Hough Transform for Detecting Space Curves in Digital 3D Models

Chiara Romanengo¹ · Silvia Biasotti¹ · Bianca Falcidieno¹

Received: 4 June 2021 / Accepted: 24 December 2021 / Published online: 17 January 2022
© The Author(s) 2022

Abstract

We present and analyse the Hough transform (HT) to recognise and approximate space curves in digital models, a problem that is not currently addressed by the standard HT. Our method works on meshes and point clouds and applies to models even incomplete or affected by noise, thus being suitable for the analysis of digital models deriving from 3D scans. In our approach we take advantage of a recent HT formulation for algebraic curves to define both parametric and implicit space curve representations. We also provide a comparative analysis of the HT-based method when dealing with both curve representations, discussing the computational performance and the approximation accuracy of both strategies.

Keywords Hough transform · Space curves · Implicit and parametric curves · Algebraic geometry

Mathematics Subject Classification 68U05 · 14H50 · 65S05

1 Introduction

Feature curves are a key element in the analysis and synthesis of digital shapes. They are significant for shape interpretation and perception studies support these curves as an effective choice for representing the salient parts of a 3D model [5]. Consider, for example, the objects depicted in Fig. 1, we naturally focus on the helix curves characterising the horn profile or the shell. However, when 3D models are acquired by scanning real objects, the resulting geometry does not explicitly encode these curves, especially when it is affected by noise, due to measurement uncertainty and sampling resolution, or miss some parts, due to the occlusion during the acquisition or other factors. In particular, in applications like the digitisation of archaeological artefacts, these objects might be damaged, and then curves are partially missing. This makes it difficult to extract feature curves in their entirety from the digital model, typically a mesh, or directly from the point cloud, and to approximate them in the model, Fig. 1b, d.

In this paper we address the problem of recognising space curves and providing a mathematical representation of them. For this purpose we make use of the Hough transform, which is well known for recognising curves in the plane and surfaces in space, but has not yet been sufficiently explored for space curves. In order to apply the Hough transform to space curves we need some operational assumptions: if the curves are expressed in a parametric form, we assume that the family of curves in which to search for the solution possesses a regular parameterization, while for curves in implicit form we assume that the equations that define these curves are analytical, more details are given in Sect. 4.2.

Our HT-based recognition method recognises a large number of space curves, including both those that directly have a space representation, such as the helix of Fig. 2, and those that can be represented with a projection onto quadrics or planes. This method deals with curves expressed in both implicit and parametric form; it evaluates the quality of the curve approximation, for both implicit and parametric representations, and it is robust to noise and outliers. In particular, for recognising curves expressed in implicit form, we extend the approach described in [27], which required the intermediate step of projecting the curves on a plane, while for parametric curve representations, we generalise to space curves the method for plane curves proposed in [15]. Since the HT formulation for parametric curves has to be defined family by family, we also propose an automatic procedure able to overcome this

✉ Silvia Biasotti
silvia@ge.imati.cnr.it
Chiara Romanengo
chiara.romanengo@ge.imati.cnr.it
Bianca Falcidieno
bianca@ge.imati.cnr.it

¹ Genova, Italy

Fig. 1 3D models of a horn and a shell (a, c). In **b** the two helices that delimit the horn profile and in **d** the helix characterising the shell

step of the algorithm for all the families of space curves with linear coefficients (notice that the curve parameter can be nonlinear). Our solution generalises the method proposed in [20] where only families of space curves with three linear parameters were considered.

Since our technique for identifying space curves on 3D shapes works for both parametric and implicit curve representations, pros and cons of both forms are also discussed, highlighting when to select one instead of the other or when the two curves formulations are not inter-changeable. In particular, the quality of the curve approximation through the HT-based recognition for the two forms of representation is compared, evaluating their advantages and drawbacks.

The remainder of the paper is organised as follows. Section 2 overviews previous work in feature curve recognition focusing on HT-based recognition methods. Section 3 introduces terminologies and basic concepts of space curves and the Hough transform for curves. Section 4 details the HT-based recognition procedure and how it may tackle space curves in implicit and parametric form. Section 5 shows the results of the method applied to digital models of 3D objects. It provides also a comparative analysis of the HT-based recognition algorithm for curves represented either in parametric or implicit form with respect to both a number of measures of quality of the curve fitting and the computational time. Concluding remarks end the paper.

2 Previous Work

The literature on HT-based recognition witnesses its large success in pattern recognition, and it is too vast to be summarised in a few lines; here we refer to [17] for a survey and we just mention some of the works we consider fundamental in its development: the original definition in [12] for straight line detection, the extension to circles and ellipses in [8], the identification in images of profiles without an analytical representation in 2D [2]. More recently, the HT was generalised

in [4] to classes of (plane) curves whose algebraic forms were known and to low-degree piece-wise polynomial curves, see [6]. The algorithms proposed to address these curves slightly differ if the curves were represented in parametric [15] or implicit form [26].

When dealing with 3D data, the HT was generally referred to the recognition of surfaces, either planes, see for instance [13] and its recent evolution [1], or simple primitives, such as ellipsoids [3], because a single, implicit equation in the Euclidean 3D space represents a surface. In the case of lines in space, the common strategy was to limit the search to straight lines, as in [18], or to project the data on planes. For instance, the approach in [27] implemented the theory described in [4] by projecting the space curves on a fitting plane. In this case, the HT was evaluated for a planar approximation of the curve and the outcome was then re-projected on the object surface. The method showed its effectiveness for the recognition of features in archaeological artefacts, in particular eyes and mouths in votive statues, and it was extended in [21] to more complex decorations, such as floral bands and Greek frets. Projecting on planes, the curves in [21,27] were represented by a single, implicit equation.

A first attempt to recognise space curves was made in a recent work [20]. It used HT and parametric curve representations. Its goal was to detect and insert spatial feature curves as constraints in a mesh by means of poly-curves composed of curved segments. However, this method also suffered some limitations because it could not automatically solve the system behind the HT expression only for families of space curves with three linear parameters, that is, the number of parameters was required to be equal to the number of equations.

Beyond the context of the HT, space curve fitting is still an open problem. In fact, projections on planes of particular families of curves were often used in this context as well. For instance, we mention the piece-wise splines of low-degree polynomials like the spline approximation in [7]; however, this approach needed a local, planar curve interpolation; therefore, it was not able to recognise entire curves

and to complete missing parts. Some methods fitted the data with some specific family of curves, such as Samper et al. [22], for studying the best curve fitting the Gaudí architecture among three conical and three hyperbolic-cosine curve types; Harary and Tal [11], for fitting line drawings and silhouettes with *3D Euler spirals*; and Lv et al. [14] with nose curves defined as geodesic curves crossing the nose bridge on triangle meshes. However, these methods defined only specific case-driven solutions and algorithms for setting the curve parameters.

