Discrete Polynomial Curve Fitting to Noisy Data

Fumiki Sekiya[1] and Akihiro Sugimoto[2]

[1] Graduate School of Advanced Integration Science, Chiba University
sekiya@chiba-u.jp
[2] National Institute of Informatics
Tokyo, Japan
sugimoto@nii.ac.jp

Abstract. A discrete polynomial curve is defined as a set of points lying between two polynomial curves. This paper deals with the problem of fitting a discrete polynomial curve to given integer points in the presence of outliers. 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 method that effectively achieves a solution guaranteeing local maximality by using a local search, called rock climging, with a seed obtained by RANSAC. Experimental results demonstrate the effectiveness of our proposed method.

Keywords: Curve fitting, Discrete polynomial curve, Local optimal, Outliers, Consensus set, RANSAC, Discrete geometry.

1 Introduction

Fitting geometric models such as lines or circles is an essential task in image analysis and computer vision, and it is indeed used in feature detection and many other procedures. Though several methods exist for model fitting, they use continuous models even in a discrete environment. The method of least-squares is most common for curve fitting. This method minimizes the sum of squared residuals from all data, and the solution can be analytically computed. This method is, however, fatally susceptible to the presence of outliers: just one outlier can cause a great impact on estimation results. In order to enhance robustness, minimizing the sum of other functions has been proposed. For example, the method of least-absolute-values minimizes the sum of absolute residuals from all data. The method of least median of squares [5] minimizes the median of squared residuals, resulting in tolerating up to half the data being outliers. This means, however, that it does not work in the presence of more outliers. On the other hand, RANdom SAmple Consensus (RANSAC) [2] is commonly used in computer vision. This method maximizes the number of inliers, and work regardless of the fraction of outliers. However, its random approach takes a long time to ensure high accuracy.

R.P. Barneva et al. (Eds.): IWCIA 2012, LNCS 7655, pp. 59–74, 2012.

In discrete spaces, it is preferable to use discretized models rather than continuous ones because the representation of the models is also discrete. Discrete model fitting in the 2D discrete space is studied for lines [1, 6], annuluses [7], and polynomial curves [4]. For lines and annuluses, methods that work for a data set including 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 remains to be reported. This paper aims at developing a method for discrete polynomial curve fitting for a given set of discrete points including outliers.

We formulate the 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. 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 sence of the set inclusion. Experimental results demonstrate the efficiency of our method.

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

$$P = \{(x, y) \in \mathbb{R}^2 : y = a_1 x^k + a_2 x^{k-1} + \cdots + a_k x + a_{k+1},\ a_1 \neq 0\}, \quad (1)$$

where $a_1, \ldots, a_{k+1} \in \mathbb{R}$.

We define the discretization of eq. (1), namely, a *discrete polynomial curve*, by

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

where $f(x) = a_1 x^k + a_2 x^{k-1} + \cdots + a_k x + a_{k+1}$, and w is a given constant real value. a_i, k and w are respectively called the *coefficient*, the *degree*, and the *width* of the discrete polynomial curve ($i = 1, \ldots, k+1$). 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 a Digital Level Layer (DLL) [3].

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

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

where the coordinates of p_j are finite values, 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. 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 $(a_1, \ldots, a_{k+1})$ of $D \in \mathbb{D}_{k,w}$ having the maximum number of inliers.

A consensus set having 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 $(a_1, \ldots, a_{k+1})$. 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 $(x', y') \in S$, $(a_1, \ldots, a_{k+1})$ determining $D \in \mathbb{D}_{k,w}$ such that $D \ni (x', y')$ satisfies

$$0 \leq -x'^k a_1 - \cdots - x' a_k - a_{k+1} + y' \leq w \ . \tag{3}$$

We call the set of such points in the parameter space the *level layer* for (x', y'). (x', y') is a lower critical point when the left-hand side equality is satisfied, and is an upper critical point when the right-hand side equality is satisfied. For a consensus set $C = \{(x_1, x_1), \ldots, (x_m, y_m)\}$, we have $(a_1, \ldots, a_{k+1})$ that satisfies

$$\begin{cases} 0 \leq -x_1^k a_1 - \cdots - x_1 a_k - a_{k+1} + y_1 \leq w \ , \\ \vdots \\ 0 \leq -x_m^k a_1 - \cdots - x_m a_k - a_{k+1} + y_m \leq w \ . \end{cases} \tag{4}$$

Letting P_C be the convex polytope (the intersection of these level layers) defined by eq. (4), P_C is the set of $(a_1, \ldots, a_{k+1})$ determining $D \in \mathbb{D}_{k,w}$ such that $D \supset C$ but not $S \cap D = C$. Therefore, D determined by $(a_1, \ldots, a_{k+1})$ 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 level layers for the points in $C' \backslash C$.

