PAPER

Minimax Geometric Fitting of Two Corresponding Sets of Points and Dynamic Furthest Voronoi Diagrams

Keiko IMAI†, Shigeo SUMINO††, *and* Hiroshi IMAI†††, *Members*

SUMMARY This paper formulates problems of fitting two corresponding sets of points by translation, rotation and scaling, and proposes efficient algorithms for the fitting. The algorithms are based on the theory of lower envelopes, or Davenport-Schinzel sequences, and linearization techniques in computational geometry, and are related to dynamic furthest Voronoi diagrams.

key words: *computational geometry, lower envelopes, linearization, Davenport-Schinzel sequences*

1. Introduction

Matching or fitting two similar sets of n points is a fundamental problem in image processing and pattern recognition. Generally, the main step in solving this fitting problem is to establish a good correspondence between the two sets. However, there are cases in practice such that a one-to-one correspondence between two sets of points, S and T, is given in advance, and the S set of points must be fitted with the T set of points by applying geometric operations to S such as translation, rotation and/or scaling.

Such problems arise in an industrial robot attaching a pin-grid-array type LSI (large-scale-integrated-circuit) chip to a board by using visual sensors [15]. The robot first takes an image of the pins of the LSI chip using a visual sensor. Then it tries to fit the LSI package to the corresponding patterns on the board in the best way by translating and rotating the LSI chip, matching the image with stored patterns (scaling the image is sometimes required). The patterns are a collection of disks or squares of the same size arranged according to a regular grid. Figure 1 illustrates the case with square patterns. Recent ball-grid-array type LSI chips correspond to the disk case.

This problem is a geometric fitting problem of a set S of points representing pins with a set T of regular grid points. This geometric fitting problem is of the minimax type. That is, the maximum among the distances between two corresponding points in S and T should be minimized by the geometric operations. This is because the pins can be attached to the corresponding patterns correctly if this minimax distance is not greater than the radius of the disks or half the side length of the squares, and a minimax location gives the most reliable attachment of the LSI chip. If the patterns are disks, Euclidean distance suffices as the distance between points in this problem, while if the patterns are squares, L_∞ distances with respect to the axis of the set T must be used. This geometric fitting problem is a special case of a calibration problem in image processing, where fitting is done nonlinearly (e.g., see [12], [23]). The case for pin-grid-array and ball-grid-array type LSIs corresponds to the minimax type fitting problem under standard geometric transformations, and this enables us to solve the problem optimally and efficiently.

This paper thus considers the following problems: Given two sets of points in a plane, $S = \{s_i \mid i = 1,\ldots,n\}$ and $T = \{t_i \mid i = 1,\ldots,n\}$, such that s_i is associated with t_i, translate, rotate (or transform in a more complicated way) and/or scale the set of points S simultaneously so that the maximum of the L_2 or L_∞ distances between t_i and the transformed s_i is minimized. The algorithmic complexity of geometric fitting problems is dependent upon which geometric operations are used in each problem, so that the following notation will be used. There are basically three operations: Translation, Rotation, and Scaling, which are abbreviated as T, R, S, respectively. As noted above, for calibration in image processing, a more general transformation is needed instead of rotation, which is repre-

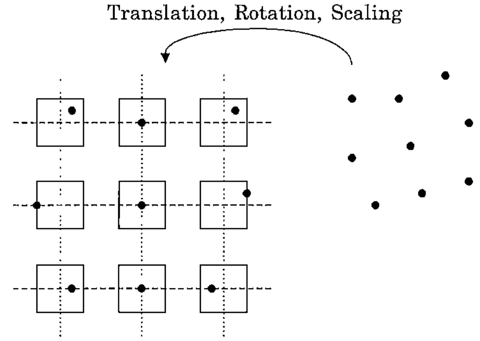

Fig. 1 Geometric fitting of distorted grid points with regular grid points in L_∞ norm.

Manuscript received October 22, 1997.
Manuscript revised June 11, 1998.

†The author is with the Department of Information and System Engineering, Chuo University, Tokyo, 112–8551 Japan.
††The author is with Central Research Laboratory, Hitachi Co., Kokubunji-shi, 185–8601 Japan.
†††The author is with the Department of Information Science, the University of Tokyo, Tokyo, 113–0033 Japan.

Table 1 Summary of the time and space complexity of geometric fitting problems where $(t(n), s(n))$ means the time complexity is $O(t(n))$ and the space complexity is $O(s(n))$.

geometric operations	L_2 distance	L_∞ distance
T (translation)	$(n,\ n)$ [13]§	$(n,\ n)$§
TS (translation/scaling)	$(n(\log n)^4 \log\log n, n)$ [14]§	$(n,\ n)$ [13]§
TR (translation/rotation)	$(n^2\lambda_7(n)\log n,\ n)$† or $(n^3 \log n,\ n^3)$‡	$(n\log n,\ n)$
TG (translation/general transform)	$(n^3 \log^* n \log n,\ n)$	$(n \log^* n \log n,\ n)$
TSR (translation/scaling/rotation)	$(n^4 \log n,\ n^4)$‡	$(n^2,\ n^2)$§
TSG (translation/scaling/general transform)	$(n^4 \log^* n \log n,\ n^4)$†	$(n^4(\log^* n)^2, n^4(\log^* n)^2)$¶

§these cases can be reduced to existing problems; †$\lambda_s(n)$ is the maximum length of (n, s) Davenport-Schinzel sequence and $\lambda_{2s+1}(n) = O(n \cdot \alpha(n)^{O(\alpha(n)^{s-1})})$ for $s \geq 2$, where $\alpha(n)$ is the functional inverse of Ackermann's function, and $\lambda_s(n) = O(n \log^* n)$ when s is regarded as a constant (e.g., see [21], [22]); ‡these are based on the output-size sensitive convex hull algorithm by Seidel [18], and these bounds are in the worst case, hence practical bounds would be better; ¶another kind of output-sensitive algorithm.

sented by low-order polynomials or rational functions. This general operation will be abbreviated as G. Then a geometric fitting problem with some of these four operations will be denoted by placing the corresponding characters before the term "fitting." For example, a geometric fitting problem with translation and rotation will be referred to as the TR-fitting problem (see Table 1).

