Discrete Polynomial Curve Fitting Guaranteeing Inclusion-Wise Maximality of Inlier Set

Fumiki Sekiya[1(✉)] and Akihiro Sugimoto[2]

[1] Department of Informatics, SOKENDAI (The Graduate University for Advanced Studies), Tokyo, Japan
sekiya@nii.ac.jp

[2] National Institute of Informatics, Tokyo, Japan
sugimoto@nii.ac.jp

Abstract. This paper deals with the problem of fitting a discrete polynomial curve to 2D noisy data. We use a discrete polynomial curve model achieving connectivity in the discrete space. We formulate the fitting as the problem to find parameters of this model maximizing the number of inliers i.e., data points contained in the discrete polynomial curve. We propose a method guaranteeing inclusion-wise maximality of its obtained inlier set.

1 Introduction

Curve fitting to noisy data (i.e., containing outliers) is an essential task in many applications such as object recognition, image segmentation and shape approximation. Continuous curve models have been used for fitting in most cases even though data dealt with in a computer are discrete.

The method most commonly used for continuous curve fitting in the presence of noise is RANdom SAmple Consensus (RANSAC) [1], which uses random sampling to estimate model parameters, and then choose the ones having the largest number of inliers, i.e., data points explained by the parameters. For its robustness and simplicity, RANSAC is used in a wide range of problems in computer vision. The main drawback of RANSAC (and most of its variants) is however that it does not guarantee any deterministic properties on its output. It also requires an empirical error threshold to define an inlier, which affects the output. Another popular approach for the task is to use the Hough transform [2,3], which allows to find model parameters consistent with many data points in the space of the model parameters. This method however requires to manually set the resolution to discretize the parameter space, which affects the output.

As long as a continuous curve model is fitted to discrete data, an error threshold is required to determine if a data point is explained by the model. By using a discrete curve model, on the other hand, we can define an inlier without an empirical error threshold. A discrete curve model is defined as a set of discrete points to represent a discretized curve. For curve discretization, it has been considered to be important to preserve the topological properties (e.g., connectivity)

C.-S. Chen et al. (Eds.): ACCV 2016 Workshops, Part II, LNCS 10117, pp. 477–492, 2017.
DOI: 10.1007/978-3-319-54427-4_35

of an original curve [4–9]. For example, a Jordan curve (i.e., simple closed curve) in a 2D image allows to partition the image into two connected regions. Such a property is useful in computer graphics, computer vision and image processing (see [10]), and therefore should be preserved in discretization. Based on the idea, several discrete curve models have been developed [11–16] to achieve some consistent topological properties in the discrete space. It is therefore preferable to use such a model for curve fitting.

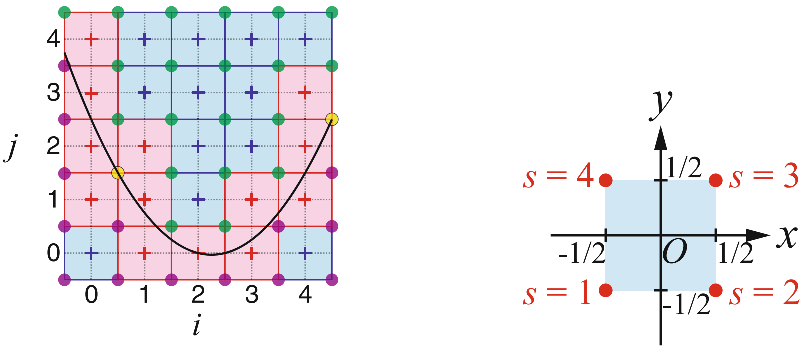
Curve fitting to noisy data in 2D has been studied for discrete lines [17–20], discrete circles [21–24] and discrete polynomial curves [25]. For lines and circles, models having connectivity have been used. For polynomial curves, on the other hand, only one type of discrete polynomial curve model without guaranteeing any topological property has been used. Discrete polynomial curve fitting therefore has yet to be studied for a model having consistent topological properties.

In this paper we deal with the problem of fitting a discrete polynomial curve to 2D noisy data. We use the discrete curve model introduced by Toutant et al. [16] to define our discrete polynomial curve. This is because this model guarantees connectivity in the discrete space [26], and is closely related to the morphological discretization [27–29]. To be precise, this model corresponds to the morphological discretization with a structuring element called the k-adjacency flake [15], which is defined for $k = 0, 1$ in 2D and achieves different topological properties depending on k. Note that in this paper we limit ourselves to define our discrete polynomial curve only for $k = 0$. We formulate our problem as to find parameters of this model that maximize the number of inliers, where an inlier is defined as a point contained in the discrete polynomial curve.

We propose for this problem a method guaranteeing inclusion-wise maximality of its obtained inlier set (i.e., there exists no larger inlier set in the sense of set inclusion). Note that an inclusion-wise maximal inlier set does not necessarily have the maximum cardinality. Our method runs in the space of parameters (coefficients) of the discrete polynomial curve model. In the parameter space a discrete polynomial curve is represented by a point, while a data point or a set of data points gives a feasible region shaped like a polytope where any discrete polynomial curve represented by a point in the region contains the data point(s). Given any initial inlier set, the method adds new data points to the inlier set one by one with tracking its feasible region in the parameter space, until inclusion-wise maximality is achieved. The feasible region is generally an infinite set so that it is impossible to store all its points in a computer. We solve this problem by focusing only on a finite number of points corresponding to the notion of the vertices of a polytope. Our method thus does not require any discretization of the parameter space, which is a major difference from the Hough transform.

2 Problem Formulation

A continuous polynomial curve of degree d in the xy-plane is represented by $y - \sum_{l=0}^{d} a_l x^l = 0$ with coefficients $a_0, \ldots, a_{d-1} \in \mathbb{R}$ and $a_d \in \mathbb{R} \setminus \{0\}$. Toutant et al. [16] introduced its discretized form in $\mathbb{Z}^2$, i.e., a *discrete polynomial curve* $D(a_0, \ldots a_d)$ by



(a) $D\left(a_0, \ldots, a_d\right)$. (b) $\left(x_s, y_s\right)$.

Fig. 1. Discrete polynomial curve. (a) shows $D\left(a_0, \ldots, a_d\right)$ (red integer points) for $d = 2$ and $\left(a_0, a_1, a_2\right) = (2.5, -2.25, 0.5)$ with its continuous counterpart (depicted in black). For $(i, j) \in \mathbb{Z}^2$ and $s \in \{1, \ldots, 4\}$, $\left(i + x_s, j + y_s\right)$ is depicted in green, purple or yellow: green if $\left(j + y_s\right) - \sum_{l=0}^{d} a_l \left(i + x_s\right)^l > 0$; purple if $\left(j + y_s\right) - \sum_{l=0}^{d} a_l \left(i + x_s\right)^l < 0$; yellow if $\left(j + y_s\right) - \sum_{l=0}^{d} a_l \left(i + x_s\right)^l = 0$. $(i, j) \in \mathbb{Z}^2$ is in $D\left(a_0, \ldots, a_d\right)$ if $\left\{\left(i + x_s, j + y_s\right) \,\middle|\, s = 1, \ldots, 4\right\}$ contains green and purple points, or an yellow point. In (b), $\left(x_s, y_s\right)$ is depicted in red for $s = 1, \ldots, 4$. (Color figure online)

$$D\left(a_0, \ldots, a_d\right) = \left\{ (i, j) \in \mathbb{Z}^2 \;\middle|\; \begin{array}{c} \min\limits_{s \in \{1, \ldots, 4\}} \left[\left(j + y_s\right) - \sum\limits_{l=0}^{d} a_l \left(i + x_s\right)^l\right] \\ \leq 0 \leq \\ \max\limits_{s \in \{1, \ldots, 4\}} \left[\left(j + y_s\right) - \sum\limits_{l=0}^{d} a_l \left(i + x_s\right)^l\right] \end{array} \right\}, \quad (1)$$

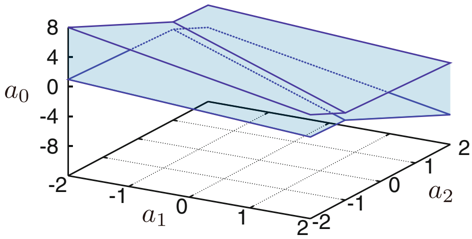
where $\left(x_1, y_1\right) = \left(-\frac{1}{2}, -\frac{1}{2}\right)$, $\left(x_2, y_2\right) = \left(\frac{1}{2}, -\frac{1}{2}\right)$, $\left(x_3, y_3\right) = \left(\frac{1}{2}, \frac{1}{2}\right)$ and $\left(x_4, y_4\right) = \left(-\frac{1}{2}, \frac{1}{2}\right)$. See Fig. 1 for an illustration of $D\left(a_0, \ldots, a_d\right)$.

