Ray-surface intersection using recursive bounding box reduction technique
by Shriram Venkatraman

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science
Montana State University

Abstract:
In many practical applications, complex geometries are represented with the help of Non-Uniform B-Spline functions or other mathematical functions. A problem may arise when it is required to render such surfaces using known techniques like ray-tracing. One such problem during ray-tracing is to find the intersection of a ray incident on to the B-spline surface from a given point at a given direction. This thesis provides a solution to find points of intersection between the ray and the surface. The technique used here consists of reducing the convex hull of the surface recursively until its size is infinitesimally small and can be considered as the point of intersection.

RAY-SURFACE INTERSECTION USING RECURSIVE BOUNDING BOX REDUCTION TECHNIQUE

by

Shriram Venkatraman

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

January, 1995

APPROVAL

of a thesis submitted by

Shriram Venkatraman

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

1/27/95
Date

J Dunbigh Stanley
Chairperson, Graduate Committee

Approved for the Major Department

1/27/95
Date

J Dunbigh Stanley
Head, Major Department

Approved for the College of Graduate Studies

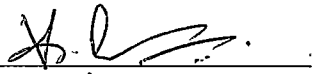
1/27/95
Date

Graduate Dean

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the US Copyright law. Request for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copright holder.

Signature 

Date 11/27/95

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my graduate committee members, *Dr. J. Denbigh Starkey*, *Dr. Gary Harkin*, and *Ray Babcock* and the rest of the faculty members from the Department of Computer Science for their unflagging effort and able support extended to me at every stage of my thesis and the graduate program.

TABLE OF CONTENTS

List of Figures

ABSTRACT

In many practical applications, complex geometries are represented with the help of Non-Uniform B-Spline functions or other mathematical functions. A problem may arise when it is required to render such surfaces using known techniques like ray-tracing. One such problem during ray-tracing is to find the intersection of a ray incident on to the B-spline surface from a given point at a given direction. This thesis provides a solution to find points of intersection between the ray and the surface. The technique used here consists of reducing the convex hull of the surface recursively until its size is infinitesimally small and can be considered as the point of intersection.

Introduction

In many real life applications involving geometrics, increasing attention has been paid to mathematical techniques of object representation in computer graphics. Parametric spline curves and surfaces are capable of modeling a wide variety of shapes. One such application that has used parametric surfaces is Boron Neutron Capture Therapy (BNCT) which is a tool for planning the treatment of certain inoperable brain tumors. In BNCT a 3D representation of the brain is formed based on the physician's interaction with planar images of the brain, originally produced from medical imaging modalities. All the bodies are modeled as B-Spline surfaces. When working with data or control points taken from MRI scan images of the brain, one problem that can arise is that the boundaries of the approximating functions of the brain and an existing tumor can overlap. To find such a boundary in order to prove the existence of a tumor, neutron rays are bombarded on the brain surface. The tumor reacting with neutron rays will be shown in a different color. The problem now is to find exactly at what point the neutron ray hits the surface, so that the point of intersection can be shown in different color. This thesis deals with the problem of finding the points of intersection of the neutron ray with the surface, and presents a solution. Although many other solutions may exist, the given solution is unique in such a way that it attempts to find the intersection points in the spatial domain itself, instead of numerical approaches.

Review of B-Splines

Three-dimensional, or space, curves and surfaces play an important role in engineering, design and manufacture of a diverse range of products, e.g., automobiles, ship hulls, aircraft fuselages and wings etc. They also play an important role in the description and interpretation of physical phenomena in the field of physics and medical science.

Surfaces are frequently described by a net of curves lying in orthogonal cutting planes plus three-dimensional feature or detail lines. An example is shown in Figure 1. These section curves are obtained by digitizing a physical model or a drawing and then fitting a mathematical curve to the digitized data. There are many techniques for obtaining a mathematical curve from digitized or *known* data, namely, cubic spline, parabolically blended curves, etc. These methods are generally referred to as curve *fitting* techniques, characterized by the fact that the derived mathematical curve *passes* through each and every data point.

