How to Cover a Point Set with a V-Shape of Minimum Width*

Boris Aronov and Muriel Dulieu

Department of Computer Science and Engineering, Polytechnic Institute of NYU, Brooklyn, NY 11201-3840, USA; aronov@poly.edu, mdulieu@gmail.com

Abstract. A balanced V-shape is a polygonal region in the plane contained in the union of two crossing equal-width strips. It is delimited by two pairs of parallel rays that emanate from two points \(x, y\), are contained in the strip boundaries, and are mirror-symmetric with respect to the line \(xy\). The width of a balanced V-shape is the width of the strips. We first present an \(O(n^2 \log n)\) time algorithm to compute, given a set of \(n\) points \(P\), a minimum-width balanced V-shape covering \(P\). We then describe a PTAS for computing a \((1 + \varepsilon)\)-approximation of this V-shape in time \(O((n/\varepsilon) \log n + (n/\varepsilon^{3/2}) \log^2 (1/\varepsilon))\). A much simpler constant-factor approximation algorithm is also described.

1 Introduction

Motivation. The problem we consider in this paper was motivated by the following curve reconstruction question: One is given a set of points sampled from a curve in the plane. The sample is noisy in the sense that the points lie near the curve, but not exactly on it. One would like to reconstruct the original curve from this data. Clearly one has to make some assumptions about the point set and the curve: If the curve is “too wiggly” or the noise is too large, little can be done. One approach is to assume that the curve is smooth and the sample points lie not too far from it; see [12, 13] and references therein. Roughly speaking, one can then approximate a stretch of a curve by an elongated rectangle (or strip) whose width is determined both by the curvature of the curve and the amount of noise. Refining this approximation allows one to reconstruct the location of the curve and its normal vector.

Complications arise when a curve makes a sharp turn, as it does not have a well-defined direction near the point of turn. It has been suggested [12, 18]...
that one approach to handle this situation is to replace fitting the set of points corresponding to a smooth arc of a curve with a strip by fitting with a wedge-like shape that we call a "balanced V-shape;" perhaps one might incorporate it in an algorithm such as that of [16]. It is meant to model one thickened turn in a piecewise-linear curve; refer to the figure and precise definitions below.

In this paper, we construct a slower exact algorithm for identifying a V-shape that best fits a given set of points in the plane, then a faster constant-factor approximation algorithm, and finally a considerably more involved algorithm that produces a $(1 + \varepsilon)$-approximation, for any positive $\varepsilon$.

The problem we solve is a new representative of a widely studied class of problems, namely geometric optimization or fitting questions; see [1, 3–5] and references therein. Generally, the problem is to find a shape from a given class that best fits a given set of points. Classical examples of such problems are linear regression in statistics, the computation of the width of a point set (which constructs a minimum-width strip covering the set), computing a minimum enclosing ball, cylinder, or ellipsoid, a minimum-width spherical or cylindrical shell, or a small number of strips of minimum width, covering the point set; see [1,10].

Previous work most closely related to our problem is that of Glozman, Kedem, and Shpitalnik [17]. They compute a double-ray center for a planar point set $S$. A double-ray center is a pair of rays emanating from a common apex, minimizing the Hausdorff distance between $S$ and the double ray. While the shape they consider is not exactly a V-shape, it is similar enough to be used for the same purpose. The exact algorithm they present runs in $O(n^3 \alpha(n) \log^2 n)$ time, however, in contrast to our near-quadratic-time algorithm; here $\alpha(n)$ is the inverse Ackermann function.

Another paper closely related to our problem is that of Agarwal, Procopiuc, and Varadarajan [2]. It concerns the 2-line-center problem studied extensively in the past; see the references in [2]. The goal is to cover a given set of points by two strips of minimum possible width. One application is fitting two lines to a point set. There had been several previously known near-quadratic-time exact algorithms for the problem. An $O(n \log n)$-time 6-approximation algorithm, and an $O(n \log n + n \varepsilon^{-2} \log(1/\varepsilon) + \varepsilon^{-7/2} \log(1/\varepsilon))$-time $(1 + \varepsilon)$-approximation algorithm were presented in [2]. A V-shape covering a point set is a special case of covering a point set by two strips, so some of the tools from [2] apply to our problem as well.

**Problem statement and results.** In this paper, we focus on the class of polygonal regions in the plane that we call balanced V-shapes. A balanced V-shape has two vertices $x$ and $y$ and is delimited by two pairs of parallel rays. One pair of parallel rays emanate from $x$ and $y$ on one side of the line $xy$ and the other pair of rays emanate from $x$ and $y$ on the other side.
of \(xy\), symmetrically with respect to \(xy\) (see the above figure). In particular, a balanced V-shape is completely contained in the union of two crossing strips of equal width. Its width is the width of the strips.

Consider a point set \(P\) of \(n\) points in the plane. We describe, in Section 3, an \(O(n^2 \log n)\) time algorithm that computes a balanced V-shape with minimum width covering \(P\).

Our algorithm actually identifies a particular type of V-shapes that we call “canonical” (see below for definitions) and enumerates all minimum-width canonical V-shapes covering \(P\); as some degenerate \(n\)-point sets have \(\Theta(n^2)\) such V-shapes (see Section 4), this approach will probably not yield a subquadratic algorithm. This leaves open the problem of how quickly one can identify just one minimum-width V-shape covering \(P\).

In Section 5, we present an \(O(n \log n)\) algorithm that constructs a V-shape covering \(P\) with width at most 13 times the minimum possible width. In Section 6, we show how to construct a \((1 + \varepsilon)\)-approximation in time \(O((n/\varepsilon) \log n + (n/\varepsilon^{3/2}) \log^2(1/\varepsilon))\), starting with the 13-approximation obtained earlier.

2 Reduction to canonical V-shapes

In the remainder of this paper, for simplicity of presentation and unless noted otherwise, we assume that the points of \(P\) are in general position: no three points are collinear and no two pairs of points define parallel lines. All algorithms can be adapted to degenerate inputs without asymptotic slowdown.

