Computer-Aided Design 46 (2014) 258–262

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Technical note

# Modeling piecewise helix curves from 2D sketches

Nicolas Cherin, Frederic Cordier *, Mahmoud Melkemi

*LMIA, Université de Haute Alsace, France*

## HIGHLIGHTS

- Our method reconstructs a piecewise helix curve from a 2D sketch.
- Helices are computed such that their projection matches the 2D curve.
- An optimization is performed to minimize the tangent discontinuity of the helices.

## ARTICLE INFO

*Keywords:*
3D reconstruction
Piecewise helix
Sketch-based modeling

## ABSTRACT

We describe a method for a reconstructing piecewise helix curve from its 2D sketch. The system takes as input a hand-drawn polygonal curve and generates a piecewise helix curve such that its orthogonal projection matches the input curve. The first step is an algorithm to generate a set of helices such that their orthogonal projection approximates the input curve. This step is followed by a global optimization to minimize the tangent discontinuity of the junctions of the helices while keeping the fitting error small.



## 1. Introduction

Helices are curves whose curvature and torsion are constant. These curves are very common in nature and among man-made objects: hair curls, coiled phone cords, spring coils, tendrils of climbing plants, etc. In this paper, we present a method to fit a curve which is the orthogonal projection of a piecewise helix curve to a 2D polygonal curve provided by the user. The main difficulty resides in the fact that the helix and the curve that is the projection of the helix have different curvature.

## 2. Related work

One of the most common approaches [1,2] of sketch-based modeling is to let the user draw the 3D shape using straight line segments. The main disadvantage of these approaches is that they are limited to the reconstruction of rectilinear shapes; they cannot be applied to free-form curves. Several researchers have worked on the modeling of 3D surfaces [3,4] and 3D curves using sketches. In the method proposed by Cohen et al. [5], the user generates a 3D curve by drawing the curve and its shadow on the floor plane. Other researchers have proposed a 3D reconstruction method by assuming that the reconstructed curves are mirror-symmetric [6].

Several researchers have worked on the problem of fitting a piecewise helix curve, a so-called super-helix, to a 3D polygonal curve. Piuze et al. [7] proposed an optimization method to fit piecewise generalized helicoids to 3D curves. Goriely et al. [8] have shown a method to compute the polyhelices that pass through an arbitrary set of points. A polyhelix is a curve consisting of a sequence of connected helical segments. Ghosh [9] introduced the bi-helices, a $G^1$ curve smoothly connecting 2 helices. Recently, Derouet-Jourdan et al. [10] have presented an algorithm to approximate Bezier curves with $G^1$ piecewise helices.

The closest work to ours is a sketch-based interface for modeling virtual hair for 3D characters [11,12]. In this paper, the authors describe a method to create 3D helices from a 2D sketch. Their method only works under the assumption that the main axis of the reconstructed helix is parallel to the sketching plane. In contrast, our method is able to reconstruct piecewise helix curves with any orientation.

In this paper, we address a different problem, which is the fitting of a curve which is the orthogonal projection of a helix to a 2D polygonal curve. In contrast to the problem of fitting a helix to a 3D polygonal curve, the curvature of the 2D polygonal curve cannot be directly used for generating the helix. This is because the helix curve and its projection have different curvature. To the best of our knowledge, such a problem has never been addressed until now.

## 3. Overview

A key element of our approach is a method to compute a helix curve such that its orthogonal projection matches the entire or a portion of a 2D polygonal curve $C_I$. In Section 5, we describe some properties of the projected helix curve. Based on these properties,

* Corresponding author.
*E-mail address:* fredcord@gmail.com (F. Cordier).

0010-4485/$ – see front matter 
http://dx.doi.org/10.1016/j.cad.2013.08.042


**Fig. 1.** Overview of the approach: the first step, which is shown in (b), consists of fitting a set of projected helix curves $\{H_1, H_2, H_3\}$ to different parts of the input polygonal curve $C_I$ as shown in (a). In the second step, which is shown in (c), the helices curves $\{H_{3D,1}, H_{3D,2}, H_{3D,3}\}$ are computed in 3D from their projection $\{H_1, H_2, H_3\}$. Then, a fine fitting is performed to reduce the tangent discontinuity at the junctions of these helices.

we present a method to compute the fitting of a projected helix curve to the input curve $C_I$ (Section 6).