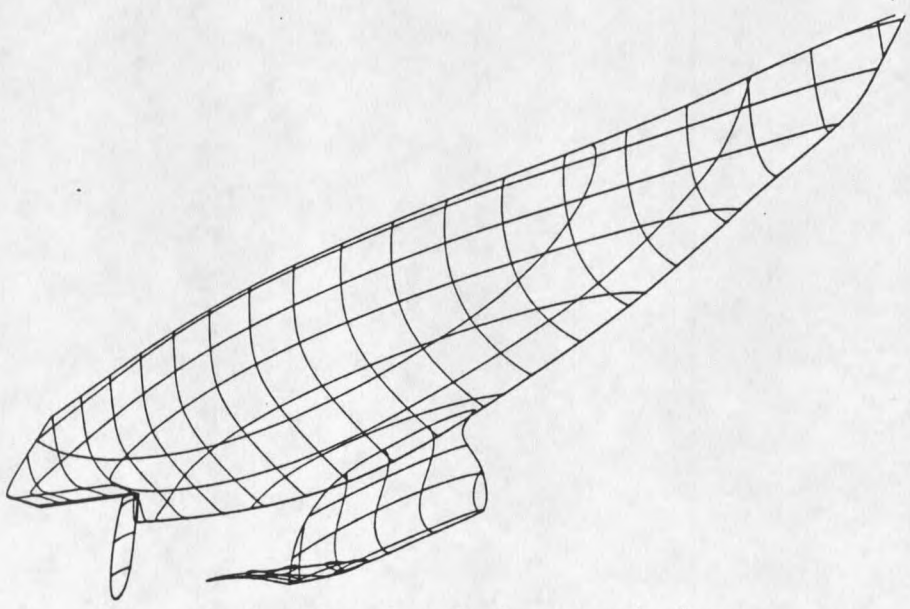


Figure 1: A surface described as a mesh of curves

Alternatively, the mathematical description of a space curve is generated *ab initio*, i.e., without any prior knowledge of the shape of the curve or surface. There are two techniques, namely, Bézier curves, and their powerful generalization **B-spline** curves, and they are characterized by the fact that few points on the curve pass through the control points which define the curve. These methods are referred to as curve *fairing* techniques.

Bézier Curves

The *fitting technique* for curve generation constrained the curves or surface to pass through existing data points. In many cases they produce excellent results, for instance, they are particularly suited for shape description, where the basic shape is arrived at by experimental evaluation or mathematical calculation. Examples are aircraft wings, engine manifolds, mechanical and/or structural parts. But in real time interactive graphics environment curve fitting methods are rendered ineffective. There is another class of shape design problems that is *both* aesthetic and *functional.* These problems are frequently termed *ab initio* design where the shape of the curve is not known *apriori.*

An alternate method of shape description suitable for *ab initio* design of free- form curves and surfaces was developed by Pierre Bézier. A Bézier curve is determined by its defining polygon as shown in Figure 2.

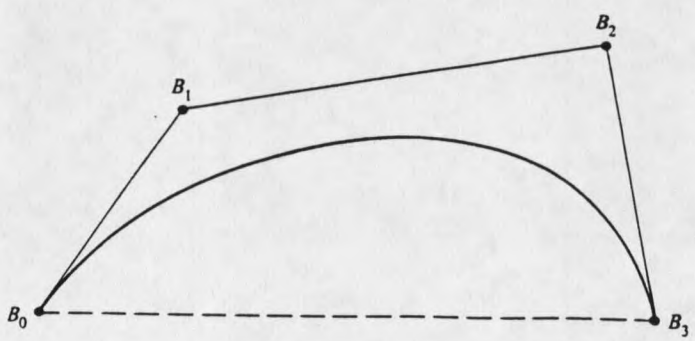


Figure 2: A Bézier curve and its defining polygon

Mathematically a parametric Bézier curve is defined by

$$P(t) = \sum_{i=0}^{n} B_i \, J_{n,i}(t) \qquad\qquad 0 \leq t \leq 1$$

where $J_{n,i}(t)$ is called the Bézier or Bernstein basis. $J_{n,i}(t)$ is also called the blending function where n is the degree of the basis function and thus of the polynomial curve segment and it is one less than the number of points in the defining Bézier polygon. Some other examples of four point Bézier polygons and the resulting cubic curves are shown in Figure 3.