At the best of our knowledge, also methods based on a learning process are limited to planar curves or curves that are projected on a plane and to curves represented as groups of poly-lines. For instance, approaches based on co-occurrence analysis relied on an interactive learning phase or assumed there are repetitions for every type of curve to be identified and sketched [10]. Due to the characteristics of the detected curves, this kind of methods was adopted mainly for recognising parts of buildings (such as windows, doors, etc.) and features in architectural models that are similar to strokes.

3 Preliminary Concepts

We overview the terminology, definitions and concepts on space curves (Sect. 3.1) and Hough transform (Sect. 3.2) that we use throughout the paper.

3.1 Parametric and Implicit Representations of Space Curves

In general, a curve is defined through a continuous function $\gamma : I \rightarrow X$ from an interval I of the real numbers into a topological space X . In case X is of three dimensions, such as the Euclidean space $\mathbb{R}^3$, the curve is called *space curve*, while if X is a plane, it is called *plane curve*. Space curves that do not lie on a plane are called *skew curves*.

In this work we consider curves that can be expressed with parametric and implicit equations, with reference to a Cartesian, a cylindrical or a spherical system of coordinates. Varying $t \in I$, the equations in parametric form $(x, y, z) = (x(t), y(t), z(t))$ describes the points of the curve. To be analytically represented with implicit equations in a three-dimensional system of Cartesian coordinates, curves must be expressed as the intersection of at least two surfaces. In general, a given curve can be realised by intersection in many ways, corresponding to different equivalent systems of equations representing the same curve.

An example of a skew curve is the cylindrical helix shown in Fig. 2. A parametric form and an implicit representation in

Fig. 2 A cylindrical helix with $a = 3$ and $b = 0.5$

cylindrical coordinates of this type of curve are, respectively:

$$\gamma(t) = \begin{cases} x = a \cos t \\ y = a \sin t \\ z = bt \end{cases} \quad \gamma(\rho, \theta, z) : \begin{cases} \rho - a = 0 \\ z - b\theta = 0 \end{cases} \quad (1)$$

The two curve representations are not interchangeable, and the transition from implicit to parametric equations and vice versa is generally not simple and often impossible to solve, at least with elementary procedures. Depending on the task, a parametric representation is best suited for generating points along a curve, whereas an implicit representation is most convenient for determining whether a given point lies on a specific curve [23]. These facts motivate the definition of a recognition method that can handle both implicit and parametric curve representations.

3.2 Hough Transform for Curves

It is well known that the original HT definition is based on the *point-line duality* as follows: points on a straight line, defined by an equation, correspond to lines in the parameter space that intersect in a single point. This point uniquely identifies the coefficients in the equation of the original straight line.

The duality concept extends to curves, both in implicit [25] and parametric form [4]. Given a family $\mathcal{F} = \{\mathcal{C}_{\mathbf{a}}\}$ of curves that depend on a set of parameters $\mathbf{a} = (a_1, \dots, a_n) \in U' \subset \mathbb{R}^n$, U' an open set of $\mathbb{R}^n$, a general point P in the space corresponds to a locus, Γ_P , in the parameter space U' . As P varies on a given curve $\mathcal{C}_{\mathbf{a}}$ from $\mathcal{F}$, a set of curves Γ_P is generated. If the set of curves Γ_P meets in one and only one point $\bar{\mathbf{a}} \in U'$, the family of curves $\mathcal{F}$ verifies the so-called regularity condition and the intersection point defines the parameters of the best fitting curve $\mathcal{C}_{\bar{\mathbf{a}}}$. The duality concept is fundamental for the HT-based recognition algorithm. Indeed, the HT translates the curve recognition problem into detecting which value of the parameters that determine the family of curves $\mathcal{F}$ corresponds to the curve best fitting a given set of points (such a value may be nonunique).

4 HT-Based Recognition Procedure for Space Curves

In this section we describe how the HT-based recognition procedure works for space curves expressed either in implicit or parametric form. Our method recognizes space curves starting from clusters of points identified in digital models (meshes or point clouds). These clusters can be supplied directly or extracted with a preprocessing step. Therefore the recognition procedure consists of a step for the extraction of clusters of points (Sect. 4.1), to identify sets of points to be potentially fit with known curves (called feature points); the actual recognition algorithm (Sect. 4.2) and a recognition criterion, which decides whether the recognised curve approximates the cluster of points with an error less than a given threshold ε (Sect. 4.3). The computational complexity is described in Sect. 4.2.2. To judge the quality of the HT approximation, in Sect. 4.4, we evaluate the curve fitting measure we adopted comparing it with other criteria.

4.1 Extraction of Feature Points Clusters

Unless point clusters are provided directly, a 3D model (a mesh or a point cloud) is pre-processed to identify sets of feature points. These sets are therefore candidates to fit known curves with the HT-based recognition method, analogously to [21, 27]. In our implementation, we have chosen the mean curvature as the filter criterion: for triangle meshes we apply the algorithm in Toolbox Graph package;¹ for point clouds, we utilize the polynomial fitting of osculating jets provided by MeshLab.² In the range of variation of the values assumed by the mean curvature, feature points correspond to points whose value falls within the first $m\%$ or the last $M\%$ of the values. The thresholds m and M influence the number of feature points; in our implementation, usual values are 15% and 85%, respectively. Then, to detect the possible curve elements, the points are into clusters clustered by using the DBSCAN algorithm [9]. The DBSCAN aggregates points that are close by marking isolated points in low-density regions as outliers. The DBSCAN algorithm requires two parameters: the threshold used as the radius of the density region (a real positive number), and the minimum number of points necessary to form a region (a positive integer). In our implementation, we use the DBSCAN routine provided in MATLAB.³ The outcome of this pre-processing phase is a set of num clusters CP_i of feature points, $i = 1, \dots, num$. As an example, a 3D model of a screw is shown in Fig. 3a,

together with some clusters of feature points selected using a given mean curvature value (Fig. 3b).

We highlight that our recognition method does not depend on a specific technique of feature point extraction, nor on the particular digital model, but it deals with the recognition of curves outlined by feature points. We refer to [5] for an overview on methods for extracting feature points and to the SHREC benchmark for an evaluation of methods for feature curves estimation [16].

4.2 HT-Based Recognition Algorithm for Space Curves

The HT-based recognition procedure looks for a family of curves $\mathcal{F}$ within the dictionary that best fits each cluster of points, expressed in parametric or implicit form. In several application domains, such as cultural heritage and CAD, the user often knows which curves to look for; then, the family of curves can be given as input. Otherwise, we try the entire catalogue and a quality criterion returns only the curve of the family that best fits. Some working hypotheses are assumed by our algorithm. In case a curve C_a is in parametric form, we assume that the system defining C_a with respect to the Cartesian coordinates x , y , and z can be analytically solved with respect to the parameter variables $\mathbf{a} = (a_1, \dots, a_n)$. In case of curves in implicit form, we assume, as in [25], that the functions F and G in the definition of C_a and Γ_P are analytical with respect to the Cartesian coordinates x , y , z and the parameter variables $\mathbf{a}$, $\mathbf{a} = (a_1, \dots, a_n)$, $n \geq 2$.

