ELSEVIER

Computational Geometry 19 (2001) 1–33

Computational Geometry
Theory and Applications

www. elsevier.nl/locate/comgeo

Efficient randomized algorithms for robust estimation of circular arcs and aligned ellipses ☆

David M. Mount [a,1], Nathan S. Netanyahu [b,c,*,2]

[a] *Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA*

[b] *Department of Mathematics and Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel*

[c] *Center for Automation Research, University of Maryland, College Park, MD 20742, USA*

Communicated by J. Urrutia; received 1 December 1997; accepted 18 September 2000

Abstract

Fitting two-dimensional conic sections (e.g., circular and elliptical arcs) to a finite collection of points in the plane is an important problem in statistical estimation and has significant industrial applications. Recently there has been a great deal of interest in *robust estimators*, because of their lack of sensitivity to outlying data points. The basic measure of the robustness of an estimator is its *breakdown point*, that is, the fraction (up to 50%) of outlying data points that can corrupt the estimator. In this paper we introduce *nonlinear* Theil–Sen and repeated median (RM) variants for estimating the center and radius of a *circular arc*, and for estimating the center and horizontal and vertical radii of an *axis-aligned ellipse*. The circular arc estimators have breakdown points of $\approx 21\%$ and 50%, respectively, and the ellipse estimators have breakdown points of $\approx 16\%$ and 50%, respectively. We present randomized algorithms for these estimators, whose expected running times are $O(n^2 \log n)$ for the circular case and $O(n^3 \log n)$ for the elliptical case. All algorithms use $O(n)$ space in the worst case.

Keywords: Circular arc fitting; Aligned ellipse fitting; Robust estimation; Theil–Sen estimator; RM estimator; Randomized algorithms; Computational geometry; Arrangements; Range searching

☆ A preliminary version of this report was presented at the Fifth Canadian Conference on Computational Geometry [32].

* Corresponding author.

E-mail addresses: mount@cs.umd.edu (D.M. Mount), nathan@macs.biu.ac.il (N.S. Netanyahu).

[1] Part of this research was done while the author was visiting the Max-Planck Institut für Informatik, Saarbrücken, Germany. The support of the National Science Foundation under grant CCR–9310705 is gratefully acknowledged.

[2] This research was carried out, in part, while the author was also affiliated with the Center of Excellence in Space Data and Information Sciences, Code 930.5, NASA Goddard Space Flight Center, Greenbelt, MD 20771.

0925-7721/01/$ – see front matter
PII: S0925-7721(01)00009-8

1. Introduction

Fitting a curve, for example, a straight line or a circular arc, to a finite collection of data points in the plane is a fundamental problem in statistical estimation, with numerous industrial applications. Although methods such as ordinary least squares (OLS) are well understood and easy to compute, they are known to suffer from the phenomenon that a small number of outlying points can perturb the function of fit by an arbitrarily large amount. For this reason, there has been a growing interest in a class of estimators, called *robust estimators* [22,25,38], which do not suffer from this deficiency. Define the *breakdown point* of an estimator to be the fraction of outlying data points (up to 50%) that may cause the estimator to take on an arbitrarily large aberrant value. (See Donoho and Huber [13] and Rousseeuw and Leroy [38] for exact definitions.) The breakdown point of an estimator is a measure of its robustness. For example, the (asymptotic) breakdown point of OLS is zero because even a single outlying data point can have an arbitrarily large effect on the estimator. Examples of robust line estimators include the following.

Theil–Sen estimator: The slope of the line of fit is taken to be the median [3] of the set of $\binom{n}{2}$ slopes that result by passing a line through each pair of distinct points in the data set [40,47]. (The intercept is defined analogously, in terms of line intercepts.) In the plane, the Theil–Sen estimator has a breakdown point of $\approx 29.3\%$.

This problem has been studied under the name of *slope-selection* in the field of computational geometry. The problem is to determine the slope of any given rank. There exist asymptotically optimal algorithms for this problem, which run in $O(n \log n)$ time and $O(n)$ space. These include algorithms by Cole et al. [10], Katz and Sharir [27] and Brönnimann and Chazelle [4].

It should be noted that all of the above algorithms rely on fairly complicated techniques. There are simpler, practical *randomized* algorithms by Matoušek [29], Dillencourt et al. [12] and Shafer and Steiger [41]. (These are *Las Vegas* randomized algorithms, meaning that they always produce correct results, and on any input, the expected running time, when averaged over the random choices made in the algorithm, is $O(n \log n)$. All the randomized algorithms presented here will be of this same type.)

RM estimator: Siegel's *repeated median* (RM) estimator [42] of a set of n distinct points in the plane $\{p_1, p_2, \ldots, p_n\}$ is defined as follows. For each point p_i, let θ_i denote the median of the $n-1$ slopes of the lines passing through p_i and each other point of the set. The RM-slope, θ^*, is defined to be the median of the multiset $\{\theta_i\}$. The RM-intercept is defined analogously, in terms of line intercepts. The RM estimator has a breakdown point of 50%, and the best known algorithm for its computation, due to Matoušek, Mount and Netanyahu [31], is randomized and runs in $O(n \log n)$ expected time.

LMS estimator: Rousseeuw's *least median of squares* (LMS) estimator [37] is defined to be the line that minimizes the median of the squared residuals. LMS has a breakdown point of 50%. The best algorithms known for LMS, due to Souvaine and Steele [44] and Edelsbrunner and Souvaine [15], run in $O(n^2)$ time. (Recently, Mount et al. [34] have presented a Las Vegas approximation algorithm that runs in $O(n \log n)$ time.)

[3] For the purposes of this paper we define the *median* of an m element multiset to be an element of rank $\lceil m/2 \rceil$.

Mount and Netanyahu [33] showed that it is possible to extend the algorithmic results for computing Theil–Sen and RM line estimators to higher dimensions. In dimension d, the problem is to fit a $(d-1)$-dimensional hyperplane to the given data points. Specifically, they showed that d-dimensional Theil–Sen and RM estimators (having breakdown points of $1-(1/2)^{1/d}$ and 50%, respectively) can be computed by randomized algorithms in $O(n^{d-1}\log n)$ expected time and $O(n)$ space, for fixed $d>2$.

In this paper we consider a generalization of these estimators to a *nonlinear* domain in the plane. In particular, given n data points in the plane, we consider the problems of robustly fitting either a circular arc or the arc of an aligned ellipse to the points. (An *aligned ellipse* is an ellipse whose axes are parallel to the coordinate axes.) In the case of the circular arc estimator (CAE), we return the center coordinates and radius of the circle. For the aligned ellipse estimator, we return the center, horizontal radius and vertical radius of the aligned ellipse. We have chosen to present two different types of curves as evidence that our algorithmic methodology is extendable to a wide variety of curves and formulations.

We generalize the definitions of the Theil–Sen and RM line estimators to circular arc estimation in the following natural way. (The ellipse estimators will be presented later in Section 4.) Consider a given set of distinct points $p_i=(x_i,y_i)$, for $1\leqslant i\leqslant n$, which are hypothesized to lie on a circular arc. It will simplify the presentation significantly to assume that the points are in general position. For example, we assume that no three points are collinear and no four points are cocircular. (These assumptions can be overcome at the expense of a great number of special cases, which would need to be considered.) The result of the estimator is a triplet $(\widehat{a},\widehat{b},\widehat{r})$, containing the coefficients of the circle equation $(x-\widehat{a})^2+(y-\widehat{b})^2=\widehat{r}^{\,2}$.

Theil–Sen circular arc estimator: For each triplet (i,j,k), $1\leqslant i<j<k\leqslant n$, consider the circle passing through the three points p_i, p_j and p_k. Let $a_{i,j,k}$, $b_{i,j,k}$ and $r_{i,j,k}$ denote the parameters of this circle. The estimate is given by the median values, over all $\binom{n}{3}$ triples of points, of each of the above parameters,

$$\widehat{a}=\operatorname{med} a_{i,j,k},\quad \widehat{b}=\operatorname{med} b_{i,j,k}\quad\text{and}\quad \widehat{r}=\operatorname{med} r_{i,j,k}.$$

RM circular arc estimator: As before, associate parameters $a_{i,j,k}$, $b_{i,j,k}$ and $r_{i,j,k}$ with each triplet (i,j,k), such that $i\neq j\neq k$. The center's estimated coordinates are given by

$$\widehat{a}=\operatorname*{med}_i \operatorname*{med}_{j\neq i} \operatorname*{med}_{k\neq i,j} a_{i,j,k},\qquad \widehat{b}=\operatorname*{med}_i \operatorname*{med}_{j\neq i} \operatorname*{med}_{k\neq i,j} b_{i,j,k},$$

and the estimated radius is given by

$$\widehat{r}=\operatorname*{med}_i \operatorname*{med}_{j\neq i} \operatorname*{med}_{k\neq i,j} r_{i,j,k}.$$

Somewhat more intuitively, each parameter of the Theil–Sen CAE is defined by considering all triples of points, computing the circle passing through the points of each triple, and then selecting the median value of the corresponding parameter over all these circles. For the repeated median CAE, each triplet (i,j,k) determines a single circle, and hence a unique parameter value. For each pair (i,j) we take the median parameter over all $n-2$ choices of the third point. For each singleton, i, we consider the median over all $n-1$ choices of a second point, and so on. Incidentally, the choice of median plays no significant role in the design of our algorithms or their efficiency. Elements of any fixed rank could be used anywhere that medians are mentioned.

Observe that generalizing the above definitions of the estimators to other types of curves and other choices of parameterizations is straightforward, provided that, for some k, each k-tuple of points uniquely

defines an object of the desired class. These estimators are *non-hierarchical*, meaning that the parameters are defined independently of one another (as opposed to deriving the parameters sequentially, and using the values of known parameters to reduce the search dimension for the other parameters).

It follows from standard arguments [38] that, because they are based on triples of observations, the breakdown points of the Theil–Sen and RM CAE's are $1 - \sqrt[3]{1/2} \approx 21\%$ and 50%, respectively. Fig. 1(a) depicts the Theil–Sen and RM circular arc estimators (versus a least squares fit) obtained for a data set having 20% outliers. Fig. 1(b) shows the same estimators obtained with 40% outlying data.

Recently Stein and Werman [45] have independently introduced similar robust estimators for fitting general 2-D conic sections. Their estimators have the nice property of being rotationally equivariant, meaning that rotating the points through some angle and then computing the estimator is equivalent to computing the estimator of the original point set and then rotating it through the same angle. Our radius estimator is rotationally equivariant, but our estimators for the center coordinates of the circle are not (because they depend on the choice of a coordinate system). Unfortunately, we know of no methods for computing their estimators other than brute force.

Before stating our results, we digress momentarily to consider an issue which is central to nonlinear curve fitting. It is well-known that there is a simple method of reducing the problem of fitting algebraic curves to the linear problem of fitting hyperplanes in higher dimensions, through a process called *linearization* (see, e.g., [3]). For example, fitting a circle of the form $(x - a)^2 + (y - b)^2 = r^2$ to a set of data points (x_i, y_i) in the plane, can be reduced to the problem of fitting a plane in 3-D space. This is done by first writing the circle formula in a form that is linear in the parameters $X = x$, $Y = y$ and $Z = x^2 + y^2$, i.e.,

$$(x^2 + y^2) = 2ax + 2by + (r^2 - a^2 - b^2),$$
$$Z = 2aX + 2bY + (r^2 - a^2 - b^2).$$

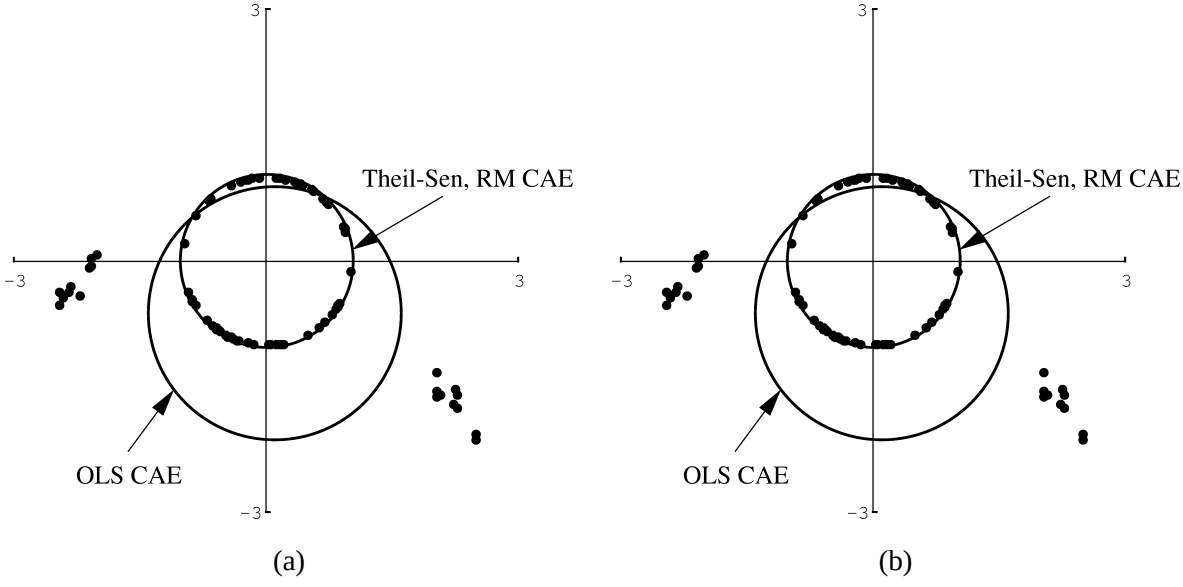

Fig. 1. Theil–Sen and RM CAE's, and an OLS fit, for data sets having (a) 20% outliers and (b) 40% outliers.

Then a plane of the form $Z = AX + BY + C$ is fitted to the points $(X_i, Y_i, Z_i) = (x_i, y_i, x_i^2 + y_i^2)$ in 3-D space. Finally, the original parameters are extracted from the transformed parameters. (The corresponding transformation for aligned ellipses produces a linear problem in 4-D.)

