Pattern Analysis and Applications manuscript No.
(will be inserted by the editor)

Recognizing Convex Polygons with Few Finger Probes

Sumanta Guha · **Kiêu Trọng Khánh**

Received: date / Accepted: date

Abstract The problem considered is that of recognizing if a given convex polygon belongs to a known collection by applying so-called finger probes (i.e., probes by laser-like rays that each return the location of contact). Existing approaches use a number of probes that is linear in the number of sides of the polygon. The current premise is that probing is expensive, while computing is not. Accordingly, a method is proposed that recognizes a polygon (arbitrarily oriented) from the given collection, with high probability, using only a constant number of finger probes, at the cost of fairly large computing resources, particularly, in setting up and applying a range tree data structure. Analysis, though partly heuristic, is validated with software and experimental results.

Keywords Finger probe · kd-tree · polygon recognition · probe-based searching · probing · range tree

1 Introduction

The problem of recognizing an object by means of probes arises in domains ranging from robotics to security. Because of its significance the problem has been studied extensively – see [12–14] for surveys of methodologies. In this paper we consider finger probes, which are probes of an object by directed rays, the outcome of each probe being the coordinates of the point of contact of the ray with the object (see Figure 1). Finger probes model devices that shoot laser or sonar beams.

Cole and Yap [4] initiated the theoretical study of finger probes. Research since then has focused primarily on determining the shape of an object given that it belongs to some restricted class (often, convex polygons or polytopes, e.g., [4,6,13]), or recognizing an object given that it belongs to a known finite collection (called model-based probing and, again, often, the collection being of convex polygons, e.g., [1,2,8,9,13]). Results proved require, typically, $O(k)$ probes, where k is a bound on the number of sides of the polygons or polytopes.

Freimer et al [8] do examine the problem of locating a polygon with a minimum number of finger probes: however, they work in a deterministic setting (assuming, moreover, that the orientation of a polygon is fixed) and, consequently, their required number of probes is still linear in the size of the polygons.

Our premise in this paper is the following: the process of physical probing itself is expensive relative to the computing power available to process the outcome of the probes. This is justified by realistic scenarios. For example, consider a fast-moving conveyor belt carrying cargo as depicted in Figure 1. Each item is in a scanning area for a short duration, during which time some number of beams are fired at it, and the outcome transmitted over a network to a central computer. The objective is to determine – based upon stored information about the shapes of a collection of known objects – a possible match. The constraints on the size of the scanning area, the number and location of the probing devices, the turnaround time of the physical beams, and network delay imply the desirability of being able to recognize an item with *very few probes*. On the other hand, it is equally reasonable to assume that the central computer that processes probe outcomes is powerful with large RAM and processor speed.

Preliminary version appeared in *Proceedings of the 11th International Conference on Computer Analysis of Images and Patterns* (CAIP 2005), Versailles, France.

S. Guha · K. T. Khánh
Computer Science & Information Management Program
Asian Institute of Technology
P.O. Box 4, Klong Luang
Pathumthani 12120
Thailand
E-mail: [guha, kieutrong.khanh]@ait.ac.th

Accordingly, we propose a method where an object from a given finite collection can be detected with *very high probability* using $O(1)$ finger probes. We restrict our investigation to collections of *convex polygons* in this paper. The polygon under investigation can be arbitrarily oriented. Our method is based on the observation that the probability that the outcome of *four* finger probes from one convex polygon in a random collection will match another is zero. We preprocess the collection, therefore, that, given four probe outcomes, we can determine possible matches: as we allow for noise and computational error, our method, typically, picks up a few possible matches, even though the theoretical probability of more than one is zero. With additional probes, that are processed in groups of four, the number of possible matches drops rapidly to one.

The principle underlying the pre-processing is to encode data about polygons in the given collection as a set of points in high-dimensional space – a set that is subsequently searched in the process of finding a match. Our primary search structure is a range tree: for a given set P of points in n-dimensional space, a range tree is a geometric data structure that allows efficient orthogonal range searching, i.e., reporting of points of P that lie in some query box B with axes-parallel sides. Because of the size of the parameters involved, our particular range tree requires a large amount of time to construct and space to store, which we suggest is acceptable given that computing resource is not a constraint in order to minimize the number of probes. Following preliminary theoretical analysis we have performed several experiments to assess our method. The results are encouraging.

The primary significance of our contribution we feel is in proposing a probabilistic approach to the very practical problem of model-based recognition using few probes, and formulating a practical algorithm.