Once a cluster of points CP_i is selected, the method works in 4 main steps:

- *Initialization of the parameter space and the accumulator function.* A region $\mathcal{T}$ of the parameter space is determined by studying the relations between the main geometric characteristics of the family $\mathcal{F}$ and the lengths of the minimum bounding box of CP_i . An array $\delta_i = (\delta_{i,1}, \dots, \delta_{i,n})$ of integers is given in input to define the size of initial discretization of $\mathcal{T}$. Then, for each component $j = 1, \dots, n$, the space $\mathcal{T} = \prod_{j=1}^n [(1 - \bar{p})a_j^*, (1 + \bar{p})a_j^*]$ is divided into intervals of length $d_j = \frac{2\bar{p}}{\delta_{i,j}} a_j^*$, where $\bar{p}$ is a percentage which is set differently depending on the family considered (a typical value is 0.1). More in detail, considering an estimate of the parameters $\mathbf{a}^* = (a_1^*, \dots, a_n^*)$ determined on the basis of the CP_i size, we define the set of samples of samples of $\mathcal{T}$ as

$$a_{j,s_j} = (1 - \bar{p})a_j^* + s_j d_j, \\ s_j = 0, \dots, \delta_{i,j} - 1, \quad j = 1, \dots, n.$$

We will discuss a strategy for its refinement in Sect. 4.3. Figure 4a shows the point cluster CP_1 of Fig. 3b and its

¹ <http://www.ceremade.dauphine.fr/~peyre/matlab/graph/content.html>.

² <https://www.meshlab.net/>.

³ The Mathworks, Inc., Natick, Massachusetts, MATLAB version 9.7.0.1247435 Update 2 (R2019b) (2019) and discard isolated points.

Fig. 3 In **a** a 3D model of a screw; in **b** the clusters of points which correspond to a mean curvature value less than 0.1. The column on the right lists some of the clusters found, identified by a number and a colour (black for cluster #1)

bounding box. Then, the cells of $\mathcal{T}$ are uniquely identified by the coordinates of their centre. At the same time, an accumulator function $\mathcal{H}_i$ of dimension $\prod_{j=1}^n \delta_{i,j}$ is initialised; note that each entry of $\mathcal{H}_i$ corresponds to a cell of $\mathcal{T}$.

- *Estimation of the accumulator function.* An entry of the accumulator function $\mathcal{H}_i$ is increased by 1 each time the HT of a point (Γ_P) intersects the corresponding cell. The resulting accumulator function for the cluster CP_1 is shown in Fig. 4b. Since this step differs for curves in implicit or parametric form, more details are given in Sect. 4.2.1.
- *Identification of a potential fitting curve.* The maximum value reached by $\mathcal{H}_i$ is found. The coordinates of the cell centre $\bar{\mathbf{a}}$ corresponding to this value are the parameters of a potential fitting curve $\mathcal{C}_{\bar{\mathbf{a}}}$ for the set of points CP_i . Figure 4c provides the fitting curve for the cluster CP_1 . Note that if more local maxima are obtained in different cells, they essentially correspond to different curves of the family.
- *Evaluation of the approximation accuracy.* To automatically determine if a curve $\mathcal{C}_{\bar{\mathbf{a}}}$ satisfactorily approximates the cluster of points CP_i , we define the *Mean Fitting Error* (MFE) as:

$$E(CP_i, \mathcal{C}_{\bar{\mathbf{a}}}) := \frac{1}{|CP_i|} \sum_{P \in CP_i} d(P, \mathcal{C}_{\bar{\mathbf{a}}})/l_i, \quad (2)$$

where d is the Euclidean distance and l_i is the diagonal of the minimum bounding box containing CP_i . Note that the MFE provides an estimation of the deviation between $\mathcal{C}_{\bar{\mathbf{a}}}$ and CP_i and it is a size-independent percentage. The smaller it is, the better the curve approximation. If $\mathcal{H}_i$

Fig. 4 In **a** the cluster CP_1 (corresponding to Cluster #1 in Fig. 3b) represented with the minimum bounding box (in red) that contains it; in **b** the accumulator function $\mathcal{H}_i$; in **c** the recognised curve, corresponding to the maximum value of $\mathcal{H}_i$

presents two or more local maxima, this distance has to be calculated for each potential curve solution. The curve having lowest error is kept.

In Algorithm 1 we provide the pseudocode of the HT based recognition method common to both parametric and implicit forms. The methods for the two forms differ in the evaluation of the intersection between the cells and the Hough transforms of the points, see Sect. 4.2.1.

4.2.1 Estimation of the Accumulator Function for Space Curves in Implicit and Parametric Form

The recognition method adopts different strategies depending on whether the space curves are in implicit or parametric form, and this difference is in the evaluation of the accumulation function.

Algorithm 1 HT based recognition

```

1: Input: A set of feature points  $CP_i$ , a family of curves  $\mathcal{F}$ , an array
    $\delta_i \in \mathbb{N}^n$ 
2: Output: The parameters of the curves in  $\mathcal{F}$  that best fit the feature
   points

3:  $T = \text{InitSpaceParameters}(CP_i, \delta_i)$  /*discretization of
   the interval range of the family parameters
   according to  $\delta_i$ */
4:  $\mathcal{H}_i^0 = \text{InitAccumulatorMatrix}(T)$  /*zero value
   everywhere*/
5:  $\mathcal{H}_i = \text{EstimateAccumulatorFunction}(\mathcal{H}_i^0, \mathcal{F}, \Gamma_P)$  /*this
   function differs if the curve is in implicit
   or parametric form*/
6:  $\bar{\mathbf{a}} = \arg \max \mathcal{H}_i$  and consider the curve  $C_{\bar{\mathbf{a}}}$  in  $\mathcal{F}$ 
7:  $MFE = MFE(C_{\bar{\mathbf{a}}}, CP_i)$  /*Estimation of the fitting
   quality*/
8: return  $(\bar{\mathbf{a}}, MFE)$ 

```

Parametric Form

For curves in parametric form, the generic analytical expression of Γ_P needs to be derived directly from the parametric curve equations of the family [15]. The same procedure applies to our extension to space curves; we show an example of its calculation in Sect. 5.1. In this section, we present an automatic solution when the system of equations defining the family can be analytically solved with respect to the parameters $\mathbf{a}$. This solution extends the method in [20], in which only systems with a number of parameters equal to the number of equations were considered. In particular, we propose to automatically derive Γ_P by exploiting the Moore–Penrose pseudo-inverse [19] of the matrix $M(t)$ that defines the coefficients of $\mathbf{a}$. Then, if the family of curves in parametric form admits a matrix representation as $(x(t), y(t), z(t)) = M(t)\mathbf{a}$, Γ_P can be calculated directly

$$\Gamma_P : \mathbf{A}(t) = M^\dagger(t)(x_P, y_P, z_P).$$

Note that the parameter t that appears in the parametric formulation of Γ_P is the same as in the parametric equation of the family of curves $\mathcal{F}$ and it is necessarily linear. The evaluation of the intersection is translated into an inequality between the components of the sample points of Γ_P and the coordinates of the cells endpoints for each component:

$$a_{j,s_j} \leq A_j(t) < a_{j,s_j+1}$$

$s_j = 0, \dots, \delta_{i,j} - 1$. This inequality must be verified for at least one parameter t for each component j of $\mathbf{A} = (A_1, \dots, A_n)$.