Finding $C_{\max}$ is equivalent to finding the convex polytope(s) for $C_{\max}$ in the parameter space. Fig. 1 shows an example of level layers in the case of $k = 1$. Note that an intersection of level layers is always a convex polygon in this case.

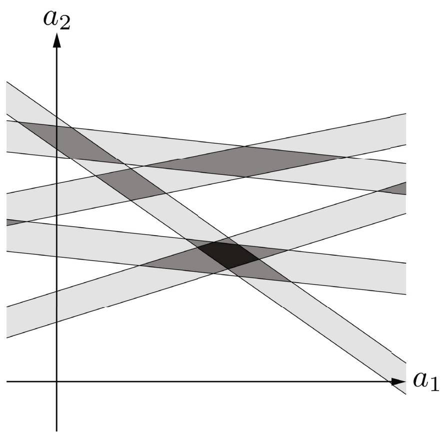


Fig. 1. An example of level layers in the case of $k = 1$. The darkness is proportional to the number of inliers.

If we define $F(a_1, \ldots, a_{k+1})$ = the number of inliers of D determined by $(a_1, \ldots, a_{k+1})$, then the discrete polynomial curve fitting problem is equivalent to seeking

$$\underset{(a_1,\ldots,a_{k+1})}{\arg\max}\ F(a_1, \ldots, a_{k+1}) \tag{5}$$

for given S, k, and w.

3 Properties of Discrete Polynomial Curves

A polynomial curve of degree up to k is uniquely determined by different $k + 1$ 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. A discrete polynomial curve $D \in \mathbb{D}_{k,w}$ with $k+1$ critical points $(s_1, t_1), \ldots, (s_{k+1}, t_{k+1})$ such that $s_i \neq s_j$ for all $i \neq j$, is identified as a point $(a_1, \ldots, a_{k+1})$ in the parameter space satisfying

$$\begin{cases} -s_1^k a_1 - \cdots - s_1 a_k - a_{k+1} + t_1 = c_1 \ , \\ \qquad\qquad \vdots \\ -s_{k+1}^k a_1 - \cdots - s_{k+1} a_k - a_{k+1} + t_{k+1} = c_{k+1} \ , \end{cases} \tag{6}$$

where

$$c_i = \begin{cases} 0 & \text{if } (s_i, t_i) \text{ is a lower critical point} \\ w & \text{if } (s_i, t_i) \text{ is an upper critical point} \end{cases} \quad (i = 1, \ldots, k + 1) \ .$$

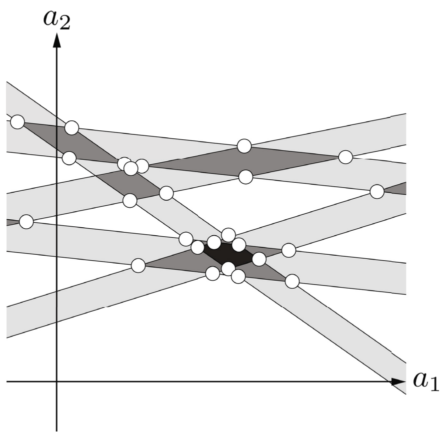


Fig. 2. Discrete polynomial curves of $\mathbb{G}_{S,k,w}$ in the parameter space. They are the intersection points of the boundaries of level layers; the white points represent them.

