# Fitting by two-joint orthogonal chains

J. M. Díaz-Báñez[*] M. A. López[†] C. Seara[‡] I. Ventura[§]

27th July 2009

**Abstract**

We study the problem of fitting a two-joint orthogonal polygonal chain to a set $S$ of $n$ points in the plane, where the objective function is to minimize the maximum orthogonal distance from $S$ to the chain. We show that this problem can be solved in $\Theta(n)$ time if the orientation of the chain is fixed, and in $\Theta(n \log n)$ time when the orientation is not a priori known.

## 1 Introduction and definitions

Fitting a curve of a certain type to a given point set in the plane is a fundamental problem with applications in fields as diverse as statistics, computer graphics, and artificial intelligence. A special case of this problem is the so called *polygonal approximation problem* or *polygonal fitting problem*, where a polygonal chain with $k$ corners or joints is fitted to a data set so as to minimize the approximation error according to some agreed upon metric. This problem is closely related to that of approximating a piecewise-linear curve with $n$ edges by one with fewer edges, except that the input is now also a chain. Applications of this problem arise in cartography, pattern recognition, and graphic design [5, 8, 19], and has received much attention in computational geometry [1, 6, 12, 15, 21].

In the *Min-Max problem* a polygonal chain with $k$ joints is fitted to a data set with the goal of minimizing the maximum vertical distance from the input points to the chain (the *Chebyshev error*). This problem was first posed by Hakimi and Schmeichel [13] and solved in $O(n^2 \log n)$ time. The complexity has since been improved, first by Wang et al. [24] to $O(n^2)$ time and then by Goodrich [11] to $O(n \log n)$ time.

We consider the case in which the approximating curve is an *orthogonal polygonal chain*, i.e., a chain of consecutive orthogonal line segments where the extreme segments are half-lines with the same slope, *the slope of the orthogonal polygonal chain.* The case in which this slope is given was first solved by Díaz-Báñez and Mesa [9] in $O(n^2 \log n)$ time, and subsequently improved by Wang [23] to $O(n^2)$ time, and by Lopez and Mayster [18] to $\min\{n^2, nk \log n\}$ time. Very recently, Fournier and Vigneron [10] give an $O(n)$ time algorithm if the points are sorted by their $x$-coordinates, and an $O(n \log n)$ time algorithm for the unsorted case. For the unsorted case these authors prove the optimality if $k = \Theta(n)$ and give an $\Omega(n \log k)$ lower bound for the decision problem.

[*]Departamento de Matemática Aplicada II, Universidad de Sevilla, Spain, dbanez@us.es. Partially supported by project MEC-MTM2006-03909.

[†]Department of Mathematics, University of Denver, 2360 South Gaylord Street, Denver, CO 80208, USA.

[‡]Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Spain, carlos.seara@upc.edu. Supported by projects MEC-MTM2006-01267 and DURSI 2009SGR1040.

[§]Departamento de Matemática Aplicada II, Universidad de Sevilla, Spain, iventura@us.es. Partially supported by project MEC-MTM2006-03909.

Let $S = \{p_1, \ldots, p_n\}$ be a set of $n$ points in the plane in general position. A *$k$-orthogonal polygonal chain* with orientation $\theta$, $\mathcal{O}_{k,\theta}$, $k \geq 1$, $0 \leq \theta < \pi$, is a chain of $2k-1$ consecutive orthogonal segments such that the extreme segments are in fact half-lines with slope $\tan(\theta)$. Thus, $\mathcal{O}_{k,\theta}$ consists of $k$ segments with slope $\tan(\theta)$ and $k-1$ segments with slope $\tan(\theta + \frac{\pi}{2})$ (Figure 1). Clearly, $\mathcal{O}_{k,\theta}$ is always monotone with respect to its orientation.

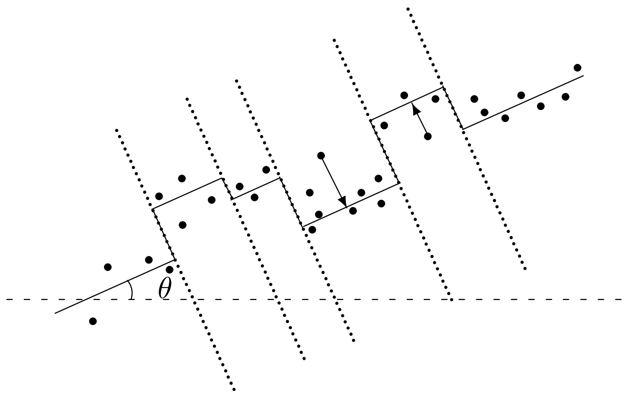

Figure 1: A 6-orthogonal polygonal chain dividing the plane into 6 strips.

We deal with the problem of fitting a $k$-orthogonal polygonal chain $\mathcal{O}_{k,\theta}$ to the set $S$. Fitting $\mathcal{O}_{k,\theta}$ to $S$ means to locate $\theta$-oriented segments $s_i(\theta)$, $i = 1, \ldots, k$, according to a given optimization criterion. We consider the Min-Max criterion, illustrated in Figure 1 and defined as follows. Let $l_i(\theta)$ be the line passing through $p_i \in S$ with orientation $\theta + \frac{\pi}{2}$. The fitting distance between $p_i$ and $\mathcal{O}_{k,\theta}$, denoted by $d_f(p_i, \mathcal{O}_{k,\theta})$, is given by

$$d_f(p_i, \mathcal{O}_{k,\theta}) = \min_{p \in l_i(\theta) \cap \mathcal{O}_{k,\theta}} d(p_i, p).$$

Notice that $d_f$ is not the Euclidean distance. However, we can assume that this distance is the Euclidean distance between $p_i$ and a point on a segment with orientation $\theta$ in $\mathcal{O}_{k,\theta}$. The *error tolerance* of $\mathcal{O}_{k,\theta}$ with respect to $S$, denoted by $\mu(\mathcal{O}_{k,\theta}, S)$, is the maximum fitting distance between the points of $S$ and $\mathcal{O}_{k,\theta}$, i.e.,

$$\mu(\mathcal{O}_{k,\theta}, S) = \max_{p_i \in S} d_f(p_i, \mathcal{O}_{k,\theta}).$$

**Definition 1** *The $k$-fitting problem for $S$ with the Min-Max criterion consists of finding an orthogonal polygonal chain $\mathcal{O}_{k,\theta}$ such that its error tolerance $\mu(\mathcal{O}_{k,\theta}, S)$ is minimized.*

Notice that if the orientation of $\mathcal{O}_{k,\theta}$ is fixed, for example $\theta = 0$, then the $k$-fitting problem consists of finding an $x$-monotone rectilinear path formed by $2k-1$ segments with minimum error tolerance where the fitting distance is just the vertical distance [9].

