Costanza Conti, Lucia Romani, Daniela Schenone: Semi-automatic spline fitting of planar curvilinear profiles in digital images using the Hough transform, Pattern Recognition, Volume 74, 2018, Pages 64-76

https://doi.org/10.1016/j.patcog.2017.09.017

Semi-automatic spline fitting of planar curvilinear profiles in digital images using the Hough transform

Costanza Conti[a], Lucia Romani[b], Daniela Schenone[b,*]

[a]*Dipartimento di Ingegneria Industriale, Università di Firenze, Viale Morgagni, 40/44 - 50134 Firenze, Italy*
[b]*Dipartimento di Matematica e Applicazioni, Università di Milano-Bicocca, Via R. Cozzi 55 - 20125 Milano, Italy*

Abstract

We develop a novel method for the recognition of curvilinear profiles in digital images. The proposed method, semi-automatic for both closed and open planar profiles, essentially consists of a preprocessing step exploiting an edge detection algorithm, and a main step involving the Hough transform technique. In the preprocessing step, a Canny edge detection algorithm is applied in order to obtain a reduced point set describing the profile curve to be reconstructed. Also, to identify in the profile possible sharp points like cusps, we additionally use an algorithm to find the approximated tangent vector of every edge point. In the subsequent main step, we then use a piecewisely defined Hough transform to locally recognize from the point set a low-degree piecewise polynomial curve. The final outcome of the algorithm is thus a spline curve approximating the underlined profile image. The output curve consists of polynomial pieces connected G^1 continuously, except in correspondence of the identified cusps, where the order of continuity is only C^0, as expected. To illustrate effectiveness and efficiency of the new profile detection technique we present several numerical results dealing with detection of open and closed profiles in images of different type, i.e., medical and photographic images.

Keywords: Hough Transform, Profile recognition, Spline fitting, G^1-continuity, Cusps

1. Introduction

The Hough transform is a well-established and robust technique used in image analysis and digital image processing to recognize shapes in images with noisy backgrounds (see [12] and references quoted therein). The Hough transform was introduced in 1962 as a computational tool to detect straight lines in

*Corresponding author

Email addresses: `costanza.conti@unifi.it` (Costanza Conti), `lucia.romani@unimib.it` (Lucia Romani), `daniela.schenone@unimib.it` (Daniela Schenone)

images [9]. Later, in 1972, it was extended to detect also circles and ellipses [8]. Other more recent applications of the Hough transform technique for a faster detection of circles in images and for the complete detection of segments of lines with particular regard to their width, can be found in [10] and [18], respectively. A successive development of the feature detection procedure via the Hough transform was the introduction of the generalized Hough transform, aimed at extracting from images more complicated shapes that cannot be represented analytically in 2D space [3]. The idea behind the generalized Hough transform is that an arbitrary, non-analytic (but fixed) shape can be detected using the principle of template matching, namely by means of a predefined look-up table of a prototypal shape. More recently, exploiting algebraic geometry arguments, the Hough transform definition has been further extended to include also special classes of curves whose algebraic forms are known, but are significantly more complicated than straight lines or conics [1, 4, 5, 11]. Specifically, irreducible algebraic plane curves like elliptic curves, curves with 3 convexities, Wassenaar curves, conchoids of Slüse and piriform curves have been considered and the standard line and conic detection algorithm has been extended to the detection of such curves. To provide an efficient procedure for describing a real profile as a collection of different algebraic pieces from the same family, a *pixel–continuous piecewise* Hough transform has been recently introduced [15]. The main drawbacks of this method are that it is not automated, as it needs to know in advance a family of curves that can reasonably approximate the shape to be recognized, and the output curve is not continuous, but only pixel–continuous.

Prompted by the mentioned drawbacks, in this paper we develop a new technique for the semi-automated recognition of both closed and open curvilinear profiles in 2D digital images, that requires as input neither a family of predefined curves nor a predefined look-up table of a prototypal shape. Our method retains one of the main benefits of the Hough transform, i.e., the robustness with respect to background noise. Differently from the previously developed algorithm, our procedure approximates the sought profile with a G^1 continuous spline curve, eventually containing C^0 junctions where cusps occur.

The organization of the paper is as follows. In Section 2 we introduce some basic definitions and results about the Hough transform for straight lines and algebraic curves. In Section 3, the core of our paper, we present the general idea behind the new spline-based profile recognition method via piecewise Hough transform. In Section 4 we get more into details and provide a sketch of our algorithm. To test the validity of the suggested approach, in Section 5 we present several application examples dealing with the recognition of profiles coming from images of different type. Two of them are slice images from computer tomography (CT) scans, with or without noise, while two other application examples deal with photographic images, with or without noise. Finally, in the last section we draw conclusions.

2. The Hough Transform for lines and algebraic curves recognition

We first introduce the essential terminology and notation that is used throughout this paper. Then we present some known definitions and results about the Hough transform technique for straight lines and algebraic curves (for more details we refer the reader to [4, 8, 11, 13, 14, 15]).

Let K be either $\mathbb{R}$ or $\mathbb{C}$. For a positive integer n, we denote by $\mathbb{A}^n_K$ the n-dimensional affine space over K. For an n-tuple $\mathbf{\Lambda} = (\Lambda_1, \ldots, \Lambda_n)$ of indeterminates we denote by $K[\mathbf{\Lambda}; X, Y]$ the polynomial ring over K with indeterminates $\mathbf{\Lambda}, X, Y$, and by $F(\mathbf{\Lambda}; X, Y)$ a polynomial in $K[\mathbf{\Lambda}; X, Y]$. Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n) \in K^n$ be a point of $\mathbb{A}^n_K$, and let $P \in \mathbb{A}^2_K$ be a point of coordinates (x_P, y_P) in K^2. Then we define the polynomial $f_{\boldsymbol{\lambda}}(X, Y)$ as $F(\boldsymbol{\lambda}; X, Y)$, that is the polynomial $F(\mathbf{\Lambda}; X, Y)$ specialized in $\boldsymbol{\lambda}$. Conversely the polynomial $F(\mathbf{\Lambda}; x_P, y_P)$ in $K[\mathbf{\Lambda}]$ is $F(\mathbf{\Lambda}; X, Y)$ specialized in (x_P, y_P).

Since an algebraic curve $\mathcal{C}$ in $\mathbb{A}^2_K$ is defined as the zero locus of a polynomial f in $K[X, Y]$, a parameter dependent family of algebraic curves can be described by the polynomials $f_{\boldsymbol{\lambda}} \in K[\mathbf{\Lambda}; X, Y]$ as

$$\mathcal{F} = \{\mathcal{C}_{\boldsymbol{\lambda}} : f_{\boldsymbol{\lambda}}(X, Y) = 0 \mid \boldsymbol{\lambda} \in \mathcal{U}\}, \quad \text{with} \quad \mathcal{U} \subset K^n \text{ an open set.}$$

We continue by denoting by I the digital image which contains the profile that we want to recognize, and by $\mathcal{F}$ the family of curves that we plan on using. We call the affine space $\mathbb{A}^2_K$, onto which I and the curves of the family $\mathcal{F}$ are represented, the *image space*. Conversely, the affine space whose dimension is the number of free parameters in $\mathcal{F}$, is called the *parameter space*.

We next recall the definition of *Hough transform* of a point Q with respect to a line $\mathcal{L}$, based on the point–line duality notion.

Definition 1. Let $Q = (x_Q, y_Q)$ be a point in $\mathbb{A}^2_K$ and let

$$\mathcal{L} = \left\{\ell_{a,b} : Y - aX - b = 0 \mid (a, b) \in K^2\right\},$$

be a family of straight lines in the affine plane $\mathbb{A}^2_K(X, Y)$. The *Hough transform of Q with respect to $\mathcal{L}$* is the line $y_Q - Ax_Q - B = 0$ lying in the parameter plane $\mathbb{A}^2_K(A, B)$, namely $\Gamma_Q(\mathcal{L}) = \{y_Q - Ax_Q - B = 0\}$.