In most cases, the reconstructed helix curve does not fit the entire input curve $C_I$; it is rare that the user draws a curve which is the exact projection of a helix curve. Thus, the matching is done in a recursive manner with several helices. After fitting the first helix $H_1$ to $C_I$, we identify the parts of $C_I$ for which the matching is not good and compute the fitting on those parts with the next helix $H_2$. This process is repeated recursively until all parts of the input curve have been fitted with a helix. The result is a piecewise helix curve that approximates the entire curve $C_I$. This process is explained in Section 6.

Once the coefficients of the piecewise helix curve have been generated, we compute its 3D position. The resulting curve is $G_1$ discontinuous since it does not guarantee the tangent continuity at the junctions between adjacent helices. Thus, we modify the helix coefficients such as to minimize the tangent discontinuity at the helix junctions while maintaining a small matching error. This last step is explained in Section 7.

The user draws a 2D polygonal curve $C_I$ on the plane $(z = 0)$, the so-called sketching plane. This 2D polygonal curve $C_I$ is specified by a sequence of $p$ vertices $v_1, v_2 \ldots v_p$ and $p-1$ line segments connecting consecutive vertices. This curve is the orthogonal projection of a piecewise helix curve onto the sketching plane $(z = 0)$ (Fig. 1(a)). This implies that the $x$ and $y$ coordinates of the 3D vertices of the curves are known. The $z$ coordinates have to be computed.

## 4. Properties of helix curve

### 4.1. Helix curve

A helix curve is a 3D curve whose curvature and torsion are constant. The equation of the helix curve that passes through the origin of the coordinate system and whose main axis is parallel to the $y$-axis is:

$$H_{\text{Prim}}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} a(\cos(t) - 1) \\ bt \\ a\sin(t) \end{pmatrix} \quad \text{with } a \geq 0 \text{ and } b \geq 0.$$

$a$ and $b$ are respectively the radius and the pitch of the helix. Now, we consider a helix curve with an arbitrary orientation and its orthogonal projection onto a plane $(z = 0)$. Among the three possible rotations (about the $x$, $y$ and $z$-axes), only the one about the $x$-axis changes the curvature of the projected helix. The $y$-rotation is reversible by substituting $t$ with $t = t + \alpha$ and the $z$-rotation is reversible by applying a rotation in the plane $(x, y)$. Without loss of generality, we only take into account the rotation about the $x$-axis. Let $c$ be the rotation angle about the $x$-axis. The parametric equation $H_{3D,R(t)}$ of the helix with a rotation is:

$$H_{3D,R}(t) = \begin{pmatrix} H_{3D,R,x}(t) \\ H_{3D,R,y}(t) \\ H_{3D,R,z}(t) \end{pmatrix} = \begin{pmatrix} a(\cos(t) - 1) \\ bt\cos(c) - a\sin(c)\sin(t) \\ bt\sin(c) + a\cos(c)\sin(t) \end{pmatrix}.$$

The parametric equation of the helix curve after the projection onto the plane $(z = 0)$ is:

$$H(t) = \begin{pmatrix} H_{3D,R,x}(t) & H_{3D,R,y}(t) & 0 \end{pmatrix}^{\mathrm{T}}.$$

### 4.2. Point of maximum curvature of the projected helix curve

The curvature of the projected helix curve $H(t)$ is:

$$\kappa_H(t) = \frac{|a(b\cos(c)\cos(t) - a\sin(c))|}{\left(a^2\sin(t)^2 + (b\cos(c) - a\sin(c)\cos(t))^2\right)^{\frac{3}{2}}}.$$

**Proposition 1.** *The value of $\kappa_H(t)$ is maximum when $t$ is a multiple of $\pi$ (i.e. $t = 0$ or $t = \pi$).*

This property is demonstrated in the Appendix. Given a polygonal curve $C_I$, we identify the vertex $v_s$ for which the curvature is maximal. The above property implies that $t$ is equal to 0 at $v_s$; that is, we have two points, $v_s$ and $H(0)$ for which $t$ has the same value. This vertex $v_s$ is used as a starting point for computing the fitting of the projected helix to $C_I$.