We focus here on the case where $k$ is small, in fact $k = 2$, and the points in $S$ are not sorted. We study the 2-fitting problem for $S$ with fixed orientation (the *oriented 2-fitting problem*) and the problem of finding the best orientation for fitting a two-joint orthogonal polygonal chain to $S$ (the *un-oriented 2-fitting problem*). We also consider the extension of the problem to three-dimensions where an orthogonal polygonal chain is a configuration of orthogonal planes. See Chen and Wang [7] for recent results on some variants of this problem including NP-hard results in three dimensions.

*Outline of the paper.* In Section 2 we study the oriented fitting problem in the plane. In Section 3 we study the un-oriented 2-fitting problem in the plane. Finally, in Section 4, we study the oriented 2-fitting problem in three-dimensions.

## 2 The oriented fitting problem

In this section we consider the oriented $k$-fitting problem for $S$, i.e., the case where the orientation $\theta$ of the $k$-orthogonal chain that fits $S$ is fixed. Without loss of generality we assume that $\theta = 0$. Thus, we are looking for an $x$-monotone rectilinear path, $\mathcal{O}_{k,0}$ or $\mathcal{O}_k$, consisting of an alternating sequence of $k$ horizontal and $k-1$ vertical segments with minimum error tolerance.

Often, the algorithms proposed in the literature for these kind of fitting problem assume that the input points are given in sorted order. Recently, Fournier and Vigneron [10] give an $O(n \log n)$ time algorithm for the oriented $k$-fitting problem when the points are unsorted and prove its optimality when $k = \Theta(n)$. The running time of the algorithm from Lopez and Mayster [18] is $\min\{n^2, nk \log n\}$ which is $O(n \log n)$ when $k$ is a constant. For the sorted case, Fournier and Vigneron [10] present an optimal $O(n)$ time algorithm and an $\Omega(n \log k)$ time lower bound for the decision problem. Here we consider the oriented $k$-fitting problem for the case $k = 2$ for the unsorted case.

Let $S = \{p_1, \ldots, p_n\}$, where $p_i = (x_i, y_i)$. We can compute $y_{\max} = \max\{y_1, \ldots, y_n\}$ and $y_{\min} = \min\{y_1, \ldots, y_n\}$ in linear time. The oriented 1-fitting problem then is solved in $O(n)$ time by finding the horizontal line $y = (y_{\max} + y_{\min})/2$.

Let $\mathcal{O}_2$ denote an optimal solution to the oriented 2-fitting problem for a set $S$. Then $\mathcal{O}_2$ consists of two horizontal half-lines joined by a vertical segment contained in a vertical line $\ell^*$ which partitions $S$ into subsets $S_1$ and $S_2$, namely the points of $S$ to the left and to the right of $\ell^*$ respectively. Since $\ell^*$ must minimize the maximum error tolerance of $S_1$ and $S_2$, the following is apparent.

**Lemma 1** *Line $\ell^*$ separates the two points in $S$ with $y$-coordinates $y_{\min}$ and $y_{\max}$.*

**Linear time algorithm for oriented 2-fitting.**

Any vertical line $\ell$ between two points of $S$ induces a candidate solution to the oriented 2-fitting problem whose cost is given by $\max\{\epsilon_1, \epsilon_2\}$, where $\epsilon_1$ ($\epsilon_2$) denotes the tolerance of the subset of $S$ to the left (right) of $\ell$. Our algorithm performs a binary search based on the following observation:

**Lemma 2** *If $\ell$ is not optimal and $\epsilon_1 < \epsilon_2$ ($\epsilon_1 > \epsilon_2$) then $\ell^*$ lies to the right (left) of $\ell$.*

We now outline the algorithm. Let $\ell$ denote the vertical line through the median $x$-coordinate of $S$. Partition $S$ into subsets $S_1$ and $S_2$ to the left and right of $\ell$, respectively. Compute also the tolerances $\epsilon_1$ of $S_1$ and $\epsilon_2$ of $S_2$, and store the two witness pairs of points responsible for the tolerances. All this can be computed in $O(n)$ time. If $\epsilon_1 = \epsilon_2$, stop the algorithm, as $\ell^* = \ell$. If $\epsilon_1 < \epsilon_2$, in $O(n/2)$ time compute the median of $S_2$, reset $\ell$ as the vertical line at this median value, compute the subsets $S_2'$ and $S_2''$ of the bipartition of $S_2$ produced by the new median, compute the tolerances $\epsilon_2'$ and $\epsilon_2''$ of $S_2'$ and $S_2''$. Compute $\max\{\epsilon_2', \epsilon_2''\}$ to decide the next move of $\ell$ (left or right). Store the tolerance values and the corresponding witness pairs as temporary values. Next compute and update the tolerances once we know the next move of $\ell$. (If $\epsilon_1 > \epsilon_2$ we proceed in a symmetric way). Compare the new tolerance values and continue recursively translating the line $\ell$ left or right by computing the new median of a subset with half of the points and updating the new tolerance values (left and right) from the old ones. At all times we have two unions at the extremes and an unknown zone in between containing at most two strips. The points in the zone are known but, in general, are not sorted.