Thanks to algebraic geometry arguments, the Hough transform definition can be extended to include also special classes of curves that retain the one-to-one correspondence between curve $(C_{\boldsymbol{\lambda}})$ and parameters $(\boldsymbol{\lambda})$ that is proper of the families of lines [4, 11]. Specifically, let us consider a polynomial $F(\mathbf{\Lambda}; X, Y) \in K[\mathbf{\Lambda}; X, Y]$. Suppose now that F has degree d in the indeterminates X, Y. Then $F(\mathbf{\Lambda}; X, Y)$ can be written as

$$F(\mathbf{\Lambda}; X, Y) = \sum_{i,j \geq 0, i+j \leq d} g_{i,j}(\mathbf{\Lambda})\, X^i Y^j, \quad g_{i,j} \in K[\mathbf{\Lambda}].$$

Denoting by $\mathcal{F}$ the family of algebraic plane curves of the same degree d, the Hough transform of a point with respect to $\mathcal{F}$ is defined as follows.

Definition 2. Let $P = (x_P, y_P)$ be a point in $\mathbb{A}^2_K$ and let $\mathcal{F}$ be the family of algebraic irreducible plane curves of the same degree d given by

$$\mathcal{F} = \{\mathcal{C}_{\boldsymbol{\lambda}} : f_{\boldsymbol{\lambda}}(X, Y) = 0 \mid \boldsymbol{\lambda} \in \mathcal{U}\}, \quad \text{with} \quad \mathcal{U} \subset K^n \text{ an open set.}$$

The zero locus $\Gamma_P(\mathcal{F})$ of the polynomial $F(\boldsymbol{\Lambda}; x_P, y_P)$,

$$\Gamma_P(\mathcal{F}) = \{F(\boldsymbol{\Lambda}; x_P, y_P) = 0\}, \tag{1}$$

is called *the Hough transform of the point P with respect to the family $\mathcal{F}$ of algebraic curves.* $\Gamma_P(\mathcal{F})$ is a hypersurface in $\mathbb{A}^n_K$.

From the previous definition we easily see that

$$P \in \mathcal{C}_{\boldsymbol{\lambda}} \iff f_{\boldsymbol{\lambda}}(x_P, y_P) = F(\boldsymbol{\lambda}; x_P, y_P) = 0 \iff \boldsymbol{\lambda} \in \Gamma_P(\mathcal{F}). \tag{2}$$

Moreover, as a direct consequence of (2) we conclude that (see Figure 1)

$$\boldsymbol{\lambda} \in \bigcap_{P \in \mathcal{C}_{\boldsymbol{\lambda}}} \Gamma_P(\mathcal{F}).$$

We emphasize that, not for every family of irreducible algebraic curves there is a one-to-one correspondence between a curve of the family and a point in the parameter space. If this correspondence exists, the family $\mathcal{F}$ is called *HT-regular*. However, if a family $\mathcal{F}$ of algebraic curves is not HT-regular, we can always consider an open subset $\overline{\mathcal{U}}$ of $\mathcal{U}$ such that $\mathcal{F}_{|_{\overline{\mathcal{U}}}}$ is HT-regular [4].

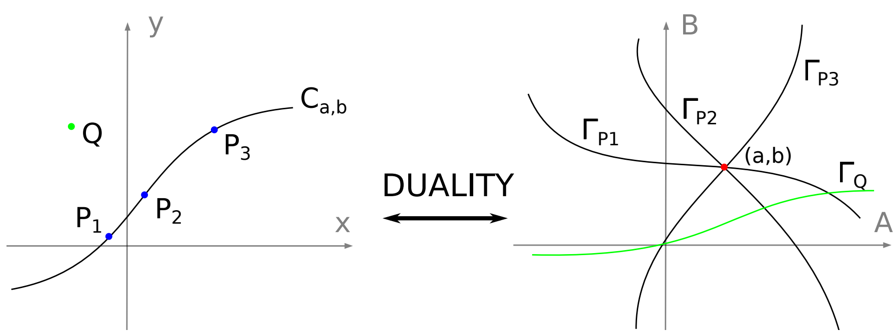

Figure 1: Duality-type correspondence between the image space (left) and the parameter space (right): each point P_i of the curve $C_{a,b}$ in the image space is transformed into a curve Γ_{P_i} in the parameter space, in such a way that all curves Γ_{P_i} meet at one point. This point (a, b) identifies the original curve to be detected. If $\mathcal{F}$ is HT-regular, then all Γ_{P_i} meet at one and only one point.

The Hough transform technique is mainly used for the detection of parametric shapes in images. To reduce the number of points for which one has to compute it, the Hough transform is usually applied after preprocessing the original image by an edge detector like, e.g., Canny edge detection algorithm [7]. The preprocessing step also eliminates the degree of freedom represented by the grey level in the image (in fact, this is out of the game of the recognition task). To apply the Hough transform, a discretization of the parameter space is required, which possibly exploits bounds on the parameter values computed by using either the cartesian or the parametric form of the curve in the image space. Once the parameter space is discretized in cells, an accumulator function is defined on it. The value of the accumulator function in a cell of the discretized space corresponds to the number of times the Hough transforms of selected points of interest reach that cell. The parameter values characterizing the curve to be detected in the image space correspond to the parameter values identifying the cell where the accumulator function reaches its maximum.

For further details on the standard profile recognition algorithm based on the Hough transform, we refer the reader to [3, 4, 12] and references therein.

3. A spline-based profile recognition technique via piecewise Hough transform

The new method we are going to present in this paper is based on the observation that the existing profile detection algorithms based on the Hough transform suffer from the following drawbacks:

- they cannot detect unknown shapes since they all require as input either a family of curves that can approximate the profile or a template for its shape (i.e., a predefined look-up table);
- a preprocessing step is needed to orient the image, or at least two more free parameters are needed to consider the right orientation.

On the contrary, our method does not require knowledge of the shape of the underlined searched profile, be it described by a look–up table as in [3], or by a family of algebraic curves as in [4]. Our idea to recognize profiles in images consists in combining splines (i.e., piecewise polynomial functions) with Hough transform. The Hough transform is in fact applied piecewisely i.e., exploiting only a specific subset of the point set identifying the whole profile. Indeed, we apply the Hough transform technique to subsets of the original dataset with respect to families of curves passing through a point of the last recognized curve piece. The process stops when a given percentage of points is recognized. If the profile to be recognized is closed, then one extra step is needed to connect the first and last curve pieces. Since, in general, the use of piecewise curves of low degree is preferred, we will confine ourselves to consider polynomial pieces of degree between 3 and 6. This choice is also supported by the observation that such degrees allow us to establish a good compromise between the number of curve pieces and the accuracy of the final fitting.

A detailed description of the novel profile recognition procedure is presented in the following subsections. We start by discussing the variations of our method with respect to the standard HT approach. We then continue by illustrating the pre-processing step aimed at reducing the number of points of which one

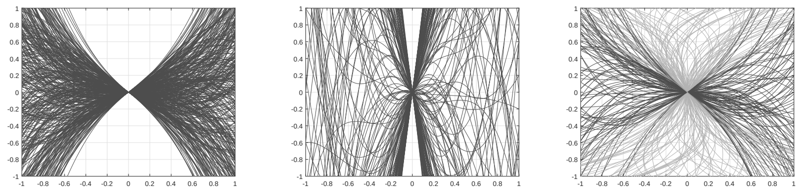

Figure 2: Four hundred curves of the family $\mathcal{F}_Y = \{\mathcal{C}_{a,b,c} : Y = aX^3 + bX^2 + cX \mid (a,b,c) \in \mathbb{R}^3\}$ with randomly chosen coefficients $(a,b,c) \in [-1,1]^3$ (left). Four hundred curves of the family $\mathcal{F}_Y = \{\mathcal{C}_{a,b,c} : Y = aX^3 + bX^2 + cX \mid (a,b,c) \in \mathbb{R}^3\}$ with randomly chosen coefficients $(a,b,c) \in [-10,10]^3$ (center). Two hundred curves of the family $\mathcal{F}_Y = \{\mathcal{C}_{a,b,c} : Y = aX^3 + bX^2 + cX \mid (a,b,c) \in \mathbb{R}^3\}$ with randomly chosen coefficients $(a,b,c) \in [-1,1]^3$ in dark gray, and two hundred curves of the family $\mathcal{F}_X = \{\mathcal{C}_{a,b,c} : X = aY^3 + bY^2 + cY \mid (a,b,c) \in \mathbb{R}^3\}$ with randomly chosen coefficients $(a,b,c) \in [-1,1]^3$ in light gray (right).

has to compute the Hough transform and also at identifying the presence of cusps in the reconstructed profile. Next, we describe the strategy to construct the piecewisely defined profile and, at last, we discuss how the free parameters are set.

