



Heuristics to isolate two closed B-Spline surfaces
by Rajeev Chawla

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in
Computer Science
Montana State University
© Copyright by Rajeev Chawla (1994)

Abstract:

In many real life applications involving geometries, a problem may arise if any two objects are intersecting where they should not be. This thesis presents a solution to this problem. The particular application that has been looked into is BNCT_Rtpe which is a tool for planning the treatment of certain types of inoperable brain tumors. In it a 3-D 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 closed B-Spline surfaces. For successful working of all the tools in the system, none of the bodies in this representation should intersect. In this thesis seven heuristics have been presented to isolate any two given closed B-Spline surfaces. The heuristic is based on marking the regions of intersection and then finding a vector based on these regions of intersection along which one of the surfaces should be moved so as to minimally separate it from the other surface. The last heuristic successfully minimally isolates any two given surfaces without destroying their relative orientation in space. All the methods preserve original shapes and differ primarily in the direction and magnitude of the movement of the surfaces. Future work includes truncation of the surfaces midway between the regions of intersection.

HEURISTICS TO ISOLATE TWO CLOSED B-SPLINE SURFACES

by
Rajeev Chawla

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

Montana State University
Bozeman, Montana
May 1994

71378
C3995

ii

APPROVAL

of a thesis submitted by

Rajeev Chawla

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.

5/31/94
Date

J. Dumbig Stanley
Chairperson, Graduate Committee

Approved for the Major Department

5/31/94
Date

J. Dumbig Stanley
Head, Major Department

Approved for the College of Graduate Studies

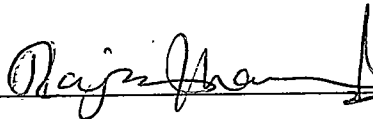
5/31/94
Date

A. Brown
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 U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Signature 

Date May 31, 1994

ACKNOWLEDGMENTS

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

TABLE OF CONTENTS

INTRODUCTION	1
REVIEW OF B-SPLINE SURFACES	2
HEURISTICS FOR ISOLATION	4
Maximum Distance	4
Maximum Distance in the Region of Intersection	6
Perpendicular Vector	8
Weighted Average Perpendicular Vector	9
Center of Mass	10
Weighted Average of CM and the Weighted Average Perpen- dicular Vector	11
RESULTS	13
REFERENCES	14

LIST OF FIGURES

1	Maximum Distance Between the Two Surfaces	5
2	Maximum Distance in the Regions Of Intersections	7
3	Finding Approximate Perpendicular Vector to $\mathfrak{R}$	9
4	The Perpendicular Vector Method	10
5	Weighted Average of Perpendicular Vectors	11
6	The Center of Mass Method	12

ABSTRACT

In many real life applications involving geometries, a problem may arise if any two objects are intersecting where they should not be. This thesis presents a solution to this problem. The particular application that has been looked into is BNCT_Rtpe which is a tool for planning the treatment of certain types of inoperable brain tumors. In it a 3-D 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 closed B-Spline surfaces. For successful working of all the tools in the system, none of the bodies in this representation should intersect. In this thesis seven heuristics have been presented to isolate any two given closed B-Spline surfaces. The heuristic is based on marking the regions of intersection and then finding a vector based on these regions of intersection along which one of the surfaces should be moved so as to minimally separate it from the other surface. The last heuristic successfully minimally isolates any two given surfaces without destroying their relative orientation in space. All the methods preserve original shapes and differ primarily in the direction and magnitude of the movement of the surfaces. Future work includes truncation of the surfaces midway between the regions of intersection.

INTRODUCTION

In many practical applications, complex geometries are represented with the help of Non-Uniform Rational B-Spline functions or other similar mathematical functions. A problem may arise if any two objects in this representation are intersecting where they should not be. The particular application in which this problem was encountered is BNCT_Rtpe [15].

