**Placing the Largest Similar Copy of a Convex Polygon Among Polygonal Obstacles***

L. Paul Chew
Klara Kedem

TR 89-964
January 1989

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*This work was supported by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research Contract N00014-88-K-0591, ONR Grant N00014-86-K-0281, NSF Grant DMC-86-17355, and MSI Grant U03-8300.

# Placing the Largest Similar Copy of a Convex Polygon Among Polygonal Obstacles

*L. Paul Chew & Klara Kedem*

Department of Computer Science
Cornell University
Ithaca, NY 14853

*ABSTRACT*

Given a convex polygon $P$ and an environment consisting of polygonal obstacles, we find the largest similar copy of $P$ that does not intersect any of the obstacles. Allowing translation, rotation, and change-of-size, our method combines a new notion of Delaunay triangulation for points and edges with the well-known functions based on Davenport-Schinzel sequences producing an almost quadratic algorithm for the problem. Namely, if $P$ is a convex $k$-gon and if $Q$ has $n$ corners and edges then we can find the placement of the largest similar copy of $P$ in the environment $Q$ in time $O(k^4 n\ \lambda_4(kn) \log n)$, where $\lambda_4$ is one of the almost-linear functions related to Davenport-Schinzel sequences. If the environment consists only of points then we can find the placement of the largest similar copy of $P$ in time $O(k^2 n\ \lambda_3(kn) \log n)$.

Work on this paper has been supported by Defense Advanced Research Project Agency grant N0014-88-K-0591, Mathematical Sciences Institute grant U03-8300, National Science Foundation grant DMC-86-17355, and Office of Naval Research grant N00014-86-K-0281.

## 1. Introduction

Given a pattern and a piece of material containing faults, the goal is to cut out a piece, geometrically similar to the pattern, that does not contain any of the faults. For instance, we might wish to cut the largest possible square from a piece of sheet metal that has some small holes in it. We show that if the pattern is a convex polygon and if the faults can be represented by polygonal boundaries then such a problem can be solved in time $O(k^4 n\ \lambda_4(kn) \log n)$, where $k$ is the size of the pattern, $n$ is the size of the environment, and $\lambda_4$ is one of the almost-linear functions related to Davenport-Schinzel sequences [ASS].

A number of authors have studied related placement problems using various assumptions about the object to be placed, the motions allowed, and the environment the object is to be placed within. We describe some of these results in the following paragraphs. Assume $P$ is a $k$-gon that we wish to place in a polygonal environment $Q$ of size $n$.

A number of papers [SCKLPS, Ch, AB] allow $P$ to both translate and rotate. In [SCKLPS] the polygon containment problem is solved in time $O(kn\ \lambda_6(kn) \log kn)$ where $P$ is convex and $Q$ is a closed, not necessarily simple, polygonal region of $n$ edges and corners. Chazelle [Ch] studied the problem for the case where $P$ and $Q$ are arbitrary simple polygons and showed that the naive algorithm takes time $O(k^3 n^3 (k+n) \log (k+n))$. A more restricted case of the polygon containment problem, in which both $P$ and $Q$ are convex was also studied by Chazelle, who solved this case in time $O(kn^2)$. Avnaim and Boissonnat [AB] present an algorithm where both $P$ and $Q$ are non-convex, possibly-not-connected polygons that runs in time $O(k^3 n^3 \log kn)$ and is optimal in the worst case.

A different version of the polygon containment problem was studied by Fortune [Fo1], (cf. also [LS] and [CD]): find the largest homothetic copy of $P$ inside $Q$. In other words, translation and change-of-size are allowed, but rotation is not. This problem is solved for a convex $P$ and an arbitrary polygonal region $Q$ in time $O(kn \log kn)$ by constructing a generalized Voronoi diagram of $Q$ under a convex distance function induced by $P$.

In this paper, we allow $P$ to **translate, rotate,** and **change-in-size,** thus solving the most general motion problem. Our polygon $P$ is a convex $k$-gon and $Q$ is an arbitrary polygonal environment consisting of $n$ vertices and edges. The problem is that of finding the largest similar copy of $P$ that can be placed in $Q$. We solve it in time $O(k^4 n \, \lambda_4(kn) \log n)$, and if the environment consists only of $n$ points (no edges) in time $O(k^2 n \, \lambda_3(kn) \log n)$.

The correspondence between placement problems and motion planning problems has been noted before ([AB], [Ch], [LS] and many others). We exploit the results of this paper to achieve an $O(k^4 n \, \lambda_4(kn) \log n)$ algorithm for planning a general motion of a convex polygonal body amidst polygonal obstacles. Our time bound does not differ by much from the one achieved by Kedem and Sharir [KS], ($O(kn \, \lambda_6(kn) \log kn)$) for this problem, even though their algorithm provides a motion plan in which the object is almost constantly touching the obstacle walls while our algorithm provides a path with high-clearance. Our results on this problem will be described in a forthcoming paper.

The algorithm we develop, although it involves some constant-time subproblems that can be tedious to solve, is fairly straightforward. As is often the case, the analysis is more complicated than the algorithm itself.

