scrowe@hemel.bull.co.uk (Simon Crowe) writes:

>I am looking for an algorithm to determine if a given point is bound by a

>polygon. Does anyone have any such code or a reference to book containing

>information on the subject ?

Well, it's been a while since this was discussed so i take the liberty of

reprinting (without permission, so sue me) Eric Haines reprint of the very

interesting discussion of this topic...

/Jonas

------------------------ X snip snip X ----------------------------

"Give a man a fish, and he'll eat one day.

Give a man a fishing rod, and he'll laze around fishing and never do anything."

With that in mind, I reprint (without permission, so sue me) relevant

information posted some years ago on this very problem. Note the early use of

PostScript technology, predating many of this year's papers listed in the

April 1st SIGGRAPH Program Announcement posted here a few days ago.

-- Eric

Intersection Between a Line and a Polygon (UNDECIDABLE??),

by Dave Baraff, Tom Duff

From: deb@charisma.graphics.cornell.edu

Newsgroups: comp.graphics

Keywords: P, NP, Jordan curve separation, Ursyhon Metrization Theorem

Organization: Program of Computer Graphics

In article [...] ncsmith@ndsuvax.UUCP (Timothy Lyle Smith) writes:

> I need to find a formula/algorithm to determine if a line intersects

> a polygon. I would prefer a method that would do this in as little

> time as possible. I need this for use in a forward raytracing

> program.

I think that this is a very difficult problem. To start with, lines and polygons are semi-algebraic sets which both contain uncountable number of points. Here are a few off-the-cuff ideas.

First, we need to check if the line and the polygon are separated. Now, the Jordan curve separation theorem says that the polygon divides the plane into exactly two open (and thus non-compact) regions. Thus, the line lies completely inside the polygon, the line lies completely outside the polygon, or possibly (but this will rarely happen) the line intersects the polyon.

Now, the phrasing of this question says "if a line intersects a polygon", so this is a decision problem. One possibility (the decision model approach) is to reduce the question to some other (well known) problem Q, and then try to solve Q. An answer to Q gives an answer to the original decision problem.

In recent years, many geometric problems have been successfully modeled in a new language called PostScript. (See "PostScript Language", by Adobe Systems Incorporated, ISBN # 0-201-10179-3, co. 1985).

So, given a line L and a polygon P, we can write a PostScript program that draws the line L and the polygon P, and then "outputs" the answer. By "output", we mean the program executes a command called "showpage", which actually prints a page of paper containing the line and the polygon. A quick examination of the paper provides an answer to the reduced problem Q, and thus the original problem.

There are two small problems with this approach.

(1) There is an infinite number of ways to encode L and P into the reduced problem Q. So, we will be forced to invoke the Axiom of Choice (or equivalently, Zorn's Lemma). But the use of the Axiom of

Choice is not regarded in a very serious light these days.

(2) More importantly, the question arises as to whether or not the PostScript program Q will actually output a piece of paper; or in other words, will it halt?

Now, PostScript is expressive enough to encode everything that a Turing Machine might do; thus the halting problem (for PostScript) is undecidable. It is quite possible that the original problem will turn out to be undecidable.

I won't even begin to go into other difficulties, such as aliasing, finite precision and running out of ink, paper or both.

A couple of references might be:

1. Principia Mathematica. Newton, I. Cambridge University Press, Cambridge, England. (Sorry, I don't have an ISBN# for this).

2. An Introduction to Automata Theory, Languages, and Computation. Hopcroft, J and Ulman, J.

3. The C Programming Language. Kernighan, B and Ritchie, D.

4. A Tale of Two Cities. Dickens, C.

From: td@alice.UUCP (Tom Duff)

Summary: Overkill.

Organization: AT&T Bell Laboratories, Murray Hill NJ

The situation is not nearly as bleak as Baraff suggests (he should know better, he's hung around The Labs for long enough). By the well known Dobbin-Dullman reduction (see J. Dullman & D. Dobbin, J. Comp. Obfusc. 37,ii: pp. 33-947, lemma 17(a)) line-polygon intersection can be reduced to Hamiltonian Circuit, without(!) the use of Grobner bases, so LPI (to coin an acronym) is probably only NP-complete. Besides, Turing-completeness will no longer be a problem once our Cray-3 is delivered, since it will be able to complete an infinite loop in 4 milliseconds (with scatter-gather.)

From: deb@svax.cs.cornell.edu (David Baraff)

Well, sure its no worse than NP-complete, but that's ONLY if you restrict yourself to the case where the line satisfies a Lipschitz condition on its second derivative. (I think there's an '89 SIGGRAPH paper from Caltech that deals with this).

Jonas Yngvesson email: jonas-y@isy.liu.se
Dept. of Electrical Engineering voice: +46-(0)13-282162
University of Linkoping, Sweden fax : +46-(0)13-139282