In Section 2 we formalize the problem and describe our approach to solving it. Section 3 describes the data structures that need to be set up and the subsequent algorithm. We analyze our algorithm in Section 4. Experimental results and the software used are described in the concluding Section 5.

2 Problem and Solution Plan

We are given a collection $\mathscr{C} = \{C_0, \ldots, C_{m-1}\}$ of m mutually disjoint convex polygons on a plane. Our problem is to determine if some target polygon C, assumed to lie on the same plane as the collection $\mathscr{C}$, can be obtained by a 2D rigid transformation t (i.e., by translation and rotation on the plane) of some $C_i \in \mathscr{C}$ – if so, then we say that C_i is a *match* for C. We are only allowed finger probes on C. Now, r finger probes on C yield r points $q_0, q_1, \ldots, q_{r-1}$ on the boundary of C (say, the points lie in the given order, either clockwise or counterclockwise, on the boundary). We first formalize a trivial necessary condition for a polygon $C_i \in \mathscr{C}$ to be a match for C based on the results of the r probes of C.

Lemma 1 *A necessary condition for $C_i \in \mathscr{C}$ to be a match for the target polygon C is if the polygon P, with vertices at the r probed points $q_0, q_1, \ldots, q_r$, can be inscribed in C_i by rigid transformation.*

Proof If C_i is a match for C, then there exists a rigid transformation t such that $t(C_i) = C$, so that $t^{-1}(C) = C_i$, which implies that the vertices of $t^{-1}(P)$ lie on the boundary of C_i.

This leads to the following proposition in case of three finger probes:

Proposition 1 *Say that the target polygon C has been probed at three points q_0, q_1, q_2 on its boundary. A necessary condition for $C_i \in \mathscr{C}$ to be a match for C is that there exist three points p_0, p_1, p_2 on the boundary of C_i such that $d(q_i, q_{i+1}) = d(p_i, p_{i+1})$, $0 \leq i \leq 2$ (addition in the subscripts being modulo 3), and so that the order q_0, q_1, q_2 and p_0, p_1, p_2 on the plane is the same (either both counterclockwise or clockwise).*

Proof The proposition follows from the preceding lemma, and that a rigid transformation is an orientation-preserving Euclidean transformation.

For example, see Figure 2, where the triangle T gives a match between the target polygon C and the polygon C_0 from a collection of three polygons.

For a pair of straight-line segments s_0, s_1 define (see Figure 3):

$$\max(s_0, s_1) = \sup\{d(p,q) : p \in s_0, q \in s_1\} \text{ and}$$
$$\min(s_0, s_1) = \inf\{d(p,q) : p \in s_0, q \in s_1\}$$

Assume that the result of probing a target polygon C are three non-collinear points q_0, q_1, q_2 on the plane such that the order q_0, q_1, q_2 is counterclockwise. Let $d_0 = d(q_0, q_1)$, $d_1 = d(q_1, q_2)$, $d_2 = d(q_2, q_0)$. By Proposition 1, a necessary condition for $C_i \in \mathscr{C}$ to be a match for C is that there exist points p_0, p_1 and p_2 on the boundary of C_i such that $d(p_i, p_{i+1}) = d_i$, $0 \leq i \leq 2$, and so that the order p_0, p_1, p_2 is counterclockwise as well. If the points p_0, p_1, p_2 are assumed to lie on the edges e_0, e_1, e_2 of C_i, respectively, this in turn implies the following necessary condition for C_i to be a match for C:

There exist edges e_0, e_1, e_2 of C_i (not necessarily distinct) so that:

(1) The order e_0, e_1, e_2 around C_i is counterclockwise (if at least two of e_0, e_1 and e_2 are identical then the order can be assumed to be either of counterclockwise or clockwise).

(2) Not all three of e_0, e_1, e_2 are identical (for, otherwise, p_0, p_1, p_2 are collinear).

(3) $\min(e_i, e_{i+1}) \le d_i \le \max(e_i, e_{i+1})$, $0 \le i \le 2$.

Physical probing is never exact as noise cannot be fully eliminated from the process, and, moreover, exact arithmetic is computationally infeasible as well. Therefore, we heuristically relax condition (3) to:

(3') $\min(e_i, e_{i+1}) - \varepsilon \le d_i \le \max(e_i, e_{i+1}) + \varepsilon$, $0 \le i \le 2$

where ε is a user-specified small constant.

Call a triple of edges (e_0, e_1, e_2) from C_i satisfying these conditions (1), (2) and (3') a *candidate triple of edges* from C_i w.r.t. q_0, q_1, q_2. Note that conditions (1) and (2) are independent of the results of probes, while (3') depends on the knowing the results.