The problem with linearization is that it computes estimators for the "transformed" parameters, and not the parameters that were supplied as part of the user's original formulation. Thus, it is doubtful that the statistical properties of the derived parameters would be the same as those that would result from the above definitions. In fact, it was noted by many that estimators based on linearization often yield *biased* results. See, e.g., Joesph [26], Cabrera and Meer [5] and Netanyahu et al. [35]. Also, Rosin [36] observed empirically that applying linearization to a robust ellipse estimator of a similar type to the ones considered in this paper leads to inaccurate results. Thus, it seems desirable that fitting be applied to the original parameters of the circle or (aligned) ellipse.

The problem of fitting circular arcs to a given set of points in the plane has been studied extensively in the fields of pattern recognition and computer vision. Several representative examples include the papers by Landau [28], Takiyama and Ono [46], Thomas and Chan [48], Chaudhuri [6], Chaudhuri and Kundu [7], Joseph [26], Yi et al. [53], Wu et al. [51] and Yuen and Feng [52]. Unfortunately, most of these methods—even if posed in (geometric) terms that lead to less biased results—are either based on a least squares approach or make simplistic assumptions with regards to noise. Thus, they are likely to be sensitive to outlying data. Amir [2] has introduced an alternative technique, the "cord (*sic.*)" method, which is presumably more robust. His "Hough-like" [14,24] technique is applicable, primarily, to edge data from an image. In general, however, the method is likely to be sensitive to quantization effects due to discretization of the parameter space. More recently, Rosin [36] proposed a robust method, based on a 5-D Theil–Sen variant, for the more general case of ellipse fitting. His method assumes that the data are preordered according to some criterion, and that it suffices to consider merely $O(n^2)$ 5-tuples (selected in a specific manner) to achieve good estimation. However, Rosin's heuristic may not always return the correct computational result (by definition of the Theil–Sen ellipse estimator). Another alternative approach would be to compute Theil–Sen and RM (circular arc and ellipse) estimators in a brute-force manner with respect to a (relatively) small random sample of the given data points. While such Monte-Carlo-like algorithms would run considerably faster, the estimates they return could differ substantially from the exact values (computed by definition). In principle, these estimates vary according to some probability density function, so one could bound the probability that an estimate deviates (from its exact value) by less than a pre-specified amount. In the presence of outliers, however, with finite probability the resulting estimates could deviate *arbitrarily* with respect to their exact values. In other words, the above probabilistic approach provides, essentially, no guarantee of accuracy. All of the above remarks suggest, therefore, that deriving computationally efficient algorithms for *exact* median-based circular arc and ellipse estimators is a valid goal to pursue.

The Theil–Sen and RM circular arc estimators can be computed by a brute-force implementation of their definitions. This would require $O(n^3)$ time for each, since this is the number of triples that would have to be computed. Moreover, the brute-force algorithm for the Theil–Sen estimator would require $O(n^3)$ space. For the case of aligned ellipses, these running times grow to $O(n^4)$ because of the extra degree of freedom. In this paper we present conceptually simple randomized algorithms to compute the above estimators. For the circular arc case, the algorithms run in $O(n^2 \log n)$ expected time. For the aligned ellipse case, the algorithms run in $O(n^3 \log n)$ expected time. In all cases the algorithms use optimal $O(n)$ storage in the worst case. (Note that solving the problems by linearization would result in algorithms that are no more efficient than these.) The algorithms always terminate and return the exact

results (relative to the precision used in the underlying arithmetic). As mentioned earlier, randomization does not affect the accuracy of the results, only the running time. However, the stated expected running times hold with high probability. Furthermore, variations in running time are completely independent of input data point distribution, and depend only on the random choices made by the algorithm. (Removal of randomization is possible, but would make the algorithms significantly more complex.)

The paper is organized as follows. In Section 2 we give an overview of our algorithmic methodology for computing the circular arc estimators. The general framework is similar to the interval contraction scheme described in earlier papers. (See, for example, [12,31].) In Section 3 we provide a more detailed discussion concerning generalizations of the various "building blocks" needed for the nonlinear estimators considered. These include intersection counting in arrangements of pseudolines, intersection sampling, and range searching. Section 4 demonstrates how to extend our algorithmic methodology to aligned ellipse fitting, and Section 5 contains concluding remarks.

2. The algorithmic framework

We begin by presenting algorithms for computing the Theil–Sen and RM estimators for the circular arc case. The general approach applies to both the circular and elliptical cases, and is based on a generalization of the randomized algorithms for the line estimators presented by Dillencourt et al. [12] and Matoušek et al. [31]. First, the problems are mapped onto a *dual* setting, which allows us to identify circles in primal space with points in dual space. Instead of considering circles passing through triples of points in the *primal* plane, we visualize the problem in one of n dual planes. In the ith dual plane, a point with coordinates (a, b) is associated with the unique circle in the primal plane, that passes through p_i, and has center (a, b). We will show that the set of circles in the primal plane that pass through p_i and any two other data points are in 1–1 correspondence with the $O(n^2)$ vertices (intersection points) of an arrangement of $n-1$ lines in the dual plane. (In particular, it will be shown that the line arrangement of the ith dual plane is formed of the $n-1$ perpendicular bisectors of the segments $p_i p_j$, for all $j \neq i$.) The value of each estimatimated parameter will be realized by one of the vertices in one of the above arrangements. Our algorithms will employ techniques for searching arrangements to locate the intersection point of interest in a dual plane, that is, the circle of interest in the primal plane. Sections 2.1 and 2.2 will describe the relationship between the line arrangements and the required estimators.

To determine the intersection points of interest, we generalize the technique introduced in [12,31] to a *region contraction* scheme. Intuitively, we identify a region of the dual plane that contains the intersection point of interest. Through random sampling of intersection points, we can identify a subregion which will contain, with high probability, the desired intersection point. Using methods to count the intersection points lying within a given region of an arrangement, we verify our choice of the contracted subregion. Based on the results of this verification, we recurse either on this subregion or on some other subregion. Algorithmic techniques for sampling and counting intersections will be explained in Section 3. We will show that in the expected case, after a constant number of contraction stages, the algorithm locates the desired intersection point. A high-level illustration of the basic elements of the algorithm is shown in Fig. 2.

Before presenting the algorithmic details, we present the basic probability theoretic result on which the region contraction technique relies. The lemma below follows from Lemmas 3.1 and 3.2 in [12], and is outlined here for the sake of completeness. Intuitively, it states that given a set of n numbers from which

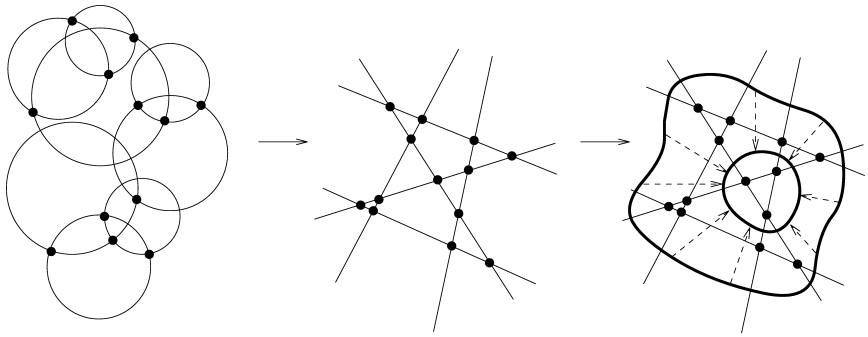

Fig. 2. Overview of the algorithms. Circles in primal space are associated with intersection points of line arrangements in dual space. Locating the desired intersection point is carried out through iterative region contraction.

we can sample at random, we can compute a small confidence interval for the kth smallest member of the set in time that is essentially independent of n. There is a tradeoff between the running time, the degree of confidence, and the size of the confidence interval (measured as the number of elements of the set that lie within the interval).

Lemma 2.1. *Given a set of numbers $X = \{x_1, x_2, \ldots, x_n\}$, given k $(1 \leqslant k \leqslant n)$, and given $m > 0$, in $O(m)$ time we can compute an interval $[x_{\mathrm{lo}}, x_{\mathrm{hi}}]$, such that with probability $1 - 1/\Omega(\sqrt{m})$, the kth smallest element of X lies within this interval. Furthermore, with this same probability, the number of elements in X that lie within the interval is at most $n/\Omega(\sqrt{m})$.*

Proof. Sample (with replacement) m of the elements of X, and select the elements x_{lo} and x_{hi} from this sample of respective ranks,

$$k_{\mathrm{lo}} = \max\left(1, \left\lfloor \frac{mk}{n} - \frac{3\sqrt{m}}{2} \right\rfloor\right), \qquad k_{\mathrm{hi}} = \min\left(m, \left\lceil \frac{mk}{n} + \frac{3\sqrt{m}}{2} \right\rceil\right).$$

This can be done in $O(m)$ time using any fast (possibly randomized) selection algorithm (see, e.g., [11, 19,20]).

The kth smallest element is less than x_{lo} if and only if fewer than k_{lo} sampled elements are less than the kth smallest element. Since the probability that a given element is less than or equal to the kth smallest is k/n, it follows that in m samples, the number of sampled values that are less than the kth smallest is a binomial random variable with mean mk/n and standard deviation not greater than $\sqrt{m}/2$. The probability that fewer than k_{lo} sampled elements are less than x_k is essentially the probability that this random variable is at least three standard deviations below its mean value. By applying Chernoff's bounds (see, e.g., [9,17]) and Chebyshev's inequality [18], it follows that this probability is $1/\Omega(\sqrt{m})$. See Lemmas 3.1 and 3.2 in [12] for complete details. A similar argument applies for k_{hi}. The probability that the kth smallest element does lie within the interval is, therefore, $1 - 1/\Omega(\sqrt{m})$.

Since a fraction of the sample of size $O(\sqrt{m})/m$ lies within the interval (by definition) it follows that the expected fraction of X that lies within the interval is $nO(\sqrt{m})/m = n/\Omega(\sqrt{m})$. Again, Chernoff's bounds can be invoked to show that this occurs with at least the stated probability. □

2.1. Theil–Sen circular arc estimator

In this subsection we present a high-level description of our algorithm for computing the Theil–Sen circular arc estimator. We focus on the computation of the estimated radius, $\widehat{r}$, since the algorithms for the center's estimated coordinates, $\widehat{a}$ and $\widehat{b}$, are simpler variants of this case. Recall that the radius estimator corresponds to the median over all radii, $r_{i,j,k}$, of circles passing through p_i, p_j and p_k, where $1 \leqslant i < j < k \leqslant n$. As mentioned earlier, it is assumed that the points are in general position. In particular, no three points are collinear (so that each triple defines a circle) and no four points are cocircular (so the circles determined by different triples are distinct).

We begin by introducing the dual transformation, which identifies circles in the primal plane with points in a dual plane. For $1 \leqslant i \leqslant n-2$, consider a fixed data point p_i. Let R_i denote the multiset of radii of circles passing through p_i and two other data points of increasing indices. (We order the triples by increasing index so that each circle is associated with a unique ordered triple.) Clearly the Theil–Sen radius estimator is just the median of the multiset $\bigcup_{i=1}^{n-2} R_i$. For each $j > i$, consider the centers of all circles passing through p_i and p_j. This locus is clearly the perpendicular bisector, $b_{i,j}$, of the two points. Thus, the circles passing through p_i and p_j in the primal plane are in 1–1 correspondence with the points of $b_{i,j}$ (in the ith dual plane) that are associated with p_i. In particular, let B_i denote the line arrangement in the ith dual plane defined by the set of $n-i$ bisectors $\{b_{i,j},\ j > i\}$. For any $k > j$, the intersection of lines $b_{i,j}$ and $b_{i,k}$ is the center of the circle passing through p_i, p_j and p_k. Thus, the vertices of this line arrangement are in 1–1 correspondence with the centers of the circles determined by p_i and any two other points of higher index. For the purposes of computing the Theil–Sen radius, each vertex in this arrangement can be associated with the radius of the corresponding circle. (The radius is simply the Euclidean distance from the vertex to p_i.)

Thus we have reduced the task of computing the Theil–Sen radius estimator, $\widehat{r}$, to the following problem. Given a set of $n-2$ line arrangements (that correspond to $n-2$ dual planes), where a vertex of each arrangement is associated with a radius (its distance to a fixed point in the plane), determine the median radius over all the vertices in these arrangements. We do not compute these arrangements explicitly, but they provide a convenient perspective from which to view the computation.

As mentioned before, we compute $\widehat{r}$ by a region contraction scheme. We maintain an interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$, which contains $\widehat{r}$. The initial interval is $(0, +\infty]$. (This is analogous to the convention adopted in [12], namely that an interval is treated as half-open half-closed.) The interval is contracted through a series of stages. We will argue that each stage runs in $O(n^2 \log n)$ expected time and requires $O(n)$ space, and that the number of stages in the expected case is $O(1)$.

Let us describe the operations performed during a typical stage in greater detail. Let $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$ be the current interval, and assume that $\widehat{r} \in (r_{\mathrm{lo}}, r_{\mathrm{hi}}]$. We consider separately the line arrangement associated with each p_i, $1 \leqslant i \leqslant n-2$. In the ith dual plane, the locus of the centers of circles that pass through p_i and whose radius lies in this half-open half-closed interval is an annulus centered at p_i with radii r_{lo} and r_{hi}. Let $A_i(r_{\mathrm{lo}}, r_{\mathrm{hi}})$ denote this annulus. (Note that for $r_{\mathrm{lo}} = 0$, the annulus is a disk with its center removed, and for $r_{\mathrm{hi}} = +\infty$, the outer disk of the annulus spans the entire plane.) We will maintain three counts: I_i, W_i and O_i which denote, respectively, the number of intersections *inside* the circle $r = r_{\mathrm{lo}}$,

within the annulus $A_i(r_{\mathrm{lo}}, r_{\mathrm{hi}})$ and *outside* the circle $r = r_{\mathrm{hi}}$. Let I, W, O, denote, respectively, the sums of I_i, W_i and O_i over i. (In terms of the primal plane, the count W is equal to the number of circles that pass through three data points and whose radius lies within the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$. The counts I and O have similar interpretation but for radii less than (or equal to) r_{lo} and greater than r_{hi}, respectively.) Since we assume that $\widehat{r}$ lies within $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$, it follows that

$$I < \left\lceil \frac{\binom{n}{3}}{2} \right\rceil \leqslant I + W.$$

We are searching for the median intersection, that is, the intersection of rank $k = \lceil \binom{n}{3}/2 \rceil - I$ from the current annulus. If there are only a small number of intersection points within the annulus, we can simply enumerate them and select the element of the desired rank by brute force. Otherwise, we will apply Lemma 2.1 to find a contracted confidence interval for the desired radius. Since we do not have ready access to the various arrangements, we need a method to randomly sample intersection points from the these arrangements in an efficient manner. This procedure will be presented later in Section 3.2. Assuming that we can sample intersection points, we apply Lemma 2.1 to the sample to determine a contracted confidence interval $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$ for the desired radius. However, since we cannot be sure that the desired radius is in this interval, we partition the original interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$ into three subintervals, and then count the number of intersections within each subinterval to determine which one contains the median radius. This counting procedure will be presented later in Section 3.1. Once the counts are known, we can contract to one of the three subintervals (expecting it to be $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$), and proceed to the next stage. The detailed procedure for the Theil–Sen radius estimator can now be given.