Let $P = \left\{\left(i_p, j_p\right) \in \mathbb{Z}^2 \,\middle|\, p = 1, \ldots, n\right\}$ be a finite set (i.e., $n < \infty$) of integer points (data). For a discrete polynomial curve $D\left(a_0, \ldots, a_d\right)$, a point $\left(i_p, j_p\right)$ $(p = 1, \ldots, n)$ is called an *inlier* if $\left(i_p, j_p\right) \in D\left(a_0, \ldots, a_d\right)$, while otherwise it is called an *outlier*. Our goal is to find $D\left(a_0, \ldots, a_d\right)$ that maximizes the number of inliers for given data P and a degree d, where we permit $a_d = 0$ so that discrete polynomial curves of degree less than d are covered as well.

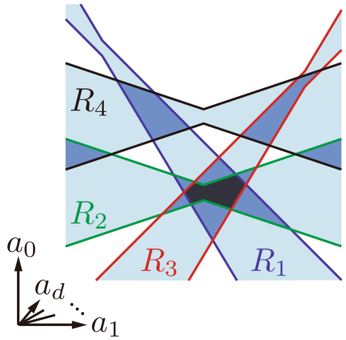
When d is fixed, $D\left(a_0, \ldots, a_d\right)$ is determined only by $a_0, \ldots, a_d$. We therefore consider the problem in the *parameter space* $\left\{\left(a_0, \ldots, a_d\right)\right\} = \mathbb{R}^{d+1}$, instead of the *data space* $\mathbb{Z}^2$ where P resides. A discrete polynomial curve in the data space is represented as a point in the parameter space. A data point in P, on the other hand, is represented as a region in the parameter space, which is defined as follows.

For $p = 1, \ldots, n$, we define the *feasible region* R_p for the pth data $\left(i_p, j_p\right)$ by

$$R_p = \left\{ \left(a_0, \ldots, a_d\right) \in \mathbb{R}^{d+1} \;\middle|\; \begin{array}{c} \min\limits_{s \in \{1, \ldots, 4\}} h_{(p,s)}\left(a_0, \ldots, a_d\right) \\ \leq 0 \leq \\ \max\limits_{s \in \{1, \ldots, 4\}} h_{(p,s)}\left(a_0, \ldots, a_d\right) \end{array} \right\}, \quad (2)$$



(a) R_p.



(b) Parameter space.

Fig. 2. Feasible regions in the parameter space. (a) shows R_p for $d = 2$ and $(i_p, j_p) = (1, 0)$. Each point in the data space is represented in the parameter space by an unbounded concave polytope like this. (b) shows intersections among the feasible regions for four individual data points, which are indexed from 1 to 4. $(a_0, \ldots, a_d)$ in a darker region has a larger number of inliers in the data space.

where

$$h_{(p,s)}\left(a_0, \ldots, a_d\right) = \left(j_p + y_s\right) - \sum_{l=0}^{d} \left(i_p + x_s\right)^l a_l. \tag{3}$$

See Fig. 2(a) for an illustration of R_p. We remark that $(i_p, j_p) \in D(a_0, \ldots, a_d)$ iff $(a_0, \ldots, a_d) \in R_p$.

We also define a feasible region for a set of data. For $\Pi \subset \{1, \ldots, n\}$, we define the *feasible region* R_Π for the set $\left\{(i_p, j_p) \;\middle|\; p \in \Pi\right\}$ of data by $R_\Pi = \bigcap_{p \in \Pi} R_p$, which is also written as

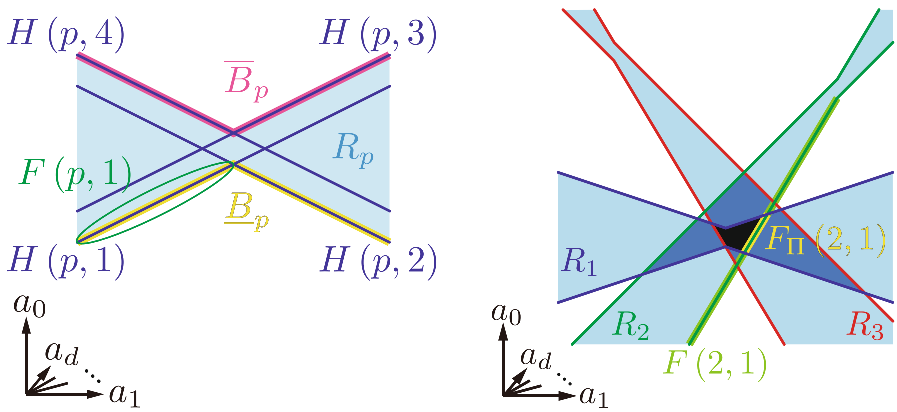
$$R_\Pi = \left\{ \left(a_0, \ldots, a_d\right) \in \mathbb{R}^{d+1} \;\middle|\; \begin{array}{c} \max\limits_{p \in \Pi} \min\limits_{s \in \{1,\ldots,4\}} h_{(p,s)}\left(a_0, \ldots, a_d\right) \\ \leq 0 \leq \\ \min\limits_{p \in \Pi} \max\limits_{s \in \{1,\ldots,4\}} h_{(p,s)}\left(a_0, \ldots, a_d\right) \end{array} \right\}. \tag{4}$$

We remark that R_Π may be bounded or unbounded, convex or concave, and connected or disconnected as can be seen in Fig. 2(b) (e.g., $R_{\{1,2,3\}}$ is bounded and convex, while $R_{\{2,4\}}$ is unbounded and disconnected).

$R_\Pi = \emptyset$ if no $(a_0, \ldots, a_d) \in \mathbb{R}^{d+1}$ satisfies $(i_p, j_p) \in D(a_0, \ldots, a_d)$ for $\forall p \in \Pi$. Our problem is therefore formulated as follows.

Problem 1. Given P and d, find $\Pi \subset \{1, \ldots, n\}$ that has the maximum cardinality providing $R_\Pi \neq \emptyset$, and $(a_0, \ldots, a_d) \in \mathbb{R}^{d+1}$ satisfying $(a_0, \ldots, a_d) \in R_\Pi$ for that Π.

We remark that the solution Π is not necessarily unique. In the example in Fig. 2(b), the data index set Π that we would like to find is $\{1, 2, 3\}$.



(a) $F(p,s) = H(p,s) \cap \left(\underline{B}_p \cup \overline{B}_p\right)$. (b) $F_{\varPi}(p,s) = F(p,s) \cap R_{\varPi}$.

Fig. 3. Concepts to represent a flat part on the surface of a feasible region. In (b), $\varPi = \{1,2,3\}$.

3 Properties of Feasible Regions

3.1 Concepts and Notations

Our approach to find an inclusion-wise maximal inlier set is as follows: starting from an arbitrary $\varPi \subset \{1,\ldots,n\}$ satisfying $R_{\varPi} \neq \emptyset$, we iteratively search $p \in \{1,\ldots,n\} \setminus \varPi$ such that $R_{\varPi \cup \{p\}}$ $(= R_{\varPi} \cap R_p) \neq \emptyset$ and add it to $\varPi$, where every time we update $\varPi$ ($\varPi := \varPi \cup \{p\}$) we compute its corresponding $R_{\varPi}$. By repeating this procedure until there is no such p, we can ensure an inclusion-wise maximal inlier set. It is however impossible to store all points in $R_{\varPi}$ in a computer, since $R_{\varPi}$ is generally an infinite set when $R_{\varPi} \neq \emptyset$. We therefore focus only a finite number of points in $R_{\varPi}$ that correspond to the notion of the vertices of a polytope (Fig. 4). As a vertex of a polytope is defined as an intersection point of flat parts on the surface of the polytope called facets, we first need a notion for $R_{\varPi}$ corresponding to a facet of a polytope.