Eq. (6) has the unique solution in $(a_1, \ldots, a_{k+1})$ because it has $k+1$ linearly independent equations. □

We remark that eq. (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. 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 boundaries of the level layers for all points in a given $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$,

$$\begin{array}{l} -x_i^k a_1 - \cdots - x_i a_k - a_{k+1} + y_i = 0 \\ -x_i^k a_1 - \cdots - x_i a_k - a_{k+1} + y_i = w \end{array} \quad (i = 1, \ldots, n) \ . \tag{7}$$

Note that the boundaries of the two level layers 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, $(a_1, \ldots, a_{k+1})$ determining D satisfies at least $k+1$ independent equations in eq. (7). Therefore, D is an intersection point of the boundaries of the level layers identified by these equations. Fig. 2 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 $(a_1, \ldots, a_{k+1})$ (in the sense that it maximizes the number of inliers) by brute-forth 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 $m(\leq k)$ different x-coordinates is not maximum. Let $X_1, \ldots, X_m$ be these x-coordinates. Then, P_C is written by

$$\begin{cases} L_1 \leq -X_1^k a_1 - \cdots - X_1 a_k - a_{k+1} \leq U_1 \ , \\ \vdots \\ L_m \leq -X_m^k a_1 - \cdots - X_m a_k - a_{k+1} \leq U_m \ , \end{cases} \tag{8}$$

where $L_i, U_i \in \mathbb{R}$, and $U_i - L_i \leq w$ for $i = 1, \ldots, m$. Since the points in S have at least $k+1$ different x-coordinates, there exists a point $(X, Y) \in S \backslash C$ such that $X \neq X_i$ for $i = 1, \ldots, m$. The level layer for (X, Y) is

$$0 \leq -X^k a_1 - \cdots - X a_k - a_{k+1} + Y \leq w. \tag{9}$$

There exists at least one solution $(a_1, \ldots, a_{k+1})$ satisfying both of eq. (8) and eq. (9). 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. 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.

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 $(a_1, \ldots, a_{k+1})$ in $P_{C_{\max}}$ satisfies

$$\begin{cases} 0 \leq -u_1^k a_1 - \cdots - u_1 a_k - a_{k+1} + v_1 \leq w \ , \\ \vdots \\ 0 \leq -u_{k+1}^k a_1 - \cdots - u_{k+1} a_k - a_{k+1} + v_{k+1} \leq w \ , \end{cases} \tag{10}$$

which can be rewritten as

$$\begin{cases} -u_1^k a_1 - \cdots - u_1 a_k - a_{k+1} + v_1 = d_1 \ , \\ \vdots \\ -u_{k+1}^k a_1 - \cdots - u_{k+1} a_k - a_{k+1} + v_{k+1} = d_{k+1} \ , \end{cases} \tag{11}$$

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

$$\begin{pmatrix} a_1 \\ \vdots \\ a_{k+1} \end{pmatrix} = \begin{pmatrix} -u_1^k & \cdots & -u_1 & 1 \\ \vdots & \ddots & \vdots & \vdots \\ -u_k^k & \cdots & -u_k & 1 \\ -u_{k+1}^k & \cdots & -u_{k+1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} d_1 - v_1 \\ \vdots \\ d_k - v_k \\ d_{k+1} - v_{k+1} \end{pmatrix} . \tag{12}$$

Denoting the (i, j) entry of the inverse matrix by m_{ij} allows eq. (12) to be written as

$$a_i = \sum_{j=1}^{k+1} m_{ij}(d_j - v_j) \quad (i = 1, \ldots, k+1) \; . \tag{13}$$

Eq. (13) shows that a_i is linear in $d_1, \ldots, d_{k+1}$. Therefore, the set of $(a_1, \ldots, a_{k+1})$ satisfying eq. (10) is bounded since $0 \leq d_i \leq w$. $P_{C_{\max}}$ 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 are found by the brute-forth search. 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-forth search requires up to $2^{k+1}\binom{|S|}{k+1}$ 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 having k upper 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, $(a_1, \ldots, a_{k+1})$ of $D' \in N_D$ satisfies the same k independent equations as that of D in eq. (7). Therefore, $(a_1, \ldots, a_{k+1})$ corresponding to D' is on the intersection line of the k hyperplanes that are the boundaries of the level layers identified by these equations. Thus, the neighboring relations are determined by the intersection lines of k boundaries of level layers. We call these lines neighboring lines. Fig. 3 shows an example of neighbors in the parameter space when $k = 1$. In this case, the neighboring lines are identical to the boundaries of level layers 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. Algorithm 2 describes the concrete procedure of rock climbing.