The geometric fitting problem has been investigated in computational geometry by Alt, Mehlhorn, Wagener and Welzl [2] in connection with the congruence problem. In fact, the optimization version of the approximate congruence problem that they considered is exactly the geometric fitting problem that we have defined so far. In that paper, the decision version is also considered, which determines whether two corresponding sets of points can be fitted so that the maximum distance between the corresponding pairs of points is less than or equal to a given tolerance ϵ. They show that the decision version of the geometric fitting problem between two corresponding sets of n points in L_2 and L_∞ norms can be solved in $O(n^3 \log n)$ time, which would yield an $O(n^3(\log n)^2)$-time algorithm for the optimization problem. In this paper, we give a much better algorithm for the L_∞ problem and a slightly better algorithm for the L_2 problem. Our approach, which is described below, is quite different from theirs. With regard to the cited paper [2], it should be mentioned that the approximate congruence problem for two sets of points without any prescribed correspondence is also discussed, and polynomial-time algorithms for it are given there.

The geometric fitting problem can be regarded in general as the problem in dynamic computational geometry introduced by Atallah [3]. In fact, the fitting problem in L_2 norm will be treated as a smallest enclosing circle problem for moving points. Some special cases of the fitting problem then reduce to existing problems in (dynamic) computational geometry as shown in the above part of Table 1. Especially, it is noted that the TS-fitting problem in L_2 norm coincides with the dynamic minimum enclosing circle problem considered by Megiddo [14].

In this paper, we formulate the geometric fitting problem as that of finding a minimum point on the upper envelope of n multivariate functions, and give efficient algorithms for TR-, TSR-, TG- and TSG-fitting problems in L_2 and L_∞ norms. Our results are summarized in Table 1. Note that some of them are "output-size sensitive" algorithms, and would behave better in practice.

The combinatorial complexity of the upper envelope of multivariate functions has been a hot research topic [1], [3], [5]–[9], [17], [20]–[22]. In fact, our analysis of the TR-fitting problem in L_2 norm gives an algorithm to construct furthest Voronoi diagrams for moving points together with an upper bound of the combinatorial complexity of such diagrams (for dynamic Euclidean Voronoi diagrams, see also [1], [7]–[9] and an original conference paper version [10] of this paper). In our analysis, nice structures of this practical fitting problem are used to decompose the problem into one-dimensional Davenport-Schinzel sequences or to apply a linearization technique (using this linearization technique was suggested to the authors by Sharir [19]).

The analysis using the Davenport-Schinzel sequence provides a slightly worse time bound in some cases than linearization for the rotation operation, but besides rotation it can be applied to a general transformation. The result for the general transformation is obtained by simply replacing the result for the rotation along this line. Also, it should be noted that by rotation and general transformation, L_∞ norm may change, but, as is seen from Fig. 1, we have a meaningful direction in our problems and it is invariant in this sense.

2. Preliminaries on Envelopes

Let f_i be a d-variable real-valued function on a domain $D \subseteqq \mathbf{R}^d$ ($i = 1, \ldots, n$). Define a function f on D taking the maximum value of f_i at each point, i.e.,

$$f(x) = \max_{i=1,\ldots,n} f_i(x) \qquad (x \in D)$$

The graph $y = f(x)$ of $f(x)$ is the upper envelope of the graphs $y = f_i(x)$ of $f_i(x)$. With each point $x \in D$, associate an index set $I(x)$ of indices attaining the maximum in f, i.e., $I(x) = \{i \mid f(x) = f_i(x)\}$. Divide the domain D into a subdivision of maximally connected regions of pairs $(x, I(x))$. The obtained subdivision of D is called *the maximum diagram* of f_i ($i = 1, \ldots, n$). In other words, the maximum diagram is obtained by projecting the pointwise maximum of f_i onto D. The maximum diagram consists of k-dimensional faces ($k = 0, \ldots, d$). The combinatorial complexity of the maximum diagram is defined to be the number of all faces. If a point x is efficiently located in the maximum diagram to find $I(x)$, $f(x)$ can be efficiently computed as $f_i(x)$ for $i \in I(x)$.

In the case of $d = 1$, the combinatorial complexity of the upper envelope has been investigated in terms of Davenport-Schinzel sequences. The following is a primary result: When f_i ($i = 1, \ldots, n$) is a continuous function on D, and each pair of the function intersects at most a constant s times, the combinatorial complexity of the upper envelopes is bounded by $\lambda_s(n)$, the maximum length of (n, s) Davenport-Schinzel sequence. $\lambda_s(n)$ is an almost linear function in n, and in fact is $O(n \log^* n)$. When f has a constant number of discontinuous points, the combinatorial complexity is bounded by $\lambda_{s+2}(n)$. See [21] for details.

Higher-dimensional cases are more difficult to analyze, and in fact the problem treated in this paper is concerned with these cases. For related results, see [1], [7]–[9].

3. Problem Formulation

Here we formulate the TSR-fitting problem. The TR-fitting problem is a special TSR-fitting, whose scaling factor is fixed at 1. Recall that we are given two sets of points, $S = \{s_j = (x_j, y_j) \mid j = 1, \ldots, n\}$ and $T = \{t_j = (u_j, v_j) \mid j = 1, \ldots, n\}$, where s_j is made to correspond to t_j. The problem is described so far by fixing T and translating/rotating/scaling S. However, scaling T by a factor $\alpha (> 0)$ is equivalent to scaling S by a factor $1/\alpha$, and makes no problem in discussing the maximum of L_2 and L_∞ distances, so we will scale T below.

The problem can be stated compactly by considering it in the complex number plane. Identify s_j and t_j with complex numbers $x_j + \mathrm{i}y_j$ and $u_j + \mathrm{i}v_j$, respectively. Rotating the set S of points by an angle θ ($0 \leqq \theta < 2\pi$) and then translating it by making the origin $z = x + \mathrm{i}y$, point s_j is mapped to $s_j \mathrm{e}^{\mathrm{i}\theta} - z$. Scaling the set T of points by a factor $\alpha \geq 0$, point t_j is mapped to αt_j. Hence, the L_p distance between mapped s_j and t_j is $\|s_j \mathrm{e}^{\mathrm{i}\theta} - \alpha t_j - z\|_p$, where $\|\cdot\|_p$ denotes the L_p norm. Thus the TSR-fitting problem in L_p norm is expressed as follows:

$$\min_{z,\, 0 \leqq \theta < 2\pi,\, \alpha \geqq 0} \ \max_{j=1,\ldots,n} \|s_j \mathrm{e}^{\mathrm{i}\theta} - \alpha t_j - z\|_p$$

Defining a point $p_j(\theta, \alpha) = x_j(\theta, \alpha) + \mathrm{i}y_j(\theta, \alpha)$ in the complex number plane by

$$x_j(\theta, \alpha) = x_j \cos\theta - y_j \sin\theta - \alpha u_j$$
$$y_j(\theta, \alpha) = x_j \sin\theta + y_j \cos\theta - \alpha v_j$$

the problem is rewritten as

$$\min_{0 \leqq \theta < 2\pi,\ \alpha \geqq 0} \left(\min_{z} \ \max_{j=1,\ldots,n} \|p_j(\theta, \alpha) - z\|_p \right)$$

Fixing θ and α, the problem becomes the minimum enclosing circle and square (parallel to the axis) problem for n points $p_j(\theta, \alpha)$ in the case of $p = 2$ and ∞, respectively. As $p_j(\theta, \alpha)$ ($j = 1, \ldots, n$) move in the plane by two parameters θ and α, the minimum circle enclosing them changes, and we want to find the smallest.