The most malignant brain tumors are not well treated by any currently accepted procedure. Boron Neutron Capture Therapy [7] is an idealized form of cancer therapy providing a mechanism to destroy cancerous cells without injuring adjacent normal cells. Neutron Capture Therapy was first proposed in 1932, but the next sixty years were required for the necessary technologies to evolve.

BNCT_Rtpe is a tool for planning this treatment. The physician is provided with several images of a given patient, originally produced from medical imaging modalities such as CT-SCAN, MRI etc. A 3-D representation of the brain is formed, based on the physician's interaction with the planar images. The brain along with all the anatomical objects inside it are modeled as B-Spline surfaces [15]. This project brought up the problem of interfering surfaces, because for successful working of all the tools in the system (in particular, an existing Monte Carlo model of neutron transport), none of the bodies in this representation should intersect. These heuristics have been designed as a general tool for minimally moving the interfering bodies (closed B-Spline surfaces), while maintaining their relative orientation in space, so as to just isolate them. The vector along which the surfaces are to be moved depends on the regions of intersection which may be easily found by using any of the common bounding box techniques [1, 6, 8, 9, 11, 12, 13, 14].

REVIEW OF B-SPLINE SURFACES

B-Splines are well known for their excellent geometric properties and characteristics and are very well suited for curve and surface design and representation.

- o They are able to exactly represent complex geometric shapes.
- o By manipulating the control points, they provide the flexibility to design a large variety of shapes.
- o B-Splines are invariant with respect to affine transformations.
- o B-Splines have a clear geometric interpretation, making them particularly useful for designers.
- o Provide the ability to model imbedded flat regions and lines of sharp discontinuities.
- o Provide the ability to make localized changes in the shapes.

B-Spline surfaces are defined as the set of points obtained by evaluating equation 1 [1, 3, 4, 5, 6, 10, 11] for all values of u and v between some u_{min} , v_{min} and u_{max} and v_{max} .

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

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

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

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

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

$N_{i,k}(u)$ and $M_{j,l}(v)$ are the B-Spline basis functions in the biparametric u and v directions respectively. x_i and y_j are the elements of knot vectors and $B_{i,j}$ are the vertices of a defining polygon net.

HEURISTICS FOR ISOLATION

We present the six heuristics, which are also the steps in which the final heuristic evolved. Given any two closed B-Spline surfaces $S1$ and $S2$, where $Q(a, b)$ and $P(l, m)$ are the sets of points defining $S1$ and $S2$ respectively,

$$\begin{aligned} Q(a, b) &= (Q(a, b)_x, Q(a, b)_y, Q(a, b)_z), \\ P(l, m) &= (P(l, m)_x, P(l, m)_y, P(l, m)_z), \end{aligned}$$

a is the number of points in the u parametric direction and b is the number of points in the v parametric direction. l and m are similarly defined for $S2$. Our aim is to find a vector

$$\vec{M} = iM(x) + jM(y) + kM(z)$$

such that, when it is added to one of the surfaces, it gets isolated from the other surface. After having moved one of the surfaces, we check again for intersection and if they are still intersecting, follow the same procedure for isolation. This procedure is repeated until they get isolated. Once they are isolated, one of the surfaces can be moved back along the same vector such that the two surfaces are just separated by some distance ϵ . These heuristics preserve the shapes of the surfaces $S1$ and $S2$.

Maximum Distance

Given two closed surfaces, find the maximum distance between the two surfaces. I.e., find

