mathematics

MDPI

Article

Weighted Quasi-Interpolant Spline Approximations of Planar Curvilinear Profiles in Digital Images

Andrea Raffo * and Silvia Biasotti

Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes", Consiglio Nazionale delle Ricerche, Via de Marini 6, 16149 Genoa, Italy; silvia.biasotti@ge.imati.cnr.it

* Correspondence: andrea.raffo@ge.imati.cnr.it

Abstract: The approximation of curvilinear profiles is very popular for processing digital images and leads to numerous applications such as image segmentation, compression and recognition. In this paper, we develop a novel semi-automatic method based on quasi-interpolation. The method consists of three steps: a preprocessing step exploiting an edge detection algorithm; a splitting procedure to break the just-obtained set of edge points into smaller subsets; and a final step involving the use of a local curve approximation, the Weighted Quasi Interpolant Spline Approximation (wQISA), chosen for its robustness to data perturbation. The proposed method builds a sequence of polynomial spline curves, connected C^0 in correspondence of cusps, G^1 otherwise. To curb underfitting and overfitting, the computation of local approximations exploits the supervised learning paradigm. The effectiveness of the method is shown with simulation on real images from various application domains.

Keywords: spline functions; quasi-interpolation; image processing; G^1 continuity

check for updates

Citation: Raffo, A.; Biasotti, S. Weighted Quasi-Interpolant Spline Approximations of Planar Curvilinear Profiles in Digital Images. *Mathematics* **2021**, *9*, 3084. https://doi.org/10.3390/math9233084

Academic Editors: Domingo Barrera, María José Ibáñez and Sara Remogna

Received: 22 October 2021
Accepted: 27 November 2021
Published: 30 November 2021

1. Introduction

Approximating curvilinear profiles in digital images with piecewise polynomials is very attractive in many application domains, as it leads to more compact and less wiggly representations of borders and contours. In this work, we study the applicability of the quasi-interpolation paradigm for approximating open and closed curves that can be modeled as 1-manifolds, that is, topological spaces wherein each point has a neighborhood that is homeomorphic to the Euclidean space of dimension 1; an extension to piecewise 1-manifolds is straightforward by decomposing the curve in 1-manifolds and proceeding one segment at a time [1].

The term quasi-interpolation has been interpreted differently by different authors, depending on the context of application. We here call quasi-interpolant any linear operator L of the form:

$$Lf := \sum_{j \in J} \mu_j(f) g_j, \tag{1}$$

where $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ is a function being approximated, μ_j are linear functionals, $g_j : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ are functions at our disposal (see, for example [2,3]). The coefficients $\mu_j(f)$ are, in general, one of the following types: linear combinations of given values of the function f to be approximated (*discrete type*); linear combinations of values of derivatives of f, of order at most d (*differential type*); linear combinations of weighted mean values of f (*integral type*). Equation (1) can be interpreted as a "reconstruction" formula: given some input data sampled from the true function f, it creates a tentative reconstruction Qf.

The origin of quasi-interpolation is traditionally traced back to Bernstein's approximation [4], where the functions g_j are Bernstein polynomials. A rather simple generalization, known as Variation Diminishing Spline Approximation (VDSA), generalizes this construction to B-splines (see, for example [5,6]). Since its inception, quasi-interpolation has been studied to obtain methods that apply to different domains and with the aim of increasing the order of convergence: recent developments include univariate and tensor-product spaces [7–9], triangular meshes [10–13], quadrangulations [14] and tetrahedra partitions [15], among others.

Unlike traditional quasi interpolation spline methods, which generally focus on the approximation of functions with strong regularity conditions, Weighted Quasi Interpolant Spline Approximations (wQISAs) [16,17] aims to provide local approximations of point clouds which can be affected by artifacts such as noise and outliers. However, a paradigm to compute global approximations was, so far, missing; indeed, the only two global approximations presented in [16] are obtained by manually processing the set of edge points, thanks to the quite simple geometry of the studied profiles. In this paper, we introduce a novel method that exploits wQISAs to construct global approximations of planar curvilinear profiles, with application to digital image processing. More specifically, the main contributions of this paper are:

- The introduction of a novel algorithm, based on Weighted Quasi-Interpolant Spline Approximations, to compute global approximations of planar curvilinear profiles;
- The validation of the method on real images from different application domains with respect to different evaluation measures.

The smoothness of a curve is generally distinguished between parametric continuity C^n and geometric continuity G^n [18,19]. A curve $\mathbf{p}(t)$ is said to be C^n continuous at $t = t_0$ if it is continuous at t_0 and if the first n left and right derivatives of $\mathbf{p}$ match at t_0, that is,

$$\lim_{t \to t_0^-} \mathbf{p}^{(m)}(t) = \lim_{t \to t_0^+} \mathbf{p}^{(\mathbf{m})}(t), \quad m = 0, \ldots, n.$$

A curve $\mathbf{p}(t)$ is said to be G^n continuous at $t = t_0$ if, up to regular re-parametrization, it is C^n continuous at $t = t_0$. Recently, the concept of fractional continuity has been proposed for generalized fractional Bézier curves [20]. In our context, the global approximation of a planar profile is computed by gluing together local approximations, by imposing a C^0 continuity in correspondence of cusps, and a G^1 continuity otherwise; while it is possible to consider the more general case of G^n (or C^n) continuity, G^1 continuity has proved to be sufficient in many practical applications (see, for example [21,22]).

The remainder of the paper is organized as follows. In Section 2, we provide some notation and background knowledge about the wQISA family. In Section 3, the core of our paper, we introduce our reconstruction algorithm. To test the validity of the suggested approach, Section 4 presents several experimental results on images from different application domains and with respect to different evaluation measures. Concluding remarks end the paper.

2. Preliminary Concepts

This section is meant to list some basic definitions regarding the wQISA family, as well as to introduce the essential terminology and notation. Because in this paper we focus on 1-manifolds, we restrict our attention to a univariate formulation of wQISA; for more details on the more general multivariate setting and an extended analysis of the theoretical properties, we refer the reader to [16].

Given a degree p, a knot vector is said to be $(p+1)$-*regular* if no knot occurs more than $p+1$ times and each boundary knot occurs exactly $p+1$ times.

Let $\mathcal{P} \subset \mathbb{R}^2$ be a point cloud and $p \in \mathbb{N}^*$. Let $\mathbf{x}$ be a $(p+1)$-regular knot vector with boundary knots $x_1 = \cdots = x_{p+1} = a$ and $x_{n+1} = \cdots = x_{n+p+1} = b$. A *Weighted Quasi Interpolant Spline Approximation* of degree p to the point cloud $\mathcal{P}$ over the knot vector $\mathbf{x}$ is defined by:

$$f_w(x) := \sum_{i=1}^{n} \hat{y}_w(x_i^*) \cdot B[\mathbf{x}_i](x), \tag{2}$$

where $x_i^* := (x_i + \ldots + x_{i+p})/p$ are the *knot averages*, the expression,

$$\hat{y}_w(u) := \frac{\sum\limits_{(x,y)\in\mathcal{P}} y \cdot w(x,u)}{\sum\limits_{(x,y)\in\mathcal{P}} w(x,u)}, \tag{3}$$

gives the *control polygon estimator* with weight function $w : \mathbb{R} \times \mathbb{R} \to [0, +\infty)$; $B[\mathbf{x}_i] : \mathbb{R} \to [0, 1]$ denotes the B-spline of degree p, which is uniquely determined by the *local knot vector* $\mathbf{x}_i = [x_i, \ldots, x_{i+p+1}]$. Note that, given a point cloud, its wQISA depends on the following inputs: a spline space, uniquely defined by a degree and a (global) knot vector, and a weight function (possibly, with free parameters to be tuned). An example of weight function is given by the *k-Nearest Neighbours* (*k*-NN) average:

$$w(x,u) := \begin{cases} 1/k, & \text{if } x \in N_k(u) \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where $k \in \mathbb{N}^*$; $N_k(u)$ denotes the neighborhood of u defined by the k closest points of the point cloud; here, k is the free parameter.