When a general transformation G, mentioned in the introduction, is used instead of rotation in this formulation, $p_j(\theta, \alpha)$ should be changed as

$$x_j(\theta, \alpha) = f_{x_j, y_j}(\theta) - \alpha u_j$$
$$y_j(\theta, \alpha) = g_{x_j, y_j}(\theta) - \alpha v_j$$

where $f_{x_j,y_j}(\theta)$ and $g_{x_j,y_j}(\theta)$ are low-order polynomial or rational functions of θ with coefficients determined by x_j and y_j (θ does not correspond to an angle here, and these two functions depend on a model used in calibration and hence this θ is a parameter independent of the scaling factor α). We assume that these functions are well behaved and that the degrees of these functions are constants independent of n.

Finding a solution to this kind of problem requires the solution of a system of polynomial equations whose size and degree are constants independent of n. We assume that ideally such a system can be solved in a constant time.

4. Geometric Fitting Problem in L_2 Norm

This section describes two algorithms for each of the TR- and TSR-fitting problems in L_2 norm. Results for the TG- and TSG-fitting can be obtained by modifying one of two algorithms for the TR- and TSR-fitting problem. The TR-fitting problem is first related to constructing furthest Voronoi diagrams for moving points, and an algorithm for traversing the diagram efficiently in linear space is given. Also, applying the linearization technique, the problem is reduced to enumerating the faces of the intersection of n halfspaces in 7-dimensional space. For the TSR-fitting problem, a completely different approach is first taken which can be used in the minimax problem for multivariate convex functions. Linearization is also demonstrated as useful in solving this problem.

4.1 TR-Titting Problem in L_2 Norm

In the TR-fitting problem, the scaling factor α is set to be 1, and, throughout this subsection, $p_i(\theta, \alpha) = p_i(\theta, 1)$ will be written simply as $p_i(\theta)$. Rewriting the general formulation given in Sect. 3 to this case, the problem is stated as follows:

$$\min_{0 \leqq \theta < 2\pi,\, x,\, y} f(\theta, x, y)$$

where

$$f(\theta, x, y) = \max_{i=1,\ldots,n} f_i(\theta, x, y)$$
$$f_i(\theta, x, y) = (x - x_i(\theta, 1))^2 + (y - y_i(\theta, 1))^2.$$

$f(\theta, x, y)$ corresponds to the upper envelope of n functions $f_i(\theta, x, y)$ with three variables. Fixing $\theta = \theta'$, consider the upper envelope of n functions $f_i(\theta', x, y)$ of x and y. As is well known, the maximum diagram of $f_i(\theta', x, y)$ for fixed θ' is the furthest Voronoi diagram of points $p_i(\theta')$ $(i = 1, \ldots, n)$ [4], [16]. The furthest Voronoi diagram for n fixed points in the plane can be constructed in $O(n \log n)$ time, and, given the diagram, the minimum enclosing circle of the points can be computed in linear time (see again [4], [16]). We will make use of this property for the dynamic case.

4.1.1 Algorithm Using Davenport-Schinzel Sequences of a Single Variable

In this paper we assume that, for any θ, $p_i(\theta)$ $(i = 1, \ldots, n)$ are pairwise distinct, to simplify our discussions.

The minimum enclosing circle of points in a plane has at least two points on its boundary. To solve the dynamic problem for moving points $p_k(\theta)$, we may consider the following constrained problem: Find the minimum enclosing circle of moving points $p_k(\theta)$ $(k = 1, \ldots, n;\ 0 \leqq \theta < 2\pi)$ having $p_i(\theta)$ and $p_j(\theta)$ on its boundary for a given pair of i and j. For fixed θ, such an enclosing circle having $p_i(\theta)$ and $p_j(\theta)$ on its boundary may not exist. However, if $p_i(\theta^*)$ and $p_j(\theta^*)$ is on the boundary of the optimal enclosing circle of the dynamic problem, that enclosing circle is also an optimal solution to the constrained problem. Hence, solving this constrained problem for every pair of i and j among $\binom{n}{2}$ pairs, the original problem can be solved simply by picking the smallest.

Fix i and j. Firstly, let us investigate when there exists an enclosing circle of $p_k(\theta)$ $(k = 1, \ldots, n)$ with $p_i(\theta)$ and $p_j(\theta)$ on its boundary. This corresponds to the existence of a Voronoi edge of $p_i(\theta)$ and $p_j(\theta)$ in the furthest Voronoi diagram.

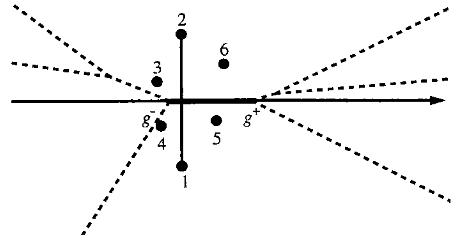

Fig. 2 A Voronoi edge of $p_1(\theta)$ and $p_2(\theta)$, where $g^+(\theta) = g_3^+(\theta)$, $g^-(\theta) = g_6^-(\theta)$, $g_5^+(\theta) = g_6^+(\theta) = +\infty$, $g_3^-(\theta) = g_4^-(\theta) = -\infty$.

The center of a circle having $p_i(\theta)$ and $p_j(\theta)$ on its boundary lies on the perpendicular bisector of line segment $\overline{p_i(\theta)p_j(\theta)}$ connecting $p_i(\theta)$ and $p_j(\theta)$. When θ varies, this bisector also moves. We orient this bisector so that, with respect to the oriented bisector, $p_i(\theta)$ is in its right side and $p_j(\theta)$ is in its left side. Regard this oriented bisector as a coordinate axis, with the middle point of the line segment $\overline{p_i(\theta)p_j(\theta)}$ as its origin.