Implicit Form

For space curves expressed in implicit form, we consider that Γ_P can be evaluated through the zero loci of its implicit functions and Γ_P crosses the cell if its value on the point centre of the cell is zero. Then, we use the symbolic calculation taking advantage of the local approximation of the zero loci of a set of analytic functions in terms of Taylor series, following the strategy defined in [25] which compares the infinity norm of the evaluation of a function $f = (f_1, f_2)$ in a point with two bounds B_1 and B_2 that depend on the norms of the Jacobian and Hessian matrices of f and the Moore–Penrose pseudo-inverse of the Jacobian matrix. In our implementation, we use the system *CoCoA*⁴ for the symbolic manipulation of polynomials and matrices, since it gives an exact estimation of the zeros of a curve in implicit form on the cells of the parameter space. In our approach, the functions f_1 and f_2 coincide with the functions defining the system of equations of Γ_P , while the point of the evaluation is the centre C of the considered cell. In particular, if:

- $\|(f_1(C), f_2(C))\|_\infty > B_1$ then Γ_P does not cross the cell and the corresponding entry of $\mathcal{H}_i$ does not increase;
- $\|(f_1(C), f_2(C))\|_\infty < B_2$ then Γ_P crosses the cell and the corresponding entry of $\mathcal{H}_i$ increases by 1;
- $B_2 < \|(f_1(C), f_2(C))\|_\infty < B_1$ the corresponding entry of $\mathcal{H}_i$ increases by a value of indeterminacy ξ . By convention, this value of indeterminacy is fixed to 0.5.

In the zones of uncertainty, it is possible to choose whether to locally refine the sampling.

Example

Considering the family of cylindrical helices $\mathcal{F} = \{\gamma_{\mathbf{a}}\}$ represented by the equations in implicit and parametric form (1), here we show an example of how to get the respective HT equations. In the parametric case, in general we consider the matrix of the coefficients $M(t) = \begin{bmatrix} \cos t & \sin t & 0 \\ 0 & 0 & t \end{bmatrix}^T$ and we exploit the Moore–Penrose pseudo-inverse $M^\dagger(t)$ for curves with linear, parameter coefficients:

$$\Gamma_P : (A(t), B(t))^T = M^\dagger(t)(x_P, y_P, z_P)^T.$$

Regarding the implicit form, the equation of Γ_P is simply obtained by applying the transformation from Cartesian to cylindrical coordinates and by substituting the coordinates

⁴ J. Abbott, A. M. Bigatti, G. Lagorio, CoCoA-5: a system for doing Computations in Commutative Algebra, Available at <http://cocoa.dima.unige.it>.

of point $P = (\rho_P, \theta_P, z_P)$ for the variables ρ, θ , and z and considering the parameters A and B as new variables:

$$\Gamma_P : \begin{cases} \rho_P - A = 0 \\ z_P - B\theta_P = 0 \end{cases}.$$

The best fitting curves of the cluster CP_1 of Fig. 4 obtained with the method in parametric and implicit form are, respectively

$$\gamma(t) = \begin{cases} x = 36.40 \cos t \\ y = 36.40 \sin t \\ z = 3.57t \end{cases} \quad \gamma(\rho, \theta, z) : \begin{cases} \rho - 37.98 = 0 \\ z - 3.61\theta = 0 \end{cases} \quad (3)$$

4.2.2 Computational Complexity

The complexity of the HT-based recognition algorithm depends on the size of the discretization of the space of the parameters $\mathcal{T}$. As previously discussed, $\mathcal{T}$ is subdivided into $M = \prod_{j=1}^n \delta_{i,j}$ cells, where n is the number of parameters of the family $\mathcal{F}$ (in the curves proposed in this paper, $n = 1, 2, 3$). Since the accumulator function $\mathcal{H}_i$ has the same dimension of $\mathcal{T}$, the evaluation of $\mathcal{H}_i$ requires $O(M)$ operations for each point in a cluster CP_i .

Therefore, the computational complexity for each cluster is $O(ML)$, where L represents the number of elements of CP_i on which we evaluate the HT accumulator function.

In summary, the algorithm has the same theoretical complexity for curves in implicit and parametric form, but in the first case it resorts to use symbolic computations that, in practice, could become a computational bottleneck for densely sampled curves. Moreover, for parametric curves the HT implementation as well as being faster is simpler and more intuitive.

4.3 Goodness of the Approximation and Refinement Strategy

Given a set of points, a family of space curves and an array of integers, possibly different for each parameter, Algorithm 1 returns in output the best fitting curve $\mathcal{C}_{\bar{\mathbf{a}}} \in \mathcal{F}$ and the quality of the approximation MFE. The value of the MFE has to be compared with the fitting threshold ε (in our experiments, typical values of ε are 3–5%). In the case the value of MFE is smaller than ε the curve is a good solution, while if the MFE is greater than 2ε , the family $\mathcal{F}$ is excluded and the algorithm starts again by selecting another family of curves of our dictionary. Finally, if $\text{MFE} \in (\varepsilon, 2\varepsilon)$ a new denser sampling of the parameter space is considered, unless the values $\delta'_{i,j}$ exceed a value C that represents the highest possible refinement for each parameter (in our experiments, we

Algorithm 2 Refinement Algorithm

```

1: Input: A set of feature points  $CP_i$ , a fitting threshold  $\varepsilon$ , a family of
   curves  $\mathcal{F}$ 
2: Output: The parameters  $\bar{\mathbf{a}}$  of the curve in  $\mathcal{F}$  that best fits  $CP_i$ 
3:  $\bar{\mathbf{a}} \leftarrow \emptyset$ ,  $\text{MFE} \leftarrow \infty$ ,
4:  $\delta_i := (\delta_{i,j})_{j=1,\dots,n}$ ,  $\delta_{i,j} \leftarrow 16 \forall j$ 
5: while ( $\bar{\mathbf{a}} = \emptyset$  and  $\text{MFE} \geq \varepsilon$  and  $\delta_{i,j} \leq C \forall j$ ) do /*  $C$  is the
   maximum acceptable size */
6:   ( $\bar{\mathbf{a}}, \text{MFE}$ ) = HTBasedRecognition( $CP_i, \mathcal{F}, \delta_i$ ) /* Algorithm 1
   */
7:   if ( $\text{MFE} \leq \varepsilon$ ) then
8:     return  $\bar{\mathbf{a}}$ ,
9:   end if
10:  if ( $\varepsilon < \text{MFE} \leq 2 * \varepsilon$ ) then
11:    Resize( $\delta_i$ ),  $\bar{\mathbf{a}} \leftarrow \emptyset$ 
12:  end if
13:  if ( $\text{MFE} > 2 * \varepsilon$ ) then
14:    return  $\bar{\mathbf{a}} \leftarrow \emptyset$ 
15:  end if
16: end while