Clearly, the time complexity of the algorithm is $T(n) = T(n/2) + O(n) = O(n)$. By Lemma 1 the optimal line $\ell^*$ will be located between the two points in $S$ with $y$-coordinates $y_{\min}$ and $y_{\max}$. The algorithm stops when either the tolerance values $\epsilon_1$ and $\epsilon_2$ are equal, or when translating $\ell$ left and right the bigger of the two tolerances switches sides. In this last case the solution will be the best of the two. Since the algorithm performs a binary search on a unimodal function, the method is correct. Notice that the solution (position of line $\ell$ or bipartition of $S$) is not unique because in an optimal solution some points can belong to $S_1$ or $S_2$ without changing the solution. Notice also that our algorithm does not sort the input points. We have the following result.

**Theorem 1** *The oriented 2-fitting problem can be solved in $\Theta(n)$ time and space.*

By the $\Omega(n \log k)$ lower bound for the decisional oriented $k$-fitting problem [10] in the unsorted case, it is clear that if $k = \omega(1)$ there is no linear time algorithm. Thus, we raise the following open question more from a theoretical than from a practical point of view.

**Open problem 1** *For which values of $k \geq 3$ does there exist a linear time algorithm for the oriented $k$-fitting problem?*

## 2.1 An $O(n \log n)$-time algorithm

We now describe an $O(n \log n)$-time algorithm for the oriented 2-fitting problem whose interest derives not from its time complexity but from the fact that it will be used as a preprocess step in the $O(n \log n)$-time algorithm for the un-oriented 2-fitting problem discussed in Section 3. We start by introducing a basic tool.

In [16, 20] the maxima problem for a point set $S$ in the plane is considered. Concretely, given two points $p_i, p_j \in S$, the following dominance relation is established: *$p_i$ dominates $p_j$ ($p_j \prec p_i$), if $x_j \leq x_i$ and $y_j \leq y_i$.* The relation $\prec$ is a partial order in $S$. A point $p_i \in S$ is called *maximal* if there does not exists $p_j \in S$ such that $i \neq j$ and $p_i \prec p_j$. The maxima problem consists of finding all the maximal points of $S$ under dominance. One can formulate maxima problems for each quadrant in the plane. We are interested in the set of maxima points for $S$ with respect to the four quadrants which form the *quadrant hull* of $S$ (Figure 2). Each set of maxima has a *total ordering* and can be organized as a height balanced search tree [20].

**Theorem 2** [16] *The maxima problem for $S$ with respect to any of the four quadrants can be solved optimally in $\Theta(n \log n)$ time and $O(n)$ space.*

**$O(n \log n)$-time algorithm for oriented 2-fitting.**

1. Let $x_{\max}$, $x_{\min}$, $y_{\max}$, and $y_{\min}$ denote the respective maximum and minimum of the $x$ and $y$-coordinates of the points in $S$. Without loss of generality assume $x_{\min} = y_{\min} = 0$ and $p_1 = (0, y_1)$, $p_n = (c, y_n)$, $p_i = (x_i, y_{\max})$, and $p_j = (x_j, 0)$ (Figure 2), i.e., the rectangle with corners at $(0, 0)$ and $(c, y_{\max})$ is the axis-parallel bounding box for $S$. Assume further that $p_i$ is strictly to the left of $p_j$ and thus, by Lemma 1, the vertical line $\ell$ lies between $p_i$ and $p_j$. (If both have the same $x$-coordinate the solution is trivial.)

   By Theorem 2, in $O(n \log n)$ time we can compute the *rectilinear convex hull* of $S$ formed by the staircases structure as in Figure 2. Notice that staircases of opposite quadrants can intercross. Since $p_i$ is to the left of $p_j$, then the third quadrant staircase gives the lower point on the left of $\ell$ and the first quadrant staircase gives the upper point on the right of $\ell$.

2. By Lemma 1 the vertical line $\ell := (x = a)$ is between $x_i$ and $x_j$. In order to find its right location we do a *binary search* over the points in the staircase structure (first and third quadrant) in $O(\log n)$ time getting the best balance between the error tolerance on the left and on the right side of $\ell$, i.e.,

$$\min_{x_i \le a < x_j} a \quad \text{subject to } \max_{x_k \le a}\{y_{\max} - y_k\} \ge \max_{x_m > a} y_m \tag{1}$$

or

$$\min_{x_i < a \le x_j} a \quad \text{subject to } \max_{x_k \le a}\{y_{\max} - y_k\} \le \max_{x_m > a} y_m \tag{2}$$

For at least one of the equations (1) or (2) there exists a solution. In case (1) the error tolerance of $S$ is given by the points to the left of $\ell$ and, in case (2), by the points to the right of $\ell$. In constant time compute this error tolerance given by the difference between the bigger and smaller $y$-coordinates of the points to the left or right of line $\ell$.

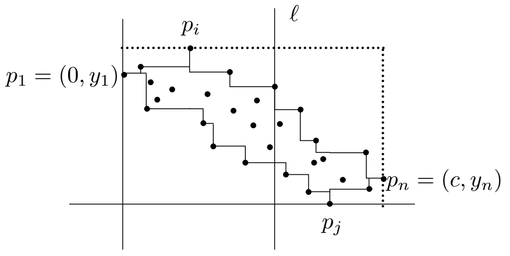

Figure 2: A quadrant hull of $S$ formed by the maxima points of $S$.

If $p_i$ is to the right of $p_j$, then the algorithm is similar with the obvious changes: the second quadrant staircase gives the upper point to the left of $\ell$ and the fourth quadrant staircase gives the lower point to the right of $\ell$.

The above staircase structure can be used to design $O(n \log n)$ time algorithms for the oriented 3-fitting and 4-fitting problems. We consider 3-fitting first. By Lemma 1 at least one of the two vertical lines, $\ell_1$ and $\ell_2$, containing the vertical segments of the approximating chain must be between $y_{\max}$ and $y_{\min}$. Assume that $\ell_1$ is to the left of $\ell_2$. There are at most a linear number of locations for $\ell_1$ between two consecutive points of the staircase structure. For each of these locations doing binary search over the staircase structure to the right of $\ell_1$ we can compute the optimal location for $\ell_2$ in $O(\log n)$ time. Details are omitted but the binary search depends on whether the location of $\ell_2$ is either between $y_{\max}$ and $y_{\min}$ or to the right of $y_{\min}$.