### 4.3. Defining a local coordinate frame

In order to compute how well the projected helix curve fits to the polygonal curve $C_I$, we must compute the distance between the points of $C_I$ and the corresponding points of the projected helix curve. Thus, a preliminary step is to align the two curves so that the point coordinates of the two curves can be compared. To do so, we define a local coordinate frame at the point $t = 0$. The $y$-axis is the unit vector parallel to the curve tangent at $t = 0$ and the $x$-axis is normal to the $y$-axis such that $(x, y)$ is oriented clockwise. The unit tangent of the projected helix curve $H(t)$ is:

$$\hat{T}_H(t) = \begin{pmatrix} -\dfrac{a\sin(t)}{\sqrt{a^2\sin(t)^2 + (b\cos(c) - a\sin(c)\cos(t))^2}} \\ \dfrac{b\cos(c) - a\sin(c)\cos(t)}{\sqrt{a^2\sin(t)^2 + (b\cos(c) - a\sin(c)\cos(t))^2}} \\ 0 \end{pmatrix}.$$

## 5. Estimation of the curvature and tangent of the polygonal curve

Similar to the projected helix curve, the computation of the curvature and unit tangent vector of the polygonal curve $C_I$ is also needed. In this paper, we adopt the method described by Lewiner et al. [13] which has been shown to provide good performance compared to other methods. Let $v_i$ be a vertex of $C_I$ and for which we want to estimate the curvature. The key idea is to compute a quadratic parametric curve $C_i(t)$ to approximate the polygonal curve $C_I$ in the vicinity of $v_i$. The equation of this curve is:

$$C_i(t) = \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \begin{pmatrix} x_i + x'_i t + \dfrac{1}{2}x''_i t^2 \\ y_i + y'_i t + \dfrac{1}{2}y''_i t^2 \end{pmatrix}$$

$x_i$ and $y_i$ are the position of the vertex $v_i$. $x'_i$, $x''_i$, $y'_i$ and $y''_i$ are the 1st and 2nd derivatives respectively of the $x$ and $y$ coordinates of $v_i$; $t$ is the curve parameter. Given the quadratic parametric curve $C_i(t)$ which approximates the curve $C_I$ in the vicinity of $v_i$, we compute the curvature $\kappa_{vi}$ and unit tangent vector $\hat{T}_{vi}$ at the point $v_i$.

## 6. Fitting the projected helix to $C_I$

The input of the algorithm is a polygonal curve $C_I$ with uniform sampling. The curve $C_I$ is composed of $p$ vertices $v_1, \ldots, v_p$ connected with line segments of roughly equal length.

First, we compute the curvature for all vertices of $C_I$ using the method described in Section 5 and identify the vertex $v_s$ whose curvature is maximal. This vertex is the seed vertex from which we start the fitting of the projected helix. The goal is to fit a projected helix curve to the part of the polygonal curve in the neighborhood of $v_s$.

### 6.1. Alignment of $C_I$ to the helix curve

Next, we compute the unit tangent vector $\hat{T}_{v_s}$ at the vertex $v_s$ belonging to $C_I$ using the method described in Section 5 and build a local coordinate frame $F_{v_s}$ whose origin is $v_s$ and its $y$-axis is $\hat{T}_{v_s}$; its $x$-axis is normal to the $y$-axis and its orientation is defined such that $(x, y)$ is counterclockwise. Then, we align the coordinate frame $F_{v_s}$ to $F_H$, $F_H$ being the local coordinate frame of the projected helix curve at $t = 0$. This alignment requires applying a translation, rotation and possibly a reflection to the vertex coordinates of $C_I$. After this transformation, vertices of $C_I$ and points of the projected helix curve are now defined in the same coordinate system.

### 6.2. Estimating the helix coefficients using two vertices of $C_I$

The next step is to estimate the unknown coefficients $a$, $b$ and $c$ of the projected helix so that it best approximates the curve $C_I$ in the neighborhood of $v_s$. Note that $t$ is also unknown except at $v_s$ and its value changes along the projected helix curve. This is done with two steps. First, we choose a vertex $v_n$ in the neighborhood of $v_s$ and compute the unknown coefficients $a$, $b$ and $c$ with $v_s$ and $v_n$. Second, we estimate the overall matching error of the projected helix to the polygonal curve $C_I$ with the coefficients $a$, $b$ and $c$ previously computed. This process of choosing $v_n$, computing the unknown coefficients for $v_n$ and $v_s$ and computing the overall matching error is repeated several times and the values of $a$, $b$ and $c$ that give the lowest matching error are chosen.

