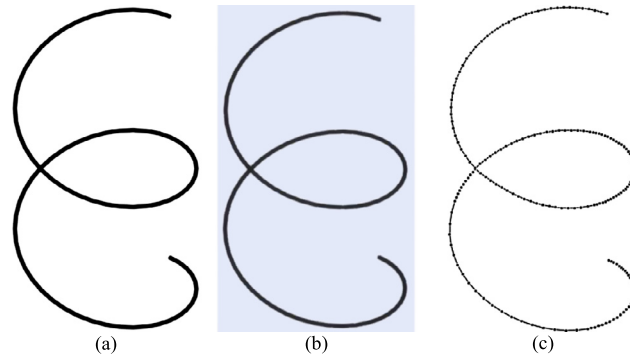


Fig. 9. The helix to be printed (a); the scan image of the printed helix (b) and the vectorization of the scan image (c).

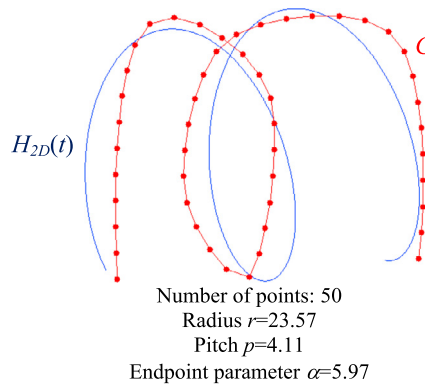


Fig. 10. Helix fitting to a hand-drawn curve. The hand-drawn curve is shown in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

its coordinates with noises. For this set of experiments, the number of points of C is 600. The metric to measure the quality of the helix fitting is computed as follows. We first find the point of C whose distance to its corresponding point on the helix is the largest. The error metric is then calculated by dividing this distance by the length of the curve C . In all the figures, this error is shown in percentage.

In the first experiment which is shown in Fig. 13, the radius r , pitch p and endpoint parameter α of the helix to generate the curve C have been set to 4, 1 and 2π respectively. The coordinates of the curves have been modified with two types of noises with increasing level of magnitude: high frequency noise and low frequency noise for the curves in the first row and second row of Fig. 8 respectively. As one can see, our algorithm succeeded in reconstructing the helix; the parameters (radius, pitch and endpoint parameter) of the reconstructed helix are very close to those of the helix that has been used to generate the curve C .

The second experiment which is shown in Fig. 14 is the same as the first one except that the helix segment to generate the curve C is much shorter; its endpoint parameter α has been set to 0.75π instead of 2π . As one can see, the quality of the reconstruction is lower than that of the first experiments; this is explained by the fact that the length of the helix to reconstruct is too short.

The third experiment in Fig. 15 shows quality of the reconstruction with helices of different lengths; noise of magnitude 0.5 has been added. It appears clearly that the quality of the reconstruction highly depends on the length of the helix (i.e. endpoint parameter α). The reconstruction of long helices (i.e. whose endpoint parameter α is equal or larger than π) gives much better results than the reconstruction of short helices (i.e. whose endpoint parameter α is smaller than π). If the curve C is too small, it does not contain enough information to reconstruct the helix accurately.

Our algorithm has also been tested in real condition with scan images of projected helices. The purpose of this experiment is to demonstrate the effectiveness of our method for the 3D reconstruction of helices from digitized images of engineering drawings. A set of helices with predefined parameters have been printed onto a paper; a bitmap image of this paper has been produced with scanner and converted into a vector image using the approach (Hilaire and Tombre, 2006) (see Fig. 9). Unlike the previous experiments for which the noise is artificially generated, the noise in this experiment is produced by the process of printing and scanning images of projected helices.

The experiments have been conducted with twenty printed helices whose radius and pitch are 2 and 1 respectively; the table below shows the largest fitting error and largest difference between the pitch of the initial helix and the pitch of the reconstructed helix. The largest difference for the radius is also shown.