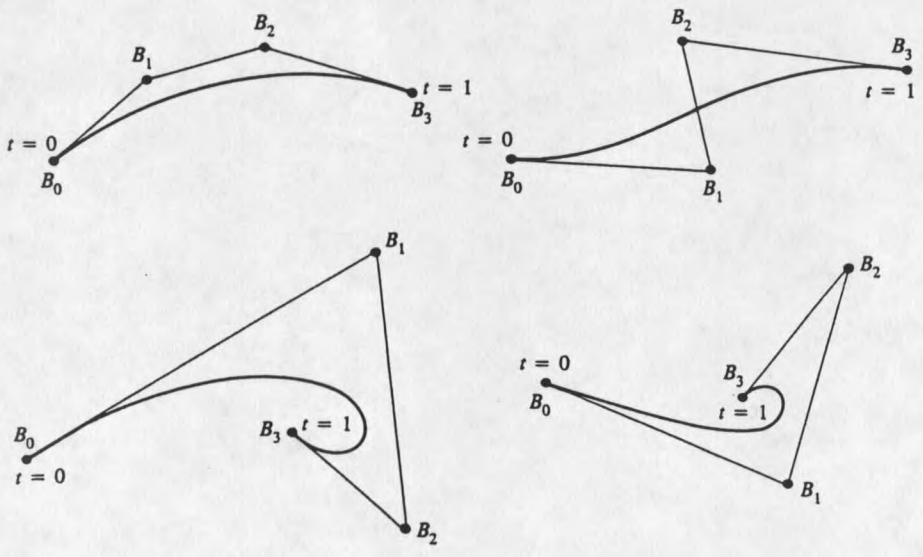


Figure 3: Bézier polygons for cubic curves

B-Spline Surface

From a mathematical point of view, a curve or surface generated by using the vertices of a defining polygon is dependent on some interpolation or approximation scheme to establish the relationship between the curve or surface and the polygon. This scheme is provided by the choice of the basis function. As noted above the Bernstein basis produces Bézier curves generated by the given equation. Two characteristics of the Bernstein basis, however, limit the flexibility of the resulting curves. First the number of specified polygon vertices fixes the order of the resulting polynomial which defines the curve. For example, a cubic curve must be defined by the polygon with four vertices and three spans. The only way to reduce the degree of the curve is to reduce the number of vertices, and vice-versa.

The second limiting characteristic is due to the global nature of the Bernstein basis, which means the value of the blending function $J_{n,i}(t)$ is nonzero for all parameter values over the entire curve. Since any point on a Bézier curve is a result of blending values of all the defining vertices, a change in one vertex is felt throughout the curve. This eliminates the ability to produce a local change within a curve. The same argument holds for a surface too.

There is another basis, called the B-Spline basis (from Basis Spline), which contains the Bernstein basis as a special case. This basis is generally nonglobal. The nonglobal behaviour of B-splines is due to the fact that each vertex of the control polygon B_i is associated with a unique basis function. Thus each vertex affects the shape of the spline only over a range of parameter values where its associated basis function is nonzero. The B-spline basis also allows the order of the basis function to be changed without changing the number of defining polygon vertices. B-spline surfaces are defined as the set of points obtained by evaluating the following equation for all values of u and v between some u_{min}, u_{max} and v_{min} and v_{max}. This recursive definition was

suggested by Cox and de Boor.

$$Q(u,v) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}\, N_{i,k}(u)\, M_{j,l}(v)$$

where

$N_{i,k}(u)$ and $M_{j,l}(v)$ are B-spline basis functions in u and v directions respectively and are defined as

$$N_{i,1}(u) = \begin{cases} 1, & \text{if } x_i \le u \le x_{i+1} \\ 0, & \text{otherwise.} \end{cases}$$

and

$$M_{j,1}(v) = \begin{cases} 1, & \text{if } y_j \le v \le y_{j+1} \\ 0, & \text{otherwise.} \end{cases}$$

Further,

$$N_{i,k}(u) = \frac{(u - x_i)\, N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)\, N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

and

$$M_{j,l}(v) = \frac{(v - y_i)\, N_{j,l-1}(v)}{y_{j+l-1} - y_j} + \frac{(y_{j+l} - v)\, N_{j+1,l-1}(v)}{y_{j+l} - y_{j+1}}$$

$N_{i,k}(u)$ and $M_{j,l}(v)$ are the basis functions in the biparametric u and v directions. x_i and y_j are the elements of a special kind of vector called the *knot* vector, whose elements are sequence of monotonically increasing numbers. There are various types of knot vectors and each of them influence the surface in a special way. The control points of the defining polygon is a (n+1) x (m+1) table of 3D points given by $B_{i,j}$.

k and l are the order of the B-splines in u and v directions.