Figure 1 shows the approximation of a dataset of 250 points when a 5-NN weight is considered. The point cloud is sampled from the analytic function $F(x) = x \sin(\pi/2x)$, and is eventually perturbed with Gaussian noise $\epsilon \sim N(0, s(x))$, where

$$s(x) := \frac{1}{\sqrt{7}(0.5 + e^{3x-1})}. \tag{5}$$

The approximation is computed by a quadratic spline space containing $n = 20$ B-splines over a uniform knot vector.

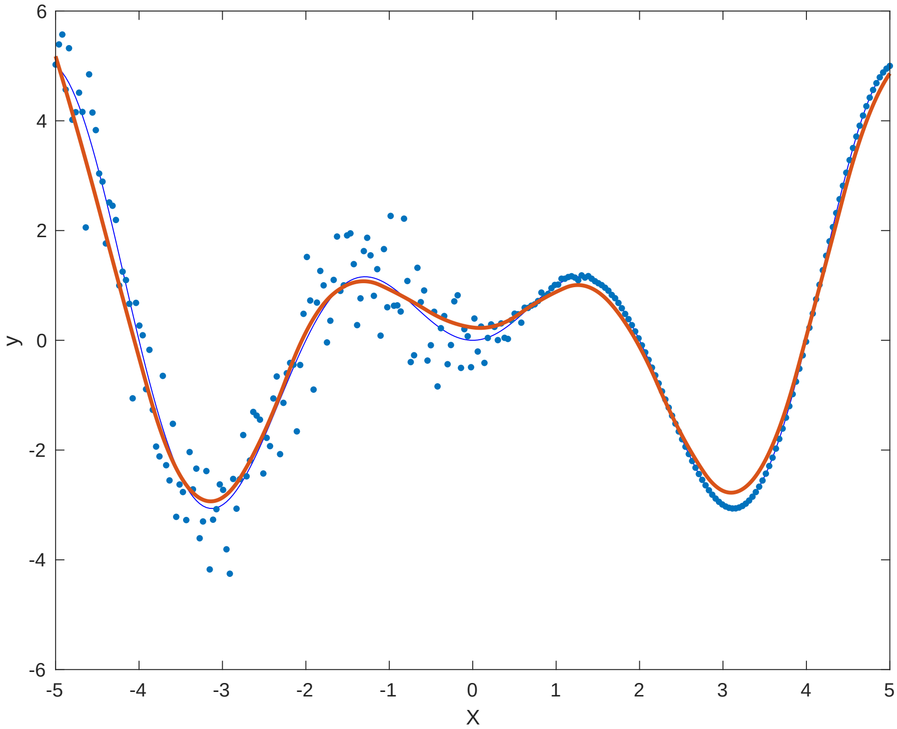


Figure 1. Variable noise approximation. The analytic function $F(x) = x \sin(\pi/2x)$ is sampled and then perturbed with (non-uniform) Gaussian noise of mean 0 and the standard deviation of Equation (5). The resulting wQISA approximation, obtained by a 5-NN weight function, is displayed red.

3. Curvilinear Profile Approximation via a Quasi-Interpolation-Based Technique

In this section, we design an algorithm that piecewisely applies wQISA to the approximation and recognition of open and closed planar curvilinear profiles. The method consists of three main steps:

- Pre-processing step (see Section 3.1);
- Point cloud split (see Section 3.2);
- Local approximation via wQISA (see Section 3.3).

3.1. Pre-Processing Step

Firstly, we run an edge detection algorithm to an input a (grayscale) image, hereinafter denoted by I: in our case, we used the Canny edge detection [23], as it is considered among the most effective edge detection techniques [24,25]. To give an example, when considering the mushroom of Figure 2, we note that the Canny edge detection algorithm provides a much better result than other popular methods, see Figure 2: indeed, the Canny edge detection provides a much cleaner set of edge points, and is the only one able to identify the whole curvilinear profile without wrongly decomposing it into subparts.

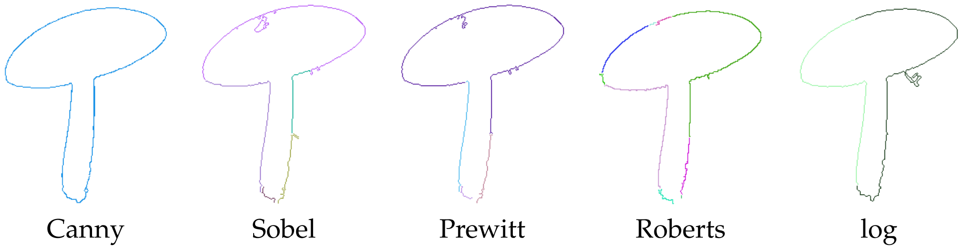


Figure 2. Different edge detection methods applied to an image presented in Section 4. Different colors identify different 8-connected components.

The pixels identified as discontinuity points by the edge detection algorithm are then processed to identify 8-connected components. Their centres are maintained in $\mathcal{P}$. For the sake of conciseness, we will hereinafter suppose that $\mathcal{P}$ consists of a single 8-connected component; in case of more than one 8-connected component, the reconstruction method will be applied as many times as their number.

For every point in $\mathcal{P}$, the local discrete tangent vector is computed (for example, by means of `regionprops` in MATLAB). All those points with a sharp change in the unit tangent vector are stored in $\mathcal{P}_0$, and will be eventually used to impose a C^0 continuity instead of G^1 continuity; to put it another way, the order of continuity will be only C^0 in correspondence with the identified cusps, as expected. A graphical illustration of the C^0 and G^1 continuities is given in Figure 3.

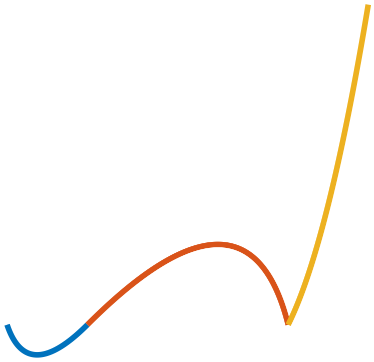


Figure 3. Graphical illustration of the C^0 and G^1 continuities: the curves in blue and red are connected with G^1 continuity; the curves in red and yellow are connected with C^0 continuity.

3.2. *Point Cloud Split*

Differently from the method in [26] that iteratively covered a curvilinear profile with disks of fixed radius, in our pipeline the point cloud $\mathcal{P}$ is split into subsets, each of which is meant to be approximated independently. The key idea behind our procedure is loosely inspired by the concept of *tubular neighborhoods*—see, for example [27]. A graphical illustration of two possible iterations is given in Figure 4.