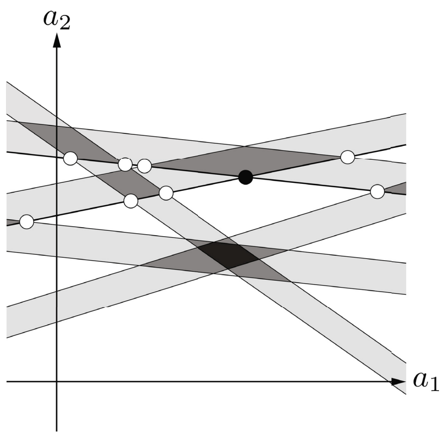


Fig. 3. An example of the neighbors ($k = 1$). The neighbors of the black point are depicted with white points. They are on the neighboring lines, i.e., lines passing through the black point.

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.

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* $P_{C_{\text{local}}}$.

Proof. Let C be the consensus set of the current discrete polynomial curve.

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 $(a_1, \ldots, a_{k+1})$ in $P_{C_{\text{local}}}$ satisfies $F(a_1, \ldots, a_{k+1}) = |C_{\text{local}}|$. 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 level layer for (x', y'). Therefore, each vertex of $P_{C'}$ is on an edge or a vertex of P_C as illustrated in Fig. 4 . This means that we can obtain all the vertices of $P_{C'}$ by propagating the neighboring relation

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 $A = A'$ **then**
    Break out of the loop
  else
    $A := A'$
  end if
end loop
return A

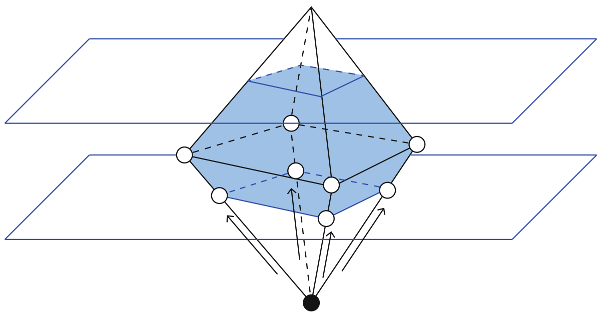


Fig. 4. P_C (black) and $P_{C'}$ (blue). Each vertex of $P_{C'}$ is on an edge or a vertex of P_C. Suppose that the black point corresponds to the current polynomial curve. Then the white points are the neighbors in P_C.

from the current vertex of P_C. Furthermore, any $(a_1, \ldots, a_{k+1})$ in P_C satisfies $F(a_1, \ldots, a_{k+1}) \geq |C|$. 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, we can always find all the vertices of a $P_{C_{\text{local}}}$ by rock climbing. Therefore, we can generate all $(a_1, \ldots, a_{k+1})$'s determining D such that $D \supset C_{\text{local}}$ from these vertices.

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 rock climbing 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.

5 Experiments

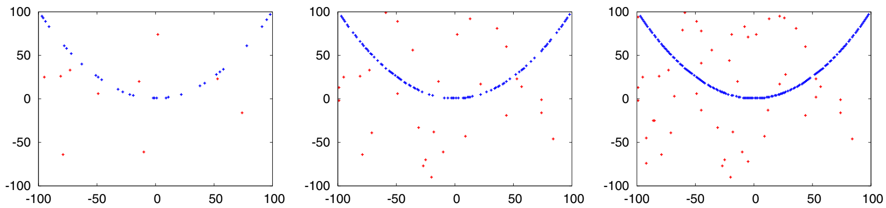
To demonstrate the effectiveness of our proposed method, we compared our method with RANSAC under two different scenarios. First, we fixed the ratio between inliers and outliers among input points and changed the number of input points. Then, we evaluated the computational time required to obtain the maximum number of inliers. Second, we fixed the number of input points and changed the ratio between inliers and outliers. Then, we evaluated the computational time. In both cases, we observed that our method outperforms RANSAC.

For the first scenario, we set $k = 2, w = 1$ and fixed the ratio of outliers among input points to be 25%, 50%, 75%. For each fixed ratio, we generated five different discrete input point sets S, where $|S|$ was changed by 40 from 40 to 200 (see Fig. 5 for examples). In each set, integer points satisfying $0 \leq y - 0.01x^2 \leq w$ $(-100 \leq x \leq 100)$ were randomly generated for inliers (blue points) and integer points that do not satisfy this inequality were randomly generated within $[-100, 100] \times [-100, 100]$ for outliers (red points). We remark that we designed in each fixed outlier ratio, all the five input point sets have the same optimal solutions in $\mathbb{G}_{S,k,w}$ $(k = 2, w = 1)$. (Data-sets having different outlier ratios do not have the same optimal solutions.)