Let $p \in \{1,\ldots,n\}$ be any data index. For $s = 1,\ldots,4$, we first define $H(p,s) = \left\{(a_0,\ldots,a_d) \in \mathbb{R}^{d+1} \mid h_{(p,s)}(a_0,\ldots,a_d) = 0\right\}$. $H(p,s)$ is a hyperplane included in R_p (Fig. 3(a)), which determines a flat part on the surface of R_p. To represent the surface of R_p, we then define $\underline{B}_p = \{(a_0,\ldots,a_d) \in \mathbb{R}^{d+1} \mid \min_{s\in\{1,\ldots,4\}} h_{(p,s)}(a_0,\ldots,a_d) = 0\}$ and $\overline{B}_p = \{(a_0,\ldots,a_d) \in \mathbb{R}^{d+1} \mid \max_{s\in\{1,\ldots,4\}} h_{(p,s)}(a_0,\ldots,a_d) = 0\}$. $\underline{B}_p$ and $\overline{B}_p$ are the "lower" and "upper" boundaries (with a_0 considered as the height) of R_p (cf. Fig. 3(a)). We remark that $\underline{B}_p$ is determined only by $s = 1,2$ while $\overline{B}_p$ is determined only by $s = 3,4$, and that $\underline{B}_p \cap \overline{B}_p = \emptyset$. The flat part of $\underline{B}_p \cup \overline{B}_p$ (i.e., facet of R_p) determined by $H(p,s)$, for $s = 1,\ldots,4$, is then represented by $F(p,s) = H(p,s) \cap \left(\underline{B}_p \cup \overline{B}_p\right)$ (Fig. 3(a)). Note that $\underline{B}_p \cup \overline{B}_p = \bigcup_{s\in\{1,\ldots,4\}} F(p,s)$.

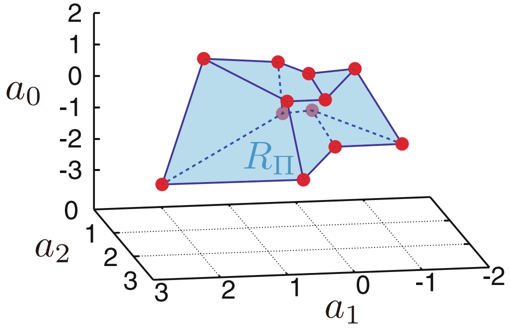


Fig. 4. C_Π for $d = 2$ and $\Pi = \{1, 2, 3\}$ with $(i_1, j_1) = (-2, 3)$, $(i_2, j_2) = (0, 0)$, $(i_3, j_3) = (2, 5)$. Points in C_Π are depicted in red. (Color figure online)

Let $\Pi \subset \{1, \ldots, n\}$ be any set of data indices. Since $R_\Pi = \bigcap_{p \in \Pi} R_p$, the flat part of the boundary of R_Π (i.e., facet of R_Π) determined by $H(p, s)$, for $(p, s) \in \Pi \times \{1, \ldots, 4\}$, is obtained as a subset of $F(p, s)$. Namely, it is represented by $F_\Pi(p, s) = F(p, s) \cap R_\Pi$. See Fig. 3(b) for an illustration of $F_\Pi(p, s)$. We remark that $F_\Pi(p, s)$ can be empty for some $(p, s) \in \Pi \times \{1, \ldots, 4\}$.

We now define a subset of R_Π corresponding to the vertices of a polytope. In $\mathbb{R}^{d+1}$, $d + 1$ hyperplanes intersect at one point when their normal vectors are linearly independent. We therefore can specify a finite subset of R_Π by enumerating $(a_0, \ldots, a_d) \in \bigcap_{\lambda=1}^{d+1} F_\Pi(p_\lambda, s_\lambda)$ for all $(p_1, s_1), \ldots, (p_{d+1}, s_{d+1}) \in \Pi \times \{1, \ldots, 4\}$ such that $H(p_1, s_1), \ldots, H(p_{d+1}, s_{d+1})$ are linearly independent. Namely, C_Π identifies a subset (i.e., vertices) of R_Π:

$$C_\Pi = \left\{ \begin{array}{r|l} (a_0, \ldots, a_d) & \text{there exist} \\ \in \mathbb{R}^{d+1} & (p_1, s_1), \ldots, (p_{d+1}, s_{d+1}) \in \Pi \times \{1, \ldots, 4\} \\ & \text{such that } (a_0, \ldots, a_d) \in \bigcap_{\lambda=1}^{d+1} F_\Pi(p_\lambda, s_\lambda) \\ & \text{and } H(p_1, s_1), \ldots, H(p_{d+1}, s_{d+1}) \text{ are} \\ & \text{linearly independent} \end{array} \right\}. \tag{5}$$

See Fig. 4 for an illustration of C_Π. We remark that C_Π is a finite set since $\Pi \times \{1, \ldots, 4\}$ has only a finite number of elements. We also define the family of sets of $d + 1$ elements of $\Pi \times \{1, \ldots, 4\}$ determining elements of C_Π:

$$\Psi_\Pi = \left\{ \begin{array}{r|l} \{(p_1, s_1), \ldots, (p_{d+1}, s_{d+1})\} & H(p_1, s_1), \ldots, H(p_{d+1}, s_{d+1}) \\ \subset \Pi \times \{1, \ldots, 4\} & \text{are linearly independent and} \\ & \text{their intersection point is in} \\ & \bigcap_{\lambda=1}^{d+1} F_\Pi(p_\lambda, s_\lambda) \end{array} \right\}. \tag{6}$$

We remark that different sets in Ψ_Π may determine the same element of C_Π.

3.2 Updated Feasible Region by an Additional Inlier

Here we give four properties of R_Π (Theorems 1–4) which are important for enabling the approach described in the beginning of Sect. 3.1, where R_Π is represented by C_Π. We start with the following lemma required to prove three of those properties, which states the condition for m ($m = 1, \ldots, d$) flat parts $F_\Pi(p_1, s_1), \ldots, F_\Pi(p_m, s_m)$ on the surface of R_Π to contribute to determining a point in C_Π. The proof of this lemma is provided in Appendix A.

Lemma 1. *Let $\Pi \subset \{1, \ldots, n\}$ be a data index set such that $R_\Pi \neq \emptyset$ and R_Π is bounded, and let $(p_1, s_1), \ldots, (p_m, s_m)$ be m ($m = 1, \ldots, d$) elements of $\Pi \times \{1, \ldots, 4\}$. There exists a set in Ψ_Π containing $(p_1, s_1), \ldots, (p_m, s_m)$ if (i) $\bigcap_{\lambda=1}^{m} F_\Pi(p_\lambda, s_\lambda) \neq \emptyset$ and (ii) $H(p_1, s_1), \ldots, H(p_m, s_m)$ are linearly independent.*

Since R_Π is represented by C_Π in our approach, it is important that $C_\Pi \neq \emptyset$ whenever $R_\Pi \neq \emptyset$. This can be proven under the condition that R_Π is bounded. Note that $\Psi_\Pi \neq \emptyset$ is equivalent with $C_\Pi \neq \emptyset$.

Theorem 1. *For $\Pi \subset \{1, \ldots, n\}$ such that $R_\Pi \neq \emptyset$ and R_Π is bounded, $\Psi_\Pi \neq \emptyset$.*

Proof. It is obvious that there exists $(p, s) \in \Pi \times \{1, \ldots, 4\}$ satisfying $F_\Pi(p, s) \neq \emptyset$. From Lemma 1, then, there exists a set in Ψ_Π containing (p, s). $\Psi_\Pi \neq \emptyset$, accordingly. $\square$

From Theorem 1, for any $\Pi \subset \{1, \ldots, n\}$ such that R_Π is bounded, we can always obtain a point in R_Π by computing C_Π. There are however $\binom{4|\Pi|}{d+1}$ ways to pick $d+1$ different elements from $\Pi \times \{1, \ldots, 4\}$, so that checking all those combinations to compute C_Π is not practical for Π with a large number of elements. In the following we give a relationship between Ψ_Π and $\Psi_{\Pi \cup \{p\}}$, for $p \in \{1, \ldots, n\} \backslash \Pi$, with which we can reduce the computational cost for obtaining $\Psi_{\Pi \cup \{p\}}$ when we have Ψ_Π.