Since the local coordinate frame of $C_I$ at vertex $v_s$ has been aligned with the coordinate frame of the projected helix curve $H(t)$ at $t = 0$, the curve point $H(0)$ and $v_s$ have their coordinates equal to 0. $\kappa_{vs}$ is the curvature of $C_I$ at the vertex $v_s$. We select one vertex $v_n = \begin{pmatrix} x_{vn} & y_{vn} \end{pmatrix}^{\mathrm{T}}$ in the neighborhood of $v_s$; its unit tangent vector is $\hat{T}_{vn} = \begin{pmatrix} x_{\hat{T}vn} & y_{\hat{T}vn} \end{pmatrix}^{\mathrm{T}}$. We suppose that the projected helix curve passes through the vertices $v_s$ and $v_n$. In particular, $v_n$ is located on the projected helix curve at $t = \alpha$. Then, we write the system of equations:

$$v_n = H(\alpha)$$
$$\hat{T}_{v_n} = \hat{T}_H(\alpha).$$

The first equation implies that the points $H(\alpha)$ and $v_n$ must have the same coordinates and the second equation implies that $\hat{T}_{v_s}$, the unit tangent vector at $v_n$, and $\hat{T}_H(\alpha)$, the unit tangent vector at $H(\alpha)$, must be equal.

The above system of equations does not have a solution; it is solved in the least square sense using an optimization method whose unknown variables are $a, b$ and $c$ and whose objective function is:

$$E(a, b, c) = \beta_D \|v_n - H(\alpha)\|^2 + (1 - \beta_D) \left\| \hat{T}_{v_n} - \hat{T}_H(\alpha) \right\|^2.$$

Also note that $v_n$ can be any vertex in the neighborhood of $v_s$. In our implementation, we set $\alpha$ to $\pi/4$. $\beta_D$ is a weight to balance the importance of minimizing either $\|v_n - H(\alpha)\|^2$ or $\|\hat{T}_{v_n} - \hat{T}_H(\alpha)\|^2$. The unknown variables $a$, $b$ and $c$ are not linearly related. Methods such as the simplex search method proposed by Lagarias et al. [14] are used to solve such an optimization problem. Fig. 2 shows projected helices computed for different vertices $v_n$.

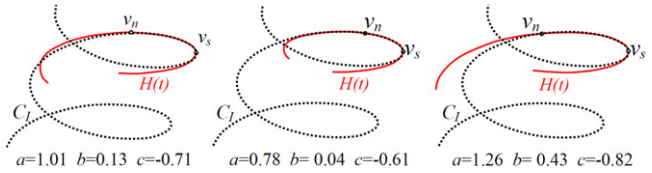


**Fig. 2.** Helix curve (shown in red) generated from different vertices $v_n$. The coefficients $a$, $b$ and $c$ of each helix are shown below the curve. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

### 6.3. Estimating the fitting of the helix to other vertices of $C_I$

Now that we have computed the helix coefficients for the two vertices $v_s$ and $v_n$, the next step to check the fitting of the projected helix curve to the other vertices. The goal is to find the number of consecutive vertices along $C_I$ for which the fitting is good. Let $v_{s+j}$ be a vertex in the neighborhood of $v_s$ and whose index is $s+j$ with $j$ starting from 1. The vertex of index $s+j$ is the vertex after a walk of $j$ vertices from $v_s$ and by following the positive orientation of the curve.

For each vertex $v_{s+j}$, we first compute the parameter $t_{s+j}$ such that the point on the projected helix curve $H(t_{s+j})$ is the closest point to $v_{s+j}$. Next we compute a matching error which measures the difference of coordinates between $v_{s+j}$ and its counterpart $H(t_{s+j})$ on the projected helix curve:

$$E(v_{s+n}, t_{s+n}) = \|v_{s+n} - H(t_{s+n})\|^2.$$

The iterations over $v_{s+j}$ stop when the matching error $E(v_{s+n}, t_{s+n})$ is larger than a user-defined threshold $E_{\mathrm{Max}}$. This threshold $E_{\mathrm{Max}}$ is small enough so that the two points are considered to have the same location. Typically, its value is set to 0.1% of the total curve length of $C_I$.

The same process is repeated for the adjacent vertices $v_{s-k}$ in the other side of $v_s$ with the vertex index $k$ starting from 1. Finally, we compute the value $j + k$ which is the number of consecutive vertices for which the matching error is below the threshold $E_{\mathrm{Max}}$. This number of vertices indicates the portion of the curve $C_I$ which is well fitted with the projected helix curve whose coefficients have been calculated.

