Proceedings of the International conference on Computing Technology and Information Management, Dubai, UAE, 2014

The *Rotating Calipers*: An Efficient, Multipurpose, Computational Tool

Godfried T. Toussaint
New York University Abu Dhabi
United Arab Emirates
gt42@nyu.edu

ABSTRACT

A paper published in 1983 established that the *rotating calipers* paradigm provides an elegant, simple, and yet powerful computational tool for solving several geometric problems. In the present paper the history of this tool is reviewed, and stock is taken of the rich variety of computational two-dimensional problems and applications that have been tackled with it during the past thirty years.

KEYWORDS

Rotating calipers, design and analysis of algorithms, computational geometry, geometric complexity, computer graphics, computer vision, combinatorial optimization, statistics

1 INTRODUCTION

The *rotating calipers* paradigm constitutes a powerful, simple, elegant, and computationally efficient tool that can solve a wide variety of geometric problems in practice. The basic idea first appeared in the 1978 Ph.D. thesis of Michael Shamos, where it was applied to the computation of the maximum distance between the elements of a convex set: the diameter of a convex polygon in the plane [40]. Later I coined the name "Rotating Calipers" for this procedure, and generalized it in a number of ways to solve several other geometric problems. In 1983 I presented some of these results at an IEEE conference in Athens, Greece [46]. Since then the rotating calipers paradigm has been generalized further to solve other problems in two as well as three dimensions. In the present paper the thirty-year history of this tool is reviewed, focusing on the progress made with it on geometry problems and applications, in two-dimensional space.

2 THE ROTATING CALIPERS

The elegant and simple algorithm that appeared in Shamos' thesis [40], showing that the diameter of a convex *n*-sided polygon may be computed in O(*n*) time in the worst case, resembles rotating a pair of calipers through 360º, once around the polygon. This concept is illustrated in Fig. 1, which contains a convex polygon, and its two horizontal lines of support (lower and upper) at vertices p_i and p_j, respectively. Note that the lines of support are *directed*, as indicated by their arrows. This permits the specification of the direction of rotation, so that if the lines are rotated in a clockwise direction while being pivoted about vertices p_i and p_j, the angles θ_i and θ_j that the lines make with vertices p_{i+1} and p_{j+1}, will decrease.

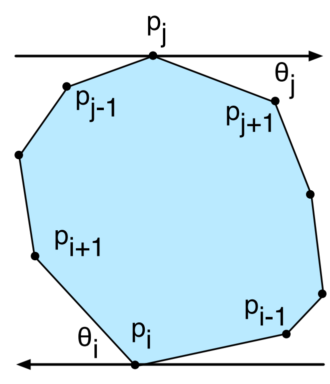

Figure 1. The rotating calipers.

2.1 The Diameter of a Convex Polygon

The *diameter* of a polygon *P* is the maximum distance between a pair of points in *P*. Since P contains an infinite number of points, searching all of them is out of the question. To obtain an efficient algorithm we need a characterization of the diameter in terms of a finite subset of the points in *P*. It is easy to show by a contradiction argument, that the diameter of P is determined by a pair of vertices of P. Therefore the diameter may be calculated by examining the distances between all the pairs of vertices of P, and selecting the maximum distance. However, this naïve (brute force) algorithm requires a number of operations that grows as the square of the number of vertices in the polygon. A more fruitful characterization narrows the set of candidates to be searched down to linear size. Several approaches have been tried in the past in order to speed up diameter-finding algorithms [5], and some characterizations have proved to be incorrect [2]. However, a valid characterization was obtained by Shamos [40] via the pairs of vertices, such as p_i and p_j in Fig. 1. These vertices are *antipodal*, meaning that they admit parallel lines of support. The diameter of a polygon is determined by two antipodal vertices. Furthermore, a polygon with n vertices has O(n) antipodal pairs, assuming that all the vertices of P that have an angle of 180º have been removed, which is a straightforward matter. The rotating calipers provide a simple O(n) procedure for searching all the antipodal pairs to find the maximum. The idea is to place a pair of parallel lines of support in any orientation, say horizontal, as in Fig. 1, and then "rotate" the lines, while keeping them as support lines of the polygon, until they are horizontal again. Such a procedure will visit all pairs of antipodal vertices. The crucial observation that makes this seeming infinite continuous process finite is that the rotation can hop from vertex to vertex. Observe in Fig. 1, that as the two lines of support rotate, the vertices p_i and p_j maintain their antipodality property until one of the lines lies flush with an edge of P. In Fig. 1, θ_j is smaller than θ_i, and thus the line advances from p_j to p_{j+1} identifying the next antipodal pair p_i and p_{j+1}. At each step all that is needed is a comparison of two angles to determine which of the two is smaller, which can be done in constant time.

2.2 The Width of a Convex Polygon

The *width* of a polygon P is the minimum distance between a pair of parallel lines of support of P. As with the diameter definition, to obtain an algorithm, the width must be characterized by a finite subset of the lines of support that can be searched efficiently. The following property yields such a characterization. Let p_i and p_j be two vertices of the convex polygon that admit parallel lines of support, and have internal angles less than 180º, as in the example of Fig. 1. If no edge of P lies on the support lines, there exists a preferred direction of rotation for the support lines such that their separation distance decreases. This implies that the width of the polygon is characterized by a vertex and an edge (a vertex-edge pair) that are antipodal, *i.e.,* such that one of the lines of support lies flush with an edge, as shown in Fig. 2. This characterization of the width of a convex polygon found application to the segmentation of plane curves in the work of Ichida and Kiyono [21]. The characterization has also led to several algorithms for computing the width of a polygon. A useful property in this regard is the fact that for a line that contains any edge of a convex polygon P, the perpendicular distance between the line and the vertices of P, as they are traversed in order, defines a *unimodal* function [45]. Kurozumi and Davis [24], and Imai and Iri [22], independently proposed algorithms for computing the width of a convex polygon by visiting each edge of the polygon, and for each edge searching for the vertex furthest from it (in a perpendicular sense). Since this distance function is unimodal the algorithms in [22] and [24] apply binary search to locate these vertices, for each edge of P. This approach results in algorithms with O(n log n) worst-case time complexities.

