Ann Math Artif Intell (2015) 75:135–162
DOI 10.1007/s10472-014-9425-7

Fitting discrete polynomial curve and surface to noisy data

Fumiki Sekiya · Akihiro Sugimoto

Published online: 11 June 2014

Abstract Fitting geometric models such as lines, circles or planes is an essential task in image analysis and computer vision. This paper deals with the problem of fitting a discrete polynomial curve to given 2D integer points in the presence of outliers. A 2D discrete polynomial curve is defined as a set of integer points lying between two polynomial curves. We formulate the problem as a discrete optimization problem in which the number of points included in the discrete polynomial curve, i.e., the number of inliers, is maximized. We then propose a robust method that effectively achieves a solution guaranteeing local maximality by using a local search, called rock climbing, with a seed obtained by RANSAC. We also extend our method to deal with a 3D discrete polynomial surface. Experimental results demonstrate the effectiveness of our proposed method.

Keywords Curve fitting · Surface fitting · Discrete polynomial curve · Discrete polynomial surface · Local optimal · Outliers

1 Introduction

Fitting geometric models such as lines, planes or circles is an essential task in image analysis and computer vision. It can be used in many procedures such as object recognition, shape approximation, and image segmentation. Though many methods exist for model fitting, in most cases they use continuous models even in a discrete environment.

F. Sekiya (✉)
Graduate School of Advanced Integration Science, Chiba University, Chiba, Japan
e-mail: sekiya@nii.ac.jp

Present Address:
F. Sekiya
Graduate University for Advanced Studies SOKENDAI, Hayama, Japan

A. Sugimoto
National Institute of Informatics, Tokyo, Japan
e-mail: sugimoto@nii.ac.jp

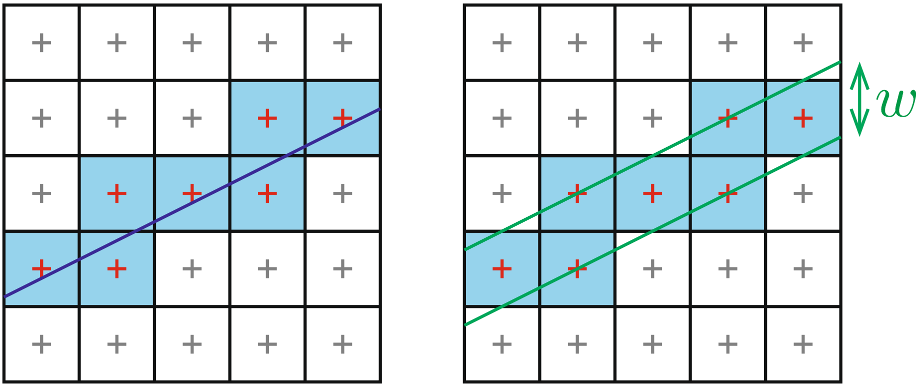
The method of least squares is most commonly used for model fitting. This method estimates model parameters by minimizing the sum of squared residuals from all data, where the solution can be analytically obtained. This method is, however, fatally susceptible to the presence of outliers: just one outlier can bring a great impact on estimation results. In order to enhance robustness, minimizing other criteria has been proposed (for details, see Chapter 1 of [25]). For example, the method of least absolute value (also known as least absolute deviation or L_1 regression) [19] minimizes the sum of absolute residuals from all data. The method of least median of squares [24] minimizes the median of squared residuals, resulting in tolerating up to half the data being outliers; it does not work in the presence of more outliers. With these method, the solution cannot be obtained analytically (i.e., in a closed form). Therefore, some optimization techniques are required for obtaining the solution as learned in [27].

In computer vision and image analysis, on the other hand, RANdom SAmple Consensus (RANSAC) [14] is widely used. This method aims at maximizing the number of inliers (i.e., data points consistent with a given model with allowing some error threshold), and it works regardless of the fraction of outliers. There exist many variants of RANSAC such as [11, 23, 28]. However, their random approach takes long time to ensure high accuracy, in particular, optimality. Moreover, most of them do not characterize any deterministic property such as local optimality on its obtained results. Another popular method in these fields is the one using the Hough transform [13, 16]. This method finds model parameters consistent with a large number of data points in the discretized space of the model parameters. It works regardless of outlier ratio, and is commonly used for detecting simple models with a small number of parameters, such as lines and circles. However, its output depends on a resolution of the parameter space determined by a user in an ad-hoc manner. In case the number of model parameters becomes large, the required time complexity and space complexity are both significantly expensive. Though randomized Hough transform [30] reduces the complexity in time and space, optimality of the obtained solution is not guaranteed.

In discrete spaces, it is preferable to use discrete models rather than continuous ones. An example is illustrated in Fig. 1, where we consider line fitting in a 2D discrete space. Figure 1(a) shows how discrete data are obtained from an original continuous object, i.e., discretization of a continuous line: the dark blue line is discretized into the light blue unit squares (pixels) intersecting with the line. They are represented by the coordinates of their centers (the red points). However, there exists no continuous line explaining all these points. Thus, a discretized object generally cannot be represented by a continuous model. This is why discrete models are introduced.

A discrete model is classically defined as the result of discretization locally applied to a continuous model, such as Bresenham's algorithms [6, 7]. A more recent approach [3, 4, 15, 29] is to use global analytical description, in which a discrete model is defined as the integer coordinate solutions of a finite set of inequalities. Such global description allows to know easily if an arbitrary point is explained by a given discrete model. A (analytical) discrete line in 2D is defined, for example, as the set of discrete points $(x, y) \in \mathbb{Z}^2$ satisfying $0 \leq y - (ax + b) \leq w$ where w is a non-negative constant. This model can represent a discretized line by collecting discrete points lying between two parallel lines as shown in Fig. 1(b), where the two green lines depict $y = ax + b$ and $y = ax + b + w$. Therefore, it is reasonable to use a discrete model for fitting in a discrete spaces.

Fitting of analytical discrete models is studied for lines [1, 8, 9, 12, 22, 31], annuluses (circles) [17, 18, 20, 32], and polynomial curves [21] in 2D, and for planes [2, 8–10, 12,



(a) Discretization of a continuous line.

(b) Discrete line.

Fig. 1 Advantage of fitting a discrete model to discrete data. Consider line fitting in a 2D discrete space as an example. (**a**) shows discretization of a continuous line; no continuous line can represent the obtained set of discrete points (the red points). On the other hand, a discrete line can represent the discretization by collecting discrete points lying between two parallel lines, as shown in (**b**).

22, 31] and polynomial surfaces [21] in 3D. In particular, [17, 18, 31, 32] guarantee the optimality of an obtained set of inliers by using discrete models. For discrete line fitting and discrete annulus fitting in 2D, and for discrete plane fitting in 3D, methods that work for a data set that include outliers, i.e., points that do not describe the model, have been developed. However, such a method that deals with outliers for discrete polynomial curves and surfaces remains to be reported. This paper aims at developing a method for discrete polynomial curve and surface fitting to a given set of discrete points in the presence of outliers.

We formulate the 2D discrete polynomial curve fitting problem as a discrete optimization problem where the number of inliers is maximized. We then propose a method that guarantees its output to achieve local optimal. Our proposed method combines RANSAC and a local search, named *rock climbing*. Namely, starting with a seed obtained by RANSAC, our method iteratively and locally searches for equivalent or better solutions to increase the number of inliers. Our method guarantees the obtained set of inliers is local maximum in the sense of the set inclusion. It works regardless of the fraction of outliers. We also show that the rock climbing can be directly applied to the 3D discrete polynomial surface fitting problem. Experimental results demonstrate the robustness and efficiency of our method for our fitting problems. We remark that a part of this work was presented in [26].

This paper is organized as follows. In Section 2, we formulate the 2D discrete polynomial curve fitting problem. We first define a discrete polynomial curve and formulate the fitting problem. We then reformulate the problem in the parameter space. After stating the properties of discrete polynomial curves in Section 3, we propose rock climbing that iteratively and locally improves the solution in Section 4. Section 5 discusses the computational complexity of our proposed method. The extension to the case of 3D discrete polynomial surface fitting problem is addressed in Section 6. In Section 7, we show some experimental results that demonstrate the efficiency of our method both for 2D discrete polynomial curves and 3D discrete polynomial surfaces.

2 Discrete polynomial curve fitting problem

2.1 Definitions of notions

A (continuous) polynomial curve of degree k in the Euclidean plane is defined by

$$P = \{(x, y) \in \mathbb{R}^2 : y = \sum_{i=0}^{k} a_i x^i,\ a_k \neq 0\}, \tag{1}$$

where $a_0, \dots, a_k \in \mathbb{R}$. We define a discretization of (1), namely, a *discrete polynomial curve*, by

$$D = \{(x, y) \in \mathbb{Z}^2 : 0 \leq y - f(x) \leq w\}, \tag{2}$$

where $f(x) = \sum_{i=0}^{k} a_i x^i$, and w is a constant uniquely determined as the absolute difference between the y-coordinates of two adjacent points in the discrete space (in other words, w corresponds to the resolution of the discrete space). In this paper, we identify a 2D discrete space as $\mathbb{Z}^2$, and therefore $w = 1$. a_i, k and w are respectively called the *coefficient*, the *degree*, and the *width* of the discrete polynomial curve ($i = 0, \dots, k$). Geometrically, D is a set of integer points lying between two polynomial curves $y = f(x)$ and $y = f(x) + w$, and w is the vertical distance between them. We remark that D. is called a Digital Level Layer (DLL) [15].

We define several notions for a discrete polynomial curve. For a finite set of discrete points (data)

$$S = \{p_j \in \mathbb{Z}^2 : j = 1, 2, \dots, n\},$$

with the coordinates of p_j being finite, and a discrete polynomial curve D, $p_j \in D$ is called an *inlier*, and $p_j \notin D$ is called an *outlier* of D. The set of inliers is called the *consensus set* of D which is denoted by C. We remark that $C = S \cap D$. Two polynomial curves $y = f(x)$ and $y = f(x) + w$ are called the *support lines* of D. In particular, we call $y = f(x)$ the *lower support line*, and $y = f(x) + w$ the *upper support line*. Points on the support lines are called *critical points* of D. In particular, a point on the lower support line is called a *lower critical point*, while that on the upper support line is an *upper critical point*.

2.2 Description of the discrete polynomial curve fitting problem

Let $\mathbb{D}_{k,w}$ be the set of all discrete polynomial curves of degree up to k with width w. The problem of discrete polynomial curve fitting is described as follows:

Input A set of discrete points S, a degree k, and a width w.
Output A $(k + 1)$-tuple of coefficients $(a_0, \dots, a_k)$ of $D \in \mathbb{D}_{k,w}$ having the maximum number of inliers.

A consensus set of the maximum number of inliers, denoted by $C_{\max}$, is called the *maximum consensus set*. We remark that not less than one optimal solution can exist.