With the above in mind, our plan is straightforward. First, pre-process to construct a data structure so that, given the result q_0, q_1, q_2 of three finger probes, one can rapidly find all candidate triples of edges w.r.t. q_0, q_1, q_2 from each polygon in the collection $\mathscr{C}$. Subsequently, use this data structure to determine matches given sets of four probes.

3 Plan Implementation

3.1 Pre-processing and Data Structures

For $0 \le i \le m-1$, let

$$E_i^2 = \{(e_0, e_1) : e_0, e_1 \text{ are edges of } C_i\}$$

Let

$$E^2 = \cup\{E_i^2 : 0 \le i \le m-1\}$$

For each $(e_0, e_1) \in E^2$, define the point

$$P(e_0, e_1) = (\min(e_0, e_1),\ \max(e_0, e_1))$$

in $\mathbb{R}^2$.

Construct a 2-dimensional *range tree* (see [5]) $\mathscr{R}$ on the set of points

$$\{P(e_0, e_1) : (e_0, e_1) \in E^2\}$$

(which is allowed to contain duplicates, that arise in case $P(e_0, e_1) = P(e'_0, e'_1)$, for distinct pairs (e_0, e_1) and (e'_0, e'_1) from E^2).

At each point $P(e_0, e_1) \in \mathscr{R}$ put a pointer back to (e_0, e_1). Since the polygons in $\mathscr{C}$ are mutually disjoint, so that a straight-line segment can be an edge of at most one member of $\mathscr{C}$, this pointer identifies, as well, the polygon C_i amongst whose edges are e_0 and e_1.

Given $(e_0, e_1) \in E_i^2$, let

$$\begin{aligned} E(e_0, e_1) = \{e_2 :\ & e_2 \text{ is an edge of } C_i \text{ and } e_0, e_1, e_2 \text{ is} \\ & \text{counterclockwise around } C_i \text{ and } e_0, e_1, e_2 \\ & \text{are not all identical}\} \end{aligned}$$

In other words, $E(e_0, e_1)$ consists of the edges e_2 of C_i such that the triple (e_0, e_1, e_2) satisfy conditions (1)and (2) above.

For each $e_2 \in E(e_0, e_1)$, define the point

$$Q(e_2) = (\min(e_0, e_2), \max(e_0, e_2), \min(e_1, e_2), \max(e_1, e_2))$$

in $\mathbb{R}^4$.

For each $(e_0, e_1) \in E_i^2$, construct a 4-dimensional range tree $\mathscr{R}(e_0, e_1)$ on the set of points

$$\{Q(e_2) : e_2 \in E(e_0, e_1)\}$$

(which again is allowed to contain duplicates, that arise if $Q(e_2) = Q(e'_2)$, for distinct edges e_2 and e'_2).

At each point $P(e_0, e_1)$ of $\mathscr{R}$ place a pointer to $\mathscr{R}(e_0, e_1)$. The entire structure $\mathscr{R}'$ consisting of $\mathscr{R}$ with $\mathscr{R}(e_0, e_1)$, for all $(e_0, e_1) \in E^2$, is equivalent to a 6-dimensional range tree on the set of points

$$\begin{aligned} (&\min(e_0, e_1),\ \max(e_0, e_1),\ \min(e_0, e_2), \\ &\max(e_0, e_2),\ \min(e_1, e_2),\ \max(e_1, e_2)\ ) \end{aligned}$$

for all triples (e_0, e_1, e_2), such that $(e_0, e_1) \in E^2$ and $e_2 \in E(e_0, e_1)$.

3.2 Algorithm

Given the result q_0, q_1, q_2, q_3 of four finger probes – we assume that they are no three collinear and that the given order is counterclockwise – we proceed as follows:

Step 1: Search $\mathscr{R}'$ to report the points that lie in the 6-dimensional box

$$\begin{aligned} B = [&-\infty,\ d(q_0, q_1) + \varepsilon] \times [d(q_0, q_1) - \varepsilon,\ \infty] \times \\ [&-\infty,\ d(q_0, q_2) + \varepsilon] \times [d(q_0, q_2) - \varepsilon,\ \infty] \times \\ [&-\infty,\ d(q_1, q_2) + \varepsilon] \times [d(q_1, q_2) - \varepsilon,\ \infty] \end{aligned}$$

which determines the set S of triples (e_0, e_1, e_2) such that (e_0, e_1, e_2) is a candidate triple of edges from some C_i w.r.t. q_0, q_1, q_2, as membership in B verifies condition (3') above.