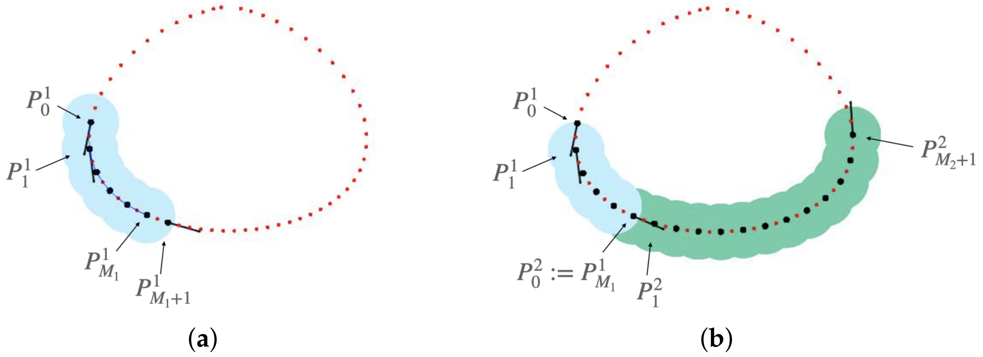


Figure 4. Graphical illustration for the computation of local tubular neighbourhoods for two iterations of the algorithm. For the sake of clarity, we here consider an over-simplified situation, with a noise-free point cloud represented in red. (**a**) First iteration; (**b**) Second iteration.

3.2.1. Constructing the k-th Local Tubular Neighborhood, $k \geq 1$

Given an initial point P_0^k, where $k \geq 1$, and an input radius $R_1 > 0$, let $\mathcal{B}(P_0^k, R_1)$ denote the (2-)ball of center P_0^k and radius R_1. We build a sequence of points $P_0^k, \ldots, P_{M_k+1}^k$ such that:

- For any index $1 \leq i \leq M_k + 1$, the point P_i^k is computed from P_{i-1}^k as follows:
  - *Advance step*. Set Q_i^k to be the farthest point from P_{i-1}^k in $\mathcal{B}(P_{i-1}^k, R_1) \cap \mathcal{P}$, among all those points that were not contained in any previously-computed balls.
  - *Smoothing step*. Set P_i^k to be the point in $\mathcal{P}$ which is the closest to the barycenter of the points in $\mathcal{B}(Q_i^k, R_2) \cap \mathcal{P}$, where $R_2 < R_1$ is an input value that should account for the amount of noise.
- The index $i = M_k$ is the first one for which either of the following criteria holds true:
  - *Angle criterion*. The angle between the tangent vectors in P_0^k and $P_{M_k+1}^k$ is above the threshold α_0.
  - *Point criterion*. The ball $\mathcal{B}(P_{M_k+1}^k, R_1) \cap \mathcal{P}$ does not contain new points (i.e., points which were not contained in previously-computed balls).
  - *Loop criterion*. The ball $\mathcal{B}(P_{M_k}^k, R_1)$ contains the initial point P_0^1.

When stopping because of the angle criterion, we define the *k-th local tubular neighbourhood* as:

$$\mathcal{P}_k := \left(\bigcup_{i=1}^{M_k - 1} \mathcal{B}(P_i^k, R_1) \cap \mathcal{P} \right) \Bigg\backslash \left(\bigcup_{j=1}^{k-1} \mathcal{P}_j \right). \tag{6}$$

Its candidate edge points are $E_0^k := P_0^k$ and $E_1^k := P_{M_k}^k$; finally, we set $P_0^{k+1} := P_{M_k}^k$ to be the initial point for the subsequent local tubular neighborhood. On the contrary, when stopping because of either the point or loop criteria, the k-th local tubular neighbourhood is defined as:

$$\mathcal{P}_k := \left(\bigcup_{i=1}^{M_k} \mathcal{B}(P_i^k, R_1) \cap \mathcal{P} \right) \Bigg\backslash \left(\bigcup_{j=1}^{k-1} \mathcal{P}_j \right), \tag{7}$$

and its candidate edge points are $E_0^k := P_0^k$ and $E_1^k := P_{M_k+1}^k$, where $P_{M_k+1}^k := P_0^1$ in case the loop criterion holds.

When stopping because of the point criterion, a new search is conducted from the initial point P_0^1 backward, to take into account the possibility of open profiles.

Note that the whole splitting procedure merely relies on the following input: the initial point P_0^1, the radii R_1 and R_2, and the threshold α_0. A more detailed discussion on the parameter setting is left to Section 3.4.

3.2.2. Updating the k-th Local Tubular Neighborhood, $k \geq 1$

The local tubular neighbourhoods $\mathcal{P}_1, \ldots, \mathcal{P}_N$ allow us to locally represent the curvilinear profile of interest. However, their construction does not take into account the points in $\mathcal{P}_0$, that is, the points having a sharp change in the tangent vector.

Suppose that a cusp $\tilde{P}$ is inside the two balls $\mathcal{B}(P_i^k, R_1)$ and $\mathcal{B}(P_{i+1}^k, R_1)$. Then, we split the k-th local tubular neighbourhood into two new local tubular neighbourhoods:

- One will contain all balls $\mathcal{B}(P_1^k, R_1), \ldots, \mathcal{B}(P_{i-1}^k, R_1)$, plus the reduced ball
$$\mathcal{B}(P_i^k, \min\{R_1, ||P_i^k - \tilde{P}||_2\}).$$
- The other one will contain all balls $\mathcal{B}(P_{i+1}^k, R_1), \ldots, \mathcal{B}(P_{M_k-1}^k, R_1)$—and $\mathcal{B}(P_{M_k}^k, R_1)$, in case of Equation (7)—plus the reduced ball
$$\mathcal{B}(P_{i+1}^k, \min\{R_1, ||P_{i+1}^k - \tilde{P}||_2\}).$$

The cusp $\tilde{P}$ will be set to be a common edge point of the two new local tubular neighbourhoods.

3.3. *Local Approximation via wQISA*

For any $\mathcal{P}_k$, the local parametrization is computed as follows:

1. *Change of coordinates*. Apply a roto-translation $\rho : \mathbb{R}^2 \to \mathbb{R}^2$ to $\mathcal{P}_k$, so that the line segment
$$< \rho(E_0^k), \rho(E_1^k) >$$
lies on the x-axis. This corresponds to assuming that our local approximation can be locally flattened onto the x-axis without any overlap; note that this assumption relies on the parameter α_0.
2. *Local spline space*. Define $\boldsymbol{\tau}_j^k := [\tau_1, \ldots, \tau_{j+6}]$ as the j-th 3-regular knot vector where:
    - For any $j \in \mathbb{N}$, $\tau_1 = \tau_3 = E_0^{k,\rho,x}$ and $\tau_{j+4} = \tau_{j+6} = E_1^{k,\rho,x}$, being $E_i^{k,\rho,x}$ the abscissa of the roto-translation of E_i^k with respect to the map ρ. Note that the map ρ can be chosen so that $E_0^{k,\rho,x} < E_1^{k,\rho,x}$.
    - The knots $\tau_3, \ldots, \tau_{j+4}$ are uniformly sampled in $[E_0^{k,\rho,x}, E_1^{k,\rho,x}]$.
3. *Local wQISA approximation*. Compute a local approximation for $\mathcal{P}_k$ by applying Equation (2) with k-NN weight function; note that the parameter k in the weight function does not necessarily equal the index k in the sequence of local tubular neighborhoods.

A discussion on the tuning of the free parameters (i.e., knot vector length and k of the k-NN weight function) is provided in Section 3.4.

Given that for any local tubular neighbourhood, the corresponding knot vector is $(p+1)$-regular, the final approximation will be automatically C^0 continuous. To impose G^1 continuity, we exploit $(p+1)$-regularity: the tangents to the curve at the boundaries are determined by its 2nd and $(n-1)$th control points; we project the 2nd and $(n-1)$th onto the straight lines determined by the tangent vectors. A similar reasoning can be applied to interpolate the left and right derivatives at cusps, up to a multiplicative constant.