Then, for $k \neq i, j$, points p on this axis satisfying

$$d(p, p_k(\theta)) \leqq d(p, p_i(\theta)) = d(p, p_j(\theta))$$

is given as an interval

$$[g_k^-(\theta), g_k^+(\theta)]$$

where $d(\cdot, \cdot)$ is the Euclidean distance between two points, and $g_k^-(\theta)$ and $g_k^+(\theta)$ may take a value of $-\infty$ and $+\infty$. Note that this interval is the Voronoi edge of $p_i(\theta)$ and $p_j(\theta)$ in the furthest Voronoi diagram. See Fig. 2. When $p_k(\theta)$ is collinear with $p_i(\theta)$ and $p_j(\theta)$ and is not on the line segment connecting $\overline{p_i(\theta)p_j(\theta)}$, the interval is empty, and we regard $g_k^-(\theta) = +\infty$ and $g_k^+(\theta) = -\infty$. Define $g^+(\theta)$ and $g^-(\theta)$ by

$$g^+(\theta) = \min_{k \neq i,j} g_k^+(\theta) \qquad g^-(\theta) = \max_{k \neq i,j} g_k^-(\theta)$$

Then, the following lemma holds, which states the relation between the furthest Voronoi diagram and the minimum enclosing circle (e.g., see [16]).

Lemma 4.1: (1) For fixed θ, there exists a circle containing all $p_k(\theta)$ and having $p_i(\theta)$ and $p_j(\theta)$ on its boundary if the intersection of all the intervals for $p_k(\theta)$ is nonempty, i.e., $g^-(\theta) \leqq g^+(\theta)$.

(2) When such a circle exists, the center of the minimum enclosing circle with $p_i(\theta)$ and $p_j(\theta)$ on its boundary is at

$$\begin{cases} g^+ & \text{if } g^-(\theta) \leqq g^+(\theta) < 0 \\ 0 & \text{if } g^-(\theta) \leqq 0 \leqq g^+(\theta) \\ g^- & \text{if } 0 < g^-(\theta) \leqq g^+(\theta) \end{cases}$$

on the axis. $\square$

Thus, having $g^+(\theta)$ and $g^-(\theta)$ at hand, the constrained problem for i, j can be solved. These two functions correspond to the lower and upper envelopes, respectively, of graphs of $g_k^+(\theta)$ and $g_k^-(\theta)$, and can be computed using the theory of Davenport-Schinzel sequences (see Sect. 2).

When a pair of these functions among $g_k^+(\theta)$ and $g_k^-(\theta)$ intersect (not at $\pm\infty$), the corresponding four points, including $p_i(\theta)$ and $p_j(\theta)$, become cocircular. The following lemma bounds the number of intersections.

Lemma 4.2: Four points among $p_k(\theta)$ $(k = 1, \ldots, n)$ become cocircular at most six times.

Proof: Four points $p_k(\theta)$ $(k = 1, 2, 3, 4)$ become cocircular when the following holds:

$$\begin{vmatrix} 1 & x_1(\theta,1) & y_1(\theta,1) & x_1(\theta,1)^2 + y_1(\theta,1)^2 \\ 1 & x_2(\theta,1) & y_2(\theta,1) & x_2(\theta,1)^2 + y_2(\theta,1)^2 \\ 1 & x_3(\theta,1) & y_3(\theta,1) & x_3(\theta,1)^2 + y_3(\theta,1)^2 \\ 1 & x_4(\theta,1) & y_4(\theta,1) & x_4(\theta,1)^2 + y_4(\theta,1)^2 \end{vmatrix} = 0$$

In the fourth column, all the quadratic terms $\cos^2\theta$ and $\sin^2\theta$ disappear, since $\cos^2\theta + \sin^2\theta = 1$. Then, the determinant is a six-degree polynomial of $\cos\theta$ and $\sin\theta$, and there are at most six roots. □

Lemma 4.3: The combinatorial complexity of g^+ and g^- is $O(\lambda_8(n))$. These functions can be computed in $O(\lambda_7(n) \log n)$ time.

Proof: For g^+, any two functions g_k^+ and g_l^+ intersect at most six times by Lemma 4.2. Hence, the combinatorial complexity of g^+ is $O(\lambda_8(n))$ (see Sect. 2). For g^-, similar. The time complexity follows from [6] (see also [17] for output-size sensitive results). □

Theorem 4.1: All the vertices of the maximum diagram for the collection F of n functions f_i can be computed in $O(n^2\lambda_7(n) \log n)$ time and $O(n)$ space, and the TR-fitting problem in L_2 norm can be solved in the same time and space.

Proof: For each pair of i, j, we solve the constrained problem to find the minimum enclosing circle having $p_i(\theta)$ and $p_j(\theta)$ on its boundary for moving $p_k(\theta)$ $(k = 1, \ldots, n;\ 0 \leqq \theta < 2\pi)$. Then, the smallest circle among them is an optimal solution to the TR-fitting problem. This takes $O(n^2\lambda_7(n) \log n)$ time and $O(n)$ space in total. □

4.1.2 Algorithm Using the Linearization Technique

This section gives an application of the linearization technique. First, expand the objective function given in Sect. 3:

$$\begin{aligned} &\|p_i(\theta,\alpha) - z\|_2^2 \\ &\quad = x^2 + y^2 + x_i^2 + y_i^2 + (u_i^2 + v_i^2)\alpha^2 \\ &\qquad +2u_i\alpha x + 2v_i\alpha y \\ &\qquad +2(-u_ix_i - v_iy_i)\alpha\cos\theta + 2(u_iy_i - v_ix_i)\alpha\sin\theta \\ &\qquad +2x_i(-x\cos\theta - y\sin\theta) + 2y_i(x\sin\theta - y\cos\theta) \end{aligned}$$

Note that $x^2 + y^2$ appears in common for each i, and $\alpha = 1$ here. Then, the TR-fitting problem is expressed as minimizing the maximum of linear functions under three nonlinear constraints.

$$\begin{aligned} \min \max_{i=1,\ldots,n} \ & a_{i0} + \sum_{d=1}^{6} a_{id} X_d \\ \text{s.t.} \ & X_3^2 + X_4^2 = 1 \\ & X_5 = -X_1X_3 - X_2X_4 \\ & X_6 = X_1X_4 - X_2X_3 \end{aligned}$$

where

$$\begin{aligned} a_{i0} &= x_i^2 + y_i^2 + u_i^2 + v_i^2 & & \\ a_{i1} &= 2u_i, & a_{i2} &= 2v_i \\ a_{i3} &= 2(-u_ix_i - v_iy_i), & a_{i4} &= 2(u_iy_i - v_ix_i) \\ a_{i5} &= 2x_i, & a_{i6} &= 2y_i \end{aligned}$$

and

$$\begin{aligned} & X_1 = x, \quad X_2 = y, \\ & X_3 = \cos\theta, \quad X_4 = \sin\theta, \\ & X_5 = -x\cos\theta - y\sin\theta, \\ & X_6 = x\sin\theta - y\cos\theta. \end{aligned}$$

This optimization problem can be interpreted nicely as follows. Consider the upper envelope of n linear functions $Z = a_{i0} + \sum_{d=1}^{6} a_{id}X_d$. Then the problem is to find a point with the minimum Z-coordinate on the intersection of this envelope and hypersurfaces determined by three constraints. A crucial observation here is that the size of this upper envelope is $O(n^3)$ (the upper bound theorem for convex polytopes; e.g., see [4]). since it is the boundary of the intersection of n upper halfspaces in 7-dimensional space.