In other words, S consists of those triples (e_0, e_1, e_2) of edges from some polygon $C_i \in \mathscr{C}$ such that the image $p_j = t(q_j)$, by some rigid transformation t, *may approximately* lie on e_j, for $j = 0, 1, 2$.

Step 2: For each $(e_0, e_1, e_2) \in S$, search $\mathscr{R}(e_0, e_1)$ to report the points that lie in the 4-dimensional box

$$\begin{aligned} B(e_0, e_1) = [&-\infty,\ d(q_0, q_3) + \varepsilon] \times [d(q_0, q_3) - \varepsilon,\ \infty] \times \\ [&-\infty,\ d(q_1, q_3) + \varepsilon] \times [d(q_1, q_3) - \varepsilon,\ \infty] \end{aligned}$$

which determines the set $S(e_0, e_1)$ of edges e_3 such that (e_0, e_1, e_3) is a candidate triple of edges from C_i (the polygon amongst whose edges are e_0 and e_1) w.r.t. q_0, q_1, q_3.

Let $\overline{S}$ denote the set of quadruples (e_0, e_1, e_2, e_3) such that $(e_0, e_1, e_2) \in S$ and $e_3 \in S(e_0, e_1)$. Then $\overline{S}$ contains precisely those quadruples (e_0, e_1, e_2, e_3) of edges from some polygon $C_i \in \mathscr{C}$ such that the image $p_j = t(q_j)$, by some rigid transformation t, *may approximately* lie on e_j, for $j = 0, 1, 2, 3$.

Step 3 For each quadruple $(e_0, e_1, e_2, e_3) \in \overline{S}$ we must now verify if *indeed* the image $p_j = t(q_j)$, by some rigid transformation t, lies approximately on the edge e_j, $0 \leq j \leq 3$.

3a: We first verify as follows if the triangle with vertices at q_0, q_1, q_2 can be mapped by some rigid transformation t so that the image of q_j lies on e_j, $j = 0, 1, 2$: Imagine placing the triangle $T = q_0 q_1 q_2$ so that q_0 lies on e_0 and q_1 on e_1, then "sliding" the base $q_0 q_1$ of T so that q_0 travels along e_0 and q_1 along e_1, and determining if there is (approximately) an intersection with e_2 of the locus of the top q_2 of T by the sliding motion. In fact, to allow for approximation, we first construct a rectangle R_2 of length $l_2 + 2\varepsilon$ and width 2ε, where l_2 is the length of edge e_2 and ε is the earlier user-specified constant, and place it centered about e_2; next, we determine if the locus A of q_2, which is an arc of an ellipse (see, e.g., [7] for the *trammel construction* of an ellipse), intersects R. For example, Figure 4 shows q_0 sliding along e_0 from the position a to a', q_1 sliding along e_1 from b to b', and q_2 traveling along the arc A of an ellipse from c to c'.

If A does not intersect R_2, then the quadruple (e_0, e_1, e_2, e_3) is *rejected*; if it does intersect R_2, in say the arc A', then we proceed to the next stage.

3b: Again, to allow for approximation, construct a rectangle R_3 of length $l_3 + 2\varepsilon$ and width 2ε, where l_3 is the length of edge e_3, and place it centered about e_2. Next, exactly as in the previous step, slide the base, but this time precisely in the range so that the locus of q_2 is is A', and determine if the locus of q_3 for this range, which is again an elliptical arc, say A'', intersects R_3.

If A does not intersect R_3, then the quadruple (e_0, e_1, e_2, e_3) is *rejected*; if it does intersect R_3, then (e_0, e_1, e_2, e_3) is *accepted*, and the polygon C_i, amongst whose edges are e_j, $0 \leq j \leq 3$, is a possible match for the target polygon C.

Let $\mathscr{C}'$ be the subset of polygons of $\mathscr{C}$ that are declared as possible matches for C by the preceding procedure.

We shall prove in the next section that if $C_i \in \mathscr{C}$ is, in fact, a match for C, then, after the preceding procedure, with high likelihood, $\mathscr{C}'$ will contain *precisely* C_i.

4 Analysis

Our analysis is partly heuristic – we back it up with experiments reported in the next section.

Proposition 2 *Given a random triangle T and a random convex polygon C with a bounded number of edges, the number of different inscriptions of T in C (equivalently, the number of different rigid-body transformations that inscribe T in C) is $O(1)$ with probability 1.*

Proof Let F be the set of distinct inscriptions of triangle $T = p_0 p_1 p_2$ in C. We'll show that F is of bounded cardinality with probability 1.