2.3 Discrete polynomial curve fitting in the parameter space

A discrete polynomial curve of $\mathbb{D}_{k,w}$ is identified as a point in the parameter space $(a_0, \dots, a_k) = \mathbb{R}^{k+1}$. We formulate the problem of discrete polynomial curve fitting as an optimization problem in the parameter space to obtain the maximum consensus set.

Given a point $(x', y') \in S$, $(a_0, \dots, a_k)$ of $D \in \mathbb{D}_{k,w}$ such that $D \ni (x', y')$ satisfies

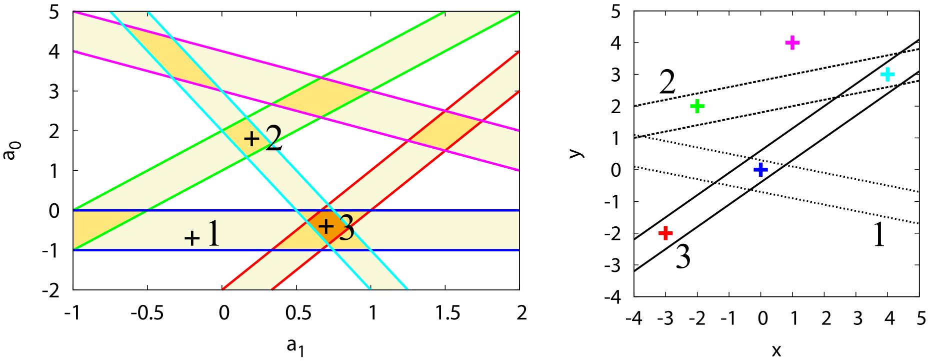
$$0 \leq -\sum_{i=0}^{k} x'^{i} a_i + y' \leq w \ . \tag{3}$$

We call the set of such points in the parameter space the *feasible region* for (x', y'). In particular, we call the set of points satisfying one of the two equalities the *feasible boundaries*; we call the one corresponding to the left-hand-side equality the *lower feasible boundaries*, and the other the *upper feasible boundaries*. (x', y') is a lower (upper resp.) critical point of the discrete polynomial curve corresponding to a point on the lower (upper resp.) feasible boundary. For a consensus set $C = \{(x_1, x_1), \dots, (x_m, y_m)\}$, we have $(a_0, \dots, a_k)$ that satisfies

$$\begin{cases} 0 \leq -\sum_{i=0}^{k} x_1^i a_i + y_1 \leq w \ , \\ \qquad\qquad \vdots \\ 0 \leq -\sum_{i=0}^{k} x_m^i a_i + y_m \leq w \ . \end{cases} \tag{4}$$

Letting P_C be the region defined by (4) (the intersection of these feasible regions). We remark that the region is a convex polytope if it is bounded. P_C is the set of $(a_0, \dots, a_k)$ determining $D \in \mathbb{D}_{k,w}$ such that $S \cap D \supseteq C$. Note that $S \cap D = C$ does not always hold. Therefore, D determined by $(a_0, \dots, a_k)$ in P_C contains at least $|C|$ inliers. For an arbitrary consensus set C' such that $C' \supset C$, $P_{C'} \subset P_C$ since $P_{C'}$ is the intersection of P_C and the feasible regions for the points in $C' \backslash C$.

Finding $C_{\max}$ is equivalent to finding the region for $C_{\max}$ in the parameter space. Figure 2 shows an example of correspondence between the parameter space and the primal space in the case of $k = 1$. Note that an intersection of feasible regions in this case is a convex polygon as long as it is bounded.



(a) Parameter space.

(b) Primal space.

Fig. 2 Examples of the parameter space (**a**) and the corresponding primal space (**b**) in the case of $k = 1$ and $w = 1$. Correspondence between the feasible regions in the parameter space and the points in the primal space are indicated by the same colors. The three points numbered from 1 to 3 in the parameter space determine the three discrete polynomial curves with the same numbers in the primal space (they are represented by their support lines). Note that the darkness of a region in the parameter space is proportional to the number of inliers.

If we denote by $F(a_0, \ldots, a_k)$ the number of inliers of D determined by $(a_0, \ldots, a_k)$, then the discrete polynomial curve fitting problem is equivalent to seeking

$$\underset{(a_0,\ldots,a_k)}{\arg\max}\, F(a_0, \ldots, a_k) \tag{5}$$

for given S, k, and w.

3 Properties of discrete polynomial curves

A polynomial curve of degree k is uniquely determined by $k+1$ different points on the curve. Theorem 1 states that a discrete polynomial curve also has a similar property.

Theorem 1 *A discrete polynomial curve $D \in \mathbb{D}_{k,w}$ is uniquely determined by $k+1$ critical points having $k+1$ different x-coordinates where each of them is specified whether it is an upper or a lower critical point.*

Proof If a discrete polynomial curve $D \in \mathbb{D}_{k,w}$ has $k+1$ critical points (s_1, t_1), …, (s_{k+1}, t_{k+1}) such that $s_i \neq s_j$ for all $i \neq j$, then the coefficients of D must satisfy

$$\begin{cases} -\sum_{i=0}^{k} s_1^i a_i + t_1 = c_1 \ , \\ \qquad\qquad \vdots \\ -\sum_{i=0}^{k} s_{k+1}^i a_i + t_{k+1} = c_{k+1} \ , \end{cases} \tag{6}$$

where

$$c_j = \begin{cases} 0 & \text{if } (s_j, t_j) \text{ is a lower critical point} \\ w & \text{if } (s_j, t_j) \text{ is an upper critical point} \end{cases} \quad (j = 1, \ldots, k+1) \ .$$

(6) can be rewritten as

$$\begin{pmatrix} 1 & s_1 & s_1^2 & \cdots & s_1^k \\ 1 & s_2 & s_2^2 & \cdots & s_2^k \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & s_{k+1} & s_{k+1}^2 & \cdots & s_{k+1}^k \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} t_1 - c_1 \\ t_2 - c_2 \\ \vdots \\ t_{k+1} - c_{k+1} \end{pmatrix} . \tag{7}$$

The $(k+1) \times (k+1)$ matrix in the left-hand side is a Vandermonde matrix. Therefore, its determinant equals to $\prod_{1 \le i < j \le k+1} (s_i - s_j)$, and cannot be zero since $s_i \neq s_j$ for all $i \neq j$ $(i, j = 1, \ldots, k+1)$. This means that D is uniquely determined. □

We remark that in general (6) does not have a solution if $s_i = s_j$ for $\exists i, j$ $(i \neq j)$.

Theorem 1 indicates that the set of all discrete polynomial curves in $\mathbb{D}_{k,w}$ generated from $k+1$ points in S is finite where the $k+1$ points are used as critical points (note that different specifications for the $k+1$ critical points whether they are lower or upper generate different discrete polynomial curves). The set is denoted by $\mathbb{G}_{S,k,w}$. $\mathbb{G}_{S,k,w}$ is not empty iff the points in S have at least $k+1$ different x-coordinates.

Assume that $\mathbb{G}_{S,k,w}$ is not empty. To identify a discrete polynomial curve in $\mathbb{G}_{S,k,w}$, we consider $2n$ hyperplanes that are the feasible boundaries for all the points in a given $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$,

$$\begin{matrix} -\sum_{i=0}^{k} x_j^i a_i + y_j = 0 \\ -\sum_{i=0}^{k} x_j^i a_i + y_j = w \end{matrix} \quad (j = 1, \ldots, n) \ . \tag{8}$$

Note that the feasible boundaries for $(x'_1, y'_1) \in S$ and $(x'_2, y'_2) \in S$ are parallel iff $x'_1 = x'_2$. Since $D \in \mathbb{G}_{S,k,w}$ has at least $k+1$ critical points with $k+1$ different x-coordinates, $(a_0, \ldots, a_k)$ determining D satisfies at least $k+1$ independent equations in (8). Therefore, D is an intersection point of the feasible boundaries identified by these equations. Figure 3 shows an example of discrete polynomial curves of $\mathbb{G}_{S,k,w}$ in the parameter space. We remark that for an arbitrary consensus set C, any discrete polynomial curve of $\mathbb{D}_{k,w}$ determined by a vertex of P_C is an element of $\mathbb{G}_{S,k,w}$.

Since $\mathbb{G}_{S,k,w}$ is a finite set, if it contains an element having the maximum consensus set, then we can find the optimal $(a_0, \ldots, a_k)$ (in the sense that it maximizes the number of inliers) by a brute-force search in $\mathbb{G}_{S,k,w}$.

Theorem 2 *If $\mathbb{G}_{S,k,w}$ is not empty, then there exists $D \in \mathbb{G}_{S,k,w}$ such that $S \cap D = C_{\max}$.*

To prove Theorem 2, we need the following lemma.

Lemma 1 *If $\mathbb{G}_{S,k,w}$ is not empty, then the points in $C_{\max}$ have at least $k+1$ different x-coordinates.*

Proof We show that a consensus set C whose points have $m \le k$ different x-coordinates is not maximum. Let $X_1, \ldots, X_m$ be these x-coordinates. Then, P_C is written by

$$\begin{cases} \overline{Y}_1 - w \le \sum_{i=0}^{k} X_1^i a_i \le \underline{Y}_1 \,, \\ \qquad\qquad \vdots \\ \overline{Y}_m - w \le \sum_{i=0}^{k} X_m^i a_i \le \underline{Y}_m \,, \end{cases} \tag{9}$$

where $\overline{Y}_j$ is the maximum y-coordinate among the points in C on $x = X_j$, while $\underline{Y}_j$ is the minimum y-coordinate among them $(j = 1, \ldots, m)$. Since the points in S have at least

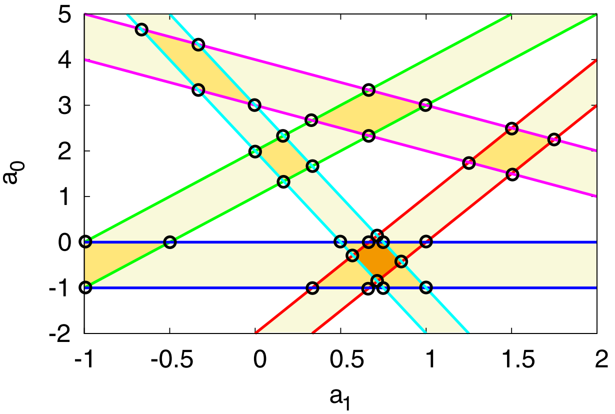


Fig. 3 Discrete polynomial curves in $\mathbb{G}_{S,k,w}$ in the parameter space ($k = 1, w = 1$). They are the intersection points of the feasible boundaries; the black points represent them.