```

observed that a good choice for the value C is 256). In particular, for curves expressed in implicit form the refinement could adaptively split only the cells with uncertainty, while in the case of curves in parametric form, all cells are bisected (i.e. $\delta'_i = 2\delta_i$). The refinement process repeats until $\text{MFE} \in (\varepsilon, 2\varepsilon)$ and $\delta_{i,j}$ is smaller than C for all j . The refinement process successfully ends if a curve $\mathcal{C}_{\bar{\mathbf{a}}} \in \mathcal{F}$ with $\text{MFE} \leq \varepsilon$ is found. A new family of curves $\mathcal{F}'$ is selected if the values of the array δ_i become too large or $\text{MFE} > 2\varepsilon$.

In Algorithm 2 we provide the pseudocode of the refinement strategy.

4.4 Quality Measures

In Sect. 4.2 we have proposed the *Mean Fitting Error* as the criterion to evaluate the quality of the approximation obtained by our method. In addition, here we compare it with two other distances, namely the *direct Hausdorff distance* and the *Goodness of Fit* (GoF).

- *Direct Hausdorff distance.* The *direct Hausdorff distance* from the points $a \in A \subset \mathbb{R}^3$ to the points $b \in B \subset \mathbb{R}^3$ is defined as follows:

$$d_{d\text{Haus}}(A, B) = \max_{(a \in A)} \min_{(b \in B)} d(a, b),$$

with d the Euclidean distance. Note that it is not a metric, since it is not symmetric. In our tests, we consider the sampled curve $\mathcal{C}_{\bar{\mathbf{a}}}$ as set A and the cluster of points CP_i as set B .

- *Goodness of Fit.* We use the notion of *Goodness of Fit* (GoF) as defined in [15]. In the notation introduced in Sect. 4, let $\mathcal{T}'$ be a region subset of $\mathcal{T}$ consisting of the cells of the accumulator matrix whose value is greater than $p\%$ of the maximum value reached by $\mathcal{H}_i$. In our

experiments, we fix $p = 80$. For each cluster CP_i , we consider the subset S of the points P_l of CP_i such that $\Gamma_{P_l} \cap \mathcal{T}' \neq \emptyset$. Let D_l be the Euclidean distance of the point P_l from the curve $\mathcal{C}_{\bar{a}}$, given by $D_l = d(P_l, \mathcal{C}_{\bar{a}}) = \min_{P \in \mathcal{C}_{\bar{a}}} \|P_l - P\|_2$.

$$GoF := \frac{D_1 + D_2 + \dots + D_{\#S}}{\#S},$$

where $\#S$ is the cardinality of the set S .

Note that in previous works [20,21], only the GoF has been considered for the evaluation of the approximation accuracy. However, using MFE makes the algorithm more automatic, since it does not require any input parameter. In any case, in our experiments, the percentage values of the MFE are very similar to those of the GoF.

In order to make these measures comparable with the MFE, in our experiments, they are also weighted on the length of the diagonal of the bounding box of the points. The GoF distance provides a lower result, but the percentage value is very similar to that of the MFE, as shown in the experiments on digital models of real objects in Sect. 5.2.

5 Experimental Results and Analysis of the HT-Based Recognition Algorithm

In this section we show some examples of spatial curves recognised and approximated with our method. Furthermore, to evaluate the efficiency and accuracy of the approximations, we carry out a comparative analysis of both the quality of the resulting curves, by evaluating the MFE, the GoF, the direct Hausdorff distance measures and the execution time for the curves expressed in parametric and implicit form. In fact, the algorithm in these two cases makes use of different algebraic techniques, in the first, it uses an analytical solution, while in the second it employs symbolic calculation.

5.1 Examples

The examples of models selected here are mainly meshes, because the repositories we work on are mainly made up of these models. However, we have the original scans of some objects, such as the shell in Fig. 1c, and therefore we worked on them directly on the point cloud. The models shown here represent man-made objects, natural objects, industrial components, mainly stored in various repositories

(e.g. Visionair,⁵ Sketchfab⁶ and the STARC⁷ repository). For these models, we do not possess a representation in continuous smooth patches such as B-splines or other CAD representations; therefore, we do not have the exact, mathematical formulation of their space curves. In Fig. 5 we show the results obtained applying our method to eight 3D models. Once the feature points are aggregated in clusters, to each group we apply the curve recognition method described in Sect. 4. The two columns show, respectively, the set of feature points identified by the clustering step with the recognised curve and the points highlighted on the corresponding model. In particular, Fig. 5a–f presents some examples of recognition by means of families of classical space curves. The feature points in Fig. 5a, d, f are approximated with a family of helices of implicit and parametric equations:

$$\begin{aligned} I : & \begin{cases} b^2 x^2 \left(\frac{z}{c}\right)^{\frac{2m}{s}} + a^2 y^2 \left(\frac{z}{c}\right)^{\frac{2n}{s}} - a^2 b^2 \left(\frac{z}{c}\right)^{\frac{2m}{s}} \left(\frac{z}{c}\right)^{\frac{2n}{s}} = 0 \\ x - a \left(\frac{z}{c}\right)^{\frac{n}{s}} \cos \left(\left(\frac{z}{c}\right)^{\frac{1}{s}}\right) = 0 \end{cases} \\ P : & \begin{cases} x = at^n \cos t \\ y = bt^m \sin t \\ z = ct^s \end{cases} \end{aligned} \quad (4)$$

with different values of n , m , and s : $n = \frac{2}{5}$, $m = \frac{1}{5}$, $s = \frac{16}{25}$ for Fig. 5a, $n = m = \frac{1}{2}$, $s = \frac{3}{2}$ for Fig. 5d and $n = m = \frac{1}{2}$, $s = 1$ for Fig. 5f. The last one is the well-known parabolic helix.

The points selected on the model in Fig. 5b are recognised with a clelia curve. The implicit and parametric equations of this family are:

$$I : \begin{cases} \rho - a \cos n\theta = 0 \\ z - b \sin n\theta = 0 \end{cases}, \quad P : \begin{cases} x = a \cos nt \cos t \\ y = a \cos nt \sin t \\ z = b \sin nt \end{cases}$$

with $n = \frac{4}{5}$.