We will find it convenient to consider a larger class of objects, namely V-shapes. A (not necessarily balanced) V-shape (refer to the figure below) is a polygonal region similar to a balanced V-shape except that the widths of its two arms need not be the same. More formally, a V-shape \(V\) is a polygonal region bounded by two pairs of parallel rays emanating from its two vertices \(x\) and \(y\). One pair of parallel rays (left rays \(X_1\) and \(Y_1\)) lies on the left side of the directed line \(xy\), while the other pair (right rays \(X_2\) and \(Y_2\)) lies on its right side. The inner rays \(X_i\) emanate from \(x\), while outer rays \(Y_i\) emanate from \(y\). \(X_1 \cup X_2\) is the inner boundary of \(V\), while \(Y_1 \cup Y_2\) is its outer boundary. The left arm of \(V\), \(L = L(V)\), is its portion on the left of \(xy\); i.e., it is the region bounded by rays \(X_1\) and \(Y_1\) and segment \(xy\). The width of the left arm, width(\(L(V)\)), is the distance between \(X_1\) and \(Y_1\). The right arm and its width are defined analogously. The width of \(V\), width(\(V)\), is the larger of the widths of its two arms. \(V\) is contained in the union of two strips \(S_1\) and \(S_2\): \(S_i\) is delimited by the lines containing \(X_i\) and \(Y_i\), respectively; we refer to \(S_1\) and \(S_2\) as the left and right strip of \(V\), respectively.

A minimum-width balanced V-shape can be obtained from a minimum-width V-shape by widening the narrower arm until the widths of the arms are equal.
In the remainder of the paper, the \( n \)-point set \( P \) is fixed. To avoid trivial cases, we assume that \( n \geq 5 \). By the general position assumption, all points of \( P \) cannot be collinear, nor can \( P \) be covered by a V-shape of zero width. We need not consider V-shapes with all points in one strip as according to Lemma 1, such a V-shape does not have minimum width.

**Lemma 1.** In a positive-width minimum-width V-shape \( V \) covering \( P \), it is not possible that one of the strips already contains \( P \) in its entirety.

**Proof.** Suppose \( S_1 := S_1(V) \) covers \( P \). Let \( w > 0 \) be its width. We argue that there is a V-shape covering \( P \) of width \( w/2 + \varepsilon \), for any positive \( \varepsilon \), so \( V \) does not have minimum width. Indeed, let \( \ell \) be the median line of \( S_1 \). Cut \( S_1 \) by \( \ell \) into two parallel strips of width \( w/2 \). They cover \( P \). They do not form a V-shape, but they can be approximated arbitrarily closely by a V-shape near \( P \), by placing its vertex \( y \) sufficiently far to the left of \( P \) along \( \ell \), \( x \) at the rightmost point of \( \text{CH}(P) \cap \ell \), and the boundary rays near-parallel to \( \ell \). \( \square \)

Unless otherwise stated, the only particular V-shapes we will be interested in are the ones we call canonical. A V-shape is canonical, if the bounding rays of each arm pass through exactly three points of \( P \); more precisely if \( |X_i \cap P| + |Y_i \cap P| \geq 3 \), for \( i = 1, 2 \) (recall that, by our general position assumption, \( |X_i \cap P|, |Y_i \cap P| \leq 2 \)); in addition, we require that each arm of a canonical V-shape covering \( P \) is locally of minimum width, i.e., neither arm can be narrowed by an infinitesimal motion.

The reason why we consider only canonical V-shapes is that at least one minimum-width V-shape covering \( P \) is canonical (see Lemma 2 below), so we can confine the search to canonical V-shapes and discard any non-canonical V-shapes considered by our algorithm.

**Lemma 2.** At least one minimum-width V-shape covering \( P \) is canonical.

**Proof.** In order to prove this lemma, we first explain why we can assume that \( |X_i \cap P| + |Y_i \cap P| \geq 3 \). We then discuss how the boundary points may be positioned on the arms.

By Lemma 1, in no minimum-width covering V-shape one strip covers \( P \) completely. Hence in the following we disregard this possibility.

We present a sequence of transformations, starting with a minimum-width V-shape covering \( P \), which do not increase its width and end in a canonical V-shape. We begin by translating its outer boundary in the direction of first \( Y_1 \) and then \( Y_2 \) to ensure that each of the outer rays contains a point of \( P \); this point might be \( y \). Now translating the inner boundary, first in the direction opposite to that of \( X_1 \), and then that of \( X_2 \), we guarantee that each of \( X_1, X_2 \) contains a point of \( P \). (By Lemma 1, an outer ray \( Y_i \) cannot meet its corresponding inner ray \( X_i \) without meeting a point of \( P \), as we started with a minimum-width V-shape.) Now consider an arm (say, left) of the resulting V-shape. We will further transform it so that \( |X_1 \cap P| + |Y_1 \cap P| > 2 \). We have so far ensured that each of \( X_1 \) and \( Y_1 \) contains at least one point. If exactly one point is present on each left ray, \( X_1 \) and \( Y_1 \) can be rotated so that the width of \( S_1 \) shrinks. This
process stops either when $S_1$ collapses to a line (in which case it is easy to check that $Y_1$ contains two points of $P$ and $X_1$ contains at least one) or when three points $s_1, s_2, s_3$ lie in $X_1 \cup Y_1$, say $s_1$ and $s_2$ on one ray and $s_3$ on the other. In the latter case, unless the angles $\angle s_3 s_1 s_2$ and $\angle s_1 s_2 s_3$ are acute, the rotation can proceed, further narrowing $S_1$. Corollary 1 follows from this last condition.

Repeating the process with the right arm, we arrive at a covering V-shape whose width is no larger than that of the original V-shape, with the property that (a) it satisfies Corollary 1 if there is no zero-width arm, (b) each bounding ray contains a point of $P$, and (c) $|X_1 \cap P| + |Y_1 \cap P| \geq 3$ and $|X_2 \cap P| + |Y_2 \cap P| \geq 3$; i.e., the resulting covering V-shape is canonical and as good or better in terms of width. Hence, indeed, it is sufficient to examine only canonical V-shapes. □

Corollary 1. Let $P$ be a point set and $V$ be a minimum-width canonical V-shape covering it. Let $s_1, s_2, s_3 \in X$ be the points on the boundary of a non-zero-width arm of $V$, with $s_1$ and $s_2$ on one ray and $s_3$ on the other. Then the angles $\angle s_3 s_1 s_2$ and $\angle s_1 s_2 s_3$ are acute.

