# Efficient Algorithms for Robust Circular Arc Estimators

David M. Mount
Department of Computer Science
and Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland, U.S.A.

Nathan S. Netanyahu*
NASA Goddard Space and Flight Center
Greenbelt, Maryland, U.S.A.
and
Center for Automation Research
University of Maryland
College Park, Maryland, U.S.A.

## Abstract

In this paper we introduce two robust estimators for fitting a circular arc through a set of points in the plane. We introduce *nonlinear* Theil-Sen and repeated median (RM) variants for *circular arc estimation* (CAE). We present randomized algorithms for these estimators for $n$ points in the plane, each running in $O(n^2 \log n)$ expected time, and requiring $O(n)$ space.

## 1 Introduction

Fitting a function (e.g., a straight line) to a finite collection of data points in the plane is a fundamental problem in statistical estimation, with numerous 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* (e.g., [9], [16]), that 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, e.g., Rousseeuw and Leroy [16] 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.

The estimators discussed in this paper are generalizations of the following robust line estimators for a set of $n$ distinct points in the plane $\{p_1, p_2, \ldots, p_n\}$.

**Theil-Sen estimator:** The slope of the line of fit is taken to be the median of the set of $\binom{n}{2}$ slopes that result by passing a line through each pair of distinct points in the data set [21], [17]. A number of $O(n \log n)$ algorithms (deterministic and randomized) have been proposed for computing this estimator (e.g., Cole et al. [6], Matoušek [12], and Dillencourt et al. [8]).

**RM estimator:** Siegel's *repeated median estimator* (RM) [18] is defined as follows. For each point $p_i = (x_i, y_i)$, let $\theta_i$ 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*, $\theta^*$, is defined to be the median of the multiset $\{\theta_i\}$. An $O(n \log n)$ randomized algorithm has been proposed by Matoušek, Mount, and Netanyahu [13].

In Mount and Netanyahu [14], we extended our algorithmic methodology for the Theil-Sen and RM line estimators to higher dimensions. Specifically, we showed that $d$-dimensional Theil-Sen and RM estimators (having breakdown points of $1-(1/2)^{1/d}$ and 0.5, respectively) can be computed by randomized algorithms in $O(n^{d-1} \log n)$ expected time and $O(n)$ space, for fixed $d > 2$.

It should be noted that by the linearity of the regression model assumed, all of the above proposed methods automatically apply to curve fitting, for curves that can be analytically represented as a linear combination of $d$ parameters. For example, although $y = ax^2 + bx + c$ is a nonlinear function of $x$, parabola fitting does reduce to 3-D linear fitting, since any parabola can be expressed as a linear combination of three parameters characterizing the parabola (e.g., $a$, $b$, and $c$). Put differently, if each data point $(x_i, y_i)$ in $E^2$ is mapped to a corresponding point $({x_i}^2, x_i, y_i) \equiv (X_i, Y_i, Z_i)$ in $E^3$ ($i = 1, \ldots, n$), then the problem of fitting a parabola to a given set of points (i.e., estimating $a$, $b$, and $c$) is trivially reduced to the problem of fitting the plane $Z = aX + bY + c$ to a corresponding set of $n$ points. In this paper we address the issue of function fitting in a nonlinear domain, i.e., where a function in question may not be expressed as a linear combination of a specified set of its parameters.

*This research was completed while the author held a National Research Council NASA Goddard Associateship. The support of the Air Force Office of Scientific Research under Grant AFOSR-91-0239 (while a doctoral candidate at the University of Maryland) is gratefully acknowledged.

The problem of circular arc fitting in the plane has been considered recently by a number of researchers [10], [22], [20], [3], [4]. Unfortunately, all of these methods are based on a least squares approach, which suggests sensitivity to outlying data. Amir [2] has introduced an alternative technique, the c[h]ord method, which is presumably more robust, but suffers from quantization effects. All of the above remarks suggest that deriving efficient median-based algorithms for circular arc fitting should be a natural goal to pursue.