It is clear that for the oriented 4-fitting problem with three vertical lines $\ell_1$, $\ell_2$, and $\ell_3$, we can proceed in a similar way, first fixing the location of the median line, say $\ell_2$, in each of the linear number of possible locations, and then finding the locations of $\ell_1$ (to the left of $\ell_2$) and $\ell_3$ (to the right of $\ell_2$) by binary search on the staircase structure.

As a consequence of the discussion above, both the oriented 3- and 4-fitting problems can be solved in $O(n \log n)$ time and $O(n)$ space.

## 3 The un-oriented 2-fitting problem

In this section we consider the problem of fitting $S$ using an un-oriented 2-orthogonal polygonal chain $\mathcal{O}_{2,\theta}$ with free orientation $\theta$. Notice that the un-oriented 1-fitting problem for $S$ is equivalent to the problem of computing the width of $S$. If we know the convex hull of $S$ this problem can be solved in $O(n)$ time using rotating calipers [14]. Otherwise, computing the width of $S$ has an $\Omega(n \log n)$-time lower bound [17]. Therefore the un-oriented 1-fitting problem for $S$ can be solved optimally in $\Theta(n \log n)$ time.

Before studying the un-oriented 2-fitting problem we introduce some notation and tools which will be useful later. We start by reviewing some definitions and results from Avis et al. [2] concerning the computation of un-oriented $\Theta$-maxima points of a planar point set $S$.

**Definition 2** [2] *A ray from a point $p \in S$ is called a maximal ray if it passes through another point $q \in S$. A cone is defined by a point $p$ and two rays $C$ and $D$ emanating from $p$. A point $p \in S$ is an un-oriented $\Theta$-maximum with respect to $S$ if and only if there exist two maxima rays, $C$ and $D$, emanating from $p$ with an angle at least $\Theta$ between them so that the points of $S$ lie outside the ($\Theta$-angle) cone defined by $p$, $C$ and $D$ (Figure 3(a)).*

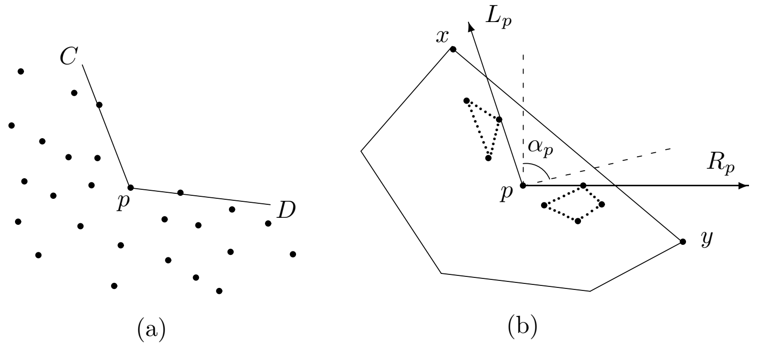

Figure 3: Un-oriented $\Theta$-maximum with respect to $S$.

**Theorem 3** [2] *All un-oriented $\Theta$-maxima points of $S$ for $\Theta \geq \pi/2$ can be computed in $O(n \log n)$ time and $O(n)$ space, and the algorithm is optimal for fixed values of $\Theta$.*

For $\Theta = \pi/2$, the output of the algorithm of Theorem 3 is the list of all the un-oriented $\pi/2$-maxima points that are apices of the wedges that have bounding rays (crossing an edge of $CH(S)$) with aperture angle at least $\pi/2$. For every such maximal point $p$ the output also contains the two rays $L_p$ and $R_p$ defining the widest empty wedge from $p$ (Figure 3(b)). Since the aperture angle is at least $\pi/2$, then each maximal point $p$ can have at most three disjoint wedges. We can translate in constant time each one of these wedges by an orientation interval in $\mathbb{S}^2$ corresponding to directions of bisectors of wedges with aperture angle $\pi/2$ contained in those wedges (Figure 3(b)). Thus, each maximal point $p \in S$ can have at most three disjoint orientation intervals in $\mathbb{S}^2$, say $\alpha'_p$, $\alpha''_p$, and $\alpha'''_p$, such that for each orientation inside this interval the point $p$ is $\pi/2$-maximum. Notice that all the points in the convex hull of $S$ are $\pi/2$-maximum, and that the total number of orientation intervals is linear.

Now we consider the un-oriented 2-fitting problem. An optimal solution for this problem is given by an orthogonal polygonal chain $\mathcal{O}_{2,\theta}$ with orientation $\theta$ such that the error tolerance of $S$ with respect to $\mathcal{O}_{2,\theta}$ is minimum. Clearly, this is equivalent to the problem of determining a line $\ell_\theta$ with slope $\tan(\pi/2+\theta)$ that splits $S$ into subsets $S_{l_\theta}$ and $S_{r_\theta}$, where $e^u_{l_\theta}$ and $e^b_{l_\theta}$ ($e^u_{r_\theta}$ and $e^b_{r_\theta}$) are the points responsible for the error tolerance of $S_{l_\theta}$ ($S_{r_\theta}$) and such that $\mathcal{O}_{2,\theta}$ minimizes the error tolerance of $S$ over all values of $\theta$. Accordingly, the error tolerance is given by the following formula, where $d_\theta(p,q)$ denotes the distance between parallel lines through $p$ and $q$ with orientation $\theta$.

$$\mu(\mathcal{O}_{2,\theta}, S) = \max_{p_i \in S} d(p_i, \mathcal{O}_{2,\theta}) = \max\{d_\theta(e^u_{l_\theta}, e^b_{l_\theta}), d_\theta(e^u_{r_\theta}, e^b_{r_\theta})\}.$$

Let $y_{\min,\theta}$ and $y_{\max,\theta}$ be the minimum and maximum $y$-coordinates of the points in $S$ when the coordinate system is rotated by angle $\theta$. The following lemma is a generalization of Lemma 1.