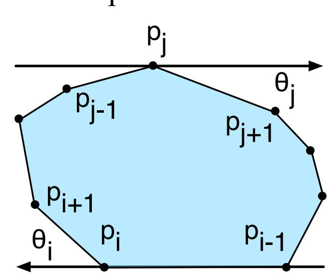

Figure 2. The rotating calipers and the *width* of P.

Houle and Toussaint [20] show that an O(n) time algorithm is achievable by avoiding binary search altogether, and using the rotating calipers instead. To initialize the algorithm, a line of support is constructed through any edge, such as p_{i-1} and p_i in

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

Fig. 2, and the vertex furthest from this line is found (p_j in Fig. 2). At each step during the rotation of the lines, the succeeding edge selected is the one that makes the smaller angle with its line of support. This process yields the vertex opposite the edge in only constant time.

2.3 The Minimum-Area Enclosing Rectangle

The algorithm described in the previous sub-section, for computing the width of a polygon with the rotating calipers, represents an application of the original calipers that use two parallel lines of support to compute the diameter, to a different problem. However, the rotating calipers tool itself has also been generalized in several ways. One generalization introduced in [46] uses more than two rotating lines of support. One nice example uses four lines of support to tackle a problem that arises in the areas of image processing and computer vision [11], [39], [50]-[52], optimal packing and layout problems in manufacturing [13], and automatic tariffing in goods traffic [18]. The problem is that of computing the minimum-area enclosing rectangle of a convex n-sided polygon. The usefulness of this rectangle in packing problems is obvious, but it also has applications to shape analysis. For example, the rectangularity of a shape may be measured by the difference in the areas of the shape and its smallest enclosing rectangle [39]. Freeman and Shapira [13] showed that the smallest rectangle must have one of its four edges flush (collinear) with an edge of the polygon, as illustrated in Fig. 3. The algorithm they propose involves visiting every edge of P, such as edge $[p_{i-1}\ p_i]$ in Fig. 3, and locating the three associated extreme vertices that complete the enclosing rectangle, such as p_t, p_j and p_s in Fig. 3. Their algorithm inspects all the vertices of P to find these three extreme vertices, leading to a total computational complexity of $O(n^2)$. If on the other hand each of these extreme vertices is located in $O(\log n)$ steps using binary search, instead of a linear scan (which is valid due to the unimodality property of the distances involved [45]), then the solution may be found in $O(n \log n)$ time. However, the rotating calipers with four lines of support solves this problem elegantly in $O(n)$ time, as follows. To initialize the procedure any edge of the polygon is selected as the base of a candidate for the smallest enclosing rectangle, and the three extreme vertices are found by a linear scan of all the vertices, as in [13]. The rest of the algorithm proceeds in a manner similar to that used in the algorithm for computing the width of the polygon, except that here the four lines of support are rotated by the smallest of the four angles that the lines make with their succeeding clockwise edges, as shown in Fig. 3. In this way every edge of the polygon generates its candidate rectangle as the support lines make a full revolution around the polygon. Furthermore each candidate rectangle is generated in constant time, resulting in a total time complexity of $O(n)$. An alternate approach to solve this problem in $O(n)$ time, that uses a data structure known as a *star*, is described in [47].

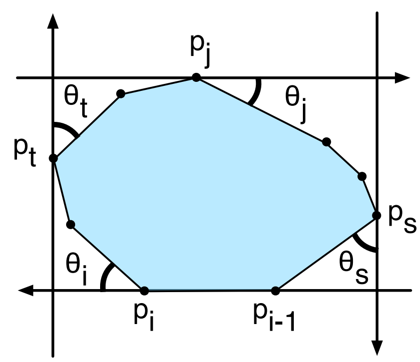

Figure 3. The minimum-area enclosing rectangle.

In closing this sub-section it is worth noting that the rotating calipers can also be used to find minimum-area squares [10], minimum-perimeter enclosures [28], and the densest double-lattice packing of a convex polygon [29].

2.4 The Maximum Distance Between Two Convex Polygons

The maximum distance between two convex polygons, P and Q, arises in several applications including pattern recognition, cluster analysis, and unsupervised learning [12]. It is defined as the largest distance determined by a point in P and a point in Q. As with the diameter of a single polygon, the search for the maximum distance may be restricted to the vertices of P and Q,

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

denoted by p_1, p_2, … , p_n and q_1, q_2, … , q_n, respectively. This maximum distance between P and Q is given by:

$$d_{\max}(P,Q) = \max\{d(p_i, q_j)\},\ i, j=1, 2, \ldots, n, \quad (1)$$

where maximization is done over all i and j, and $d(p_i, p_j)$ is the Euclidean distance between p_i and p_j. Bhattacharya and Toussaint [6] showed that two sets of points may be partitioned into eighteen subsets such that the maximum distance between the two sets is equal to the largest of the eighteen diameters of each of these subsets. Furthermore, the diameter of each subset may be computed with the rotating calipers algorithm applied to their convex hulls. If the two sets are convex polygons, their algorithm runs in O(n) time. However, a simpler and direct O(n) time algorithm for the case of convex polygons was later discovered by Toussaint and McAlear [49]. Their algorithm follows from the generalization of the notion of an antipodal pair of points for a single polygon, as illustrated in Fig. 4.

