On Detecting the Orientation of Polygons and Polyhedra

B. K. Natarajan

CMU-RI-TR-86-18

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

December 1986

Table of Contents

List of Figures

Abstract

This paper concerns the use of ray sensors to detect the orientation of polygonal and polyhedral objects moving on a belt or slide. The problem is abstracted to the computational domain and the following results obtained. For polygons of n vertices, (n) sensors are sufficient and ($n/2$) necessary. For convex polyhedra of n vertices, ($6n$) sensors are sufficient and ($n/4$) necessary. Non-convex polyhedra cannot be effectively handled with such sensors.

1. Introduction

This paper deals with detecting the orientation of polygonal and polyhedral objects using simple ray sensors. In robot-automated assembly, precise knowledge of the orientation of the components involved is essential if the robot is to succesfully pick up and assemble them. Sensorless manipulation to achieve unique orientation is discussed in [Lozano-Perez, 1986], [Natarajan, 1986] and [Erdmann and Mason, 1986]. In this paper, we discuss a method that uses sensing without manipulation to determine orientation. Needless to say, the mode of sensing is a key ingredient of the problem. Positioning a camera in front of the object of interest shifts the problem into the domain of machine-vision and pattern recognition. Tactile sensing as applied to a related problem is explored in [Bernstein, 1984]. We consider single-ray optic sensors that can do no more than detect the presence of the object along their sensing axes.

We propose an application-oriented scheme for the use of the ray sensors and obtain the following results within the framework of the scheme. For n-sided polygons in the plane, $(n/2)$ sensors are necessary and n sensors sufficient to determine orientation. For n-vertex convex polyhedrons $(n/4)$ sensors are necessary and $(6n)$ sufficient. The orientation of non-convex polyhedrons cannot be effectively determined using these sensors.

2. Detecting the Orientation of Polygons

Consider the guide plane consisting of a plane guideway with a lip as shown in Fig. 2.1. The polygonal object of interest slides along the guide plane with one of its edges resting on the lip. The aim is to detect the orientation of the object as it slides by. This paradigm is intended to capture the practical application of objects travelling on conveyor belts and slides. To aid in determining the orientation, we are provided with sensors that work as follows. A sensor possesses a distinguished direction called its sensing axis. The *location* of a sensor is its position in space and the direction of its sensing axis - an ordered pair (s_1, s_2), where s_1 is the position and s_2 is the direction of the sensing axis of the sensor. (Needless to say, it is desirable that the sensor position is such that the sensor does not collide with the object as it slides by.) When positioned at any location, the sensor can detect the intersection of its axis with the object. Given a polygonal object, how do we locate these sensors along the guide plane and how many do we need to uniquely determine the

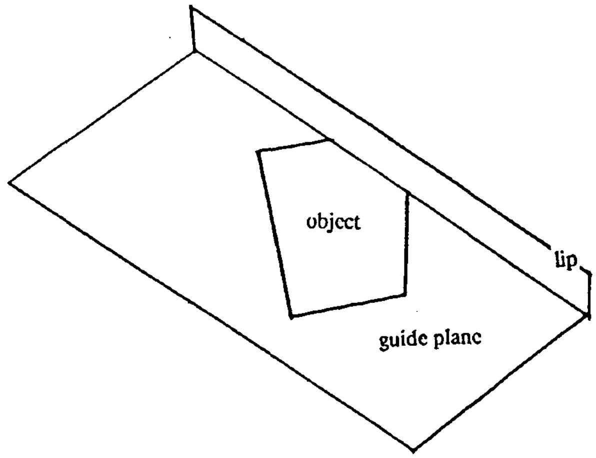

Figure 2-1: The guide plane

orientation of the object?

One possibility is to locate the sensors above the guide plane with their axes normal to the guide plane. Consider a string of such sensors, evenly spaced along a line normal to the lip of the guide plane. Then, if we observe the sensors at evenly spaced time intervals, we can obtain a "bitmap" of the object. The difficulty with this is that the spacing (and hence the number of sensors) and the number of time intervals depends on how "different" any two orientations of the object are. For example, a rectangle that is almost square would require a large number of sensors rather than just one. It is desirable to look for solutions that require the same number of sensors for all polygons with the same number of sides and that read these sensors once (or a constant number of times). With this in view, we propose the following scheme. Pick a point on the lip of the guide plane and read the sensors at the instant the base edge of the object first contacts this point. The issue now is to determine the number of sensors and their location, given the polygon whose orientation is to be detected.