$k+1$ different x-coordinates (because $\mathbb{G}_{S,k,w}$ is assumed not to be empty), there exists a point $(X,Y) \in S \backslash C$ such that $X \neq X_j$ for $j=1,\dots,m$. The feasible region for (X,Y) is

$$0 \leq -\sum_{i=0}^{k} X^i a_i + Y \leq w\ . \tag{10}$$

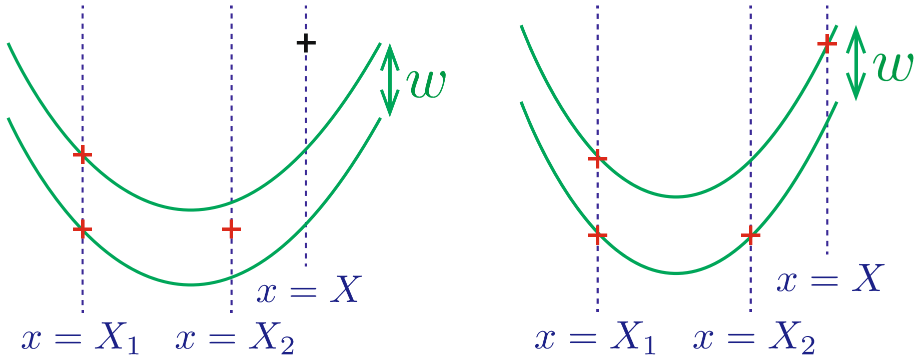
By combining (9) and (10), and introducing h_j and h such that $\overline{Y}_j - w \leq h_j \leq \underline{Y}_j$ for $j=1,\dots,m$ and $Y-w \leq h \leq Y$, we obtain

$$\begin{pmatrix} 1 & X_1 & X_1^2 & \cdots & X_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_m & X_m^2 & \cdots & X_m^k \\ 1 & X & X^2 & \cdots & X^k \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ h \end{pmatrix}\ . \tag{11}$$

(11) has at least one solution in $(a_0,\dots,a_k)$, from the same discussion used for (7) in the proof of Theorem 1. Therefore, there exists at least one discrete polynomial curve $D' \in \mathbb{D}_{k,w}$ such that $D' \supset C \cup \{(X,Y)\}$, which concludes that C is not maximum. □

Lemma 1 states that a consensus set whose points have less than $k+1$ different x-coordinates is not maximum. Figure 4 illustrates Lemma 1 in the primal space. Therefore, we need not consider such consensus sets in proving Theorem 2. We now give the proof of Theorem 2.

Proof If $P_{C_{\max}}$ is bounded, then each of its vertices corresponds to an element of $\mathbb{G}_{S,k,w}$, from which Theorem 2 is immediately obtained. Therefore, we only have to show that $P_{C_{\max}}$ is bounded.



(a) Consensus set having less than $k+1$ x-coordinates (not maximum).

(b) Consensus set having $k+1$ x-coordinates (possibly maximum).

Fig. 4 Illustration of Lemma 1 in the primal space. Assume $k=2$. The consensus set C (*red points*) in (**a**) is not maximum since it has only two ($< k+1$) x-coordinates; there exists a possible consensus set $C' \supset C$ having not less than $k+1$ x-coordinates as shown in (**b**).

Since $\mathbb{G}_{S,k,w}$ is not empty, there exist at least $k+1$ points $(u_1, v_1), \ldots, (u_{k+1}, v_{k+1}) \in C_{\max}$ such that $u_i \neq u_j$ for all $i \neq j$ thanks to Lemma 1. Any $(a_0, \ldots, a_k)$ in $P_{C_{\max}}$ satisfies

$$\begin{cases} 0 \leq -\sum_{i=0}^{k} u_1^i a_i + v_1 \leq w \ , \\ \qquad\qquad \vdots \\ 0 \leq -\sum_{i=0}^{k} u_{k+1}^i a_i + v_{k+1} \leq w \ , \end{cases} \tag{12}$$

which can be rewritten as

$$\begin{cases} -\sum_{i=0}^{k} u_1^i a_i + v_1 = b_1 \ , \\ \qquad\qquad \vdots \\ -\sum_{i=0}^{k} u_{k+1}^i a_i + v_{k+1} = b_{k+1} \ , \end{cases} \tag{13}$$

where $0 \leq b_j \leq w$ $(j = 1, \ldots, k+1)$. We thus obtain

$$\begin{pmatrix} a_0 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} 1 & u_1 & \cdots & u_1^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & u_k & \cdots & u_k^k \\ 1 & u_{k+1} & \cdots & u_{k+1}^k \end{pmatrix}^{-1} \begin{pmatrix} v_1 - b_1 \\ \vdots \\ v_k - b_k \\ v_{k+1} - b_{k+1} \end{pmatrix} . \tag{14}$$

We remark that the inverse matrix always exists. Denoting the (i, j) entry of the inverse matrix by m_{ij} allows (14) to be written as

$$a_{i-1} = \sum_{j=1}^{k+1} m_{ij}(v_j - b_j) \ \ (i = 1, \ldots, k+1) \ . \tag{15}$$

(15) shows that a_i is linear in $b_1, \ldots, b_{k+1}$. Therefore, the set of $(a_0, \ldots, a_k)$ satisfying (12) is bounded since $0 \leq b_j \leq w$. $P_{C_{\max}}$ is its subset, and consequently is bounded. □

Theorem 2 states that the consensus sets $\{S \cap D : D \in \mathbb{G}_{S,k,w}\}$ contain all the maximum consensus sets. Therefore, if $\mathbb{G}_{S,k,w}$ is not empty, then all the maximum consensus sets (optimal solutions to our problem) are found by a brute-force search. We remark that such maximality cannot be guaranteed with a continuous model. Hereafter, we assume that $\mathbb{G}_{S,k,w}$ is not empty, which almost always holds.

4 Discrete polynomial curve fitting algorithm

RANSAC iteratively generates model parameters by randomly sampling points from a given set to find the ones describing a largest number of points in the set. Finding all the maximum consensus sets by RANSAC requires to compute the consensus sets for all the discrete polynomial curves of $\mathbb{G}_{S,k,w}$, which is computationally expensive and impractical. In fact, the brute-force search requires up to $2^{k+1}\binom{|S|}{k+1}$ iterations. So the number of iterations is usually fixed with a sufficiently large number. RANSAC, however, does not guarantee any property on its obtained result after such iterations. In this section, we propose a method that effectively achieves a solution guaranteeing local optimality in the sense of the set inclusion by introducing a local search.

We define neighbors in $\mathbb{G}_{S,k,w}$ for our local search. When $D \in \mathbb{G}_{S,k,w}$ is given, we define neighbors of D denoted by N_D as the discrete polynomial curves of $\mathbb{G}_{S,k,w}$ having k upper

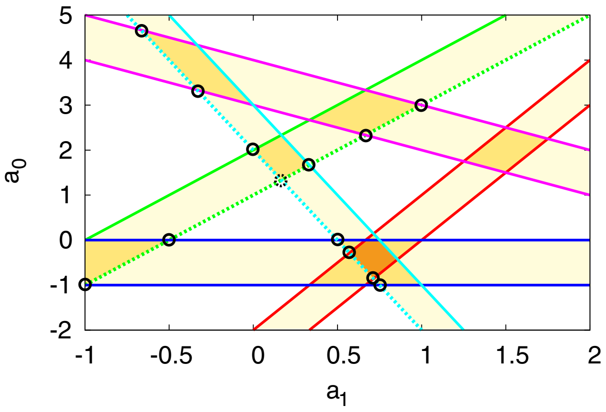


Fig. 5 An example of neighbors ($k = 1$). The neighbors of the dotted-circle point are depicted with solid-circle points. They are on the neighboring lines, i.e., dotted lines passing through the dotted-circle point.

and lower critical points all of which are identical with those of D where the x-coordinates of the critical points are different from each other. Note that $D \notin N_D$. Then, $(a_0, \dots, a_k)$ of $D' \in N_D$ satisfies the same k independent equations as that of D among the $2n$ equations in (8). We remark that a neighbor can be easily computed by solving in $(a_0, \dots, a_k)$ the system consisting of these k equations and another equations in (8). Therefore, $(a_0, \dots, a_k)$ of D' is on the intersection line of the k hyperplanes that are the feasible boundaries identified by these equations. Thus, the neighboring relations are determined by the intersection lines of k feasible boundaries. We call these lines *neighboring lines*. Figure 5 shows an example of neighbors in the parameter space when $k = 1$. In this case, the neighboring lines are identical to the feasible boundaries themselves. We call D' having at least the same number of inliers a *good* neighbor of D.

Our method consists of two steps (Algorithm 1). In the first step, we use RANSAC to obtain a seed for the second step. In the second step, we introduce a local search, called *rock climbing*, to increase the number of inliers. Given an initial seed (discrete polynomial curve) obtained by RANSAC, rock climbing searches the discrete polynomial curves having a largest number of inliers among the seed and its neighbors, and then iterates this procedure using the obtained curves as new seeds. We remark that if there is more than one discrete polynomial curve with the same largest number of inliers, then rock climbing uses all of them as new seeds in the next iteration (this is where rock climbing differs from the method of gradient ascent). Algorithm 2 describes the concrete procedure of rock climbing. We remark that rock climbing searches discrete polynomial curves of degree *up to* k.

Algorithm 1 Our method

```
Input: A set of discrete points $S$, a degree $k$, a width $w$, a number of iterations $t$
       for RANSAC.
Output: A set of discrete polynomial curves.
  Run RANSAC with $t$ iterations.
  Run rock climbing using a seed obtained by RANSAC.
  return The output of rock climbing.
```

Algorithm 2 Rock climbing

```
Input: $S$, $k$, $w$, an initial discrete polynomial curve $D_{\text{init}} \in \mathbb{G}_{S,k,w}$.
Output: A set $A$ of discrete polynomial curves.
  $A := \{D_{\text{init}}\}$
  loop
    $A'$ :=A set of discrete polynomial curves in $\left(A \cup \bigcup_{D \in A} N_D\right)$ having a largest number
    of inliers
    if $A = A'$
      Break out of the loop
    else
      $A := A'$
    end if
end loop
return $A$
```

Remark 1 The coefficients of a discrete polynomial curve are obtained by solving the linear equation system in eq. (7). The condition number of a Vandermonde matrix, however, exponentially increases with the size of the matrix [5]. In theory, therefore, a numerical solution obtained by our method is not necessarily stable, i.e., a small change in s_j $(j = 1, \ldots, k+1)$ may cause a significant change in a_i $(i = 0, \ldots, k)$, when k is large. Though the numerical evaluation in stability of obtained solutions is beyond the scope of this paper, we experimentally observe that the solution obtained by our method is numerically stable for degrees $k = 2$, 3 and 4. Accordingly, our method is effective from the practical point of view because fitting polynomial curves of high degrees is not in a great demand.

A consensus set C is called *local maximum* when no consensus set exists that is a superset of C. We denote a local maximum consensus set by C_{local}.

Theorem 3 *Rock climbing outputs discrete polynomial curves that correspond to all the vertices of a $P_{C_{\text{local}}}$, and therefore we can generate all $(a_0, \ldots, a_k)$'s of D such that $S \cap D = C_{\text{local}}$ from those obtained vertices.*

Proof Let C be the consensus set of the current discrete polynomial curve ($\in \mathbb{G}_{S,k,w}$).