Algorithm 1 (Theil–Sen circular arc radius estimator).

(1) Set the initial interval to $(r_{\mathrm{lo}} = 0, r_{\mathrm{hi}} = +\infty]$. Initialize $I := O := 0$ and $W := \binom{n}{3}$.

(2) Repeat the following steps until $W = 1$ (or more practically, until $W = \mathrm{O}(n)$, after which brute-force enumeration followed by a standard fast selection procedure can be used).

(2a) For each $1 \leqslant i \leqslant n-2$, consider the set of the intersection points of arrangement B_i that lie in annulus $A_i(r_{\mathrm{lo}}, r_{\mathrm{hi}})$. Consider the union of this set of intersection points. Using the methods to be described in Section 3.2, randomly sample (with replacement) $m = n$ elements from this union.

(2b) For each sampled intersection point, compute its radius with respect to the corresponding point, p_i. These radii will all lie in the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$.

(2c) Let $k := \lceil \binom{n}{3}/2 \rceil - I$. (That is, k is the rank of $\widehat{r}$ in the set of radii that remain under consideration.) Applying Lemma 2.1, let

$$k_{\mathrm{lo}} := \max\left(1, \left\lfloor \frac{mk}{W} - \frac{3\sqrt{m}}{2} \right\rfloor\right), \qquad k_{\mathrm{hi}} := \min\left(m, \left\lceil \frac{mk}{W} + \frac{3\sqrt{m}}{2} \right\rceil\right).$$

Employ any fast selection algorithm to determine the elements r'_{lo} and r'_{hi} of the respective ranks k_{lo} and k_{hi} from the set of sampled radii. (We expect the median radius to lie in the interval $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$.)

(2d) Partition the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$ into three disjoint subintervals $(r_{\mathrm{lo}}, r'_{\mathrm{lo}}]$, $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$ and $(r'_{\mathrm{hi}}, r_{\mathrm{hi}}]$. (Recall, we treat each annulus/interval as if it is open on the left and closed on the right, so that no intersection lies in more than one annulus/interval.) Using the method to be described in Section 3.1, for $1 \leqslant i \leqslant n-2$, count the number of intersections lying in each of the associated

annuli $A_i(r_{\mathrm{lo}}, r'_{\mathrm{lo}})$, $A_i(r'_{\mathrm{lo}}, r'_{\mathrm{hi}})$ and $A_i(r'_{\mathrm{hi}}, r_{\mathrm{hi}})$. (Note that, for each i, only two annuli need to be counted, since the third count can be inferred from the other two.)

(2e) Based on these counts, determine which of the three subintervals contains the median radius. Make this the new interval, and update the counts I, W and O, accordingly.

Note that other than the counting and sampling subtasks (Section 3 below), the algorithm makes no assumptions about the geometric structure of the arrangements. This is important, since the same procedure can be used for the other circular parameters, as well as for the ellipse parameters to be discussed in Section 4.

To derive the running time of the above algorithm, first consider the number of stages required until the algorithm terminates. According to Lemma 2.1, with probability at least $1 - 1/\Omega(\sqrt{n})$, the median radius is contained within the annulus associated with the interval $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$, and the count W will be reduced by a factor of $1/\Omega(\sqrt{n})$ for the next stage. If we repeat this process t times, the magnitude of W decreases by a factor of $1/\Omega(n^{t/2})$. Since we began with a count of $\binom{n}{3}$ intersections, it is expected that within a constant number of stages ($t = 6$), we will have satisfied, with high probability, the termination condition of step (2).

It is easy to verify that each of the steps of the algorithm can be performed in $O(n)$ time, except for the intersection counting and sampling subtasks. Later we will show that these tasks can be performed in $O(n \log n)$ time and $O(n)$ space for each arrangement, and hence take $O(n^2 \log n)$ time in total. Therefore, the total expected running time over the constant number of contraction stages is $O(n^2 \log n)$.

The algorithms for $\widehat{a}$ and $\widehat{b}$ are similar, but the corresponding regions are different. Recall that for $\widehat{r}$, the annulus, $A_i(r_{\mathrm{lo}}, r_{\mathrm{hi}})$, is the locus of the centers of circles passing through p_i and whose radius lies within the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$. For $\widehat{a}$, the analogous region is the set of centers of circles the x-coordinate of which lies within some interval $(a_{\mathrm{lo}}, a_{\mathrm{hi}}]$. Clearly this is just the vertical strip, $a_{\mathrm{lo}} < x \leqslant a_{\mathrm{hi}}$, in the dual plane. Similarly, for $\widehat{b}$, we have a horizontal strip. Subject to the missing subtasks we have the following result.

Theorem 2.1. *The non-hierarchical Theil–Sen circular arc estimator can be computed in* $O(n^2 \log n)$ *expected time and* $O(n)$ *space.*

2.2. Repeated median circular arc estimator

In this subsection we present a randomized algorithm for the RM circular arc estimator. Again, we only present the computation of the most illustrative case, namely the radius parameter, $\widehat{r}$. (The modifications needed to compute the other parameters, $\widehat{a}$ and $\widehat{b}$, are largely the same as those mentioned in the previous subsection.)

Recall that $r_{i,j,k}$ is the radius of the circle passing through data points p_i, p_j and p_k. To compute the RM radius, $\widehat{r}$, we first define

$$\widehat{r}_i \triangleq \operatorname*{med}_{j \neq i} \operatorname*{med}_{k \neq i,j} r_{i,j,k}.$$

Intuitively, $\widehat{r}_i$ is the *radius estimator* associated with a fixed point p_i. The algorithm we present computes $\widehat{r}_i$ in expected $O(n \log n)$ time for each i, $1 \leqslant i \leqslant n$. Any fast selection algorithm may then be applied to compute $\widehat{r} = \operatorname{med}_i \widehat{r}_i$. Thus, the total running time of the algorithm will be $O(n^2 \log n)$.

Fixing i, $1 \leqslant i \leqslant n$, we focus on the computation of $\widehat{r}_i$. By definition, this estimate corresponds to a 2-D RM computation over all radii, $r_{i,j,k}$, of circles passing through p_i and two other distinct points p_j and p_k ($j \neq i$, $k \neq i, j$). As before, let $b_{i,j}$ denote the perpendicular bisector of the segment joining p_i and p_j, and let $B_i = \{b_{i,j},\ j \neq i\}$ denote the planar line arrangement of these $n-1$ bisectors in the ith dual plane. (Note that the index ordering used in the Theil–Sen case is not used here.) Recall that the intersection point of the lines $b_{i,j}$ and $b_{i,k}$ in this arrangement is the center of the circle passing through p_i, p_j and p_k, and hence the vertices of the arrangement are in 1–1 correspondence with the circles determined by p_i and any two other distinct points. Specifically, each vertex of the arrangement is associated with the radius of such a circle. For each bisector, $b_{i,j}$, the median among the radius values of all the intersection points on this bisector is called the *median radius* for the bisector. In terms of the primal plane, this is the median radius among all circles passing through both p_i and p_j. Thus, we have reduced the problem of computing $\widehat{r}_i$ to that of determining the median of the $n-1$ median radii.

To compute $\widehat{r}_i$, we apply the same interval contraction technique presented in the randomized algorithms for the RM line estimator [31]. We maintain an interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$ which contains $\widehat{r}_i$. The initial interval is $(0, +\infty]$. The interval is contracted through a series of stages. We will argue that each stage runs in $O(n \log n)$ expected time and requires $O(n)$ space, and that the number of stages in the expected case is $O(1)$.

Let us describe the operations performed during a typical stage in greater detail. Let $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$ be the current interval, and assume that $\widehat{r}_i \in (r_{\mathrm{lo}}, r_{\mathrm{hi}}]$. Recall from the previous subsection that in the ith dual plane, the locus of the centers of circles that pass through p_i and whose radius lies in this interval is the annulus, $A_i(r_{\mathrm{lo}}, r_{\mathrm{hi}})$. For each bisector, $b_{i,j}$, in the arrangement, we will maintain three counts: I_j, W_j and O_j, which denote, respectively, the number of intersection points on the bisector, $b_{i,j}$, that lie *inside* the circle $r = r_{\mathrm{lo}}$, *within* the annulus $A_i(r_{\mathrm{lo}}, r_{\mathrm{hi}})$ and *outside* the circle $r = r_{\mathrm{hi}}$. (In terms of the primal plane, these counts correspond to the number of circles that pass through p_i, p_j and one other data point, and whose radius is smaller than (or equal to) r_{lo}, contained in the interval, and greater than r_{hi}.) Depending on the relationship between I_j, $I_j + W_j$ and the median index $\lceil (n-2)/2 \rceil$, we can determine whether the median radius for $b_{i,j}$ is less than (or equal to) r_{lo}, within the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$, or greater than r_{hi}. The bisectors of the arrangement are partitioned accordingly into three subsets $\mathcal{I}$, $\mathcal{W}$ and $\mathcal{O}$, respectively. (These sets will be represented as sets of indices of points.) Let I, W and O denote the respective cardinalities of these sets. Since we assume that $\widehat{r}_i$ lies within $\mathcal{W}$, it follows that $I < \lceil (n-1)/2 \rceil \leqslant I + W$. A bisector is a *candidate* to provide the RM radius $\widehat{r}_i$ if it lies in $\mathcal{W}$. In particular, the candidate whose median radius is of rank $\lceil (n-1)/2 \rceil - I$ yields the desired radius.

As in the Theil–Sen algorithm, we are searching for an element of a given rank in a set, $\mathcal{W}$, but the task is complicated by the fact that the elements of these sets are themselves medians of other sets. To meet the stated complexity, we do not have time to compute all of these sets explicitly, but we will show that it is possible to compute median radii implicitly. Later in Section 3.3 we discuss how this is done. Assuming for now that this can be solved, we present an overview of the algorithm.

For some suitably chosen constant $\beta < 1$ (whose value will be derived in Section 3.3), we randomly sample $O(n^\beta)$ elements from the set $\mathcal{W}$. For each sampled element j (representing the point p_j in the primal plane or the bisector, $b_{i,j}$, in the dual arrangement), we determine the median radius of the bisector. Using the median radii of these sampled bisectors, we can apply Lemma 2.1 to construct a contracted confidence interval $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$ for $\widehat{r}_i$. However, since we cannot be sure that the desired radius is in this interval, we partition the original interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$ into three subintervals (one of which is the newly contracted interval), and then count the number of median radii falling within each subinterval. From

these counts we can determine which subinterval contains the RM radius (expecting it to be $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$). We then continue searching in this subinterval.

We note that instead of computing the median radius for each of the n^β sampled bisectors (step (2b) below), one could compute *estimates* of these median radii, analogously to the description in [31]. (The main modification would require sampling of intersections points from each sampled bisector, after which an individual median estimate would be arrived at for each bisector. Otherwise, the algorithm would remain intact.) Although the latter is theoretically less efficient (by an additional $\log n$ factor [31]), it is likely to be more attractive from a practical standpoint, as it avoids dealing with the sophisticated data structures that are required for the (theoretically) improved version. (See Section 3.3, for specific details.) In any case, the detailed procedure for the ith RM radius estimate $\widehat{r}_i$ can now be given.

Algorithm 2 (RM circular arc radius estimator for p_i).

(1) Set the initial interval to $(r_{\mathrm{lo}} = 0, r_{\mathrm{hi}} = +\infty]$. Initialize counts $I_j := O_j := 0$, $W_j := n-2$, for all $j \neq i$. Initialize sets $\mathcal{I} := \mathcal{O} := \emptyset$ and $\mathcal{W} := \{j,\ j \neq i\}$. Initialize counts $I := O := 0$ and $W := n-1$.

(2) Repeat the following steps until $W = 1$ (or more practically, until $\sum_{j\in\mathcal{W}} W_j = \mathrm{O}(n)$, after which brute-force enumeration can be used).

(2a) Let β be a constant $(0 < \beta < 1)$, whose value will be given later in Section 3.3. Set $m := \lceil (n-1)^\beta \rceil$, and sample m bisectors $b_{i,j}$ from $\mathcal{W}$ randomly, with replacement.

(2b) Using the method described in Section 3.3 for each sampled bisector, compute its median radius with respect to p_i. These radii will all lie in the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$.

(2c) Let $k := \lceil (n-1)/2 \rceil - I$, that is, k is the rank of $\widehat{r}_i$ among the median radii of the elements of $\mathcal{W}$. Applying Lemma 2.1, let

$$k_{\mathrm{lo}} := \max\left(1, \left\lfloor \frac{mk}{W} - \frac{3\sqrt{m}}{2} \right\rfloor\right), \qquad k_{\mathrm{hi}} := \min\left(m, \left\lceil \frac{mk}{W} + \frac{3\sqrt{m}}{2} \right\rceil\right).$$

Employ any fast selection algorithm to determine the elements r'_{lo} and r'_{hi} of the respective ranks k_{lo} and k_{hi} from the sampled median radii. (We expect the median radius to lie in the interval $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$.)