3.1. Variations with respect to the standard HT approach

The families of algebraic curves that we use for the piecewise recognition of the profile have the general form

$$\mathcal{F}_{d,X} = \{C^d_\lambda : X = \lambda_d Y^d + \ldots + \lambda_1 Y + \lambda_0\}, \qquad 3 \le d \le 6,$$

or

$$\mathcal{F}_{d,Y} = \{C^d_\mu : Y = \mu_d X^d + \ldots + \mu_1 X + \mu_0\}, \quad 3 \le d \le 6.$$

Some of the parameters of the two families of curves (λ_i, $i = 0, ..., d$ for $\mathcal{F}_{d,X}$ and μ_i, $i = 0, ..., d$ for $\mathcal{F}_{d,Y}$) are fixed by some analytical conditions for each piece of the curve. We remark that we consider two families of curves, rather than just one, to overcome an intrinsic limit of the standard Hough transform consisting in the discretization of a limited region of the parameter space. Therefore the choice of two families allow for a wider variety of possible curves. This fact is highlighted in Figure 2. While the first two pictures show empty regions corresponding to the selection of a single family of curves (independently of the size of the discretized domain of variation of the parameters), the third shows the advantage of selecting two families of curves.

Instead of using the standard Hough transform technique, in our method we give to each point a particular weight. Precisely, in the accumulator matrix the weight of each point P in the set of points whose distance from a fixed point P^* is less than ε_1 (hereinafter denoted by $\mathcal{U}_{P^*,\varepsilon_1}$) is chosen to be inversely proportional to the squared euclidean distance between P and P^*. In this way, the points closer to the point P^* are privileged. To be more precise, the weight associated to each point P with respect to the starting point P^* of the curve is

$$\omega_{P^*}(P) = \frac{1}{dist(P^*, P)^2}, \tag{3}$$

where *dist* is the distance function defined as

$$dist : (P_k, P_j) \longmapsto \sqrt{(x_{P_k} - x_{P_j})^2 + (y_{P_k} - y_{P_j})^2}.$$

In the standard Hough transform technique, each Hough transform, passing through a cell in the discretization of the parameter space, contributes to the corresponding entry of the accumulator matrix by adding 1. On the other hand, for the point-distance weighted Hough transform, given a starting point P^*, the Hough transform of a point P of the dataset contributes to the entries of the accumulator matrix by adding $\omega_{P^*}(P)$. The resulting HT method is therefore called *point-distance weighted Hough transform.*

3.2. Pre-processing step

Let I be the digital image containing the profile to be recognized. The pre–processing step starts with the application of a Canny edge detection algorithm [7] to the image I. All of the pixels identified as discontinuity points by the edge detection form the so called *dataset*, hereinafter denoted by $\mathcal{D}$. Afterwards, the pre-processing step may involve two different stages. The first one consists in the segmentation of the obtained dataset by identifying 8–connected components in the image (this stage can be performed by using the MATLAB functions `bwlabel` and `bwconncomp`), and selecting the ones that are part of the profile we are interested in approximating. The second stage is the application of the algorithm proposed in [2], which takes as input a point cloud and selects the internal and external boundary of this dataset. With some images a combination of both stages is also considered. For example, to choose the dataset relative to the internal profile of the vertebra (Figures 6 and 11) we consider the connected components that have points that are part of the internal boundary.
Once the previous stages have been performed, we compute the local discrete tangent vector for every pixel of $\mathcal{D}$ (this computation can be easily performed by using the MATLAB function `regionprops`). Then we look for points of the profile in which there is a sharp change in the tangent vector of the profile. In case they occur, we keep track of them by defining a new point set called $\mathcal{D}_0$. This will be successively needed by our algorithm to select the junction points where the recognized curve has to be only C^0 and not G^1–continuous.

3.3. Piecewise construction of the profile

As already mentioned, the profile constructed by our method is defined piecewisely, by applying the Hough transform technique to subsets of the original dataset with respect to two different families of algebraic curves. The two families of algebraic curves are suitably defined to ensure that the outcome of the algorithm is a G^1 continuous spline curve, eventually containing some sharp points (cusps). Since the construction of the first spline piece is a bit different from all the successive ones, we separate the description of the construction in two corresponding parts. For shortness, in both descriptions we denote by d the degree of the families of algebraic curves used.

To construct the first piece of the profile, a starting point P_0 has to be selected (in this regard, see the following Subsection on free parameters setting). Then, by requiring that the curves of both families pass through the point P_0, one parameter for each family is fixed (i.e., λ_0 for $\mathcal{F}_{d,X}$ and μ_0 for $\mathcal{F}_{d,Y}$). The explicit conditions on the parameters are specifically

$$\lambda_0 = x_{P_0} - (\lambda_d y_{P_0}^d + \ldots + \lambda_1 y_{P_0}) \quad \text{and} \quad \mu_0 = y_{P_0} - (\mu_d x_{P_0}^d + \ldots + \mu_1 x_{P_0}).$$

The two families of curves identified by these parameters are denoted as $\mathcal{F}_{d,X,P_0}$ and $\mathcal{F}_{d,Y,P_0}$, respectively. The idea to compute the first piece of the profile curve, is to apply the Hough transform to the subset $\mathcal{U}_{P_0,\varepsilon_1}$ of $\mathcal{D}$, namely to the set of all points in $\mathcal{D}$ whose distance from P_0 is less than a fixed number ε_1. We emphasize that the Hough transform technique is applied with respect to both families $\mathcal{F}_{d,X,P_0}$ and $\mathcal{F}_{d,Y,P_0}$ so that two approximating curves are obtained. The one having the smallest average distance from the approximated points, is chosen to be the curve which better approximates the portion of the profile in $\mathcal{U}_{P_0,\varepsilon_1}$. We refer to this first approximating curve as $\mathcal{C}_1$, and we point out that $\mathcal{C}_1$ represents only the profile piece that approximates the points in $\mathcal{U}_{P_0,\varepsilon_1}$. The curve piece $\mathcal{C}_1$ is considered an acceptable piece of the final profile if at least d^2+1 points from $\mathcal{U}_{P_0,\varepsilon_1}$ are approximated by $\mathcal{C}_1$. A point P in $\mathcal{U}_{P_0,\varepsilon_1}$ is said to be approximated by $\mathcal{C}_1$ if a discrete estimation of the euclidean distance between P and $\mathcal{C}_1$ is less than a fixed number ε_2. We remark that we choose d^2+1 as the minimum number of points that has to be approximated in order to consider a curve piece suitable for the final profile, because the space of bivariate polynomials with degree less than or equal to d has dimension $\frac{(d+2)(d+1)}{2}$. Although our families of curves usually do not have d^2+1 free parameters, it is a reasonable bound to be considered [6].