We start the algorithm by creating an initial *Edge Delaunay Triangulation* of the polygonal environment using our convex polygon ($P$), at some fixed orientation $\theta$, as the distance function. The Edge Delaunay Triangulation has three types of edges: simple edges, that connect corners or points of $Q$; wedges, that connect points or corners of $Q$ to sides of $Q$; and ledges, that connect pairs of $Q$ sides. We treat the above features as *generalized triangulation edges*. The exact definition of this new notion of the Delaunay triangulation is dealt with in detail in section 2.

We observe that in most cases, as $\theta$ varies just slightly, the combinatorial structure of the triangulation does not change. For each generalized edge in this triangulation, we determine at which angle $\theta$ this edge will cease to be valid due to interaction with its immediate neighbors. This can be determined by checking the two triangles directly adjacent to the edge. We place all the generalized edges in a priority queue, ordered by the angle $\theta$ at which they are to be eliminated. At each succeeding stage of the algorithm, we determine which edge will be the next to disappear as $\theta$ increases. We then eliminate that edge from the Delaunay triangulation, add the appropriate new edge, and update the priority queue information for the new edge and its neighbors. Note that a new edge can change the priority for its neighbors.

As we update the triangulation, we determine, for each triangle, the range of $\theta$ where the triangle exists. With this information we can determine, in constant time, the maximum size of $P$, for the given range of $\theta$, that can be placed in contact with the three obstacles that induced the triangle. As triangles are processed we remember the placement for the largest size of $P$ so far. Thus, by the end of the algorithm, we have the required placement and orientation of the largest similar copy of $P$.

It is easy to see that the time for this algorithm is bounded by a constant times (the number of edges eliminated as $\theta$ changes) times (the time to do a priority queue operation). A priority queue can be implemented to run in time $O(\log m)$ per operation where $m$ is the maximum number of items in the queue. Since there are never more than $O(n)$ edges in the queue at any one time, each priority queue operation takes time $O(\log n)$.

The main contribution of this paper is an analysis proving a bound on the number of edge changes in the edge Delaunay triangulation as $\theta$ changes. In outline, our analysis goes as follows. Changes in the combinatorial structure of the triangulation occur only at *critical orientations* which we analyze in a manner somewhat similar to [KS]. We choose a corner of $Q$ and a corner of $P$ and attach these corners to make a *hinge*. Call this hinge $H$. We define a family of functions for this hinge, one function for each possible *contact* in $Q$. (A contact is a vertex of $P$ against an edge of $Q$, or an edge of $P$ against a vertex of $Q$.) For contact $C$, $f_C(\theta)$ is the smallest possible size of $P$ such that $P$, at angle $\theta$ and retaining hinge $H$, maintains contact $C$. We show that the lower envelope of this family of functions is closely related to the Delaunay triangulation. We argue that the number of *breakpoints* on the lower envelope is $O(\lambda_4(kn))$. Hence, over all $kn$ possible hinges the total number of breakpoints is $O(kn\ \lambda_4(kn))$. The number of critical orientations is shown to be proportional to the number of breakpoints (the exact proportion depends on $k$).

The paper is organized as follows. In section 2 we discuss the Edge Delaunay Triangulation. In section 3 we analyze the size functions and their lower envelopes. Section 4 deals with the critical orientations where changes occur in the Delaunay triangulation, Conclusions and further research are briefly discussed in Section 5.

## 2. Edge Delaunay Triangulations

Just as the standard Delaunay triangulation is the dual of the standard Voronoi diagram the *Edge Delaunay Triangulation* (EDT) is the dual of the *Edge Voronoi Diagram* (EVD). In the literature, the EVD is sometimes called a *generalized Voronoi diagram,* but there are so many different possible generalizations of the Voronoi diagram that it seems worthwhile to use a name that indicates which generalization is meant. For an EVD, the initial data consists of points and line segments between points, while for a standard Voronoi diagram, the initial data consists only of points. The initial data items are called *sources*. For the EVD there are two kinds of sources: open line segments and points (including the endpoints of the line segments). The Voronoi region for a given source is, as for the standard Voronoi diagram, the portion of the plane that is closer to the given source than to any other source. The resulting Voronoi diagram divides the plane into regions with boundaries consisting of line segments and segments of parabolas. An $O(n \log n)$ divide-&-conquer algorithm for building the EVD has been developed by Yap [Ya]. Fortune's sweepline technique [Fo2] can also be used to build the EVD in $O(n \log n)$ time.

The following intuitive method for building an EVD can be used to gain an understanding of its dual, the EDT. Approximate each line-segment source by placing $p$ points, equally spaced, along the line-segment. Determine the standard Voronoi diagram using the original source points and the approximating points; the original line-segment sources are ignored. Voronoi boundaries between points along the same line segment are colored red; all other Voronoi boundaries are colored black. In the limit, as $p$ increases, the black boundaries give us the EVD.

Note that for each value of $p$ there is a well-defined Delaunay triangulation. Intuitively, the EDT is simply the limit of these Delaunay triangulations as $p$ goes to infinity. (See Fig. 1.) This might seem to imply that the EDT requires infinitely many edges. Fortunately, as we explain below, such prospective edges appear in simply defined groups, so that entire groups of such edges can be represented by a simple triangle or trapezoid.