(2d) Partition the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$ into three disjoint subintervals $(r_{\mathrm{lo}}, r'_{\mathrm{lo}}]$, $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$ and $(r'_{\mathrm{hi}}, r_{\mathrm{hi}}]$. (As in the Theil–Sen case, we treat each annulus/interval as if it is open on the left and closed on the right, so that no intersection lies in more than one annulus/interval.) Using the method to be described in Section 3.1, for each bisector $j \in \mathcal{W}$, count the numbers of intersections on $b_{i,j}$ lying in each of the associated annuli $A_i(r_{\mathrm{lo}}, r'_{\mathrm{lo}})$, $A_i(r'_{\mathrm{lo}}, r'_{\mathrm{hi}})$ and $A_i(r'_{\mathrm{hi}}, r_{\mathrm{hi}})$. (As noted in the Theil–Sen case, only two of these counts need be computed, since the third count can be inferred from the other two.) Based on these counts and the value of I_j, determine, for each $j \in \mathcal{W}$, whether its median radius lies within the first, second or third subinterval.

(2e) From this information, determine which of the three subintervals contains the median radius. Make this the new interval. Update the counts I_j, W_j and O_j, for each $j \in \mathcal{W}$ and update the sets $\mathcal{I}$, $\mathcal{W}$, $\mathcal{O}$ and their respective cardinalities I, W and O accordingly.

To derive the running time of the above algorithm we first consider the number of stages required until the algorithm terminates. According to Lemma 2.1, with probability $1 - 1/\Omega(n^{\beta/2})$, the repeated median is contained within the interval $(r'_{\mathrm{lo}}, r'_{\mathrm{hi}}]$, and so W is reduced by a factor of $1/\Omega(n^{\beta/2})$. If we repeat this process t times, the number of candidate lines decreases by a factor of $1/\Omega(n^{t\beta/2})$. Since we began

with $n-1$ candidates, after a constant number of stages, $t \in O(2/\beta)$, we will have satisfied, with high probability, the termination condition given in step (2).

It is easy to verify that each of the steps of the algorithm can be performed in $O(n)$ time, except for the subtasks of intersection counting (and sampling) and computing median radii. We will show later that each of these tasks can be performed in $O(n \log n)$ time and $O(n)$ space. Combining this with the fact that the expected number of stages is a constant, in $O(n \log n)$ expected time we can compute $\widehat{r}_i$ for a given i. Thus, in total $O(n^2 \log n)$ expected time (together with an additional $O(n)$ time to compute the medians of the $\widehat{r}_i$), we can compute the RM radius. As mentioned before, the minor modifications for computing $\widehat{a}$ and $\widehat{b}$ can be performed within the same time and space bounds. Therefore, we have the following result.

Theorem 2.2. *The non-hierarchical RM circular arc estimator can be computed in* $O(n^2 \log n)$ *expected time and* $O(n)$ *space.*

3. Building blocks

The algorithms presented in the last two sections assumed the existence of routines for performing the basic counting and sampling subtasks, as well as finding median radii, which were omitted from the descriptions of the algorithms. In this subsection we present these basic building blocks. Our presentation will be somewhat more general than needed, in order to accommodate the requirements of the ellipse estimator (to be presented in Section 4) and possibly other curve estimators.

3.1. Intersection counting

Recall that one of the principal building blocks in the algorithms presented earlier was that of counting the number of intersection points of a line arrangement lying within a given annulus. This problem is a variation of the problem (considered, for example, in [12,31]) of counting the number of intersection points of a line arrangement lying within a vertical strip. We present a simple solution to a generalization of this problem. In particular, we assume that we are given an arrangement of *pseudolines*, that is a system of planar curves satisfying the properties that any pair of curves intersects in at most one point, and that curves intersect transversally (that is, the curves cross one another as opposed to intersecting tangentially). To simplify the presentation, we make the assumption that curves are in general position, so that no three curves intersect in a single point. We show that given n pseudolines and a closed region of the plane with a connected boundary, it is possible to compute the intersections on each pseudoline that occur within the region in $O(n \log n)$ time, provided that the following *boundary intersection properties* hold:

(i) each pseudoline intersects the boundary of this region an even number of times,
(ii) the number of intersections between a pseudoline and the boundary is bounded above by some constant, and
(iii) the intersections of pseudolines along the region's boundary can be cyclically sorted in $O(n \log n)$ time (see Fig. 3(a)).

If the region's boundary is not connected but consists of a constant number of connected components (as is the case with an annulus), one or more pseudoline segments can be added to form a *channel*

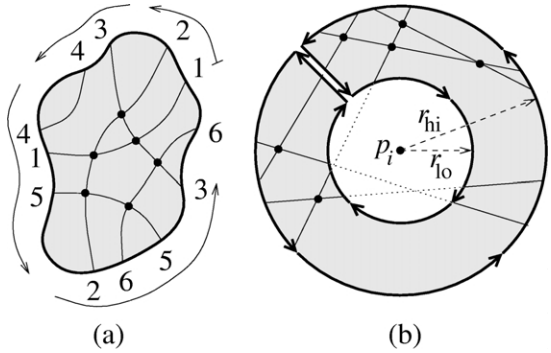

Fig. 3. (a) Generalized intersection/inversion counting using the modified stack mechanism described below ($2 + 2 + 1 + 1 = 6$ pseudoline intersections are reported). (b) Its application to CAE by introducing a "channel", so as to transform the annulus to a connected, bounded region.

connecting the parts of the boundary (as shown in Fig. 3(b)). Note that an $O(n^2)$ solution would be trivial, since this is the maximum number of pairwise intersections possible.

The solution is quite simple, and is based on a generalization of inversion counting (see [12,31]). First, for each pseudoline we determine its intersections with the boundary of the region. If a pseudoline's intersection with a region is not connected, then we break the pseudoline into a collection of connected *pseudosegments* (or just *segments* for short). The intersections between segments will be counted individually for each segment and later added together. We sort the $O(n)$ endpoints of the segments cyclically along the boundary of the region. Each segment has two entries in this list, one for each endpoint. Let us assume that the segments have been indexed according to the appearance of their first endpoint in this order.

We first describe a procedure which counts for each segment the number of intersections with segments of higher index. We maintain a type of stack into which we can insert elements only at the top, but we can remove elements from any position. Initially the stack is empty. We process the elements of the list in the following manner. When a segment endpoint is encountered in the list, if this is the first endpoint encountered for this segment, then it is pushed onto the top of the stack. Otherwise, if this is the second endpoint, we locate the entry for this segment in the stack, count the number of entries lying above it on the stack, and remove this segment from the stack. The count associated with each segment is the desired number of intersections.

To see the correctness of this procedure, observe that when a segment s_i is removed from the stack, the stack entries on top of this segment correspond to segments s_j, $j > i$, whose first intersection with the boundary was encountered between the two occurrences of s_i's endpoints, but whose second endpoint has yet to be encountered. Thus the order of intersections of these segments along the boundary alternates as follows:

$$\ldots s_i \ldots s_j \ldots s_i \ldots s_j \ldots$$

Because pseudoline segments start and end on the boundary, it follows that s_i and s_j intersect at least once (and hence exactly once) within the region. Furthermore, from the properties of pseudolines it is easy to see that all intersections will be counted in this manner.

Recall that we have only counted intersections with segments of higher index. For the Theil–Sen estimator, where we need a total count of all pseudoline intersections occurring within the region, the above procedure suffices, since each intersection is counted exactly once (by the segment of lower index). For the purpose of the repeated median estimator, it is important to compute intersection counts individually for each segment. By running the procedure twice, first counterclockwise and then clockwise around the boundary, and then summing the two counts for each segment, we will have counted all of the segment's intersections.

The stack can be implemented by a simple modification of virtually any type of a balanced binary search tree, for example, a red–black tree (see, e.g., [21] or [11, Chapters 14 and 15]). The tree is modified for the purposes of counting in the following manner. The segments that are currently on the stack are stored in the leaves of the tree. They are ordered so that the top/bottom of the stack is the leftmost/rightmost leaf of the tree. Unlike a normal binary search tree, where elements are accessed in a top-down manner using key values, each segment is associated with a *finger pointer* to the leaf of the tree containing this segment. If the segment does not appear in the tree, the finger pointer is null. The tree is augmented with parent links so that the path from each leaf to its ancestors can be traversed in a bottom-up manner. Each internal node has an additional field containing the number of leaves in the left subtree rooted at this node. It is a simple programming exercise to augment the procedures for virtually any type of balanced binary tree to maintain this additional information.

Pushing a segment s_i onto the stack is performed by first finding the top of the stack, that is, the leftmost leaf in the tree, and then inserting s_i as the new leftmost leaf in the tree and rebalancing the tree as necessary.

To delete a segment s_i from the stack, the finger pointer for this segment is accessed to locate the corresponding leaf in the tree. To count the number of elements lying above of s_i in the stack (that is, to the left of s_i in the tree), the path from this leaf to the root is traversed. Whenever this path travels to a parent node from its right child, the leaf count for the left child is added to the count. It is easy to see that this will count all leaves lying to the left of s_i. The leaf containing s_i is then deleted from the tree, and the tree is rebalanced.

For example, in Fig. 3(a), suppose we start the traversal in counterclockwise order starting at the rightmost endpoint of segment 1. The algorithm pushes, in turn, 1, 2, 3 and 4. On encountering the second endpoint of 4 no intersections are reported, since 4 is already on top of the stack. Segment 4 is then removed from the stack. On seeing the second endpoint of segment 1, the leaves of the tree are (from left to right) 3, 2, 1. The removal of leaf 1 counts its two intersections with segments 3 and 2. After this, 5 is pushed onto the stack, and then 2 is removed. At the time of the removal of 2, the leaves of the tree are 5, 3, 2, and the removal of 2 counts its two intersections with 5 and 3 (the intersection with 1 was already counted). The algorithm continues in this fashion, and returns a total count of 6 intersections.

Both insertion and deletion can be performed in $O(\log n)$ time, from standard results on (balanced) binary trees. Since a total of $O(n)$ segment endpoints are processed, we have the following result.

Lemma 3.1. *Consider a set of n pseudolines in the plane, and a closed planar region satisfying the boundary intersection properties given above. Then in $O(n \log n)$ time and $O(n)$ space, it is possible to compute, for each pseudoline, the number of intersections between it and the other pseudolines that occur within this region.*

3.2. Intersection sampling

The intersection sampling problem is that of randomly sampling (with replacement), say, m intersection points of an arrangement of pseudolines that lie within a given region of the plane. (Recall, this procedure is required for step (2a) of Algorithm 1, and may be required for step (2b) of Algorithm 2, should median radii be estimated rather than computed.) In the previous subsection we have already discussed how to count these intersection points. We use a standard trick to transform the procedure described for counting into a sampling routine. This trick is generally applicable to a situation where the cardinality of some finite set is arrived at by accumulating a number of positive counts, and where an efficient method exists for associating elements of the set with each increment of the count. Although this method was described in [12], we reintroduce it here in terms of the more general intersection counting algorithm presented in the previous subsection.

It follows from this procedure that the number of intersection points is broken down into a number of cumulative $O(n)$ counts. Let C, c_i denote, respectively, the total count and the incremental count associated with segment s_i. According to the previous subsection, each count, c_i, is the number of entries lying above s_i in the stack, or equivalently, the number of leaves lying to the left of a given leaf s_i in the balanced tree. For each of the leaves s_j (that account for c_i), there is a corresponding intersection point between s_i and s_j that lies within the region. Furthermore, for any k, $1 \leqslant k \leqslant c_i$, in $O(\log n)$ time we can determine the kth leaf (from the left) in this tree, using a simple tree search based on the leaf counts stored in the nodes of the tree.

To sample an arbitrary set of m intersection points, we first generate a random sample of m integers in the range from 1 to C. (To determine C, we apply the counting procedure first.) We sort these integers, letting $E = (e_1 \leqslant e_2 \leqslant \cdots \leqslant e_m)$ denote the sorted indices to be sampled. Intuitively, for each e_j, when the intersection counter is incremented to a value $\geqslant e_j$, we sample the corresponding intersection. In particular, we reapply the counting procedure, but with the following modification. As the counter is incremented from, say, C' to $C' + c_i$, we determine all the elements $e_j \in E$, such that $C' < e_j \leqslant C' + c_i$. (This can be done easily by maintaining the index of the most recently accessed element of S.) For each element e_j, we select (from the left) the leaf of rank $e_j - C'$ in the tree, and determine the corresponding intersection point as explained in the previous paragraph. Since each of the m samples can be computed in $O(\log n)$ time, and since the counting procedure takes $O(n \log n)$ time, the entire procedure runs in $O((n + m) \log n)$ time.

Lemma 3.2. *Given the same hypotheses of Lemma* 3.1, *a sample of m intersection points can be computed in* $O((n + m) \log n)$ *time.*

Since the number of sampled elements m is at most n (in both invocations of this procedure, in the RM case), it follows that the total time to perform sampling is $O(n \log n)$.

3.3. Finding median radii

In this subsection we describe how to determine the median radius for the sampled bisectors in the arrangement (as needed for step (2b) of the RM algorithm). Recall that there is a fixed point p_i, and an arrangement of $n - 1$ bisectors $b_{i,j}$ for each $j \neq i$, and that each intersection point in this arrangement is the center of a circle passing through p_i and two other data points. The radius of this circle is associated

with each intersection point. We want to compute the *median radius* of a bisector $b_{i,j}$, which is defined to be the median radius value among intersection points on this bisector. It will be somewhat easier to describe this problem in terms of the primal plane. The corresponding problem is as follows. Given a set of $n-2$ circles passing through two fixed points p_i and p_j and any third data point, return the median radius over this set.

Recall that we need to solve this problem for a given a set of $\lceil (n-1)^\beta \rceil$ bisectors (where the value $\beta < 1$ was left unspecified). Using results from the theory of *range searching* [8,16,23,30,49], we will show that there exists $\gamma < 1$ (depending on the complexity of the range searching algorithms), such that, after $O(n \log n)$ preprocessing (which will be common to all of the bisectors), the median radius for each bisector can be computed in $O(n^\gamma \log n)$ expected time. Thus the total expected running time will be $O(n \log n + n^\beta n^\gamma \log n)$. We set $\beta = 1 - \gamma$, so the total expected running time will be $O(n \log n)$.