Before starting with the description of the construction of the other pieces of the profile curve, we need to initialize some variables in terms of the curve piece $\mathcal{C}_1$ defined at first. Precisely, in the construction of the $(k+1)^{\text{th}}$-piece of curve, the new starting point P_k of the curve piece $\mathcal{C}_{k+1}$ is chosen as follows. We first compute the arclength of the curve $\mathcal{C}_k$, taking P_{k-1} as the starting point in which the arclength is zero. Then we define P_k as the point on $\mathcal{C}_k$ with the greatest arclength. If the curve piece we intend to construct is G^1–joined to $\mathcal{C}_k$ in P_k, then we compute the unit tangent vector of $\mathcal{C}_k$ in P_k and we call it $\mathbf{u}_k$. In this case the families of curves that we use for the recognition of $\mathcal{C}_{k+1}$ depend on both P_k and $\mathbf{u}_k$, which means that $\lambda_1, \lambda_0, \mu_1$, μ_0 are fixed in the following way:

$$\lambda_1 = \frac{1}{\tan(\mathbf{u}_k)} - (d\lambda_d y_{P_k}^{d-1} + \ldots + 2\lambda_2 y_{P_k}), \quad \lambda_0 = x_{P_k} - (\lambda_d y_{P_k}^d + \ldots + \lambda_1 y_{P_k}),$$

or

$$\mu_1 = \tan(\mathbf{u}_k) - (d\mu_d x_{P_k}^{d-1} + \ldots + 2\mu_2 x_{P_k}), \quad \mu_0 = y_{P_k} - (\mu_d x_{P_k}^d + \ldots + \mu_1 x_{P_k}),$$

where $\tan(\mathbf{u}_k)$ stands for the tangent of the angle between $\mathbf{u}_k$ and the x-axis. We thus denote the corresponding new families of curves as $\mathcal{F}_{d,X,P_k,\mathbf{u}_k}$ and

$\mathcal{F}_{d,Y,P_k,\mathbf{u}_k}$. The subset of points to which we apply the Hough transform with respect to the two families of curves previously computed, is now $\mathcal{U}_{P_k,\varepsilon_1}$ (i.e., the set of all points in $\mathcal{D}$ whose distance from P_k is less than ε_1), excluded the points that have already been approximated by $\mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$, $\mathcal{C}_k$. Next we compute the approximating curve $\mathcal{C}_{k+1}$ as it was done for $\mathcal{C}_1$. If less than d^2+1 points of the dataset have a distance from the curve $\mathcal{C}_{k+1}$ that is smaller than ε_2 (namely $\mathcal{C}_{k+1}$ cannot be considered an acceptable piece of the final profile curve), then a new starting point, close to the previous one, is chosen on $\mathcal{C}_k$ and the last part of the process here described is repeated. To make sure that our procedure provides the curve piece $\mathcal{C}_{k+1}$ in a reasonable number of repetitions, we limit the number of times that the point P_k can be changed. This limit is chosen to be the degree of the curve $\mathcal{C}_k$. If this limit is reached, we do an extra step, in which we choose manually a new point Q_0 in the dataset $\mathcal{D}$, closer than ε_1 to P_k, and we join P_k to Q_0 as we would do for the last step of a closed profile.

3.4. Free parameters setting

The outcome of the novel recognition algorithm implementing the described procedure depends on the choice of the following free parameters:

- A vector $\boldsymbol{d}$ of positive integers, where the k^{th} entry is the degree of the two families of algebraic curves used for the recognition of the k^{th} polynomial piece of the sought profile. As specified below, the value of each entry of $\boldsymbol{d}$ depends on how many constraints we want to impose on each profile piece. In our procedure the number of constraints is kept as small as possible to reduce the computational costs of the Hough transforms.

- A real positive value ε_1, radius of a 2-dimensional ball centered at the starting point of the degree-d polynomial piece and containing the points to which we locally apply the Hough transform technique (Figure 3). The parameter ε_1 has to be at least bigger than d^2+1 times the length of the side of a pixel in the image space since we want to approximate at least d^2+1 points of the dataset. The upper bound of ε_1 is $1/8$ of the length of the shortest edge of the input image.

- A real positive value ε_2 providing the biggest acceptable distance between each point of the dataset and the closest point on the approximating curve, to consider that point approximated (Figure 3). We choose ε_2 as the length of the side of a pixel in the image space.

- The starting point $P_0 = (x_{P_0}, y_{P_0})$ in $\mathcal{D}$ for the piecewise curve to be constructed. This point needs to be chosen on the profile in an appropriate way. For example, if in the pre-processing step some points with sharp orientation variations have been found, they are good candidates for the starting point P_0. In case an appropriate point is not easily found, the staring point is an input given by the user sufficiently close to the searched profile. This is what makes the method semi-automatic.

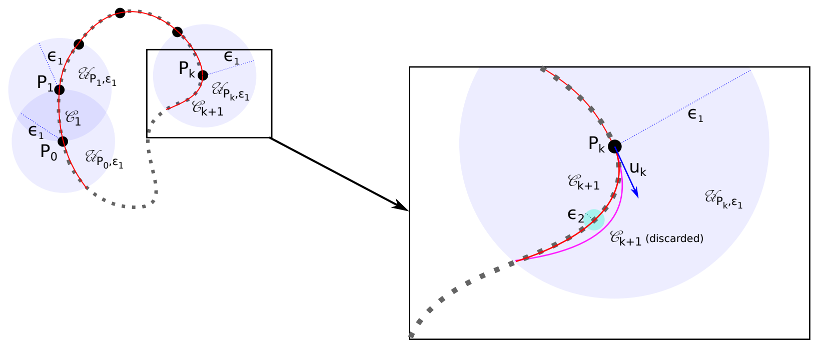

Figure 3: Graphical illustration of the meaning of the parameters ε_1 and ε_2.

Before concluding this section, we would like to detail how $\boldsymbol{d}$ is chosen. To keep the computational cost of the method low, we use families of curves with three free parameters only. Keeping the number of free parameters equal to 3 in each curve piece, the degree d of the two families of curves used for the recognition of every single polynomial piece depends on the smoothness order we request at the connecting points. Precisely,

- $d = 3$ (from first to last piece of the profile), if the piecewise curve has to be globally C^0 continuous;
- $d = 3$ for the first piece and $d = 4$ for all the following ones, if the piecewise curve is required to be globally G^1 continuous and is open;
- $d = 3$ for the first piece, $d = 4$ for all the following ones, if the piecewise curve is required to be globally G^1 continuous, and $d = 6$ for the last piece if the curve is closed.

In Figure 4 an example of a closed G^1 continuous profile curve is illustrated. The number of free parameters of the families of curves used for the recognition is always 3. Thus, the degree is $d = 3$ for the first polynomial piece of the curve, given the fact that we only have to interpolate P_0. Instead, $d = 4$ for every other polynomial piece of curve given the fact that we interpolate both the point P_k and its unit tangent vector. For the last piece of curve we have $d = 6$ as we interpolate both P_N, P_0 and their corresponding unit tangent vectors.

4. Sketch of the algorithm for the recognition of open and closed profiles

In this Section we provide the sketch of the recognition algorithm derived from the method described in Section 3. Either closed or open curves, which can be either globally G^1-continuous or G^1-continuous with some C^0 junction points, can be obtained as output.

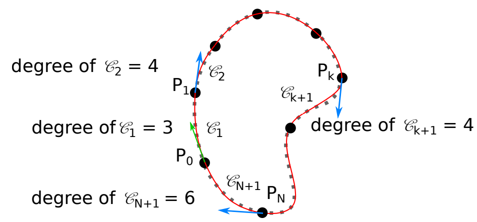

Figure 4: Closed profile made of G^1-joined polynomial pieces having degree between 3 and 6. The number of free parameters for every piece is equal to 3, but the interpolation conditions used in their construction are different.

Let P_0 be the starting point, $\mathcal{D}$ the initial dataset, $\mathcal{D}_0$ the points of the profile in which there is a sharp change of orientation. Moreover let $\mathcal{F}_{d,X}$ and $\mathcal{F}_{d,Y}$ be the general families used for the recognition of each curve piece. For referring to specific families of curves passing through the point P we use $\mathcal{F}_{d,X,P}$ and $\mathcal{F}_{d,Y,P}$ or $\mathcal{F}_{d,X,P,\mathbf{u}}$ and $\mathcal{F}_{d,Y,P,\mathbf{u}}$ if a unit tangent vector $\mathbf{u}$ in P is also fixed.