Imagine sliding T around on the inside of C with its base $p_0 p_1$ lying on C (as in Figure 4). When the endpoints p_0 and p_1 of the base lie on two fixed different edges of C, then the locus of p_2 is elliptic and, therefore, it can intersect an edge of C, which is a straight segment, at most twice (actually, in the given situation, at most once, but we do not need this stronger fact for the proof). When the endpoints, and, therefore, the entire base $p_0 p_1$ lies on one fixed edge of C, the locus of p_2 is a straight segment parallel to that edge. Such a segment can intersect an edge of C at most once, *except* when another edge of C is parallel to the edge on which the base lies and at a distance from it equal to that of p_2 from $p_0 p_1$, as depicted in Figure 5(a), when they overlap and intersect in any infinite set. However, the probability is 0 that a random convex polygon with a bounded number of edges has two parallel ones.

During a complete rotation of T around C, p_2 describes a union of a bounded number of distinct elliptic and straight loci. We conclude that the cardinality of F is indeed bounded with probability 1.

Corollary 1 *Given a random convex quadrilateral Q and a random convex polygon C with a bounded number of edges, the probability that Q can be inscribed in C (equivalently, that there exists a rigid-body transformations that inscribes Q in C) is* 0.

Proof Let the quadrilateral $Q = p_0 p_1 p_2 p_3$, and consider the triangle $T = p_0 p_1 p_2$. We know from the proposition that the set F of distinct inscriptions of triangle $p_0 p_1 p_2$ in C is of bounded cardinality with probability 1.

Now, Q can be inscribed in C if and only if p_3 lies on C for some inscription of $T = p_0 p_1 p_2$ in C (see Figure 5(b)). However, the set of positions of p_3 corresponding to an inscription of T in C is of bounded cardinality with probability 1, as such is true from F; moreover, the probability that subset of the plane of bounded cardinality, and, therefore, measure zero, intersects a random convex polygon C is 0. The result follows.

This leads to the next important corollary which probabilistically justifies our strategy of using four probes to distinguish polygons from a collection of convex ones.

Corollary 2 *Given a random collection $\mathscr{C}$ of convex polygons, the probability that a quadrilateral $p_0p_1p_2p_3$, inscribed in a particular polygon $C_i \in \mathscr{C}$, can be inscribed by rigid transformation in another polygon $C_j \in \mathscr{C}$ ($j \neq i$) is 0.*

Proof Follows easily from the previous corollary.

Proposition 3 *A range tree on a set of N points in $\mathbb{R}^d$ (i.e., a d-dimensional range tree) can be constructed in $O(N\log^{d-1}N)$ and it uses $O(N\log^{d-1}N)$ space. Such a range tree can be queried to report the points in a d-dimensional box in $O(log^d N + R)$ time, where R is the number of reported points.*

Proof We refer the reader to existing literature, e.g., [5].

Proposition 4 *The time to construct and space to store the data structures required for the algorithm of Section 3.2 are both $O(mk^3\log^5(mk^3))$, where m is the cardinality of the collection $\mathscr{C}$ of convex polygons, and k is a bound on the number of edges of polygons belonging to $\mathscr{C}$.*

The time to query the structure and report a match, if there is one, is $O(m\log^4 k)$ with high probability.

Proof The time to construct, as well as space to store $\mathscr{R}'$, is $O(mK^3\log^5(mk^3))$, by Proposition 3, as $\mathscr{R}'$ is a 6-dimensional range tree containing $O(mk^3)$ points. This proves the claim for the construction time and space for our data structures.

As the cardinality of $\mathscr{C}$ is m, it follows from Proposition 2 that the number of points reported in Step 1 of the procedure is $O(m)$ with high probability. How close the probability actually is to 1 depends on two items:

(i) The smallness of the user-specified constant ε.
(ii) The smallness of the length of each edge of the polygons in $\mathscr{C}$ relative to the length of the boundary of the polygon to which the edge belongs.

The smaller the edges are relative to the entire boundary, the fewer the number of *false positives*. A false positive is a triple (e_0, e_1, e_2) that satisfies the necessary conditions (1)-(3) of Section 2, *without* there being a rigid transformation t so that the images $p_j = t(q_j)$ lie (approximately) on e_j, for $j = 0,1,2$. We make the heuristic assumption that the number of false positives per polygon in $\mathscr{C}$ is $O(1)$.

Accordingly, the cardinality of the set S of triples created in Step 1 is $O(m)$ with high probability. Therefore, by Proposition 3, the time spent in Step 1 is $O(log^6(mk^3) + m)$ with high probability.