We can generalize the definitions of the Theil-Sen and RM line estimators to circular arcs in a natural way. Consider a given set of distinct points $p_i = (x_i, y_i)$, for $i = 1, \ldots, n$, that are hypothesized to lie on a circular arc. We make the *general position* assumptions that no three points are collinear and no four points are cocircular. The output of the estimator is the triple $(\hat{a}, \hat{b}, \hat{r})$ containing the coefficients of the circle equation $(x-\hat{a})^2 + (y-\hat{b})^2 = \hat{r}^2$, that fits the data. For each triplet $(i, j, k)$ of distinct indices, $1 \leq i, j, k \leq n$, let $a_{i,j,k}$, $b_{i,j,k}$, and $r_{i,j,k}$ denote the parameters of the unique circle passing through points $p_i$, $p_j$, and $p_k$. The circular arc estimators are defined as follows.

**Theil-Sen CAE:** Defined by taking the median values, over all $\binom{n}{3}$ elements, of each of the above sets, i.e.,

$$\hat{a} = \operatorname*{med}_{i<j<k} a_{i,j,k}, \quad \hat{b} = \operatorname*{med}_{i<j<k} b_{i,j,k}, \quad \hat{r} = \operatorname*{med}_{i<j<k} r_{i,j,k}.$$

**RM CAE:** Defined by taking the median over $i$, of the median over $j$, of the median over $k$ of these parameters, i.e.,

$$\hat{a} = \operatorname*{med}_{i} \operatorname*{med}_{j \neq i} \operatorname*{med}_{k \neq i,j} a_{i,j,k}, \quad \hat{b} = \operatorname*{med}_{i} \operatorname*{med}_{j \neq i} \operatorname*{med}_{k \neq i,j} b_{i,j,k},$$

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

We conjecture that the breakdown points of the above defined Theil-Sen and RM circular arc estimators are identical to their 3-D linear counterparts, i.e., $\approx 21\%$ and $50\%$, respectively. Observe that the definitions described above can be applied to determining the parameters of any type of function, provided that for some $k$, each $k$-tuple of points uniquely defines the parameters of the function.

Recently Stein and Werman [19] have independently introduced similar robust estimators for fitting two-dimensional conic sections in general. It is claimed in their paper that estimating the more "natural" parameters of a curve in question (e.g., *geometric* parameters such as location, radius, etc.) as opposed to estimating the coefficients of its linear representation results in superior estimators, as far as maintaining, for example, the property of statistical (rotation) equivariance. Since no method other than a brute force computation of these estimators is suggested in [19], it is a valid objective to pursue alternative, computationally efficient algorithms according to the above given definitions.

Brute force algorithms for computing the Theil-Sen and RM circular arc estimators would require $O(n^3)$ time. Moreover, the naive Theil-Sen estimator would require $O(n^3)$ space which is rather inefficient. In this paper we present randomized algorithms to compute the above estimators. The algorithms are conceptually simple, run in $O(n^2 \log n)$ expected time, and use linear storage. It should be noted that in contrast to the (hierarchical) computation of the radius estimate in [14], this paper presents efficient algorithms that compute $\hat{r}$ *by definition*, i.e., the radius estimate is computed independently of previously obtained estimates of the center's coordinates.

## 2 The Algorithms

In this section we present the algorithmic framework for our algorithms. Because of space limitations we only present the computation of the most illustrative case, the radius parameter for the RM circular arc estimator. The computation of other parameters and for the Theil-Sen estimator are simpler variations of this method.

To compute the estimated radius, $\hat{r}$, we first define

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

Intuitively $\hat{r}_i$ is the *radius estimate* associated with a fixed point $p_i$. The algorithm to be presented computes $\hat{r}_i$ in (expected) $O(n \log n)$ time for each $i = 1, \ldots, n$. A standard fast selection algorithm may be applied at this stage to compute $\hat{r} = \operatorname*{med}_{i} \hat{r}_i$. Thus, the total running time of the algorithm will be $O(n^2 \log n)$.