For every point P and every real number ε_1, $\mathcal{U}_{P,\varepsilon_1}$ is defined as the set of points of $\mathcal{D}$ that lie inside the 2-dimensional ball of center P and radius ε_1.

By $\mathtt{HT}(\mathcal{F}_d, \mathcal{U}_{P,\varepsilon_1})$ we mean the computation of the Hough Transform of the points in $\mathcal{U}_{P,\varepsilon_1}$ with respect to the family $\mathcal{F}_d$.

Sketch of the algorithm.

1. input: $\mathcal{D}$, $\mathcal{D}_0$, ε_1, ε_2, $\mathbf{d}$.
2. `if` $\#\mathcal{D}_0 > 0$ `do`
   choose P_0 in $\mathcal{D}_0$
   `else`
   choose P_0 in $\mathcal{D}$
   `end`
3. compute $\mathcal{U}_{P_0,\varepsilon_1}$
4. `if` $\#\mathcal{U}_{P_0,\varepsilon_1} < (d(1))^2 + 1$ `do`
   change P_0
   `endif`
5. compute $\mathcal{C}_X = \mathtt{HT}(\mathcal{F}_{d(1),X,P_0}, \mathcal{U}_{P_0,\varepsilon_1})$ and $\mathcal{C}_Y = \mathtt{HT}(\mathcal{F}_{d(1),Y,P_0}, \mathcal{U}_{P_0,\varepsilon_1})$
6. call $\mathcal{C}_1$ the best approximating curve between $\mathcal{C}_X$ and $\mathcal{C}_Y$
7. compute $\mathcal{U}_{\mathcal{C}_1}$, i.e., the set of points of $\mathcal{D}$ approximated by the curve $\mathcal{C}_1$ (in other words find the points of $\mathcal{D}$ whose distance from the curve $\mathcal{C}_1$ is smaller than ε_2)
8. `if` $\#\mathcal{U}_{\mathcal{C}_1} < (d(1))^2 + 1$ `do`
   choose another starting point P_0 and `go to` 3.
9. define P_{new} as the arclength-wise most distant point in $\mathcal{C}_1$ from P_0
10. compute $\mathbf{u}_{P_{\text{new}},\mathcal{C}_1}$, i.e., the unit tangent vector to $\mathcal{C}_1$ in P_{new}
11. compute $\mathcal{D} = \mathcal{D} \setminus \mathcal{U}_{\mathcal{C}_1}$
12. `while` $\#\mathcal{D} > 0$, `do`

12.1. set $P_k = P_{\text{new}}$ and $\mathbf{u}_k = \mathbf{u}_{P_{\text{new}},\mathcal{C}_k}$
12.2. compute $\mathcal{U}_{P_k,\varepsilon_1}$
12.3. `if` $\#\mathcal{U}_{P_k,\varepsilon_1} < (d(k+1))^2 + 1$ `do`
choose another starting point P_{new} on $\mathcal{C}_{k+1}$
`go to` 12.1
`endif`
12.4. compute $\mathcal{C}_X = \mathtt{HT}(\mathcal{F}_{d(k+1),X,P_k,\mathbf{u}_k}, \mathcal{U}_{P_k,\varepsilon_1})$ and $\mathcal{C}_Y = \mathtt{HT}(\mathcal{F}_{d(k+1),Y,P_k,\mathbf{u}_k}, \mathcal{U}_{P_k,\varepsilon_1})$
12.5. call $\mathcal{C}_{k+1}$ the best approximating curve between $\mathcal{C}_X$ and $\mathcal{C}_Y$
12.6. compute $\mathcal{U}_{\mathcal{C}_{k+1}}$, i.e., the set of points of $\mathcal{D}$ approximated by the curve $\mathcal{C}_{k+1}$ (in other words find the points of $\mathcal{D}$ whose distance from the curve $\mathcal{C}_{k+1}$ is smaller than ε_2)
12.7. `if` $\#\mathcal{U}_{\mathcal{C}_{k+1}} < (d(k+1))^2 + 1$ `do`
choose another starting point P_{new} on $\mathcal{C}_k$
`if` $\#\mathcal{U}_{\mathcal{C}_{k+1}} < (d(k+1))^2 + 1$ for more than $d(k)$ times `do`
12.7.1. choose a new point Q_0 on the curve, do the extra step for closed profiles to connect the last P_k with Q_0 (see Remark 1)
12.7.2. start again the algorithm with Q_0 as the starting point
`endif`
`go to` 12.1
`endif`
12.8. define P_{new} as the arclength-wise most distant point in $\mathcal{C}_{k+1}$ from P_k
12.9. `if` G^1 recognition `do`
compute $\mathbf{u}_{P_{\text{new}},\mathcal{C}_{k+1}}$
`endif`
12.10. compute $\mathcal{D} = \mathcal{D} \setminus \mathcal{U}_{\mathcal{C}_{k+1}}$
13. `endwhile`

Remark 1. In the case of an open profile (see, e.g., Figures 5 and 6 in Subsection 5.1), the algorithm stops when enough points are considered approximated. If the profile is closed and has to be G^1 continuous at the closure point, then we need to perform an extra step in which we use the Hough transform technique with respect to a family of curves which pass through the first and the last point, and have given unit tangent vectors in those points.

Before concluding the section we would like to remark on the computational cost of our algorithm. It is a known result that the computational cost of the standard Hough transform is $\mathcal{O}(A^{m-2})$ [17], where A is the dimension of the image space and m is the number of parameters of the family of curves. The computational cost of our algorithm is therefore

$$2\sum_{k=1}^{N} d(k)\,\mathcal{O}(A(k)^{m(k)-2}),$$

where N is the number of curve pieces (which becomes $N+1$ in case of closed profiles), $A(k)$ the dimension of the image space in the k^{th} step (concerning the construction of the k^{th} curve piece), $m(k)$ the number of parameters of the family of curves in the k^{th} step. The factor two is due to the two families of curves used in the algorithm, while the factor $d(k)$ is the maximum number of times that the algorithm can change the starting point P_k before a new one, Q_0, is manually selected.

5. Experimental results

This section is to provide some application examples of our new method to several real images. We consider CT scans and photographic images, also with addition of synthetic noise. In the first case, we recognize two piecewise G^1-continuous curves to be associated, respectively, to the external and internal profiles of a vertebra, and a G^1-continuous curve with some C^0 junctions, to be associated to the left/right external profile of a hip. In the second case, we recognize piecewise continuous curves (again G^1-continuous with some C^0 junctions) to be associated with the profiles of a hand and a face, also with addition of synthetic noise. The method is implemented with datasets that, after a suitable rescaling, are in the region $[-4,\ 4]\times[-4,\ 4]$ and $[-10,\ 10]\times[-10,\ 10]$ of the Cartesian plane for the medical images and for the photographic images, respectively. Every pixel is 0.0333 units in the Cartesian plane in which the shape is recognized. The parameter ε_1 is chosen in the real interval $[0.8, 1.5]$ and ε_2 is chosen equal to 0.03, that is it approximately coincides with the number of units that correspond to a pixel in the original image. Moreover, in case the parameter space used for the recognition is of dimension two, the region we discretize is $[-1,\ 1]\times[-1,\ 1]$, with 100×100 cells. In the case of three parameters, the discretized region of the parameter space is $[-1,\ 1]\times[-1,\ 1]\times[-1,\ 1]$, with $40\times 40\times 40$ cells.

5.1. Medical images

For the recognition of the external and internal profile of the vertebra (given in Figures 5 and 6 top left), the free parameters of the algorithm have been selected as in Table 1.

	Vertebra ext.	Vertebra int.	Hip left	Hip right
ε_1	0.8	1.2	1	1
ε_2	0.03	0.03	0.03	0.03
$\mathbf{d}$	$(3, 4, 4, ..., 4)$	$(3, 4, 4, ..., 4)$	$(3, 4, 4, ..., 4, 6)$	$(3, 4, 4, ..., 4, 6)$

Table 1: The choice of the free parameters for the medical images in Figures 5, 6, 7, 8, 9, 10.