All canonical minimum-width V-shapes fall into the following three categories:

**both-outer** Each outer ray contains exactly two points of $P$, and each inner ray contains at least one.

**inner-outer** On one arm of the V-shape, the inner ray contains exactly two points of $P$; on the other arm, the outer ray contains exactly two points. The other rays contain at least one point of $P$.

**both-inner** Each inner ray contains exactly two points of $P$ and each outer ray contains at least one.

3 Computing a canonical minimum-width V-shape

To find a canonical minimum-width V-shape covering $P$, we will search independently for the best solution for each of the three types identified above and output the V-shape that minimizes the width. Let $H$ be the convex hull of $P$.

**V-shapes of both-outer type.** Consider a covering V-shape $V$ with outer rays $Y_1, Y_2$ containing edges $e_1, e_2$ of $H$, respectively; refer to the figure on the right. Let $\ell$ be the bisector of the angle $\angle Y_1 Y_2$. Notice that $V$ is not minimal unless its width is given by the largest distance from a point in $P$ to its closest outer ray. Therefore, we can assume that points of $P$ left of $\ell$ belong to the left arm of $V$ and points right of $\ell$—to its right arm.

Thus, given $Y_1, Y_2$, and $\ell$ it is sufficient to determine the furthest point from $Y_1$ to the left of $\ell$ and the furthest point from $Y_2$ to the right of $\ell$. The larger distance determines the width of $V$. This can be accomplished by building a data
structure $D(P)$ on $P$ that supports the following queries: Given a halfplane $h$ and a direction $d$, return an extreme point of $P \cap h$ in direction $d$. $O(n^2)$ queries are sufficient to enumerate all choices of $e_1, e_2$ and identify the best both-outer-type V-shape. $D(P)$ can be constructed in $O(n^2 \log n)$ time and supports logarithmic-time queries, resulting in total running time of $O(n^2 \log n)$.

$D(P)$ is constructed as follows: We build the arrangement $A = A(P^*)$ of lines dual to points of $P$. Cells of $A$ correspond to different ways to partition $P$ by a line. We construct a directed spanning tree $T$ of the cells of $A$, starting with the bottommost cell and allowing only arcs from a cell $f$ to a cell immediately above $f$ and sharing an edge with it; we use $P_f \subset P$ to denote the convex hull of the set of points whose dual lines lie below $f$. Using $T$ as the history tree, we store the convex hull $P_f$ for every face $f \in A$, in a fully persistent version [14] of the semi-dynamic convex hull data structure of [21]. We also preprocess $A$ for point location. Given a query (say, upper) half-plane $h$ and direction $d$, we locate the face $f$ of $A$ containing the point dual to the bounding line of $h$ and consult the data structure associated with $f$ and storing $P_f = P \cap h$ to find the extreme point of $P_f$ in direction $d$, all in logarithmic time.

**V-shapes of inner-outer type.** In this section, we describe how to find a minimum-width canonical V-shape covering $P$ and having exactly one edge of $\text{CH}(P)$, say $e$, on its outer boundary; it contains two points of $P$ on the inner bounding ray of its other arm. We handle each choice of $e$ independently, in $O(n \log n)$ time, yielding overall $O(n^2 \log n)$ running time.

Having fixed an edge $e$ of $\text{CH}(P)$, consider a (minimum-width canonical) V-shape $V$ covering $P$ that has $e$ on its boundary. For ease of description, suppose $Y_1 \ni e$, $X_2$ contains two points $p_1, p_2 \in P$, while both $Y_2$ and $X_1$ contain at least one point of $P$ each, denoted $p$ and $q$, respectively; see the figure on the right.

Let $\ell$ be the line containing $e$, and $\ell'$ be the line containing $e' := p_1 p_2$. Set $Q := S_1 \cap P$ and $Q' := P \setminus Q$. We observe that

a) $Q$ is the set of points of $P$ at distance at most $\text{dist}(q, \ell)$ from $\ell$;

b) $p_1 p_2$ is an edge of $\text{CH}(Q')$;

c) $Y_2$ is contained in a supporting line $\ell_{p_1 p_2}$ of $\text{CH}(Q')$ (which must also be a supporting line to $\text{CH}(P)$ for $V$ to cover $P$) parallel to $\ell'$; this line lies on the same side of $\ell'$ as $Q'$;

d) $\text{width}(V) = \max(\text{width}(S_1), \text{width}(S_2)) = \max(\text{dist}(q, \ell), \text{dist}(\ell', \ell_{p_1 p_2}))$.

Our algorithm enumerates all choices for the point $q$, in order of decreasing distance from $\ell$. For the current choice of $q$, it maintains (the boundary of)

\[2\text{ If width}(R(V)) = 0, we have } Q' \subset \ell' \text{ and } Y_2 \subset \ell'.\]
CH(Q'), say as an AVL tree, and, for each edge e' of CH(Q'), the distance from ℓ' to the furthest point of CH(P) to the right of (i.e., “beyond”) ℓ'. Edges with distances are stored in a min-heap; the minimum such distance gives the minimum width for S_2 for the current choice of S_1; the larger of the two widths determines the width of the current V-shape. We record the best width of any V-shape encountered in the process.

The algorithm is initialized with the set Q' containing the two points of P furthest from ℓ (the case where Q' contains only one point treated by the both-outer case as the zero-width strip S_2 can be rotated until it contains one edge of CH(P)). A generic step of the algorithm involves moving the current point q from Q to Q'. We update the convex hull of Q' by computing the supporting tangents from q to the old hull, in O(log n) time. For the two new hull edges e_1, e_2, we compute the corresponding supporting lines ℓ_{e_1}, ℓ_{e_2} of CH(P), also in logarithmic time. We add the new edges with the corresponding widths to the min-heap, after removing from it the entries of all the eliminated edges of CH(Q'). The root of the min-heap yields the best width for S_2 for the current partition {Q, Q'}. The algorithm requires presorting points by distance from ℓ and then a linear number of balanced-search-tree and heap operations (since the number of edges inserted is less than 2n and each cannot be deleted more than once), for a total running time of O(n log n) for a fixed e, as claimed.