A direct approach based on this formulation is to construct the intersection of the envelope with the three hypersurfaces, and to find a bottom point on the intersection. In the following, however, we take a slightly different approach, and use this formulation to construct the maximum diagram of F described at the beginning of Sect. 4.1, assuming nondegeneracies. Algorithmically, these approaches are almost the same, but, regarding our algorithm in the latter way makes it easier to use the underlying geometric properties of the problem.

In this formulation, vertices, edges and faces of the maximum diagram of F correspond to the intersection of each of d-dimensional faces for $d = 3, 4$ and 5, respectively, of the envelope with three hypersurfaces. Here, the envelope is a 7-dimensional convex polyhedron, and a d-dimensional face is called a d-face. In discussing relations between these faces, we need a facial graph of the envelope. Roughly, the facial graph of the envelope is an acyclic directed graph with one source and one sink, such that the nodes in this graph are the faces of the envelope and an arc connects faces Φ and Ψ iff Ψ is a facet of Φ (cf. [18]). The facial graph can be constructed by the output-size sensitive convex hull algorithm of Seidel [18] using duality, which in the worst case requires $O(n^3 \log n)$ time. In the worst case, the size of the facial graph becomes $O(n^3)$ in total, but practically it would be less.

We first consider computing all the vertices of the maximum diagram using this formulation. Consider 3-faces of the envelope. A 3-face is part of the intersection of four hyperplanes among n hyperplanes $Z = a_{i0} + \sum_{d=1}^{6} a_{id} X_d$ which is bounded by upward halfspaces determined by the other $n-4$ hyperplanes. The intersection between a 3-face and three hypersurfaces of the constraints is a point, which corresponds to a vertex in the maximum diagram.

Consider four hyperplanes containing a 3-face, and further consider the intersection of these four hyperplanes and the three hypersurfaces. This intersection consists of a constant number of points, to be called candidate points. Not all the candidate points really lie on the envelope, since points on the envelope should be contained in upward halfspaces determined by the other $n-4$ hyperplanes. A naive approach to compute the points on the envelope among the candidates is to check, for each candidate point, whether the candidate point is contained in the $n-4$ halfspaces. However, since the number of 3-faces are $O(n^3)$, and the test for each candidate by this approach takes $O(n)$ time, this approach requires $O(n^4)$ time.

We can enumerate all the vertices of the maximum diagram in $O(n^3)$ time, given the facial graph at hand, as follows. In the naive approach, we test $n-4$ halfspaces. For the 3-face we are considering, these $n-4$ halfspaces may really determine the boundary of the 3-face, but in ordinary cases only some of them are truly determining the boundary of the 3-face. In fact, the boundary of the 3-face is determined by 2-faces incident to the 3-face in the facial graph. By the nondegeneracy assumption, there is a hyperplane contributing to a 2-face incident to the 3-face but not contributing to the 3-face. We have to test only such hyperplanes in checking whether candidate points are on the envelope or not.

This leads to the following lemma.

Lemma 4.4: If the facial graph of this envelope is given, all the vertices of the maximum diagram of F can be computed in linear time and space with respect to the size of the facial graph (strictly, the total number of nodes of 3- and 4-faces and arcs connecting them). □

From this lemma, we see that, given the facial graph, we can enumerate all the vertices of the maximum diagram in $O(n^3)$ time and space.

To compute the edges and faces of the maximum diagram, we compute the intersection of 4- and 5-faces with the hypersurfaces in $O(n^3 \log n)$ time by making full use of the underlying geometry. We here skip the details of this step of our algorithm, and just state the result:

Theorem 4.2: The TR-fitting problem in L_2 norm can be solved in $O(n^3 \log n)$ time and $O(n^3)$ space in the worst case. □

4.2 TSR-Fitting Problem in L_2 Norm

Following Sects. 3, 4.1, the TSR-fitting problem can be formulated as the problem of finding a bottom point on the upper envelope of n functions $f_i(x, y, \theta, \alpha)$ of four variables for each point p_i as follows:

$$\min_{x,y,\theta,\alpha} \max_{i=1,\dots,n} f_i(x, y, \theta, \alpha)$$

where

$$f_i(x, y, \theta, \alpha) \equiv (x - x_i(\theta, \alpha))^2 + (y - y_i(\theta, \alpha))^2.$$

For fixed $\theta = \theta_0$, this function $f_i(x, y, \theta_0, \alpha)$ is convex with respect to x, y and α. Using this property, we first consider reducing this problem of four variables to the one-dimensional Davenport-Schinzel sequence by introducing many new functions from f_i. We then consider applying the linearization technique to this problem.

4.2.1 Algorithm Using Davenport-Schinzel Sequence of a Single Variable

The function $f_i(x, y, \theta, \alpha)$ is not convex, but, for fixed θ, it is convex with respect to x, y and α. We show below that this convexity can be utilized to get a nontrivial time bound for the TSR- and TSG-fitting problems in a theoretically easy way. We first provide the following basic lemma for convex functions. In fact, we have already used this property for fixed $\alpha = 1$, and we here omit its proof (a proof may be found in [10]).

Lemma 4.5: Let $\phi_i(\boldsymbol{x})$ be a convex function on a convex domain D in $\mathbf{R}^d$ $(i = 1, \dots, n)$. For $I \subseteqq S \equiv \{1, \dots, n\}$, let $\phi_I(\boldsymbol{x})$ be a function whose value at $\boldsymbol{x} \in D$ is the pointwise maximum of $\phi_i(\boldsymbol{x})$ $(i \in I)$. We denote the family of subsets of S whose sizes are $d+1$ by $\mathcal{S}_{d+1}$. Then we have

$$\min_{\boldsymbol{x} \in D} \max_{i=1,\dots,n} \phi_i(\boldsymbol{x}) = \max_{I \in \mathcal{S}_{d+1}} \min_{\boldsymbol{x} \in D} \phi_I(\boldsymbol{x}). \qquad \square$$

For fixed θ, $f_i(x, y, \theta, \alpha)$ is convex with respect to x, y and α, and this lemma can be used. We define a function $r_{ijkl}(\theta)$ for $i < j < k < l$ by

$$r_{ijkl}(\theta) = \min_{\alpha \geqq 0, x, y} \max_{h=i,j,k,l} f_h(x, y, \theta, \alpha).$$

$r_{ijkl}(\theta)$ is the minimum, for $\alpha \geqq 0$, of the square of the radius of the minimum enclosing circle of $p_i(\theta, \alpha)$, $p_j(\theta, \alpha)$, $p_k(\theta.\alpha)$ and $p_l(\theta, \alpha)$. Then, from Lemma 4.5, we have the following.