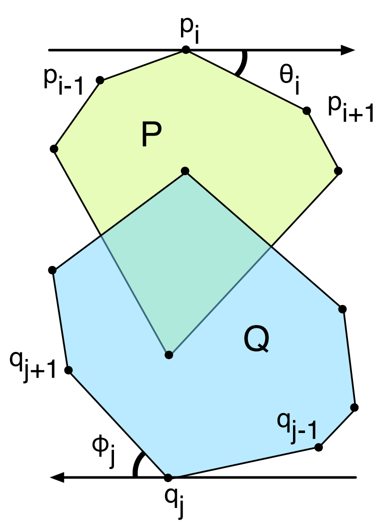

Figure 4. The maximum distance between two convex polygons is determined by an *antipodal pair between* them.

An *antipodal pair between the sets P* and *Q*, is defined as a pair of vertices $p_i \in P$ and $q_j \in Q$, such that they admit parallel lines of support of P and Q, at p_i and q_j, respectively, with the added restrictions that the support lines are oriented in opposite directions, and each polygon lies to the right of its support line. Note that both polygons need not be contained in the parallel strip defined by the two support lines, as is the case for the configuration in Fig. 4. If one polygon is smaller than the other, or if it lies inside the other, a polygon may protrude outside this strip. It is shown in [49] that the maximum distance between the sets is determined by an antipodal pair between the sets. There are only a linear number of such pairs, and they can be searched in O(n) time by rotating the calipers in the same manner as was done in the diameter algorithm. In Fig. 4, the pair p_i and q_j are one candidate pair, and the next candidate pair is obtained in O(1) time by rotating the calipers in a clockwise manner by the smaller of the angles θ_i and φ_j.

2.5 Minkowski Sum of Two Convex Polygons

Consider two points r and s in the plane, denoted by $r(x_r,y_r)$ and $s(x_s,y_s)$, specified by their x and y coordinates. The Minkowski sum (also called the vector sum) of r and s is the new point $t(x_t,y_t)$, where $x_t = x_r+x_s$ and $y_t = y_r+y_s$. The Minkowski sum of two convex polygons P and Q, denoted by $P \oplus Q$, is the set of points obtained by the Minkowski addition of each and every point in P with each and every point in Q. The Minkowski sums of polygons in the plane, and polyhedra in space, find application in spatial planning problems in the field of robotics [25], [26]. The Minkowski sum of two convex polygons, P and Q may be characterized in terms of their vertices, thus making it computable. In particular, $P \oplus Q$ is a convex polygon, and has at most $2n$ vertices, which are Minkowski sums of the vertices of P with those of Q. This characterization implies the following algorithm: first compute all $\Theta(n^2)$ pairwise Minkowski additions of the vertices of P and Q, and then compute the convex hull of the resulting set. Using an efficient O(n log n) time convex hull algorithm, such as Graham's algorithm [15], yields an O(n^2 log n) time algorithm for computing the Minkowski sum. However, a much faster O(n) time algorithm may be obtained by exploiting a characterization of the Minkowski sum in terms of a modification of the notion of an antipodal pair of vertices. Two vertices $p_i \in P$ and $q_j \in Q$ are defined as being *co-podal* if, and only if, they admit parallel directed

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

lines of support of P and Q, at $p_i \in P$ and $q_j \in Q$, respectively, such that the support lines are oriented in the same direction, and each polygon lies to the right of its support line, as illustrated in Fig. 5. The following characterization of the Minkowski sum of P and Q may now be obtained: the vertices of $P \oplus Q$ are the Minkowski sums of *co-podal* pairs of vertices of P and Q. This characterization permits the computation of $P \oplus Q$ by rotating the calipers in the manner as shown in Fig. 5, where the pair p_i and q_j is a candidate pair of vertices to be summed. The subsequent candidate pair is obtained in O(1) time by rotating the calipers in a clockwise manner by the smaller of the angles θ_i and φ_j. Let $z_k = p_i \oplus q_j$ denote a vertex of $P \oplus Q$ that has been computed. Then the succeeding vertex $z_{k+1} = p_i \oplus q_{j+1}$ if $\varphi_j < \theta_i$, $z_{k+1} = p_{i+1} \oplus q_j$ if $\theta_i < \varphi_j$, and $z_{k+1} = p_{i+1} \oplus q_{j+1}$ if $\theta_i = \varphi_j$. Since there are no more than $2n$ vertices in $P \oplus Q$ it follows that O(n) time suffices to compute it.

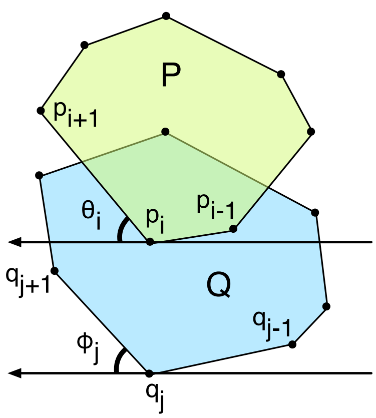

Figure 5. The Minkowski sum of two convex polygons.