Working through the entire set P (except for the endpoints of e), in order of decreasing distance from ℓ, growing Q' and shrinking Q, we obtain a sequence of fewer than n V-shapes which include all the canonical minimum-width V-shapes covering P with e on its outer boundary and two other points of P lying on the opposite arm’s inner boundary (it may include non-canonical V-shapes as well, but it is not difficult to check that every combination (e, q, e', ℓ_{e'}) examined by the algorithm yields a valid V-shape covering P, which is sufficient for our purposes).

To summarize, inner-outer type V-shapes can be handled in total time O(n^2 log n).

V-shapes of both-inner type. Now a covering V-shape V has points a, b of P on its inner ray X_1 and points c, d on its inner ray X_2; refer to Figure 1; points a, b, c, d are in convex position, in this counterclockwise order. It is known [20]
that there are at most \( O(n^2) \) such wedges \( W = W(a, b, c, d) \) determined by a quadruple of points \( a, b, c, d \in P \) and empty of points of \( P \); note that \( W \) completely determines \( V \), and, given \( W \), one can construct the corresponding \( V \) in \( O(\log n) \) time, so it is sufficient to enumerate all empty wedges \( W \).

For a pair \( a, b \in P \), we compute all pairs \( c, d \), so that \( W(a, b, c, d) \) is an empty wedge. Let \( Q(a, b) \) be the set of all points of \( P \) lying to the left of the directed line \( ab \).

**Observation 1.** \( W(a, b, c, d) \), in the above notation, is an empty wedge if and only if line \( cd \) supports \( CH(Q) \) and separates segment \( ab \) from \( Q = Q(a, b) \) (and \( a, b, c, d \) are in this counterclockwise order).

Now enumerating all \( k \) pairs \( c, d \) for a fixed choice of \( a, b \) can be done in time \( O((k + 1) \log n) \), as follows. While handling V-shapes of both-outer type we constructed a data structure \( D(P) \) which, for a given line (here \( ab \)), produces a balanced search tree storing the convex hull of the points of \( P \) lying to one side of the line (here \( Q = Q(a, b) \)). Using \( D(P) \), we find the point \( z \) of \( Q \) closest to the line \( ab \) and traverse the boundary of \( CH(Q) \) in both directions from \( z \), to list all \( k \) edges \( cd \) of \( CH(Q) \) satisfying the conditions of the above observation. Since all such edges are consecutive, it is sufficient to examine \( k + 2 \) edges of \( CH(Q) \). Repeating the procedure for all choices of \( a, b \) and recalling that the number of empty wedges is at most quadratic, we deduce that the enumeration algorithm runs in time \( O(n^2 \log n) \).

### 4 Maximum number of canonical minimum-width V-shapes

How far is our algorithm from optimality? In Figure 2, starting with the vertex set of two congruent regular \( k \)-gons, for a suitably large \( k \), we sketch a construction of a set of \( n \) points with \( \Theta(n^2) \) distinct covering minimum-width V-shapes. The idea is that a minimum-width covering V-shape would consist essentially of two independently chosen minimum-width strips, each covering one \( k \)-gon. The point set is highly degenerate. However, perturbing it slightly yields a point set with \( \Theta(n^2) \) canonical V-shapes with width arbitrarily close to minimum possible. This is an indication that any algorithm that explicitly enumerates canonical covering V-shapes may have to spend \( \Omega(n^2) \) time on this input, thus it is unlikely that any algorithm taking our approach can run much faster. On the other hand, for this specific input one can encode the \( \Theta(n^2) \) optimal V-shapes in \( \Theta(n) \) space. This leaves open the possibility that identifying a single minimum-width covering V-shape may still be possible in subquadratic worst-case time.

### 5 A 13-approximation algorithm

Given a set of points \( P \), let \( w \) be the minimum value such that \( P \) can be covered by a V-shape of width \( w \). We present an algorithm that computes a V-shape
Fig. 2: A sketch of a construction of a set with many minimum-width covering V-shapes. Points below and above the two regular \( k \)-gons are added to raise the minimum width of any covering V-shape to that of the \( k \)-gons.

covering \( P \) of width at most \( 13w \) in time \( O(n \log n) \). For this purpose, we use the \( O(n \log n) \) time 6-approximation algorithm for the 2-line-center problem presented by Agarwal, Procopiuc, and Varadarajan [2]. Recall that the 2-line-center problem is the following: Given a set \( P \) of \( n \) points in \( \mathbb{R}^2 \), cover \( P \) by two congruent strips of minimum width. We start with the following observation which follows from the fact that the union of the two strips of any V-shape covering \( P \) contains \( P \).

**Observation 2.** If \( w' \) is the width of two congruent strips of minimum width covering \( P \), \( w' \leq w \).

Our 13-approximation algorithm proceeds as follows. Use the 6-approximation algorithm of [2] to compute two congruent strips of width \( w'' \) that cover \( P \), with \( w' \leq w'' \leq 6w' \). (It is possible that the two strips just computed are such that a V-shape defined by them contains \( P \). In this case we return that V-shape. This clearly produces a 6-approximation, due to Observation 2. In the remainder of this section, we will assume that this is not the case, in other words, one of the two strips has points of \( P \) on both sides of it.) Find the median lines \( \ell_1 \) and \( \ell_2 \) of the strips. For all points in each strip, project them orthogonally onto \( \ell_1 \) and \( \ell_2 \) respectively (the points in the intersection of the strips are duplicated and projected onto both \( \ell_1 \) and \( \ell_2 \)). Let \( P' \) be the resulting set of projected points. Compute an exact minimum width V-shape \( V' \) covering \( P' \) in \( O(n \log n) \) time. The desired approximate V-shape \( V \) is obtained by widening \( V' \) by \( w''/2 \) in all directions.