Step 2 involves a search of a 4-dimensional range tree containing at most k points for each point in the set S created in Step 1. By Proposition 2, with high probability – again how high depends on the same items (i)-(ii) described for Step 1 – at most one of these searches will report a single point. Therefore, with high probability, the time spent in Step 2 is $O(m(\log^4 n + O(1))) = O(m\log^4 n)$.

Each verification in Step 3 takes $O(1)$ time, as it involves $O(1)$ tests to detect the intersection between the arc of an ellipse and a rectangle.

Totaling the time for Steps 1-3, the claim for the query time follows.

5 Experimental Results and Conclusions

We wrote the data structures, as described in Section 3.1, in Java, using range tree code from CGAL [3], and, as well, using kd-trees (another data structure for orthogonal range searching [5], code from [10]), in place of range trees. In our experiments the kd-tree, even though theoretically less efficient, was actually quicker than the range tree, evidently due to the constants involved.

Our code, both source and executable, is available online [11]. A set of 50 polygons is displayed on-screen for the user to select one and probe with the mouse. With each successive probe, polygons that match the current set of probe points based upon our algorithm are highlighted, until only the selected polygon remains.

Our (heuristic) method to generate random convex polygons is to generate random sets of points and find their convex hulls. We tested our algorithm on batches of 50 randomly-generated convex polygons each. From each batch we randomly chose 30 polygons, and randomly probed each. In particular, for each polygon, we first probed it four times and, if this did not yield a unique match, increased the number of probes, each time taking the intersection of the matches for each subset of four, until we were left with a single polygon.

Figure 6 summarize our experimental results. Four probes almost invariably resulted in a large number of matches – often more than 20 out of 50 polygons – which we attribute to our method allowing for noise. Increasing the number of probes to eight always resulted in a unique match in our experiments. Moreover, execution times for our experiments were negligible.

We conclude that initial results are encouraging for our proposed probabilistic approach to the extremely practical problem of recognizing objects from a collection of suspects by applying a bounded number of probes and, therefore, in constant time. Extensions along the following lines should be investigated:

1. Compression of the data structures involved in our approach would improve its efficiency. Possibly, keeping the same general principle of codifying polygon data as points in a high-dimensional space and then search-

ing geometrically, one might investigate variations of the strategy proposed in this paper.

2. Most importantly, methods must be extended to 3-dimensional objects for the majority of practical applications.
3. The software bears improving, and further experimentation is required to support the heuristic analysis.
4. Most welcome, of course, would be a sharper theoretical analysis.

Acknowledgements We thank Chansophea Chuon and Nguyen Tan Khoa for help with the software. We thanks the referees of this and the earlier conference version for several suggestions that helped improve the presentation significantly.

References

1. P. Belleville, T. C. Shermer, Probing Polygons Minimally is Hard, *Computational Geometry: Theory and Applications* **2** (1993), 255-265.
2. H. J. Bernstein, Determining the Shape of a Convex n-sided Polygon using $2n+k$ Tactile Probes, *Information Processing Letters* **22** (1986), 255-260.
3. CGAL, Computational Geometry Algorithms Library, `http://www.cgal.org`.
4. R. Cole, C. K. Yap, Shape from Probing, *Journal of Algorithms* **8** (1987), 19-38.
5. M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Second Edition, Springer-Verlag, 2000.
6. D. P. Dobkin, H. Edelsbrunner, C. K. Yap, Probing Convex Polytopes, *Proc. 18th ACM Symposium on the Theory of Computing*, 1988, 424-432.
7. H. W. Eves, *A Survey of Geometry*, Revised Edition, Allyn and Bacon, 1965.
8. R. Freimer, S. Khuller, J. S. B. Mitchell, C. Piatko, K. Romanik, D. Souvaine, Localizing an object with finger probes, *Proc. SPIE*, Vol. 2356, Vision Geometry III (R. A. Melter, A. Y. Wu, Eds.) 1995, 272-283.
9. E. Joseph, S. S. Skiena, Model-Based Probing Strategies for Convex Polygons, *Computational Geometry: Theory and Applications* **2** (1992), 209-221.
10. S. D. Levy, KDTree - A Java class for KD-tree search (exact and nearest-neighbor), `http://www.cs.wlu.edu/~levy`.
11. PolyRecognition: polygon recognition software, `http://www.cs.ait.ac.th/~guha/papers/PolyRecognition.zip`.
12. K. Romanik, Geometric Probing and Testing - A Survey, DIMACS Technical Report 95-42, 1995.
13. S. S. Skiena, *Geometric Probing*, Ph.D. Thesis, Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1998.
14. S. S. Skiena, Interactive Reconstruction via Geometric Probing, *Proceedings of the IEEE* **80** (1992), 1364-1383