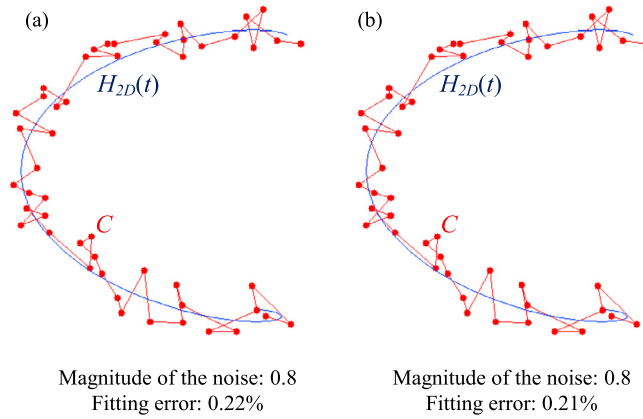


Fig. 11. Comparison of the helix fitting of our method (a) with the one using the Levenberg–Marquardt method (b).

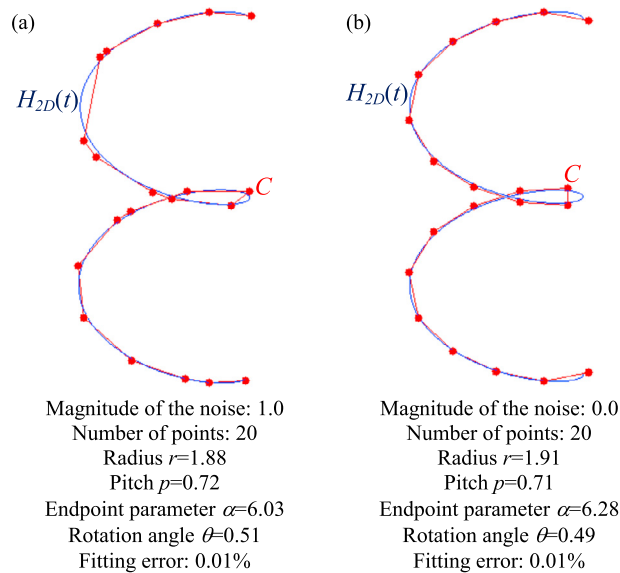


Fig. 12. Helix fitting to a curve with irregular sampling (a) and with uniform sampling (b).

Largest fitting error	Largest difference for the pitch	Largest difference for the radius
0.04%	0.01	0.007

Our algorithm has also been tested with hand-drawn curves. An example of helix fitting to hand-drawn curves is shown in Fig. 10.

7.3. Evaluation of the approximate solution

Our method does not compute the exact solution to the helix fitting; this is because the computation of the radius r and the pitch p of the helix and its rotation matrix R are done separately although these three variables are interdependent. In order to show that our method computes a solution which is close to the optimal one, we compare our algorithm with another one that computes the parameters r , p and the rotation matrix R all together using a non-linear optimization method. The optimization problem of the helix fitting is defined as follows:

$$\min_{r,p,R} \|(M_{U,\alpha} M_{rp})R - M_C\|_F^2 \quad \text{subject to } R^T R = I \tag{12}$$

This optimization problem is rewritten so as to remove the non-linear constraints $R^T R = I$. R is a rotation matrix which is decomposed into three rotation matrices R_x , R_y and R_z about the x , y and z -axis respectively. Let θ_x , θ_y and θ_z be the rotation for the x , y and z axis respectively. Then, R_x , R_y and R_z are defined as follows