Our algorithm is a modification of the randomized binary search presented in [31], where each probe of the binary search will require $O(n^\gamma)$ time. Consider the circles passing through p_i and p_j. We maintain an interval $(r_{\text{lo}}, r_{\text{hi}}]$ which will bound the median radius at all times. (This interval is independent of the radius interval used in the repeated median algorithm.) The initial radius is $(0, +\infty]$. The search implicitly partitions the set of circles into three subsets: those whose radius is less than (or equal to) r_{lo}, whose radius lies within the interval, and those whose radius is greater than r_{hi}. Three counts are maintained, one for each subset. Let us assume for now that given any radius interval, it is possible to count the number of circles whose radius value lies within the interval, and furthermore, that it is possible to return a random element from this set. (These will be explained later.) Each probe of the randomized binary search begins by sampling a random circle whose radius value r_{mid} lies within the interval. We then partition the interval $(r_{\text{lo}}, r_{\text{hi}}]$ into two subintervals $(r_{\text{lo}}, r_{\text{mid}}]$ and $(r_{\text{mid}}, r_{\text{hi}}]$. Finally, we count the number of circles associated with each interval. (It is sufficient to count within one interval, since the other count can be derived from the latter.) Using these counts, and based on the number of radius values that are smaller than (or equal to) r_{lo}, we can determine which subinterval contains the median radius, update the counts, and then recurse on the right subinterval. The algorithm terminates when the interval is associated with a single circle.

Since each circle is chosen at random from among the circles whose radius value lies within the current interval, it follows from a simple probabilistic argument that at least 1/4 of the remaining intersection points are expected to be eliminated with each probe. Thus, based on properties of a Bernoulli trial, it can be shown that the expected number of probes that will take the search to terminate is $O(\log n)$. All that remains is to describe how to perform counting and sampling (in the primal domain). We show that these tasks can be reduced to the problem of a range counting query in the plane. This is illustrated in Fig. 4.

Lemma 3.3. *Given two data points p_i and p_j and an interval $(r_{\text{lo}}, r_{\text{hi}}]$, the circles passing through these two data points and any other third data point, and whose radius is in the interval, are in $1-1$ correspondence with the set of data points lying within the union of two regions of the plane, each being the symmetric difference of two circular disks.*

Proof. Clearly, the centers of circles passing through p_i and p_j lie on the bisector, $b_{i,j}$, and can be viewed as elements of a linearly ordered set. Associated with each element of this set (i.e., with each center point on the bisector) is a unique radius. As a center point travels along the bisector, it is easy to see that the radius of the corresponding circle is a unimodal function, attaining its minimum at the midpoint between

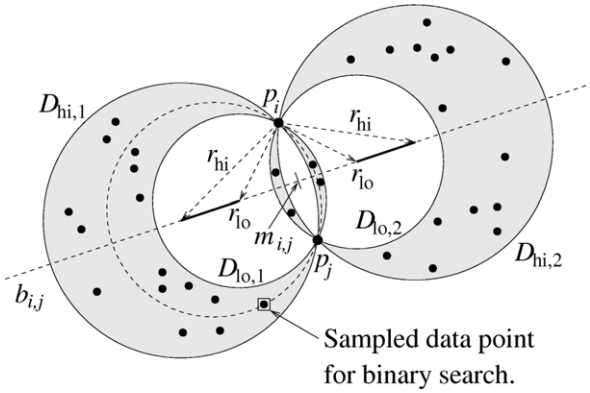

Fig. 4. The set of circles that pass through p_i and p_j and whose radius is contained in the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$ lie within the shaded region.

p_i and p_j (denoted by $m_{i,j}$), and growing monotonically to $+\infty$ toward either extreme. Thus, for a given radius interval, $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$, any circle passing through p_i and p_j and whose radius lies within this interval has its center lying within one of two segments on the bisector $b_{i,j}$, each residing on opposite sides of $m_{i,j}$. (These segments are shown as thick lines in Fig. 4.) If r_{lo} is less than half the distance between p_i and p_j, then these two segments share a common endpoint at $m_{i,j}$.

Associated with the endpoints of these two segments are four closed circular disks $D_{\mathrm{lo},1}$, $D_{\mathrm{hi},1}$ and $D_{\mathrm{lo},2}$, $D_{\mathrm{hi},2}$, where $D_{\mathrm{lo},i}$ has radius r_{lo} and $D_{\mathrm{hi},i}$ has radius r_{hi} $(i = 1, 2)$. Let S denote the union of the symmetric differences of $D_{\mathrm{lo},i}$ and $D_{\mathrm{hi},i}$, for $i = 1, 2$. (Observe that p_i and p_j are not in S, since they belong to both disks, for $i = 1, 2$.) It is an easy geometric exercise to verify that for any circle passing through p_i and p_j, if its radius lies within the interval $(r_{\mathrm{lo}}, r_{\mathrm{hi}}]$, then its boundary lies entirely within S (except at p_i and p_j). Otherwise, its boundary is entirely disjoint from S. It follows immediately that the data points lying within S are in 1–1 correspondence with the circles passing through p_i, p_j and one other data point. □

This lemma provides us with a way to reduce the task(s) of counting (sampling) circles that pass through p_i and p_j to a range counting query over the set of data points, where the range can be described in terms of a constant number of Boolean operations on circles. Agarwal and Matoušek [1] have shown that there exists $\gamma < 1$, such that these queries can be solved in $O(n^\gamma)$ time and $O(n)$ space after $O(n \log n)$ preprocessing. Because the result of the range query can be identified as the disjoint union of $O(n^\gamma)$ leaves in a balanced tree, it follows that it is possible to apply the methods of the previous subsection to sample one such leaf at random. Note that preprocessing need only be applied once to the set of data points.

The modifications needed for the other circle parameters, namely $\widehat{a}$ and $\widehat{b}$, are quite simple. In particular, it is easy to show that the circles passing through p_i, p_j and any other third data point have their center lying within a given vertical or horizontal strip. Thus, finding the median values of these parameters can be reduced to a range query over a single symmetric difference of two circular disks.

4. An extension to fitting aligned ellipses

In this section we consider the more general problem of robust fitting of an ellipse whose axes are aligned with the coordinate axes to a set of points in the plane. We call such an ellipse an *aligned ellipse*. We assume that the ellipse is presented in the form

$$\frac{(x-a)^2}{A^2} + \frac{(y-b)^2}{B^2} = 1.$$

The point (a, b) is the *center* of the ellipse, and the values A and B are the *horizontal* and *vertical radii* of the ellipse, respectively. Aligned ellipses are of interest, for example, in applications such as automated document processing, where they are frequently used in charts and diagrams. One of the reasons for considering aligned ellipses is that they have fewer degrees of freedom than general ellipses (4 versus 5), which leads to a reduction in the asymptotic complexity of computing the corresponding estimators.

The higher level structure of the randomized algorithms for circular arc fitting presented in Sections 2.1 and 2.2 can be generalized readily to the ellipse problem. Most of this section is devoted to establishing the lower-level geometric details involved with the dual transformation, intersection counting and sampling in arrangements, and range searching. The main difficulty in extending the method from circles to ellipses is that the intuitive geometric arguments, which sufficed in the case of circles, lead to significantly longer and more technical algebraic arguments in the case of the ellipse. Furthermore, because of the larger number of degenerate and special cases of points, it will greatly simplify the presentation to make the following *general position assumptions* throughout this section:

(i) no two points share the same x-coordinate or same y-coordinate,
(ii) no three points are collinear,
(iii) there is at most one aligned ellipse passing through any four points, and
(iv) no four points determine an ellipse whose center is the midpoint of two of the four points.

First, let us consider the generalizations of the Theil–Sen and repeated median estimators to the case of aligned ellipses. In contrast to the circular case, where three points uniquely determine a circle, four points are generally needed to determine a unique aligned ellipse. However, not all quadruples of points define an aligned ellipse (e.g., if the points are not in convex position). A quadruple (i, j, k, l), $1 \leqslant i \neq j \neq k \neq l \leqslant n$, is *feasible* if there exists an aligned ellipse passing through the corresponding points. From Assumption (iii) it follows that every quadruple of data points determines at most one aligned ellipse. A k-tuple of points, for $k < 4$, is *feasible* if there is an extension to a feasible quadruple. Given a feasible quadruple of points, define $a_{i,j,k,l}$, $b_{i,j,k,l}$, $A_{i,j,k,l}$ and $B_{i,j,k,l}$ to be the parameters of the aligned ellipse passing through these points. Our estimators return the center coordinates and horizontal and vertical radii of the ellipse, that is, the quadruple $(\widehat{a}, \widehat{b}, \widehat{A}, \widehat{B})$ corresponding to the coefficients of the above ellipse equation. Generalizing the linear and circular cases, aligned ellipse estimators can be defined formally as follows.

Theil–Sen aligned ellipse estimator: Computes for each feasible quadruple (i, j, k, l), $1 \leqslant i < j < k < l \leqslant n$, the parameters of the corresponding ellipse. The estimator is given by the median values, over (up to) $\binom{n}{4}$ elements, of each of the above sets, i.e.,

$$\widehat{a} = \operatorname{med} a_{i,j,k,l}, \qquad \widehat{b} = \operatorname{med} b_{i,j,k,l},$$

and

$$\widehat{A} = \operatorname{med} A_{i,j,k,l}, \qquad \widehat{B} = \operatorname{med} B_{i,j,k,l}.$$

RM aligned ellipse estimator: Computes for each feasible quadruple (i, j, k, l), $1 \leqslant i \neq j \neq k \neq l \leqslant n$, the parameters of the corresponding ellipse. The center's estimated coordinates are given by

$$\widehat{a} = \operatorname*{med}_{i} \operatorname*{med}_{j\neq i} \operatorname*{med}_{k\neq i,j} \operatorname*{med}_{l\neq i,j,k} a_{i,j,k,l}, \qquad \widehat{b} = \operatorname*{med}_{i} \operatorname*{med}_{j\neq i} \operatorname*{med}_{k\neq i,j} \operatorname*{med}_{l\neq i,j,k} b_{i,j,k,l},$$

and the estimated radii are given by

$$\widehat{A} = \operatorname*{med}_{i} \operatorname*{med}_{j\neq i} \operatorname*{med}_{k\neq i,j} \operatorname*{med}_{l\neq i,j,k} A_{i,j,k,l}, \qquad \widehat{B} = \operatorname*{med}_{i} \operatorname*{med}_{j\neq i} \operatorname*{med}_{k\neq i,j} \operatorname*{med}_{l\neq i,j,k} B_{i,j,k,l}.$$

This definition is understood to involve only singletons, pairs and triples that are feasible.

Somewhat more intuitively, each parameter of the Theil–Sen (aligned) ellipse estimator is defined by considering all feasible quadruples of points, computing the ellipse passing through each quadruple, and then selecting the median value of the corresponding parameter over all these ellipses. For the repeated median estimator, each feasible quadruple (i, j, k, l) determines a single ellipse, and hence a unique parameter value. For each triplet (i, j, k), we take the median parameter over all $n - 1$ choices of the fourth point. For each pair, (i, j), we consider the median over all $n - 2$ choices of a third point, and so on. It follows from standard arguments [38] that because they are based on quadruples of observations, the breakdown points of the Theil–Sen and RM (aligned) ellipse estimators are $1 - \sqrt[4]{1/2} \approx 16\%$ and 50%, respectively.

The Theil–Sen and the RM aligned ellipse estimators can be computed by a brute-force implementation of their definitions in $O(n^4)$ time. The Theil–Sen estimator requires $O(n^4)$ space, whereas the RM requires linear space. We show that both can be computed by randomized algorithms running in $O(n^3 \log n)$ expected time and $O(n)$ space. Perhaps more importantly, we indicate the necessary steps in generalizing the methodology presented in the previous sections to other classes of curves. We focus primarily on the most illustrative case, the RM horizontal radius estimator, $\widehat{A}$. (The basic building blocks for the Theil–Sen case are a subset of those needed for the RM. Regarding the other RM parameters, the vertical radius, $\widehat{B}$, is symmetric to $\widehat{A}$, and the center coordinates $\widehat{a}$ and $\widehat{b}$ are algebraically simpler.)

The general structure of the algorithm is the same as that presented in Section 2. The principal elements that need to be modified are the geometric components used in computing the RM estimator for circles: (1) the *bisector arrangement* in the dual plane, (2) the *annular regions* used in region contraction and (3) the *ranges* used in the computation of median radii. The remainder of this section is organized as follows. First we transform the problem of computing the RM horizontal radius to a somewhat simpler form for the purposes of presentation. In Section 4.1 we provide some technical results on the algebraic structure of aligned ellipses, which will be needed later. In Section 4.2 we show that the bisector arrangement in the circular case generalizes to an arrangement of axis-aligned hyperbolas in the ellipse case. In Section 4.3 we provide a generalization of the annular regions which were used in the circular case, and show that the arrangement of hyperbolas satisfies the boundary intersection properties of Section 3.1 required for intersection point counting and sampling. Finally, in Section 4.4 we show how the ranges used in the circular case for finding median radii, can be generalized to ranges based on the symmetric differences of ellipses. Although some of the algebraic derivations presented in this section are rather tedious, they are relatively straightforward and help illuminate a number of interesting properties of this important class of conics [4].

[4] Some of the longer derivations presented here have been verified with the help of the *Mathematica* software system [50]. Copies of the Mathematica scripts containing these derivations are available from the authors.

We begin by transforming the RM horizontal radius problem into a somewhat simpler form. Recall that in the circular case, three points are needed to determine a circle. We fixed one point, p_i, and reduced the problem to a repeated median calculation over $O(n^2)$ possible remaining pairs. Analogously, in the case of aligned ellipses where four points are needed, we fix two points, p_i and p_j, and reduce the problem to a repeated median calculation over the $O(n^2)$ remaining pairs. We first define

$$\widehat{A}_{i,j} \triangleq \operatorname*{med}_{k\neq i,j} \operatorname*{med}_{l\neq i,j,k} A_{i,j,k,l}.$$

$\widehat{A}_{i,j}$ can be interpreted as a *horizontal radius estimator* for two fixed points, p_i, p_j. (It plays, essentially, the analogous role of $\widehat{r}_i$ in Section 2.2.) In the remainder of the section, we argue that $\widehat{A}_{i,j}$ can be computed in expected $O(n\log n)$ time and $O(n)$ space. Thus, using standard algorithms for computing medians in $O(n)$ time, this leads immediately to an algorithm, for computing $\widehat{A}$, whose expected running time is $O(n^3\log n)$ and which requires $O(n)$ space.