$$|\vec{M}|_{max} = \max[|Q(a, b) - P(l, m)|]$$

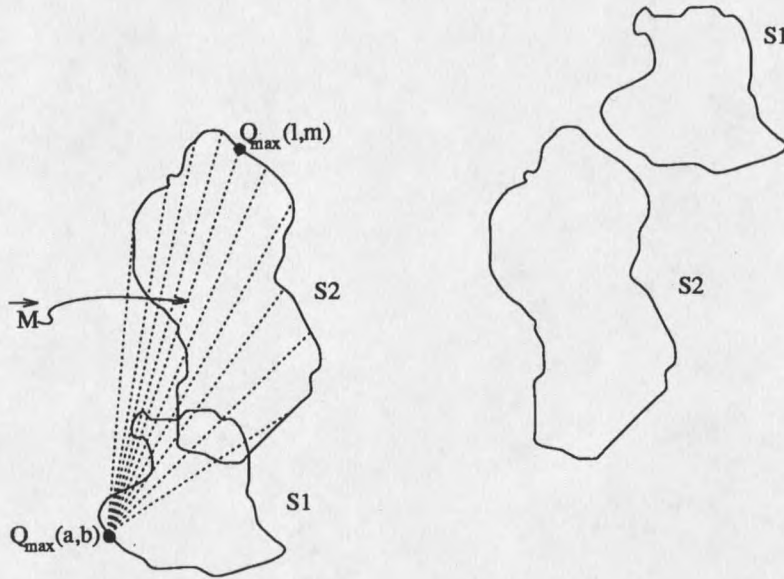


Figure 1: Maximum Distance Between the Two Surfaces

for all a, b, l, m . Also if $Q_{max}(a,b)$ and $P_{max}(l,m)$ are the points on the boundaries of the two surfaces corresponding to $|\vec{M}|_{max}$, then

$$\begin{aligned} M(x) &= [Q_{max}(a,b)]_x - [P_{max}(l,m)]_x, \\ M(y) &= [Q_{max}(a,b)]_y - [P_{max}(l,m)]_y, \quad \text{and} \\ M(z) &= [Q_{max}(a,b)]_z - [P_{max}(l,m)]_z, \end{aligned}$$

where

$$\begin{aligned} Q(a,b) &= ([Q(a,b)]_x, [Q(a,b)]_y, [Q(a,b)]_z) \quad \text{and} \\ P(l,m) &= ([P(l,m)]_x, [P(l,m)]_y, [P(l,m)]_z). \end{aligned}$$

Surface $S1$ crosses over to the other side of surface $S2$. So the result is not at all acceptable(Figure 1), as the relative orientation of the surfaces might change to a considerable extent.

Maximum Distance in the Region of Intersection

Here, the search for the maximum magnitude vector is limited to the regions of intersection. If $R1(n)$ and $R2(p)$ represent the set of points in the regions of intersections, on the first and second surface boundaries respectively, then

$$|\vec{M}|_{max} = \max[|R1(n) - R2(p)|]$$

for all n, p . Here,

$$\vec{M} = i * mv_x M(x) + j * mv_y M(y) + k * mv_z M(z),$$

$$M(x) = |R1_{max}(n)_x - R2_{max}(p)_x|,$$

$$M(y) = |R1_{max}(n)_y - R2_{max}(p)_y|, \text{ and}$$

$$M(z) = |R1_{max}(n)_z - R2_{max}(p)_z|.$$

$R1_{max}(n)$ and $R2_{max}(p)$ are the points on the boundaries of the two surfaces corresponding to $|\vec{M}|_{max}$ and

$$R1_{max}(n) = (R1_{max}(n)_x, R1_{max}(n)_y, R1_{max}(n)_z),$$

$$R2_{max}(p) = (R2_{max}(p)_x, R2_{max}(p)_y, R2_{max}(p)_z).$$

mv_x, mv_y, mv_z have the values 1 or -1. The rule here is that a surface is moved in the direction in which it extends more than the other surface. So if S1 extends in the negative x direction more than S2, then $mv_x = -1$ else $mv_x = 1$ etc. If $\min x1 = \min[Q(a, b)_x]$, $\max x1 = \max[Q(a, b)_x]$ and $\min x2$ and $\max x2$ are similarly defined for S2, then:

if $\min x1 \geq \max x2$	$mv_x = 1$
else if $\max x1 \leq \min x2$	$mv_x = -1$
else if $\max x1 > \max x2$	$mv_x = 1$
else if $\min x1 > \min x2$	$mv_x = 1$
else	$mv_x = -1$

An example is shown in Figure 2.