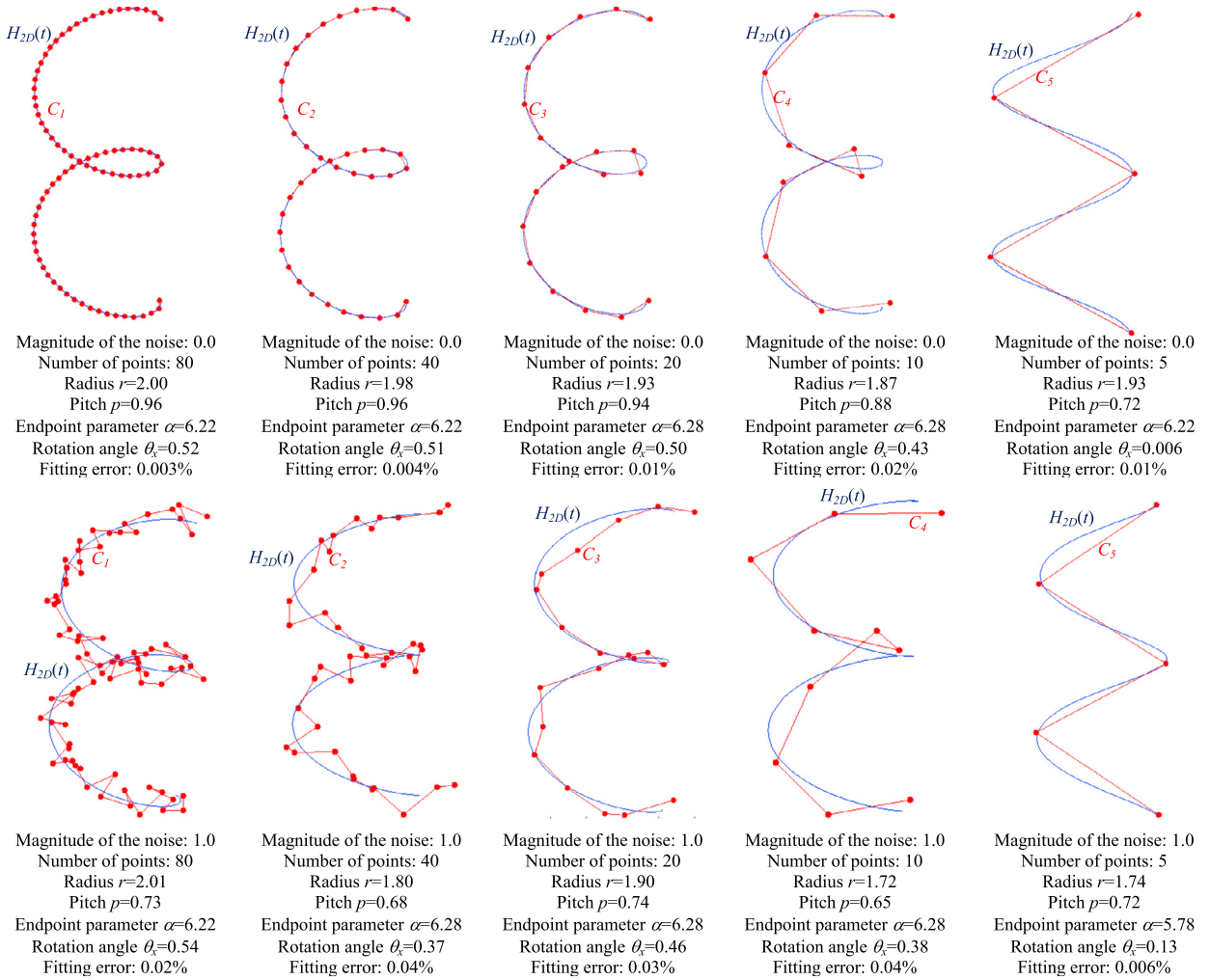


Fig. 13. Helix fitting with different levels of sampling and different levels of noise. The values r , p , α and θ_x are the radius, the pitch, the end point parameter and the orientation angle with respect to the x axis respectively.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \quad R_y = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \quad R_z = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation matrix is defined as $R = R_z R_y R_x$. The main advantage of such representation is that the orthonormality constraints are not any more needed; the optimization problem (12) becomes:

$$\min_{r, p, \theta_x, \theta_y, \theta_z} \|(M_{U, \alpha} M_{rp}) R_z R_y R_x R_{Pro} - M_C\|_F^2 \tag{13}$$

R_{Pro} is the orthogonal projection matrix defined as follows:

$$R_{Pro} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

We solve this optimization problem using the Levenberg–Marquardt algorithm (Levenberg, 1944) which is available in Matlab; any other non-linear optimization method could be actually used. A set of experiments have been conducted on a set of input curves. These curves have been generated by orthogonally projecting a helix whose radius, pitch and length are 2, 4.5 and 3.3 respectively. The coordinates of the curve points have then been modified with different levels of uniform noise. For each of these curves, we have run our algorithm to compute the helix coefficients r and p as well as its rotation matrix R . Then these values have been used as initial values to the Levenberg–Marquardt algorithm which has been run to solve the optimization problem (13). The purpose is to show how much the solution given by our algorithm can be improved using the Levenberg–Marquardt algorithm. The results are listed in the table below:

Magnitude of noise	Fitting error of our method	Fitting error after applying the Levenberg–Marquardt optimization
0.00	0.03%	0.02%
0.20	0.04%	0.02%
0.40	0.09%	0.02%
0.60	0.20%	0.16%
0.80	0.21%	0.2%

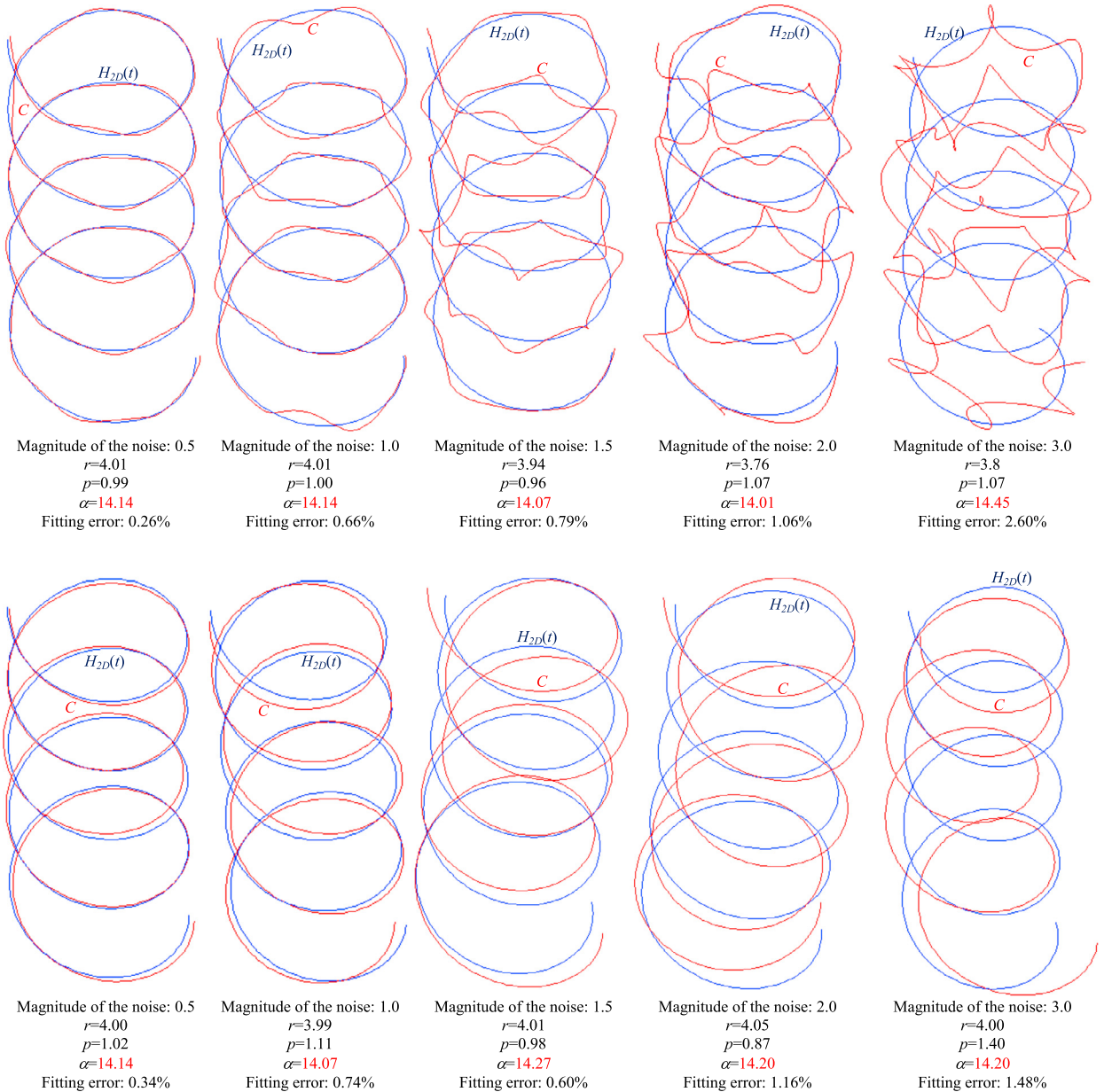


Fig. 14. The input curve C (shown in red) has been generated by uniformly sampling a helix and by orthogonally projecting the sampled helix onto the (x, y) plane. The parameters of the helix to generate the curve C are: $r = 4$, $p = 1$ and $\alpha = 2\pi$. Uniform noise with different levels of magnitude has been added to the coordinates of the curve C . The helix fitting has been computed for each of these curves. The parameters (radius r , pitch p , the parameter of the endpoints α) of the fitted helix (depicted in blue) are shown for the different levels of noises and with different levels of frequency. The number of points is 600 for the curve C . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