2.6 The Convex Hull of Two Convex Polygons

There exist applications where it is required to compute the convex hull of two convex polygons. For example, the divide and conquer approach to computing the convex hull of a set of n points in the plane repeatedly (recursively) merges the convex hulls of two smaller subsets of the points [33]. A linear time merge step is sufficient to yield an algorithm for the convex hull of the set that runs in O(n log n) time. The convex hull of two convex polygons, P and Q, consists of three types of edges: edges of P, edges of Q, and edges that connect vertices of P with those of Q. The latter edges are called *bridges* (also *common tangents*). In Fig. 6 the dashed line L_B that supports polygons P and Q at vertices p_i and q_j, respectively, determines a bridge of the convex hull of P and Q.

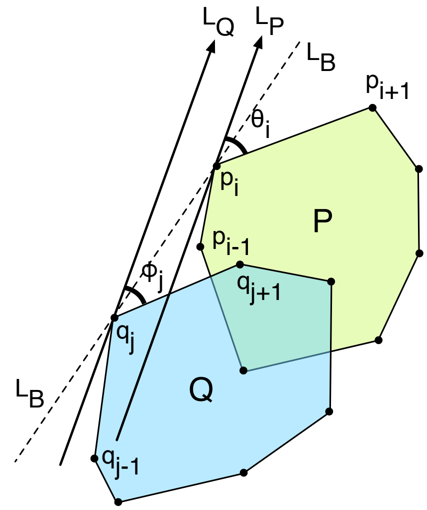

Figure 6. The convex hull of two convex polygons.

The convex hull of two convex polygons may therefore be computed by rotating clockwise two directed lines of support (one on each polygon) oriented in the same direction, such that each polygon is to the right of its line of support, as was done for the Minkowski sum problem. Whenever the two lines of support are not collinear (overlapping) one of them lies to the left of the other. For instance, in Fig. 6, line L_Q lies to the left of L_P. Later in the process line L_P will lie to the left of L_Q. This implies that the lines must, at some time during the rotation, completely overlap, and whenever they do so they identify a bridge. Thus, the convex hull of the two polygons may be obtained by rotating the calipers a full revolution, and at each step outputting the vertex of either P or Q, that lies on the leftmost supporting line. Since each vertex of P and Q is visited only once, and there are O(n) bridges, O(n) time suffices for the entire computation.

The common tangents between two convex polygons have been computed with the rotating calipers in the context of solving special cases of the travelling salesperson problem (TSP) in which there are nested convex obstacles [1].

2.7 Intersecting Two Convex Polygons

Computing the intersection of two convex polygons is a fundamental operation that occurs in many applications. For example, the divide and conquer approach to computing the intersection of a set of n half-planes, repeatedly merges the intersections of two smaller subsets of the half-planes, which are (perhaps unbounded) convex polygonal sets [34]. A linear-time algorithm for intersecting two convex polygons, that uses the *slab method*, was described by Michael Shamos in his thesis [40], which leads to an O(n log n) time algorithm for the half-plane intersection problem. An alternate O(n) time algorithm was proposed by O'Rourke [31]. A transparently clear O(n) algorithm along with an easy proof of correctness, that uses the rotating calipers was presented in [44]. The algorithm exploits the fact that if two convex polygons intersect, then there exists an intersection point corresponding to each bridge in the convex hull of the two polygons, as illustrated in Fig. 7, where the dashed line L_B identifies a bridge determined by vertices $q_j \in Q$ and $p_i \in P$, and x denotes the intersection point corresponding to this bridge. The algorithm in [44] first determines if the two polygons intersect. If they do then the rotating calipers are used to find the convex hull as described in Sub-section 2.6, after which for each bridge the corresponding intersection point is found by a simple step-down procedure along the convex chains searching for the two intersecting edges.

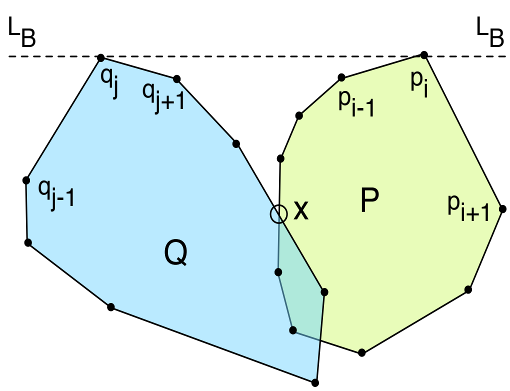

Figure 7. A bridge and its corresponding intersection point x of two intersecting convex polygons.

2.8 The Critical lines of Support of Two Convex Polygons

Given two disjoint convex polygons P and Q, the critical support lines are the two lines that separate P and Q, such that they are both support lines for P and Q. In Fig. 8 the two critical support lines, L_P and L_Q, are the dashed lines. One line contains vertices p_{i+1} and q_j, and the other contains p_{i-1} and q_{j-1}. Intuitively, the two critical support lines may be obtained by rotating any line that separates the two polygons, in clockwise and counterclockwise directions as much as possible, while maintaining the separability of the two polygons. Critical support lines find application to a variety of problems, some of which are described in the following subsections. For two convex polygons they can be computed in linear time using the rotating calipers in a manner similar to that of computing the maximum distance, by initially placing the two parallel lines oriented in opposite directions such that each polygon is to the right of its support line, as in Fig. 4. During the rotation phase of the calipers the critical support lines are detected each time that both support lines are collinear (overlap each other), as in Fig. 8.