**Theorem 3.** This algorithm computes a 13-approximation of a minimum-width V-shape covering \( P \).

**Proof.** Let \( V_{\text{best}} \) be a minimum-width covering V-shape of \( P \), \( V' \) — a minimum-width covering V-shape of \( P' \), and \( V_{\text{apx}} \) — the approximate covering V-shape.
computed by the algorithm. As the points of \( P \) have been moved by a distance of at most \( w''/2 \) to form \( P' \), width(\( V' \)) \( \leq \) width(\( V_{\text{best}} \)) + \( w'' \). Since \( V_{\text{apx}} \) is a widened version of \( V' \), it contains the points of \( P \). Moreover, width(\( V_{\text{apx}} \)) \( \leq \) width(\( V' \)) + \( w'' \) \( \leq \) width(\( V_{\text{best}} \)) + 2\( w'' \) \( \leq \) width(\( V_{\text{best}} \)) + 12\( w' \) \( \leq \) 13\( w' \) by Observation 2. □

**Remark.** Using the \((1 + \varepsilon)\)-approximation algorithm of [2] in place of their 6-approximation algorithm in our procedure, we can attain any approximation factor larger than three for the minimum-width V-shape. The running time remains \( O(n \log n) \), with the constant of proportionality depending on the quality of the approximation. We do not discuss this extension further, since we present our own \((1 + \varepsilon)\)-approximation algorithm for the problem in Section 6.

### 5.1 Minimum-width V-shape for points on two lines

We now describe how to compute the minimum-width V-shape \( V' \) covering a given point set \( P' \) contained in the union of two lines \( \ell_1, \ell_2 \) in the plane. Put \( z := \ell_1 \cap \ell_2 \).

Let \( P'_r := P' \cap \ell_1 \) and \( P'_l := P' \cap \ell_2 \). If \( P' \) is covered by a zero-width V-shape, which is easy to check, we are done. From now on we assume that this is not the case, i.e., that \( \ell_1, \ell_2 \) do not already form a V-shape containing \( P' \), so \( \ell_1 \) separates some two points of \( P'_2 \) and/or \( \ell_2 \) separates some two points of \( P'_1 \). The convex hull \( CH(P') \) has three or four vertices. Moreover, by reasoning similar to that of Section 3, the outer boundary of \( V' \) contains two, three, or four vertices of \( CH(P') \) (in the case where an outer ray is contained in \( \ell_1 \) or \( \ell_2 \), we consider only the extreme points). Before describing how we handle these cases, we need a technical lemma.

**Lemma 3.** Given a line partitioning \( P' \) into \( P'_r, P'_l \) and given their convex hulls \( CH(P'_r), CH(P'_l) \), the minimum-width canonical V-shape \( V' \) of \( P' \) containing \( P'_r \) in one strip and \( P'_l \) in the other can be computed in constant time; some points of \( P' \) might lie in both strips of \( V' \).

**Proof.** The convex hulls \( CH(P'_r) \) and \( CH(P'_l) \) have at most four vertices each. It must be the case that the boundary of one arm of \( V' \) contains an edge \( e_r \) of \( CH(P'_r) \) or an outer common tangent of \( CH(P'_r) \) and \( CH(P'_l) \), and the other arm boundary contains an edge \( e_l \) of \( CH(P'_l) \) or an outer common tangent to \( CH(P'_r) \) and \( CH(P'_l) \). There is a constant number of possible pairs of such edges. Let \( S(X, e) \) be the minimum-width strip covering a set \( X \) and parallel to \( e \). For each such pair of edges \( e_r, e_l \), check whether \( S(P'_r, e_r) \) and \( S(P'_l, e_l) \) form a V-shape. Return the canonical V-shape of minimum width among all V-shapes so generated. □

Now we consider the different types of canonical V-shapes covering \( P' \) and describe how to find a minimum-width V-shape of each type.

**Case 1:** An outer bounding ray of \( V' \) contains an edge \( e \) of \( CH(P') \). Let \( \ell \) be the line containing \( e \). For all points \( p \) of \( P' \), draw a line \( \ell_p \) through \( p \) and parallel
to \( \ell \). Apply Lemma 3 to (the partition induced by) \( \ell_p \). This can be implemented to run in overall time \( O(n \log n) \).

In the remaining cases, each of the outer rays of \( V' \) contains precisely one vertex of \( \text{CH}(P') \) and each inner ray contains two points of \( P' \).

Case 2: An inner ray of \( V' \) lies on \( \ell_1 \) or \( \ell_2 \). Suppose an inner ray of \( V' \) is contained in \( \ell_1 \). Draw two lines parallel to \( \ell_1 \) and very close to it, one to the left of \( \ell_1 \), one to the right of \( \ell_1 \). Apply Lemma 3 to each of these two lines.

Case 3: Point \( z = \ell_1 \cap \ell_2 \) lies between the two arms of \( V' \). Draw the two lines passing through \( z \) and bisecting the angles between \( \ell_1 \) and \( \ell_2 \). Apply Lemma 3 to each of these two lines.

Case 4: Point \( z \) is inside one arm of \( V' \). For each pair of consecutive points \( p,q \in P' \) on \( \ell_1 \) or on \( \ell_2 \) not separated by \( z \), apply Lemma 3 to the perpendicular bisector of the segment \( pq \).

Now we argue that the last procedure returns the best minimum-width V-shape \( V' \) of \( P' \) with two points on its outer boundary and \( z \) in one of its arms, correctly handling case 4 and thereby concluding our description.

Let \( x \) and \( y \) be the vertices of \( V' \). For ease of presentation, rotate the entire picture so that \( y \) lies below \( x \); refer to figure 3a. Let \( s_3 \) be the point of \( P \) on \( Y_1 \), \( s_1, s_2 \) be the points on \( X_1 \), with \( s_1 \) closer to \( x \) than \( s_2 \). Similarly let \( s_4 \) be the point on \( Y_2 \), \( s_5, s_6 \) be the points on \( X_1 \), with \( s_5 \) closer to \( x \) than \( s_6 \). As \( \ell_1 \) and \( \ell_2 \) don’t intersect between the two arms of \( V' \), \( s_1 \) and \( s_3 \) lie on one line, \( s_2 \) and \( s_6 \) lie on the other line. Let \( s_1 \) and \( s_5 \) lie on \( \ell_1 \), and \( s_2 \) and \( s_6 \) lie on \( \ell_2 \), without loss of generality.