For $\Pi \subsetneq \{1, \ldots, n\}$ and $p \in \{1, \ldots, n\} \setminus \Pi$, we define $\Phi^1_{\Pi,p}$ to be the family of sets each of whose set is obtained by replacing an element of a set in Ψ_Π with (p, s) for $s \in \{1, \ldots, 4\}$:

$$\Phi^1_{\Pi,p} = \left\{ \begin{array}{r} \{(p_1, s_1), \ldots, (p_d, s_d)\} \\ \cup \{(p, s)\} \end{array} \;\middle|\; \begin{array}{l} \{(p_1, s_1), \ldots, (p_d, s_d)\} \text{ is a subset} \\ \text{of a set in } \Psi_\Pi \text{ and } s = 1, \ldots, 4 \end{array} \right\}. \tag{7}$$

We also define $\Phi^2_{\Pi,p}$ to be the family of sets each of whose set is obtained by replacing two elements of a set in Ψ_Π with $(p, 1)$ and $(p, 2)$, or $(p, 3)$ and $(p, 4)$:

$$\Phi^2_{\Pi,p} = \left\{ \begin{array}{r} \{(p_1, s_1), \ldots, (p_{d-1}, s_{d-1})\} \\ \cup \{(p, s), (p, s')\} \end{array} \;\middle|\; \begin{array}{l} \{(p_1, s_1), \ldots, (p_{d-1}, s_{d-1})\} \text{ is} \\ \text{a subset of a set in } \Psi_\Pi \\ \text{and } (s, s') = (1, 2), (3, 4) \end{array} \right\}. \tag{8}$$

We then have the following relationship among Ψ_Π, $\Psi_{\Pi \cup \{p\}}$, $\Phi^1_{\Pi,p}$ and $\Phi^2_{\Pi,p}$.

Theorem 2. *For Π such that R_Π is bounded, $\Psi_{\Pi \cup \{p\}} \subset \Psi_\Pi \cup \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$.*

Proof. We assume $\Psi_{\Pi \cup \{p\}} \neq \emptyset$; otherwise the statement is obviously true. Let $\psi = \{(p_1, s_1), \ldots, (p_{d+1}, s_{d+1})\}$ be a set in $\Psi_{\Pi \cup \{p\}}$. We show $\psi \in \Psi_\Pi \cup \Phi^1_{\Pi,p} \cup$

$\Phi^2_{\Pi,p}$. We assume that m of $p_1, \ldots, p_{d+1} \in \Pi \cup \{p\}$ are equal to p and the others belong to Π. If ψ contains $(p, \underline{s})$ and $(p, \overline{s})$ for any $\underline{s} \in \{1, 2\}$ and $\overline{s} \in \{3, 4\}$, then $\bigcap_{\lambda=1}^{d+1} F_{\Pi \cup \{p\}}(p_\lambda, s_\lambda) = \emptyset$ from $F(p, \underline{s}) \cap F(p, \overline{s}) = \emptyset$ (recall that $F(p, \underline{s}) \subset \underline{B}_p$, $F(p, \overline{s}) \subset \overline{B}_p$ and $\underline{B}_p \cap \overline{B}_p = \emptyset$), contradicting $\psi \in \Psi_{\Pi \cup \{p\}}$. $m \geq 3$ is therefore impossible, because in that case $(p, \underline{s})$ and $(p, \overline{s})$ are necessarily contained in ψ. We therefore have $m \leq 2$, where when $m = 2$ the two elements of ψ corresponding to p are either $(p, 1)$ and $(p, 2)$, or $(p, 3)$ and $(p, 4)$.

We first consider the case of $m = 0$. We then have $(p_1, s_1), \ldots, (p_{d+1}, s_{d+1}) \in \Pi \times \{1, \ldots, 4\}$. From $\psi \in \Psi_{\Pi \cup \{p\}}$, $\bigcap_{\lambda=1}^{d+1} F_{\Pi \cup \{p\}}(p_\lambda, s_\lambda) \neq \emptyset$ and $H(p_1, s_1), \ldots, H(p_{d+1}, s_{d+1})$ are linearly independent. Since $F_\Pi(p_\lambda, s_\lambda) \supset F_{\Pi \cup \{p\}}(p_\lambda, s_\lambda)$ $(\lambda = 1, \ldots, d+1)$, we have $\bigcap_{\lambda=1}^{d+1} F_\Pi(p_\lambda, s_\lambda) \neq \emptyset$. $\psi \in \Psi_\Pi$, consequently.

We next consider the case of $m = 1, 2$. Without loss of generality, we assume $p_1, \ldots, p_{d+1-m} \in \Pi$. We prove $\psi \in \Phi^m_{\Pi,p}$ by showing that assuming otherwise leads to a contradiction. Namely, we assume that there exists no such set in Ψ_Π that contains $(p_1, s_1), \ldots, (p_{d+1-m}, s_{d+1-m})$. Lemma 1 then suggests that $\bigcap_{\lambda=1}^{d+1-m} F_\Pi(p_\lambda, s_\lambda) = \emptyset$ (which implies $\bigcap_{\lambda=1}^{d+1-m} F_{\Pi \cup \{p\}}(p_\lambda, s_\lambda) = \emptyset$) or $H(p_1, s_1), \ldots, H(p_{d+1-m}, s_{d+1-m})$ are not linearly independent. This contradicts $\psi \in \Psi_{\Pi \cup \{p\}}$. □

With Theorem 2, we do not have to evaluate all the sets of $d+1$ elements of $(\Pi \cup \{p\}) \times \{1, \ldots, 4\}$ to compute $\Psi_{\Pi \cup \{p\}}$, but only those in $\Psi_\Pi \cup \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$. For $\{(p_1, s_1), \ldots, (p_{d+1}, s_{d+1})\} \in \Psi_\Pi$, $\{(p_1, s_1), \ldots, (p_{d+1}, s_{d+1})\} \in \Psi_{\Pi \cup \{p\}}$ is verified as soon as the corresponding $(a_0, \ldots, a_d) \in C_\Pi$ satisfies $(a_0, \ldots, a_d) \in R_p$. This is because that we have $(a_0, \ldots, a_d) \in \bigcap_{\lambda=1}^{d+1} F_\Pi(p_\lambda, s_\lambda)$ from $(a_0, \ldots, a_d) \in C_\Pi$, and $F_\Pi(p_\lambda, s_\lambda) \cap R_p = F_{\Pi \cup \{p\}}(p_\lambda, s_\lambda)$ $(\lambda = 1, \ldots, d+1)$. For $\{(p_1, s_1), \ldots, (p_{d+1}, s_{d+1})\} \in \Phi^m_{\Pi,p}$ $(m = 1, 2)$, on the other hand, we have to check if $\bigcap_{\lambda=1}^{d+1} H(p_\lambda, s_\lambda)$ has the unique element $(a_0, \ldots, a_d)$, and $(a_0, \ldots, a_d) \in \bigcap_{\lambda=1}^{d+1} F_{\Pi \cup \{p\}}(p_\lambda, s_\lambda)$. We remark that here it suffices to evaluate $(a_0, \ldots, a_d) \in \bigcap_{\lambda=1}^{d+1} F_\Pi(p_\lambda, s_\lambda)$ (i.e., we do not have to verify $(a_0, \ldots, a_d) \in R_p$) because $p_\lambda = p$ for some $\lambda \in \{1, \ldots, d+1\}$ so that we have $(a_0, \ldots, a_d) \in H(p, s_\lambda) \subset R_p$. The computational cost for evaluating $(a_0, \ldots, a_d) \in R_\Pi$, which is required for checking if $(a_0, \ldots, a_d) \in F_\Pi(p_\lambda, s_\lambda)$, can be reduced by using the following property of R_Π.