We first consider the case of $C = C_{\text{local}}$. Any two vertices of a convex polytope are reachable with each other by tracing edges of the polytope. This means that we can obtain all the vertices of $P_{C_{\text{local}}}$ by propagating the neighboring relation from the current vertex, since each edge of P_C is a part of a neighboring line. Furthermore, any $(a_0, \ldots, a_k)$ in $P_{C_{\text{local}}}$ satisfies $F(a_0, \ldots, a_k) = |C_{\text{local}}|$. Consequently, we can obtain all the vertices of $P_{C_{\text{local}}}$ by iteratively searching good neighbors.

If $C \neq C_{\text{local}}$, then a consensus set $C' = C \cup (x', y')$ exists where $(x', y') \in S \backslash C$. $P_{C'}$ is the intersection of P_C and the feasible region for (x', y'). Therefore, each vertex of $P_{C'}$ is on an edge or a vertex of P_C as illustrated in Fig. 6. This means that we can obtain all the vertices of $P_{C'}$ by propagating the neighboring relation from the current

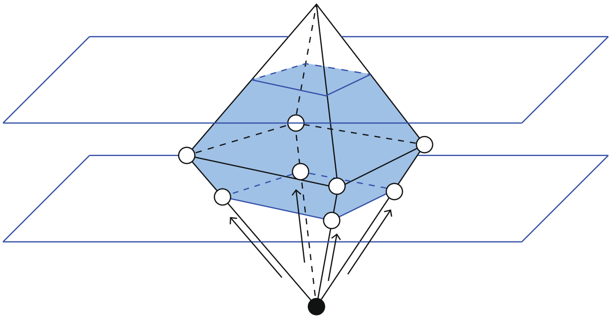
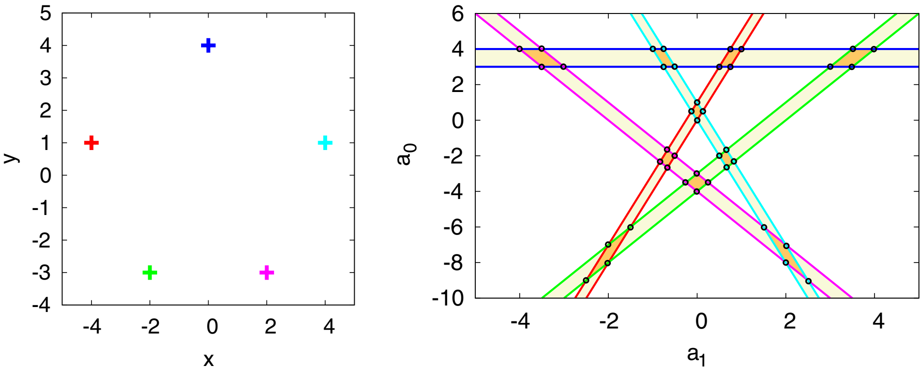


Fig. 6 P_C (black) and $P_{C'}$ (blue). Note that the two parallel planes are feasible boundaries. Each vertex of $P_{C'}$ is on an edge or a vertex of P_C. Suppose that the black point corresponds to the current polynomial curve. Then the white points are the neighbors in P_C.

vertex of P_C. Furthermore, any $(a_0, \dots, a_k)$ in P_C satisfies $F(a_0, \dots, a_k) \geq |C|$. Consequently, we can obtain all the vertices of $P_{C'}$ by iteratively searching good neighbors. This discussion holds as long as $C \neq C_{\text{local}}$. By repeating this procedure, we finally obtain $C' = C_{\text{local}}$. □

From Theorem 3, rock climbing guarantees local maximality of an obtained consensus set in the sense of set inclusion. It should be noted that our method does not always terminate immediately at a local optimal consensus set. Rock climbing examines every neighbor to seek good ones, and does not terminate as long as good neighbors exist.

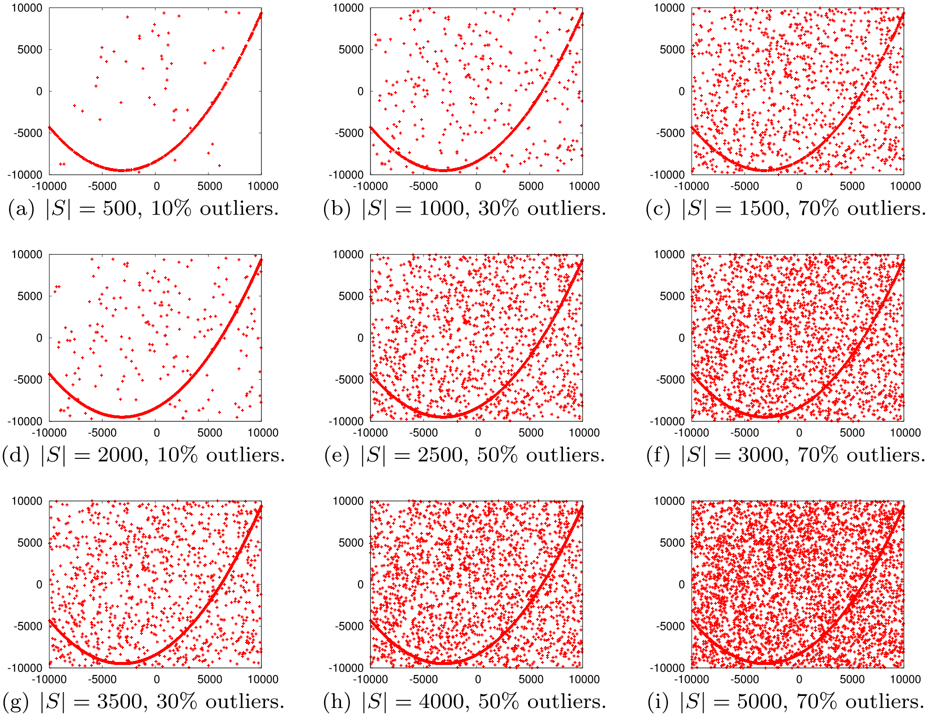
Rock climbing has possibilities of not achieving a global optimum. Its output depends on an initial seed. Having a "good" seed will be preferable. That is why we use RANSAC to obtain an initial seed having as many inliers as possible. We will experimentally demonstrate this issue in Section 7.



(a) S in the primal space.

(b) S in the parameter space.

Fig. 7 An example of S for which every element of $\mathbb{G}_{S,k,w}$ has only $k+1$ inliers ($k = 1, w = 1$). (**a**) shows the S in the primal space, while (**b**) shows the corresponding feasible regions in the parameter space. The *black points* in the parameter space correspond to the discrete polynomial curves in $\mathbb{G}_{S,k,w}$. We can see that they all have only two inliers. For such S, rock climbing evaluates all the discrete polynomial curves in $\mathbb{G}_{S,k,w}$.



(a) $|S| = 500$, 10% outliers.
(b) $|S| = 1000$, 30% outliers.
(c) $|S| = 1500$, 70% outliers.
(d) $|S| = 2000$, 10% outliers.
(e) $|S| = 2500$, 50% outliers.
(f) $|S| = 3000$, 70% outliers.
(g) $|S| = 3500$, 30% outliers.
(h) $|S| = 4000$, 50% outliers.
(i) $|S| = 5000$, 70% outliers.

Fig. 8 Examples of input sets $S \in \mathcal{S}_2$ ($k = 2$).

5 Computational complexity of rock climbing

The computational complexity of rock climbing is given by the following theorem.

Theorem 4 *For a given k, rock climbing has the computational complexity of* $\mathcal{O}(|S|^{k+2})$.

Proof As indicated in Theorem 3, rock climbing does not terminate as long as a good neighbor exists. This means that rock climbing evaluates all the discrete polynomial curves in $\mathbb{G}_{S,k,w}$ when a given S has a property such that for any $k+1$ points in S, no other inliers exists for the discrete polynomial curve uniquely determined by the $k+1$ points (the $k+1$ points are critical points). Figure 7 shows an example of such S in the case of $k = 1$. We remark that any two elements of $\mathbb{G}_{S,k,w}$ are reachable from each other (by applying the neighboring relation at most $k+1$ times, i.e., by exchanging at most $k+1$ critical points). Therefore, the computational cost for rock climbing is proportional to the product

Table 1 Average value of $|C_{\text{initial}}|/|C_{\text{true}}|$ for each seed quality level ($k = 2$).

seed quality	$\lvert C_{\text{initial}}\rvert/\lvert C_{\text{true}}\rvert$
3/3	0.29
2/3	0.0047
1/3	0.0035
0/3	0.0035

of $|\mathbb{G}_{S,k,w}|$ and the cost of counting the number of inliers for one discrete polynomial curve. In general $|\mathbb{G}_{S,k,w}| = 2^{k+1}\binom{|S|}{k+1}$, since to determine one discrete polynomial curve requires selecting $k+1$ points among S with specifying whether each point is a lower or an upper critical point. The cost of counting the number of inliers, on the other hand, is $\mathcal{O}(|S|)$, because for each point $(x', y') \in S$ we need to verify whether $0 \leq y' - \sum_{i=0}^{k} x'^{i} a_i \leq w$ is satisfied. Accordingly, the computational complexity of rock climbing is

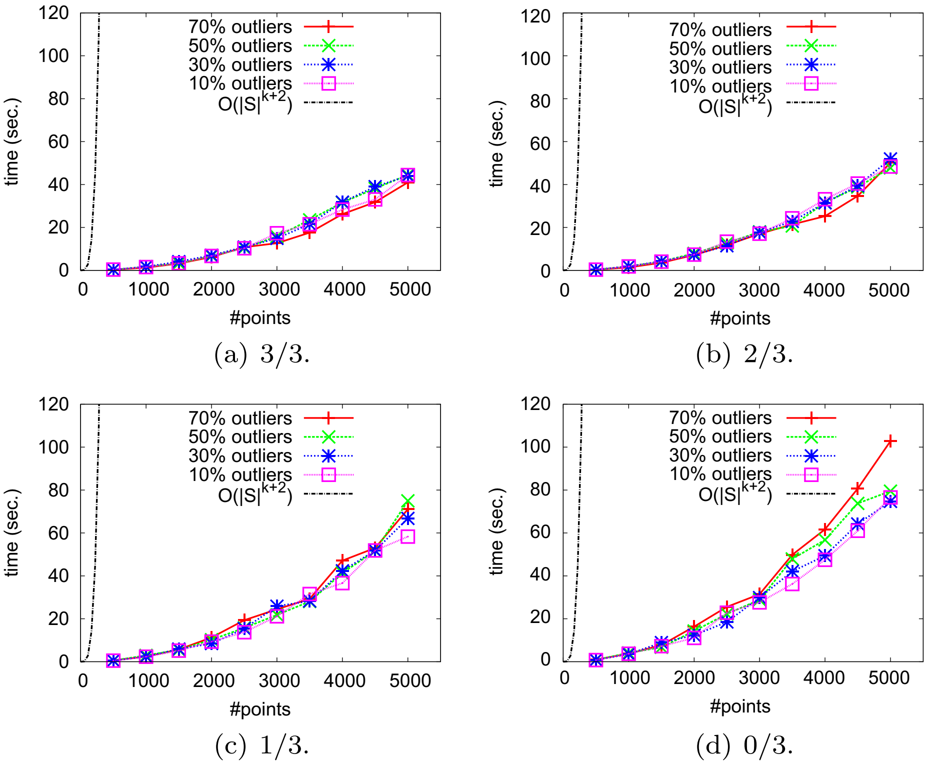
$$\begin{aligned}\mathcal{O}\left(|S|\right) \cdot |\mathbb{G}_{S,k,w}| &= \mathcal{O}\left(|S| \times 2^{k+1}\binom{|S|}{k+1}\right)\\ &= \mathcal{O}(|S|^{k+2}).\end{aligned}$$