The three points of \( P' \) on one arm boundary cannot lie on the same line \( \ell_i \), as they form a triangle. Therefore either each line \( \ell_i \) contains three boundary points belonging to three different boundary rays, or one line contains four boundary points from four boundary rays, and the other line contains two boundary points from the two inner rays.

We consider all possible cases:

- a) \( \ell_1 \) contains \( s_1, s_5, \) and \( s_4 \), and \( \ell_2 \) contains \( s_3, s_2, \) and \( s_6 \) (\( \ell_1 \) contains \( s_1, s_3, \) and \( s_5 \), and \( \ell_2 \) contains \( s_2, s_4, \) and \( s_6 \) is a symmetric case).
- b) \( \ell_2 \) contains four boundary points, \( \ell_1 \) contains two boundary points.
- c) \( \ell_2 \) contains two boundary points, \( \ell_1 \) contains four boundary points.

In each case, we prove either that the configuration of the points is impossible, or that the angles \( \alpha_1 \) and \( \alpha_2 \) (see figure 3a) between \( \ell_1 \) and the inner boundary rays are acute. When the angles are acute, the perpendicular bisector of \( s_1s_5 \) separates the points of \( P' \) belonging to each arm of \( V' \). Therefore Lemma 3 can be applied and our handling of case 4 is justified.

We consider the cases in turn:

- a) \( \ell_1 \) contains \( s_1, s_5, \) and \( s_4 \), and \( \ell_2 \) contains \( s_3, s_2, \) and \( s_6 \) (see figure 3a). As the angle \( \angle s_4s_5s_6 \) is acute, its opposite angle \( \alpha_2 \) is acute. What is left to prove is that \( \alpha_1 \) is acute as well. The angle \( \angle s_1s_3s_5 \) is acute, hence \( \alpha_3 \) is obtuse, and so is \( \alpha_6 \). But the angle \( \angle s_5s_6s_4 \) is acute, hence \( \ell_2 \) does not intersect \( \ell_1 \) in the right arm of \( V' \), so \( \ell_1 \) and \( \ell_2 \) intersect in the left arm. More
(a) Three boundary points on $\ell_1, \ell_2$. (b) Four boundary points on $\ell_2$

(c) Four boundary points on $\ell_1$.

Fig. 3: $V'$
precisely $\ell_1$ intersects the segment $s_3s_2$. The opposite angle of $\alpha_1$ is smaller than the angle $\angle s_2s_3s_1$, which is acute. Therefore so is $\alpha_1$.

b) $\ell_2$ contains four boundary points, $\ell_1$ contains two boundary points. Let $\ell_{s_2s_3}$ be the line containing $s_2, s_3$, $\ell_{s_4s_5}$ be the line containing $s_6, s_4$, and $\ell_{xy}$ be the line containing $x, y$ (see figure 3b). By Corollary 1, as the angle $\angle s_3s_2s_1$ is acute, $\ell_{s_2s_3}$ forms an acute angle $\alpha_3$ with $\ell_{xy}$. Similarly, as the angle $\angle s_5s_6s_4$ is acute, $\ell_{s_4s_5}$ forms an acute angle $\alpha_4$ with $\ell_{xy}$. But as $\ell_{s_3s_2} = \ell_{s_6s_4}$, $\alpha_3$ and $\alpha_4$ are supplementary, a contradiction.

c) $\ell_2$ contains two boundary points, $\ell_1$ contains four boundary points (see figure 3c). By Corollary 1, the angles $\angle s_2s_3s_1$ and $\angle s_4s_5s_6$ are acute, therefore their opposite angles $\alpha_1$ and $\alpha_2$ are acute as well.

6 A $(1 + \varepsilon)$-approximation algorithm

In this section we describe how to construct, given a point set $P$ and a real number $\varepsilon > 0$, a V-shape $V$ covering $P$, with $\text{width}(V) \leq (1 + \varepsilon)w_{\text{opt}}$, where $w_{\text{opt}}$ is the width of a minimum-width V-shape covering $P$.

We start by recalling the notion of an anchor pair used in [2]. Given a V-shape $V$ covering $P$, fix one of the strips of $V$, say $S_1$. We say that a pair of points $p, q \in P \cap S_1$ is an anchor pair, if $\text{dist}(p, q) \geq \text{diam}(P \cap S_1)/2$. Lemma 3.3 in [2] describes how to identify at most 11 pairs of points in $P$, such that, for any two-strip cover of $P$, at least one of the pairs is an anchor pair for one of the strips; the algorithm requires $O(n \log n)$ time. As covering by a V-shape is a special case of covering by two strips, the definition and the algorithm apply here as well.

We show how to, given a potential anchor pair $p, q$, construct a $(1 + \varepsilon)$-approximation of the minimum-width V-shape covering $P$ for which $p, q$ is an anchor pair. More precisely, below we prove

**Lemma 4.** Given a potential anchor pair $p, q \in P$, we can construct, in time $O((n/\varepsilon) \log n + (n/\varepsilon^{3/2}) \log^2(1/\varepsilon))$, a V-shape covering $P$, of width at most $1 + \varepsilon$ times the minimum width of any V-shape covering $P$ for which $p, q$ is an anchor pair.

Applying this procedure at most 11 times, we obtain our desired approximation algorithm:

**Theorem 4.** A V-shape covering $P$ and of width at most $(1 + \varepsilon)w_{\text{opt}}$ can be constructed in time $O((n/\varepsilon) \log n + (n/\varepsilon^{3/2}) \log^2(1/\varepsilon))$.