For an EDT, there are three kinds of connections between sources: a simple *edge*, a connection between two point sources; a *wedge*, a connection between a point source and a line-segment source; and a *ledge*, a connection between two line-segment sources. A simple edge is just a straight line between two points. A wedge is a triangle (possibly a degenerate triangle, i.e., a line-segment) with one vertex at a point and the other two vertices on a line segment. Intuitively, a wedge is an infinite set of edges connecting a point source to an infinite set of approximating points on a line-segment source. A ledge is a trapezoid (possibly a degenerate trapezoid, i.e., a triangle or a line-segment). The two parallel sides connect a line-segment source (call it $a$) to another line-segment source (call it $b$). The remaining sides are formed by continuous portions of $a$ and $b$. Intuitively, a ledge is an infinite set of (parallel) edges connecting infinitely many approximating points on $a$ to infinitely many approximating points on $b$. Note that because ledges are trapezoids, an EDT is actually not quite a triangulation.

Note that our EDT differs from another kind of Delaunay triangulation for edges that has appeared in the literature. This other type of Delaunay triangulation has been called a generalized Delaunay triangulation [LL] or a constrained Delaunay triangulation [Che]. Very roughly, the constrained Delaunay triangulation is a cross between the standard Delaunay triangulation and the visibility graph. The EDT is the dual of the EVD while the constrained Delaunay triangulation is not [LL].

The remainder of this section consists of a more careful definition of what is meant by an EDT. We start by defining a *D−graph*, a first approximation to the EDT of source graph $S$. Intuitively, a D-graph is just $S$ with some extra edges and with some extra vertices added to line segments of $S$.

**Definition.** Let $S$ be a straight-line planar graph (called the *source graph*). A straight-line planar graph $G$ is called a *D−graph* of $S$ if

(1) each vertex of $S$ appears in $G$, and

(2) each edge of $S$ appears in $G$, possibly with extra vertices (called *D−vertices*) placed along it.

The edges and vertices in $G$ that correspond to edges and vertices of $S$ are called *source edges* and *source vertices,* respectively. Edges of $G$ that do not correspond to edges in $S$ are called *D−edges*. A region of $G$ is called a *wedge* if it is a triangle with a source edge on one side and a source vertex as the opposite vertex. A region of $G$ is called a *ledge* if it is a quadrilateral with source edges on opposite sides. The quadrilateral can be degenerate with one edge of zero length.

The following definitions restrict the D-edges to make them into Delaunay edges. In the following definitions, the meaning of *circle* depends on the distance function being used. For instance, an $L_1$ *circle* is a square tipped at 45°; an $L_2$ *circle* is a standard circle; for a convex distance function, a *circle* is a shape similar to and at the same orientation as the distance-defining convex shape.

**Definition.** A D-graph, $G$, has the *empty circle property* if for each D-edge, $e$, there is a *circle*, $C$, such that

(1) the endpoints of $e$ are on the boundary of $C$, and

(2) $C$ is empty (no source vertex or source edge of $G$ intersects the interior of $C$).

**Definition.** Let $G$ be a D-graph of the source graph $S$. $G$ is called an *Edge Delaunay Triangulation (EDT)* of $S$ if all of the following hold:

(1) $G$ has the empty circle property;

(2) adjacent wedges of G do not have identical sources;

(3) adjacent ledges of G do not have identical sources; and

(4) $G$ is maximal in the sense that no new D-edges can be added without violating one of these restrictions.

Note that if there is an empty *circle* that goes through 4 sources then there is more than one possible EDT. In this case all the possible EDTs are equally valid.

It is easy to show that if the number of vertices and edges in the source graph is $n$ then there are $O(n)$ vertices and edges in the EDT (to see this, imagine shrinking each line-segment source into a point). The EDT can be built in time $O(n \log n)$ by first building the EVD. For our purposes, since the rest of our algorithm runs in time roughly $O(n^2 \log n)$, it may be worthwhile to use a simpler incremental algorithm that runs in time $O(n^2)$.

## 3. Expansion Functions and Their Lower Envelopes

Let $P$ be a convex polygon having $k$ sides, and let $Q$ be a two-dimensional environment having polygonal boundaries with a total of $n$ corners. We assume there is a reference point $p$ and a reference vector $pq$ in $P$, such that a placement of $P$ in the plane can be represented by the quadruple $(x, y, \theta, \delta)$, where $(x, y)$ are the coordinates of $p$, $\theta$ is the orientation of $pq$, and $\delta$ is the *expansion factor* of $P$.

For the rest of this section we discuss a family of functions associated with placing a particular corner of $P$ on a particular corner of $Q$. We call this pair of corners the *hinged corner* $H$. We define a *contact pair* $C$ to be a *point* contact pair if a side of $P$ touches a corner of $Q$, and a *side* contact pair if a corner of $P$ touches a side of $Q$. We define an *expansion function* $E_{HC}(\theta)$ to be the minimal expansion factor of $P$ at orientation $\theta$ so that $P$ is hinged at $H$ and touches $Q$ at the contact pair $C$. (See Fig. 2).