Lemma 4.6: For the TSR-fitting problem, we have

$$\min_{x,y,\theta,\alpha}\max_{i=1,\ldots,n} f_i(x,y,\theta,\alpha)=\min_{\theta}\max_{i<j<k<l} r_{ijkl}(\theta).$$

□

Thus, the problem is reduced to computing the upper envelope of $O(n^4)$ functions of one variable. By this definition, two functions may coincide in some interval. However, modifying the functions by case analysis leads us to an ordinary problem of Davenport-Schinzel sequences. Similarly, we can compute intersections between two functions among $r_{ijkl}(\theta)$ $(i<j<k<l)$ by case analysis. Note that $r_{ijkl}(\theta)$ can be represented by the upper envelope of four functions $f_h(x,y,\theta,\alpha)$ $(h=i,j,k,l)$, which can be computed in a constant time.

Theorem 4.3: The TSG-fitting problem in L_2 norm can be solved in $O(n^4\log^* n\log n)$ time and $O(n^4)$ space.

□

4.2.2 Algorithm Using the Linearization Technique

Using the linearization technique as in Sect. 4.1.2, the TSR-fitting problem is reduced to the problem of minimizing the maximum of linear functions of seven variables with three nonlinear constraints and an inequality constraint as follows.

$$\begin{aligned}
&\min\max_{i=1,\ldots,n} a_{i0}+\sum_{d=1}^{7}a_{id}X_d\\
&\text{s.t.}\quad X_3^2+X_4^2=X_7,\quad X_7\geqq 0,\\
&\qquad X_5X_7=-X_1X_3-X_2X_4,\\
&\qquad X_6X_7=X_1X_4-X_2X_3
\end{aligned}$$

where

$$\begin{aligned}
&a_{i0}=x_i^2+y_i^2\quad a_{i1}=2u_i,\quad a_{i2}=2v_i\\
&a_{i3}=2(-u_ix_i-v_iy_i),\qquad a_{i4}=2(u_iy_i-v_ix_i),\\
&a_{i5}=2x_i,\qquad a_{i6}=2y_i,\quad a_{i7}=u_i^2+v_i^2
\end{aligned}$$

and

$$\begin{aligned}
&X_1=\alpha x,\qquad X_2=\alpha y,\\
&X_3=\alpha\cos\theta,\quad X_4=\alpha\sin\theta,\\
&X_5=-x\cos\theta-y\sin\theta,\\
&X_6=x\sin\theta-y\cos\theta,\\
&X_7=\alpha^2.
\end{aligned}$$

In this case, the envelope of n linear functions of seven variables has the combinatorial complexity of $O(n^4)$. Again, enumerating appropriate faces in a clever way using the structure of the maximum diagram of original n functions of four variables, we obtain the following theorem.

Theorem 4.4: The TSR-fitting problem in L_2 norm can be solved in $O(n^4\log n)$ time and $O(n^4)$ space in the worst case. □

5. Geometric Fitting Problem in L_∞ Norm

Recalling the formulation of the TSR-fitting problem in L_∞ norm, if θ and α are fixed, the problem becomes the minimum enclosing square (parallel to the axis) problem, and so the problem is trivially solved. That is, the TSR-fitting problem is to minimize

$$\min_{x,y}\max_{i=1,\ldots,n}\{|x_i(\theta,\alpha)-x|,\ |y_i(\theta,\alpha)-y|\}$$

for $0\leqq\theta<2\pi$, $\alpha\geqq 0$. For fixed θ and α, this maximum can be easily minimized for x, y as

$$\begin{aligned}
h(\theta,\alpha)=\max\{&(x_{\max}(\theta,\alpha)-x_{\min}(\theta,\alpha)),\\
&(y_{\max}(\theta,\alpha)-y_{\min}(\theta,\alpha))\}/2
\end{aligned}$$

where

$$\begin{aligned}
x_{\max}(\theta,\alpha)&=\max_{i=1,\ldots,n}x_i(\theta,\alpha)\\
x_{\min}(\theta,\alpha)&=\min_{i=1,\ldots,n}x_i(\theta,\alpha)\\
y_{\max}(\theta,\alpha)&=\max_{i=1,\ldots,n}y_i(\theta,\alpha)\\
y_{\min}(\theta,\alpha)&=\min_{i=1,\ldots,n}y_i(\theta,\alpha)
\end{aligned}$$

$x_{\max}$ and $x_{\min}$ correspond to the upper and lower envelope of n functions $x_i(\theta,\alpha)$. Similarly for $y_{\max}$ and $y_{\min}$. These envelopes are envelopes of functions of a single variable in the case of TR-fitting, and those of bivariate functions in the case of the TSR-fitting problem. If these functions $x_{\max}$, $x_{\min}$, $y_{\max}$ and $y_{\min}$ are computed, the TR- and TSR-problems can be solved by merging or overlaying four 1- and 2-dimensional subdivisions of the functions, respectively, and then solving subproblems of small size. The number of subproblems is the size of the overlaid subdivision.

5.1 TR-Fitting Problem in L_∞ Norm

In this case, $\alpha=1$, and $x_{\max}$, $x_{\min}$, $y_{\max}$ and $y_{\min}$ are the upper or lower envelopes of n functions of one variable, and any pair of functions $x_i(\theta,1)$ and any pair of functions $y_i(\theta,1)$ intersect at most twice. Hence, the combinatorial complexity of these functions is just $O(n)$, and the final overlaid subdivision consists of $O(n)$ intervals. To compute these functions, it takes $O(n\log n)$, and afterwards it takes only $O(n)$ time, and the following holds.

Theorem 5.1: The TR-fitting problem in L_∞ norm can be solved in $O(n\log n)$ time. □

5.2 TSR-Fitting Problem in L_∞ Norm

In this section, we first describe a general approach that can be applicable to the TSG-fitting problem, and then discuss applying the linearization technique to the TSR-fitting problem by making use of its special structures.

5.2.1 Algorithm Using the Davenport-Schinzel Sequences of a Single Variable

In this section we discuss computing the maximum diagram of $x_{\max}(\theta, \alpha)$ by using Davenport-Schinzel sequences. Solving the TSR- and TSG-fitting problem after computing the diagrams for $x_{\max}$, $x_{\min}$, $y_{\max}$, $y_{\min}$ will be discussed at the end of Sect. 5.2.2.