We first prove that it is sufficient to consider those V-shapes $V$ with anchor pair $p, q$, for which the strip containing $p, q$ has one of a small set of fixed directions. Setting $\beta := \sin^{-1}\min\{\varepsilon \cdot \text{width}(V)/(6d(p, q)), 1\}$ and $\gamma := \beta + \sin^{-1}\min\{1, \text{width}(V)^2/d(p, q)\}$, we prove the following

**Lemma 5.** Let $V$-shape $V$ cover $P$, and let $p, q$ be an anchor pair for $S_1(V)$. Rotating $S_1(v)$ by an angle at most $\beta$ does not increase the width of the V-shape by more than a factor of $1 + \varepsilon/3$, and the angle between $pq$ and the direction of the rotated strip cannot exceed $\gamma$. 
Proof. Put \( w := \text{width}(V) \). Let \( B \) be the minimum bounding box of \( P \cap S_1 \). More precisely, it is the shortest rectangle cut out of \( S_1 \) by two lines perpendicular to \( S_1 \) and containing \( P \cap S_1 \); refer to Figure 4. Let \( s \) and \( t \leq w \) be the length (along the axis of \( S_1 \)) and width of \( B \), respectively. Let \( S'_1 \) be the minimal parallel strip containing \( B \cap V \), whose direction is \( \alpha \leq \beta \) away from that of \( S_1 \) (there are two choices for \( S'_1 \), corresponding to rotating clockwise and counterclockwise; only one is shown; the argument applies to both cases). Then

\[
\text{width}(S'_1) \leq s \sin \alpha + t \cos \alpha \leq 2d(p,q) \sin \alpha + w \\
\leq w(1 + 2\frac{d(p,q)}{w} \sin \alpha) \leq t(1 + \varepsilon/3),
\]

since \( \sin \alpha \leq \sin \beta \leq \varepsilon w/(6d(p,q)) \). Now replace \( S_1 \) by \( S'_1 \) to obtain a new V-shape \( V' \) covering \( P \). Its width is \( \min\{\text{width}(S'_1), \text{width}(S_2)\} \leq (1 + \varepsilon/3)w \), as claimed.

Observe that in the above construction, the angle between \( pq \) and the direction of \( S'_1 \) cannot exceed

\[
\alpha + \sin^{-1}(\min\{1, t/d(p,q)\}) \leq \beta + \sin^{-1}(\min\{1, \text{width}(V)/d(p,q)\}) = \gamma.
\]

We conclude that enumerating all V-shapes that contain \( p,q \) in their strip \( S_1 \) and whose directions are (a) at most \( \gamma \) away from that of \( d(p,q) \) and (b) spaced at most \( \beta \) apart, would yield a V-shape whose existence is claimed in Lemma 4. The number of directions to be tested is at most \( O(\gamma/\beta) = O(1/\varepsilon) \).

Given a candidate anchor pair \( p,q \), the algorithm proceeds by starting with the direction \( pq \). Since we need not consider V-shapes whose width is larger than the approximate width \( w_{\text{apx}} \) computed in Section 5 (this is where the 13-approximation algorithm is used to bootstrap our \( 1 + \varepsilon \)-approximation), we replace \( \text{width}(V) \) by the smaller \( w_{\text{apx}}/13 \) in the definition of \( \beta \) above and by the larger \( w_{\text{apx}} \) in the definition of \( \gamma \), thereby erring on the conservative side in each case. Having computed (conservative estimates of) \( \beta \) and \( \gamma \), we enumerate the \( O(1/\varepsilon) \) directions of the form \( \theta_i := \theta_{pq} + i\beta \), where \( \theta_{pq} \) is the direction of \( pq \) and
$i$ is an integer ranging from $-\lceil\gamma/\beta\rceil$ to $\lceil\gamma/\beta\rceil$. It remains to explain how to deal with one such direction $\theta := \theta_i$.

**Lemma 6.** One can compute a canonical V-shape $V$ covering $P$ with one arm in given direction $\theta$ and width at most $1 + \varepsilon/3$ times the minimum width of any such V-shape, in time $O(n \log n + (n/\varepsilon^{1/2}) \log^2(1/\varepsilon))$.

**Proof.** We use an approach similar to that of the inner-outer case of our exact algorithm with a slight twist.

Let $\ell$ be a line in direction $\theta$ supporting $\text{CH}(P)$. We again let $q$ be the furthest point from $\ell$ in $Q := P \cap S_1$ and let $Q' := P \setminus Q$. When $q$ is fixed, the minimum-width V-shape is determined by the minimum-width strip $S_2$ covering $Q'$ and not “splitting” $P$, i.e., such that it does not have points of $P$ on both sides of it. It is easy to ensure that $S_2$ does not split $P$ by observing that a direction of $S_2$ lying between the directions of the common outer tangents to $\text{CH}(Q)$ and $\text{CH}(Q')$ is never useful. Depending on the side where the lines supporting these tangents cross, a minimal strip $S_2$ covering $Q'$ and lying in the range between them either crosses $Q$ (and therefore $P$) or completely covers $Q$ (and therefore $P$). In the former case, $S_1$ and $S_2$ do not form a legal V-shape covering $P$ and in the latter they form a covering V-shape with one empty strip, which never yields minimum width by reasoning as in Lemma 1.

The width of the resulting V-shape is the maximum of $\text{dist}(q, \ell)$ and (the restricted) width($S_2$). The algorithm proceeds by processing points $q$ in order of decreasing distance to $\ell$, keeping track of $\text{dist}(q, \ell)$ and a coreset for $Q'$, which is a subset of $Q'$ with the property that its directional width, in every direction, is at least $1 - \varepsilon/3$ that of $Q'$ (and, expanding the corresponding minimal strip containing the subset by a factor of $1 + \varepsilon/3$, we get a strip covering $Q'$). Chan [11], in Theorem 3.7 and remarks in Section 3.4, describes a streaming algorithm that maintains an $O(1/\varepsilon^{1/2})$-size coreset at an amortized cost of $O((1/\varepsilon^{1/2}) \log^2(1/\varepsilon))$ per insertion. For a fixed $q$, we go through the coreset (after computing its convex hull, if necessary), and determine the narrowest strip covering it and satisfying our angle constraints. The maximum of that and $\text{dist}(q, \ell)$ gives the width of the minimum-width V-shape whose boundary passes through $q$.\(^3\) The amortized cost per point is dominated by the $O((1/\varepsilon^{1/2}) \log^2(1/\varepsilon))$ cost of insertion. Together with presorting points by distance from $\ell$, the total cost is then $O(n \log n + (n/\varepsilon^{1/2}) \log^2(1/\varepsilon))$. \(\square\)

Combining Lemmas 5 and 6 yields the procedure claimed in Lemma 4 and thereby completes our description of the $(1 + \varepsilon)$-approximation algorithm.