The points selected on the model in Fig. 5c are recognised with a pancake curve of implicit and parametric equations:

$$I : \begin{cases} x^2 + y^2 - a^2 = 0 \\ a^2 z - b(2x^2 - a^2) = 0 \end{cases}, \quad P : \begin{cases} x = a \cos t \\ y = a \sin t \\ z = b \cos 2t \end{cases}.$$

Finally, the points in Fig. 5e are approximated with a family of sphero-cylindrical curves, called hippopede of implicit

⁵ The Shape Repository, <http://visionair.ge.imati.cnr.it/ontologies/shapes/>.

⁶ Publish & find 3D models online, <https://sketchfab.com/>.

⁷ STARC repository, <http://public.cyi.ac.cy/starcRepo/>.

Fig. 5 Results obtained by our method applied to some 3D models

and parametric equations:

$$\begin{aligned}
 I : & \begin{cases} x^2 + y^2 + z^2 - a^2 = 0 \\ c^2(x - d)^2 + b^2y^2 - b^2c^2 = 0 \end{cases} \\
 P : & \begin{cases} x = d + b \cos t \\ y = c \sin t \\ z = \pm \sqrt{a^2 - (d + b \cos t)^2 - c^2 \sin^2 t} \end{cases}
 \end{aligned} \quad (5)$$

in the special case in which $d = a - b$.

In the case of curves that can be approximated by quadrics or planes, we can exploit the richness of the dictionary of plane curves [24], much ampler than the one of spatial curves. To do that, we propose a space curve representation that is modelled as the intersection of a paraboloid and a cylinder with a plane curve as its directrix. This type of representation also includes the degenerate case in which the paraboloid coincides with a plane. Figure 5g, h shows examples of points recognised with a family of curves obtained as projections of families of plane curves on particular types of surfaces. The points in Fig. 5g are recognised with the family of curves of

equations:

$$I : \begin{cases} bx^m + a^m y^m - a^m b = 0 \\ z - cx^2 - cy^2 = 0 \end{cases}$$

$$P : \begin{cases} x = at \\ y = \pm(b - bt^m)^{\frac{1}{m}} \\ z = ca^2 t^2 + c(b - bt^m)^{\frac{2}{m}} \end{cases}$$

obtained as the intersection of a paraboloid and a cylinder with a Lamet curve (see [24]) as its directrix. Figure 5h represents an example of the degenerate case in which the paraboloid coincides with a plane. In this case the family of curves used for recognition, called astroid (see [24]), is given by the equations:

$$I : \begin{cases} (x^2 + y^2 - a^2)^3 + 27a^2 x^2 y^2 = 0 \\ z - b = 0 \end{cases}$$

$$P : \begin{cases} x = a \cos^3 t \\ y = a \sin^3 t \\ z = b \end{cases}$$

The HT-based recognition technique is robust to noise and missing parts, as in the case of Fig. 5c, d, where, respectively, a portion of the decoration on the fragment of a vase bottom was eroded and some parts of the profile of the horn are missing. In this example, we show the recognition of pieces of a single curve; similarly, we are able to recognise the other elements of the relief of Fig. 5c and the second curve of the horn of Fig. 1b. An example of robustness to noise is given in Fig. 6, where the models of the objects presented in the first row have been perturbed by adding a Gaussian noise. More in details, on average, the points have a perturbation in norm L^2 of 5% (Fig. 6a) and 0.3% (Fig. 6b), with respect to the points of the original model. Also in these cases the method identifies the two space curves without any problem. The family of curves used for the points in Fig. 6a is a family of helices [Eq. (4)] with $n = m = \frac{3}{2}$ and $s = \frac{9}{4}$, while the one selected for Fig. 6b is a family of sphero-cylindrical curves [Eq. (5)] with $d = a$.

Note that all the curves used in the experiments of this section have a parametric representation, which is linear with respect to the parameters $\mathbf{a}$, with the exception of the sphero-cylindrical and hippopede ones, recognised in the examples in Figs. 6b and 5e, respectively. In these cases, the expression of Γ_P needs to be derived family by family of curves. As an example, we show the parametric expression of Γ_P that is obtained by solving the system of equations that corresponds

to the hippopede curve:

$$\Gamma_P : \begin{cases} A = \sqrt{x_P^2 + y_P^2 + z_P^2} \\ B = -\frac{x_P}{1 - \cos t} + \frac{\sqrt{x_P^2 + y_P^2 + z_P^2}}{1 - \cos t} \\ C = \frac{y_P}{\sin t} \end{cases}.$$

We have used this formulation of Γ_P in our experiments.

5.2 Comparative Analysis

Table 1 compares the HT based algorithm for the same families of mathematical space curves, when the data are sampled on a random perturbation error. More in detail, on average, the points have a perturbation of 2% with respect to the exact formulation.

The parameters of the mathematical curves and those recognised by the HT algorithm on samples of the curve perturbed with noise are listed in Table 1. In each column, we report the family of space curves, the parameters of the mathematical curve, the curve parameters that differ from the original ones as recognised by the HT algorithms, for the implicit and parametric curve expressions, respectively, on the perturbed curve samples. For a more accurate analysis we consider two different steps of sampling, a coarser one ($\frac{1}{100}$ of the curve length) and a finer one ($\frac{1}{500}$ of the curve length).

The experiments are performed on a laptop equipped with an Intel Core i7 processor (at 1.3 GHz). All the routines are implemented in MATLAB, with the exception of the symbolic calculations that are implemented on the open source toolkit, CoCoA.

Table 2 summarises the comparative analysis for our experiments with digital models of real objects. We list, from left to right, the 3D model; the MFE, the GoF, and the d_{dHaus} distances; the computational time in seconds. All these measures are reported for both the implicit and parametric curve expressions. The computational time refers to the estimation of the accumulator function step of the algorithm, i.e. the symbolic calculations or the inequalities estimations, because the other steps of the algorithm take on average 0.5 s per cluster. Finally, in the rightmost column, we indicate the number of points L in each cluster, i.e. the result of the pre-processing step. The number L depends on the sample density of the model and the size of the feature curve. The MFE measure and the GoF highlight that a better precision is achieved by the HT-based algorithm for approximating curves in implicit form with respect to the parametric ones. On the other hand, the algorithm for curves in parametric form is computationally more efficient. The computational time of the HT based recognition algorithm for curves in parametric form can be further optimised when a family of

Fig. 6 3D original models (first row) and the same models modified by adding a Gaussian noise (second row). Alongside, the recognised feature curves

Table 1 Comparison between the mathematical curve parameters and those recognised by our algorithm over perturbed curves, with two sampling steps: $\frac{1}{100}$ (sample 1) and $\frac{1}{500}$ (sample 2) of the curve length

