Robust Point Location in Approximate Polygons (Extended Abstract)

A. James Stewart
Computer Science Department
Cornell University
jstewart@cs.cornell.edu

Introduction

Most geometric algorithms assume that perfect "real" arithmetic is available. When these algorithms are implemented they often fail because this assumption is not borne out; that is, these algorithms are not *robust.* This failure occurs because either the input or the intermediate calculations are imprecise, leading to inconsistent decisions by the algorithm.

This paper presents a framework for reasoning about robust geometric algorithms which operate on polygons. *Robustness* is formally defined and a data structure called an *approximate polygon* is introduced and used to reason about polygons constructed of edges whose positions are uncertain.

A robust algorithm for point location in an approximate polygon is described. The interesting aspect of this algorithm is that in addition to the polygon's position being uncertain, the point's position in the plane does not have to be known; only the point's *signature* is important (that is, its left/right relations to the edges of the polygon). The point location algorithm has immediate practical application to solid modeling, particularly in the robust intersection of polyhedra.

An approximate polygon could, by shifting its edges back and forth within their error bounds, induce a large number of different line arrangements. In each of these arrangements some points with a given signature α may or may not appear, and if they appear, they may be to the interior or to the exterior of the polygon which induces the arrangement. An interesting *uniqueness theorem* is presented which states that in all such line arrangements, the points with signature α in each arrangement are always to the same side of the polygon which induces that arrangement.

Background

The theory of approximate polygons is based upon the "representation and model" approach of Hoffmann, Hopcroft, and Karasick [3]. In this approach the algorithm operates on a computer representation, but presents output as though it were operating on some mathematical model corresponding to the representation.

An approximate polygon is a computer *representation* of some real, mathematical polygon, the *model.* The model is rarely explicitly constructed by the algorithm. An approximate polygon P_{rep} can be thought of as a set of constraints on the topology and position of the implicit model polygon. Any real polygon P satisfying these constraints is considered a model for P_{rep}.

Under the representation and model approach, the definition of robustness is very close to that of Fortune [2]. Consider a geometric problem $\mathcal{P}$ as a function from an input space consisting of *models* to an output space, $\mathcal{P} : \mathcal{I} \rightarrow \mathcal{O}$, and consider an algorithm $\mathcal{A}$ as function from a different input space consisting of *representations* to the same output space, $\mathcal{A} : \mathcal{R} \rightarrow \mathcal{O}$. Given a representation x_{rep}, the set of its models is denoted $\textsc{models}(x_{rep})$. This leads to a definition of robustness:

> An algorithm $\mathcal{A}$ for a problem $\mathcal{P}$ is *robust* if
>
> $$\forall\ x_{rep} \in \mathcal{R},\ \exists\ x \in \textsc{models}(x_{rep})$$
>
> such that $\mathcal{A}(x_{rep}) = \mathcal{P}(x)$.

Note that we can pick an arbitrary $x \in \textsc{models}(x_{rep})$. It could be that there are two models x^1 and x^2 such that $\mathcal{P}(x^1) \neq \mathcal{P}(x^2)$. In this case the algorithm could choose to output either $\mathcal{P}(x^1)$ or $\mathcal{P}(x^2)$ and would still be considered to be robust. This leads to a definition of consistency:

A problem $\mathcal{P}$ and a representation $\mathcal{R}$ are *consistent* if

$$\forall\, x_{rep} \in \mathcal{R},\ \ \forall\, x^1, x^2 \in \textsc{models}(x_{rep}),$$

$$\mathcal{P}(x^1) = \mathcal{P}(x^2).$$

Definitions

An approximate polygon closely mirrors the appearance of a real polygon, as shown in Figure 1. The approximate polygon consists of an ordered list of *bands* corresponding to the edges of the model. The position of the bands in the plane constrains the line equations of the model. A formal definition of approximate polygons is given in the full paper.