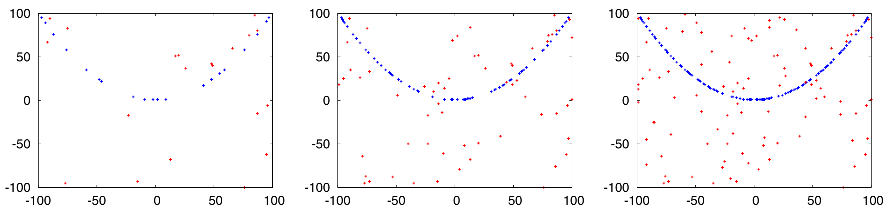
To these data-sets, we applied our method 100 times independently where we set $t = 1000$ (the number of iteration for our RANSAC step). We then evaluated the computational time to obtain $C_{\max}$ (a consensus set having the maximum number of inliers) in terms of the required number of samplings there. Note that one sampling takes the same computational time and thus the number of samplings can be a measurement for the computational time. For comparison, we applied RANSAC alone without setting any limited number of iterations, and terminated it when $C_{\max}$ is obtained.

The average number of samplings over the 100 trials is given in Table 1 and illustrated in Fig. 6. We see that our method finds $C_{\max}$ more than twice faster than RANSAC and that the difference of required numbers of samplings to find $C_{\max}$ drastically becomes larger as the number of input points increases. From Fig. 6, we can also observe that regardless of outlier ratios, the required number of samplings has a similar behavior depending on the number of input points. Namely, the required number of samplings slowly increases and is not exponentially affected by the number of input points for our method while it exponentially increases for RANSAC. We can thus conclude that the number of input points has far less impact on our method than RANSAC.

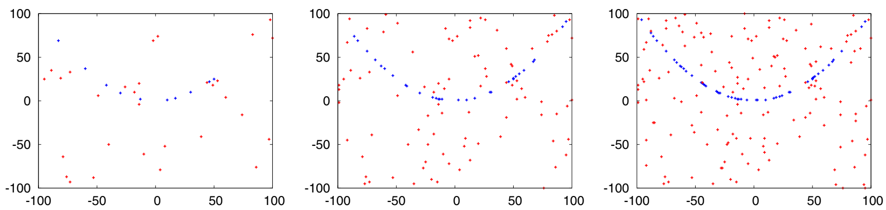
For the second scenario, we again set $k = 2, w = 1$ and fixed the number of input points to be 200. We generated nine different discrete input point sets, where the ratio of outliers was changed by 10% from 10% to 90% (see Fig. 7 for examples). In each set, inliers and outliers are generated in the similar way as the first scenario. To these data-sets, we applied our method 100 times independently and evaluated the required number of samplings to obtain $C_{\max}$. We also applied RANSAC alone using the same condition as the first scenario case.



(a) $|S| = 40$, 25% outliers. (b) $|S| = 120$, 25% outliers. (c) $|S| = 200$, 25% outliers.



(d) $|S| = 40$, 50% outliers. (e) $|S| = 120$, 50% outliers. (f) $|S| = 200$, 50% outliers.



(g) $|S| = 40$, 75% outliers. (h) $|S| = 120$, 75% outliers. (i) $|S| = 200$, 75% outliers.

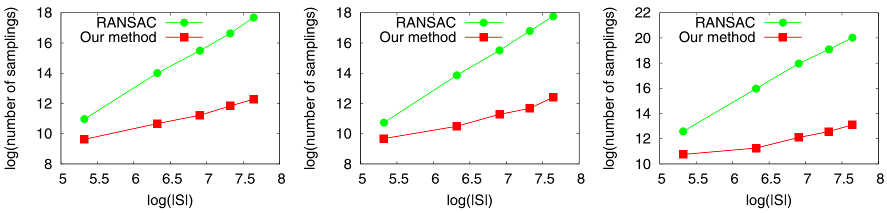
Fig. 5. Input set examples with different numbers of points and different outlier ratios ($k = 2$). (a), (b), (c) are for 25% outliers; (d), (e), (f) are for 50% outliers; (g), (h), (i) are for 75% outliers.