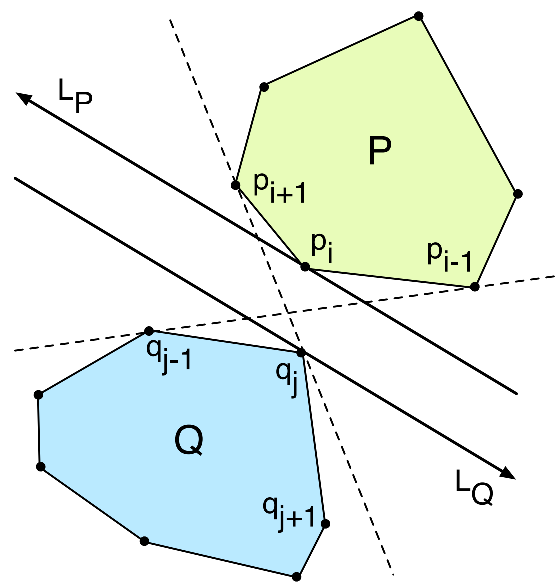

Figure 8. The two critical support lines (dashed) determined by two disjoint convex polygons.

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

2.9 The Widest Separating Strip Between Two Convex Polygons, and Machine Learning

In the context of pattern classification and machine learning, two polygons P and Q may be thought of as regions in a feature space that enclose points of the training data for a two-class discrimination problem. Any separating line then is a linear classification rule that can be used to classify future patterns (points) depending on whether they lie on one side or the other of this line. In order for the classifier to make more confident decisions it is desired to pick the separating line that is furthest from the two polygons. Such a line may be chosen as the centerline of the widest empty strip that separates the polygons. Such classifiers are referred to as large-margin classifiers [14], (also wide separation of sets [19]) and make up the geometric backbone of support vector machines [27]. For the case of two planar convex polygons the widest empty strip is determined by either one vertex from each polygon, or by a vertex of one polygon and an edge of the other. Furthermore, this strip may be detected when the support lines are oriented in between the directions of the two critical support lines, and may be found in O(n) time with the rotating calipers.

The widest empty strip problem is closely related to the problems of fitting lines to data [36], finding transversals of sets [4], and linear approximation of objects [37], [38], all of which have been solved efficiently using the rotating calipers.

2.10 The Grenander Distance Between Two Convex Polygons

Ulf Grenander [16] proposed that the distance between two convex polygons be measured by comparing the lengths of the connecting segments of their critical support lines, to the lengths of the polygonal chains spanned by these segments. To be more precise let the critical support lines be supported at p_i and p_r in P, and q_j and q_s in Q, and refer to Fig. 9. The supporting vertices of the critical support lines partition the convex polygons into two polygonal chains: the *inner chains* facing the intersection point of the support lines, and the complementary *outer chains*. The *supporting chords* of P and Q are the line segments that connect q_j to p_i and p_r to q_s, shown as bold red line segments in Fig. 9. The Grenander distance is defined as the sum of the lengths of the supporting chords, less the sum of the lengths of all the edges of the polygons that belong to the inner chains. Clearly, once the critical support lines have been computed with the rotating calipers, O(n) time suffices for the computation of the lengths of the chords and chains, and therefore the Grenander distance may be computed in O(n) time.

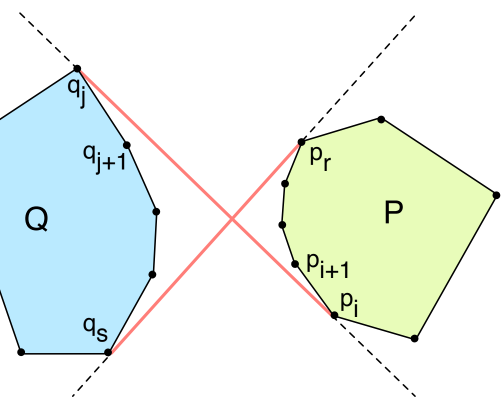

Figure 9. The Grenander distance between two convex polygons.

2.11 Optimal Strip Separation in Medical Imaging and Solid Modeling

In several contexts such as medical imaging it is required to construct a solid model by stitching parallel polygonal slices together. A problem arises during the interpolation when the solid object is bounded by a single contour in one slice, and two contours in the adjoining slice. The computational geometric problem that results is the following [3]. Given two linearly separable polygons P and Q, and a third convex polygon R, it is required to compute the separating strip between P and Q, that covers the largest area of R. Barequet and Wolfers [3] present a linear-time algorithm for computing this optimum strip using the rotating calipers. They also consider the case when the polygon R is not convex, but in this case the running time of their algorithm is quadratic in the size of the input.

2.12 Aperture Angle Optimization for Visibility Problems in Graphics and Computer Vision

In several disciplines such as computer graphics, computer vision, robotics, operations research, visual inspection, and accessibility analysis in the manufacturing industry, the notion of visibility is fundamental. Often models assume that a camera can see in all directions, in effect idealizing the camera's aperture angle to 360º. In a more realistic model the aperture angle is much smaller than 360º. Furthermore, if the camera is movable, it is desirable to compute the maximum and minimum aperture angles that the camera may need as it travels in a constrained space. Let *P* and *Q* be two disjoint convex polygons in the plane. For a given a point *x* in *P*, the aperture angle at *x* with respect to *Q* is defined as the angle of the cone with apex at *x*, that contains *Q*, and has its two rays that emanate from *x* tangent to *Q*. The critical lines of support play singular roles in computing the extreme aperture angles, and may be efficiently computed with the rotating calipers [7].