We focus now on the computation of $\hat{r}_i$. By definition, this estimate corresponds to a two-dimensional RM computation over all radii, $r_{i,j,k}$, of circles passing through $p_i$, $p_j$, and $p_k$ ($j \neq i$, $k \neq i, j$), for a *fixed* point $p_i$. For $j \neq i$, let $\{b_{i,j}, j \neq i\}$ denote the perpendicular bisector of $p_i$ and $p_j$. Consider the planar line arrangement of the $n-1$ lines $\{b_{i,j}, j \neq i\}$. Observe that the intersection of the 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 $O(n^2)$ vertices of this arrangement are essentially the centers of the circles determined by $p_i$ and any two other points. The distance of each vertex from $p_i$ is the radius of the corresponding circle. Thus we have reduced the problem to that of determining the median distance from a fixed point to every vertex of an arrangement of $n-1$ lines. We use the term *intersection point* to denote a vertex in this arrangement. Henceforth all radius values are to be interpreted as being distances from $p_i$.

To compute $\hat{r}_i$ we apply the same interval contraction scheme, presented in the randomized algorithms for the Theil-Sen and RM line estimators (see [8], [12], and [13], respectively). We maintain an interval $[r_{lo}, r_{hi}]$, which defines an annulus centered at $p_i$ that contains $\hat{r}_i$. (The initial interval is $[0, +\infty]$.) The interval is contracted through a series of stages. During each stage a subinterval $[r'_{lo}, r'_{hi}]$ that contains $\hat{r}_i$ is constructed.

Let us describe the operations of a typical stage in greater detail. Suppose that $[r_{lo}, r_{hi}]$ is the current interval, and that $\hat{r}_i \in [r_{lo}, r_{hi}]$. The current interval corresponds to an annulus whose center is the point $p_i$ and whose radii are $r_{lo}$ and $r_{hi}$. Let $A_i(r_{lo}, r_{hi})$ denote this annulus. For each perpendicular bisector, $b_{i,j}$, we maintain three counts: $I_j$, $A_j$, and $O_j$, which denote the number of the bisector's intersection points lying inside the circle $r = r_{lo}$, inside the annulus $A_i(r_{lo}, r_{hi})$, and outside the circle $r = r_{hi}$, respectively. (Counting the number of intersections per bisector will be discussed in the next section.) Depending on the relationship between $\lceil (n-2)/2 \rceil$, $I_j$, and $I_j + A_j$, we can determine whether the median radius associated with $b_{i,j}$ lies inside the circle $r = r_{lo}$, within the annulus $A_i(r_{lo}, r_{hi})$, or outside the circle $r = r_{hi}$. The set of bisectors $\{b_{i,j}, j \neq i\}$ is partitioned accordingly into three subsets $I$, $A$, and $O$, respectively. Since we assume that $\hat{r}_i$ lies within $A$, it follows that $|I| < \lceil (n-1)/2 \rceil \leq |I| + |A|$. A bisector is a *candidate* to provide the two-dimensional RM, $\hat{r}_i$, if it lies in $A$. In particular, the candidate whose median radius (i.e., median distance of its intersections to the point $p_i$) is of rank $\lceil (n-1)/2 \rceil - |I|$ within $A$ is the desired candidate.

We apply essentially the same randomized interval contraction algorithm given in [13] for the linear RM estimator (where $I$, $A$, and $O$ play the roles of $L$, $C$, and $R$, respectively). We omit the details here, but the idea is as follows. Recall that the point $p_i$ is fixed. For some suitably chosen constant $\beta < 1$ (whose value will be defined in Subsection 3.2), randomly sample $O(n^\beta)$ bisectors. For each sampled bisector $b_{i,j}$ determine the point on this bisector from the line arrangement with the median distance from $p_i$. This task is described in Subsection 3.2. Using the median distances of these sampled lines, construct a confidence interval for the desired RM radius, and derive a contracted annulus, $A_i(r'_{lo}, r'_{hi})$, with respect to this confidence interval. Apply an intersection counting procedure (described below) to determine whether the contracted annulus indeed contains $\hat{r}_i$, and if so recurse on this interval. (This will occur with high probability.) Otherwise recurse on one of the two annuli lying on either side of the contracted annulus.

The algorithm presented in [13] makes no special use of the linearity of the RM line estimation problem, except in three subtasks. For CAE, these subtasks are: (1) Counting the number of intersections of a set of lines within an annulus, (2) randomly sampling (lines or intersection points) from a set of lines that intersect an annulus, and (3) determining the point on a line of an arrangement with the median distance from $p_i$. In the next section we will show that each of these tasks can be performed in $O(n \log n)$ time and $O(n)$ space. Applying the same analysis of [13], it follows that each stage takes $O(n \log n)$ expected time, and the number of stages in the expected case is $O(1)$. Repeating for each point $p_i$ gives the $O(n^2 \log n)$ time, as claimed.