The overall fitting process requires using the two algorithms described in Sections 6.2 and 6.3 alternately. We choose a vertex $v_n$ among the 2 by NMAX neighbors of $v_s$ and compute the helix coefficients using $v_s$ and $v_n$. Then we compute how well the projected helix curve with the computed coefficients can fit the rest of the curve. We choose the coefficients $a$, $b$ and $c$ for which the helix curve covers the largest portion of the polygonal curve $C_I$.

### 6.4. Approximating $C_I$ with multiple helices

A single helix curve is usually not sufficient to approximate the entire curve $C_I$. In such a case, we build a piecewise helix curve to approximate $C_I$.

The process is done iteratively. After fitting a helix to $C_I$, we identify the part of $C_I$ whose vertices have not been fitted yet. The fitting of the next helix is done on those vertices. As mentioned in Section 6, the fitting of a new projected helix start by finding $v_s$, the vertex of highest curvature among the vertices which have not been fitted yet.

The output is a set of projected helices: $\{H_1, H_2 \ldots H_n\}$ sorted according to their location along $C_I$. Each projected helix $H_i$ has a common vertex with each of its two adjacent neighbors $H_{i-1}$ and $H_{i+1}$. These common vertices are denoted as junction vertices. In addition, the projected helix orientations are consistent with each other, i.e. the curve $H_i$ begins at the last endpoint of the previous curve $H_{i-1}$.

## 7. 3D reconstruction and fine fitting

Once the entire curve $C_I$ has been approximated with projected helix curves, the next step is to compute the 3D coordinates of the vertices of $C_I$. We start at the first helix located at one endpoint of $C_I$ and continue the 3D reconstruction by processing adjacent helices until the opposite endpoint of $C_I$. The $z$-coordinates are computed using the equation $H_{3D,R}(t)$ and the helix coefficients previously computed.

Note that there exist two helices that match the same projected helix curve onto the plane ($z = 0$). These two helices are mirror symmetric with respect to the plane ($z = 0$). The equations of the 3D helix $H_{3D}(t)$ and its symmetric counterpart $H_{3D,S}(t)$ are:

$$H_{3D}(t) = \begin{pmatrix} H_{3D,x}(t) \\ H_{3D,y}(t) \\ H_{3D,z}(t) \end{pmatrix} = \begin{pmatrix} a(\cos(t) - 1) \\ bt\cos(c) - a\sin(c)\sin(t) \\ bt\sin(c) + a\cos(c)\sin(t) \end{pmatrix}$$

$$H_{3D,S}(t) = \begin{pmatrix} H_{3D,x}(t) & H_{3D,y}(t) & -H_{3D,z}(t) \end{pmatrix}^{\mathrm{T}}.$$

These two helices have the same coordinates except the $z$-coordinate which has opposite sign.

When processing the next helix $H_{3D,i+1}$, the translation along the $z$-axis is calculated such that the first endpoint of $H_{3Di+1}$ has the same $z$-coordinate as the last endpoint of $H_{3D,i}$. This is to ensure that the endpoints of neighboring helices have the same location. In addition, we choose the helix curve $H_{3D}(t)$ or its symmetric counterpart $H_{3D,S}(t)$ such that the tangent vectors of $H_{3D,i}$ and $H_{3D,i+1}$ at the junction between these two helices are the most parallel.

As one may see, the curvature of the generated piecewise helices is discontinuous at the points located at the junction between adjacent helices. This is because the coefficients of the helices have been computed independently for each helix. To overcome this problem, we modify the coefficients of the helices in such a way to make the tangents of the helices at the junction vertices as equal as possible while keeping the distance between the projected piecewise helix and the polygonal curve $C_I$ small. We formulate this problem as an optimization problem for which the objective function measures (1) the tangent difference at the junction vertices of the piecewise helix and (2) the overall matching error which is the distance between the projected piecewise helix and the vertices of $C_I$.

Let $H_{3D,1}, H_{3D,2} \ldots H_{3D,i} \ldots H_{3D,n}$ be the set of $n$ helices that compose the piecewise helix. Let $H_1, H_2 \ldots H_i \ldots H_n$ be the set of $n$ projected helices that approximate the curve $C_I$, $H_i$ being the projection of $H_{3D,i}$. Let $a_i$, $b_i$ and $c_i$ be the coefficients of the projected helix $H_i$. Each projected helix $H_i$ fits to a portion of $C_I$. Let $V_i = \{v_{i,1}, v_{i,2} \ldots v_{i,j} \ldots v_{i,n_i}\}$ be the set of $n_i$ vertices located along the part of $C_I$ which is fitted with $H_i$. Each set of vertices $V_i$ is associated with a transformation matrix. The purpose of this transformation matrix is to align the vertices $V_i$ with the projected helix $H_i$ in the 2-dimensional space $(x, y)$ (see Section 6.1).