With these two recognitions some extra remarks are needed. We start with the case of the external profile. From the dataset obtained by the Canny edge detection algorithm (Figure 5 top center) we proceed by identifying a sub-set of points that are part of the external profile only. This is possible by the help of the algorithm in [2] capable of recognising the boundary points. The result is in Figure 5 top right. We then apply our algorithm to the corresponding rescaled points shown in Figure 5 bottom left. A spline curve made of 19 pieces (see Figure 5 bottom center) is the recognized profile superposed on the original image in Figure 5 bottom right. The errors for this recognition are presented in Table 2.

	Vertebra ext.	Vertebra int.	Hip left	Hip right
MAE	0.0234	0.0372	0.0350	0.0427
RMSE	0.0338	0.0431	0.0576	0.0871

Table 2: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for the spline profiles obtained from the medical images.

Next, we discuss the case of the internal profile. From the dataset obtained by the Canny edge detection algorithm (Figure 6 top center) we identify, again by the algorithm in [2], a sub-set of points that are part of the internal profile only. Then, we apply the MATLAB function `bwlabel` to separate the different connected components obtained with the edge detection (Figure 6 top right). The final dataset to which we apply our algorithm is obtained by considering the points of the connected components that are touched by the internal boundary. The result is in Figure 6 bottom left. A spline curve made of 11 pieces (see Figure 6 bottom center) is the recognized profile superposed on the original image in Figure 6 bottom right. The errors for this recognition are presented in Table 2.

For completeness the results of the application of the algorithm in [2] for both external and internal profile of the vertebra, are in Figure 11 left-center.

In Figure 7 we present a recognition with background noise, meaning uniformly distributed noise independent from the profile. The profiles we want to recognize are again the external and internal profiles of the vertebra in Figure 5 and Figure 6 top left. In both cases, the ratio between the number of noise points with respect to the number of points on the profiles is 2 to 1. In Figure 7 left (top-down) we can see the dataset including the points obtained after applying the edge detection algorithm and the synthetic noise points. The profiles are recognized with a spline curve made of 20 and 11 pieces (see Figure 7 center, top-down). The recognized curves superimposed on the original image are in Figure 7 right (top-down).

Remark 2. We remark that, the Hough transform technique for algebraic curves has been already used to recognize profiles of vertebras by using elliptic curves and Slüse conchoids (see [11, 15, 16]). In both cases, the image needs a roto-translation as a pre-processing step for the recognition to be effective. In [15] an iterative pixel-connected version of the Hough transform algorithm is presented, but it is still based on the a priori knowledge of a family of potentially approximating algebraic curves. Moreover, in the Example presented in [15, Sect. 4, Fig. 2], a set of 7 different curves is used, but from [15, Sect. 4, Fig.2(d)] we can see that the final curve is composed of at least 25 pieces.

Next we consider the recognition of a CT slice of a hip bone. As far as we know, the recognition of this profile is new in the Hough transform literature. For the recognition of the left and right external profile of the hip (given in Figures 8 and 9 top left), the free parameters of the algorithm have been selected as in Table 1. For these two examples the errors for the approximations are again presented in Table 2.

As for the case of the vertebra, to identify the points of the dataset that are part of the external profile, we apply the algorithm in [2] (see Figure 11 right). Then we apply the MATLAB function `bwlabel` to separate the different connected components obtained with the edge detection. Further, to identify the two datasets corresponding to the hip left and hip right images, we proceed with the recognition of the two connected components that are touched by the external boundary. A spline curve made of 26 pieces (see Figure 8 bottom center) is the recognized profile, which is shown superposed on the original image in Figure 8 bottom right. Instead, the spline curve made of 21 pieces (in Figure 9 bottom center) is the recognized profile shown in Figure 9 bottom right.

In Figure 10 we present a recognition with background noise, meaning uniformly distributed noise independent from the profile. The profiles we want to recognize are again the left and right profiles of the hip in Figures 8 and 9 top left. In both cases, the ratio between the number of noise points with respect to the number of points on the profiles is 2 to 1. In Figure 10 left (top-down) we can see the dataset including the points obtained by applying the edge detection algorithm and the synthetic noise points. The profiles are recognized with a spline curve made of 28 and 21 pieces, respectively (see Figure 10 center, top-down). The recognized curves superimposed on the original image are in Figure 10 right (top and down).

5.2. Photographic images

For the recognition of the external profiles of the hand and the face (given in Figures 12 and 13), the free parameters of the algorithm have been selected as in Table 3. In both cases, our algorithm is applied to sub-sets of points produced by the Canny edge detection algorithm. They are the external connected components identified by means of the MATLAB function `bwlabel`. The errors for these recognitions are presented in Table 4.

The hand profile in Figure 12 top left is recognized with a spline curve made of 64 pieces visible in Figure 12 bottom center. The errors for this recognition are presented in Table 4.

For the recognition of the external profile of the face in Figure 13 top left, the free parameters of the algorithm have been selected as in Table 3. A spline curve made of 26 pieces (in evidence in Figure 13 bottom center) is the recognized profile superposed on the original image in Figure 13 bottom right.

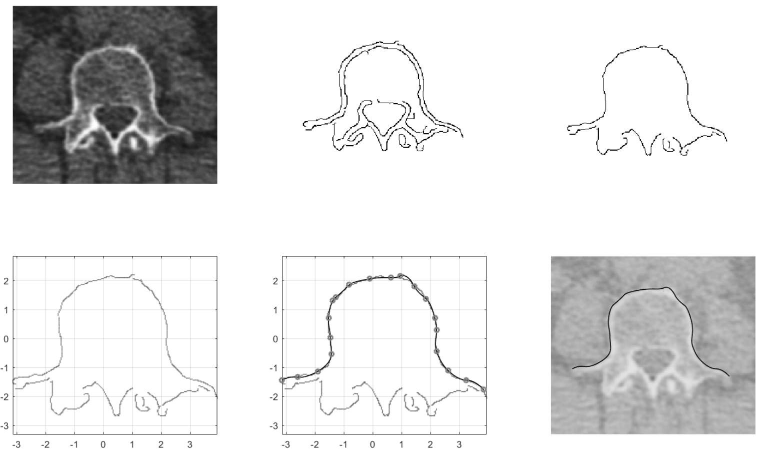

Figure 5: Recognition of the external profile of a vertebra from a slice of a CT scan. Top: The original CT slice image of a vertebra (left). Output of the Canny edge detection algorithm (center). Points of the image belonging to the external profile and selected using the algorithm in [2], as shown in Figure 11 (left). Bottom: The rescaled dataset points, obtained by edge detection (left). The recognized G^1-connected piecewise curve, with the junction points highlighted by small circles (center). The recognized curve superimposed on the original image (right).

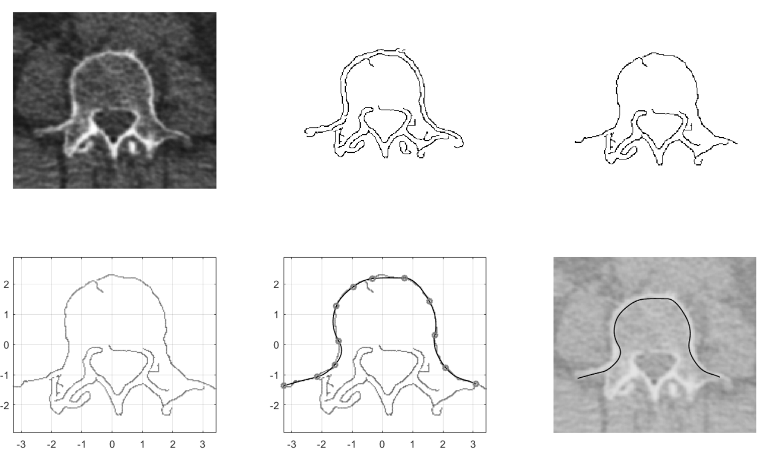

Figure 6: Recognition of the internal profile of a vertebra from a slice of a CT scan. Top: The original CT slice image of a vertebra (left). Output of the Canny edge detection algorithm (center). Points of the image belonging to the connected components that are touched by the internal boundary, obtained from [2] as in Figure 11 (center). Bottom: The rescaled dataset points, obtained by edge detection (left). The recognized G^1-connected piecewise curve, with the junction points highlighted by small circles (center). The recognized curve superimposed on the original image (right).