## 3 Building Blocks

### 3.1 Intersection Counting/Sampling

In this subsection we describe how to perform efficient counting (and sampling) of line intersections in a given annulus. The technique to be presented is a generalization of the intersection/inversion counting technique used in [8], [13], for counting/sampling line intersections in a given interval. More specifically, define a collection of *pseudolines* to be a collection of curves such that any pair of curves intersects at most once. We demonstrate below that given $n$ pseudolines and a bounded region, such that, (1) each pseudoline intersects the boundary of this region a positive, even number of times, (2) the number of intersections between a pseudoline and the boundary is bounded above by some constant, and (3) the intersections of pseudolines along the region's boundary can be cyclically ordered (see Figure 1(a)), one can compute the number of intersections (between each pair of pseudolines) in the region in time $O(n \log n)$. Also, sampling $O(n)$ intersections can be carried out in a similar time bound.

We assume that the region's boundary is connected. If not (as is the case with an annulus) a pseudoline segment can be added to form a channel joining the two parts of the boundary. For each pseudoline we determine its intersections with the boundary of the region, and break the pseudoline into a collection of pseudoline segments. The intersections will be counted individually for each segment, and counts for the segments from the same pseudoline will be combined later. First sort the $O(n)$ endpoints of the pseudoline segments along the boundary of the region. These are stored in a list $l$. We maintain a sort of stack in which we can insert elements only at the top, but can remove elements from any position. Initially the stack is empty. Process the elements of $l$ in the following manner. When a pseudoline segment endpoint, $a_i$, is encountered in $l$, if this is the first time this segment has been encountered it is pushed onto the top of the stack. Upon encountering the other endpoint of this segment, the number of stack entries on top of the element is determined and the segment is removed from the stack.

It is easy to see that because of the pseudoline property and the fact that each pseudoline intersects the boundary an even number of times, when $a_i$ is removed from the stack, any item above $a_i$ in the stack will intersect $a_i$ exactly once. Because every pseudoline intersects the boundary of the region at least once, every pseudosegment's intersections will be counted.

The stack can be implemented using a modification of any type of a balanced binary search tree, e.g., a red-black tree (see, for example, [7] for a detailed tutorial). The tree is modified for the purposes of counting in a straightforward manner. The underlying tree stores the elements of the stack, at the leaf level. Also, every internal node $N$ contains two fields: $C$, a count which corresponds to the number of leaves stored in the node's left subtree, and $I$, an increment to be added eventually to the individual (intersection) counts of the leaves of the subtree whose root is $N$. (This will be clarified below.)

We modify the basic operations supported by a red-black tree (RBT) in the following manner:

**Insertion:** A newly encountered element, $a_{i+1}$, would be inserted into the tree as a leftmost leaf. More precisely, the current leftmost leaf, $a_i$, is replaced by a subtree whose root node is red (with $C = 1, I = 0$), and whose left and right children are $a_{i+1}$ and $a_i$, respectively. Maintaining the tree requires (1) incrementing the $C$ values of all the internal nodes on the path from the newly inserted red node to the root, and (2) restoring the properties of a RBT.

**Deletion:** Deleting an element $a_i$ is done by splicing out its parent node, i.e., replacing this node with the sibling of the (deleted) element. This requires (1) updating the (intersection) count for the pseudoline segment $a_i a_i$ by adding $I$ values of nodes along the path to the root, (2) updating the $C$ and $I$ values of nodes along the above path, and (3) restoring the properties of a RBT.

**Rotation:** Left and right rotations are required to restore the properties of a RBT (see [7], Chapter 14), and are often invoked upon insertion and deletion. In addition to changing the pointer structure of the tree we also need to update appropriately the $C$ and $I$ values of the rotated nodes.

Figure 3 illustrates insertion and deletion of a node for the permutation list $1, 5, 6, 2, 3, 1, \ldots, 3$, which corresponds to the example depicted in Figure 1(a). Each node is associated with a $(C, I)$ list.