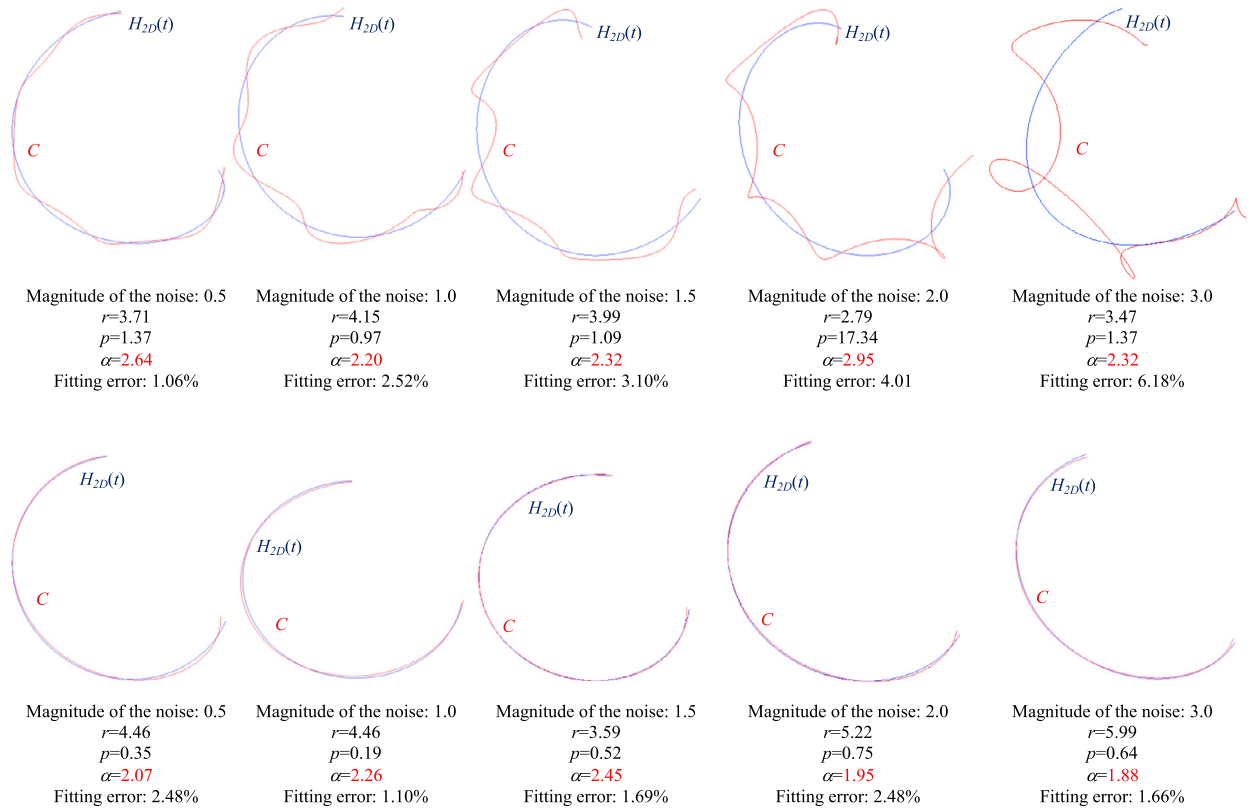


Fig. 15. The input curve C (shown in red) has been generated by uniformly sampling a helix and by orthogonally projecting the sampled helix onto the (x, y) plane. The parameters of the helix to generate the curve C are: $r = 4$, $p = 1$ and $\alpha = 2.36$. Uniform noise with different levels of magnitude has been added to the coordinates of the curve C . The helix fitting has been computed for each of these curves. The parameters (radius r , pitch p , the endpoint parameter α) of the fitted helix (depicted in blue) are shown for the different levels of noises. The number of points is 600 for the curve C shown in first row and 20 for the curve C in the second row. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