□

Theorem 4 states that rock climbing itself has the same computational complexity as a brute-force search. However, the worst case cost is rarely achieved in practice. This is because rock climbing searches for only *good* neighbors. A discrete polynomial curve $D \in \mathbb{G}_{S,k,w}$ has $|N_D| = (k+1)\,(2|S| - (k+1))$ neighbors in general. This is because $k+1$ neighboring lines pass through $(a_0, \ldots, a_k)$ determining D and each of them has at most $2|S| - (k+1)$ intersections with other feasible boundaries. Therefore, for one current seed, the computational cost required to evaluate its neighbors is $\mathcal{O}(|S|) \times (k+1)\,(2|S| - (k+1)) = \mathcal{O}(|S|^2)$. The seed pool of rock climbing contains a single solution when it starts, and the pool is iteratively updated by equivalent or better solutions. If M denotes the number of discrete polynomial curves stored once in the pool while rock climbing is running, the computational cost practically required for rock climbing is $\mathcal{O}(M \cdot |S|^2)$. This indicates that the practical efficiency of rock climbing depends on M. Note that M depends on the property of input data and a given initial seed. This practical efficiency of rock climbing is demonstrated in the experiments in Section 7.

Table 2 Percentage of acquiring C_{true} for each pair of seed quality level and outlier ratio ($k = 2$).

seed quality	outlier ratio (%)	percentage of acquiring C_{true} (%)
3/3	10	100
	30	100
	50	100
	70	100
2/3	10	100
	30	100
	50	100
	70	100
1/3	10	99.3
	30	99.3
	50	96.7
	70	95.3
0/3	10	92.0
	30	94.7
	50	89.7
	70	95.7



(a) 3/3.

(b) 2/3.

(c) 1/3.

(d) 0/3.

Fig. 9 Average time until rock climbing converges at C_{true} for each seed quality level ($k = 2$).

6 Extension to surface fitting

Our proposed method for 2D discrete polynomial curve fitting can be straightforwardly used even for 3D discrete polynomial surface fitting. This section discusses the extension to 3D discrete polynomial surface fitting.

A (continuous) polynomial surface of degree k in the 3D Euclidean space is defined by

$$P^{3\text{D}} = \left\{ (x, y, z) \in \mathbb{R}^3 : z = \sum_{i=0}^{k} \sum_{j=0}^{k-i} a_{ij} x^i y^j, a_{i(k-i)} \neq 0 \text{ for } {}^{\exists} i \right\}, \tag{16}$$

where $a_{ij} \in \mathbb{R}$. As is the case of a discrete polynomial curve, a *discrete polynomial surface* in 3D, a discretization of (16), is described by

$$D^{3\text{D}} = \{(x, y, z) \in \mathbb{Z}^3 : 0 \leq z - g(x, y) \leq w\}, \tag{17}$$

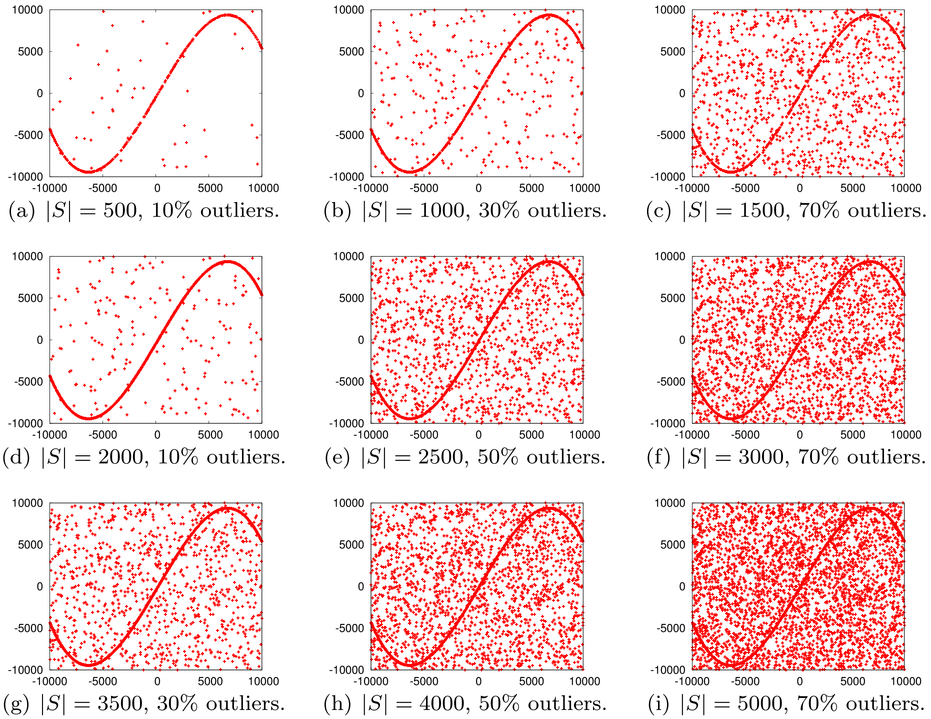
where $g(x, y) = \sum_{i=0}^{k} \sum_{j=0}^{k-i} a_{ij} x^i y^j$, and w is a constant uniquely determined as the absolute difference between the z-coordinates of two adjacent points in the discrete space.

We align coefficients a_{ij} in (16) and rename them to have $a_0, a_1, \ldots, a_{d_{k-1}}$, where d_k is the number of a_{ij}'s. Let $\mathbb{D}^{3\text{D}}_{k,w}$ be the set of all 3D discrete polynomial surfaces of degree k with a width w. Then the discrete polynomial surface fitting problem is given below.

Input A set of discrete points $S \subset \mathbb{Z}^3$, a degree k, and a width w.

Table 3 Variance of convergence time (sec.) of rock climbing for each seed level ($k = 2$).

S		seed quality			
#point	outlier ratio (%)	3/3	2/3	1/3	0/3
500	10	0.10	0.12	0.20	0.23
	30	0.13	0.13	0.18	0.38
	50	0.11	0.12	0.34	0.41
	70	0.11	0.11	0.41	0.54
1000	10	0.46	0.60	0.58	2.72
	30	0.59	0.62	0.72	2.69
	50	0.53	0.55	1.90	2.74
	70	0.55	0.49	1.20	2.64
1500	10	1.21	1.40	1.65	5.43
	30	1.63	1.60	3.17	6.30
	50	0.93	1.58	2.04	2.11
	70	0.89	1.20	2.78	3.51
2000	10	2.21	2.28	2.89	3.17
	30	2.72	1.88	2.23	3.48
	50	2.87	2.95	3.46	7.14
	70	1.95	2.92	5.16	7.98
2500	10	4.31	4.22	3.60	23.29
	30	3.03	4.17	3.24	5.69
	50	3.12	4.87	6.17	8.82
	70	3.79	4.24	7.13	12.14
3000	10	5.04	4.59	6.00	8.19
	30	3.79	5.59	8.87	10.20
	50	5.19	6.16	6.26	8.94
	70	4.03	5.72	10.73	15.46
3500	10	6.87	6.56	7.28	11.77
	30	6.84	7.11	9.50	14.04
	50	8.07	6.74	7.08	18.69
	70	4.80	7.08	8.04	24.82
4000	10	8.88	11.29	7.20	12.05
	30	11.05	9.72	12.89	15.43
	50	11.44	8.36	11.52	16.53
	70	7.99	7.77	33.22	31.09
4500	10	11.03	13.34	24.02	12.68
	30	12.22	13.28	14.71	22.14
	50	13.06	12.90	12.38	23.80
	70	11.62	11.85	18.87	49.25
5000	10	15.63	13.62	15.09	24.75
	30	14.09	15.28	18.63	17.07
	50	17.33	15.73	25.34	28.80
	70	15.18	14.28	34.79	44.96



(a) $|S| = 500$, 10% outliers.
(b) $|S| = 1000$, 30% outliers.
(c) $|S| = 1500$, 70% outliers.
(d) $|S| = 2000$, 10% outliers.
(e) $|S| = 2500$, 50% outliers.
(f) $|S| = 3000$, 70% outliers.
(g) $|S| = 3500$, 30% outliers.
(h) $|S| = 4000$, 50% outliers.
(i) $|S| = 5000$, 70% outliers.

Fig. 10 Examples of input sets $S \in \mathcal{S}_3$ ($k = 3$).

Output A d_k-tuple $(a_0, a_1 \ldots, a_{d_{k-1}})$ of coefficients of $D^{3\text{D}} \in \mathbb{D}^{3\text{D}}_{k,w}$ having the maximum number of inliers.

In the case of 3D discrete polynomial surface fitting, input points are in 3D and the surface to be searched runs over the xy-plane. This seems more complex and difficult than 2D discrete polynomial curve fitting where input points are in 2D and the curve runs over the x-axis; however, it is not true. Indeed, (17) is linear with respect to parameters like (2). This indicates that we can completely have the same discussion on 3D discrete polynomial surface fitting as 2D discrete polynomial curve fitting (the only difference is the number of parameters involved). Rock climbing can thus be directly applied to the 3D discretely polynomial surface fitting problem. As the counterparts of Theorems 2, 3 and 4, we

Table 4 Average value of $|C_{\text{initial}}|/|C_{\text{true}}|$for each quality of initial seeds ($k = 3$).

seed quality	$\lvert C_{\text{initial}}\rvert/\lvert C_{\text{true}}\rvert$
4/4	0.25
3/4	0.0069
2/4	0.0050
1/4	0.0047
0/4	0.0046

have straightforwardly the following corollaries. Let $\mathbb{G}^{3\mathrm{D}}_{S,k,w}$ be the set of all $D^{3\mathrm{D}} \in \mathbb{D}^{3\mathrm{D}}_{k,w}$ generated from d_k points in S where the d_k points are used as critical points.