Mathematical curve		Parameters	Perturbed sample 1		Perturbed sample 2	
Curve			Implicit	Parametric	Implicit	Parametric
(1)	Parabolic helix	a = 1	$a = 1.06$	$a = 0.98$	$a = 1.06$	$a = 0.98$
		b = 1	$b = 0.98$	$b = 0.95$	=	$b = 0.97$
		c = 2	=	=	=	=
(2)	Cylindrical helix	a = 1	$a = 1.06$	$a = 0.99$	=	$a = 0.98$
		b = 1	$b = 0.98$	$b = 0.95$	=	$b = 0.97$
(3)	Pancake curve	a = 2	$a = 1.98$	$a = 1.95$	=	$a = 1.98$
		b = 1	$b = 0.98$	$b = 1.05$	$b = 1.04$	=
(4)	Clelia, $n = 0.8$	a = 2	$a = 1.98$	$a = 1.96$	=	$a = 1.98$
		b = 2.5	=	=	=	=
(5)	Sphero-cylindrical curve	a = 2	=	=	=	=
		b = 1	$b = 1.06$	$b = 0.94$	=	=
	$d = a$	c = 0.5	$c = 0.50$	=	=	=
(6)	Hippopede	a = 2	=	=	=	=
	$d = a - b$	b = 1	=	$b = 1.02$	=	=
		c = 0.9	$c = 0.92$	=	=	=
(7)	Lamet curve	a = 2	=	=	=	=
	Projected	b = 20	$b = 19.91$	$b = 19.80$	$b = 19.98$	$b = 19.90$
	On a paraboloid	c = 1	$c = 0.97$	$c = 1.06$	=	=
(8)	Astroid curve projected	a = 2	=	=	=	=
		b = 1	$b = 1.06$	$b = 0.99$	=	=

The values in bold represent the target parameters. The symbol = indicates when the parameter identified equals the target one

Table 2 Quality measures and computational time for the HT-based recognition algorithm for the curves highlighted in Fig. 5, in implicit (I) and parametric (P) form

Model	MFE		GoF		$d_d\text{Haus}$		Time (s)		L
	I	P	I	P	I	P	I	P	
Figure 4c	0.0126	0.0183	0.0072	0.0125	0.0931	0.0870	1100	34	443
Figure 5a	0.0136	0.0164	0.0134	0.0139	0.0312	0.0351	201	4	96
Figure 5b	0.0177	0.0177	0.0112	0.0114	0.0236	0.0242	4012	178	2000
Figure 5c	0.0068	0.0097	0.0030	0.0034	0.0234	0.0248	376	13	922
Figure 5d	0.0082	0.0082	0.0032	0.0044	0.0248	0.0278	300	2	176
Figure 5e	0.0100	0.0112	0.0021	0.0031	0.0302	0.0259	2200	2	97
Figure 5f	0.0091	0.0120	0.0118	0.0156	0.0302	0.0312	4232	20	2027
Figure 5g	0.0038	0.0057	0.0044	0.0065	0.0402	0.0543	5200	25	585
Figure 5h	0.0107	0.0193	0.0020	0.0193	0.1112	0.0828	1.7	1	58

L represents the number of feature points

space curves can be expressed with three, linear, independent parameters by solving the linear system with the Cramer's rule as proposed in [20]. For example, in this case, the computational time decreases of 60% for a set of 400 feature points.

In summary, we can say that the implicit form guarantees a better approximation of the parameters, and therefore a greater precision, but a higher computational time due to the use of symbolic calculation. The parametric formulation instead makes the method much faster, especially if we can use Cramer's rule or Moore–Penrose pseudo-inverse.

Our experiments confirm that the GoF distance generally provides a good convergence rate. Indeed, such a distance has been tailored for contours in images that might be affected by some noise and outliers; indeed, it happens that some points of a cluster $C P_i$ do not really lie on the profile we are looking for [15], see for instance the cluster in Fig. 4c. For this reason, the GoF distance was designed to be evaluated only on the most representative characteristic points. However, this measure requires more input parameters as the percentage of the maximum to be considered and the minimum number of points for computing it, which makes the algorithm nonautomatic. On the other hand, the MFE yields an interpretation of the distances similar to the GoF and it does not require any input. On the contrary, the Hausdorff distance is designed to measure if two sets of points are globally close and depends on the extrema kept by the Euclidean distance. As confirmed by our tests, this distance is sensitive to outliers and data perturbation, while it well reflects if two dense curve samples approximate each other.

6 Concluding Remarks

In this paper we have introduced a general, HT-based algorithm for the recognition of space curves and their

approximation with known curves expressed in both implicit and parametric form. It works directly on clusters of points extracted from digital models (meshes or point clouds) and recognises feature curves, even incomplete or affected by noise. Being based on the HT concept, this method requires a priori knowledge of which curves to look for. This suggestion can be provided by the user through a template or by choosing in a curves catalogue the one or the ones most similar to what is sought.

Our approach directly addresses space curves as defined in Sect. 3.1. Since at the current state of the art the number of space curves is limited, we also introduce a constructive strategy to build a space curve as the intersection of a quadric surface and a cylinder with a plane curve as its directrix, taking advantage of the rich dictionary of plane curves.

In particular, this work extends the approach in [27] not only to detect families of space curves without necessarily projecting them on a plane, but it also improves the approximation of those that can be projected by adding the possibility of using surface primitives, such as paraboloids. Figure 7 compares the HT-based recognition of the feature curves extracted from the 3D models provided on the first row, with space curves projected on a paraboloid or a plane as in [27]. As shown in the examples in Fig. 7a, c, the use of a paraboloid better fits the set of feature points when the surface is curved, in particular in the star extremities and in the inner part of the eye. This result is also demonstrated by the value of the MFE: in (a) it is 0.0131, while in (b) it is 0.0180; in (c) it is 0.0116, while in (d) it is 0.0129.

The proposed method has several advantages: it combines the HT-based recognition of curves expressed in both implicit and parametric form in a single flow; it works indifferently on meshes or point clouds and, as it is based on the recognition of curve families, it is able to assess whether a feature curve is repeated. Furthermore, the use of the HT naturally leads to a curve recognition pipeline robust to noise, outliers, and

Fig. 7 Approximation of curves projected, respectively, on a paraboloid and on a plane: in **a**, **b** using a geometric petal curve; in **c**, **d** using a m -convexities curve (see [24])

missing data. Thanks to these characteristics, it is particularly suitable for the analysis of digital models deriving from 3D scans.

There are two limitations for the algorithm presented in Sect. 4. First, for curves in implicit form, when the choice of the parameter space and its sampling is too rough, the value of the accumulator function in some cells could be undecidable, and therefore, a space refinement is necessary. Second, for curves in parametric form, if the system of equations is not linear with respect to the parameters $\mathbf{a}$, there is no automatic way to calculate the expression of the HT and its formulation has to be calculated case by case, as for the sphero-cylindrical curves [Eq. (5)].