3.4. Free Parameters and Tuning

The initial point P_0^1 is set to be the cusp, which is the closest to the lower-left corner. If the profile does not exhibit any cusps, we can consider the point of maximum (or minimum) curvature—the closest to the lower-left corner if more than one exists .

The radii R_1 and R_2 depend on the profiles and the noise level, and are currently empirically set by the user.

For each local approximation, the optimal global knot vector $\boldsymbol{\tau}_{\text{opt}}^k$ and the optimal k_{opt} for the weight function can be found by a model assessment and selection procedure, as in statistical learning; more precisely, we proceed by performing Leave-One-Out Cross-Validation (LOO-CV, see [28]), by considering the Root Mean Square Error as a generalization error (see Section 4). This allows us to limit under- and overfitting; an example is given in Figure 5 of a local tubular neighborhood exhibiting both noise and outliers. The use of Cross-Validation for the estimation of the unknown coefficients allows us to interpret the curve fitting problem as a supervised learning problem (see, for example [29]).

To generalize the method, one could select the continuity G^k at the segment edge points by cross-validation; however, G^1 continuity has proved to be sufficient for all the examples given in Section 4.

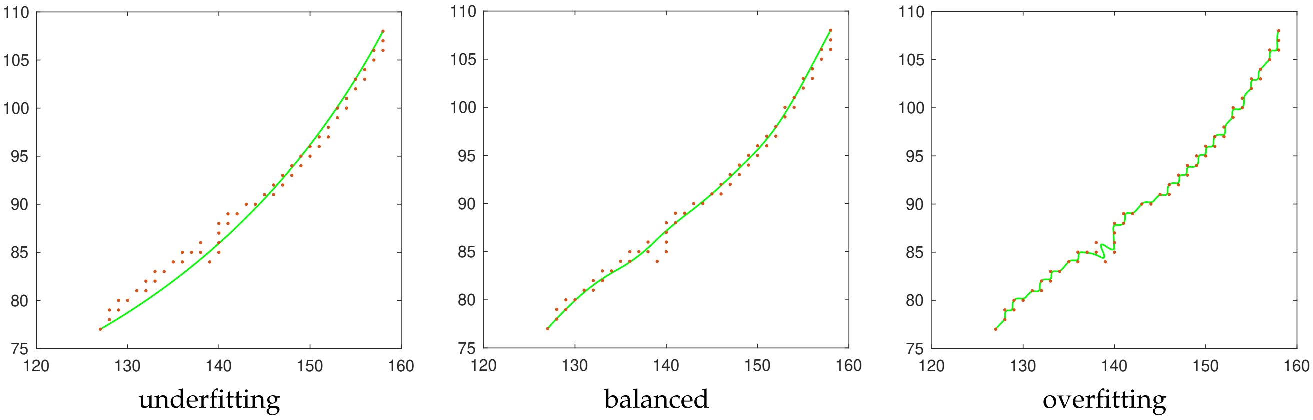


Figure 5. Three approximations for a local tubular neighborhood from Figure 6. In the left image, the local approximation consists of an oversimplification of the input data. In the right image, the local approximation models the training data too well, resulting in unwanted oscillations that reduce the prediction capability on new points. A good balance between under- and over-fitting is displayed in the central image.

3.5. Computational Complexity

The computational complexity of the parametrization step is the sum of the single wQISA iterations. On its turn, wQISA mainly depends on the k-NN weight, whose time complexity for estimating a single coefficient is proportional to $O(k\log(N))$, where N is the number of points. To further reduce the computational complexity of the k-NN search, we follow the k-d trees approach proposed in [30]. It is worth noting that, each coefficient and each local approximation being computed independently, this task is *embarrassingly parallel.*

The splitting procedure can itself rely on a k-d tree, thus it has the same computational complexity. On the other hand, it cannot be performed independently, and turns out to be the most demanding part of the algorithm.

4. Experimental Results

We evaluate our method on photographic images, animation images and CT scans. Although the approximation could theoretically make use of B-splines of any degree, we here focus on quadratic spline approximations, as they provide a sufficient flexibility for our purposes; nevertheless, one could consider the degree of each local approximation as an additional parameter to be tuned.

4.1. Performance Measures

To evaluate the quality of an approximation against the input point clouds, the following indicators are considered:

- The *Root Mean Square Error* (RMSE), that is, the square root of the *Mean Square Error*, is defined by:

$$\mathrm{RMSE} := \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2},$$

  Y_i being the abscissa of an input point with respect to its local coordinate system, $\hat{Y}_i$ its approximation, and N the cardinality of the input point cloud. Similarly, the *Mean Absolute Error* (MAE) is given by:

$$\mathrm{MAE} := \frac{1}{N}\sum_{i=1}^{N}|Y_i - \hat{Y}_i|.$$

- The (normalized) *Directed Hausdorff Distance* from the points of $a \in A \subset R^2$ to the points of $b \in B \subset R^2$ is defined as:

$$d_{\mathrm{dHaus}}(A,B) := \frac{1}{L}\max_{a\in A}\min_{b\in B} d(a,b),$$

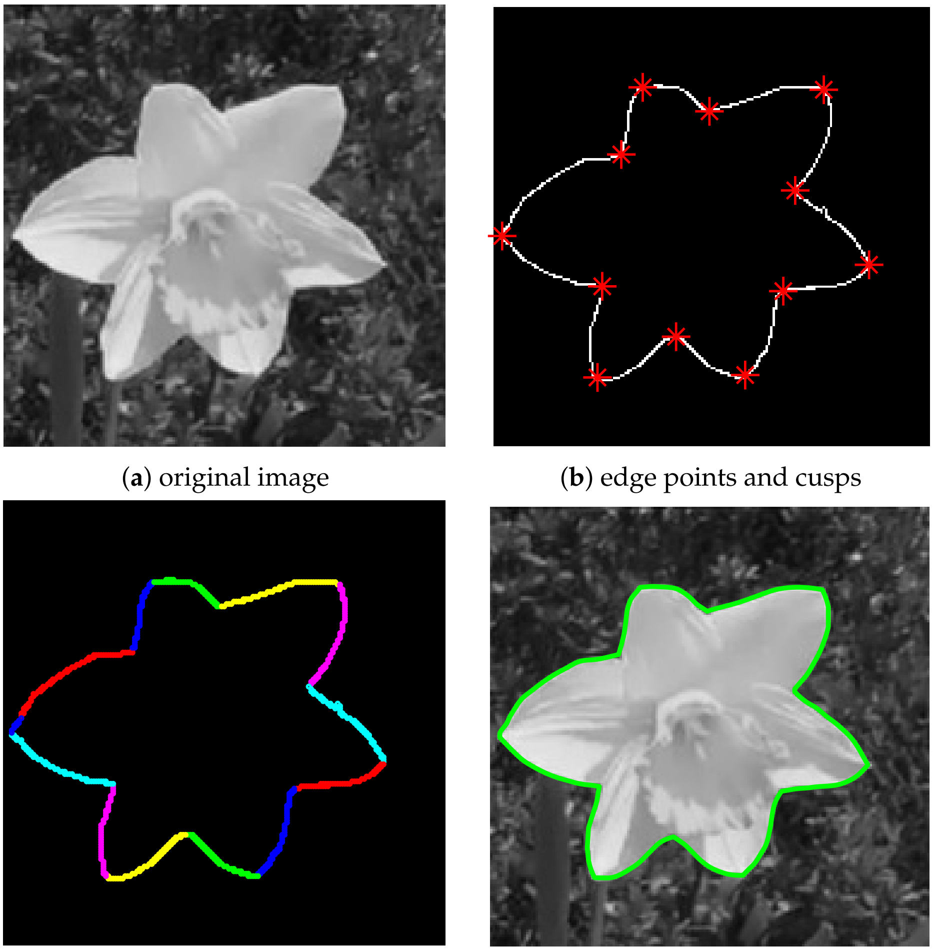
  where d is the Euclidean distance and L is the diameter of the input point cloud. In our case, we set B to be the input point cloud $\mathcal{P}$, while A is given by the points defining our spline approximation.
- The *Jaccard index*, or *Intersection over Union*, measures how overlapping two sets A and B are; it is computed as:

$$J(A,B) := \frac{|A \cap B|}{|A \cup B|},$$

  $|\cdot|$ being the cardinality of a set. The Jaccard index has been intensively applied to measure the performance of curve recognition methods for images (see, for example [31]). Here, we compare the set of pixels corresponding to $\mathcal{P}$ with the set of pixels crossed by the approximation and their 1-neighbours.

4.2. Photographic Images

We start by considering a 183 × 183 image representing a flower (Image retrieved on MathWorks®), see Figure 6. The use of the Canny edge detection algorithm and the subsequent study of the tangent vector of the profile allows us to identify all cusps, as shown in Figure 6b. The set of edge points is then split into 13 local tubular neighborhoods, see Figure 6c. The obtained approximation, superimposed to the original image, is displayed in Figure 6d. Given that plotting the local tubular neighborhoods is not particularly informative, we just report their number for the remaining examples in Section 4.5.

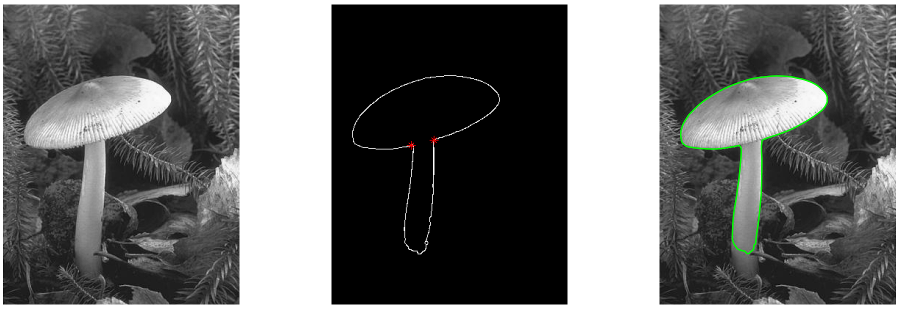


(**a**) original image (**b**) edge points and cusps

(**c**) local tubular neighborhoods (**d**) approximation

Figure 6. Recognition and approximation of a daffodil. The input image (**a**) is pre-processed to obtain the edge points and detect the cusps (**b**). The point cloud is then split into local segments, see (**c**). The approximated curve, superimposed to the original image, is given in (**d**).

Figure 7 shows a (not surprising) limitation of the method. The edge detection algorithm, applied to the 481 × 321 image displayed in Figure 7a, is not able to reveal the whole stalk of the mushroom, see Figure 7b: this results in an accurate approximation of the edge points, as displayed in Figure 7c which is nonetheless missing part of the original object. The image for this test was selected from the Berkeley Segmentation Dataset and Benchmark [32].

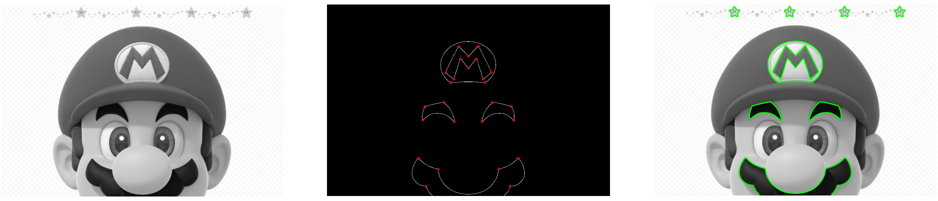


(**a**) original image (**b**) edge points and cusps (**c**) approximation

Figure 7. Recognition and approximation of a mushroom. From the original image (**a**), we extract (**b**) and approximate the edge points (**c**).

4.3. Animation Images

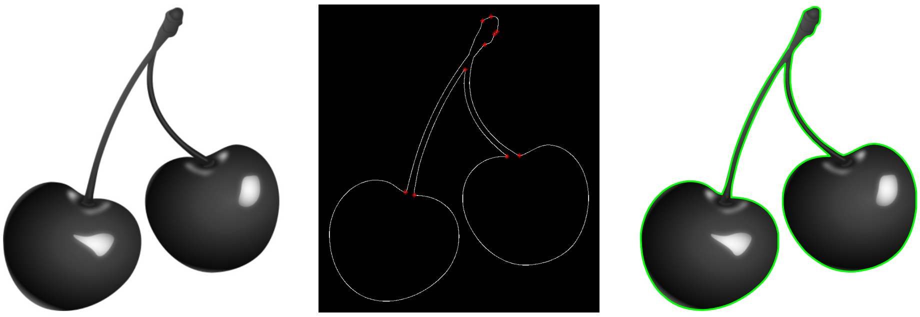
A composite example is exhibited in Figure 8. In Figure 8a, the input 984 × 600 image is shown. In Figure 8b, some sets of edge points and their cusps are displayed. Different closed and open profiles are approximated and provided in Figure 8c: the circle logo and the letter M therein, the eyebrows and the moustache. Note that all profiles have cusps.



(**a**) original image (**b**) edge points and cusps (**c**) approximations

Figure 8. Recognition and approximation of different profiles in Super Mario. From the original image (**a**), we extract (**b**) and approximate the circle logo and the letter M therein, the eyebrows and the moustache (**c**).

A single but more complex closed profile is provided in the 734 × 670 image from Figure 9a, obtained from the dataset used in [33]. The set of edge points and the detected cusps are given in Figure 9b, while the final approximation is shown in Figure 9c.

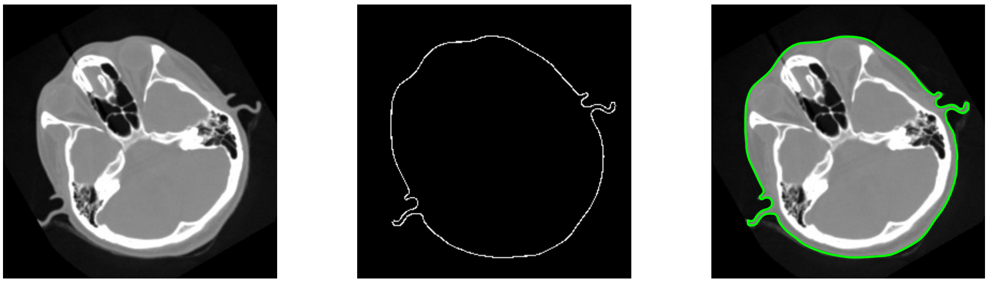


(**a**) original image (**b**) edge points (**c**) approximation

Figure 9. Recognition and approximation of a cherry profile. From the original image (**a**), we extract the edge points and identify cusps (**b**), and produce the final approximation given in (**c**).