Corollary 1 *If $\mathbb{G}^{3\mathrm{D}}_{S,k,w}$ is not empty, then there exists $D^{3\mathrm{D}} \in \mathbb{G}_{S,k,w}$ such that $S \cap D^{3\mathrm{D}} = C_{\mathrm{max}}$.*

Corollary 2 *Rock climbing, when applied to 3D discrete polynomial surface fitting, outputs 3D discrete polynomial surfaces that correspond to all the vertices of a $P_{C_{\mathrm{local}}}$.*

Corollary 3 *Rock climbing, when applied to 3D discrete polynomial surface fitting, has the computational complexity of $\mathcal{O}(|S|^{d_k+1})$.*

7 Experiments

In this section, we apply rock climbing to synthesized data-sets. We first evaluate the performance of rock climbing (Algorithm 2) depending on the quality of an initial seed. We next evaluate the computational speed of our method using initial seeds obtained by RANSAC (Algorithm 1).

Table 5 Percentage of acquiring $C_{\mathrm{true}}$for each pair of seed quality level and outlier ratio ($k = 3$).

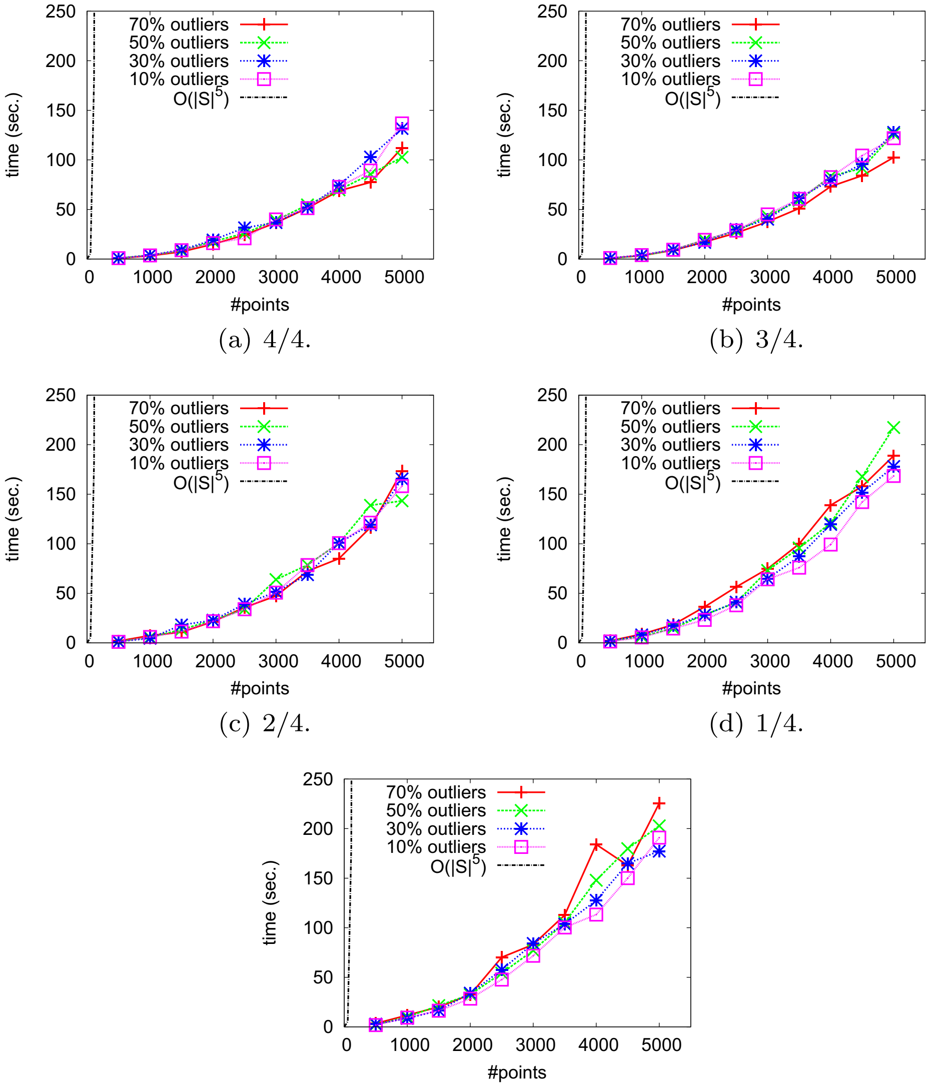
seed quality	outlier ratio (%)	percentage of acquiring C_{true} (%)
4/4	10	100
	30	100
	50	100
	70	100
3/4	10	100
	30	100
	50	99.7
	70	100
2/4	10	99.3
	30	99.3
	50	99.0
	70	96.3
1/4	10	98.3
	30	97.7
	50	96.7
	70	96.0
0/4	10	99.0
	30	98.7
	50	95.3
	70	92.0

Table 6 Variance of convergence time (sec.) of rock climbing for each seed quality level ($k = 3$).

S		seed quality				
#point	outlier ratio (%)	4/4	3/4	2/4	1/4	0/4
500	10	0.26	0.44	0.37	0.54	0.73
	30	0.38	0.42	0.49	0.76	1.36
	50	0.37	0.28	0.63	0.96	0.99
	70	0.31	0.31	2.68	1.29	2.20
1000	10	1.14	1.30	2.09	1.54	4.38
	30	1.40	1.29	1.24	2.97	4.69
	50	1.33	1.47	2.23	1.91	5.81
	70	1.61	1.17	6.23	6.48	8.83
1500	10	2.98	2.47	3.94	4.57	6.75
	30	1.99	4.06	22.77	7.18	7.75
	50	3.27	2.23	5.40	5.02	8.97
	70	2.39	2.97	4.35	11.43	8.80
2000	10	5.31	5.81	6.65	5.46	8.87
	30	5.36	5.31	7.32	8.72	19.78
	50	5.29	5.63	7.88	10.92	10.27
	70	5.06	6.60	9.34	31.44	9.81
2500	10	6.96	8.01	9.23	10.32	15.65
	30	8.65	9.89	14.04	13.66	27.26
	50	8.57	9.69	10.06	9.78	24.70
	70	7.70	7.67	18.80	48.01	37.15
3000	10	14.20	13.76	13.65	22.32	20.93
	30	12.97	11.93	15.55	19.69	51.10
	50	10.01	10.90	42.03	25.42	37.40
	70	11.24	10.95	13.77	57.35	31.02
3500	10	13.14	16.45	26.78	16.21	20.72
	30	12.22	14.70	19.51	18.48	56.71
	50	16.53	17.25	27.39	34.43	38.90
	70	15.30	16.00	30.06	53.16	53.77
4000	10	21.44	22.96	24.64	25.30	32.31
	30	21.87	25.97	26.38	27.42	43.17
	50	24.32	22.71	57.03	38.85	81.72
	70	20.45	23.28	31.77	134.31	85.80
4500	10	29.43	29.79	22.05	42.51	64.77
	30	29.54	30.91	31.77	38.36	52.20
	50	25.43	29.19	61.94	53.52	53.37
	70	24.95	29.63	46.17	60.12	51.46
5000	10	45.92	37.31	49.82	55.88	55.19
	30	44.76	32.40	45.41	52.58	44.67
	50	34.22	37.06	40.80	108.65	57.34
	70	26.17	30.70	144.36	59.71	106.59

7.1 Performance of rock climbing depending on initial seed quality

For 2D discrete polynomial curves, we first set $k = 2$, $w = 1$. We fixed the ratio of outliers among input points to be 10 %, 30 %, 50 %, 70 %, and for each fixed ratio we generated ten different input point sets S, where $|S|$ was changed by 500 from 500 to 5000 (see Fig. 8 for examples). In each S, integer points satisfying $0 \leq y - (1.0976 \times 10^{-4}x^2 + 0.6840x - 8.4283 \times 10^3) \leq w$ $(-10000 \leq x \leq 10000)$ were randomly generated and named as *true inliers*, and integer points that do not satisfy this inequality were randomly generated within



(a) 4/4.

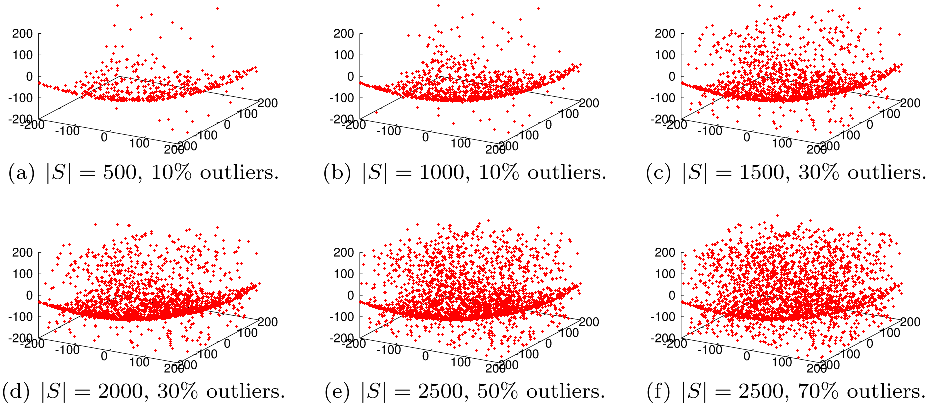
(b) 3/4.

(c) 2/4.

(d) 1/4.

(e) 0/4.

Fig. 11 Average time until convergence at C_{true} for each seed quality level ($k = 3$).



(a) $|S| = 500$, 10% outliers. (b) $|S| = 1000$, 10% outliers. (c) $|S| = 1500$, 30% outliers.

(d) $|S| = 2000$, 30% outliers. (e) $|S| = 2500$, 50% outliers. (f) $|S| = 2500$, 70% outliers.

Fig. 12 Examples of input sets $S \in \mathcal{S}_2^{3D}$ (3D, $k = 2$).

$[-10000, 10000] \times [-10000, 10000]$ and named as *true outliers*. We denote the set of these input point sets by $\mathcal{S}_2$.

For applying rock climbing to these data, we generated initial seeds of different qualities, to evaluate the effect of using a "good" initial seed. We define the quality of an initial seed by the number of true inliers used for generating it. For example, an initial seed of the highest quality is generated only from true inliers, while an initial seed of the lowest quality is generated only from true outliers. We denote the quality level by a fraction $\sharp(\text{true inliers})/(k+1)$, where the denominator is the number of points required for determining a discrete polynomial curve of $\mathbb{G}_{S,k,w}$. For $k = 2$, we have four quality levels of an initial seed, namely, 3/3, 2/3, …, 0/3. For each S, we randomly generated 30 initial seeds at each quality level. Table 1 compares the average ratio of inliers to all true inliers under different quality levels. Note that C_{initial} denotes the consensus set of an initial seed. We can see that initial seeds of high quality have a large number of inliers, which justifies our definition of seed quality in our used data-sets.

We then applied rock climbing to all S using the initial seeds generated above. Table 2 shows for each pair of seed quality level and outlier ratio, the percentage of acquiring all the true inliers C_{true} over the whole trials. We can see that initial seeds of high quality always obtained C_{true} regardless of outlier ratio, while initial seeds of low quality sometimes converged at other C_{local}.