Table 2 and Fig. 8 show the average number of samplings over the 100 trials. From these results, we can see that our method finds $C_{\max}$ more than ten times faster than RANSAC. We can also observe that in both methods, the outlier ratio does not affect the required number of samplings as far as the number of input points is the same. We remark that in our method, the required number of samplings in the case where the outlier ratio is 90% (in this case, the number of inliers is 20 while that of outliers is 180) becomes almost twice of that for the other cases. This suggests that there may be a minimum number of inliers required for our method to work effectively. Investigating this is left for future work.

So far, we had experiments only for quadratic curves ($k = 2$). In order to confirm our observations even for another order case, we conducted the same experiments under the condition of $k = 3$ and $w = 1$. As input points, we randomly generated inliers satisfying $0 \leq y - 0.0001x^3 \leq w$ $(-100 \leq x \leq 100)$

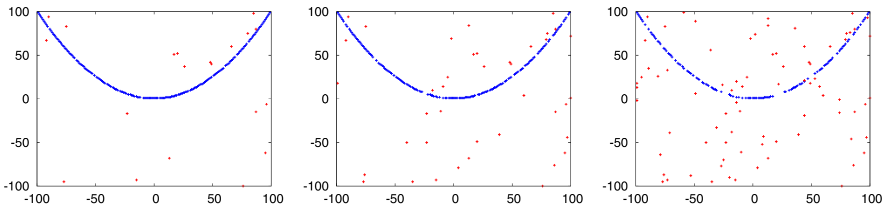
Table 1. Number of samplings ($\times 10^3$) required for achieving $C_{\max}$ ($k = 2$)

ratio of outliers (%)	$\|S\|$	40	80	120	160	200
25	our method	0.8	1.6	2.4	3.6	5.0
	RANSAC	2.0	16.5	46.1	101.7	211.2
50	our method	0.8	1.4	2.4	3.2	5.4
	RANSAC	1.7	14.9	46.6	113.1	223.4
75	our method	1.7	2.4	4.3	6.0	8.8
	RANSAC	6.1	64.4	256.1	560.1	1062.0

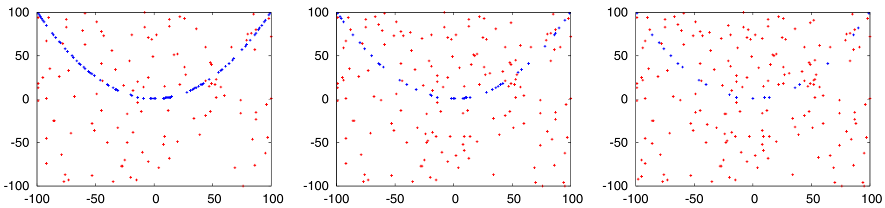


(a) 25% outliers. (b) 50% outliers. (c) 75% outliers.

Fig. 6. Required number of samplings depending on $|S|$ ($k = 2$)



(a) 10% outliers. (b) 20% outliers. (c) 40% outliers.



(d) 60% outliers. (e) 80% outliers. (f) 90% outliers.

Fig. 7. Input set examples with different outlier ratios under the same number of points ($|S| = 200$, $k = 2$)

Table 2. Number of samplings ($\times 10^3$) under different outlier ratios ($k = 2$)

ratio of outliers (%)	10	20	30	40	50	60	70	80	90
our method	4.7	4.6	4.2	4.6	4.6	4.4	4.6	4.3	7.0
RANSAC	76.7	80.1	75.9	87.7	81.2	74.9	75.8	72.0	77.4