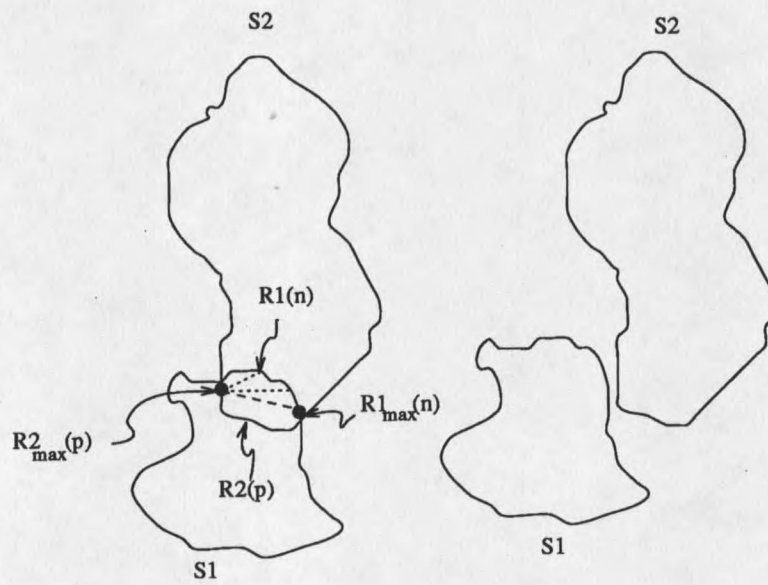


Figure 2: Maximum Distance in the Regions Of Intersections

Perpendicular Vector

The first method does not take into account the relative orientation of the two surfaces in the space and hence does not give a good direction of movement. The two surfaces are distributed around the region formed by joining the endpoints of two consecutive lines of intersection. So, a better direction of movement would be the approximate perpendicular to this region. If there are several regions of intersection, move perpendicular to the one corresponding to the region that contains $|\vec{M}|_{max}$. Again,

$$\vec{M} = i \star mv_x M(x) + j \star mv_y M(y) + k \star mv_z M(z).$$

We again find $|\vec{M}|_{max} = \max[|R1(n) - R2(p)|]$ for all n,p. Let $\mathfrak{R}$ represent the region of intersection corresponding to $|\vec{M}|_{max}$ and let L_x and L_{x+1} be the lines of intersection flanking $\mathfrak{R}$. If $\vec{V}1$ and $\vec{V}2$ are the two vectors formed by joining the opposite endpoints of the lines L_x and L_{x+1} (Figure 3), we can easily find a unit vector $\vec{V} = i\vec{V}_x + j\vec{V}_y + k\vec{V}_z$, approximately perpendicular to $\mathfrak{R}$ as

$$\vec{V} = \frac{\vec{V}1 \times \vec{V}2}{|\vec{V}1 \times \vec{V}2|} \quad (2)$$

We may want to move along $\vec{V}$ or $-\vec{V}$. Let θ be the angle between $\vec{V}$ and $\vec{M}_{max}$ where $\vec{M}_{max}$ is the vector corresponding to $|\vec{M}|_{max}$.

$$\vec{V} = \begin{cases} \vec{V} & \text{if } -90 \leq \theta \leq 90 \\ -\vec{V} & \text{otherwise} \end{cases} \quad (3)$$

Then, $mv_x = \vec{V}_x, mv_y = \vec{V}_y, mv_z = \vec{V}_z$. Also, if

$$\begin{aligned} \vec{M}_{max} &= iM_{x,max} + jM_{y,max} + kM_{z,max}, \text{ and} \\ max_dist &= \max[|M_{x,max}|, |M_{y,max}|, |M_{z,max}|], \end{aligned}$$