Table 7 Average value of $|C_{\text{initial}}|/|C_{\text{true}}|$ for each quality of initial seeds (3D, $k = 2$).

seed quality	$\lvert C_{\text{initial}}\rvert/\lvert C_{\text{true}}\rvert$
6/6	0.23
5/6	0.029
4/6	0.017
3/6	0.014
2/6	0.013
1/6	0.013
0/6	0.013

Table 8 Percentage of acquiring $C_{\mathrm{true}}$for each pair of seed quality level and outlier ratio ($k = 3$).

seed quality	outlier ratio (%)	percentage of acquiring C_{true} (%)
6/6	10	100
	30	100
	50	100
	70	100
5/6	10	100
	30	100
	50	100
	70	100
4/6	10	100
	30	100
	50	99.7
	70	99.7
3/6	10	100
	30	100
	50	99.7
	70	98.3
2/6	10	100
	30	99.3
	50	100
	70	99.7
1/6	10	100
	30	100
	50	100
	70	99.3
0/6	10	100
	30	99.7
	50	98.6
	70	96.7

Figure 9 shows for each seed quality level, the relation between the number of input points $|S|$ and the average time required to converge at C_{true}. Note that convergence times for other C_{local} were not counted here. In each graph, we depicted the average convergence times for the four different outlier ratios. For comparison, we also depicted the estimated time assuming the worst case. The estimated time was computed as $1.4 \times 10^{-8} \times |S| \times |\mathbb{G}_{S,k,w}|$ $\left(= \mathcal{O}(|S|^{k+2})\right)$ where $k = 2$. This is because that our computer [1] requires about 1.4×10^{-8} seconds to verify whether $(x, y) \in S$ is an inlier of a given $D \in \mathbb{G}_{S,k,w}$ when $k = 2$. From these graphs, we can see that rock climbing is in practice efficient regardless of outlier ratio, comparing with the estimated time. We can also see that using

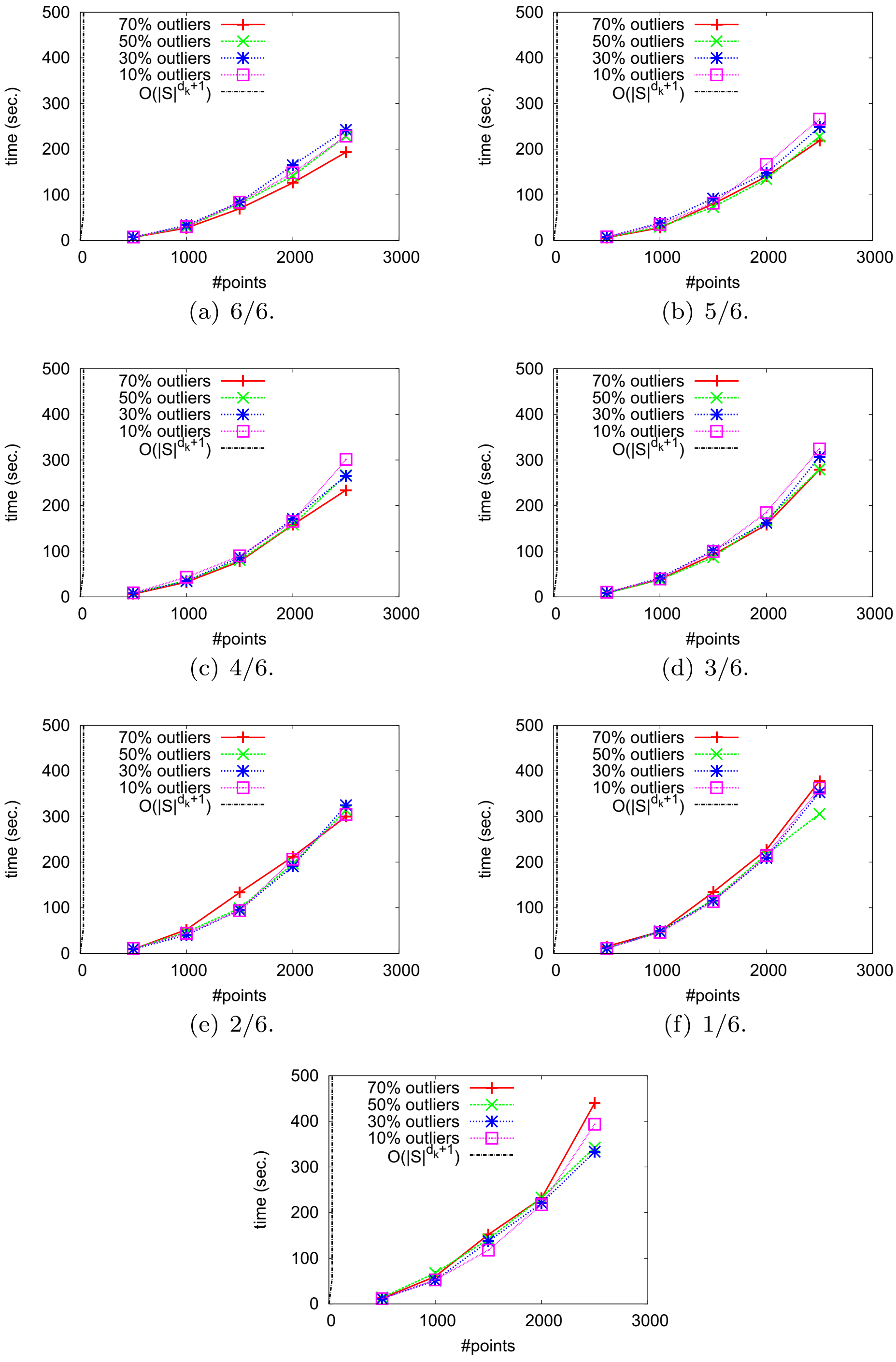
[1] CPU: Intel Core i7-3930K Processor (3.2GHz, 6-Cores, 12-Threads), memory: 16GB.

an better initial seed results in even more efficiency. Variances of the convergence time under different seed-quality levels for each S are shown in Table 3. From Table 3, we can see that the computational time required for a low-quality seed tends to be more unstable (has a larger variance) than that required for a high-quality seed. We can also see that the computational time required for a low-quality seed tends to have a larger variance when the outlier ratio is high, while that required for a high-quality seed is less affected by the outlier ratio. The variance becomes greater as the number of input points increases.

So far, we had experiments for quadratic curves ($k = 2$). In order to confirm our observations for another degree case, we executed the same experiments under the condition of $k = 3$ and $w = 1$. As input sets, we randomly generated true inliers satisfying $0 \leq y - (-1.6674 \times 10^{-8} x^3 + 1.0183 \times 10^{-5} x^2 + 2.1502x - 477.58) \leq w$ $(-10000 \leq x \leq 10000)$ and true outliers over $[-10000, 10000] \times [-10000, 10000]$ so that no true outlier satisfies this inequality (see Fig. 10 for examples). We denote the set of these input point sets by $\mathcal{S}_3$. The results are shown in Tables 4, 5, 6 and Fig. 11. Note that when $k = 3$, we have five quality levels of an initial seed, namely, 4/4, 3/4, …, 0/4. The estimated time in Fig. 11 was computed as $1.7 \times 10^{-8} \times |S| \times |\mathbb{G}_{S,k,w}|$ $\left(= \mathcal{O}(|S|^{k+2})\right)$ where $k = 3$, since our computer requires about 1.7×10^{-8} seconds to verify whether $(x, y) \in S$ is an inlier of a given $D \in \mathbb{G}_{S,k,w}$ when $k = 3$. From these results, we have the same observation as the quadratic curves case.

Table 9 Variance of convergence time (sec.) of rock climbing for each seed quality level (3D, $k = 2$).

S		seed quality						
#point	outlier ratio (%)	6/6	5/6	4/6	3/6	2/6	1/6	0/6
500	10	3.10	3.01	2.99	2.90	2.42	3.05	2.75
	30	2.23	2.76	1.98	2.18	2.49	4.61	3.83
	50	2.11	2.00	2.65	2.47	4.05	3.55	5.80
	70	2.47	2.03	1.90	3.75	3.37	9.51	5.33
1000	10	7.92	10.22	12.43	9.81	10.16	14.40	12.57
	30	12.60	10.47	8.07	10.81	8.83	14.89	20.28
	50	8.96	11.20	10.22	13.29	14.08	13.17	34.63
	70	8.67	8.31	12.75	10.62	19.05	17.47	33.67
1500	10	24.76	25.40	26.48	21.28	23.28	17.85	32.70
	30	24.69	31.08	25.24	35.63	25.36	33.15	33.79
	50	23.56	18.35	25.44	22.90	32.34	34.69	45.17
	70	20.46	22.76	23.41	43.10	73.41	50.19	93.03
2000	10	41.41	39.25	42.28	46.79	49.94	51.47	57.48
	30	48.07	35.45	41.25	44.65	36.08	44.80	48.78
	50	43.34	37.02	30.56	54.42	62.29	51.11	59.29
	70	36.62	52.45	66.11	44.27	82.55	89.80	53.56
2500	10	64.83	77.25	72.26	101.12	56.91	110.46	81.35
	30	52.27	67.35	66.57	87.29	77.60	106.30	75.10
	50	76.50	48.78	60.37	65.89	94.13	101.09	121.42
	70	52.31	48.70	71.14	72.12	85.91	195.22	314.81



(a) 6/6.

(b) 5/6.

(c) 4/6.

(d) 3/6.

(e) 2/6.

(f) 1/6.

(g) 0/6.

Fig. 13 Average time until convergence at C_{true} for each seed quality level (3D, $k = 2$).

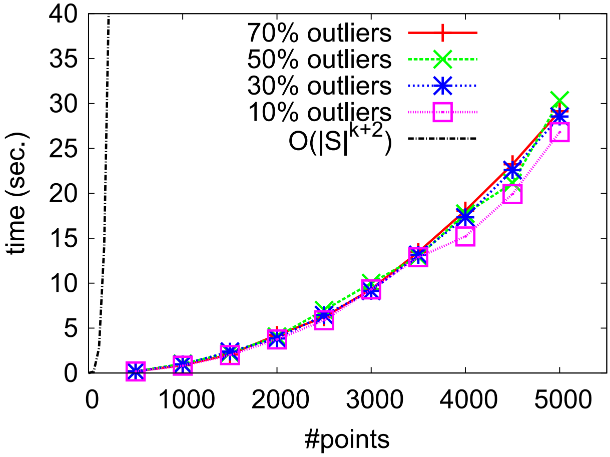


Fig. 14 Average time until our method converges at C_{true} $(k = 2)$.