The first vertex of $H_i$ is common with the last vertex $v_{i-1,n_{i-1}}$ of $H_{i-1}$. This vertex is the so-called junction vertex. Similarly, the last vertex of $H_i$ is common with the first vertex $v_{i+1,1}$ of $H_{i+1}$. Let $\{t_{i,1}, t_{i,2} \ldots t_{i,n_i}\}$ be the $n_i$ parameters of the points along the projected helix $H_i$ such that $\left\| F_i \cdot v_{i,j} - H_i\left(t_{i,j}\right) \right\|$ is minimum for all $v_{i,j} \in V_i$.

The objective function to minimize the difference of the tangent vectors at the junction vertices is defined as follows:

$$E_T(a_1, b_1, c_1 \ldots a_n, b_n, c_n) = \sum_{i=2}^{n} \left\| \hat{T}_{H_{3D,i-1}}\left(t_{i-1,n_{i-1}}\right) - \hat{T}_{H_{3D,i}}\left(t_{i,1}\right) \right\|^2.$$

$\hat{T}_{H_{3D,i-1}}\left(t_{i-1,n_{i-1}}\right)$ and $\hat{T}_{H_{3D,i}}\left(t_{i,1}\right)$ are the tangent at the second endpoint of the helix $H_{3D,i-1}$ and the first endpoint of the helix $H_{3D,i}$ respectively. The objective function to compute the distance error between $C_I$ and the projected helices is:

$$E_D(a_1, b_1, c_1 \ldots a_n, b_n, c_n) = \sum_{i=1}^{n} \left( \sum_{j=1}^{n_i} \left\| F_i \cdot v_{i,j} - H_i\left(t_{i,j}\right) \right\|^2 \right).$$

$\left\| F_i \cdot v_{i,j} - H_i\left(t_{i,j}\right) \right\|$ is the distance between a vertex of $C_I$ and its closest point along the corresponding projected helix curve $H_i\left(t_{i,j}\right)$. The equation of the projected helix curve $H_i\left(t_{i,j}\right)$ is a function of $a_i$, $b_i$ and $c_i$.

The objective function is defined as follows:

$$E(a_1, b_1, c_1 \ldots a_n, b_n, c_n) = \alpha_T E_T(a_1, b_1, c_1 \ldots a_n, b_n, c_n) + (1 - \alpha_T) E_D(a_1, b_1, c_1 \ldots a_n, b_n, c_n).$$

The weight $\alpha_T$ is a value in the interval $]0, 1[$ and which is chosen by the user. If its value is close to 1, more importance is given to reducing the tangent discontinuity at the helix junctions. The objective function is not linear in the unknown variables $a_1, b_1, c_1 \ldots a_n, b_n$ and $c_n$. We use the simplex search method proposed by Lagarias et al. [14] for computing the unknown variables such that it is minimum.

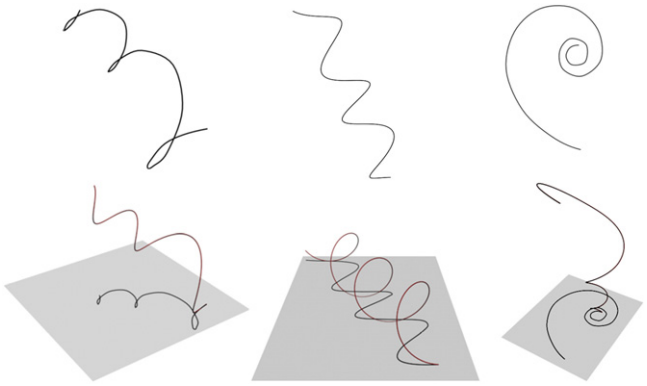

**Fig. 3.** Some examples of piecewise helices generated by our algorithm.

## 8. Results

We have implemented our algorithm as Matlab scripts and have tested our reconstruction method on different curves. The results are shown in Fig. 3. The computation time is about 1 min for curves composed of 500 vertices. Most of the computation time is spent for the fine fitting (Section 7).