Let $\bigcup \Psi_\Pi$ denote the union of all sets in Ψ_Π. For $\Pi \subset \{1, \ldots, n\}$, then, we define

$$R^*_\Pi = \left\{ (a_0, \ldots, a_d) \in \mathbb{R}^{d+1} \;\middle|\; \begin{array}{c} \max_{p \in \Pi^*} \min_{s \in \Sigma_\Pi(p)} h_{(p,s)}(a_0, \ldots, a_d) \\ \leq 0 \leq \\ \min_{p \in \Pi^*} \max_{s \in \Sigma_\Pi(p)} h_{(p,s)}(a_0, \ldots, a_d) \end{array} \right\}, \tag{9}$$

where

$$\Pi^* = \left\{ p \in \Pi \;\middle|\; (p, s) \in \bigcup \Psi_\Pi \text{ for some } s \in \{1, \ldots, 4\} \right\} \tag{10}$$

and

$$\Sigma_\Pi(p) = \left\{ s \in \{1, \ldots, 4\} \;\middle|\; (p, s) \in \bigcup \Psi_\Pi \right\}. \tag{11}$$

Note that $\Sigma_\Pi(p) \neq \emptyset$ for $p \in \Pi^*$. R^*_Π is equivalent with Eq. (4) where only $h_{(p,s)}(a_0, \dots, a_d)$ for (p, s) contained in $\bigcup \Psi_\Pi$ are involved. We now show that Eq. (9) serves as a simpler form of Eq. (4) when R_Π is bounded.

Theorem 3. *For Π such that R_Π is bounded, $R^*_\Pi = R_\Pi$.*

Proof. Suppose that $(p, s) \in \Pi \times \{1, \dots, 4\}$ is not in $\bigcup \Psi_\Pi$. Lemma 1 (in the case of $m = 1$) then implies $F_\Pi(p, s) = \emptyset$, which means that (p, s) does not contribute to determining the boundary of R_Π. □

Since the theorems above hold true only when R_Π is bounded, it is important to know when R_Π is bounded. We conclude this section by giving a sufficient condition for which R_Π is bounded. Recall that the coordinates of the pth data $(p = 1, \dots, n)$ is denoted by (i_p, j_p).

Theorem 4. *Let $\Pi = \{p_1, \dots, p_{d+1}\} \subset \{1, \dots, n\}$ be a set of $d+1$ data indices such that $\left|i_{p_\lambda} - i_{p_\mu}\right| > 1$ for $\forall \lambda \neq \mu$. R_Π is bounded. For any $\Pi' \subset \{1, \dots, n\}$ such that $\Pi' \supset \Pi$, therefore, $R_{\Pi'}$ is bounded.*

Proof. We show that a superset $R'_\Pi \subset \mathbb{R}^{d+1}$ of R_Π defined in the following is bounded.

For $\lambda = 1, \dots, d+1$, we first define

$$R'_{p_\lambda} = \left\{ (a_0, \dots, a_d) \in \mathbb{R}^{d+1} \;\middle|\; \begin{array}{c} \min\limits_{(x',y') \in S} \left[(j_{p_\lambda} + y') - \sum\limits_{l=0}^{m} (i_{p_\lambda} + x')^l a_l \right] \\ \leq 0 \leq \\ \max\limits_{(x',y') \in S} \left[(j_{p_\lambda} + y') - \sum\limits_{l=0}^{m} (i_{p_\lambda} + x')^l a_l \right] \end{array} \right\}, \tag{12}$$

where $S = \left\{ (x', y') \in \mathbb{R}^2 \;\middle|\; \max\{|x'|, |y'|\} \leq \frac{1}{2} \right\}$. S is the square having (x_1, y_1), …, (x_4, y_4) as its vertices. We therefore have $R'_{p_\lambda} \supset R_{p_\lambda}$. Since S is connected in $\mathbb{R}^2$, the intermediate value theorem allows to rewrite Eq. (12) as

$$R'_{p_\lambda} = \left\{ (a_0, \dots, a_d) \in \mathbb{R}^{d+1} \;\middle|\; \begin{array}{l} (j_{p_\lambda} + y') = \sum\limits_{l=0}^{d+1} (i_{p_\lambda} + x')^l a_l \\ \text{for some } (x', y') \in S \end{array} \right\}. \tag{13}$$

We then define R'_Π by $R'_\Pi = \bigcap_{\lambda=1}^{d+1} R'_{p_\lambda}$, which is written as

$$R'_\Pi = \left\{ (a_0, \dots, a_d) \in \mathbb{R}^{d+1} \;\middle|\; \begin{array}{l} (j_{p_\lambda} + y'_\lambda) = \sum\limits_{l=0}^{d+1} (i_{p_\lambda} + x'_\lambda)^l a_l \\ \text{for some } (x'_1, y'_1), \dots, (x'_{d+1}, y'_{d+1}) \in S \end{array} \right\}. \tag{14}$$

R'_Π is therefore obtained by collecting $(a_0, \dots, a_d) \in \mathbb{R}^{d+1}$ satisfying

Algorithm 1. Discrete polynomial curve fitting.

Require: P, d and $I \subset \{1, \ldots, n\}$ such that $R_I \neq \emptyset$ and R_I is guaranteed by Theorem 4 to be bounded.
Ensure: $\Pi \subset \{1, \ldots, n\}$ and C_Π.

Initialize $\Pi :=$ any set of $d+1$ elements of I satisfying the property in Theorem 4.
Initialize $\Pi^{\complement} := \emptyset$.
Compute Ψ_Π and C_Π.
while $I \setminus \Pi \neq \emptyset$ **do**
    $p :=$ any data index in $I \setminus \Pi$.
    Compute $\Psi_{\Pi \cup \{p\}}$ and $C_{\Pi \cup \{p\}}$ by Algorithm 2
    $\Pi := \Pi \cup \{p\}$.
end while
while $\{1, \ldots, n\} \setminus \left(\Pi \cup \Pi^{\complement}\right) \neq \emptyset$ **do**
    $p :=$ any data index in $\{1, \ldots, n\} \setminus \left(\Pi \cup \Pi^{\complement}\right)$.
    Compute $\Psi_{\Pi \cup \{p\}}$ and $C_{\Pi \cup \{p\}}$ by Algorithm 2
    if $\Psi_{I \cup \{p\}} \neq \emptyset$ **then**
        $\Pi := \Pi \cup \{p\}$.
    else
        $\Pi^{\complement} := \Pi^{\complement} \cup \{p\}$.
    end if
end while
return Π and C_Π.

$$\begin{pmatrix} j_{p_1} + y'_1 \\ j_{p_2} + y'_2 \\ \vdots \\ j_{p_{d+1}} + y'_{d+1} \end{pmatrix} = \begin{pmatrix} 1 & i_{p_1} + x'_1 & (i_{p_1} + x'_1)^2 & \cdots & (i_{p_1} + x'_1)^d \\ 1 & i_{p_2} + x'_2 & (i_{p_2} + x'_2)^2 & \cdots & (i_{p_2} + x'_2)^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & i_{p_{d+1}} + x'_{d+1} & (i_{p_{d+1}} + x'_{d+1})^2 & \cdots & (i_{p_{d+1}} + x'_{d+1})^d \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix} \tag{15}$$

for all combinations of $(x'_1, y'_1), \ldots, (x'_{d+1}, y'_{d+1}) \in S$. Since the $(d+1) \times (d+1)$ matrix in Eq. (15) is a Vandermonde matrix, its determinant is given by

$$\prod_{1 \leq \lambda < \mu \leq d+1} \left((i_{p_\mu} + x'_\mu) - (i_{p_\lambda} + x'_\lambda) \right), \tag{16}$$

none of whose factors can be zero from $\left| i_{p_\mu} - i_{p_\lambda} \right| > 1$ for $\forall \lambda \neq \mu$. For any fixed $(x'_1, y'_1), \ldots, (x'_{d+1}, y'_{d+1}) \in S$, therefore, $(a_0, \ldots, a_d)$ is uniquely determined by Eq. (15) to be a point with the coordinates of finite values. It follows from this that R'_Π is bounded. $\square$

4 Algorithm

4.1 Algorithm Ensuring Inclusion-Wise Maximal Inlier Set

Our method for discrete polynomial curve fitting, described in Algorithm 1, requires an initial inlier set and ensures an inclusion-wise maximal inlier set containing the initial set. The initial inlier set is represented by its corresponding

Algorithm 2. Update of Ψ_Π and C_Π for an additional inlier.

Require: P, d, $\Pi \subset \{1, \ldots, n\}$, $p \in \{1, \ldots, n\} \setminus \Pi$, Ψ_Π and C_Π.
Ensure: $\Psi_{\Pi \cup \{p\}}$ and $C_{\Pi \cup \{p\}}$.