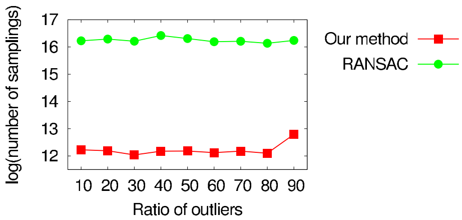
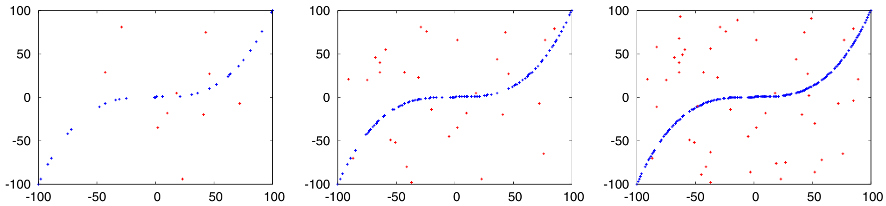
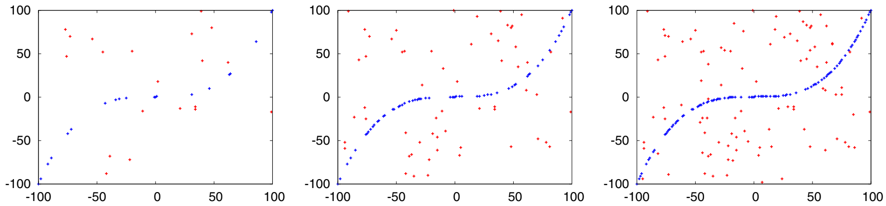


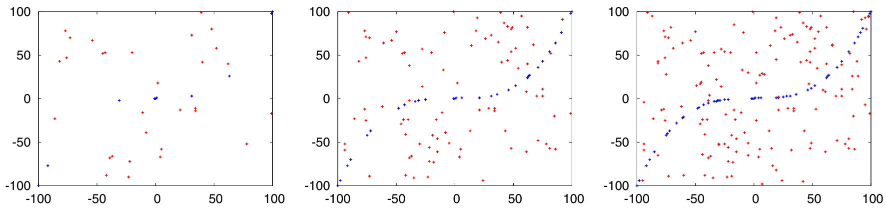
Fig. 8. Required number of samplings depending on outlier ratio ($k = 2$)



(a) $|S| = 40$, 25% outliers. (b) $|S| = 120$, 25% outliers. (c) $|S| = 200$, 25% outliers.



(d) $|S| = 40$, 50% outliers. (e) $|S| = 120$, 50% outliers. (f) $|S| = 200$, 50% outliers.

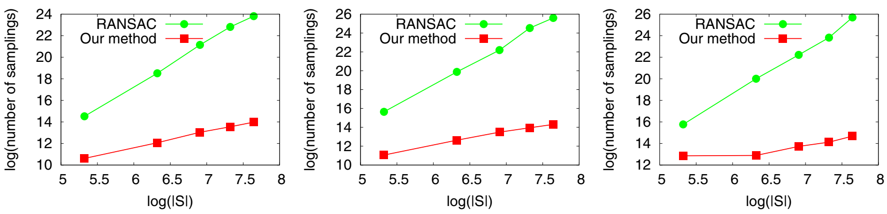


(g) $|S| = 40$, 75% outliers. (h) $|S| = 120$, 75% outliers. (i) $|S| = 200$, 75% outliers.

Fig. 9. Input set examples with different numbers of points and different outlier ratios ($k = 3$). (a), (b), (c) are for 25% outliers; (d), (e), (f) are for 50% outliers; (g), (h), (i) are for 75% outliers.

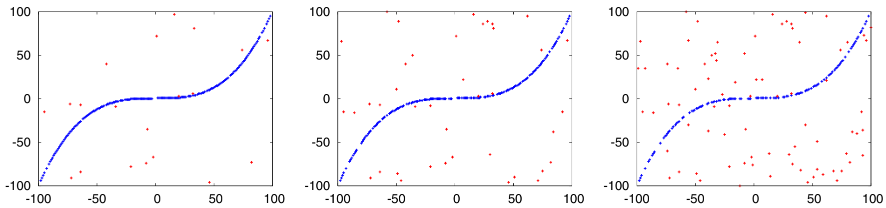
Table 3. Number of samplings ($\times 10^3$) required for achieving $C_{\max}$ ($k = 3$)

ratio of outliers (%)	$\lvert S \rvert$	40	80	120	160	200
25	our method	1.5	4.2	8.4	11.9	16.5
	RANSAC	23.5	374.2	2321.9	7380.0	14749.4
50	our method	2.1	6.3	11.4	15.8	20.2
	RANSAC	51.0	963.4	4784.3	24186.0	50754.9
75	our method	7.4	7.5	13.5	17.8	26.2
	RANSAC	55.8	1052.4	4890.6	14855.5	54525.0