### 7 Concluding remarks

As mentioned in the introduction, this work was inspired by research on curve fitting, in the situations where a curve takes a sharp turn. Besides the exact

\(^3\) More precisely, $q$ lies on the boundary of $S_1$ and may not even appear on the boundary of $V$. However, as before, all V-shapes we examine are valid and cover $P$, and the desired approximating V-shape is among them, which is sufficient.
and approximate versions of the problem studied above, it would be natural to investigate a variant that can handle a small number of outliers. A natural “peeling” approach to the problem would be to eliminate the points defining the optimal V-shape found by our exact algorithm and trying again. However, it is easy to construct an example of a point set in which removing a single point not appearing on the boundary of the minimum-width covering V-shape significantly reduces the width of the optimum V-shape.

Are there natural assumptions (perhaps in the spirit of “realistic input models” [9] or in the form of requiring reasonable sampling density) that would be relevant for the curve-fitting problem, and that would make finding the minimum-width covering V-shape easier?

Returning to the problem studied in the paper, is it possible to find an exact minimum-width covering V-shape in subquadratic time? Is the problem 3SUM-hard?

Is it possible to speed up the approximation algorithm, improving the dependence of its running time on \( \varepsilon \)? Is time \( O(n + f(\frac{1}{\varepsilon})) \) achievable?

Finally, we would like to point out that there are other “reasonable” definitions for a V-shape, if the goal is to approximate a sharp turn of a curve: One can imagine defining a V-shape as the Minkowski sum of a disk with the union of two rays emanating from a common point as in [17]. The width of the V-shape would be the diameter of the disk. Can the exact algorithm from [17] be sped up? Is there a faster approximation algorithm? Is this version of the problem better suited for curve fitting?

Acknowledgments

We thank Piyush Kumar for suggesting the problem studied in this paper, Pankaj K. Agarwal for extremely useful pointers, Alfredo Hubard for several helpful conversations, and Sariel Har-Peled for sharing his wisdom and dispelling some confusion about approximation.

References

1. P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan, Geometric approximation via coresets, in Current Trends in Combinatorial and Computational Geometry, (J. E. Goodman, J. Pach, and E. Welzl, eds.), MSRI Publications, Volume 52, Cambridge University Press, New York, 2005, pp. 1–30.
2. P. K. Agarwal, C. M. Procopiuc, and K. R. Varadarajan, A \((1 + \varepsilon)\)-approximation algorithm for 2-line-center, Comput. Geom. Theory Appl., 26(2):119–128, 2003.
3. P. K. Agarwal and S. Sen, Randomized algorithms for geometric optimization problems, in Handbook of Randomized Computation, J. Pardalos, S. Rajasekaran, J. Reif, and J. Rolim, eds., Kluwer Academic Press, The Netherlands, 2001, pp. 151–201.
4. P. K. Agarwal and M. Sharir, Algorithmic techniques for geometric optimization, in Computer Science Today: Recent Trends and Developments, Lecture Notes in Computer Science, vol. 1000 (J. van Leeuwen, ed.), Springer-Verlag, Berlin, 1995, pp. 234–253.
5. P. K. Agarwal and M. Sharir, Efficient algorithms for geometric optimization, *ACM Computing Surveys* 30:412–458, 1998.
6. H. Alt and L. J. Guibas, Discrete geometric shapes: matching, interpolation, and approximation, Chapter 3 in *Handbook of Computational Geometry*, J.-R. Sack and J. Urrutia, eds., 1999, pp. 121–153.
7. B. Aronov and M. Dulieu, How to cover a point set with a V-shape of minimum width, *Proc. Algorithms Data Structures Symp. (WADS’11)*, 2011, to appear.
8. M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed., Springer-Verlag, 2008.
9. M. de Berg, M. Katz, A.F. van der Stappen, and J. Vleugels, Realistic input models for geometric algorithms, *Algorithmica*, 34(1):81–97, 2006.
10. T. M. Chan, Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus, *Inter. J. Comput. Geom. Appl.*, 12:67–85, 2002.
11. T. M. Chan, Faster core-set constructions and data-stream algorithms in fixed dimensions, *Comput. Geom. Theory Appl.*, 35:20–35, 2006.
12. S.-W. Cheng, S. Funke, M. Golin, P. Kumar, S.-H. Poon, and E. Ramos, Curve reconstruction from noisy samples, *Comput. Geom. Theory Appl.*, 31(1–2):63–100, 2005.
13. T. K. Dey, *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*, Cambridge Monographs on Applied and Computational Mathematics, Vol. 23, Cambridge Univ. Press, 2007.
14. J. R. Driscoll, N. Sarnak, D. D. Sleator, and R. E. Tarjan, Making data structures persistent, *J. Computer System Sci.*, 38(1):86–124, 1989.
15. H. Edelsbrunner, J. O’Rourke, and R. Seidel, Constructing arrangements of lines and hyperplanes with applications, *SIAM J. Comput.*, 15:341–363, 1986.
16. S. Funke and E. A. Ramos, Reconstructing a collection of curves with corners and endpoints, in *Proc. 12th ACM-SIAM Symp. Discr. Algorithms (SODA 2001)*, 2001, pp. 344–353.
17. A. Glozman, K. Kedem, and G. Shpitalnik, Computing a double-ray center for a planar point set, *Inter. J. Comp. Geom. Appl.*, 2(9):103–123, 1999.
18. A. Mhatre and P. Kumar, Projective clustering and its application to surface reconstruction: extended abstract, *Proc. 22nd Annu. Symp. Comput. Geom.*, 2006, pp. 477–478.
19. P. Kumar, private communication, 2010.
20. R. Pinchasi, R. Radoičić, and M. Sharir, On empty convex polygons in a planar point set, *J. Combinat. Theory, Series A*, 113:385–419, 2006.
21. F. P. Preparata, An optimal real time algorithm for planar convex hulls, *Comm. ACM*, 22:402–405, 1979.