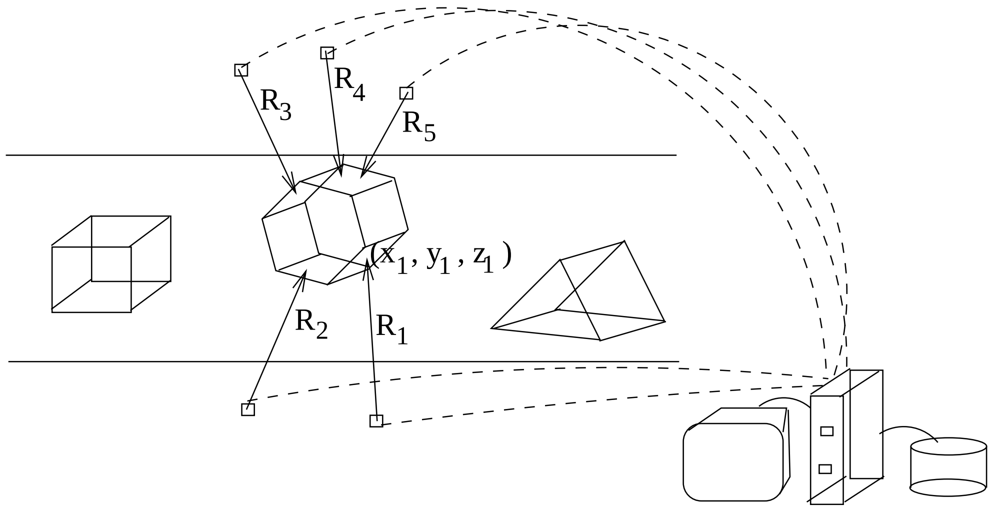

Fig. 1 Five laser probes are aimed at an item on a conveyor belt as it passes through the scanning area. Collected data is returned to a computer to determine a match with a database of known objects.

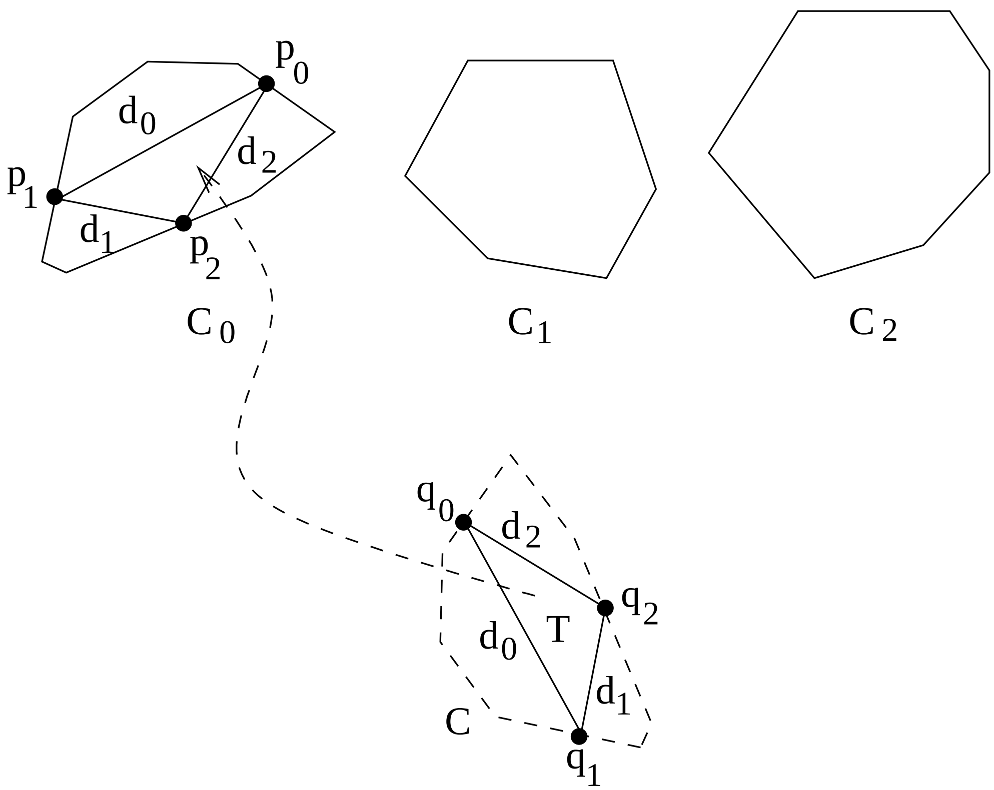

Fig. 2 Target polygon C is probed to find that triangle T which can be inscribed in C_0.