Since the points p_i and p_j will be fixed for the remainder of the discussion, let us denote them simply by $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$. It will simplify subsequent derivations to apply an affine transformation that maps these points to $(-1,-1)$ and $(1,1)$, respectively, and which preserves aligned ellipses. (We hasten to mention that, unlike linearization, we will be able to extract the true values of the estimators as defined earlier.) We first apply a translation to the plane, so that the midpoint of these two points is mapped to the origin. Next, by general position assumption (iii), we know that no two data points share the same x- or y-coordinates, and so we may apply scaling independently to the x- and y-axes by $1/(x_2 - x_1)$ and $1/(y_2 - y_1)$, respectively. Clearly, this maps p_1 and p_2 to $(-1,-1)$ and $(1,1)$, respectively. Both translation and scaling of coordinates preserve aligned ellipses. Furthermore, the transformation either preserves or (if we scale by a negative quantity) entirely reverses the order relationships among ellipse parameters. Therefore, the various parameter medians can be easily extracted from the transformed representations. (For the center coordinates this is done by an inverse translation, and for the horizontal and vertical radii this is done by multiplying by the reciprocal of the scale factor.)

4.1. Some technical results on aligned ellipses

We begin with the following technical result, which describes some of the relationships between the parameters of an aligned ellipse passing through the points p_1 and p_2.

Lemma 4.1. *Consider an ellipse*

$$\frac{(x-a)^2}{A^2} + \frac{(y-b)^2}{B^2} = 1,$$

which passes through $p_1 = (-1,-1)$ *and* $p_2 = (1,1)$. *Then*

(i) $A, B \geqslant 1$,
(ii) $bA^2 = -(1-ab)(a-b)$, *and* $aB^2 = (1-ab)(a-b)$, *and*
(iii) *if the center of the ellipse does not coincide with the origin, then the ellipse is uniquely determined from its center.*

Proof. Fact (i) is immediate from the observations that the ellipse passes through two points (p_1 and p_2) which are separated by horizontal and vertical distances of 2. Hence, the ellipse's horizontal and vertical diameters must be at least this large.

Fact (ii) is proved by a straightforward substitution of the values of $p_1 = (-1, -1)$ and $p_2 = (1, 1)$ into the ellipse equation, yielding

$$\frac{(-1-a)^2}{A^2} + \frac{(-1-b)^2}{B^2} = 1, \qquad \frac{(1-a)^2}{A^2} + \frac{(1-b)^2}{B^2} = 1,$$

and then solving these two equations for A^2 and B^2 as a function of a and b. In particular, since $A^2, B^2 \neq 0$, through simple manipulations we have the two equations

$$B^2\big(A^2 - (1+a)^2\big) = A^2(1+b)^2, \qquad B^2\big(A^2 - (1-a)^2\big) = A^2(1-b)^2.$$

Eliminating B^2/A^2 these yield

$$\big(A^2 - (1+a)^2\big)(1-b)^2 = \big(A^2 - (1-a)^2\big)(1+b)^2,$$

and after some expansion and simplification

$$4bA^2 = -4(1-ab)(a-b).$$

One part of the result follows immediately. The other part follows from the symmetry of a with b and of A with B.

To establish (iii), it is easy to verify that if one of the center's coordinates is zero, then the other must also be zero. If both center coordinates are nonzero, then the values of A and B are determined from these constraints, and hence the ellipse is uniquely determined. □

Notice that when the center coincides with the origin, there are an infinite number of ellipses passing through p_1 and p_2, and a third point is needed to uniquely determine the ellipse. This is why general position assumption (iv) was introduced.

Next, we consider properties of aligned ellipses passing through p_1, p_2 and any other third data point p_k, $k \neq 1, 2$. Excluding degenerate cases, four points are needed to uniquely define an ellipse, so these three points define a 1-dimensional family of aligned ellipses. The most convenient method to describe this family of ellipses will be to introduce a parameterization based on the ratio of the height to width of these ellipses.

Let ρ be a any positive real, and let $T[\rho]$ denote the linear transformation,

$$T[\rho](x, y) = (\rho x, y),$$

which scales the x axis by a factor of ρ. There is a unique circle passing through the transformed points $T[\rho](p_1)$, $T[\rho](p_2)$ and $T[\rho](p_k)$. Applying the inverse transformation $T[1/\rho]$ to this circle produces a unique aligned ellipse $E[\rho]$, see Fig. 5. The ratio of the height to width of the resulting ellipse is clearly ρ. The following fundamental lemma provides a parametric description of each of the ellipse parameters as a function of ρ.

Lemma 4.2. *Consider points $p_1 = (-1, -1)$, $p_2 = (1, 1)$ and $p_k = (s, t)$. The points (x, y) on the ellipse $E[\rho]$ passing through these points satisfy*

$$\frac{(x - a[\rho])^2}{A^2[\rho]} + \frac{(y - b[\rho])^2}{B^2[\rho]} = 1,$$

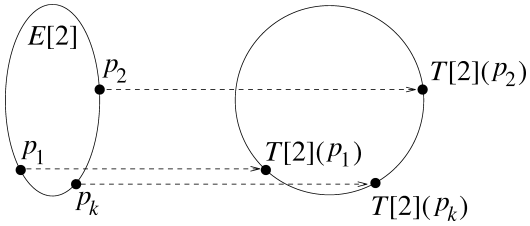

Fig. 5. Ellipse parameterization, where the horizontal axis has been scaled by a factor of $\rho = 2$.

where

$$a[\rho] = -\frac{\rho^2(1-s^2)+(1-t^2)}{2\rho^2(s-t)}, \qquad b[\rho] = \frac{\rho^2(1-s^2)+(1-t^2)}{2(s-t)},$$
$$A^2[\rho] = \frac{M(\rho,s,t)}{4\rho^4(s-t)^2}, \qquad B^2[\rho] = \frac{M(\rho,s,t)}{4\rho^2(s-t)^2},$$

and where

$$M(\rho,s,t) = \big(1+\rho^2\big)\big(\rho^2(1-s)^2+(1-t)^2\big)\big(\rho^2(1+s)^2+(1+t)^2\big).$$

Proof. Substituting the values $a[\rho]$, $b[\rho]$, $A^2[\rho]$ and $B^2[\rho]$, into the ellipse equation, and then applying straightforward simplifications yields

$$M(\rho,s,t) = \big(2x\rho^2(s-t)+\rho^2\big(1-s^2\big)+\big(1-t^2\big)\big)^2+\rho^2\big(2y(s-t)-\rho^2\big(1-s^2\big)-\big(1-t^2\big)\big)^2.$$

After a lengthy expansion, we collect common terms in ρ. The expanded equation has the form $\rho^4 N(x,y,s,t)+\rho^2 N(y,x,t,s)=0$, where

$$N(x,y,s,t) = 4(s-t)\big((s-t)-x^2(s-t)-x\big(1-s^2\big)+y\big(1-s^2\big)\big).$$

After removing common factors and simplifying, we have

$$\rho^2\big((s-t)\big(1-x^2\big)+\big(1-s^2\big)(y-x)\big)+\big((s-t)\big(1-y^2\big)+\big(1-t^2\big)(y-x)\big)=0. \tag{1}$$

The conclusion follows by substituting the coordinates of each of the points p_1, p_2 and p_k (in place of x and y) into this equation, to show that these points satisfy the equation, irrespective of ρ. □

An important feature of the family of aligned ellipses passing through three given points is that all the ellipses in the family share a fourth point in common and exhibit an interesting manner of nesting. This fact is presented in the following lemma, and is illustrated in Fig. 6.

Lemma 4.3. *Consider the set of aligned ellipses $E[\rho]$, $\rho > 0$, that pass through the three noncollinear points $p_1 = (-1,-1)$, $p_2 = (1,1)$ and $p_k = (s,t)$.*

(i) *All these ellipses share a common fourth point $q_k = (S,T)$, where*

$$S = \frac{s^2+st-2}{s-t} \quad \text{and} \quad T = -\frac{t^2+st-2}{s-t}.$$

(ii) *Define an elliptical disk to be the closed 2-D region bounded by an ellipse. Given $0 < \rho' < \rho''$, for each $\rho > 0$, if $\rho' < \rho < \rho''$ then (except at the four common points of intersection) the ellipse $E[\rho]$*

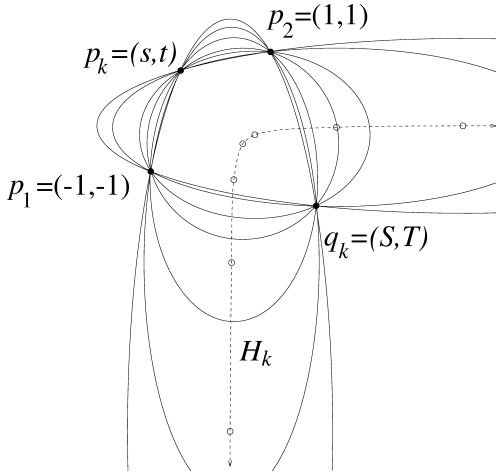

Fig. 6. Aligned ellipses passing through three points share a common fourth point.

lies entirely within the symmetric difference of the elliptical disks defined by $E[\rho']$ *and* $E[\rho'']$*, and is otherwise entirely disjoint from of this region.*

Proof. The proof of (i) is by substitution of the point (S, T) as (x, y) into Eq. (1), followed by straightforward manipulations to show that the equation holds, irrespective of ρ.

To prove (ii), we know from Lemma 4.2 that as ρ varies continuously from ρ' to ρ'', the corresponding family of ellipses $E[\rho]$ sweeps continuously over some region of the plane, starting with $E[\rho']$ and ending at $E[\rho'']$. By simple continuity and connectivity of these curves, it follows that the locus of points encountered by the sweep contains the symmetric difference of the corresponding elliptical regions.

We claim that (other than the four common points) no point outside the symmetric difference can be encountered by the sweep. First note that distinct ρ values generate distinct ellipses. From the intermediate value theorem, visiting a point outside the symmetric difference would imply that for some $\rho \in (\rho', \rho'')$ there is an intersection between $E[\rho]$ and either $E[\rho']$ or $E[\rho'']$ (in addition to the four common points). However, all of the ellipses in this family are distinct polynomial curves of order 2. Hence by Bézout's theorem [39], there can be no more than four intersection points between any two of them. □

4.2. The dual arrangement of hyperbolas

Recall that in the circular case, we considered a dual transformation in which each point in the dual plane was associated with the center of a unique circle. In the elliptical case we can define an analogous *dual plane* associated with the points p_1 and p_2, such that an ellipse can be uniquely identified with each point contained in a subset of points in this plane. In particular, given the coordinates (a, b) of a center point (other than the origin), we know from Lemma 4.1(iii) that there is at most one ellipse passing through p_1 and p_2 having these center coordinates. Unlike the circular case, not all points of the dual plane will be centers of aligned ellipses, because the values of A^2 and B^2 as determined from Lemma 4.1(ii) must both be positive. However, our algorithm will never encounter such "infeasible" center points.

Since four points are generally needed to define an ellipse, if we consider three distinct data points p_1, p_2 and p_k, there is a 1-dimensional family of ellipses that pass through these three points. We claim that the centers of these ellipses form a branch of a hyperbola, denoted by H_k. (This is the analogue of the bisector $b_{i,j}$ introduced earlier in Section 2.) This hyperbola is shown as a dashed curve in Fig. 6. This observation is a special case of the more general lemma, which states that the locus of the centers of 2-D conics that pass through four fixed points is a 2-D conic whose asymptotes are parallel to the axes of the two parabolas through the four points. (For a proof, see, e.g., [43, pp. 292–293].)

Lemma 4.4. *Given distinct points $p_1 = (-1, -1)$, $p_2 = (1, 1)$ and $p_k = (s, t)$, the locus H_k of the centers of ellipses that pass through these three points is one branch of a hyperbola with horizontal and vertical asymptotes. In particular,*

$$H_k \subseteq \{(a, b) \mid (a - a_0)(b - b_0) = K\},$$

where

$$a_0 = \frac{-(1 - s^2)}{2(s - t)}, \quad b_0 = \frac{(1 - t^2)}{2(s - t)} \quad \text{and} \quad K = a_0 b_0.$$

The asymptotes of the hyperbola are $x = a_0$ and $y = b_0$.

Proof. First we show that the set of centers satisfies the hyperbola equation presented above. Observe that the hyperbola equation above can be rewritten as

$$2ab(s - t) - a(1 - t^2) + b(1 - s^2) = 0.$$

The result follows by substituting the ellipse coefficients $a[\rho]$ and $b[\rho]$ (presented in Lemma 4.2) for a and b, respectively, into the hyperbola equation, and then verifying that the rewritten equation holds, irrespective of ρ.

To show that H_k consists of only one branch of the hyperbola, recall the transformation $T[\rho]$ introduced earlier. It suffices to show that as ρ varies continuously from 0 to ∞, the centers of the respective ellipses $E[\rho]$ trace out one branch of the hyperbola. As $\rho \to 0$ it is easy to verify from Lemma 4.2(i) that the a-coordinate of the ellipse, $a[\rho]$, approaches either $+\infty$ or $-\infty$, and as $\rho \to \infty$, the b-coordinate of the center, $b[\rho]$, approaches either $+\infty$ or $-\infty$. For all values between these extremes the center varies continuously in the plane, implying that exactly one branch is traced out. □

Observe that the hyperbola described in the previous lemma passes through the origin. In general, the branch of interest may or may not pass through the origin. This branch can be determined easily from the values of the coordinates of p_k, relative to those of p_1 and p_2.