We finally executed the same experiments for a discrete polynomial surface to evaluate the 3D case. We set $k = 2$ and $w = 1$. We generated five different input point sets S for each fixed outlier ratio 10 %, 30 %, 50 % and 70 %, where $|S|$ was changed by 500 from 500 to 2500 (see Fig. 12 for examples). In each S, we randomly generated true inliers satisfying $0 \leq z - (0.002x^2 - 0.001xy + 0.002y^2 + 0.25x - 0.25y - 150) \leq w$ $(-100 \leq x, y \leq 100)$ and true outliers over $[-100, 100] \times [-100, 100] \times [-100, 100]$ so that no true outlier satisfies this inequality. We denote the set of these input point sets by $\mathcal{S}_2^{3\text{D}}$. The results are shown in Tables 7, 8, 9 and Fig. 13. Note that we input $k = 2$, and for a 3D discrete polynomial surface of degree 2, we have seven quality levels of an initial seed, namely, 7/7, 6/7, …, 0/7. The estimated time in Fig. 13 is computed as $4.5 \times 10^{-8} \times |S| \times |\mathbb{G}^{3\text{D}}_{S,k,w}|$ $\left(= \mathcal{O}(|S|^{d_k+1})\right)$ where $k = 2$, since our computer requires about 4.5×10^{-8}

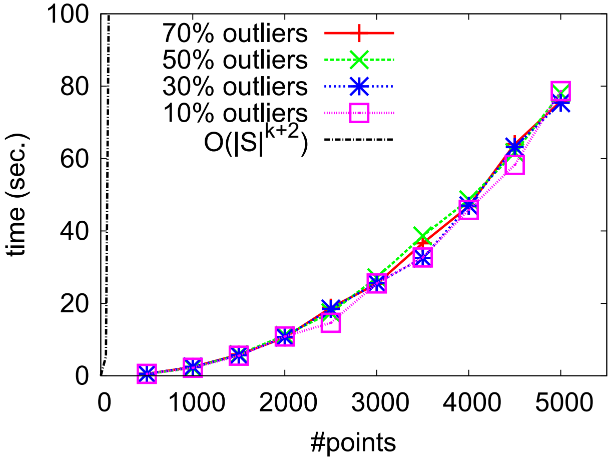


Fig. 15 Average time until our method converges at C_{true} $(k = 3)$.

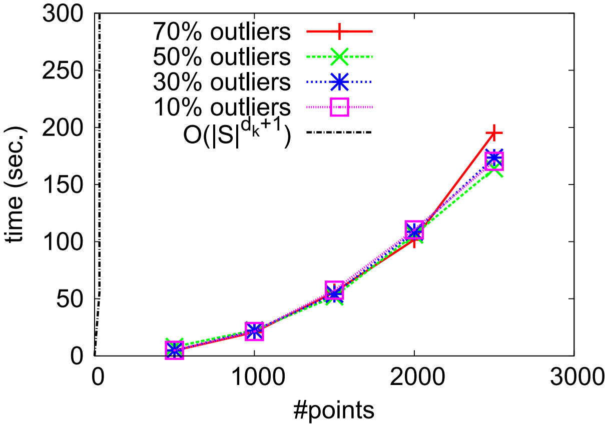


Fig. 16 Average time until our method converges at C_{true} (3D, $k = 2$).

seconds to verify whether $(x, y, z) \in S$ is an inlier of a given $D \in \mathbb{G}^{3D}_{S,k,w}$ when $k = 2$. From these results, we have the same observation as the 2D discrete polynomial curves cases.

7.2 Computational speed of our method

Now we evaluate the computational speed of our method, by applying rock climbing to all S generated in the previous subsection using initial seeds obtained by RANSAC.

We first applied our method to each $S \in \mathcal{S}_2$ (Fig. 8, for examples) 100 times independently, where for all S we input $k = 2$, $w = 1$ and $t = 1000$ (the number of iterations for RANSAC). We obtained C_{true} in every trial for all S. The average convergence time for each S is shown in Fig. 14, where the computational time required for RANSAC is included (this is far smaller than that required for rock climbing). We can see that the computational times are in general slightly shorter than those of rock climbing using the highest-quality initial seeds in the previous subsection (see Fig. 9). This indicates that using RANSAC to generate an initial seed for rock climbing is effective.

We next conducted the same experiments on $\mathcal{S}_3$, where for all S we input $k = 3$, $w = 1$ and $t = 1000$. We obtained C_{true} in every trial for all S. The average convergence time for each S is shown in Fig. 15. From these results, we have the same observation as the quadratic curves case.

We finally conducted the same experiments on $\mathcal{S}^{3D}_2$, where for all S we input $k = 2$, $w = 1$ and $t = 10000$. We obtained C_{true} in every trial for all S. Figure 16 shows the average convergence time for each S. From these results, we have the same observation as the 2D discrete polynomial curves cases.

8 Conclusion

This paper dealt with the problem of fitting a discrete polynomial curve and surface to a given set of discrete points in the presence of outliers. We formulated this problem as a discrete optimization problem where the number of inliers is maximized. We then proposed a method, called rock climbing, that iteratively improves a solution using local search. Our proposed method effectively searches solutions of the formulated problem by rock climbing using an initial seed obtained by RANSAC. We showed that our method guarantees local maximality of the obtained solution in the sense of the set inclusion. Our intensive experiments demonstrated the effectiveness of our proposed method. In particular, we observed that (i) with a good initial seed, rock climbing stably obtains a good solution even in the presence of various levels of outliers, and (ii) the computational cost for rock climbing to achieve the local optimal is cheap in practice.

Acknowledgments This work is in part supported by Grant-in-Aid for Scientific Research of the Ministry of Education, Culture, Sports, Science and Technology of Japan and by a Japanese-French joint research project called the SAKURA program.

References

1. Abdoulaye, S., Oumarou, S., Andres, E.: Extended standard hough transform for analytical line recognition. Int. J. (2013)
2. Aiger, D., Kenmochi, Y., Talbot, H., Buzer, L.: Efficient robust digital hyperplane fitting with bounded error: In: Proceedings of the 16th IAPR international conference on Discrete Geometry for Computer Imagery (DGCI2011), LNCS, Vol. 6607, pp. 223–234. Springer (2011)
3. Andres, E.: Discrete linear objects in dimension n: the standard model. Graph. model. **65**(1), 92–111 (2003)
4. Andres, E.: The supercover of an m-flat is a discrete analytical object. Theor. Comput. Sci. **406**(1), 8–14 (2008)
5. Beckermann, B.: The condition number of real vandermonde, krylov and positive definite hankel matrices. Numer. Math. **85**(4), 553–577 (2000)
6. Bresenham, J.: A linear algorithm for incremental digital display of circular arcs. Commun. ACM **20**(2), 100–106 (1977)
7. Bresenham, J.E.: Algorithm for computer control of a digital plotter. IBM Syst. J. **4**(1), 25–30 (1965)
8. Brimkov, V.E., Dantchev, S.: Digital hyperplane recognition in arbitrary fixed dimension within an algebraic computation model. Image Vis. Comput. **25**(10), 1631–1643 (2007)
9. Buzer, L.: An incremental linear time algorithm for digital line and plane recognition using a linear incremental feasibility problem: In: Discrete Geometry for Computer Imagery, pp. 372–381. Springer (2002)
10. Charrier, E., Buzer, L.: An efficient and quasi linear worst-case time algorithm for digital plane recognition: In: Discrete Geometry for Computer Imagery, pp. 346–357. Springer (2008)
11. Chum, O., Matas, J., Kittler, J.: Locally optimized RANSAC: In: Pattern Recognition, pp. 236–243. Springer (2003)
12. Dexet, M., Andres, E.: A generalized preimage for the digital analytical hyperplane recognition. Discret. Appl. Math. **157**(3), 476–489 (2009)
13. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1), 11–15 (1972)
14. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
15. Gérard, Y., Provot, L., Feschet, F.: Introduction to digital level layers: In: Proceedings of the 16th IAPR international conference on Discrete Geometry for Computer Imagery (DGCI2011), LNCS, Vol. 6607, pp. 83–94. Springer (2011)
16. Hough, V., Paul, C.: Method and means for recognizing complex patterns (1962). US Patent 3,069,654
17. Largeteau-Skapin, G., Zrour, R., Andres, E.: O (n^3log$n$) time complexity for the optimal consensus set computation for 4-connected digital circles: In: Proceedings of the 17th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI2013), LNCS, Vol. 7749, pp. 241–252. Springer (2013)
18. 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 21st IAPR International Conference on Pattern Recognition (ICPR2012), pp. 3774–3777. IEEE (2012)
19. Narula, S.C., Wellington, J.F.: The minimum sum of absolute errors regression: A state of the art survey. Int. Stat. Rev./Revue Internationale de Statistique, 317–326 (1982)

20. Phan, M.S., Kenmochi, Y., Sugimoto, A., Talbot, H., Andres, E., Zrour, R.: Efficient robust digital annulus fitting with bounded error: In: Proceedings of the 17th IAPR International Conference on Discrete Geometry for Computer Imagery (DGCI2013), LNCS, Vol. 7749, pp. 253–264. Springer (2013)
21. Provot, L., Gérard, Y.: Estimation of the derivatives of a digital function with a convergent bounded error: In: Proceedings of the 16th IAPR international conference on Discrete Geometry for Computer Imagery (DGCI2011), LNCS, Vol. 6607, pp. 284–295. Springer (2011)
22. Provot, L., Gerard, Y.: Recognition of digital hyperplanes and level layers with forbidden points: In: Combinatorial Image Analysis, pp. 144–156 (2011)
23. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus: In: Computer Vision–ECCV 2008, pp. 500–513. Springer (2008)
24. Rousseeuw, P.: Least median of squares regression. J. Am. Stat. Assoc., 871–880 (1984)
25. Rousseeuw, P.J., Leroy, A.M.: Robust regression and outlier detection. Wiley (2005)
26. Sekiya, F., Sugimoto, A.: Discrete polynomial curve fitting to noisy data: In: Proceedings of the 15th International Workshop on Combinatorial Image Analaysis (IWCIA2012), LNCS, Vol. 7655, pp. 59–74. Springer (2012)
27. Spall, J.C.: Introduction to stochastic search and optimization: estimation, simulation, and control. Wiley (2005)
28. Torr, P.H., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. Comp. Vision Image Underst. **78**(1), 138–156 (2000)
29. Toutant, J.L., Andres, E., Roussillon, T.: Digital circles, spheres and hyperspheres: From morphological models to analytical characterizations and topological properties. Discret. Appl. Math. (2013)
30. Xu, L., Oja, E.: Randomized hough transform (rht): basic mechanisms, algorithms, and computational complexities. CVGIP: Image underst. **57**(2), 131–154 (1993)
31. 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**(1), 45–57 (2011)
32. Zrour, R., Largeteau-Skapin, G., Andres, E.: Optimal consensus set for annulus fitting: In: Proceedings of the 16th IAPR international conference on Discrete Geometry for Computer Imagery (DGCI2011), LNCS, Vol. 6607, pp. 358–368. Springer (2011)