4.4. Medical Images

Figure 10 presents the study undertaken on the 256 × 256 CT image of a head (Image retrieved on https://vtk.org, accessed on 1 September 2021). For the external profile in Figure 10a, the set of edge points Figure 10b does not exhibit any cusps. The approximation, superimposed to the original image in Figure 10c, shows an overall satisfying result.



(**a**) original image (**b**) edge points (**c**) approximation

Figure 10. Recognition and approximation of the external profile of a human head. The original image (**a**) is treated to obtain the edge points (**b**), which do not contain cusps in this case; the approximation of the external profile, superimposed to the image, is given in (**c**).

We then consider a coronal slice obtained from the MedPix database (MedPix database, https://medpix.nlm.nih.gov/, accessed on 1 September 2021), see Figure 11. For our testing, we focus our attention on the lumbar spine, whose edge points are shown in Figure 11b and approximated in Figure 11c.

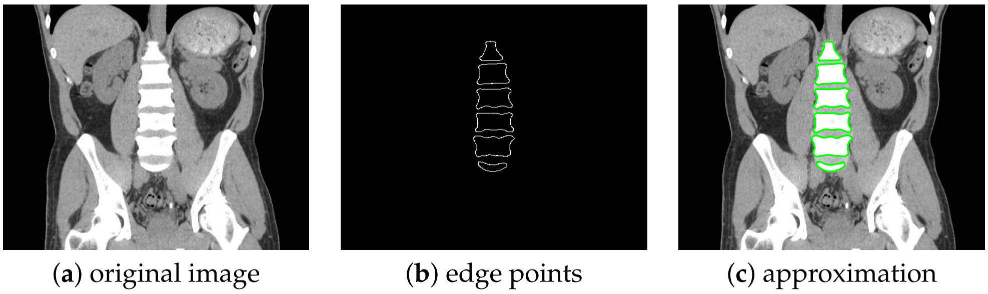


Figure 11. Recognition and approximation of different profiles from a human lumbar spine. The original image (**a**) is treated to obtain the edge points (**b**). The six approximations are given in (**c**).

For the last simulation, we consider a 512 × 512 axial X-ray CT slice of a human lumbar vertebra, see Figure 12. The image has been obtained by the repository [34]. In Figure 12b we provide the three sets of edge points chosen to be approximated, and corresponding to the external profile of the body section; the vertebra, where it is possible to distinguish body, transverse process and spinous process; the abdominal aorta. In Figure 12c we show the superimposition of three distinct approximations to the original image.

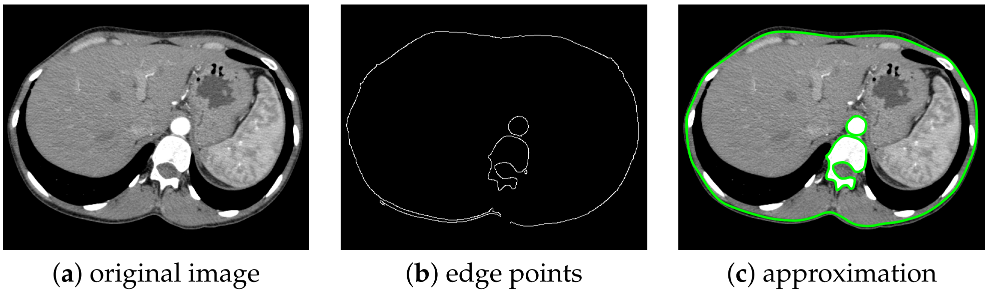


Figure 12. Recognition and approximation of different profiles in a CT slice. The original X-ray CT axial image (**a**) is processed and three sets of edge points are selected (**b**), corresponding to the body profile, the lumbar vertebra and the aorta; the three approximations obtained are then superimposed in (**c**). Note the presence of outliers in (**b**).

4.5. Quantitative Evaluation and Discussions

The indicators computed on each profile of each image are provided in Tables 1–3. When considering the RMSE, MAE and d_{dHaus}, all approximations show a quite satisfying performance. In particular, having normalized the directed Hausdorff distance, we can conclude that the error is always below 3%, with the highest values coming from the lumbar spine. Only a few profiles have a Jaccard index lower than 95%; the lowest possible is that of the body in Figure 12 (61.41% ca.). However, one should consider that this indicator is not reliable when applied to point sets containing a high number of outliers, which is the case for this specific profile, see Figure 13b: the local approximation is able to capture the underlying trend of the data and preserve the monotonicity, but due to the presence of an excessive number of erroneous points, this indicator is not fully representative of the situation. On the contrary, the edge points from Figure 9 are noisy but contain a limited number of outliers; an example of a small local tubular neighborhood and its approximation is given in Figure 13a.

Table 1. Indicators for the examples in Figures 6 and 7. For each profile, we report: the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), the normalized directed Hausdorff distance d_{dHaus} and the Jaccard index J. Here, fit1, fit2 and fit3 stand, respectively, for quadratic polynomial, piecewise cubic and smoothing spline fitting. For each figure and each indicator, the best performance is reported in bold.

Figure	Profile	Method	RMSE	MAE	d_{dHaus}	J
Figure 6	-	us	**0.001918**	**0.000007**	**0.005746**	**0.979417**
		fit1	0.002328	0.000009	0.006317	0.920266
		fit2	0.006170	0.000010	0.007131	0.923588
		fit3	0.003849	0.000009	0.006505	0.966777
Figure 7	-	us	**0.001799**	**0.000005**	**0.003917**	0.951165
		fit1	0.001944	**0.000005**	0.004911	0.931188
		fit2	0.022073	0.000006	0.004542	0.996670
		fit3	0.016233	0.000007	0.004117	**0.997780**

Table 2. Indicators for the examples in Figures 8 and 9. For each profile we report: the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), the normalized directed Hausdorff distance d_{dHaus} and the Jaccard index J. Here, fit1, fit2 and fit3 stand, respectively, for quadratic polynomial, piecewise cubic and smoothing spline fitting. For each figure and each indicator, the best performance is reported in bold.

Figure	Profile	Method	RMSE	MAE	d_{dHaus}	J
Figure 8	left eyebrow	us	**0.002513**	**0.000011**	**0.008158**	**0.983471**
		fit1	0.005680	0.000022	0.013121	0.945946
		fit2	0.027413	0.000040	0.021134	0.918919
		fit3	0.008262	0.000026	0.015892	0.931831
	right eyebrow	us	**0.002535**	**0.000011**	**0.003136**	**0.984000**
		fit1	0.007125	0.000028	0.005068	0.945946
		fit2	0.133256	0.000101	0.013090	0.972973
		fit3	0.017209	0.000039	0.004811	0.972973
	letter M	us	**0.001805**	**0.000006**	**0.002263**	0.973648
		fit1	0.001873	0.000006	0.003513	0.986486
		fit2	0.004236	0.000010	0.006810	0.959459
		fit3	0.002108	0.000007	0.003962	**1.000000**
	oval	us	**0.001593**	**0.000006**	**0.002080**	**1.000000**
		fit1	0.006431	0.000017	0.003587	**1.000000**
		fit2	0.008842	0.000022	0.004210	**1.000000**
		fit3	0.005516	0.000016	0.003233	**1.000000**
	moustache	us	**0.001507**	**0.000005**	**0.005005**	**0.991398**
		fit1	0.010994	0.000020	0.010215	0.905263
		fit2	0.023154	0.000022	0.023991	0.989474
		fit3	0.006152	0.000015	0.010013	0.989474
Figure 9	-	us	**0.000773**	**0.000001**	**0.001092**	**0.976051**
		fit1	0.002365	0.000002	0.001890	0.873673
		fit2	0.068072	0.000004	0.003109	0.957537
		fit3	0.001114	**0.000001**	0.001580	0.973503

Table 3. Indicators for the examples in Figures 10–12. For each profile we report: the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), the normalized directed Hausdorff distance d_{dHaus} and the Jaccard index J. Here, fit1, fit2 and fit3 stand, respectively, for quadratic polynomial, piecewise cubic and smoothing spline fitting. For each figure and each indicator, the best performance is reported in bold.