**Lemma 3** *Given an orientation $\theta$, an optimal solution for the un-oriented 2-fitting problem with orientation $\theta$ is defined by a line $\ell_\theta$ passing through a point of $S$ which separates the points of $S$ with $y$-coordinates $y_{\min,\theta}$ and $y_{\max,\theta}$.*

*Description of the un-oriented 2-fitting algorithm.* The goal of our approach is to adapt the $O(n \log n)$-time algorithm for oriented 2-fitting described earlier to account for continuous changes in the orientation $\theta$, looking for the optimal $\mathcal{O}_{2,\theta}$ chain in the process. To do this we update the staircase structure as $\theta$ varies and use Lemma 3 to look for an optimal solution.

• *Initialization*: The starting situation is the staircase structure formed by the four sets of maxima with respect to the four quadrants of the coordinate system when $\theta = 0$. Analogously to [16, 20], we use a height-balanced search tree to store and compute each of the four sets of maxima with insertions or deletions in optimal $O(n \log n)$ time. Thus, we compute the staircase structure and its corresponding optimal solution in $O(n \log n)$ time as we did for the oriented case.

• *Update as $\theta$ sweeps over* $[0, 90]$: As we rotate the coordinate system according to the orientation $\theta$ in discrete steps from $\theta = 0$ to $\theta = 90$ to compute the un-oriented optimal solution, we identify the four quadrants by its oriented bisectors, i.e., by the oriented lines with slopes $\tan(\theta + 45)$, $\tan(\theta + 135)$, $\tan(\theta + 225)$, and $\tan(\theta + 315)$. The staircases are formed by the four sets of maxima points in $S$ with respect to the bisectors of the current four quadrants.

The main idea of the algorithm is to rotate the coordinate system by $\theta$, and update the staircase structure by inserting or deleting points to each of the staircases as the orientation $\theta$ changes. To do this we maintain four ordered lists of the current un-oriented $\pi/2$-maxima points of $S$ with respect to the bisectors with orientations $\theta + 45$, $\theta + 135$, $\theta + 225$, and $\theta + 315$. The lists correspond to the sequences of points in the four staircases. Precisely, the staircase structure will be maintained with insertions and deletions of points induced by the changes in $\theta$. Notice that for any orientation $\theta$ the staircase structure has linear size, and updating a point on it can be done in $O(\log n)$ time as in the $\theta = 0$ case [16, 20].

As $\theta$ changes, the four staircases can be modified because either a new point of $S$ becomes $\pi/2$-maxima or some current $\pi/2$-maxima point of $S$ has to be deleted. To determine the sequence of events, as $\theta$ changes, we use Theorem 3 to pre-compute in $O(n \log n)$ time the set of all un-oriented $\pi/2$-maxima points of $S$ together with their respective orientation intervals in $\mathbb{S}^2$. These orientation intervals are the intervals where each point is un-oriented $\pi/2$-maxima for $S$. Notice that a point can be $\pi/2$-maxima for at most 3 (disjoint) orientation intervals and, consequently, the total number of changes in the staircase structure is linear.

To know in advance the sequences of events, i.e., the insertion/deletion of points in the staircase structure, along the rotation, we proceed as follows. Suppose that we have computed the orientation intervals for each point $p_i \in S$. Figure 4 represents the set of these orientation intervals. A point $p \in S$ can have at most 3 disjoint orientation intervals. We sweep the set of these intervals from 0 to $2\pi$, knowing for a particular orientation $\theta$ which are the set of $\pi/2$-maxima points for that orientation which is the set of intervals pierced by a vertical line.

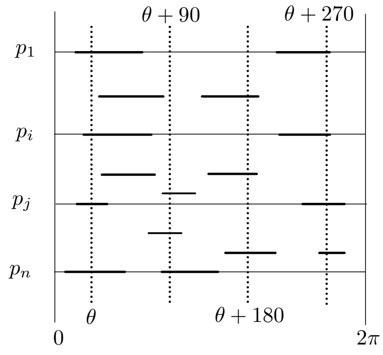

Figure 4: Sweeping the orientation intervals of the $\pi/2$-maxima points of $S$.

In fact, the algorithm performs a sweep of these intervals from $\theta = 0$ to $\theta = 90$ by vertical lines with orientations $\theta$, $\theta + 90$, $\theta + 180$, and $\theta + 270$, stopping at each event (endpoint interval) and updating the staircase structure. Since the points in $S$ are in general position, only a constant number of updates can occur at some event. We compute the optimal solution for the staircase structure between two consecutive events by computing a line $\ell_\theta$ with slope $\tan(\theta + 90)$.

Consider consecutive events $\theta_1$ and $\theta_2$. Lemma 3 implies that for a fixed value $\theta \in [\theta_1, \theta_2]$, the line $\ell_\theta$ that gives the optimal solution has to separate the points with the current $y$-coordinates $y_{\min,\theta}$ and $y_{\max,\theta}$. Thus, the optimal solution is determined by two pairs of points, either (i) $(y_{\max,\theta}, e^b_{l_\theta})$ and $(e^u_{r_\theta}, y_{\min,\theta})$ if $y_{\max,\theta}$ is on the left side of $y_{\min,\theta}$, or (ii) $(e^u_{l_\theta}, y_{\min,\theta})$ and $(y_{\max,\theta}, e^u_{r_\theta})$ if $y_{\max,\theta}$ is on the right side of $y_{\min,\theta}$, giving the error tolerance in $S_{l_\theta}$ and $S_{r_\theta}$, respectively. To compute the optimal solution between two consecutive events we use the next lemma.

**Lemma 4** *Rotating counterclockwise the coordinate system between two consecutive events, the vertical distances given the error tolerances for the staircases of the first and third (second and fourth) quadrants always increase (decrease).*

*Proof.* Figure 5 illustrates all the quadrant cases. The proof is intuitive and clear. □

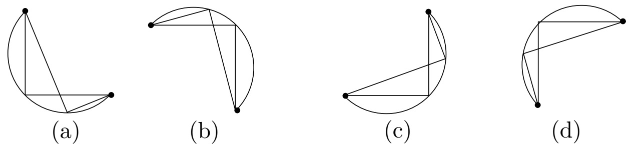

Figure 5: Variation of the vertical distance with the orientation: (a) first quadrant, (b) third quadrant, (c) second quadrant, (d) fourth quadrant.