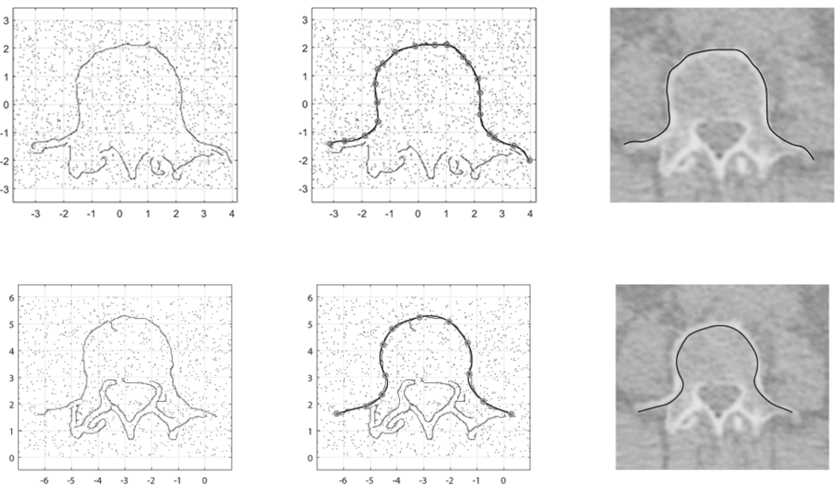

Figure 7: Recognition of the external and internal profile of a vertebra from a slice of a CT scan with synthetic noise. The pre–processing steps are not included in this Figure as they are the same as the ones in Figures 5 and 6. Left column: the rescaled dataset points, obtained by edge detection, plus the synthetic noise. Center column: the recognized G^1-connected piecewise curves with the junction points highlighted by small circles. Right column: the recognized curves superimposed on the original image.

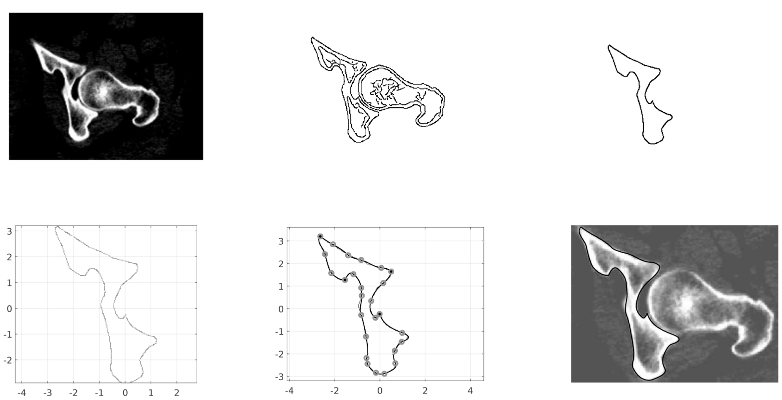

Figure 8: Recognition of the left external profile of a hip from a slice of a CT scan. Top: The original CT slice image of a hip (left). The output of the Canny edge detection algorithm (center). The connected component relative to the left part of the external profile of the hip (right). Bottom: The rescaled dataset points, obtained by edge detection (left). The recognized piecewise curve with the G^1-junction points and the subset of C^0-junction points marked in black (center). The recognized curve superimposed on the original image (right).

Finally, in Figure 14 we present a recognition with background noise. The profiles we want to recognize are the hand in Figure 12 and the face in Figure

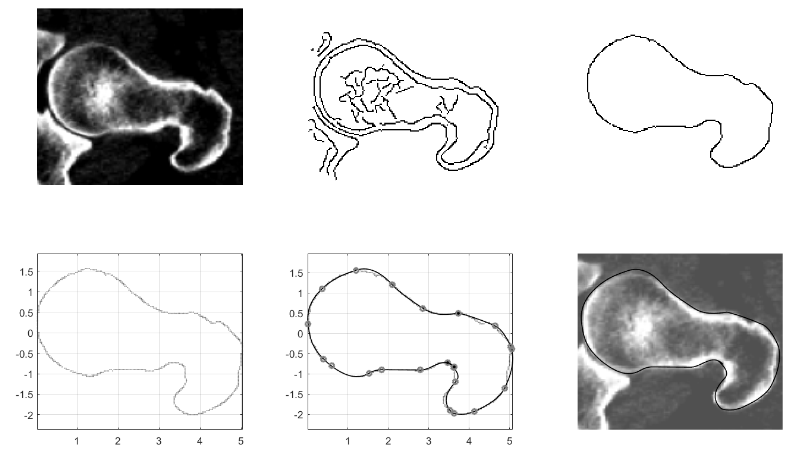

Figure 9: Recognition of the right external profile of a hip from a slice of a CT scan. Top: The original CT slice image of a hip (left). The output of the Canny edge detection algorithm (center). The connected component relative to the right part of the external profile of the hip (right). Bottom: The rescaled dataset points, obtained by edge detection (left). The recognized piecewise curve with the G^1-junction points and the subset of C^0-junction points marked in black (center). The recognized curve superimposed on the original image (right).

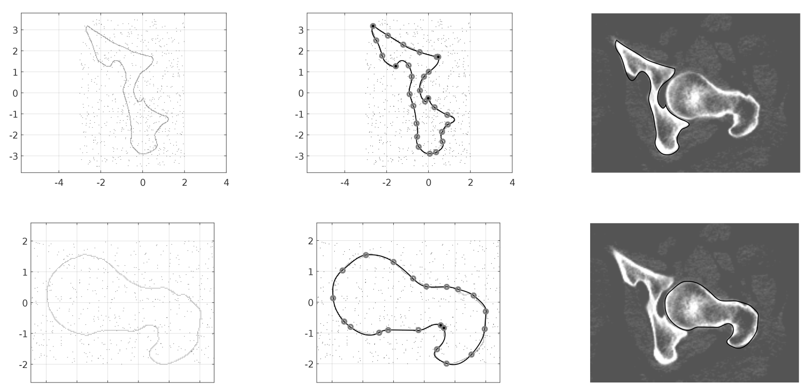

Figure 10: Recognition of the left and right external profile of a hip with synthetic noise. The pre–processing steps are not included in this Figure as they are the same as the ones in Figures 8 and 9. Left column: the rescaled dataset points, obtained by edge detection, plus the synthetic noise. Center column: the recognized G^1-connected piecewise curve with the junction points highlighted by small circles. Right column: the recognized curve superimposed on the original image.

13 top left. In both cases, the ratio between the number of noise points with respect to the number of points on the profiles is 2 to 1. In Figure 14 left

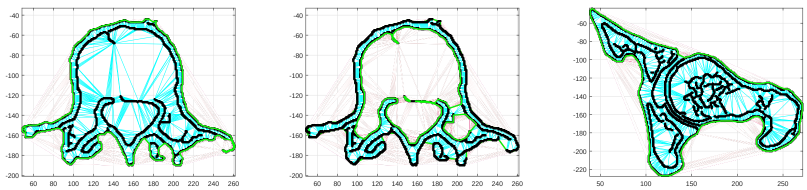

Figure 11: Output of the algorithms in [2] for points extracted by the Canny edge detection. The external boundary of the dataset for the vertebra (left), the internal boundary of the dataset for the vertebra (center) and the external boundary of the dataset for the hip (right) are highlighted in green.

	Hand	Face
ε_1	1.5	1.2
ε_2	0.03	0.03
$\mathbf{d}$	$(3, 4, 4, ..., 4)$	$(3, 4, 4, ..., 4)$

Table 3: The choice of the free parameters for the photographic images in Figures 12, 13, 14.