The result of our algorithm is invariant up to translation, rotation and uniform scale; that is, the reconstruction of two 2D curves which are the same up to a rigid transformation and uniform scale will give the same 3D piecewise helix curve. This is because a fundamental step of our algorithm is to align the projected helix curves with the input curve $C_I$ using the tangent vectors (Section 6.1).

The piecewise helix curves generated with our algorithm are not $G^0$ continuous. In practice, the distance between the endpoints of two neighboring helices is small enough so that they can be regarded as having the same position.

Our algorithm also does not produce a $G^1$ continuous curve although it tries to find the piecewise helix such that the tangent discontinuity is the smallest possible at the junction vertices. Most of the time, these tangent discontinuities are small enough so that they are not noticeable. There are some cases where high tangent discontinuity occurs in the reconstructed piecewise helix curve (Fig. 4).

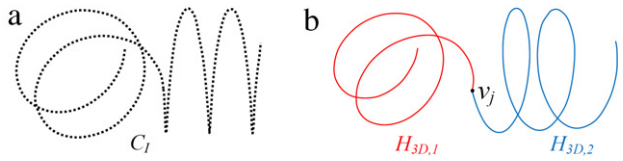


**Fig. 4.** A case where the reconstructed piecewise helix exhibits tangent discontinuity: the input polygonal curve $C_I$ is shown in (a); $v_j$ is the junction vertex. The tangents of $H_{3D,1}$ and $H_{3D,2}$ at the junction vertex have different orientation.

## 9. Conclusion

In this paper, we have described a method for computing a 3D piecewise helix curve from a 2D polygonal curve.

## Appendix

**Proposition 1.** *The value of $\kappa_H(t)$ is maximum when $t$ is a multiple of $\pi$ (i.e. $t = \pi \cdot d$ with $d \in \mathbb{N}$).*

**Proof.** To prove Proposition 1, we analyze the curvature of the ellipse $O$ which is the projection of the osculating circle $O_{3D}$ to the curve $H_{3D}(t)$. In order to determine the orientation of the osculating circle, we compute the Frenet frame along the curve $H_{3D}(t)$.

Let $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$ be the orthonormal coordinate system in which the helix curve $H_{3D}(t)$ is defined. Let $(\hat{T}_{H3D}(t), \hat{N}_{H3D}(t), \hat{B}_{H3D}(t))$ be the Frenet frame at the point $H_{3D}(t)$; this coordinate system is orthonormal and its origin is the point $H_{3D}(t)$. The parametric equation of the osculating circle $O_{3D}$ to the helix curve $H_{3D}(t)$ is defined in the Frenet frame as follows:

$$O_{3D}(u) = \left( \frac{(a^2+b^2)}{a} \sin(u) \quad \frac{(a^2+b^2)}{a} (1-\cos(u)) \quad 0 \right)^{\mathrm{T}}.$$

Let $\hat{T}_{H3D}(t)$, $\hat{N}_{H3D}(t)$ and $\hat{B}_{H3D}(t)$ be the tangent, normal and binormal vectors at the point respectively. To prove Proposition 1, we highlight the following five properties:

(P1) The osculating circle $O_{3D}$ is in the osculating plane $(\hat{T}_{H3D}(t), \hat{N}_{H3D}(t))$ of $H_{3D}(t)$ and its curvature is the same as that of $H_{3D}(t)$.

(P2) The matrix which defines the transformation from the Frenet frame $\left(\hat{T}_{H3D}(t), \hat{N}_{H3D}(t), \hat{B}_{H3D}(t)\right)$ to the global coordinate system $(\vec{X}, \vec{Y}, \vec{Z})$ is:
$$M_{F\to G}(t) = \begin{pmatrix} \hat{T}_{H3D}(t) & \hat{N}_{H3D}(t) & \hat{B}_{H3D}(t) \end{pmatrix}.$$

(P3) The matrix $M_{F\to S}(t)$ which defines the transformation from the Frenet frame $\left(\hat{T}_{H3D}(t), \hat{N}_{H3D}(t), \hat{B}_{H3D}(t)\right)$ to the orthogonal projection onto the plane $(z = 0)$ is the same as $M_{F\to G}(t)$ except that the last row is filled with zeros.