The functions have the form

$$E_{HC_1}(\theta) = \frac{c_1}{\cos(\theta + c_2)} \qquad E_{HC_2}(\theta) = c_3 \cos(\theta + c_4)$$

when $C_1$ is a side contact pair, and $C_2$ is a point contact pair and $\{c_i, i=1 \cdots 4\}$ are constants that depend on the geometry of $P$ and $Q$ at the hinged corner and at the contact pair. The domain of definition of $E_{HC}$ is a continuous angular interval $< \pi$; this follows from convexity of $P$.

The lower envelope for one hinged corner $H$ is defined as

$$\Psi_H(\theta) = \min_C E_{HC}(\theta).$$

If a function $E_{HC}$ is on the lower envelope at some $\theta$ then there is an EDT edge between $Q$'s hinged corner and the element of $Q$ from $C$; this follows from the definition of EDT. We call this EDT edge a *reported* edge (reported as being on the lower envelope). A *breakpoint* on the lower envelope occurs where $E_{HC_1}$ and $E_{HC_2}$ intersect. It happens when $P$ touches simultaneously the hinged corner of $Q$ and both the elements of $Q$ from $C_1$ and $C_2$.

To estimate the number of breakpoints along $\Psi_H$ we extend the domain of definition of the functions over the complete range $[0,2\pi)$ of $\theta$ by (as in [LS]) extending $E_{HC}$ leftwards from $\theta_1$ along a ray of some very large negative slope, and rightwards from $\theta_2$ by a ray of some very large positive slope.

**Proposition 3.1:** Two expansion functions intersect each other in at most 4 points.

**Proof:** It is easy to see that if both $E_{HC_1}$ and $E_{HC_2}$ involve point contact pairs, or if both involve side contact pairs, then they intersect in at most one point in an interval shorter than $\pi$. The ray extensions of these functions may add up to two more intersection points giving a total of 3 possible intersections. If the functions involve both a side contact pair and a point contact pair then the functions, not including the ray extensions, intersect at most twice. To see this, assume $C_1$ is the point contact pair and $C_2$ is the side contact pair. Choose one point on $B$ to be the reference point. We draw the expansion functions $E_{HC_1}$ and $E_{HC_2}$ using polar coordinates (see Fig. 3). As we vary $\theta$, maintaining touch with the contact pair $C_1$, the reference point draws a circular arc. Contact $C_2$ causes the reference point to draw a straight line (proof by similarity of triangles). The arc and the line can intersect at most twice, the ray extensions can add up to 2 more intersections, giving a total of up to 4 intersections. □

**Proposition 3.2:** The number of breakpoints on the lower envelope $\Psi_H$ is $O(kn\ \lambda_4(kn))$. If $Q$ consists only of points then the number of breakpoints on $\Psi_H$ is $O(kn\ \lambda_3(kn))$.

**Proof:** Since each pair of expansion functions for a single hinge can intersect at most 4 times, it follows from the definition of $\lambda$ (see, for instance, [At], [Ass], or [SCKLPS]) that the size of the lower envelope for this hinge is $O(\lambda_4(kn))$. Thus, over all possible hinges, the number of lower-envelope breakpoints is $O(kn\ \lambda_4(kn))$. If $Q$ consists only of points then the functions involve only point contact pairs, and the number of breakpoints on the lower envelope for one hinge is $O(\lambda_3(kn))$ and the total number of breakpoints is $O(kn\ \lambda_3(kn))$. □

## 4. Critical Orientations

In this section we prove a bound on the number of critical orientations that can occur as $\theta$ varies. By definition, a *critical orientation* is an angle at which the combinatorial representation of the EDT changes. We make use of the functions defined in the previous section. For a fixed $\theta$, these functions report, by means of the lower envelopes, some, but not all of the edges in the EDT; the lower envelopes define a subgraph of the EDT consisting of the *reported* edges. We show that the number of changes in the EDT is bounded by a constant that depends on $k$, times the number of changes in this subgraph. We know, from the previous section, that the number of changes in this subgraph is $O(kn\ \lambda_3(kn))$ for the points case and $O(kn\ \lambda_4(kn))$ for the general case.

In order to illuminate the main ideas behind our technique, we initially restrict ourselves to the special case in which the environment $Q$ consists only of points. We start with a lemma showing the relationship between the reported edges and the full EDT.

**Lemma 4.1:** If $Q$ consists only of points then, for a fixed $\theta$, every EDT edge is either a reported edge or a diagonal in a convex $l$-gon, $l \leq k$, whose sides are reported EDT edges.