We abstract the above problem as follows, limiting ourselves to convex polygons for now. Given is a convex polygon of n vertices in the plane. Label the vertices $v_1, v_2, \ldots, v_n$ in cyclic order. In an X-Y-Z frame, an orientation of the polygon is *allowable* if the polygon lies in the X-Y plane and if some edge $v_i v_{i+1}$ is parallel to the positive X-axis and v_i coincides with the origin. Denote such an orientation by $[v_i]$. Clearly there are n allowable orientations. We say two allowable orientations $[v_i]$, $[v_j]$ are equal, $[v_i] = [v_j]$, if they are the same within a cyclic permutation of the labels. Also, we will use the notation $[v_i]$ to refer to the orientation or to the set of points in the plane contained by the polygon in that orientation, depending on the context for clarification. We define the operator $\wedge$ as follows. Let $[v_i]$ be an allowable orientation and let S be a set of sensor locations.

$$[v_i] \wedge S = \{(s_1, s_2) \mid (s_1, s_2) \in S \text{ and the half line thro } s_1 \text{ in the direction of } s_2 \text{ intersects } [v_i]\}.$$

We are now ready for the problem statement.

Problem 1: Given an n-sided convex polygon P, find a set of vector pairs S such that for any two allowable orientations $[v_i]$ and $[v_j]$ of the polygon,

$$[v_i] = [v_j] \text{ iff } S \wedge [v_i] = S \wedge [v_j].$$

and for any $(s_1, s_2) \in S$ and orientation $[v_i]$ of P, $s_1 \notin [v_i]$. *We say S detects P.* □

Algorithm 1

begin

let z be the unit vector along the Z-axis.

$S = \emptyset$;

for each allowable orientation $[v_i]$ **do**

$S = S \cup \{(x,z) \mid x \text{ is a vertex of } [v_i]\}$

od

end

Figure 2-2: An algorithm for Problem 1.

Fig. 2.2 gives an algorithm for Problem 1. Given a convex polygon of n vertices, run Algorithm 1 on it.

Then, for any two allowable orientations $[v_i]$ and $[v_j]$ of the polygon,

$$[v_i] \wedge S = \{\text{vertices of } [v_i]\}$$

$$[v_j] \wedge S = \{\text{vertices of } [v_j]\}$$

and hence

$$[v_i] = [v_j] \text{ iff } [v_i] \wedge S = [v_j] \wedge S$$

Hence, Algorithm 1 solves Problem 1. Unfortunately $|S| = n^2$, which is rather large considering that this is the number of sensors in the practical application and not the number of computational steps. Can we do better? Is the absolute lower bound of $\log n$ attainable? Fig. 2.3 gives a better algorithm for the problem.

Algorithm 2

let z be the unit vector along the Z-axis.

function split(C: set of orientations): set of sensor locations

  if $|C| = 1$ then **return**

  Examine the vertices of the orientations in C to find a

  vertex point v that partitions C into C_1, C_2. i.e.,

  $C = C_1 \cup C_2$, $C_1 \neq \emptyset$, $C_2 \neq \emptyset$

  and v belongs to every orientation in C_1 and

  none in C_2.

  return $(\{(v,z)\} \cup \mathit{split}(C_1) \cup \mathit{split}(C_2))$

end

begin

  $S = \mathit{split}(\{[v_1], [v_2], \ldots, [v_n]\}$

end

Figure 2-3: A better algorithm for Problem 1.

To show that Algorithm 2 solves Problem 1, we only need show that if $|C| > 1$, the function *split* will indeed split C correctly. If $|C| > 1$, then C contains two distinct orientations. If so, they must differ at one of their vertices. Since the algorithm examines all the vertices of the orientations in C, such a vertex will be found and the split will be successful.