As a consequence of Lemma 4, the optimal solution for an interval orientation $\theta \in [\theta_1, \theta_2]$ of consecutive events has to be found in the extremes of the interval, either at $\theta_1$ or at $\theta_2$. Thus, summarizing: (1) the number of events (insertions/deletions) for the staircase structure as $\theta$ changes from 0 to 90 is linear, (2) any update can be done in $O(\log n)$ time, (3) for a fixed value of $\theta$ doing binary search in $O(\log n)$ we can compute the optimal location of the line $\ell_\theta$, its corresponding error tolerance, and maintain the minimum one. We conclude that the un-oriented 2-fitting problem can be solved in $O(n \log n)$ time and $O(n)$ space.

**The un-oriented-2-fitting-algorithm.**

We assume that for the current orientation $\theta$ the point with $y$-coordinate $y_{\max,\theta}$ is on the left of the point with $y$-coordinate $y_{\min,\theta}$; otherwise, we only update the changes in the staircase structure without computing the optimal solution. We repeat the algorithm for the otherwise case.

1. Use the algorithm from Avis et al. [2] to compute in $O(n \log n)$ time the list of the un-oriented $\pi/2$-maxima of $S$ and their orientation intervals $\mathbb{S}^2$ where each point is un-oriented $\pi/2$-maxima. Sort the arrangement of the orientation intervals according to their endpoints in such a way that when we sweep the arrangement we know which $\pi/2$-maxima points are *active* in a current sweeping orientation, and which is the next incoming endpoint (Figure 4).
2. In $O(n \log n)$ time compute the horizontal/vertical staircase structure for $S$ and the optimal solution as we did in the oriented 2-fitting problem. The staircase structure is formed by the $\pi/2$-maxima points for $S$ with orientations 45, $90 + 45$, $180 + 45$, and $270 + 45$.
3. Sweep the arrangement of the orientation intervals with the four vertical lines. Each time that we get an endpoint, either (1) a new un-oriented $\pi/2$-maxima point enters the staircase structure, or (2) an active un-oriented $\pi/2$-maxima point is deleted from the staircases. We update the changes in the staircases in $O(\log n)$ time, including also the possible changes of the points with minimum and maximum $y$-coordinates for the current orientation. Since we consider the points in general position, at most two aligned points are updated at the same time producing a constant number of changes. We use binary search to compute the separating line of the new optimal solution in $O(\log n)$ time, and store and update the information of the optimal solution.

Notice that a point enters in one of the staircases at most once, so a point is updated a constant number times and the overall running time for updating changes is $O(n \log n)$ time. The running time of the algorithm is $O(n \log n)$ and the space is $O(n)$ since the lists and the arrangement of orientation intervals have linear size[1].

The $\Omega(n \log n)$-time lower bound for the un-oriented 1-fitting problem implies the same lower bound for the un-oriented 2-fitting problem. Nevertheless we show a reduction for the un-oriented 2-fitting problem to a MAX-GAP problem proving directly the $\Omega(n \log n)$-time lower bound.

We reduce our problem to the MAX-GAP problem for points in the first quadrant of the unit circle centered at the origin of the coordinates system. This problem has an $\Omega(n \log n)$ time lower bound in the algebraic decision tree model [17]. Let $P = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be an instance of the MAX-GAP problem. Consider a symmetric copy of points in the third quadrant and a new copy of the overall circle in other position as in Figure 6. It is easy to see that the optimal un-oriented 2-fitting orthogonal chain defines the maximum gap for $P$ and vice versa. This construction can be extended for the un-oriented $k$-fitting problem, $k \geq 3$ doing $k - 1$ copies of the initial circle such that the centers are located with an adequate distance between them.

**Theorem 4** *The un-oriented 2-fitting problem can be solved in $\Theta(n \log n)$ time and $O(n)$ space.*

[1]Notice that the algorithm can maintain the rectilinear convex hull of $S$ during the rotation in $O(n \log n)$ time and $O(n)$ space, this improves a very recent $O(n^2)$ time and space algorithm for this problem [4].

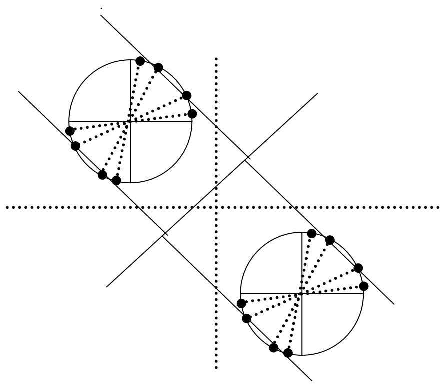

Figure 6: Construction in the proof of the lower bound for the un-oriented 2-fitting problem.

## 4 The oriented 2-fitting problem in 3D

In this section we consider the oriented 2-fitting problem in three-dimensions where a polygonal chain is interpreted as a configuration of three consecutive orthogonal planes.

The oriented 1-fitting problem for $S$ in $\mathbb{R}^3$ is equivalent to that of finding the width of $S$. To solve this problem we proceed according to how much information we have in order to fix the orientation: (1) Fixed the orientation of the solution plane, for example $(0,0,1) \in \mathcal{S}^3$, we solve this problem in $O(n)$ time by computing the points with maximum and minimum $z$-coordinates. (2) Fixed the orientation of a line contained in the solution plane (for example, $(0,1,0) \in \mathcal{S}^3$), we solve this problem by first projecting the points onto a plane orthogonal to the line (in our example, the plane $y = 0$), then computing the convex hull of the projected points in $O(n \log n)$ time, and, finally, computing the width of this hull with a rotating caliper in $O(n)$ time. The total running time is $\Theta(n \log n)$. The $\Omega(n \log n)$ time lower bound comes from the computation of the width for a set of points in two-dimensions.