In the future, we plan to extend our method by optimising the sampling of the parameter space. A suggestion in this sense could come from the approach described in [3], which proposes a first example of adaptive sampling valid only for plane curves or surfaces in space and in which the parameters of the family of curves or surfaces are assumed to be linear.

Acknowledgements This work has been developed in the CNR research activities DIT.AD004.100, DIT.AD021.080.001. The authors thank Michela Spagnuolo and the colleagues at CNR-IMATI for the fruitful discussions and Maria Laura Torrente for her feedback on space curves in implicit form.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adap-

tation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Araújo, A.M.C., Oliveira, M.M.: A robust statistics approach for plane detection in unorganized point clouds. *Pattern Recognit.* **100**, 107–115 (2020). <https://doi.org/10.1016/j.patcog.2019.107115>
2. Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* **13**(2), 111–122 (1981)
3. Beltrametti, M., Sendra, J., Sendra, J., Torrente, M.: Moore–Penrose approach in the Hough transform framework. *Appl. Math. Comput.* **375**, 125083 (2020)
4. Beltrametti, M.C., Robbiano, L.: An algebraic approach to Hough transforms. *J. Algebra* **37**, 669–681 (2012)
5. Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H.S., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S.: Where do people draw lines? *ACM Trans. Graph.* **27**(3), 1–11 (2008)
6. Conti, C., Romani, L., Schenone, D.: Semi-automatic spline fitting of planar curvilinear profiles in digital images using the Hough transform. *Pattern Recognit.* **74**(C), 64–76 (2018)
7. Daniels, J., II., Ochotta, T., Ha, K.L., Silva, T.C.: Spline-based feature curves from point-sampled geometry. *Vis. Comput.* **24**(6), 449–462 (2008)
8. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* **15**(1), 11–15 (1972)
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise.

- In: 2nd International Conference Knowledge Discovery and Data Mining, pp. 226–231 (1996)
10. Gehre, A., Lim, I., Kobbelt, L.: Feature curve co-completion in noisy data. *Comput. Gr. Forum* **37**(2), 1–12 (2018)
 11. Harary, G., Tal, A.: 3D Euler spirals for 3D curve completion. *Comput. Geom.* **45**(3), 115–126 (2012)
 12. Hough, P.V.C.: Method and means for recognizing complex patterns (1962). US Patent 3,069,654
 13. Limberger, F.A., Oliveira, M.M.: Real-time detection of planar regions in unorganized point clouds. *Pattern Recognit.* **48**(6), 2043–2053 (2015)
 14. Lv, C., Wu, Z., Wang, X., Zhou, M., Toh, K.A.: Nasal similarity measure of 3D faces based on curve shape space. *Pattern Recognit.* **88**, 458–469 (2019). <https://doi.org/10.1016/j.patcog.2018.12.006>
 15. Massone, A.M., Perasso, A., Campi, C., Beltrametti, M.C.: Profile detection in medical and astronomical images by means of the Hough transform of special classes of curves. *J. Math. Imaging Vis.* **51**(2), 296–310 (2015)
 16. Moscoso Thompson, E., Gerasimos, A., Moustakas, K., Nguyen, E.R., Tran, M., Lejemble, T., Barthe, L., Mellado, N., Romanengo, C., Biasotti, S., Falcidieno, B.: SHREC'19 track: feature curve extraction on triangle meshes. In: EG Work. 3DOR, pp. 85–92. The EG Association (2019)
 17. Mukhopadhyay, P., Chaudhuri, B.B.: A survey of Hough transform. *Pattern Recognit.* **48**(3), 993–1010 (2015)
 18. Oesau, S., Lafarge, F., Alliez, P.: Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS J. Photogramm. Remote Sens.* **90**, 68–82 (2014)
 19. Penrose, R.: On best approximate solutions of linear matrix equations. *Math. Proc. Camb. Phil. Soc.* **52**(1), 17–19 (1956)
 20. Romanengo, C., Biasotti, S., Falcidieno, B.: HT-based identification of 3D feature curves and their insertion into 3D meshes. *Comput. Gr.* **89**, 105–116 (2020)
 21. Romanengo, C., Biasotti, S., Falcidieno, B.: Recognising decorations in archaeological finds through the analysis of characteristic curves on 3D models. *Pattern Recognit. Lett.* **131**, 405–412 (2020)
 22. Samper, A., González, G., Herrera, B.: Determination of the geometric shape which best fits an architectural arch within each of the conical curve types and hyperbolic-cosine curve types: The case of Palau Güell by Antoni Gaudí. *J. Cult. Herit.* **25**, 56–64 (2017)
 23. Sederberg, T., Anderson, D., Goldman, R.: Implicit representation of parametric curves and surfaces. *Comput. Vis. Gr. Image Process.* **28**(1), 72–84 (1984)
 24. Shikin, E.V.: Handbook and Atlas of Curves. CRC Press (1995)
 25. Torrente, M., Beltrametti, M., Sendra, J.: r-norm bounds and metric properties for zero loci of real analytic functions. *J. Comput. Appl. Math.* **336**, 375–393 (2018)
 26. Torrente, M.L., Beltrametti, M.C.: Almost vanishing polynomials and an application to the Hough transform. *J. Algebra Appl.* **13**(08), 145–157 (2014)
 27. Torrente, M.L., Biasotti, S., Falcidieno, B.: Recognition of feature curves on 3D shapes using an algebraic approach to Hough transforms. *Pattern Recognit.* **73**, 111–130 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

on the surface of 3D models based on the Hough transform technique.

Chiara Romanengo Chiara Romanengo graduated in Mathematics at the University of Genoa in March 2019. Since November 2019, she is a PhD student in “Mathematics and Applications” at the University of Genoa funded by CNR and is a research associate at the CNR-IMATI. Her research interests include curves and surfaces, algebraic geometry, features on 3D models, and geometric modelling. She investigates methods for the identification and recognition of characteristic parts

Bianca Falcidieno Bianca Falcidieno is an associate Research Director at CNR-IMATI. She has been leading and coordinating research at international level in Computational Mathematics, Computer Graphics, and Shape Modeling. She authored over 250 peer-reviewed publications and books. She is a Eurographics Fellow, a Pioneer of the Solid Modeling association and awarded with the Eurographics Gold Medal and the Tosiya Kunii Award for her scientific career.

CNR-IMATI research activities on surface similarity reasoning and is involved in the creation of datasets and benchmarks, in particular she organized several tracks of the SHape REtrieval Contest (SHREC).

Silvia Biasotti Silvia Biasotti is a Senior Researcher at CNR-IMATI. She graduated in Mathematics in 1998 and got a PhD in Mathematics and Applications in 2008, both from the University of Genoa. Her research interests include: shape modelling; similarity reasoning in presence of incomplete or partial information; pattern, feature and texture recognition on surfaces; change detection on 3D data; the exploration of large collections of 3D models. She was in charge of several