As the height of a RBT is $O(\log n)$, and since the overhead for updating $(C, I)$ is $O(1)$ upon insertion, deletion, and rotation, we arrive at the following.

LEMMA 3.1 *Given a simple arrangement of pseudolines and a bounded region, counting/sampling intersections along individual pseudolines can be done in $O(n \log n)$ time.*

Due to space limitations we omit a discussion of the sampling of intersections. In principle, though, it draws directly from the above sketched counting technique, and is analogous to the sampling procedures described in [8], [13]. As mentioned earlier, Lemma 3.1 can be applied directly to counting/sampling intersections in a given annulus, by considering the bounded region that is formed by cutting the annulus along an arbitrary straight line (see Figure 1(b)).

## 3.2 Finding Line Medians

In this subsection we describe how to determine the "median intersection point" on a line of the bisector arrangement. Recall that there is a fixed point $p_i$, and an arrangement of $n-1$ bisectors $b_{i,j}$ for each $j \neq i$. Each intersection point between bisectors $b_{i,j}$ and $b_{i,k}$ in this arrangement is the location of the center of some circle passing through $p_i$, $p_j$, and $p_k$. The radius of this circle is the distance from $p_i$ to this intersection point. Given a bisector $b_{i,j}$ we wish to determine the median radius among all $n-2$ intersection points. We do this in $O(n^\gamma \log n)$ time, for some $\gamma < 1$, by an application of recent results in the theory of range searching (e.g., [5], [11], [1]).

The desired median value is determined by a binary search. The set of circles passing through $p_i$ and $p_j$ form a linearly ordered set, whose centers lie on the bisector $b_{i,j}$. For a given radius value $r$ ($r \geq \text{dist}(p_i, p_j)$), the locus of points $q$, for which the circle determined by $p_i$, $p_j$, and $q$ has radius less than $r$ can be seen to be the symmetric difference between the two open disks of radius $r$ that pass through $p_i$ and $p_j$. For a given pair of radii $r_{lo}$ and $r_{hi}$, the locus of points $q$, for which the circle determined by $p_i$, $p_j$, and $q$ determines a circle of radius between $r_{lo}$ and $r_{hi}$ can be seen to be the difference of two such sets. Such a locus can be described in terms of a constant number of Boolean operations on circles. Thus we can count the number of points $p_k$ lying within such a locus by applying range searching. The type of range is shown in Figure 2. Agarwal and Matoušek [1] have shown that such queries can be solved in $O(n^\gamma)$ time after $O(n \log n)$ preprocessing and with linear space.

To determine the median we apply the same randomized binary search presented in [13]. The only difference is that rather than using double-wedge range queries, we use range queries to the type of range described above. The value $\beta$ introduced earlier is defined to be $1 - \gamma$. Thus by applying binary search to each of $n^\beta$ lines, where each probe of the binary search performs a range count, the total running time of the median computation phase is $O(n^\beta n^\gamma \log n) = O(n \log n)$. Details will be presented in the full paper [15].

## 4 Conclusions

Efficient (randomized) algorithms for robust circular estimation were presented in this paper. In particular, it was shown that (non-hierarchical) Theil-Sen and repeated median CAEs can be computed in $O(n^2 \log n)$ (expected) time and $O(n)$ space. The algorithms derived rely on generalized techniques for intersection/inversion counting/sampling and range searching. Extending the current fitting methodology to a larger set of functions (e.g., 2-D conic sections) would be pursued as a future research topic.

**Acknowledgements:** We would like to thank Michael Dillencourt and Jiří Matoušek for their inspiring collaboration, as well as Pankaj Agarwal for bringing to our attention the important reference [1].

## References

[1] P. K. Agarwal and J. Matoušek (1992), On range searching with semialgebraic sets, in *Proc. 17th Symp. on Math. Found. of Comp. Sci.*, Lect. Notes in Comp. Sci. 629, Springer-Verlag 1992, 1–13.

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

[3] B. B. Chaudhuri (1990), Optimal circular fit to objects in two and three dimensions, *Pattern Recognition Letters*, **11**, 571–574.

[4] B. B. Chaudhuri and P. Kundu (1993), Optimum circular fit to weighted data in multi-dimensional space, *Pattern Recognition Letters*, **14**, 1–6.