**Proof:** Let $p, q \in Q$ be the ends of an arbitrary EDT edge (see Fig. 4). By the definition of the EDT there is an empty circle (in the shape of $P$) that touches points $p$ and $q$ making contacts $C_p$ and $C_q$ respectively. In an attempt to bring $P$ to a position where a corner of $P$ touches either $p$ or $q$ we slide $P$, maintaining contacts $C_p$ and $C_q$, while shrinking or expanding $P$ as necessary. If we get to a position where a corner of $P$ touches $p$ or $q$, then, by definition, $pq$ is a reported edge and we are done. If we do not succeed in touching a corner of $P$ to either $p$ or $q$ then there must be points in both translational directions that stop $P$ (i.e., some side $s_r$ of $P$ touches a stopping point $r \in Q$ creating contact $C_r$). There are now two new pairs of contacts, $C_p$ and $C_r$, and $C_q$ and $C_r$. We continue the process with each of these new pairs. The process terminates because if two contacts correspond to adjacent sides of $P$ then $P$ can be translated and shrunk, while maintaining the contacts, to show that the corresponding edge is a reported edge of the EDT. Thus we get at most $k$ stopping points that describe a convex polygon.

□

Lemma 4.1 implies that the subgraph of the EDT formed by the reported edges consists of *cells* of size at most $k$.

To study the number of changes in the EDT, we examine one such change in detail. A change occurs whenever there is a free placement of $P$ that makes 4 contacts with obstacles of $Q$. These 4 contacts correspond to two prospective EDT edges, each edge connecting a pair of contacts that are opposite each other. These are the edges that are changing at this critical orientation; as $\theta$ increases one edge is eliminated and the other replaces it.

**Theorem 4.1:** For $Q$ consisting only of points, the number of critical orientations of the EDT is $O(k^2 n\ \lambda_3(kn))$.

**Proof:** Assume first that $P$ is a quadrilateral. A change in the EDT corresponds to either a change within a cell or a change of a cell boundary. The changes of a cell boundary are reported by the breakpoints. We will estimate the number of changes within a cell. We define the *life-span* of a cell to be the range of $\theta$ where there is no change in the cell's boundary. We want to count the number of times that the diagonals of the cell interchange in the EDT during the life-span of the cell. By using similarity and the sine law, we can determine the number of orientations at which a shape similar to $P$ touches all 4 stopping points that bound the cell. For a given set of 4 stopping points there are either no such orientations, one such orientation, or $P$ can be made to touch at all orientations (infinitely many such orientations). In the first case there is no change of diagonals, and in the second there is exactly one such change. In the last case, if $P$ touches all four points at every orientation, then either choice of edge makes a valid EDT; thus, in this case, no change is necessary (i.e., this is not really a critical orientation). Thus, for any cell, if there is a diagonal change then the old diagonal disappears and does not reappear during the life-span of the cell.

A similar statement holds when $P$ is a $k$-gon. In this case each cell is also a $k$-gon referred to as a $k$-cell. For any pair of diagonals we can restrict our attention to the corresponding four boundaries of the $k$-cell. We can then use the technique outlined for the 4-cell above to show that if there is a diagonal change then the old diagonal does not reappear during the life-span of the cell.

As we show above, the total number of breakpoints is $O(kn\ \lambda_3(kn))$. As $\theta$ changes, each breakpoint of the lower envelope of the expansion functions affects up to two cells, which in turn can contribute up to $k$ new diagonals each. Thus, the total number of critical orientations is $O(k^2 n\ \lambda_3(kn))$. □

The following lemma applies to the general case in which the environment $Q$ consists of points and line segments.

**Lemma 4.2:** For a fixed $\theta$, every EDT edge is either a reported edge or a diagonal in a convex $l$-gon, $l \le 3k$, whose sides are either reported EDT edges or segments of source edges.

**Proof:** Let $p, q \in Q$ be the ends of an arbitrary EDT edge. (In this case $p$ and $q$ can be either a source point or a portion of a source edge.) As before there is an empty circle (in the shape of $P$) with contacts $C_p$ and $C_q$ (see Fig. 5). We try to bring $P$ to a position where there is a hinged corner by sliding $P$ while maintaining contacts $C_p$ and $C_q$, expanding or shrinking $P$ as necessary. We argue that either edge $pq$ is reported or the sliding of $P$ is stopped. A stop can be either a point $r \in Q$ that stops a side of $P$, or an edge $r \in Q$ that stops a corner of $P$. We continue the process with the two new pairs of contacts. If two contacts are adjacent (on $P$) then they must correspond to a reported edge.

The number of $Q$-edges that can stop vertices of $P$ is bounded by $k$, and the number of $Q$-points that can stop sides of $P$ is also bounded by $k$. Up to two EDT edges emanate from each such $Q$-point, either simple edges or wedges (not ledges). Therefore $l \le 3k$. □

**Remarks:**

1. Note that to preserve the convexity of a cell we consider the innermost boundary of a wedge to be the boundary of a cell, not the whole wedge.