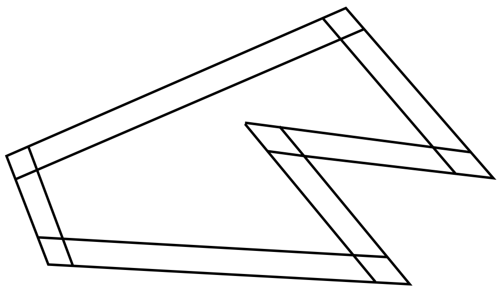

Figure 1: An Approximate Polygon

Each edge of an approximate polygon lies in a region B_i called a *band*, as shown in Figure 2. The edge must lie on a line which passes through the ends of the band; that is the line must lie completely within the shaded region of Figure 2.

It will be useful later on to talk about the *span* of a band. This is the set of points swept out by all lines which fit within the band. The *left* and *right* of a band are the set of those points to the left and right of the span. By convention, the interior of the approximate polygon is to the right of the band. In Figure 2 the shaded region is $\textsc{span}(B_i)$ and to its left and right are $\textsc{left}(B_i)$ and $\textsc{right}(B_i)$. For a band B_i, define the set of lines in the shaded region of Figure 2 as $\textsc{lines}(B_i)$.

$$\begin{aligned}
\textsc{span}(B_i) &= \{x \mid \exists\, \ell \in \textsc{lines}(B_i),\ \ell(x) = 0\} \\
\textsc{right}(B_i) &= \{x \mid \forall\, \ell \in \textsc{lines}(B_i),\ \ell(x) < 0\} \\
\textsc{left}(B_i) &= \{x \mid \forall\, \ell \in \textsc{lines}(B_i),\ \ell(x) > 0\}
\end{aligned}$$

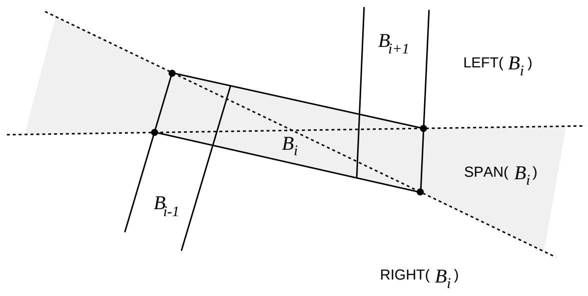

Figure 2: The $\textsc{span}$ of a Band

Robust Point Location in Approximate Polygons

The point location problem would be simple if the exact location of the point were given. However, in most practical applications the point's location is known only to be within some region of uncertainty. In particularly ill-conditioned situations this region of uncertainty can be as large as the polygon itself.

Some practical applications (geometric modelers, for example) can, from other information, logically deduce the $\textsc{left}/\textsc{right}$ status of the point with respect to each edge of the polygon. Call this $\textsc{l}/\textsc{r}$ sequence the *signature*. If the polygon's location is known exactly, then in the induced line arrangement a cell decomposition can easily determine whether all points with a given signature lie inside or outside the polygon. It is a different matter, however, when there is uncertainty in the polygon's location. If uncertainty is modeled with an approximate polygon then the following questions must be answered:

> **Question 1 (Robustness)** Given an approximate polygon P_{rep} and a signature $\alpha \in (\textsc{l}|\textsc{r})^*$, does P_{rep} have a model P in which the induced line arrangement contains a cell with signature α, and is the cell $\textsc{inside}$ or $\textsc{outside}$ the model P?

> **Question 2 (Consistency)** Consider that an approximate polygon can have two models, P^1 and P^2, which induce two different line arrangments. These two arrangements each contain a cell with signature α (call them C^1 and C^2). Then is it possible that C^1 is $\textsc{inside}$ P^1 and C^2 is $\textsc{outside}$ P^2?

If the answer to Question 2 were affirmative then the signature α and the approximate polygon P_{rep}

would not be sufficient information to determine point location, and the problem would not be *consistent.* The Uniqueness Theorem which is presented later proves that this is *not* the case.