By selecting an appropriate knot vector, the parametric intervals with care, and by not restricting the points $B_{i,j}$ to be unique, a number of closed and open surfaces can be created. A closed and an open surface are shown in Figure 4.

In general, B-splines are well known for their excellent geometric properties and characteristics. They are very well suited for surface design and representation. Some of their salient properties are :

- They are able to exactly represent geometric shapes.
- By manipulating the control points, they provide the flexibility to design a large variety of shapes.
- B-splines are invariant to affine transformation. Any affine transformation can be applied to the surface by applying to the defining polygon vertices; i.e., the curve is transformed by transforming the defining polygon vertices.
- B-splines lie within the convex hull of their defining polygon.
- B-splines generally follow the shape of the defining polygon.
- They provide the ability to make localized changes in their shapes, without affecting the entire curve/surface.

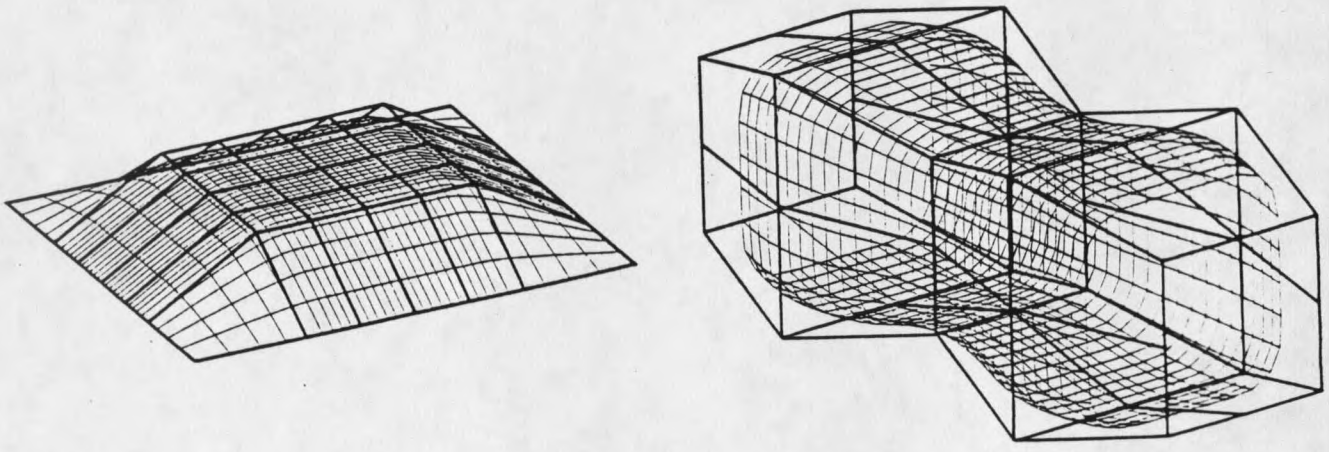


Figure 4: Open and closed B-Spline surface

Design and Description of the Algorithm

There are three steps that will need to take place in the execution of the program. The first is to create the surface or curves from the control points or data points taken from the actual scan image of the brain. Curves will most likely be created, as slices of the brain scan will be used. If this is the case then these curves will be placed on top of each other, using a common center, to create a 3D mesh surface. If the control points are taken from the three-dimensional model directly, then the surface will be created directly. Since the points used are actual control points, than an interpolation function is constructed and the surface is created using the formula given in the preceding discussion about B-splines.

The second step in the program is to create the ray. This is trivial, since, given a point of origin and the direction cosines of the ray, ray points can be generated upto any length of the ray in that direction.

The third step is to find a point of intersection between the ray and the surface created in that given direction, if such a point exists. The surface bounding box is created by finding the max and min x, y and z extents and connecting them together to make a box around the spline surface. The ray is first tested for intersection with that box, and if there is no intersection, then it is trivially accepted that no intersection of the ray and the surface exists. If, however there is an intersection then the box is divided into 8 smaller cubes and each of these cubes define a new bounding box for the next recursive step. Each of them is tested to see if they contain both the surface and the ray. The reduction continues till the size of the bounding box is as small as the pixel level. The other boxes which do not contain both the ray and the surface are rejected from further reduction. So only the smallest box which contains both the surface and the ray is taken as the