(P4) The projection of the osculating circle $O_{3D}$ onto the plane $(z = 0)$ is the ellipse $O$. The length of the major axis of $O$ is constant; it is equal to the length of the osculating circle diameter. The length of the minor axis is a function of the orientation of the osculating plane with respect to the plane $(z = 0)$. The parametric equation of the ellipse $O$ is $O(u,t) = M_{F\to S}(t)\, O_{3D}(u)$.

(P5) The curvature of the ellipse $O(u,t)$ at $u = 0$ is equal to the curvature of the projected helix curve $H(t)$. In fact, by setting the value $u$ to 0 in the equation to compute the curvature of the ellipse $O(u,t)$, we obtain the curvature expression of the projected helix $H(t)$.

To prove Proposition 1, we make the following statements. First, the orthogonal projection of the point $O_{3D}(0)$ located on the osculating circle $O_{3D}(u)$ is the point $O(0,t)$ which is located on the ellipse $O(u,t)$, $O(u,t)$ being the orthogonal projection of $O_{3D}(u)$ such that $O(u,t) = M_{F\to S}(t)\, O_{3D}(u)$.

Second, the point $O(0,t)$ is located at one of the endpoints of the major axis of the ellipse $O(u,t)$ if the vector $M_{F\to S}(t) \cdot \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^{\mathrm{T}}$ is a unit vector. This is true when $t = 0$ or $t = \pi$. Since the curvature of the ellipse is maximum at the endpoints of the major axis and the length of the major axis is constant, the curvature at $O(0,t)$ is maximum for $t = 0$ or $t = \pi$. Property (P5) states that the curvature of the ellipse $O(u,t)$ at $u = 0$ is equal to the curvature of the projected helix $H(t)$; we conclude that the curvature of $H(t)$ is maximal for $t = 0$ or $t = \pi$.

## References

[1] Brown E, Wang P. 3D object recovery from 2D images: a new approach. In: SPIE proc. robotics and computer vision, vol. 2904. 1996. p. 138–45.
[2] Shoji K, Kato K, Toyama F. 3-D interpretation of single line drawings based on entropy minimization principle. In: Proc. IEEE conf. computer vision and pattern recognition, vol. 2. 2001. p. 90–5.
[3] Cordier Frederic, Seo Hyewon. Free-form sketching of self-occluding objects. IEEE Computer Graphics and Applications 2007;27(1):50–9.
[4] Cordier Frederic, Seo Hyewon, Park Jinho, Noh Jun-yong. Sketching of mirror-symmetric shapes. IEEE Transactions on Visualization and Computer Graphics 2011;17(11):1650–62.
[5] Cohen J, Markosian L, Zeleznik R, Hughes J, Barzel R. An interface for sketching 3D curves. In: ACM I3DG 1999 symposium on interactive 3D graphics. 1999. p. 17–21.
[6] Cordier Frederic, Seo Hyewon, Melkemi Mahmoud, Sapidis NickolasS. Inferring mirror symmetric 3D shapes from sketches. Computer-Aided Design 2013;45(2):301–11.
[7] Piuze E, Kry PG, Siddiqi K. Generalized helicoids for modeling hair geometry. Computer Graphics Forum 2011;30(2):247–56.
[8] Goriely A, Neukirch S, Hausrath A. Polyhelices through *n* points. International Journal of Bioinformatics Research and Applications 2009;5(2):118–32.
[9] Ghosh S. Geometric approximation of curves and singularities of secant maps. A differential geometric approach. Ph.D. dissertation. University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science, Advisors: G. Vegter and J. Rieger. December 2010. p. 185.
[10] Derouet-Jourdan Alexandre, Bertails-Descoubes Florence, Thollot Joëlle. Floating tangents for approximating spatial curves with piecewise helices. Computer Aided Geometric Design 2013;30(5):490–520.
[11] Wither Jamie, Bertails Florence, Cani Marie-Paule. Realistic hair from a sketch. In: Shape modeling international 2007. p. 33–42.
[12] Marchal Xavier, Bertails Florence, Hétroy Franck. Reconstruction de trochodes à partir de courbes 2D. Technical report. 2009.
[13] Lewiner T, Gomes J, Lopes H, Craizer M. Curvature and torsion estimators based on parametric curve fitting. Computers & Graphics 2005;29(5):641–55.
[14] Lagarias JC, Reeds JA, Wright MH, Wright PE. Convergence properties of the Nelder–Mead simplex method in low dimensions. SIAM Journal on Optimization 1998;9(1):112–47.