	Hand	Face
MAE	0.0234	0.0372
RMSE	0.0338	0.0431

Table 4: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for the spline profiles obtained from the photographic images.

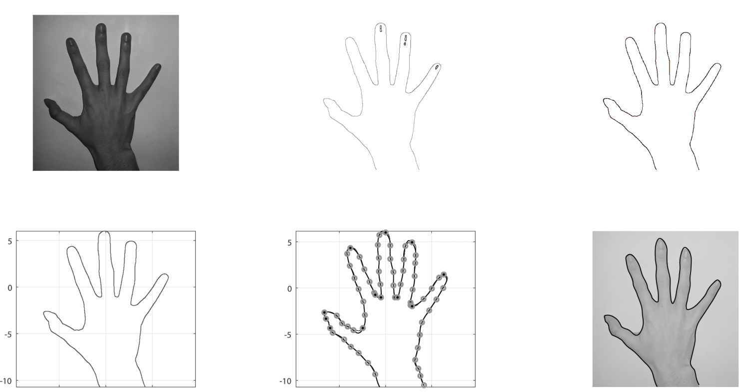

Figure 12: Recognition of the external profile of a hand. Top: The original grayscale photographic image (left). The image after the application of the Canny edge detection and of the MATLAB function `bwlabel` that separates the connected components of the edge detection (center). The connected component we want to approximate (right). Bottom: The rescaled dataset points (left). The recognized piecewise curve (center). The recognized curve superimposed on the original image (right).

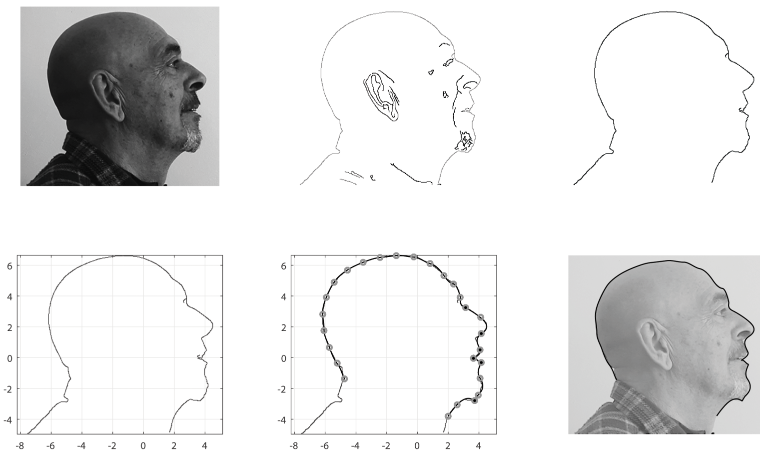

Figure 13: Recognition of the external profile of the side profile of a face. Top: The original grayscale photographic image (left). The image after the application of the Canny edge detection and of the MATLAB function `bwlabel` that separates the connected components of the edge detection (center). The connected component we want to approximate (right). Bottom: The rescaled dataset points (left). The recognized piecewise curve (center). The recognized curve superimposed on the original image (right).

(top-down) we can see the dataset including the points obtained after applying the edge detection algorithm and the synthetic noise points. The profiles are recognized with a spline curve made of 62 and 29 pieces, respectively (see Figure 14 center, top-down). The recognized curves superimposed on the original image are in Figure 14 right (top-down).

6. Conclusions and future works

This paper proposes a new method to recognize arbitrary profiles in 2D images without any previous knowledge of them. Based on the use of a piecewisely defined Hough Transform, it can provide either a globally G^1-continuous spline or a G^1-continuous spline with some C^0 junction points. The method is semi-automatic, very efficient and robust in presence of noise. An interesting topic of further research is the replacement of the Standard Hough Transform with the Randomized Hough Transform, which uses a dynamic accumulator.

References

[1] C. Athwal. *Detection of Deformable Structures in Video by Polynomial Fitting Using an Efficient Hough Transform*, pages 396–403. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

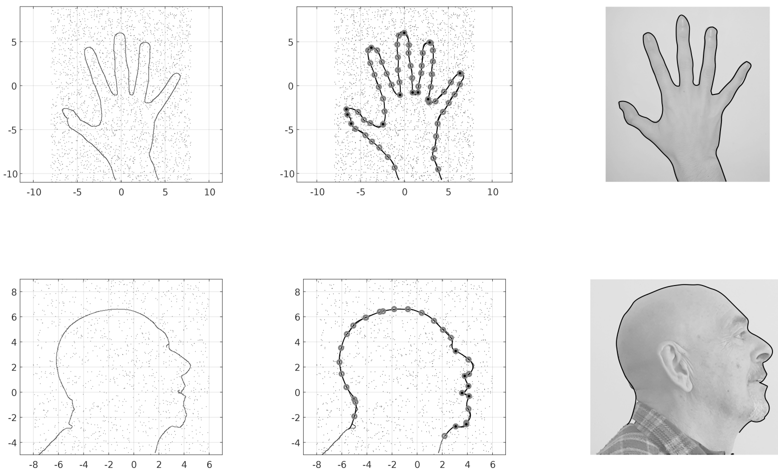

Figure 14: Recognition of the external profile of a hand and of the side profile of a face in Figure 12 and Figure 13, respectively. Left column: the rescaled dataset points, obtained by edge detection, plus the synthetic noise. Center column: the recognized G^1-connected piecewise curve with the junction points highlighted by small circles. Right column: the recognized curves superimposed on the original image.

[2] M. Awrangjeb. Using point cloud data to identify, trace, and regularize the outlines of buildings. *International Journal of Remote Sensing*, 37(3):551–579, 2016.

[3] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[4] M. C. Beltrametti, A. M. Massone, and M. Piana. Hough transform of special classes of curves. *SIAM J. Imaging Sci.*, 6(1):391–412, 2013.

[5] M. C. Beltrametti and L. Robbiano. An algebraic approach to hough transforms. *Journal of Algebra*, 371:669–681, 2012.

[6] C. Campi and M.-L. Torrente. Some finite bounds for testing the hough regularity of special classes of algebraic curves. In *Proceedings of SIMAI2016*, 2016.

[7] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986.

[8] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972.

[9] P. V. C. Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[10] T. De Marco, D. Cazzato, M. Leo, and C. Distante. Randomized circle detection with isophotes curvature analysis. *Pattern Recognition*, 48(2):411 – 421, 2015.

[11] A. M. Massone, A. Perasso, C. Campi, and M. C. Beltrametti. Profile detection in medical and astronomical images by means of the Hough transform of special classes of curves. *J. Math. Imaging Vision*, 51(2):296–310, 2015.

[12] P. Mukhopadhyay and B. B. Chaudhuri. A survey of hough transform. *Pattern Recogn.*, 48(3):993–1010, March 2015.

[13] A. Perasso, C. Campi, A. M. Massone, and M. C. Beltrametti. Spinal canal and spinal marrow segmentation by means of the Hough transform of special classes of curves. In *Image analysis and processing—ICIAP 2015. Part I*, volume 9279 of *Lecture Notes in Comput. Sci.*, pages 590–600. Springer, Cham, 2015.

[14] J. Princen, J. Illingworth, and J. Kittler. A formal definition of the hough transform: Properties and relationships. *Journal of Mathematical Imaging and Vision*, 1(2):153–168, 1992.

[15] G. Ricca, M.C. Beltrametti, and A. M. Massone. Piecewise recognition of bone skeleton profiles via an iterative hough transform approach without re-voting. In *Proc. SPIE*, volume 9413, 2015.

[16] G. Ricca, M.C. Beltrametti, and A. M. Massone. Detecting curves of symmetry in images via Hough transform. *Math. Comput. Sci.*, 10(1):179–205, 2016.

[17] G. Stockman and L. G. Shapiro. *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[18] Zezhong Xu, Bok-Suk Shin, and Reinhard Klette. Closed form line-segment extraction using the hough transform. *Pattern Recognition*, 48(12):4012 – 4023, 2015.