Since H_k denotes the locus of the centers of ellipses that pass through p_1, p_2 and p_k, if we consider $k = 3, 4, \ldots, n$, we have an arrangement of $n - 2$ curves in the plane. Observe that if two such curves H_k and H_ℓ [5] intersect at some point (other than the origin), then from Lemma 4.1(iii) there is a unique aligned ellipse whose center coincides with these points, and hence this ellipse passes through all four points, p_1, p_2, p_k and p_ℓ. This center cannot coincide with the origin, because from general position assumption (iv), the midpoint of p_1 and p_2 cannot be the center of any ellipse. Conversely, if there is an (aligned) ellipse that passes through four data points, then the hyperbolas H_k and H_ℓ must intersect at the

[5] H_ℓ denotes the locus of the centers of ellipses that pass through p_1, p_2 and p_ℓ.

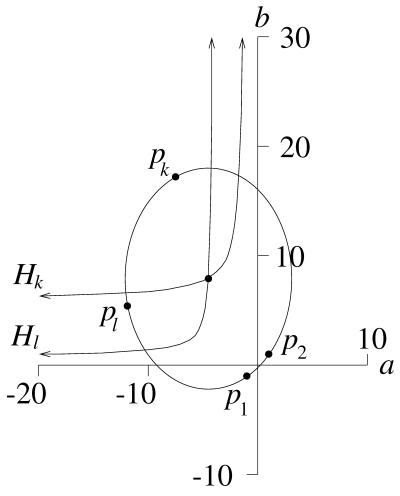

Fig. 7. Hyperbola intersections and ellipse centers.

ellipse's center. From general position assumption (iii), there is only one such ellipse. This implies that the arrangement of hyperbolas is a pseudoline arrangement analogous to the arrangement of bisectors in the circular arc case. This is illustrated in Fig. 7, and presented in the following lemma.

Lemma 4.5. *Given points p_1 and p_2, the arrangement of hyperbolas H_k, $3 \leqslant k \leqslant n$, is a pseudoline arrangement whose vertices, other than the origin, are in $1-1$ correspondence with the centers of aligned ellipses that pass through p_1, p_2 and any two other data points.*

4.3. Region contraction for the ellipse estimator

Each vertex in the arrangement of hyperbolas is associated with a unique aligned ellipse, and hence is associated with the four coefficients defining this ellipse. For a given curve H_k, define its *median horizontal radius* to be the median horizontal radius among the $O(n)$ arrangement vertices lying on this curve. (In the primal plane, this is the median horizontal radius among the $O(n)$ aligned ellipses that pass through p_1, p_2, p_k and any other fourth data point.) Thus, for the quantity of interest, $\widehat{A}_{1,2}$, the problem has been reduced to computing the median over the $O(n)$ median horizontal radii.

To compute $\widehat{A}_{1,2}$, we employ a similar *region contraction* scheme as the annulus contraction which was described in the circular case. From Lemma 4.1(ii) it follows that each point in the dual plane, which is the center of some ellipse passing through p_1 and p_2, can be associated with a unique horizontal radius,

$$A(a,b) = \sqrt{\frac{-(1-ab)(a-b)}{b}}.$$

(Points (a,b) for which this quantity is undefined are not centers of aligned ellipses. This will not be a problem because our algorithm will only evaluate the above expression for vertices of the hyperbola arrangement, and all of these points are centers of ellipses.)

For $0 < A_{\mathrm{lo}} < A_{\mathrm{hi}}$, we can associate any (half-open, half-closed) interval, $(A_{\mathrm{lo}}, A_{\mathrm{hi}}]$, with a region $R(A_{\mathrm{lo}}, A_{\mathrm{hi}})$ of the plane, such that, for all points (a,b) in this region, $A(a,b)$ is defined and lies within

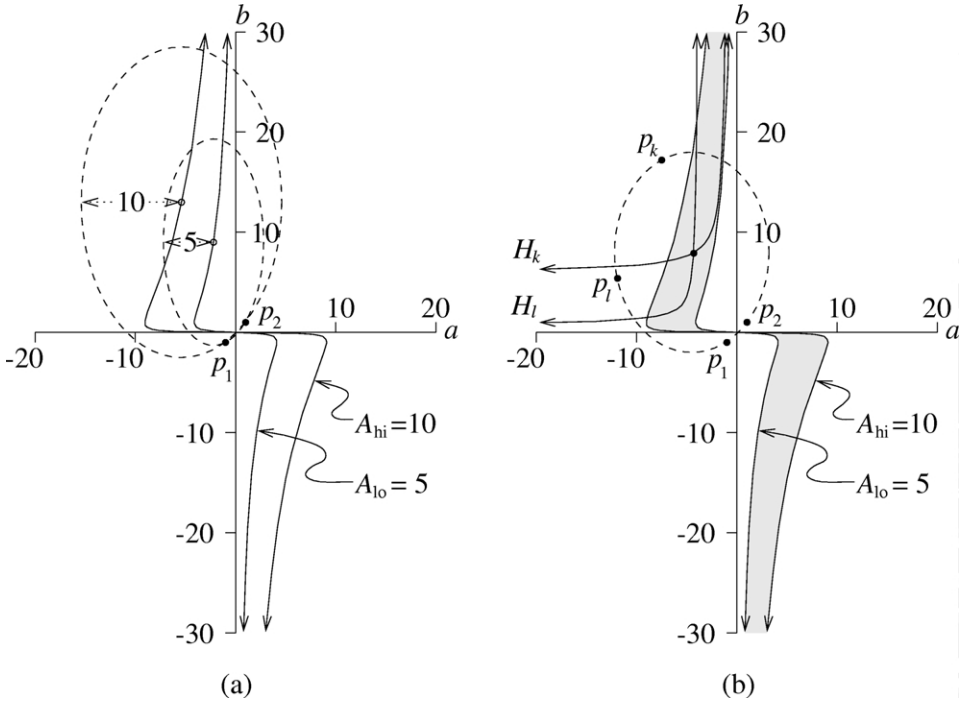

Fig. 8. (a) An example of two A-contours in the dual plane and (b) the corresponding A-region.

the interval. (In the circular arc case, the corresponding region was an annulus.) As in the circular case, the algorithm maintains an interval that contains $\widehat{A}_{1,2}$. (The initial interval is $(0, +\infty]$.) We contract the interval through a series of stages. In this subsection we establish a representation of the region that corresponds to the interval $(A_{\text{lo}}, A_{\text{hi}}]$ which, together with the hyperbola arrangement, satisfies the boundary intersection properties presented in Section 3.1. From this it will follow that the building blocks of Sections 3.1 and 3.2, i.e., intersection point counting and sampling, can be applied.

Consider a fixed horizontal radius value, $A \geqslant 1$, and let $C(A)$ denote the locus of centers of ellipses that pass through p_1 and p_2 and whose horizontal radius is A. We call this an *A-contour*. (Fig. 8(a) provides an example of two such curves, one for $A_{\text{lo}} = 5$ and the other for $A_{\text{hi}} = 10$. The figure also shows two ellipses in dashed lines, centered on these contours, one of horizontal radius 5 and the other of radius 10.) Our next result provides an analysis of this curve's structure.

Lemma 4.6. *Given any $A \geqslant 1$, $C(A)$ is a connected curve in the dual plane consisting of the points $(a(b), b)$, which, for $b \neq 0$, satisfies*

$$a(b) = \frac{b^2 + 1 - \sqrt{(b^2-1)^2 + (2bA)^2}}{2b}.$$

This curve satisfies the following properties:

(i) *it passes through and is symmetric with respect to the origin,*
(ii) *it is monotone with respect to the b-axis, and*
(iii) $\lim_{b\to\infty} a(b) = 0$.

Proof. To establish (i), recall from Lemma 4.1(ii) that $bA^2 = -(1-ab)(a-b)$. First observe that the point $(a,b) = (0,0)$ is a solution to this equation (irrespective of A), so the resulting curve passes through the origin of the dual plane. If (a,b) satisfies the equation, then so does $(-a,-b)$, implying its symmetry with respect to the origin.

By symmetry, it suffices to consider the case $b > 0$. Expanding the above equation and collecting common terms for a yields $a^2b - a(b^2+1) + b(1-A^2) = 0$. The solution of this quadratic equation in a is

$$a(b) = \frac{1+b^2 \pm \sqrt{\Delta}}{2b},$$

where the discriminant is $\Delta = (b^2-1)^2 + (2bA)^2$.

Observe that the discriminant is always positive, so two distinct real roots exist for each value of $b \neq 0$. We claim that only one root, namely the one that arises by subtracting $\sqrt{\Delta}$, yields the center of an aligned ellipse. To see this, consider the other root. From Lemma 4.1(ii), we have

$$\frac{A^2}{B^2} = -\frac{a(b)}{b} = -\frac{b^2+1+\sqrt{(b^2-1)^2+(2bA)^2}}{2b^2} < 0,$$

which is impossible. Since there is only one solution for each value of b, we have established (ii).

To prove (iii), we first divide the numerator and denominator of $a(b)$ by b^2, obtaining

$$a(b) = \frac{1+b^{-2}-\sqrt{(1-b^{-2})^2+4A^2b^{-2}}}{2b^{-1}}.$$

Computing the limit of this quantity is complicated by the square root term. As b approaches infinity, b^{-2} approaches 0. If we let $f(x) = \sqrt{(1-x)^2+4A^2x}$, we can substitute a Taylor's expansion of $f(b^{-2})$ about 0 in place of the square root term. It is easy to verify that a Taylor's expansion of $f(x)$ about 0 is

$$f(x) = 1 + (2A^2-1)x + \mathrm{O}(x^2).$$

Thus,

$$\begin{aligned} a(b) &= \frac{1+b^{-2}-f(b^{-2})}{2b^{-1}} \\ &= \frac{1+b^{-2}-(1+(2A^2-1)b^{-2}+\mathrm{O}(b^{-4}))}{2b^{-1}} = \frac{(1-A^2)}{b} - \mathrm{O}\left(\frac{1}{b^3}\right). \end{aligned}$$

Thus, $\lim_{b\to\infty} a(b) = 0$. □

The region, $R(A_{\mathrm{lo}}, A_{\mathrm{hi}})$, which corresponds to the interval $(A_{\mathrm{lo}}, A_{\mathrm{hi}}]$, can be seen as the region of the dual plane lying between two A-contours. (Fig. 8(b) illustrates such a region obtained for $A_{\mathrm{lo}} = 5$ and $A_{\mathrm{hi}} = 10$. It also shows two hyperbolas, H_k and H_ℓ, associated with the points p_k and p_ℓ, respectively. Since their intersection lies within this region, it is the center of an ellipse of horizontal radius between 5 and 10.)

To apply analogous intersection counting and sampling routines, which were presented in Sections 3.1 and 3.2, we need to present the boundary of this region in a way that allows us to sort intersections with the hyperbolas of the pseudoline arrangement in cyclic order around the boundary. This is easy to do. First orient $C_{\mathrm{lo}} = C(A_{\mathrm{lo}})$ from bottom to top and orient $C_{\mathrm{hi}} = C(A_{\mathrm{hi}})$ from top to bottom. Next, break each contour into two pieces, the portion above the origin and the portion below the origin. Joining both

pieces above the origin, and then joining both pieces below the origin provides us with the boundary of two disjoint, closed (but unbounded) regions of the dual plane. We can apply the intersection counting and sampling procedures individually to each piece, and then combine the two results.

It is easy to see that each of the resulting boundaries satisfies the boundary intersection properties presented in Section 3.1. Property (i) follows from the fact that the region is closed, (ii) follows from the fact that the limit points of each hyperbolic arc lie outside the region (this is in part a consequence of general position assumption (i)), and (iii) follows from the fact that hyperbolas and A-contours are algebraic curves of bounded degree (and so their intersections can be determined either through numeric or symbolic means), and boundary sorting can be performed by the monotonicity of the A-contours.

As a practical consideration, it should be noted that the equation derived in Lemma 4.6 has a singularity at the origin, and before applying a numerical procedure for computing intersections in this neighborhood, an alternative formulation of the equation as a function of a rather than b should be derived. Because the underlying equation is quadratic in b, this can be done symbolically.

4.4. Median radii and range queries

To be able to apply the repeated median computation, we have one remaining task, namely to establish the corresponding ranges used in finding the median horizontal radius for each curve H_k in the arrangement of hyperbolas. Recall from Section 3.3, that in the circular case the problem (in primal form) is, given a fixed data point p_i and any other data point p_j, determine the median radius among the $n-2$ circles passing through these two points and any other third data point. We showed that this could be reduced to a randomized binary search in which each probe was solved by applying a range query over the set of data points. Each range was the union of two regions, each of which was the symmetric difference of two circles.

In the case of aligned ellipses, this task is generalized as follows. We have three fixed points, p_1, p_2 and p_k. Among the $n-3$ aligned ellipses passing through these three points and any other fourth data point, determine the median horizontal radius. In the dual plane, this is equivalent to finding the intersection point on H_k in the arrangement of hyperbolas, which corresponds to the center of the aligned ellipse having the median horizontal radius value.

The randomized binary search is exactly the same as that described in Section 3.3. The only significant difference is that the type of ranges are different. The main result of this subsection, which is the analogue to Lemma 3.3, states that the corresponding range is the union of a constant number of symmetric differences of aligned ellipses. This is illustrated in Fig. 9.