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

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

[7] T. H. Cormen, C. E. Leiserson, and R. L. Rivest (1990), *Introduction to Algorithms*, MIT Press, McGraw-Hill, New York.

[8] M. B. Dillencourt, D. M. Mount, and N. S. Netanyahu (1992), A randomized algorithm for slope selection, *Int. J. Comp. Geom. and Appl.*, **2**, 1–27.

[9] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel (1986), *Robust Statistics: The Approach Based on Influence Functions*, John Wiley & Sons, New York.

[10] U. M. Landau (1987), Estimation of a circular arc center and its radius, *Computer Vision, Graphics, and Image Processing*, **38**, 317–326.

[11] J. Matoušek (1991), Efficient partition trees, *Proc. 7th ACM Symp. on Comp. Geom.*, June 1991, 1–9.

[12] J. Matoušek (1991), Randomized optimal algorithm for slope selection, *Information Processing Letters*, **39**, 183–187.

[13] J. Matoušek, D. M. Mount, and N. S. Netanyahu (1993), Efficient randomized algorithms for the repeated median line estimator, in *Proc. of the 4th ACM-SIAM Ann. Symp. on Discrete Algorithms*, January 1993, 74–82.

[14] D. M. Mount and N. S. Netanyahu (1992), Computationally efficient algorithms for high-dimensional robust estimators, in *Proc. of the 4th Canadian Conf. on Comp. Geom.*, August 1992, 257–262.

[15] D. M. Mount and N. S. Netanyahu (1993), Efficient algorithms for robust circular arc estimators, technical report in preparation.

[16] P. J. Rousseeuw and A. M. Leroy (1987), *Robust Regression and Outlier Detection*, John Wiley & Sons, New York.

[17] P. K. Sen (1968), Estimates of the regression coefficients based on Kendall's Tau, *J. Amer. Stat. Assoc.*, **63**, 1379–1389.

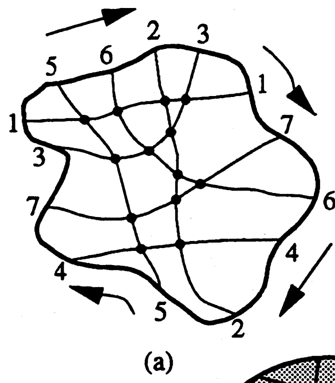
[18] A. F. Siegel (1982), Robust regression using repeated medians, *Biometrika*, **69**, 242–244.

[19] A. Stein and M. Werman (1992), Robust statistics in shape fitting, in *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, June 1992, 540–546.

[20] R. Takiyama and N. Ono (1989), A least square error estimation of the center and radii of concentric arcs, *Pattern Recognition Letters*, **10**, 237–242.

[21] H. Theil (1950), A rank-invariant method of linear and polynomial regression analysis, (Parts 1–3), *Nederlandse Akademie Wetenchappen Series A*, **53**, 386–392, 521–525, 1397–1412.

[22] S. M. Thomas and Y. T. Chan (1989), A simple approach for the estimation of circular arc center and its radius, *Computer Vision, Graphics, and Image Processing*, **45**, 362–370.