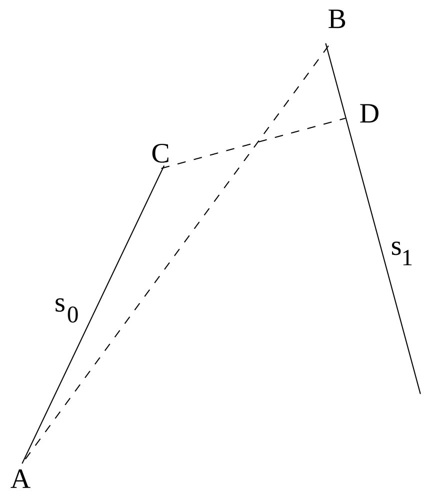

Fig. 3 Two straight-segments s_0 and s_1 with $|AB| = \max(s_0, s_1)$ and $|CD| = \min(s_0, s_1)$.

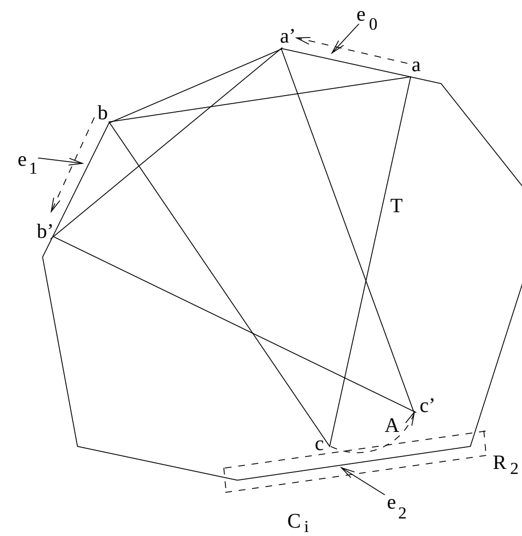

Fig. 4 Sliding triangle T with its base on two edges of the polygon C_i.

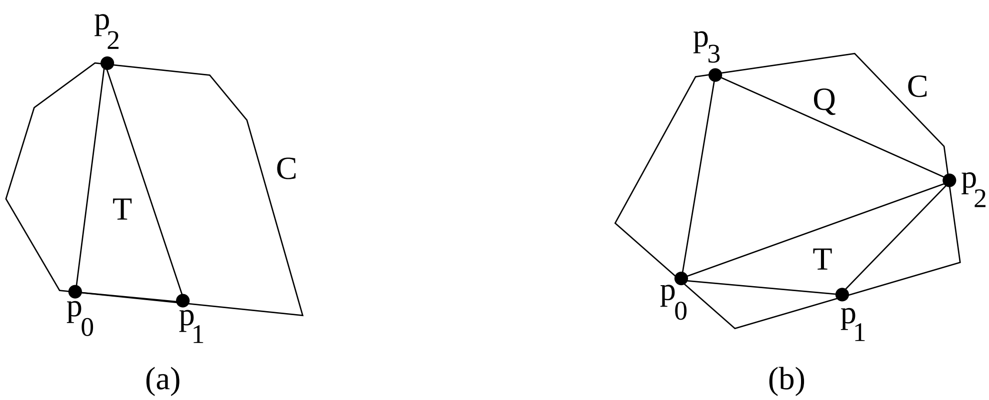

Fig. 5 (a) Convex polygon C with a triangle T inscribed on two parallel edges (b) Q is inscribed in C if p_3 lies on C for some inscription of T in C.

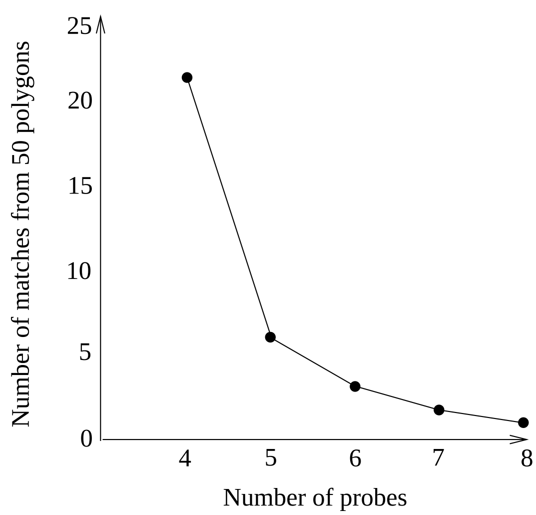

Fig. 6 Number of matches from 50 randomly-generated convex polygons vs. number of probes.