On termination, S contains at most n elements. The running time of the algorithm is not of interest here, suffice it to say that it is polynomial in n.

Next, we show a lower bound on the size of the sensor set S.

Claim 1: For infinitely many n, there exists a convex n-sided polygon P, such that if a set of vector pairs S detects P, then $|S| \geq n/2$.

Proof: For any integer m, construct a $n=2m$ sided almost-regular polygon as follows. Take a m-sided regular polygon of unit side and label the vertices in clock-wise order $v_1, v_2, \ldots, v_m$. Pick small positive $\delta, 0 < \delta \ll 1$, and visit the vertices in the same order and mark off points $u_1, u_3, u_5, \ldots, u_{2m-1}$ so that u_{2i-1} lies on edge $v_{i-1}v_i$ and is a distance δ away from v_i. Again visit the vertices in order and mark off points $u_2, u_4, \ldots, u_{2m}$ so that u_{2i} lies on edge v_iv_{i+1} and is a distance $\delta+((i-1)/n)\delta$ away form v_i. Now construct the $n=2m$ sided convex polygon P with vertices $u_1, u_2, \ldots, u_{2m}$ taken in order. See Fig. 2.4.

Let s be a vector pair such that for some two allowable orientations $[u_i]$ and $[u_j]$ of P,

$$[u_i] \wedge \{s\} \neq [u_j] \wedge \{s\}.$$

Then there exists a cyclic permutation of $u_1, u_2, \ldots, u_n$ to $t_1, t_2, \ldots, t_n$ such that for any u_i and u_j

$[u_i] \wedge \{s\} \neq [u_j] \wedge \{s\}$ iff

$u_i \in \{t_1, t_2, \ldots, t_k\}$

and

$u_j \in \{t_{k+1}, \ldots, t_n\}$

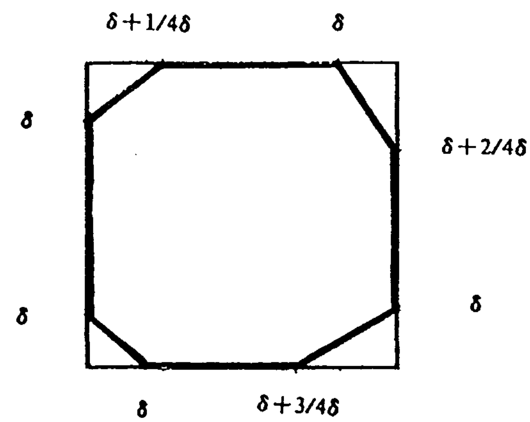

Figure 2-4: The constructed polygon.

From which it follows that at least $n/2$ vector pairs are required to separate all the pairs of orientations. □

Remark: In its present form, Algorithm 2 will not work on non-convex polygons, as it is possible for two distinct non-convex polygons to have the same vertex points. However a simple (although inelegant) modification will extend its scope to include non-convex polygons.

3. Polyhedra

We now propose an analogous scheme for determining the orientation of polyhedra. The polyhedron of interest slides along the glide plane with some face resting on the plane and some edge of this face along the lip. As in the previous section, the sensors are positioned about the guide plane and are read at the instant the edge along the lip first intersects a distinguished point on the lip. Again, we wish to determine the location and number of sensors needed to determine the orientation of a given polygon as it slides by. We first abstract the problem as follows.

In an X-Y-Z frame, an orientation of a polyhedron is *allowable* if some face of the polyhedron lies on the X-Y plane and the polygon defined by that face is in an allowable orientation as defined in the previous section. We denote the allowable orientations by $[o_1]$, $[o_2]$ etc. and use the same notation for the sets of points contained by the polyhedron in the corresponding orientations. Two orientations are equal (=) if they are the same within a cyclic permutation of the labels. We define the operator $\wedge$ exactly as in the previous section.

We are now ready for the statement of the abstracted problem.

Problem 2: Given a polyhedron P of n vertices, find a set S of of vector pairs such that for any two allowable orientations $[o_1]$ and $[o_2]$ of P,

$$[o_1] = [o_2] \text{ iff } [o_1] \wedge S = [o_2] \wedge S.$$

and for any $(s_1, s_2) \in S$ and orientation $[o_i]$ of P, $s_1 \notin [o_i]$. □

Before we give an algorithm for Problem 2, the following is necessary.