Initialize $\Psi := \emptyset$ and $C := \emptyset$.
for all $\psi \in \Psi_\Pi$ **do**
  $(a_0, \ldots, a_d) :=$ the point in C_Π corresponding to ψ.
  if $(a_0, \ldots, a_d) \in R_p$ **then**
    $\Psi := \Psi \cup \{\psi\}$ and $C := C \cup \{(a_0, \ldots, a_d)\}$.
  end if
end for
Compute $\Phi^1_{\Pi,p}$ and $\Phi^2_{\Pi,p}$ (Eqs. (7) and (8)).
for all $\psi = \{(p_1, s_1), \ldots, (p_{d+1}, s_{d+1})\} \in \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$ **do**
  if $\bigcap_{\lambda=1}^{d+1} H(p_\lambda, s_\lambda)$ has the unique point **then**
    $(a_0, \ldots, a_d) :=$ the unique point in $\bigcap_{\lambda=1}^{d+1} (p_\lambda, s_\lambda)$
    if $(a_0, \ldots, a_d) \in R^*_\Pi \cap \bigcap_{\lambda=1}^{d+1} F(p_\lambda, s_\lambda)$ **then**
      $\Psi := \Psi \cup \{\psi\}$ and $C := C \cup \{(a_0, \ldots, a_d)\}$.
    end if
  end if
end for
return $\Psi = \Psi_{\Pi \cup \{p\}}$ and $C = C_{\Pi \cup \{p\}}$.

index set I in Algorithm 1. The algorithm divides the data indices $1, \ldots, n$ into two classes Π and $\Pi^{\complement}$: those for inliers are sorted into Π, while those for outliers into $\Pi^{\complement}$. Π is first initialized to be a set of $d+1$ indices in I, for which Ψ_Π and C_Π are computed at low cost using Eq. (6). In the two while-loops, then, we add new data indices to Π one by one accordingly computing corresponding Ψ_Π and C_Π using Algorithm 2. The first while-loop in Algorithm 1 is to obtain Ψ_I and C_I, where $\Psi_{\Pi \cup \{p\}}$ in each iteration cannot be empty from $R_I \neq \emptyset$ (Theorem 1). Note that R_I is nonempty and bounded. The second while-loop is to obtain Ψ_Π and C_Π for Π such that $\Pi \supset I$, where a data index p is sorted into $\Pi^{\complement}$ if $\Psi_{\Pi \cup \{p\}} = \emptyset$, i.e., $R_{\Pi \cup \{p\}} = \emptyset$ (Theorem 1).

Algorithm 2 shows how to compute $\Psi_{\Pi \cup \{p\}}$ and $C_{\Pi \cup \{p\}}$ for $\Pi \subset \{1, \ldots, n\}$ and $p \in \{1, \ldots, n\} \setminus \Pi$ when Ψ_Π and C_Π are known. The algorithm evaluates each set in $\Psi_\Pi \cup \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$ to check if it is in $\Psi_{\Pi \cup \{p\}}$ (Theorem 2). The first for-loop is to evaluate the sets in Ψ_Π, while the second for-loop is to evaluate the sets in $\Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$. Why a set in $\Psi_\Pi \cup \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$ is verified to be in $\Psi_{\Pi \cup \{p\}}$ in this way is explained in Sect. 3.2 (after the proof of Theorem 2). In the second loop we use Theorem 3 ($R^*_\Pi = R_\Pi$) to reduce the computational cost for checking if $(a_0, \ldots, a_d) \in F_\Pi(p_\lambda, s_\lambda)$ $(= R_\Pi \cap F(p_\lambda, s_\lambda))$ for $\lambda = 1, \ldots, d+1$.

Since $\Psi_{\Pi \cup \{p\}} \neq \emptyset$ if $R_{\Pi \cup \{p\}} \neq \emptyset$ from Theorem 1, after the second loop in Algorithm 1 we obtain an inclusion-wise maximal inlier set, which is equivalently stated in the following theorem.

Theorem 5. *Let $\Pi \subset \{1, \ldots, n\}$ be a data index set obtained by Algorithm 1. There exists no $\Pi' \subset \{1, \ldots, n\}$ satisfying $\Pi' \supset \Pi$ and $R_{\Pi'} \neq \emptyset$.*

The output of Algorithm 1 depends on the initial inlier set (i.e., I), and therefore how to determine I is an important issue. Most straightforwardly we can just set I to be random $d+1$ data indices for which R_I is bounded according to Theorem 4, where $R_I \neq \emptyset$ can be evaluated by computing C_I (Theorem 1). Since our objective is to maximize the number of inliers, however, it is better to give I that is as large as possible. Note that Algorithm 1 outputs $\Pi \subset \{1, \ldots, n\}$ such that $\Pi \supset I$. To reduce the possibility of being trapped in a local optimum with a small number of inliers, it is also important for I not to be contaminated with noise. For the acquisition of such I, we can use a robust estimation algorithms such as RANSAC [1].

The output of Algorithm 1 also depends on the order in which data are added to the initial inlier set, i.e., how to choose p in Line 10. Choosing p corresponding to an outlier for the optimal solution here may make it impossible for many data to be added. The performance of the algorithm therefore might be improved by incorporating a procedure to select a "good" p, which is out of the scope of this paper.

4.2 Computational Complexity

We give the computational cost required for Algorithm 1. We remark that here we discuss the computational cost depending the number n of data where the degree d is treated as a constant. The computational cost for each iteration in the two while-loops (i.e., the computational cost for Algorithm 2), depending on $|\Pi|$, is written as $\left|\Psi_\Pi \cup \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}\right|$ multiplied by the computational cost required for checking if a set in $\Psi_\Pi \cup \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$ is in $\Psi_{\Pi\cup\{p\}}$.

We first consider the order of $|\Psi_\Pi|$. We remark that $\left|\Phi^1_{\Pi,p}\right|$ and $\left|\Phi^2_{\Pi,p}\right|$ depend on $|\Psi_\Pi|$: in fact, we generally have $\left|\Phi^1_{\Pi,p}\right| = 4\,(d+1)\,|\Psi_\Pi|$ and $\left|\Phi^2_{\Pi,p}\right| = 2\binom{d+1}{2}|\Psi_\Pi| = d\,(d+1)\,|\Psi_\Pi|$. Since a set in Ψ_Π is composed of $d+1$ elements of $\Pi \times \{1, \ldots, 4\}$, $|\Psi_\Pi|$ is bounded by the number of ways to pick $d+1$ elements of $\Pi \times \{1, \ldots, 4\}$, i.e., $\binom{4|\Pi|}{d+1} = \mathcal{O}\left(|\Pi|^{d+1}\right)$. This upper bound is reduced by removing sets of $d+1$ elements of $\Pi \times \{1, \ldots, 4\}$ containing $(p, \underline{s})$ and $(p, \overline{s})$ for any $p \in \Pi$, $\underline{s} \in \{1, 2\}$ and $\overline{s} \in \{3, 4\}$ (such sets cannot be in $\Psi_{\Pi\cup\{p\}}$ since $F(p, \underline{s}) \cap F(p, \overline{s}) = \emptyset$), which however does not change the order $\mathcal{O}\left(|\Pi|^{d+1}\right)$.

We next consider the computational cost for evaluating $\psi \in \Psi_{\Pi\cup\{p\}}$ for $\psi = \{(p_1, s_1), \ldots, (p_{d+1}, s_{d+1})\} \in \Psi_\Pi \cup \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$. For $\psi \in \Psi_\Pi$, we only have to check if the corresponding $(a_0, \ldots, a_d) \in C_\Pi$ satisfies $(a_0, \ldots, a_d) \in R_p$, which takes a constant cost $\mathcal{O}(1)$. For $\psi \in \Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}$, on the other hand, the computational cost is $\mathcal{O}(|\Pi|)$: we first have to check if $(a_0, \ldots, a_d) \in \bigcap_{\lambda=1}^{d+1} H(p_\lambda, s_\lambda)$ uniquely exists (computational cost: $\mathcal{O}(1)$), and if so, we then have to check if $(a_0, \ldots, a_d) \in \bigcap_{\lambda=1}^{d+1} F_\Pi(p_\lambda, s_\lambda)$ (computational cost: $\mathcal{O}(|\Pi|)$).