2.13 Wedge Placement Optimization Problems

Wedge placement optimization problems arise in several contexts such as visibility with bounded aperture angles, and layout design of parts in stock cutting for manufacturing. A wedge *W* may be thought of simply as an unbounded cone with a fixed angle θ at its apex. Given an *n*-vertex polygonal region, such as *R* in Fig. 10, we are interested in computing the entire region where a camera (the apex of the cone) with aperture angle θ, may be positioned so that it is as close as possible to *R*. Such a region is bounded by a concatenation of arcs (called the *wedge cloud*) determined by the apex of the cone as it travels around *R* maintaining contact with *R*. Fig. 10 shows three points on the cloud, x_i, x_j and x_k which are camera locations with a fixed aperture angle θ. The original rotating calipers can be generalized so that the lines of support form any fixed angle to each other. Teichmann [41] used such a generalization of the calipers to compute the wedge cloud of a convex polygon in O(*n*) worst-case time. Note that to maintain contact with the polygon *R*, a wedge must both rotate and translate.

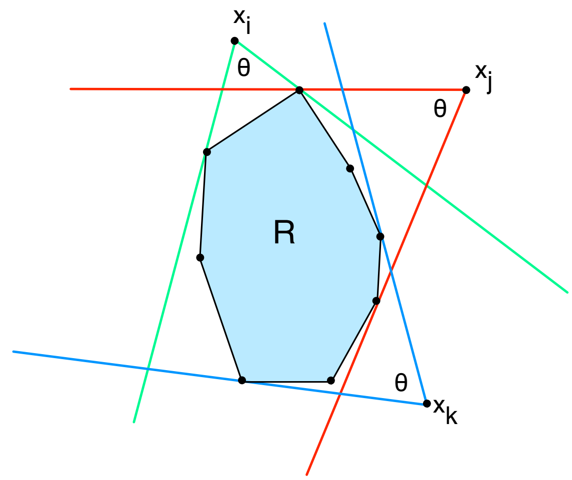

Figure 10. Wedge fitting and the *wedge cloud*.

2.14 Nonparametric Decision Rules

In the nonparametric discrimination problem we are given training data that belong to different classes, and it is desired to classify new incoming data into their respective classes. Jean-Paul Rasson and Grandville [35] proposed a geometric approach to the design of such a decision rule. Consider the two-dimensional, two-class problem, in which the classes are linearly separable, and refer to Fig. 11. As part of the training phase of the classifier, the convex hulls are computed and stored for each class. These convex hulls are the convex polygons *P* and *Q* in Fig. 11, colored light green and light blue, respectively. The decision rule for a new incoming pattern *X* is as follows. If *X* lies in *P* (or *Q*) it is classified as belonging to class *P* (or *Q*). If *X* lies outside both polygons it is classified to the class associated with the nearest polygon, where nearest is defined in terms of the area distance between *X* and each of the polygons. More precisely, the distance between *X* and *Q* is the absolute value of the difference between the area of polygon *Q* and the area of the convex hull of $Q \cup X$. Similarly, the distance between *X* and *P* is the absolute value of the difference between the area of polygon *P* and the area of the convex hull of $P \cup X$. These areas are colored dark blue and dark green, respectively. In this example the dark green area is smaller than the dark blue area, and therefore *X* would classified as belonging to class *P*. The four lines that connect *X* to the polygons are critical support lines between *X* and the

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

polygons, where X may be considered as a degenerate single-vertex polygon, and may be computed in linear time with the rotating calipers.

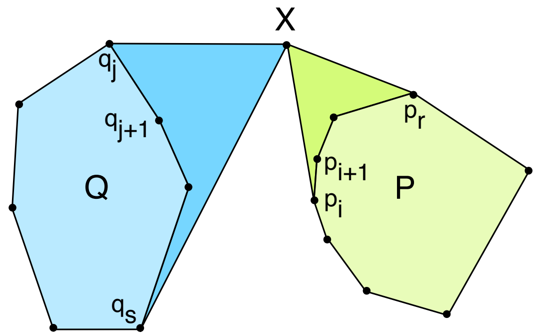

Figure 11. A geometric non-parametric decision rule.

2.15 Nice Triangulations and Quadrangulations of Planar Sets of Points

A convex polygonal *annulus* is the region in between two properly nested convex polygons, such as the blue shaded region consisting of the polygon Q less the interior of the green shaded polygon P shown in Fig. 12. This region admits a very simple triangulation by means of the rotating calipers [43]. A triangulation of a polygonal region is a partition of the region's interior into as many triangles as possible, obtained by inserting interior edges (diagonals) only between pairs of vertices, without allowing any edges to cross each other. For the annulus the calipers work in a similar manner as in the convex hull problem. Initially two parallel lines of support, oriented in the same direction, are constructed through the vertices of P and Q with lowest y-coordinates, so that the polygons lie to the right of the lines (Fig. 12). As the calipers are rotated clockwise, the pairs of vertices that come into contact with the lines of support are connected with an edge. The first few edges connected by this algorithm are shown in red. In general a region admits many triangulations, some of which have long edges, very acute triangles, or other properties deemed undesirable for some applications. One attractive property of the triangulation of the annulus obtained with the rotating calipers algorithm is that the triangulation tends to be nice, in the sense that the resulting triangles tend to be nearly regular. Furthermore, the method can be applied to more general problems such as obtaining nice triangulations of sets of points. One way of doing this is to first compute all the convex layers of the set [9], which yields a nested collection of annuli, each of which can be triangulated with the rotating calipers [42]. Another method involves first computing a spiral polygonal chain spanning the points [23], and then triangulating this spiral with the rotating calipers [42]. This latter triangulation also has the added nice property that it is *serpentine*, *i.e.,* its dual graph is a chain.