2. Note that ledges are always unreported EDT edges.

**Theorem 4.2:** The number of critical orientations of the EDT is $O(k^4 n \ \lambda_4(kn))$.

**Proof:** Again we consider pairs of EDT unreported edges. By trigonometrical calculations it can be shown that no pair of such generalized diagonals can interchange more than four times. Observe one cell in the arrangement, and look at the contribution of one change in the boundary of the cell to the number of the unreported diagonals. If one boundary element is added to the cell it can add $O(k)$ new diagonals, which in turn can interact with $O(k^2)$ other possible unreported diagonals. A new diagonal and another diagonal can exchange at most four times in the EDT. Hence one cell change can contribute $O(k^3)$ changes in the EDT. Since there are $O(kn \ \lambda_4(kn))$ cell changes, the number of critical orientations is $O(k^4 n \ \lambda_4(kn))$. □

## 5. Conclusions and further research

We have developed a bound on the number of changes that can occur in an Edge Delaunay Triangulation as its distance-defining convex shape is rotated. A relatively straightforward algorithm for finding all the different EDTs that occur is outlined in the Introduction. The bound we have developed implies that the algorithm runs in time $O(k^4 n \ \lambda_4(kn) \log n)$ where $k$ is the size of the distance-defining convex shape and $n$ is the size of the polygonal environment. Note that the big cannons used to develop the bound are not used within the algorithm itself.

Once a representation for all the different EDTs is available it is simple to solve our placement problem. (Recall that our goal is to find the largest similar copy of a convex polygon $P$ that fits within a polygonal environment.) Using $P$ as the distance-defining shape, we determine all the changes in the EDT as $P$ is rotated. Each EDT triangle corresponds to a continuous set of 3-contact placements. Using some geometric calculations, it can be shown that there are at most three maximal placements of $P$ for a given triangle (restricted to the lifetime of the triangle). Triangle lifetimes can be determined as the EDT changes are discovered. Thus, for this technique, the time to place the largest similar copy of $P$ is determined by the time needed to find all the changes in the EDT as $P$ is rotated.

We suspect that too many factors of $k$ appear in our bound on the number of changes that can occur in the EDT. It seems likely that the correct time bound is $O(kn\lambda_s(kn))$ for both the special case in which $Q$ consists of points (for which $s=3$), and for the more general case, where $Q$ consists of points and edges (for which $s=4$). An improvement in this bound would also improve the time bound for the motion planning algorithm mentioned in the Introduction.

## References

[AB] F. Avnaim and J.D. Boissonnat, Polygon placement under translation and rotation, *5th Annual Symp. on Theoretical Aspects of Computer Science*, Lecture Notes in Comp. Science 294, Springer-Verlag, New-York, 1988, pp. 322-333.

[ASS] P. Agarwal, M. Sharir and P. Shor, Sharp upper and lower bounds for the length of general Davenport-Schinzel sequences, to appear in *J. Combinatorial Theory, Series A*.

[At] M. J. Atallah, Dynamic computational geometry, *Proc. 24th IEEE Symp. on Foundations of Computer Science*, 1983, pp. 92-99.

[CD] L.P. Chew and R.L. Drysdale, Voronoi diagrams based on convex distance functions, *Proc. ACM Symposium on Computational Geometry*, 1985, pp. 235-244.

[Ch] B. Chazelle, The polygon containment problem, in *Advances in Computing Research, Vol. I: Computational Geometry*, (F.P. Preparata, Ed.), JAI Press, Greenwich, Conneticut (1983), pp. 1-33.

[Che] L.P. Chew, Constrained Delaunay Triangulations, to appear in *Algorithmica.*

[Fo1] S. Fortune, Fast algorithms for polygon containment, *Proc. 12th International Colloquium on Automata, Language and Programming,* Lecture Notes in Comp. Science 194, Springer-Verlag, New-York, 1985, pp. 189-198.

[Fo2] S. Fortune, A sweepline algorithm for Voronoi diagrams, *Algorithmica,* 2 (1987), pp. 153-174.

[KS] K. Kedem and M. Sharir, An efficient motion planning algorithm for a convex polygonal object in 2-dimensional space, to appear in *Discrete and Computational Geometry,* 1988.

[LL] D.T. Lee and A.K. Lin, Generalized Delaunay Triangulation for Planar Graphs, *Discrete and Computational Geometry,* 1 (1986), pp. 201-217.

[LS] D. Leven and M. Sharir, Planning a purely translational motion for a convex polygonal object in two dimensional space using Generalized Voronoi diagrams, *Discrete and Computational Geometry,* 2 (1987), pp. 255-270.

[SCKLPS] M. Sharir, R. Cole, K. Kedem, D. Leven, R. Pollack and S. Sifrony, Geometric applications of Davenport-Schinzel sequences, *Proc. 27th IEEE Symp. on Foundations of Computer Science,* 1986, pp. 77-86.

[Ya] C.K. Yap, An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments, *Discrete and Computational Geometry,* 2 (1987), pp. 365-393.