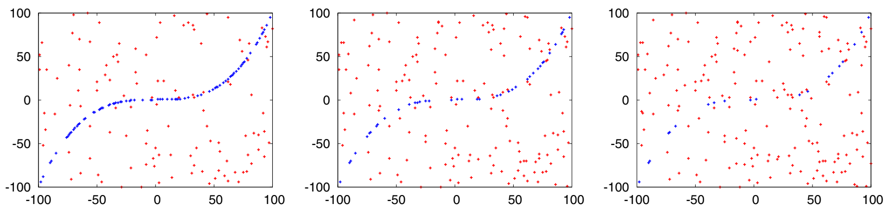


(a) 25% outliers. (b) 50% outliers. (c) 75% outliers.

Fig. 10. Required number of samplings depending on $\lvert S \rvert$ ($k = 3$)



(a) 10% outliers. (b) 20% outliers. (c) 40% outliers.



(d) 60% outliers. (e) 80% outliers. (f) 90% outliers.

Fig. 11. Input set examples with different outlier ratios under the same number of points ($\lvert S \rvert = 200$, $k = 3$)

Table 4. Number of samplings under different outlier ratios ($k = 3$)

ratio of outliers (%)	10	20	30	40	50	60	70	80	90
our method ($\times 10^4$)	1.7	1.8	1.9	1.7	1.9	2.0	1.9	2.6	5.7
RANSAC ($\times 10^7$)	2.9	2.6	2.9	2.9	2.4	2.8	2.6	2.9	3.1

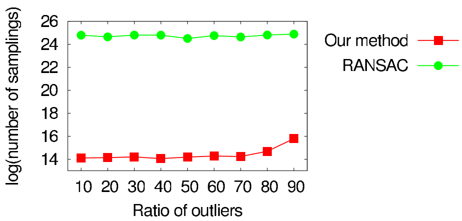


Fig. 12. Required number of samplings depending on outlier ratio ($k = 3$)

and outliers over $[-100, 100] \times [-100, 100]$ so that no outlier satisfies this inequality (see Figs. 9 and 11 for examples). The results are shown in Tables 3 and 4 and Figs. 10 and 12. From these results, we have the same observation as the quadratic curves case. We can thus conclude that our method significantly outperforms RANSAC.

6 Conclusion

This paper dealt with the problem of fitting a discrete polynomial curve to a given set of points including outliers. We formulated this problem as an optimization problem where the number of inliers is maximized. Our proposed method effectively searches solutions by rock climbing using an initial seed obtained by RANSAC. We showed that our method guarantees local maximality of inliers in the sense of the set inclusion. The effectiveness of our method was demonstrated using experiments.

Acknowledgements. 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 under the contract of 23650092 and by a Japanese-French joint research project called the SAKURA program.

References

1. Aiger, D., Kenmochi, Y., Talbot, H., Buzer, L.: Efficient Robust Digital Hyperplane Fitting with Bounded Error. In: Domenjoud, E. (ed.) DGCI 2011. LNCS, vol. 6607, pp. 223–234. Springer, Heidelberg (2011)
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)

3. Gérard, Y., Provot, L., Feschet, F.: Introduction to Digital Level Layers. In: Domenjoud, E. (ed.) DGCI 2011. LNCS, vol. 6607, pp. 83–94. Springer, Heidelberg (2011)
4. Provot, L., Gérard, Y.: Estimation of the Derivatives of a Digital Function with a Convergent Bounded Error. In: Domenjoud, E. (ed.) DGCI 2011. LNCS, vol. 6607, pp. 284–295. Springer, Heidelberg (2011)
5. Rousseeuw, P.: Least median of squares regression. Journal of the American Statistical Association, 871–880 (1984)
6. Zrour, R., Kenmochi, Y., Talbot, H., Buzer, L., Hamam, Y., Shimizu, I., Sugimoto, A.: Optimal consensus set for digital line and plane fitting. International Journal of Imaging Systems and Technology 21(1), 45–57 (2011)
7. Zrour, R., Largeteau-Skapin, G., Andres, E.: Optimal Consensus Set for Annulus Fitting. In: Domenjoud, E. (ed.) DGCI 2011. LNCS, vol. 6607, pp. 358–368. Springer, Heidelberg (2011)