Figure	Profile	Method	RMSE	MAE	d_{dHaus}	J
Figure 10	-	us	**0.001682**	**0.000005**	**0.003380**	**0.961276**
		fit1	0.008141	0.000023	0.004311	0.933333
		fit2	0.010565	0.000022	0.004200	0.933333
		fit3	0.008312	0.000022	0.003399	0.933333
Figure 11	1st	us	**0.004673**	**0.000030**	**0.006878**	**0.955224**
		fit1	0.016557	0.000129	0.009897	0.800000
		fit2	0.009030	0.000057	0.008621	0.931818
		fit3	0.006711	0.000047	0.008533	0.909091
	2nd	us	**0.002620**	**0.000012**	**0.008201**	**0.988701**
		fit1	0.004431	0.000025	0.009130	0.864407
		fit2	0.031993	0.000056	0.009287	0.881356
		fit3	0.004191	0.000021	0.008666	0.915254
	3rd	us	**0.002664**	**0.000014**	**0.000268**	**0.984456**
		fit1	0.006681	0.000039	0.008065	0.777778
		fit2	0.130043	0.000166	0.011406	0.888889
		fit3	0.004274	0.000022	0.009983	0.920635
	4th	us	**0.003160**	**0.000016**	**0.018939**	**0.960784**
		fit1	0.006387	0.000040	0.020379	0.794118
		fit2	0.072073	0.000163	0.022738	0.897059
		fit3	0.017184	0.000036	0.021314	0.941176
	5th	us	**0.002852**	**0.000015**	**0.028267**	**0.986047**
		fit1	0.005998	0.000035	0.046147	0.763889
		fit2	0.067032	0.000121	0.045282	0.916667
		fit3	0.006559	0.000030	0.039010	0.930556
	6th	us	**0.005309**	**0.000032**	**0.018585**	**0.957627**
		fit1	0.008948	0.000065	0.042655	0.794872
		fit2	0.154618	0.000278	0.058112	0.846154
		fit3	0.005987	0.000045	0.039143	0.897436
Figure 12	body	us	**0.004904**	**0.000009**	**0.001993**	0.614137
		fit1	0.023478	0.000046	0.003989	0.529583
		fit2	0.010881	0.000016	0.002923	**0.972222**
		fit3	0.005590	0.000011	0.002711	0.854167
	aorta	us.	**0.002837**	**0.000015**	**0.004369**	0.939130
		fit1	0.004833	0.000019	0.010610	0.869565
		fit2	0.080569	0.000068	0.012113	0.920290
		fit3	0.003674	0.000016	0.010002	**0.971014**
	vertebra	us	**0.002309**	**0.000009**	**0.010610**	**0.980344**
		fit1	0.008043	0.000063	0.022318	0.900000
		fit2	0.029186	0.000136	0.029725	0.900000
		fit3	0.009149	0.000072	0.021721	0.950000

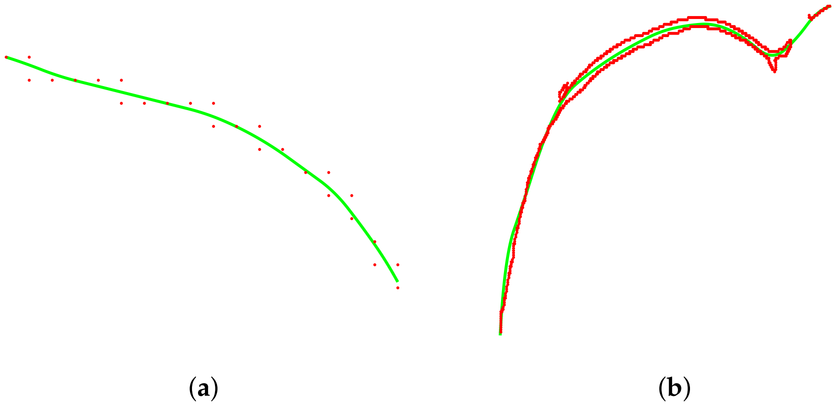


Figure 13. Example of local tubular neighborhoods (in red) and their approximations (in green): (**a**,**b**) are taken from Figures 9 and 13, respectively.

All in all, it might be concluded that the method can successfully reconstruct planar curvilinear profiles, but its success is inevitably linked to the quality of the segmentation: as observed in the qualitative discussion of Figure 7, the Canny edge detection algorithm was unable to identify the stalk of the mushroom in its entirety.

We compared the generalization capability of wQISAs with that of other fitting methods implemented in MATLAB: quadratic polynomial, piecewise cubic and smoothing spline fitting. When considering RMSE, MAE and d_{dHaus}, our methods outperforms its competitors; however, in a few cases the results are comparable. In terms of the Jaccard index, the results are more fluid: as previously observed, this is partly justified by the presence of outliers in some point clouds, which makes this indicator not always trustworthy.

To provide a complete analysis of the experimental results, Table 4 provides the number of tubular neighborhoods, the number of edge points and the execution times for all the examples shown in the paper.

Table 4. Number of tubular neighbohoods, number of edge points and execution times (in seconds) for the profiles presented in Figures 6–12.

Figure	Profile	# Tubular Neighborhoods	# Edge Points	Execution Times (s)
Figure 6	-	13	583	0.386935
Figure 7	-	11	901	0.700913
Figure 8	left eyebrow	5	363	0.530476
	right eyebrow	4	375	0.539895
	letter M	10	721	0.758750
	oval	7	649	0.833938
	moustache	9	930	1.038803
Figure 9	-	30	3758	6.910031
Figure 10	-	24	878	1.541078
Figure 11	1st	4	134	0.211553
	2nd	7	177	0.238248
	3rd	6	193	0.241169
	4th	4	204	0.259886
	5th	10	215	0.252587
	6th	5	118	0.199001
Figure 12	body	4	1726	2.786343
	aorta	5	422	0.426888
	vertebra	13	42	0.195435

5. Conclusions

In this work, we have proposed a novel method to recognize and approximate closed and open 1-manifolds in digital images. Based on the Weighted Quasi-Interpolant Spline Approximation family, our approach can provide a sequence of parametrizations connected G^1 continuously, except in correspondence with the identified cusps, where we impose C^0 continuity. For each local approximation, the number of knots in the knot vectors and the free parameter k are tuned by cross-validation, to limit under- and overfitting. Being based on the quasi-interpolation paradigm, the method is computationally efficient because it does not involve least-square approximation routines. Experimental results on a set of real digital images show a good performance on real data.

As a future development of the method, we intend to extend it to the case of space curves, that is, to non planar profiles embedded into the Euclidean space, and to curves with non-manifold configurations, such as knots and self-intersections. A typical solution for dealing with complicated image borders is to consider arcs and line segments at the same time [1]; in our case, our method can be iterated on the single 1-manifold arcs by decomposing a general profile into segments.