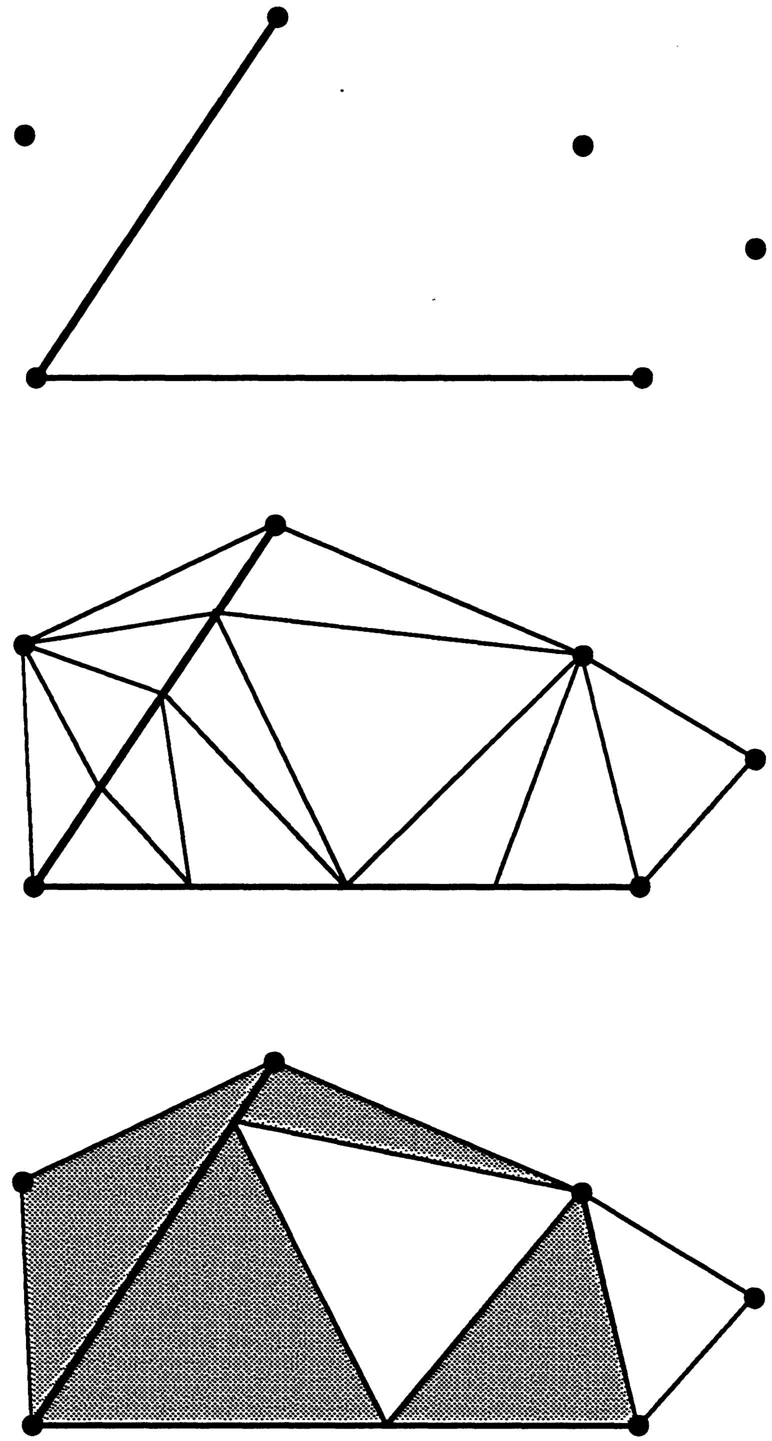

Fig. 1

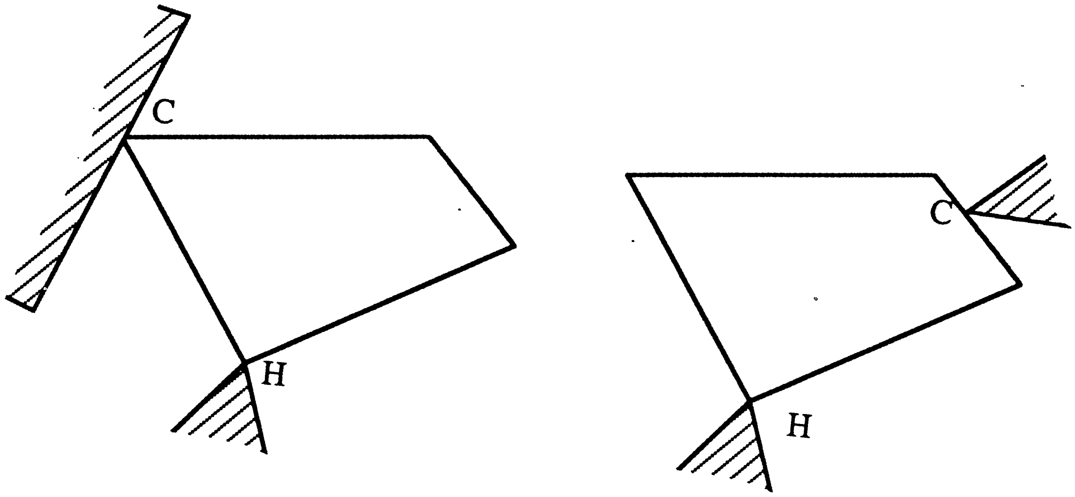
C
H
C
H

Fig. 2

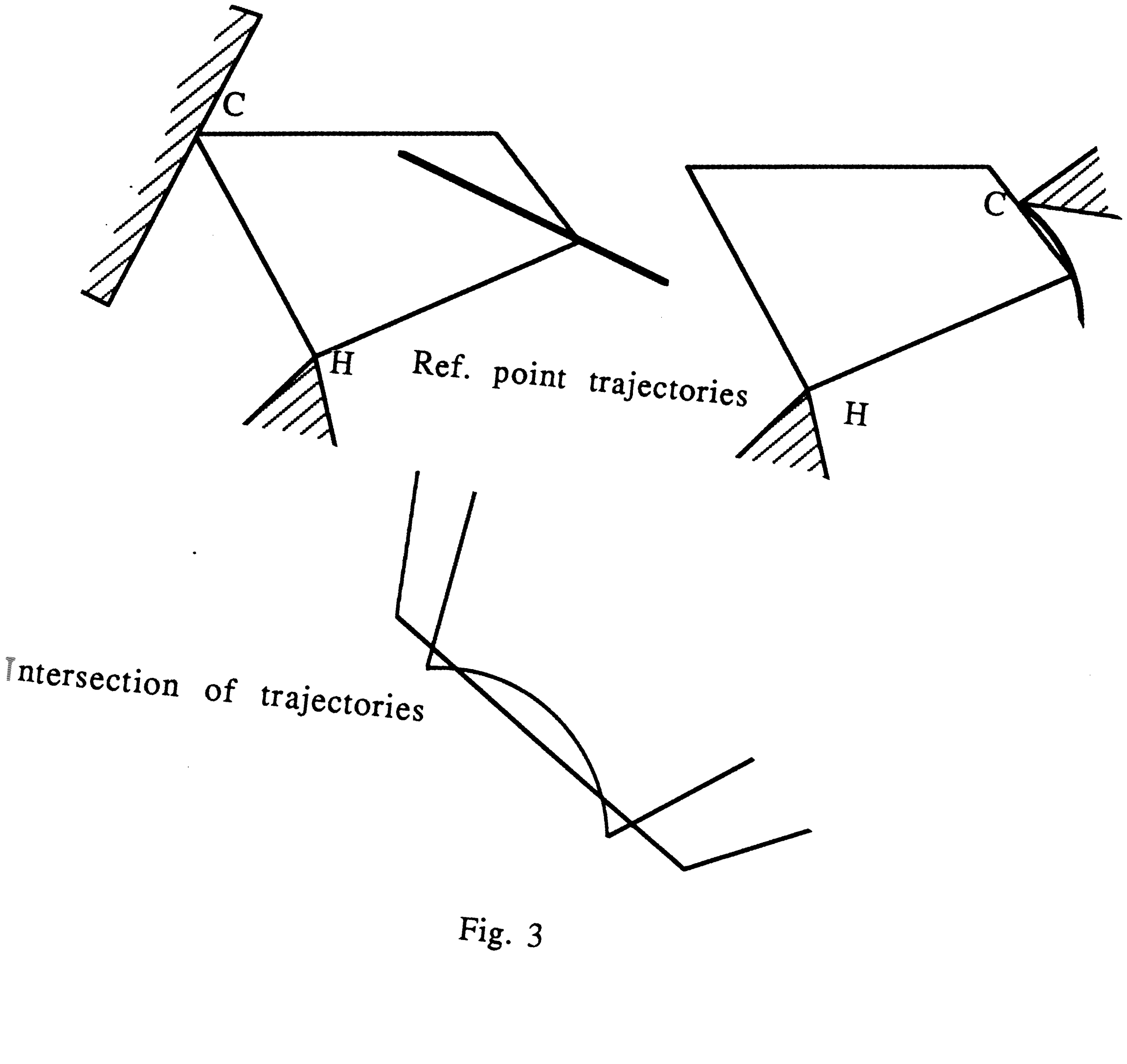
C
H