Let V_i be the region in the maximum diagram in which the function $x_i(\theta, \alpha)$ attains the maximum in $x_{\max}$ $(i = 1, \ldots, n)$. That is, V_i is expressed as follows:

$$V_i = \bigcap_{j=1,\ldots,n} \{ (\theta, \alpha) \mid x_i(\theta, \alpha) \geqq x_j(\theta, \alpha) \}.$$

Rearranging terms in $x_i(\theta, \alpha) \geqq x_j(\theta, \alpha)$, we have

$$(u_i - u_j)\alpha \leqq (x_i - x_j)\cos\theta - (y_i - y_j)\sin\theta.$$

Define three sets I_i^-, I_i^0 and I_i^+ of indices by

$$\begin{aligned} I_i^- &= \{ j \mid u_i - u_j < 0 \} \\ I_i^0 &= \{ j \mid u_i - u_j = 0 \} \\ I_i^+ &= \{ j \mid u_i - u_j > 0 \} \end{aligned}$$

Then, V_i is expressed as

$$V_i = V_i^- \cap V_i^0 \cap V_i^+$$

where

$$\begin{aligned} V_i^- &= \bigcap_{j \in I_i^-} \{ (\theta, \alpha) \mid \alpha \geqq h_j(\theta) \} \\ V_i^0 &= \bigcap_{j \in I_i^0} \{ (\theta, \alpha) \mid (x_i - x_j)\cos\theta - (y_i - y_j)\sin\theta \geqq 0 \} \\ V_i^+ &= \bigcap_{j \in I_i^+} \{ (\theta, \alpha) \mid \alpha \leqq h_j(\theta) \} \end{aligned}$$

and

$$h_j(\theta) = \frac{x_i - x_j}{u_i - u_j}\cos\theta - \frac{y_i - y_j}{u_i - u_j}\sin\theta \quad (j \in I_i^- \cup I_i^+).$$

V_i^- (resp. V_i^+) can be computed from the upper (resp. lower) envelope of functions $\alpha = h_j(\theta)$, $j \in I_i^-$ (resp. $j \in I_i^+$). Both envelopes have a combinatorial complexity of $O(n)$ and the intersection of V_i^- and V_i^+ also has a combinatorial complexity of $O(n)$, since any pair among $h_j(\theta)$ intersect at most twice in $[0, 2\pi)$. Algorithmically, the two envelopes can be computed in $O(n \log n)$ time, and then the intersection $V_i^- \cap V_i^+$ can be computed in $O(n)$ time by using the monotone property of $h_j(\theta)$ with respect to θ with the merging technique.

V_i^0 consists of the intersection of n regions each of which is a vertical slab in the (θ, α) domain, when we identify $\theta = 0$ and $\theta = 2\pi$. Hence, V_i^0 has a combinatorial complexity of $O(n)$, and can be computed in $O(n \log n)$ time.

Hence, the combinatorial complexity of V_i is $O(n)$ and can be computed in $O(n \log n)$ time. Computing V_i for each i, we see that the maximum diagram of $x_{\max}$ has a combinatorial complexity of $O(n^2)$ and can be computed in $O(n^2 \log n)$ time. These bounds for the TSR-fitting problem are worse than those given in the next section, but the arguments given so far can be applied to the TSG-fitting problem.

In the TSG-fitting problem, the combinatorial complexity of each V_i is shown to be $O(n \log^* n)$, and V_i itself can be computed in $O(n \log^* n \log n)$ time. This implies that the maximum diagram of $x_{\max}$ in the TSG-fitting has complexity $O(n^2 \log^* n)$, and can be computed in $O(n^2 \log^* n \log n)$ time. To solve the TSG-fitting problem, we have to further compute the overlaid subdivision of four diagrams of $x_{\max}$, $x_{\min}$, $y_{\max}$ and $y_{\min}$. This step can be performed as described in the next section, and we have the following theorem.

Theorem 5.2: The TSG-fitting problem in L_∞ norm can be solved in $O((n^2 \log^* n)^2)$ time and space in the worst case. □

Although the worst-case time complexity becomes almost $O(n^4)$, this seems too pessimistic, since an almost quadratic bound for the combinatorial complexity of $x_{\max}$ itself may not be so tight, and the complexity of the overlaid subdivision may not be close to $O((n^2)^2)$ for some specific transformation G.

5.2.2 Algorithm Using the Linearization Technique

Using the linearization technique, computation of the upper envelope of $x_{\max}(\theta, \alpha)$ can be done by considering the following:

$$\begin{aligned} &\max_{i=1,\ldots,n} \ x_i X_1 + y_i X_2 + u_i X_3 \\ &\quad \text{s.t.} \quad X_1^2 + X_2^2 = 1, \quad X_3 \leqq 0 \end{aligned}$$

Neglecting the constraints, this problem is to compute the upper envelope of n linear functions $Z = x_i X_1 + y_i X_2 + u_i X_3$ in 4-dimensional space. In this case, all of these functions pass through the origin, so that the envelope essentially has a 3-dimensional structure, and hence its size is just $O(n)$. Furthermore, since $X_1^2 + X_2^2 \leqq 1$ determines a convex region, the subdivision, maximum diagram, of $x_{\max}$ is seen to consist of $O(n)$ elements. Using the three-dimensional convex hull algorithm, this subdivision can be computed in $O(n \log n)$ time.

To solve the TSR-fitting problem, four subdivisions of $x_{\max}$, $x_{\min}$, $y_{\max}$ and $y_{\min}$ must be overlaid. Let K be the size of the overlaid subdivision. Since the size of each subdivision is $O(n)$, $K = O(n^2)$. Mairson and Stolfi [11] show that two planar subdivisions can be merged in $O(K' + N \log N)$ time and $O(K' + N)$ space where N is the number of vertices in the original subdivisions and K' is the number of vertices in the merged subdivision. In our case, we have to merge four subdivisions. We can compute the overlaid subdivision of four in $O(K + n\log n)$ time and $O(K + n)$ space by overlaying any pair of subdivisions among four by their algorithm and then merging the computed subdivisions by simply traversing the linked list representing the subdivisions. Given the overlaid subdivision at hand, the problem can be solved in $O(K)$ time. Thus, the following holds.

Theorem 5.3: The TSR-fitting problem in L_∞ norm can be solved in $O(K + n\log n)$ time and $O(K + n)$ space, where K is the number of distinct ordered four indices (i, j, k, l) such that x_i, x_j, y_k, y_l attain the extremal values in $x_{\max}$, $x_{\min}$, $y_{\max}$, $y_{\min}$, respectively, for some θ and α. K is $O(n^2)$ in the worst case. □

6. Concluding Remarks

We have presented efficient algorithms for several types of the geometric fitting problem. Our approach is to regard the problem as that of finding a minimum point on the upper envelope of multivariate functions. We have analyzed the combinatorial complexity of these envelopes, and devised algorithms to construct them.