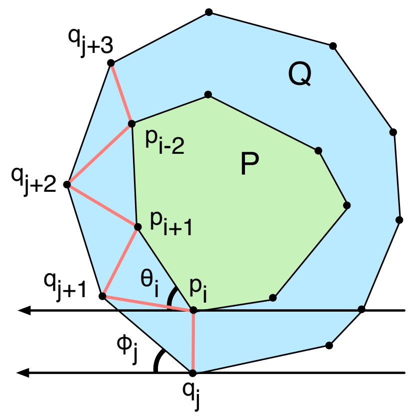

Figure 12. Triangulating a polygonal convex annulus.

3 CONCLUSION

This paper has focused on extensions of the rotating calipers, and their applications to a variety of geometric problems in the plane. The problem of computing the *minimum* distance between sets is conspicuously absent. This is because it appears difficult to crack it with the rotating calipers [48]. The rotating calipers have also been generalized to work on surfaces such as spheres and cones [17], and three-dimensional space, where supporting planes, rather than supporting lines, are rotated [8], [30]. However, a survey of this topic is beyond the scope of this paper, but will be forthcoming in the near future. Finally, the reader is referred to the web page and thesis of Hormoz Pirzadeh [32], for details of some proofs, and animation applets that help to visualize several algorithms that have been described in this paper. http://cgm.cs.mcgill.ca/~orm/welcome.html

4 ACKNOWLEDGEMENTS

This research was supported by a grant from the Provost's Office of New York University Abu Dhabi, through the Faculty of Science, in Abu Dhabi, The United Arab Emirates.

5 REFERENCES

[1] J. Abrahamson, A. Shokoufandeh, and P. Winter, "Euclidean TSP between two nested convex obstacles," Information Processing Letters, vol. 95, pp. 370-375, 2005.

[2] D. Avis, G. T. Toussaint, and B. K. Bhattacharya, "On the multimodality of distances in convex polygons," Computers and Mathematics with Applications, vol. 8, no. 2, pp. 153-156, 1982.

[3] G. Barequet, and B. Wolfers, "Optimizing a strip separating two polygons," Graphical Models and Image Processing, vol. 60, no. 3, pp. 214-221, 1998.

[4] B. K. Bhattacharya, J. Czyzowicz, P. Egyed, I. Stojmenovic, G. T. Toussaint, and J. Urrutia, "Computing shortest transversals of sets," International Journal of Computational Geometry and Applications, vol. 2, no. 4, pp. 417-436, 1992.

[5] B. K. Bhattacharya, and G. T. Toussaint, "Fast algorithms for computing the diameter of a finite planar set," The Visual Computer, vol. 3, no. 6, pp. 379-388, 1988.

[6] B. K. Bhattacharya, and G. T. Toussaint, "Efficient algorithms for computing the maximum distance between two finite planar sets," Journal of Algorithms, vol. 14, pp. 121-136, 1983.

[7] P. Bose, F. Hurtado-Diaz, E. Omaña-Pulido, J. Snoeyink, and G. T. Toussaint, "Some aperture-angle optimization problems," Algorithmica, vol. 33, pp. 411-435, 2002.

[8] C.-T. Chang, B. Gorissen, and S. Melchior, "Fast oriented bounding box optimization on the rotation group SO(3;R)," ACM Transactions on Graphics, vol. 30, no. 5, October 2011, Article 122.

[9] B. Chazelle, "On the convex layers of a planar set," IEEE Transactions on Information Theory, vol. 31, pp. 509-517, 1985.

[10] S. Das, P. P. Goswami, and C. S. Nandy, "Smallest k-point enclosing rectangle and square of arbitrary orientation," Information Processing Letters, vol. 94, pp. 259-266, 2005.

[11] S. Drazić, N. Ralević, and J. Zunić, "Shape elongation from optimal encasing rectangles," Computers and Mathematics with Applications, vol. 60, pp. 2035-2042, 2010.

[12] R. O. Duda, and P. E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

[13] H. Freeman and R. Shapira, "Determining the minimum-area encasing rectangle for an arbitrary closed curve," Communications of the A.C.M., vol. 18, pp. 409-413, 1975.

[14] Y. Freund, and R. E. Schapire, "Large margin classification using the perceptron algorithm," Machine Learning, vol. 37, no. 3, pp. 277-296, 1999.

[15] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," Information Processing Letters, vol. 1, pp. 132-133, 1972.

[16] U. Grenander, Pattern Synthesis, Springer-Verlag, New York, 1976.

[17] C. Grima, and A. Márquez, Computational Geometry on Surfaces, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.

[18] F. C. A. Groen, P. W. Verbeek, N. de Jong, and J. W. Klumper, "The smallest box around a package," Pattern Recognition, vol. 14, nos. 1-6, pp. 173-178, 1981.

[19] M. E. Houle, "Algorithms for weak and wide separation of sets," Discrete Applied Mathematics, vol. 45, pp. 139-159, 1993.

[20] M. E. Houle, and G. T. Toussaint, "Computing the width of a set," IEEE Transactions Pattern Analysis & Machine Intelligence, vol. 10, no. 5, pp. 761-765, 1988.

[21] K. Ichida, and T. Kiyono, "Segmentation of plane curves," Transactions of the Institute of Electronics and Communication Engineers of Japan, vol. 58-D, pp. 689-696, 1975.

[22] H. Imai, and M. Iri, "Polygonal approximations of a curve: Formulations and solution algorithms," in Computational Morphology, G. T. Toussaint, (Ed.), Amsterdam, The Netherlands: North-Holland, 1988, pp. 71-96.