Ref. point trajectories
C
H
Intersection of trajectories

Fig. 3

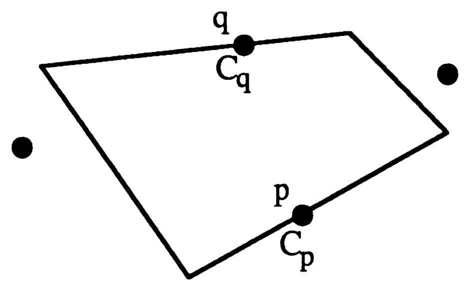


Fig. 4

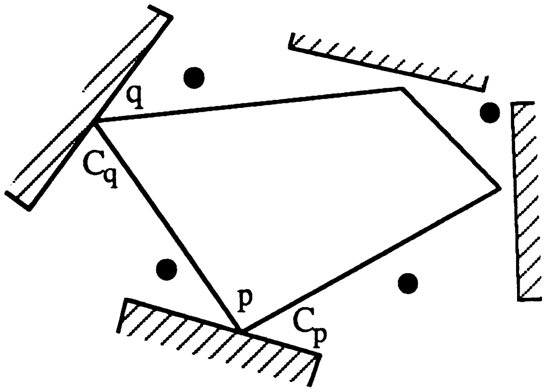


Fig. 5