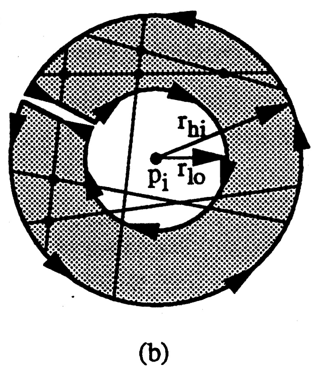
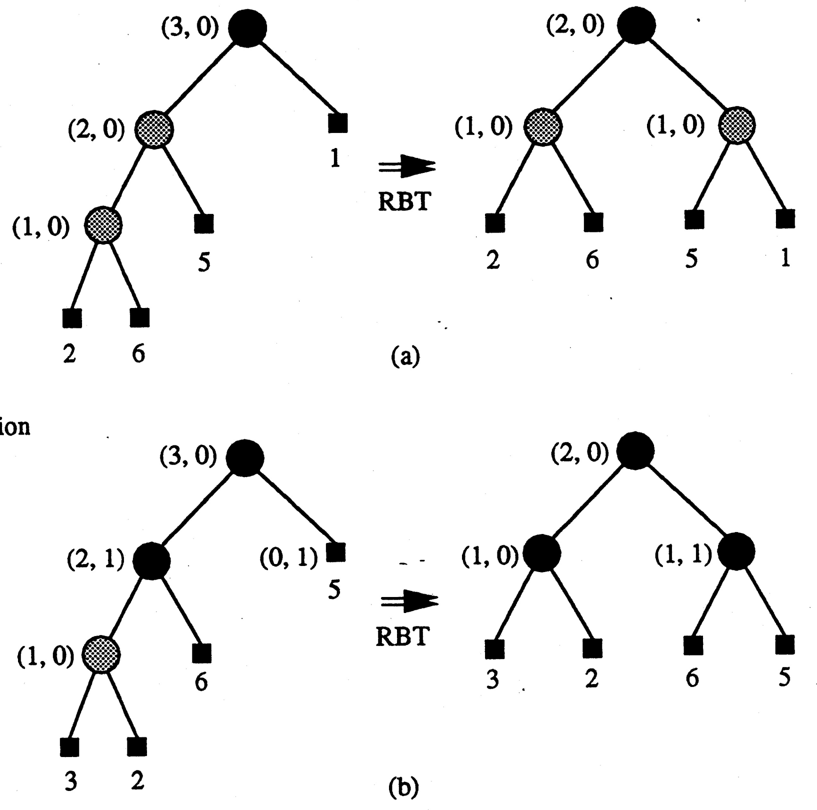


Figure 1: (a) Generalized intersection / inversion counting; (b) its applications for CAE.



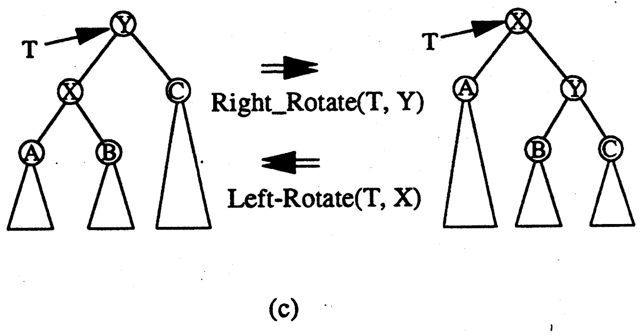


Figure 3: (a) Insertion of pseudoline 2; (b) deletion of pseudoline 1; (c) rotations to the right and to the left.

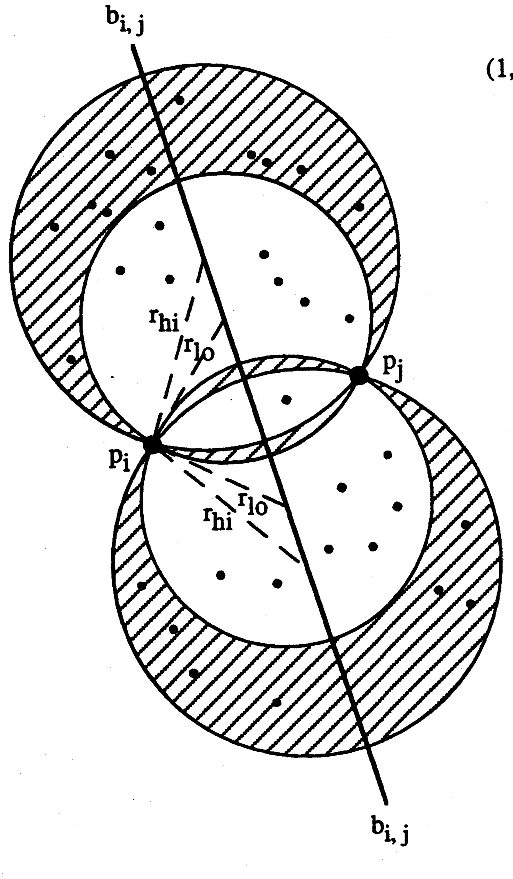


Figure 2: The range associated with $p_i$, $A_i(r_{lo}, r_{hi})$, for a sampled bisector $b_{i,j}$.