Lemma 4.7. *Given three data points p_1, p_2 and p_k and an interval $(A_{\mathrm{lo}}, A_{\mathrm{hi}}]$, the aligned ellipses passing through these three data points and any other fourth data point, and whose horizontal radii are in the interval, are in* 1–1 *correspondence with the set of data points lying within the union of a constant number of regions of the plane, each being the symmetric difference of two elliptical disks.*

Proof. The centers of aligned ellipses passing through the three given points lie on the hyperbola H_k in the dual plane. Hence, the locus of centers of those ellipses whose horizontal radius lies within $(A_{\mathrm{lo}}, A_{\mathrm{hi}}]$ is the intersection of H_k with $R(A_{\mathrm{lo}}, A_{\mathrm{hi}})$, i.e., the dual region associated with the given interval. Thus this intersection consists of some number of connected segments of H_k. From Lemma 4.6 we know that the boundaries of this region consist of two algebraic curves of bounded degree. Since H_k, too, is an

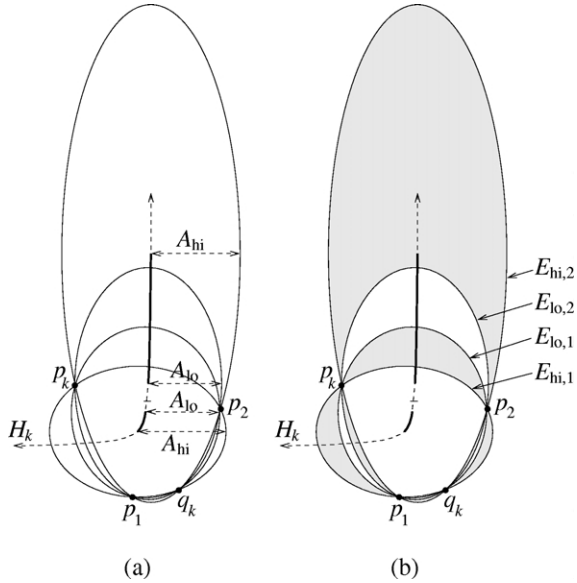

Fig. 9. Searching for the median horizontal radius: (a) the set of aligned ellipses that pass through p_1, p_2, p_k and a fourth point q_k; (b) the subset of the above ellipses whose horizontal radius is contained in the interval $(A_{\text{lo}}, A_{\text{hi}}]$ lies within the shaded region.

algebraic curve of bounded degree, the number of connected segments is some constant. These segments are shown as thick lines in Fig. 9. (A more detailed analysis would reveal that the horizontal radius is a unimodal function along H_k, and hence it can be shown that the number of segments is at most two, as it was in the circular case.)

Associated with the endpoints of these segments are a constant number c of closed elliptical disks, $E_{\text{lo},i}$, $E_{\text{hi},i}$, for $1 \leqslant i \leqslant c$. (Again, a more careful analysis would show that $c = 2$.) $E_{\text{lo},i}$ has horizontal radius A_{lo}, and $E_{\text{hi},i}$ has horizontal radius A_{hi}. Let S denote the union of the symmetric differences of $E_{\text{lo},i}$ and $E_{\text{hi},i}$, for $1 \leqslant i \leqslant c$. (Observe that p_1, p_2 and p_k are not in S, since they belong to both disks.) Because ρ varies monotonically along H_k, we can apply Lemma 4.3(ii) (where ρ' and ρ'' take on the parameter values at the endpoints of each interval). It follows that for any other data point p_ℓ, the horizontal radius of the ellipse that passes through p_1, p_2, p_k and p_ℓ (if it exists) is within the interval $(A_{\text{lo}}, A_{\text{hi}}]$ if and only if p_ℓ lies within S. (From the proof of Lemma 4.3(ii) it also follows that if p_ℓ lies within S, then such an ellipse exists.) It follows immediately, that the data points lying within S are in 1–1 correspondence with the circles passing through p_1, p_2, p_k and one other data point. □

Thus, the task of counting the number of aligned ellipses passing through the points p_1, p_2 and p_k, and whose horizontal radii lie within a given interval, can be reduced to performing a range counting query over ranges defined by a constant number of Boolean operations on sets of bounded algebraic complexity. Hence, the results for range searching of Agarwal and Matoušek [1] can be applied in this case, as well. That is, counting the number of data points in this type of range can be performed in $O(n^\gamma)$ time, for some $\gamma < 1$, after $O(n \log n)$ preprocessing and with linear space. (Note that the value of γ is different from the one used in Section 3.3, and this affects the value of $\beta = 1 - \gamma$ used in the algorithm.)

Therefore, we can apply the randomized binary search to compute the median horizontal radius for each of the sampled points in $O(n \log n)$ expected time, as in Section 3.3. Combining this with the intersection counting and sampling building blocks, which were established in the previous section, we have all the building blocks needed to generalize the repeated median algorithm to finding the horizontal radius estimate, $\widehat{A}$, for aligned ellipses. (Computing $\widehat{B}$ is carried out similarly.) To compute $\widehat{a}$ ($\widehat{b}$), note that the associated *value* of an intersection point in the hyperbola arrangement is just the a- (b-) coordinate of the point. Thus, the *a-contours* and *b-contours* are vertical and horizontal lines, respectively, and the associated regions to be contracted are just horizontal and vertical strips. Finally, the regions used for range queries are essentially the same, but each region will consist of the symmetric difference of a single pair of ellipses. This follows because the hyperbola H_k intersects each vertical or horizontal strip in a single segment. As mentioned before, the computation of the Theil–Sen estimator is a simpler variant, since only the intersection counting and sampling steps are needed. Thus, in conclusion, we have the main result of this section.

Theorem 4.1. *The nonlinear, non-hierarchical Theil–Sen and RM aligned ellipse estimators can be computed in* $O(n^3 \log n)$ *expected time and* $O(n)$ *space.*

5. Conclusions

Efficient randomized algorithms for computing robust circular arc and aligned ellipse estimators were presented in this paper. In particular, it was shown that the (non-hierarchical) Theil–Sen and repeated median circular arc estimators can be computed in $O(n^2 \log n)$ expected time and $O(n)$ space. It was also shown that the Theil–Sen and repeated median aligned ellipse estimators can be computed in $O(n^3 \log n)$ expected time and $O(n)$ space. Both algorithms rely on generalized techniques for intersection counting and sampling, and range searching.

It is natural to ask whether randomization is necessary in these results. We believe that randomization can be removed from our algorithms without increasing the asymptotic running times, but this would likely come at the expense of more complicated algorithmic techniques.

We conjecture that the methods introduced in this paper can be applied to computing similar estimators for (general) ellipses and arbitrary 2-D conic sections in $O(n^4 \log n)$ expected time and $O(n)$ space. In fact, this suggests the general conjecture that for any "reasonable parameterization" of a k-parameter planar algebraic curve, corresponding Theil–Sen and RM estimators can be computed in $O(n^{k-1} \log n)$ time and $O(n)$ space.

Acknowledgements

We thank Michael Dillencourt and Jiří Matoušek for their inspiring collaboration, as well as Pankaj Agarwal for bringing to our attention the important reference [1]. Also, we are grateful to Azriel Rosenfeld for his comments, and Kathy Bumpass for her help in preparing this paper.

References

[1] P.K. Agarwal, J. Matoušek, On range searching with semialgebraic sets, Discrete Comput. Geom. 11 (1994) 393–418.

[2] I. Amir, Algorithms for finding the center of circular fiducials, Comput. Vision Graphics Image Process. 49 (1990) 398–406.

[3] F.L. Bookstein, Fitting conic sections to scattered data, Comput. Graphics Image Process. 9 (1979) 56–71.

[4] Brönnimann, B. Chazelle, Optimal slope selection via cuttings, in: Proceedings of the Sixth Canadian Conference on Computational Geometry Saskatton, Saskatchewan, Canada, 2–6 August 1994, pp. 99–103.

[5] J. Cabrera, P. Meer, Unbiased estimation of ellipses by bootstrapping, IEEE Trans. Pattern Anal. Machine Intell. 18 (1996) 752–756.

[6] B.B. Chaudhuri, Optimal circular fit to objects in two and three dimensions, Pattern Recogn. Lett. 11 (1990) 571–574.

[7] B.B. Chaudhuri, P. Kundu, Optimum circular fit to weighted data in multi-dimensional space, Pattern Recogn. Lett. 14 (1993) 1–6.

[8] B. Chazelle, E. Welzl, Quasi-optimal range searching in spaces of finite VC-dimension, Discrete Comput. Geom. 4 (1989) 467–490.

[9] H. Chernoff, A measure of asymptotic efficiency for test of a hypothesis based on the sum of observations, Ann. Math. Statist. 23 (1952) 493–507.

[10] R. Cole, J.S. Salowe, W.L. Steiger, E. Szemerédi, An optimal-time algorithm for slope selection, SIAM J. Comput. 18 (1989) 792–810.

[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA; McGraw-Hill, New York, 1990.

[12] M.B. Dillencourt, D.M. Mount, N.S. Netanyahu, A randomized algorithm for slope selection, Internat. J. Comput. Geom. Appl. 2 (1992) 1–27.

[13] D.L. Donoho, P.J. Huber, The notion of breakdown point, in: P.J. Bickel, K. Doksun, J.L. Hodges Jr. (Eds.), A Festschrift for Erich L. Lehman, Wadsworth International Group, Belmont CA, 1983, pp. 157–184.

[14] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Comm. of the ACM 15 (1972) 11–15.

[15] H. Edelsbrunner, D.L. Souvaine, Computing median-of-squares regression lines and guided topological sweep, J. Amer. Statist. Assoc. 85 (1990) 115–119.

[16] H. Edelsbrunner, E. Welzl, Halfplanar range search in linear space and $O(n^{0.695})$ query time, Inform. Process. Lett. 23 (1986) 289–293.

[17] P. Erdös, J. Spencer, Probabilistic Methods in Combinatorics, Academic Press, New York, 1974.

[18] W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, Wiley, New York, 1950.

[19] R. Floyd, R. Rivest, Expected time bounds for selection, Comm. of the ACM 8 (1975) 165–172.

[20] C.A.R. Hoare, Proof of a program: FIND, Comm. of the ACM 13 (1970) 39–45.

[21] L.J. Guibas, R. Sedgewick, A dichromatic framework for balanced trees, in: Proceedings of the Nineteenth Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1978, pp. 1–8.

[22] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, W.A. Stahel, Robust Statistics: The Approach Based on Influence Functions, Wiley, New York, 1986.

[23] D. Haussler, E. Welzl, ε-nets and simplex range queries, Discrete Comput. Geom. 2 (1987) 127–151.

[24] P.V.C. Hough (1962) Method and means for recognizing complex patterns, U.S. Patent 3,069,654.

[25] P.J. Huber, Robust Statistics, Wiley, New York, 1981.

[26] S.H. Joseph, Unbiased least squares fitting of circular arcs, Comput. Vision Graphics Image Process. 56 (1994) 424–432.

[27] M. Katz, M. Sharir, Optimal slope selection via expanders, Inform. Process. Lett. 47 (1993) 115–122.

[28] U.M. Landau, Estimation of a circular arc center and its radius, Comput. Vision Graphics Image Process. 38 (1987) 317–326.
[29] J. Matoušek, Randomized optimal algorithm for slope selection, Inform. Process. Lett. 39 (1991) 183–187.
[30] J. Matoušek, Efficient partition trees, Discrete Comput. Geom. 8 (1992) 315–334.
[31] J. Matoušek, D.M. Mount, N.S. Netanyahu, Efficient randomized algorithms for the repeated median line estimator, Algorithmica 20 (1998) 136–150.
[32] D.M. Mount, N.S. Netanyahu, Efficient algorithms for robust circular arc estimators, in: Proceedings of the Fifth Canadian Conference on Computational Geometry, Waterloo, Ontario, Canada, August 1993, pp. 79–84.
[33] D.M. Mount, N.S. Netanyahu, Computationally efficient algorithms for high-dimensional robust estimators, CVGIP: Graphical Models and Image Processing 56 (1994) 289–303.
[34] D.M. Mount, N.S. Netanyahu, K. Romanik, R. Silverman, A. Wu, A practical approximation algorithm for the LMS line estimator, in: Proceedings of the Eighth Annual ACM–SIAM Symposium on Discrete Algorithms, New Orleans, LO, January 1997, pp. 473–482.
[35] N.S. Netanyahu, V. Philomin, A. Rosenfeld, A.J. Stromberg, Robust detection of road segments in noisy aerial images, Pattern Recogn. 30 (1997) 1673–1686.
[36] P.L. Rosin, Ellipse fitting by accumulating five-point fits, Pattern Recogn. Lett. 14 (1993) 661–669.
[37] P.J. Rousseeuw, Least median-of-squares regression, J. Amer. Statist. Assoc. 79 (1984) 871–880.
[38] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, Wiley, New York, 1987.
[39] J.G. Semple, L. Roth, Introduction to Algebraic Geometry, Oxford University Press, Oxford, 1949.
[40] P.K. Sen, Estimates of the regression coefficients based on Kendall's Tau, J. Amer. Statist. Assoc. 63 (1968) 1379–1389.
[41] L. Shafer, W.L. Steiger, Randomizing optimal geometric algorithms, in: Proceedings of the Fifth Canadian Conference on Computational Geometry, Waterloo, Ontario, Canada, August 1993, pp. 133–138.
[42] A.F. Siegel, Robust regression using repeated medians, Biometrika 69 (1982) 242–244.
[43] C. Smith, An Elementary Treatise on Conic Sections by the Method of Co-ordinate Geometry, Macmillan, London, 1956.
[44] D.L. Souvaine, J.M. Steele, Efficient time and space algorithms for least median of squares regression, J. Amer. Statist. Assoc. 82 (1987) 794–801.
[45] A. Stein, M. Werman, Robust statistics in shape fitting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Champaign, IL, June 1992, pp. 540–546.
[46] R. Takiyama, N. Ono, A least square error estimation of the center and radii of concentric arcs, Pattern Recogn. Lett. 10 (1989) 237–242.
[47] H. Theil, A rank-invariant method of linear and polynomial regression analysis (Parts 1–3), Nederlandse Akad. Wetenchappen Ser. A 53 (1950) 386–392, 521–525, 1397–1412.
[48] S.M. Thomas, Y.T. Chan, A simple approach for the estimation of circular arc center and its radius, Comput. Vision Graphics Image Process. 45 (1989) 362–370.
[49] D.E. Willard, Polygon retrieval, SIAM J. Comput. 11 (1982) 149–165.
[50] S. Wolfram, Mathematica, A System for Doing Mathematics by Computer, 2nd Edition, Addison-Wesley, Redwood City, CA, 1991.
[51] Z. Wu, L. Wu, A. Wu, The robust algorithms for finding the center of an arc, Comput. Vision Image Understanding 62 (1995) 269–278.
[52] P.C. Yuen, G.C. Feng, A novel method for parameter estimation of [a] digital arc, Pattern Recogn. Lett. 17 (1996) 929–938.
[53] S. Yi, R.M. Haralick, L.G. Shapiro, Error propagation in machine vision, Machine Vision Appl. 7 (1994) 93–114.