A line is *tangent* to a polygon iff it lies in the plane of the polygon and they intersect at exactly one point. A line is *tangent* to a polyhedron iff they intersect at exactly one point.

Claim 2: Let P be a convex polyhedron and f be a face of P. Then, any tangent to f is a tangent to (P).

Proof: A tangent T to f must pass through a vertex of f. Since T lies in the plane of f, any face of P that it intersects must intersect the plane of f. But since P is convex, all such intersections form the edges of f, and T intersects these at exactly one point. Hence T is a tangent of P. □

Fig. 2.5 presents an algorithm for the problem that is analogous to Algorithm 2. Here, to distinguish between two orientations $[o_1]$ and $[o_2]$, we first find a vertex point v belonging to $[o_1]$ say and not to $[o_2]$. Then we find a line through v that does not intersect $[o_2]$ by picking a line tangent at v to the convex hull of v and $[o_2]$. This gives the axis direction for a sensor that separates $[o_1]$ and $[o_2]$. We then pick a distant point on this line for the sensor position so as not to place the sensor within the polyhedron in any of its allowable orientations.

Once again, we make no effort to minimize the running time of the algorithm and limit our interest to the

Algorithm 3

function 3split(C:set of orientations): set of sensor locations

begin

if $|C| = 1$ **then return**

Examine the vertices of the orientations in C to find a

vertex point v that partitions C into C_1 and C_2. i.e.,

$$C = C_1 \cup C_2, C_1 \neq \emptyset, C_2 \neq \emptyset$$

and v belongs to every orientation in C_1 and none in C_2.

Construct the convex hull H of the set of points

$$\{u \mid u \text{ is a vertex of some orientation in } C_2\} \cup \{v\}$$

and pick any line L tangent to H at v.

Let x be a "distant" point on L.

return(3split(C_1) $\cup$ 3split(C_2) $\cup$ $\{(x,L)\}$)

end

begin

S = 3split({[o] | [o] is an allowable orientation of the given polyhedron})

end

Figure 3-1: An algorithm for Problem 2.

size of the solution set. Clearly this is bounded by the number of distinct allowable orientations of the polyhedron which in turn is no more than twice the number of edges of the polyhedron. Hence, $|S| \leq 6n$, where n is the number of vertices of the given polyhedron. We will now show this to be asymptotically optimal.

Claim 3: $n/4$ is a lower bound on the size of S in Problem 2.

Proof: Construct a right cylinder of *2n* vertices with the *n*-sided polygon of Claim 1 as cross-section. By the proof of Claim 1, at least *n*/2 sensors will be required to to detect its orientation. Since such a cylinder can be constructed for infinitely many *n*, the claim follows. □

Remark: The orientation of non-convex polyhedra cannot be determined by this scheme. To see this, we only need observe that two distinct orientations of a non-convex polyhedron can have identical shadows in every direction.

4. Finite Collections

It is easy to see that the foregoing algorithms and lower bounds hold for finite collections of polyhedra and polygons. In particular, it is possible to identify elements of a set of *n* convex polyhedra (polygons) with *n* sensors, and (*n*/4) sensors is a lower bound.

5. Conclusion

We proposed an application scheme for detecting the orientation of polygons and polyhedra using simple sensors. We then abstracted the scheme to the computational domain and developed algorithms that determine the location of the sensors, given the object whose orientation is to be detected. The algorithms are provably optimal with respect to the number of sensors used.

6. References

Bernstein, H.J., Determining the Shape of a Convex n-Sided Polygon using 2n+k Tactile Probes, TR. No 125, Courant Instt. of Math. Sciences, NYU, 1984.

Erdmann, M. and M.T. Mason, An Exploration of Sensorless Manipulation, 3rd IEEE Conference on Robotics and Automation, April 1986.

Lozano-Perez, T., 1986, Private Communication.

Natarajan, B.K., An Algorithmic Approach to the Automated Design of Parts Orienters, IEEE Foundations of Computer Science, 1986.