Since $\mathcal{O}(|\Psi_\Pi|) = \mathcal{O}\left(\left|\Phi^1_{\Pi,p} \cup \Phi^2_{\Pi,p}\right|\right) = \mathcal{O}\left(|\Pi|^{d+1}\right)$, the computational cost for each iteration in the two while-loops is therefore obtained as $\mathcal{O}\left(|\Pi|^{d+1}\right) \times$

$\mathcal{O}(|\Pi|) = \mathcal{O}\left(|\Pi|^{d+2}\right)$. In the first iteration $|\Pi| = d+1$, and in the last iteration $|\Pi| = n-1$ at most. The theoretical computational cost for Algorithm 1 is therefore $\sum_{m=d+1}^{n-1} \mathcal{O}\left(m^{d+2}\right) = \mathcal{O}\left(n^{d+2}\right)$.

5 Conclusions

We dealt with the problem of fitting a discrete polynomial curve to 2D noisy data, for which we proposed a method guaranteeing inclusion-wise maximality of its obtained inlier set. The method is constructed based on our investigation on the properties of the feasible regions in the parameter space corresponding to input data points. Evaluation of the practical performance of the proposed method is left for future work. This work may be extended to implicit functions ($f(x,y) = 0$) and surface fitting in 3D.

A Appendix: Proof of Lemma 1

Proof. For $m = 1, \ldots, d$, let $(p_1, s_1), \ldots, (p_m, s_m) \in \Pi \times \{1, \ldots, 4\}$ satisfy (i) and (ii) in Lemma 1. It suffices to show that there always exists $(p_{m+1}, s_{m+1}) \in \Pi \times \{1, \ldots, 4\}$ such that $\bigcap_{\lambda=1}^{m+1} F_\Pi(p_\lambda, s_\lambda) \neq \emptyset$ and $H(p_1, s_1), \ldots, H(p_{m+1}, s_{m+1})$ are linearly independent. See Fig. 5 for an illustration of this proof.

Let $(a'_0, \ldots, a'_d) \in \mathbb{R}^{d+1}$ be a point in $\bigcap_{\lambda=1}^{m} F_\Pi(p_\lambda, s_\lambda)$. No proof is required for the case where there exists $(p_{m+1}, s_{m+1}) \in \Pi \times \{1, \ldots, 4\}$ such that $(a'_0, \ldots, a'_d) \in F_\Pi(p_{m+1}, s_{m+1})$ and $H(p_1, s_1), \ldots, H(p_{m+1}, s_{m+1})$ are linearly independent. We therefore assume otherwise. Since $F_\Pi(p,s) \subset H(p,s)$ for any (p,s), we have $(a'_0, \ldots, a'_d) \in \bigcap_{\lambda=1}^{m} H(p_\lambda, s_\lambda)$. $\bigcap_{\lambda=1}^{m} H(p_\lambda, s_\lambda)$ is a $(d+1-m)$-dimensional flat ($d+1-m \geq 1$), and therefore we may consider a half-line in $\bigcap_{\lambda=1}^{m} H(p_\lambda, s_\lambda)$ running from $(a'_0, \ldots, a'_d)$. A point in the half-line is represented by $(a''_0(r), \ldots, a''_d(r))$ where $a''_l(r) = a'_l + rv_l$ ($l = 0, \ldots, d$) with some non-zero vector $(v_0, \ldots, v_d) \in \mathbb{R}^{d+1}$ and a non-negative parameter $r \in \mathbb{R}_{\geq 0}$: $(a''_0(r), \ldots, a''_d(r)) = (a'_0, \ldots, a'_d)$ for $r = 0$, and as we increase the value of r, the point traces the half-line in the direction of the vector $(v_0, \ldots, v_d)$.

Since $F_\Pi(p_\lambda, s_\lambda)$ ($\subset R_\Pi$) is bounded for $\lambda = 1, \ldots, m$, a large enough r satisfies $(a''_0(r), \ldots, a''_d(r)) \notin \bigcap_{\lambda=1}^{m} F_\Pi(p_\lambda, s_\lambda)$. Let r'_1 be the maximum value of r such that any $r \leq r'_1$ satisfies $(a''_0(r), \ldots, a''_d(r)) \in R_\Pi$ (note that this may be satisfied for some $r > r'_1$ when R_Π is concave). Let r'_2, on the other hand, be the maximum value of r satisfying $(a''_0(r), \ldots, a''_d(r)) \in \bigcap_{\lambda=1}^{m} F(p_\lambda, s_\lambda)$ (note that this is satisfied for any $r < r'_2$ since $F(p_\lambda, s_\lambda)$ is convex for $\lambda = 1, \ldots, m$), where we put $r'_2 = \infty$ if it is satisfied for any $r > 0$. Then, $r' = \min\{r'_1, r'_2\}$ is the maximum value of r such that any $r \leq r'$ satisfies $(a''_0(r), \ldots, a''_d(r)) \in \bigcap_{\lambda=1}^{m} F_\Pi(p_\lambda, s_\lambda)$ (recall that $F_\Pi(p,s) = F(p,s) \cap R_\Pi$). We now show that $(a''_0(r'), \ldots, a''_d(r')) \in F_\Pi(p_{m+1}, s_{m+1})$ for some $(p_{m+1}, s_{m+1}) \in \Pi \times \{1, \ldots, 4\}$ where $(p_{m+1}, s_{m+1}) \neq (p_\lambda, s_\lambda)$ for $\lambda = 1, \ldots, m$. We remark that, for such (p_{m+1}, s_{m+1}), $H(p_1, s_1), \ldots, H(p_{m+1}, s_{m+1})$ are linearly independent, since otherwise it is impossible to have $(a''_0(r'), \ldots, a''_d(r')) \in H(p_{m+1}, s_{m+1})$ whereas $(a''_0(0), \ldots, a''_d(0)) \notin H(p_{m+1}, s_{m+1})$.

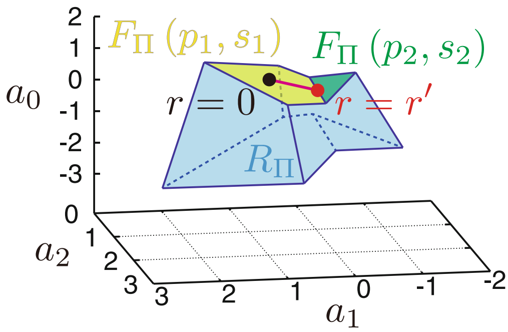


Fig. 5. Illustration for the proof of Lemma 1. We assume $m = 1$ here. Let the black point depict $(a'_0, \ldots, a'_d) = (a''_0(0), \ldots, a''_d(0))$. As we increase the value of r from zero, $(a''_0(r), \ldots, a''_d(r))$ traces a half-line in $H(p_1, s_1)$ running from $(a'_0, \ldots, a'_d)$. For $r = r'$ (intuitively speaking, just before $(a''_0(r), \ldots, a''_d(r))$ comes off $F_\Pi(p_1, s_1)$), then, $(a''_0(r), \ldots, a''_d(r))$ is in $F_\Pi(p_1,s_1) \cap F_\Pi(p_2,s_2)$ for some $(p_2,s_2) \in \Pi \times \{1,\ldots,4\}$, so that we have $m = 2$ with $F_\Pi(p_1,s_1)$ and $F_\Pi(p_2,s_2)$. $(a''_0(r'), \ldots, a''_d(r'))$ is depicted in a red point. (Color figure online)

We first assume the case of $r' = r'_1 < r'_2$: as we increase the value of r past r', $(a''_0(r), \ldots, a''_d(r))$ gets out of R_Π, i.e., $R_{p_{m+1}}$ for some $p_{m+1} \in \Pi$ (recall that $R_\Pi = \bigcap_{p\in\Pi} R_p$). We remark that $p_{m+1} \neq p_1, \ldots, p_m$ for $\lambda = 1, \ldots, m$ since any r satisfies $(a''_0(r), \ldots, a''_d(r)) \in H(p_\lambda, s_\lambda) \subset R_{p_\lambda}$. For this p_{m+1}, therefore, we have $(a''_0(r'), \ldots, a''_d(r')) \in B_{p_{m+1}}$, i.e., $(a''_0(r'), \ldots, a''_d(r')) \in F(p_{m+1}, s_{m+1})$ for some $s_{m+1} \in \{1,\ldots,4\}$. $(a''_0(r'), \ldots, a''_d(r')) \in F(p_{m+1}, s_{m+1}) \cap R_\Pi = F_\Pi(p_{m+1}, s_{m+1})$, accordingly.