The oriented 2-fitting problem is defined by three consecutive orthogonal planes. We refer to the plane $\Pi$ producing the bipartition of $S$ as the *separating plane* and to the two parallel planes that induce the error tolerances on either side of $\Pi$ as the *supporting planes*. We distinguish among three cases depending on how the orientation for the solution is constrained: (1) fixing the orientation of both the separating plane and the parallel supporting planes; (2) fixing the orientation of the separating plane; and (3) fixing the orientation of the parallel supporting planes. Next, we consider the three cases.

*Case 1: the orientation of both the separating plane and the parallel supporting planes are fixed.* Assume that the separating plane has normal $n_1$ (e.g., $(0,1,0)$) and that the (parallel) supporting planes have normal $n_2$ (e.g., $(0,0,1)$). We reduce the problems to 2D as follows. Let $n_3 = n_1 \times n_2$. We project the points in $S$ onto a plane with normal $n_3$ (e.g., $(1,0,0)$) and solve it optimally in $O(n)$ time using the algorithm of Section 2.

*Case 2: the orientation of the separating plane is fixed.* Assume that the separating plane has unit normal $n$ (e.g., $(0,1,0)$). In $O(n \log n)$ time, sort the points in $S$ along $n$, i.e., by $n \cdot p_i$ (e.g., by $y$-coordinate). According to this order, let $S_i = \{p_1, \ldots, p_i\}$ and $S_{n-i} = \{p_{i+1}, \ldots, p_n\}$ be the bipartition of $S$ given by the separating plane passing through $p_i$. In order to compute the four parallel supporting planes of $S_i$ and $S_{n-i}$ for determining which bipartition of $S$ gives the optimal solution, we project the points of $S_i$ and the points of $S_{n-i}$ onto two planes parallel to the separating plane. We work with the convex hulls of the projected points. Let $S'_i$ and $S'_{n-i}$ be the projected points of $S_i$ and $S_{n-i}$, and let $CH(S'_i)$ and $CH(S'_{n-i})$ be their respective convex hulls (Figure 7).

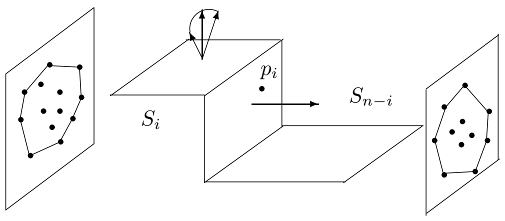

Figure 7: Sample configuration for Case (2).

To find the optimal 2-fitting solution, we use two clockwise rotating calipers which rotate simultaneously over $CH(S'_i)$ and $CH(S'_{n-i})$ in discrete steps. Each step is defined by the minimum rotating angle of the two calipers on antipodal pairs. Suppose that at some step, the rotating caliper over $CH(S'_i)$ ($CH(S'_{n-i})$) has antipodal points $q_1$ and $q_2$ ($q_3$ and $q_4$). Let $\alpha$ ($\alpha'$) be the angle of rotation with respect to the parallel supporting lines passing through $q_1$ and $q_2$ ($q_3$ and $q_4$). Let $w_1$ ($w_2$) be the width function of $CH(S'_i)$ ($CH(S'_{n-i})$) in the rotation interval and $d_1 = d(q_1, q_2)$ ($d_2 = d(q_3, q_4)$). The continuous and monotone width function $w_1$ ($w_2$) depends on $d_1$ ($d_2$) and $\cos(\alpha)$ ($\cos(\alpha')$). The minimum of the maximum of the two width values is a minimum of the upper envelope of the two functions $w_1$ and $w_2$. Thus, we compute the widths when this fact occurs (a linear number of times) and maintain the best solution.

We can update the convex hulls $CH(S'_i)$ and $CH(S'_{n-i})$ in $O(\log n)$ time when a point $p_i$ changes from $S_{n-i}$ to $S_i$ [3]. We use linear time for the task of computing an optimal solution for each partition with a total $O(n^2)$ running time. We do not know how to compute the new minimum of the maximum of the two width values in $O(\log n)$ time. Consequently, we propose the following open problem.

**Open problem 2** *$CH(S'_i)$ and $CH(S'_{n-i})$ can be updated in $O(\log n)$ time. Can the new minimum of the maxima of the two width values be computed in $O(\log n)$ time?*

*Case 3: the orientation of the parallel supporting planes is fixed.* Without loss of generality, assume that the parallel supporting planes have normal vector $n = (0, 0, 1)$. A optimal separating plane passing through a point $p_i \in S$, produces a bipartition $S_i$, $S_{n-i}$ of $S$ that separates the points with extreme projections along $n$, i.e., with minimum and maximum $z$-coordinates $z_{\min}$ and $z_{\max}$. In $O(n \log n)$ time, sort the points of $S$ by decreasing $z$-coordinate and store them in a list $L_Z$.

Let $\Pi_i$ be the separating plane containing the vertical line $v_i$ passing through $p_i$. Rotate $\Pi_i$ 180 degrees over $v_i$, starting with the $x$-axis orientation. We compute the points located in both half-spaces of $\Pi_i$ in $O(n)$ time, computing also the points $z^l_{\min}$ and $z^r_{\max}$ on each left and right half-space of $\Pi_i$ (Figure 8). Project the points of $S$ on the plane $z = 0$ and compute the dual formed by the arrangement of the lines corresponding to the projected points. The radial order of the projected points with respect to one of them can be computed in the dual in $O(n)$ time. With this order we can compute in $O(n)$ time the sequence of the points in $S \setminus \{p_i\}$ bumped by rotating $\Pi_i$ over $v_i$.

We do a radial sweep with $\Pi_i$ anchored at $v_i$ creating a list $L_r$ ($L_l$) of the current values of $z^r_{\max}$ ($z^l_{\min}$) on the right (left) half-space of $\Pi_i$. To do this, each time $\Pi_i$ bumps a new point $p_j \in S$, we do the following tasks in constant time: (i) check that the points $z_{\max}$ and $z_{\min}$ are not in the same half-space, otherwise mark this step as a *null step*, and (ii) compare the $z$-coordinate of $p_j$ with the current values $z^r_{\max}$ and $z^l_{\min}$.