[23] J. Iwerks, and J. S. B. Mitchell, "Spiral serpentine polygonization of a planar point set," In Encuentros de Geometria Computacional, (Hurtado Festschrift), A. Márquez, *et al*. (Eds.): LNCS 7579, 2012, pp. 146-154.

[24] Y. Kurozumi, and W. A. Davis, "Polygonal approximation by the minimax method," Computer Graphics and Image Processing, vol. 19, pp. 248-264, 1982.

[25] T. Lozano-Perez, "An algorithm for planning collision-free paths among polyhedral obstacles," Communications of the ACM, vol. 22, pp. 560-570, 1979.

[26] T. Lozano-Perez, "Spatial planning: A configuration space approach," IEEE Transactions on Computers, vol. 32, no. 2, pp. 108–120, 1983.

[27] M. E. Mavroforakis, and S. Theodoridis, “A geometric approach to support vector machine (SVM) classification,” IEEE Transactions on Neural Networks, vol. 17, no. 3, pp. 671-682, 2006.

[28] J. S. B. Mitchell, and V. Polishchuk, “Minimum-perimeter enclosures,” Information Processing Letters, vol. 107, pp. 120-124, 2008.

[29] D. Mount, “The densest double-lattice packing of a convex polygon,” J. E. Goodman, R. Pollack, W. Steiger, (Eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 6, 1999, pp. 245-262.

[30] J. O'Rourke, “Finding minimal enclosing boxes,” International Journal of Computer and Information Science, vol. 14, pp. 183-199, 1985.

[31] J. O’Rourke, “A new linear algorithm for intersecting convex polygons,” Computer Graphics and Image Proccessing, vol. 19, pp. 384-391, 1982.

[32] H. Pirzadeh, Computational Geometry with the Rotating Calipers, M.Sc. Thesis, School of Computer Science, McGill University, Montreal, Canada, November 1999.

[33] F. P. Preparata, and S. Hong, “Convex hulls of finite sets of points in two and three dimensions,” Communications of the ACM, vol. 20, pp. 87-93, 1977.

[34] F. P. Preparata, and D. E. Muller, “Finding the intersection of n half-spaces in time O(n log n),” Theoretical Computer Science, vol. 8, no. 1, pp. 45-55, 1979.

[35] J. P. Rasson, and V. Granville, “Geometrical tools in classification,” Journal of Computational Statistics and Data Analysis, vol. 2, pp. 105-123, 1996.

[36] C. Rey, and R. Ward, “On determining the on-line minimax linear fit to a discrete point set in the plane,” Information Processing Letters, vol. 24, pp. 97-101, 1987.

[37] J.-M. Robert, and G. T. Toussaint, “Linear approximation of simple objects,” Proceedings 9th Annual Symposium on Theoretical Aspects of Computer Science (STACS'92), Cachann France, February 1992, A. Finkel and M. Jantzen (eds.), Lecture Notes in Computer Science #577, pp. 233-244.

[38] J.-M. Robert, and G. T. Toussaint, “Computational geometry and facility location,” Proceedings of the International Conference on Operations Research and Management Science, Manila, The Philippines, December 11-15, 1990, pp. B-1 to B-19.

[39] P. L. Rosin, “Measuring shape: ellipticity, rectangularity, and triangularity,” Machine Vision and Applications, vol. 14, pp. 172-184, 2003.

[40] M.I. Shamos, “Computational geometry”, Ph.D. thesis, Yale University, 1978.

[41] M. Teichmann, Wedge Placement Optimization Problems, M.Sc. Thesis, School of Computer Science, McGill University, October 2909.

[42] G. T. Toussaint, “Quadrangulations of planar sets,” Proceedings of the 4th International Workshop on Algorithms and Data Structures (WADS'95), Kingston, Canada, August 16-18, 1995, pp. 218-227.

[43] G. T. Toussaint, “New results in computational geometry relevant to pattern recognition in practice,” In Pattern Recognition in Practice II, E. S. Gelsema and L. N. Kanal, (Eds.), North-Holland, Amsterdam, The Netherlands, 1986, pp. 135-146.

[44] G. T. Toussaint, “A simple linear algorithm for intersecting convex polygons,” The Visual Computer, vol. 1, pp. 118-123, 1985.

[45] G. T. Toussaint, “Complexity, convexity, and unimodality,” International Journal of Computer and Information Sciences, vol. 13, no. 3, pp. 197-217, 1984.

[46] G. T. Toussaint, “Solving geometric problems with the rotating calipers,” Proceedings of IEEE MELECON'83, Athens, Greece, May 1983, pp. A10. 02/1-4.

[47] G. T. Toussaint, “Pattern recognition and geometrical complexity”, Proceedings of the Fifth International Conference on Pattern Recognition, Miami Beach, December 1980, pp. 1324-1347.

[48] G. T. Toussaint, and B. K. Bhattacharya, “Optimal algorithms for computing the minimum distance between two finite planar sets,” Pattern Recognition Letters, vol. 2, pp. 79-82, 1983.

[49] G. T. Toussaint, and J. A. McAlear, “A simple O(n log n) algorithm for finding the maximum distance between two finite planar sets,” Pattern Recognition Letters, vol. 1, pp. 21-24, 1982.

[50] J. Žunić and P. L. Rosin, “A new convexity measure for polygons,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 7, pp. 923-934, 2004.

[51] J. Žunić and P. L. Rosin, “Rectilinearity measurements for polygons,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 1193-1200, 2003.

[52] J. Žunić and P. L. Rosin, and L. Kopanja, “On the orientability of shapes,” IEEE Transactions on Image Processing, vol. 15, no. 11, pp. 3478-3487, 2006.