We next assume the case of $r' = r'_2 \leq r'_1$: as we increase the value of r past r', $(a''_0(r), \ldots, a''_d(r))$ gets out of $F(p_\lambda, s_\lambda)$ for some $\lambda \in \{1,\ldots,m\}$. Without loss of generality, we assume that $\lambda = 1$, and $(a''_0(r), \ldots, a''_d(r)) \in \overline{B}_{p_1}$ for $r \leq r'$. We then have

$$h_{(p_1,s_1)}(a''_0(r), \ldots, a''_d(r)) = \max_{s\in\{1,\ldots,4\}} h_{(p_1,s)}(a''_0(r), \ldots, a''_d(r)) \text{ for } r \leq r', \tag{17}$$

while

$$h_{(p_1,s_1)}(a''_0(r), \ldots, a''_d(r)) < \max_{s\in\{1,\ldots,4\}} h_{p_1,s}(a''_0(r), \ldots, a''_d(r)) \text{ for } r > r'. \tag{18}$$

This suggests that for some $s'_1 \in \{1,\ldots,4\} \setminus \{s_1\}$ we have

$$\begin{aligned} 0 = h_{(p_1,s_1)}(a''_0(r'), \ldots, a''_d(r')) &= h_{(p_1,s'_1)}(a''_0(r'), \ldots, a''_d(r')) \\ &= \max_{s\in\{1,\ldots,4\}} h_{(p_1,s)}(a''_0(r'), \ldots, a''_d(r')), \end{aligned} \tag{19}$$

Note that $h_{(p_1,s_1)}(a''_0(r), \ldots, a''_d(r)) = 0$ (i.e., $(a''_0(r), \ldots, a''_d(r)) \in H(p_1,s_1)$) is satisfied for any r. $(a''_0(r'), \ldots, a''_d(r')) \in F(p_1,s'_1) \cap R_\Pi = F_\Pi(p_1,s'_1)$, consequently. $\square$

References

1. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**, 381–395 (1981)
2. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM **15**, 11–15 (1972)
3. Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recogn. **13**, 111–122 (1981)
4. Cohen-Or, D., Kaufman, A.: Fundamentals of surface voxelization. Graph. Models Image Process. **57**, 453–461 (1995)
5. Brimkov, V.E., Andres, E., Barneva, R.P.: Object discretization in higher dimensions. In: Borgefors, G., Nyström, I., Baja, G.S. (eds.) DGCI 2000. LNCS, vol. 1953, pp. 210–221. Springer, Heidelberg (2000). doi:10.1007/3-540-44438-6_18
6. Tajine, M., Ronse, C.: Topological properties of Hausdorff discretization, and comparison to other discretization schemes. Theoret. Comput. Sci. **283**, 243–268 (2002)
7. Lincke, C., Wüthrich, C.A.: Surface digitizations by dilations which are tunnel-free. Discret. Appl. Math. **125**, 81–91 (2003)
8. Brimkov, V.E., Barneva, R.P., Brimkov, B.: Minimal offsets that guarantee maximal or minimal connectivity of digital curves in nD. In: Brlek, S., Reutenauer, C., Provençal, X. (eds.) DGCI 2009. LNCS, vol. 5810, pp. 337–349. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04397-0_29
9. Brimkov, V.E.: On connectedness of discretized objects. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Li, B., Porikli, F., Zordan, V., Klosowski, J., Coquillart, S., Luo, X., Chen, M., Gotz, D. (eds.) ISVC 2013. LNCS, vol. 8033, pp. 246–254. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41914-0_25
10. Khalimsky, E., Kopperman, R., Meyer, P.R.: Computer graphics and connected topologies on finite ordered sets. Topol. Its Appl. **36**, 1–17 (1990)
11. Andres, E., Acharya, R., Sibata, C.: Discrete analytical hyperplanes. Graph. Models Image Process. **59**, 302–309 (1997)
12. Andres, E., Jacob, M.A.: The discrete analytical hyperspheres. IEEE Trans. Vis. Comput. Graph. **3**, 75–86 (1997)
13. Andres, E.: Discrete linear objects in dimension n: the standard model. Graph. Models **65**, 92–111 (2003)
14. Andres, E., Roussillon, T.: Analytical description of digital circles. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 235–246. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19867-0_20
15. Toutant, J.L., Andres, E., Roussillon, T.: Digital circles, spheres and hyperspheres: From morphological models to analytical characterizations and topological properties. Discret. Appl. Math. **161**, 2662–2677 (2013)
16. Toutant, J.-L., Andres, E., Largeteau-Skapin, G., Zrour, R.: Implicit digital surfaces in arbitrary dimensions. In: Barcucci, E., Frosini, A., Rinaldi, S. (eds.) DGCI 2014. LNCS, vol. 8668, pp. 332–343. Springer, Cham (2014). doi:10.1007/978-3-319-09955-2_28
17. Zrour, R., Kenmochi, Y., Talbot, H., Shimizu, I., Sugimoto, A.: Combinatorial optimization for fitting of digital line and plane. In: International Workshop on Computer Vision and Its Application to Image Media Processing, Satellite Workshop of the 3rd Pacific-Rim Symposium on Image and Video Technology, pp. 35–41 (2009)
18. Kenmochi, Y., Buzer, L., Talbot, H.: Efficiently computing optimal consensus of digital line fitting. In: International Conference on Pattern Recognition (ICPR2010), pp. 1064–1067. IEEE (2010)

19. Zrour, R., Kenmochi, Y., Talbot, H., Buzer, L., Hamam, Y., Shimizu, I., Sugimoto, A.: Optimal consensus set for digital line and plane fitting. Int. J. Imaging Syst. Technol. **21**, 45–57 (2011)
20. Sere, A., Sie, O., Andres, E.: Extended standard Hough transform for analytical line recognition. In: Proceedings of International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT2012), pp. 412–422. IEEE (2012)
21. Zrour, R., Largeteau-Skapin, G., Andres, E.: Optimal consensus set for annulus fitting. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 358–368. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19867-0_30
22. Largeteau-Skapin, G., Zrour, R., Andres, E., Sugimoto, A., Kenmochi, Y.: Optimal consensus set and preimage of 4-connected circles in a noisy environment. In: Proceedings of the International Conference on Pattern Recognition (ICPR2012), pp. 3774–3777. IEEE (2012)
23. Largeteau-Skapin, G., Zrour, R., Andres, E.: $O(n^3 \log n)$ time complexity for the optimal consensus set computation for 4-connected digital circles. In: Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B. (eds.) DGCI 2013. LNCS, vol. 7749, pp. 241–252. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37067-0_21
24. Zrour, R., Largeteau-Skapin, G., Andres, E.: Optimal consensus set for nD fixed width annulus fitting. In: Barneva, R.P., Bhattacharya, B.B., Brimkov, V.E. (eds.) IWCIA 2015. LNCS, vol. 9448, pp. 101–114. Springer, Heidelberg (2015). doi:10.1007/978-3-319-26145-4_8
25. Sekiya, F., Sugimoto, A.: Fitting discrete polynomial curve and surface to noisy data. Ann. Math. Artif. Intell. **75**, 135–162 (2015)
26. Sekiya, F., Sugimoto, A.: On connectivity of discretized 2D explicit curve. In: Ochiai, H., Anjyo, K. (eds.) Mathematical Progress in Expressive Image Synthesis II. MI, vol. 18, pp. 33–44. Springer, Heidelberg (2015). doi:10.1007/978-4-431-55483-7_4
27. Heijmans, H.: Morphological discretization. In: Geometrical Problems of Image Processing, pp. 99–106 (1991)
28. Heijmans, H.J.A.M., Toet, A.: Morphological sampling. CVGIP: Image Underst. **54**, 384–400 (1991)
29. Heijmans, H.J.A.M.: Discretization of morphological operators. J. Vis. Commun. Image Represent. **3**, 182–193 (1992)