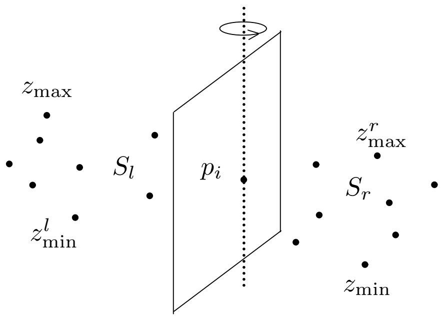

Figure 8: The plane $\Pi_i$ rotates over the vertical line passing through a point $p_i$.

The lists $L_r$ and $L_l$ have the same size, say $m$. From $L_r$ and $L_l$, in $O(n)$ time, create a list $L$ formed by the minimum of the two values sharing the same position $k$, $1 \leq k \leq m$, if neither value is labeled as *null step*. Finally, in $O(n)$ time, compute the minimum value on the list $L$ and its position. Thus, we have outlined an $O(n^2)$ time and space algorithm, whose complexity is justified by the fact that we have spent $O(n)$ time per point $p_i$, and used the dual to compute the radial orders.

**Theorem 5** *The oriented* 2*-fitting problem in 3D can be solved: in* $\Theta(n)$ *time and space if the orientations of both the separating plane and the parallel supported planes are fixed; in* $O(n^2)$ *time and* $O(n)$ *space if the orientation of the separating plane is fixed; and in* $O(n^2)$ *time and* $O(n^2)$ *space if the orientation of the parallel supporting planes is fixed.*

*Acknowledgement.* These problems were proposed and partially solved during the *Third Spanish Workshop on Geometric Optimization*, July 4-8, 2006, El Rocío, Huelva, Spain. The authors would like to thank the other workshop participants for helpful comments.

## References

[1] B. Aronov, T. Asano, N. Katoh, K. Melhorn, and T. Tokuyama. Polyline fitting of planar points under min-sum criteria. *International Journal of Computational Geometry and Aplications*, Vol. 16, Nos. 2 and 3, 2006, pp. 97–116.

[2] D. Avis, B. Beresford-Smith, L. Devroye, H. Elgindy, E. Guvremont, F. Hurtado, and B. Zhu. Unoriented Θ-maxima in the plane: complexity and algorithms. *SIAM Journal of Computing*, Vol. 28, No. 1, 1999, pp. 278–296.

[3] D. Avis, H. Elgindy, and R. Seidel. Simple on-line algorithms for convex polygons. *Computational Geometry*, G. T. Toussaint, ed., North-Holland, Amsterdam, 1985, pp. 23–42.

[4] S. W. Bae, C. Lee, H-K. Ahn, S. Choi, and K-Y. Chwa. Computing minimum-area rectilinear convex hull and L-shape. *Computational Geometry: Theory and Applications*, Vol. 42, 2009, pp. 903–912.

[5] P. J. Burt. Fast filter transforms for image processing. *Computer Graphics and Image Processing*, Vol. 16, 1979, pp. 20–51.

[6] W. S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimun error. *International Journal of Computational Geometry and Applications*, Vol. 6, 1996, pp. 59–77.

[7] D. Z. Chen and H. Wang. Approximating points by piecewise linear functions. *Manuscript.*

[8] F. Chin, A. Choi, and Y. Luo. Optimal generating kernel for image pyramids by piecewise fitting. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 14, 1992, pp. 1190–1198.

[9] J. M. Díaz-Báñez and J. A. Mesa. Fitting rectilinear polygonal curves to a set of points in the plane. *European Journal of Oper. Research*, 130, 1, 2001, pp. 214–222.

[10] H. Fournier and A. Vigneron. Fitting a step function to a point set. *LNCS*, 5193, 2008, pp. 442–453.

[11] M. T. Goodrich. Efficient piecewise-linear function approximation using the uniform metric. *Discrete and Computational Geometry*, Vol. 14, 1995, pp. 445–462.

[12] L. J. Guibas, J. E. Hershberger, J. S. B. Mitchell, and J. S. Snoeyink. Approximating polygons and subdivisions with minimum-link paths. *International Journal of Computational Geometry and Applications*, Vol. 3, 1993, pp. 383–415.

[13] S. L. Hakimi and E. F. Schmeichel. Fitting polygonal functions to a set of points in the plane. *Graphical Models and Image Processing*, Vol. 53, 1991, pp. 132–136.

[14] M. E. Houle and G. T. Toussaint. Computing the width of a set. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 5, 1988, pp. 761–765.

[15] H. Imai and M. Iri. Polygonal approximations of a curve-formulations and algorithms. *Computational Morphology*, G. T. Toussaint ed., North Holland, 1988.

[16] H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *Journal of ACM*, Vol. 22, 1975, pp. 469–476.

[17] D. T. Lee and Y. F. Wu. Geometric complexity of some location problems. *Algorithmica*, Vol. 1, 1986, pp. 193–211.

[18] M. A. Lopez and Y. Mayster. Approximating a set of points by a step function. *Journal of Visual Communication and Image Representation*, 2006, pp. 1178-1189.

[19] T. Pavlidis. Algorithms for shape analysis of contours and waveforms. *IEEE Transation Pattern Analysis Machine Intelligence*, 1980, pp. 301–312.

[20] F. P. Preparata and M. I. Shamos. *Computacional Geometry, an Introduction.* Springer, 1988.

[21] G. T. Toussaint. On the complexity of approximating polygonal curves in the plane. *Proc. International Symposium on Robotics and Automation*, Lugano, Switzerland, 1985.

[22] K. R. Varadarajan. Approximating monotone polygonal curves using the uniform metric. *Proc. 12th Annual ACM Symp. on Computational Geometry*, 1996, pp. 311-318.

[23] D. P. Wang. A new algorithm for fitting a rectilinear $x$-monotone curve to a set of points in the plane. *Pattern Recognition Letters*, Vol. 23, No. 1, 2002, pp. 329–334.

[24] D. P. Wang, N. F. Huang, H. S. Chao, and R. C. T. Lee. Plane sweep algorithms for polygonal approximation problems with applications. *LNCS*, 762, 1993, pp. 515–522.