Author Contributions: Conceptualization, A.R. and S.B.; methodology, A.R.; software, A.R.; validation, A.R.; formal analysis, A.R.; investigation, A.R.; resources, A.R. and S.B.; data curation, A.R.; writing—original draft preparation, A.R.; writing—review and editing, A.R. and S.B.; visualization, A.R.; supervision, S.B.; funding acquisition, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been made possible by the contribution granted by CNR in the framework of the Scientific Cooperation Agreement CNR-CNRST (Morocco), under the bilateral agreement n. SAC.AD002.014.032, and by the project POR FSE, Programma Operativo Regione Liguria, 2014–2020 grant number RLOF18ASSRIC/68/1.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Data available online in publicly accessible repositories, see Section 4.

Acknowledgments: The authors thank Bianca Falcidieno and Michela Spagnuolo for the fruitful discussions.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Nguyen, T.P.; Debled-Rennesson, I. Decomposition of a Curve into Arcs and Line Segments Based on Dominant Point Detection. In *Image Analysis*; Heyden, A., Kahl, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 794–805.
2. Cheney, E.W. *Approximation Theory, Wavelets and Applications*; NATO Science Series; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1995; Volume 454, pp. 37–45.
3. Sablonnière, P. Univariate spline quasi-interpolants and applications to numerical analysis. In *Rendiconti del Seminario Matematico*; Università Degli Studi di Torino/Politecnico di Torino: Turin, Italy, 2005; Volume 63, pp. 211–222.
4. Bernstein, S.N. Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités. *Commun. Société Mathématique Kharkow* **1912**, *13*, 1–2.
5. Marsden, M.; Schoenberg, I.J. *On Variation Diminishing Spline Approximation Methods*; de Boor, C., Ed.; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1988; Volume 2, pp. 247–268. [CrossRef]
6. Lyche, T.; Mørken, K. *Spline Methods Draft*; University of Oslo: Oslo, Norway, 2011.
7. Barrera, D.; Ibáñez, M.J.; Sablonnière, P.; Sbibih, D. Near-Best Univariate Spline Discrete Quasi-Interpolants on Nonuniform Partitions. *Constr. Approx.* **2008**, *28*, 237–251. [CrossRef]
8. Remogna, S.; Sablonnière, P. On trivariate blending sums of univariate and bivariate quadratic spline quasi-interpolants on bounded domains. *Comput. Aided Geom. Des.* **2011**, *28*, 89–101. [CrossRef]
9. Ibáñez, M.J.; Barrera, D.; Maldonado, D.; Yáñez, R.; Roldán, J.B. Non-Uniform Spline Quasi-Interpolation to Extract the Series Resistance in Resistive Switching Memristors for Compact Modeling Purposes. *Mathematics* **2021**, *9*, 2159. [CrossRef]

10. Remogna, S. Bivariate C2 cubic spline quasi-interpolants on uniform Powell–Sabin triangulations of a rectangular domain. *Adv. Comput. Math.* **2012**, *36*, 39–65. [CrossRef]
11. Sbibih, D.; Serghini, A.; Tijini, A.; Zidna, A. Superconvergent C^1 cubic spline quasi-interpolants on Powell-Sabin partitions. *BIT Numer. Math.* **2015**, *55*, 797–821. [CrossRef]
12. Sbibih, D.; Serghini, A.; Tijini, A. Superconvergent quadratic spline quasi-interpolants on Powell–Sabin partitions. *Appl. Numer. Math.* **2015**, *87*, 74–86. [CrossRef]
13. Eddargani, S.; Ibáñez, M.J.; Lamnii, A.; Lamnii, M.; Barrera, D. Quasi-Interpolation in a Space of C2 Sextic Splines over Powell–Sabin Triangulations. *Mathematics* **2021**, *9*, 2276. [CrossRef]
14. Sbibih, D.; Serghini, A.; Tijini, A. Superconvergent local quasi-interpolants based on special multivariate quadratic spline space over a refined quadrangulation. *Appl. Math. Comput.* **2015**, *250*, 145–156. [CrossRef]
15. Barrera, D.; Ibáñez, M.; Remogna, S. On the Construction of Trivariate Near-Best Quasi-Interpolants Based on C2 Quartic Splines on Type-6 Tetrahedral Partitions. *J. Comput. Appl. Math.* **2017**, *311*, 252–261. [CrossRef]
16. Raffo, A.; Biasotti, S. Weighted quasi-interpolant spline approximations: Properties and applications. *Numer. Algorithms* **2021**, *87*, 819–847. [CrossRef]
17. Raffo, A.; Biasotti, S. Data-driven quasi-interpolant spline surfaces for point cloud approximation. *Comput. Graph.* **2020**, *89*, 144–155. [CrossRef]
18. Barsky, B.A.; DeRose, A.D. *Geometric Continuity of Parametric Curves*; Technical Report UCB/CSD-84-205; EECS Department, University of California: Berkeley, CA, USA, 1984.
19. DeRose, T.D.; Barsky, B.A. An Intuitive Approach to Geometric Continuity for Parametric Curves and Surfaces. In *Computer-Generated Images*; Magnenat-Thalmann, N., Thalmann, D., Eds.; Springer: Tokyo, Japan, 1985; pp. 159–175.
20. Said Mad Zain, S.A.A.A.; Misro, M.Y.; Miura, K.T. Generalized Fractional Bézier Curve with Shape Parameters. *Mathematics* **2021**, *9*, 2141. [CrossRef]
21. Farin, G. *Curves and Surfaces for CAGD: A Practical Guide*, 5th ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001.
22. Mizutani, K.; Kawaguchi, K. Curve approximation by G1 arc splines with a limited number of types of curvature and length. *Comput. Aided Geom. Des.* **2021**, *90*, 102036. [CrossRef]
23. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]
24. Heath, M.; Sarkar, S.; Sanocki, T.; Bowyer, K. Comparison of Edge Detectors: A Methodology and Initial Study. *Comput. Vis. Image Underst.* **1998**, *69*, 38–54. [CrossRef]
25. Williams, I.; Bowring, N.; Svoboda, D. A performance evaluation of statistical tests for edge detection in textured images. *Comput. Vis. Image Underst.* **2014**, *122*, 115–130. [CrossRef]
26. Conti, C.; Romani, L.; Schenone, D. Semi-automatic spline fitting of planar curvilinear profiles in digital images using the Hough transform. *Pattern Recognit.* **2018**, *74*, 64–76. [CrossRef]
27. Abate, M.; Francesca, T. *Curves and Surfaces*; Springer: Mailand, Italy, 2012. [CrossRef]
28. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*, 40–79. [CrossRef]
29. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer Series in Statistics; Springer: New York, NY, USA, 2001.
30. Giraudot, S.; Cohen-Steiner, D.; Alliez, P. Noise-Adaptive Shape Reconstruction from Raw Point Sets. *Comput. Graph. Forum* **2013**, *32*, 229–238. [CrossRef]
31. Taha, A.A.; Hanbury, A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med. Imaging* **2015**, *15*, 29. [CrossRef] [PubMed]
32. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In Proceedings of the Proceedings Eighth IEEE International Conference on Computer Vision—ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 416–423.
33. Hu, Y.; Schneider, T.; Gao, X.; Zhou, Q.; Jacobson, A.; Zorin, D.; Panozzo, D. TriWild: Robust Triangulation with Curve Constraints. *ACM Trans. Graph.* **2019**, *38*, 52:1–52:15. [CrossRef]
34. Hazarika, H.J.; Handique, A.; Ravikumar, S. DICOM-based medical image repository using DSpace. *Collect. Curation* **2020**, *39*, 105–115. [CrossRef]