Some final definitions

A *signature* $\alpha(v)$ is a string in $(\text{L}|\text{R})^*$. The signature denotes the relation of the point v to each edge e_i of the polygon P. The i_{th} element of $\alpha(v)$ is the relation of the point v to edge e_i of the polygon P.

Refer to Figure 2 for the following definitions. A *half-region* is similar to a half-space, except that it has a polygonal boundary. The following *half-regions* R_i and L_i consist of those points which, in *at least one* model P, are either ON e_i or to the RIGHT or LEFT of e_i, respectively, in that model. Given some $\alpha_i(v)$, the half-region H_i is that region in whose interior v must lie if it is to have $\alpha_i(v)$ as the i^{th} component of its signature. The *interior* of the cell $\tilde{C}_\alpha$ consists of those points which have signature α in *at least one* model.

$$R_i = \text{SPAN}(B_i) \cup \text{RIGHT}(B_i)$$

$$L_i = \text{SPAN}(B_i) \cup \text{LEFT}(B_i)$$

$$H_i = \begin{cases} R_i & \text{if } \alpha_i = \text{R} \\ L_i & \text{if } \alpha_i = \text{L} \end{cases}$$

$$\tilde{C}_\alpha = \bigcap_{i=1}^{n} H_i$$

The next two lemmas will be used to construct the point location algorithm. The first lemma shows that for each point in $\tilde{C}_\alpha$ there exists some model in which the point has signature α; the second lemma shows how to determine whether the point is INSIDE or OUTSIDE that model.

Lemma 1 (Model Existence)
Given an approximate polygon P_{rep} and a signature α, construct $\tilde{C}_\alpha$ as described above. Then for each point v on the interior of $\tilde{C}_\alpha$, there exists some *model $P \in$ MODELS(P_{rep}) in which v has signature α.*

Proof Since $v \in \tilde{C}_\alpha$, for each i, $v \in H_i$ and there is some edge e_i in the band B_i which has v to the side specified by α_i. These edges join to form a model polygon P in which v has signature α. □

Lemma 2 (Point Location) *Given an approximate polygon P_{rep}, a model polygon $P \in$ MODELS(P_{rep}), and a point v which has a signature α with respect to P, the following are true:*

1. *If v is strictly to the interior of P_{rep} (that is, it does not lie on any band B_i) then α, v* INSIDE *P.*
2. *If v is strictly to the exterior of P_{rep} then v* OUTSIDE *P.*
3. *If $v \in B_i$, but $v \notin B_{i\pm1}$, then v* INSIDE *P* iff *$\alpha_i =$* R.
4. *If $v \in B_i \cap B_{i+1}$ and the $i/i+1$ corner is* convex, *then v* INSIDE *P* iff *$\alpha_i =$ R and $\alpha_{i+1} =$ R.*
5. *If $v \in B_i \cap B_{i+1}$ and the $i/i+1$ corner is* reflex, *then v* INSIDE *P* iff *$\alpha_i =$ R or $\alpha_{i+1} =$ R.*

Proof In Figure 3 the cases 1 through 5 are demonstrated by the points x_1 through x_5. □

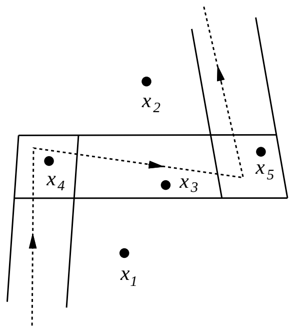

Figure 3: Cases for the Point Location Lemma

Given the Model Existence Lemma and the Point Location Lemma, a point location algorithm can be developed. This algorithm will construct the region $\tilde{C}_\alpha$, pick a point from its interior, and apply the rules of the Point Location Lemma to determine whether the point is INSIDE or OUTSIDE the model in which it has signature α. The following Uniqueness Theorem shows that if one such point is INSIDE its model polygon then *all* such points are INSIDE their respective model polygons (similarly for OUTSIDE).