It appears clearly that the solution computed by our algorithm is not the optimal one. The Levenberg–Marquardt optimization could find a solution with a lower fitting error for almost all the curves. However, the improvement is rather small especially for curves with high magnitude of noise. It is actually not possible to distinguish the difference between the helix fitting of the two methods as shown in Fig. 10. On the other hand, the computation time of our method is small; it ranges from 0.2 to 0.5 seconds for a number of points of the input curve C ranging from 100 to 600. The Levenberg–Marquardt method takes about 10 to 15 seconds.

7.4. Robustness with respect to coarse sampling

In this section, we analyze the performances of our algorithm for input curves with different levels of sampling and different level of noises. The input curve C is generated by orthogonally projecting a helix curve whose radius r , pitch p , endpoint parameter α (i.e. length of the curve) and rotation angle with respect to the x -axis θ_x are 2, 1, π and 0.8 respectively. The number of points has been set to 80, 40, 20, 10 and 5 for C_1 , C_2 , C_3 , C_4 and C_5 respectively. The helix fitting has been computed twice for each of these curves, the first time without noise (first row of Fig. 12) and the second time with uniform noises (second row of Fig. 12).

As one may observe, our algorithm successfully computes the fitting of the helix; the fitting error remains small even for the curve C_5 which has 5 points only. On the other hand, our algorithm fails to compute the rotation angle θ_x of C_5 ; its value is 0.006 and it should be 0.8. This is because C_5 has so few points that some of the curve details are missing. In particular, C_5 does not show the curve loop which is visible in other curves C_1 to C_4 .

7.5. Robustness with respect to irregular sampling

Here, we analyze the behavior of our algorithm for non-uniformly-sampled curves. As mentioned in Section 6, our algorithm starts by uniformly sampling the input curve; the line segments that compose the input curve must have equal length. In this experiment, the helix fitting to an irregularly-sampled curve and the helix fitting to the same curve with uniform sampling are compared. The results are shown in Fig. 11.