The analysis of the TR-fitting problem in L_2 norm in Sect. 3 can be used to get nontrivial bounds on the combinatorial complexity of furthest Voronoi diagrams for moving points directly. Also, it can be modified to get bounds for nearest Voronoi diagrams. For example, this approach gives an $O(n^3)$ bound on the combinatorial complexity of nearest and furthest Voronoi diagrams for points moving in different fixed directions with different constant velocity.

In this paper, we have concentrated on the geometric fitting between two corresponding sets. As noted in the introduction, it has been shown [2] that the fitting problem where no correspondence between two sets is given in advance can be solved in relatively high polynomial time. In [2], the L_2 case and the L_∞ case are treated analogously, and there is almost no difference in the results for the two cases. However, as we have demonstrated here, in solving the fitting problem with a given correspondence, the L_∞ problem is easier to solve than the L_2 problem. We might be able to improve the results in [2] for the L_∞ case. At least the T-fitting problem between two sets of points with no specified correspondence in the L_∞ case can be solved faster.

Another generalization of our results would be to extend the objects from points to more complicated geometric objects. For instance, the geometric fitting between two polygons seems to be a very important and interesting problem.

Acknowledgment

The authors would like to thank anonymous referees for their valuable comments on the original version of this paper. Part of the work of the first and third authors was supported in part by a Grant-in-Aid from the Ministry of Education, Science and Culture of Japan.

References

[1] G. Albers, L.J. Guibas, J.S.B. Mitchell, and T. Roos, "Voronoi diagrams of moving points," International Journal of Computational Geometry & Applications, vol.8, no.3, pp.365–379, 1998.
[2] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl, "Congruence, similarity, and symmetries of geometric objects," Discrete and Computational Geometry, vol.3, pp.237–256, 1988.
[3] M.J. Atallah, "Some dynamic computational geometry problems," Computers and Mathematics with Applications, vol.11, pp.1171–1181, 1985.
[4] H. Edelsbrunner, "Algorithms in Combinatorial Geometry," Springer-Verlag, 1987.
[5] H. Edelsbrunner, J. Pach, J.T. Schwartz, and M. Sharir, "On the lower envelope of bivariate functions and its applications," Proc. 28th IEEE Annual Symposium on Foundations of Computer Science, pp.27–37, 1987.
[6] J. Hershberger, "Finding the upper envelope of n line segments in $O(n\log n)$ time," Information Processing Letters, vol.33, no.4, pp.169–174, 1989.
[7] K. Imai, "Voronoi diagrams for moving points and its applications," Transactions of the Japan Society for Industrial and Applied Mathematics, vol.1, pp.127–134, 1991.
[8] H. Imai and K. Imai, "Computational Geometry," Kyoritsu Shuppan, Tokyo, 1995.
[9] K. Imai and H. Imai, "Dynamic weighted Voronoi diagrams and weighted minimax matching of two corresponding point sets," Optimization Methods and Software, 1998, in press.
[10] K. Imai, S. Sumino, and H. Imai, "Minimax geometric fitting of two corresponding sets of points," Proc. 5th ACM Symposium on Computational Geometry, pp.266–275, 1989.
[11] H.G. Mairson and J. Stolfi, "Reporting and counting intersections between two sets of line segments," Nato Adv. Study Inst. on Theoretical Foundations of Computer Graphics and CAD, Lucca, Italy, July 1987.
[12] S.J. Maybank, "Applications of algebraic geometry to computer vision," in Computational Algebraic Geometry, Nice, 1992, Progr. Math., 109, Birkhäuser, Boston, pp.185–194, 1993.
[13] N. Megiddo, "Linear programming in linear time when the dimension is fixed," J. ACM, vol.31, pp.114–127, 1984.
[14] N. Megiddo, "Dynamic location problems," Annals of Operations Research, vol.6, pp.313–319, 1986.
[15] S. Naito, "Applications of visual sensors to industrial robots," Sensor Technology, vol.7, pp.90–93, 1987.
[16] F.P. Preparata and M.I. Shamos, "Computational Geometry: An Introduction," Springer-Verlag, 1985.
[17] E.A. Ramos, "Construction of 1-d lower envelopes and applications," Proc. 13th Annual ACM Symposium on Computational Geometry, pp.57–66, 1997.
[18] R. Seidel, "Small-dimensional linear programming and convex hulls made easy," Discrete and Computational Geometry, vol.6, no.5, pp.423–434, 1991.
[19] M. Sharir, private communication, 1988.
[20] M. Sharir, "Almost tight upper bounds for lower envelopes in higher dimensions," Discrete and Computational Geometry, vol.12, pp.327–345, 1994.
[21] M. Sharir and P.K. Agarwal, "Davenport-Schinzel Se-

quences and Their Geometric Applications," Cambridge University Press, Cambridge, 1995.

[22] E. Szemerédi, "On a problem by Davenport and Schinzel," Acta Arithmetica, vol.25, pp.213–224, 1974.

[23] A. Takahashi, I. Ishii, H. Makino, and M. Nakashizuka, "A camera calibration method using parallelogrammatic grid points," IEICE Trans. Inf. & Syst., vol.E79-D, no.11, pp.1579–1587, 1996.

Keiko Imai received the B.S., M.S. and Dr.S. degrees in Mathematics from Tsuda College in 1980, 1982 and 1991, respectively. Since 1992 she has been an associate professor at the Department of Information and System Engineering, Chuo University. Her research interests include design and analysis of algorithms, data structures and computational geometry. She is a member of IPSJ, JSIAM, OR Soc. Japan and Math. Soc. Japan.

Shigeo Sumino obtained B.Eng. and M.Eng. in Department of Computer Science and Communication Engineering, Kyushu University in 1988 and 1990, respectively. Since 1990, he has been a researcher at Central Research Laboratory, Hitachi, Ltd. His research interests include human interface, groupware and computational geometry. He is a member of IPSJ.

Hiroshi Imai obtained B.Eng. in Mathematical Engineering, and M.Eng. and D.Eng. in Information Engineering, University of Tokyo in 1981, 1983 and 1986, respectively. In 1986∼1990, he was an associate professor of Department of Computer Science and Communication Engineering, Kyushu University. Since 1990, he has been an associate professor at Department of Information Science, University of Tokyo. His research interests include algorithms, computational geometry, and optimization. He is a member of IPSJ, OR Soc. Japan, ACM and IEEE.