Theorem 1 (Uniqueness) *Given an approximate polygon P_{rep} and a signature α, if for* some *model polygon in* MODELS(P_{rep}) *there is a point with signature α which is* INSIDE *the polygon, then, for* every *model polygon, all points which have signature α with respect to that polygon are* INSIDE *that polygon (similarly for* OUTSIDE*).*

The Uniqueness Theorem is the most interesting aspect of the point location problem. The proof is quite involved and is given in the full paper.

Point Location Algorithm

The Model Existence Lemma, Point Location Lemma, and Uniqueness Theorem combine to form the point location algorithm shown in Figure 4. Note that the algorithm is quite simple and never actually constructs the model polygon.

1. Compute $\tilde{C}_\alpha$.
2. If $\tilde{C}_\alpha = \emptyset$ then no model of P_{rep} induces a cell with signature α.
3. Pick a point w on the interior of $\tilde{C}_\alpha$.
4. Apply the Point Location Lemma to determine whether w is INSIDE or OUTSIDE of the models in which it has signature α.

Figure 4: Point Location Algorithm

Lemma 3 (Robustness) *The point location algorithm is robust.*

Proof This follows directly from the Model Existence Lemma and the Point Location Lemma. □

Lemma 4 (Consistency) *The approximate point location problem is consistent.*

Proof This follows directly from the Uniqueness Theorem. □

Lemma 5 (Complexity) *The point location algorithm has time complexity $\mathcal{O}(n^2)$.*

Proof Step 1 of the algorithm finds $\tilde{C}_\alpha$ by constructing in $\mathcal{O}(n^2)$ time the arrangement of the $3n$ lines defining the half-regions H_i. The other steps take constant time. □

Summary

Most geometric algorithms are not *robust*; they fail due to inexact input or with inexact intermediate computations. This paper has introduced (a) formal definitions of robustness and consistency, and (b) the notion of an *approximate polygon*, along with several of its properties. With these, one can formally develop robust and consistent algorithms that deal with inexact polygons.

One such algorithm for point location in an approximate polygon has been presented. The algorithm is particularly suited for practical application in a solid modeler because it assumes uncertainty in both the polygon position and the point position. The point location algorithm has been proved robust, and the point location problem has been shown to be consistent.

Acknowledgments

I am pleased to acknowledge the many helpful discussions which I have had with Chanderjit Bajaj and Paul Chew during the course of this work. Improvements to the algorithm were contributed by Victor Milenkovic, and this line of research was originally suggested by John Hopcroft.

My research has been supported by Nasa Training Grant NGT-50327. This work is also supported by Office of Naval Research Contract N00014-88K-0591, Office of Naval Research Grant N00014-89J-1946, and National Science Foundation Grant IRI-9006137.

References

[1] D. Dobkin and D. Silver. Recipes for geometry & numerical analysis - Part I: An empirical study. In *Annual Symposium on Computational Geometry*, pages 93–105, 1988.

[2] S. Fortune. Stable maintenance of point set triangulations in two dimensions. In *IEEE Annual Symposium on Foundations of Computer Science*, pages 494–499, 1989.

[3] C. M. Hoffmann, J. E. Hopcroft, and M. S. Karasick. Towards implementing robust geometric computations. In *Annual Symposium on Computational Geometry*, pages 106–117, June 1988.

[4] V. J. Milenkovic. *Verifiable Implementations of Geometric Algorithms Using Finite Precision Arithmetic.* PhD thesis, Carnegie Mellon University, 1988.

[5] D. Salesin, J. Stolfi, and L. Guibas. Epsilon geometry: Building robust algorithms from imprecise calculations. In *Annual Symposium on Computational Geometry*, pages 208–217, 1989.

[6] K. Sugihara and M. Iri. Construction of the Voronoi diagram for over 10^5 generators in single-precision arithmetic. In *Abstracts of the First Canadian Conference on Computational Geometry*, page 42, August 1989.