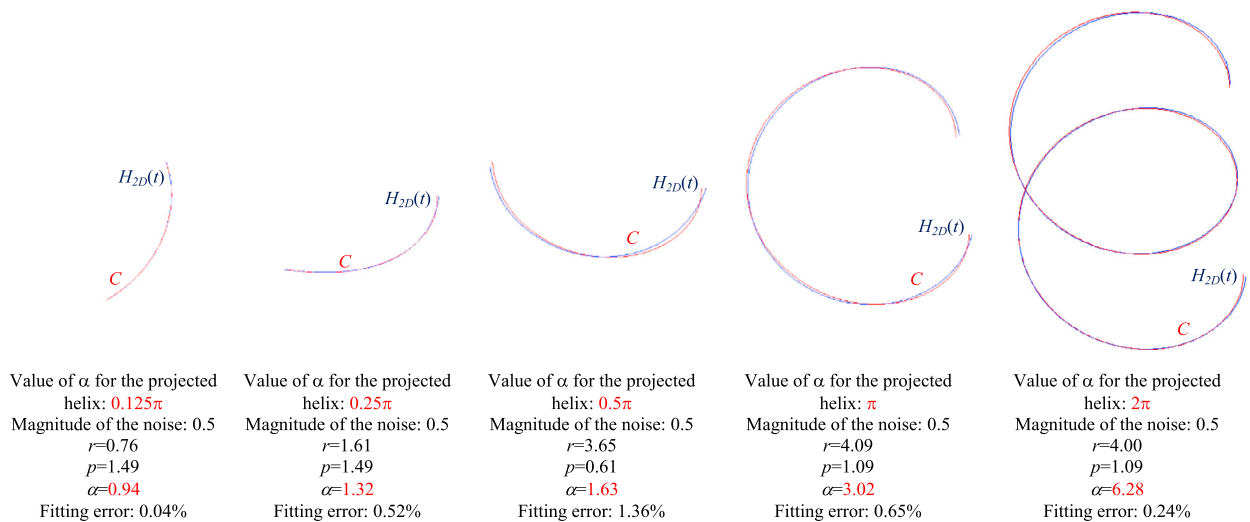


Fig. 16. The input curve C (shown in red) has been generated by uniformly sampling a helix and by orthogonally projecting the sampled helix onto the (x, y) plane. The parameters of the helix to generate the curve C are: $r = 4$, $p = 1$; the endpoint parameter α (i.e. the length of the helix) has been set from 0.125π to 2π . Noise with magnitude of 0.5 has been added to the data. The parameters (radius r , pitch p , the parameter of the endpoints α) of the fitted helix (depicted in blue) are shown for each of these endpoint parameters α . The number of points is 600 for the curve C . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As one may observe, input curves with irregular sampling do not affect much the performance of our algorithm. This is because the curve sampling and the helix fitting are used alternately in the main loop of our algorithm. If the sampling of the input curve is irregular, then this irregular sampling is removed at the next iteration.

8. Conclusion

We have described a method to compute a helix segment such that its orthogonal projection fits a planar polygonal curve. A future work would be to extend this method for the reconstruction under perspective projection.

References

- Brown, E., Wang, P., 1996. 3D object recovery from 2D images: a new approach. In: *SPIE Proc. Robotics and Computer Vision*, vol. 2904, pp. 138–145.
- Chérin, N., Cordier, F., Melkemi, M., 2014. Modeling piecewise helix curves from 2D sketches. *Comput. Aided Des.* 46, 258–262.
- Christopher, J.A., Swanson, R., Baldwin, T.O., 1996. Algorithms for finding the axis of a helix: fast rotational and parametric least-squares methods. *Comput. Chem.* 20 (3), 339–345.
- Cordier, F., Seo, H., Melkemi, M., Sapidis, N.S., 2013. Inferring mirror symmetric 3D shapes from sketches. *Comput. Aided Des.* 45 (2), 301–311.
- Derouet-Jourdan, A., Bertails-Descoubes, F., Thollot, J., 2013. Floating tangents for approximating spatial curves with piecewise helices. *Comput. Aided Geom. Des.* 30 (5), 490–520.
- Enkhbayar, P., Damdinsuren, S., Osaki, M., Matsushima, N., 2008. HELFIT: helix fitting by a total least squares method. *Comput. Biol. Chem.* 32 (4), 307–310.
- Ghosh, S., 2010. Geometric approximation of curves and singularities of secant maps. A differential geometric approach. PhD dissertation. University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science. Advisors: G. Vegter and J. Rieger, 185 pages.
- Goriely, A., Neukirch, S., Hausrath, A., 2009. Polyhelicities through n points. *Int. J. Bioinform. Res. Appl.* 5 (2), 118–132.
- Hilaire, X., Tombre, K., 2006. Robust and accurate vectorization of line drawings. *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (6), 890–904.
- Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.E., 1998. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM J. Optim.* 9 (1), 112–147.
- Levenberg, K., 1944. A method for the solution of certain problems in least-squares. *Q. Appl. Math.* 2, 164–168.
- Marchal, X., Bertails, F., Hérouy, F., 2009. Reconstruction de trochoïdes à partir de courbes 2D. Technical report.
- Nievergelt, Y., 1997. Fitting helices to data by total least squares. *Comput. Aided Geom. Des.* 14 (8), 707–718.
- Piuze, E., Kry, P.G., Siddiqi, K., 2011. Generalized helicoids for modeling hair geometry. *Comput. Graph. Forum* 30 (2), 247–256.
- Schonemann, P., 1966. A generalized solution of the orthogonal Procrustes problem. *Psychometrika* 31, 1–10.
- Shoji, K., Kato, K., Toyama, F., 2001. 3-D interpretation of single line drawings based on entropy minimization principle. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2, 90–95.
- Wither, J., Bertails, F., Cani, M.-P., 2007. Realistic hair from a sketch. *Shape Model. Int.*, 33–42.
- Zhang, Z., Du, K., 2006. Successive projection method for solving the unbalanced Procrustes problem. *Sci. China Ser. A* 49 (7), 971–986.