then

$$M(x) = M(y) = M(z) = max_dist.$$

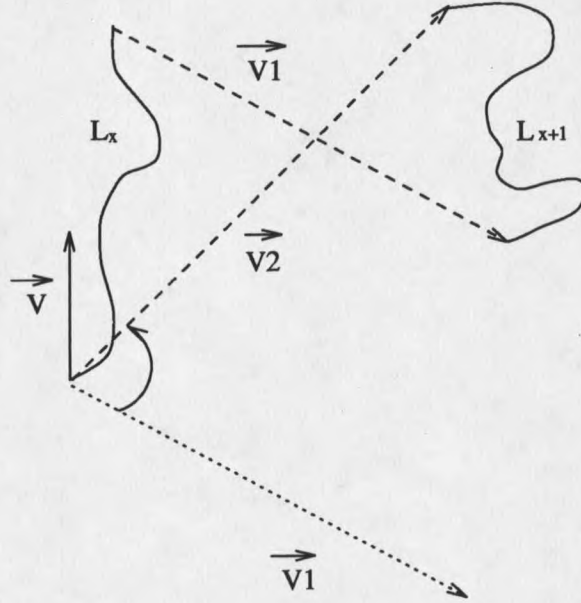


Figure 3: Finding Approximate Perpendicular Vector to $\mathcal{R}$

An example is shown in Figure 4.

Weighted Average Perpendicular Vector

A better measure of the relative distribution of the two surfaces in the space may be obtained by looking at all the regions of intersection [2]. So number the regions of intersection such that the region containing the maximum distance vector is numbered 1 and the other regions are sequentially numbered 1, 2 etc. (Figure 5), going away in both the directions (left and right) from region 1. $\vec{V}_{r,1}, \vec{V}_{r,2}, \dots, \vec{V}_{r,nr}, \vec{V}_{l,1}, \vec{V}_{l,2}, \dots, \vec{V}_{l,nl}, \vec{V}_{c,1}$ are the unit vectors (Figure 5) perpendicular to the regions $\mathcal{R}_{r,1}, \mathcal{R}_{r,2}, \dots, \mathcal{R}_{r,nr}, \mathcal{R}_{l,1}, \mathcal{R}_{l,2}, \dots, \mathcal{R}_{l,nl}, \mathcal{R}_{c,1}$ respectively (see equation 2), nr is the number of regions of intersection on the right side of region 1 and nl is the number of

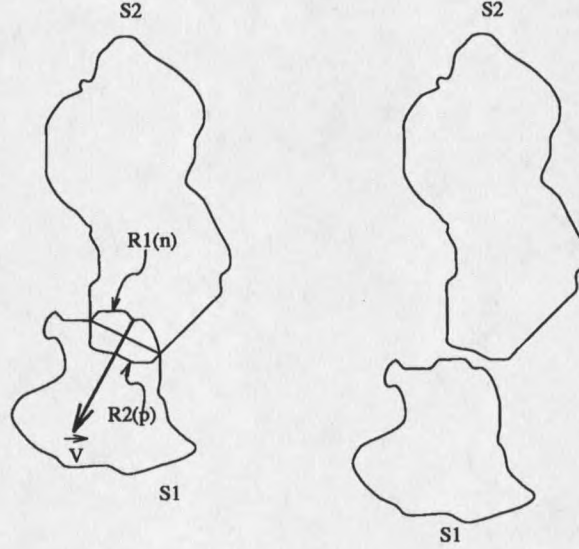


Figure 4: The Perpendicular Vector Method

regions of intersection on the left side of region 1. All the unit vectors should be modified by multiplying them by -1, wherever necessary, as explained in equation 3. So, the final unit vector $\vec{V}$ is then given by:

$$\vec{V}' = \frac{\sum_{i=2}^{nr} f(i) \vec{V}_{r,i} + \sum_{j=2}^{nl} f(j) \vec{V}_{l,j} + \vec{V}_{c,1}}{nr + nl + 1}.$$

$$\vec{V} = \frac{\vec{V}'}{|\vec{V}'|}$$

The function f can easily be found by experimenting with different functions. For most of the cases, $f(i) = 1/\sqrt{i+1}$ works very well.

Center of Mass

If the surfaces are evenly distributed around the regions of intersection,

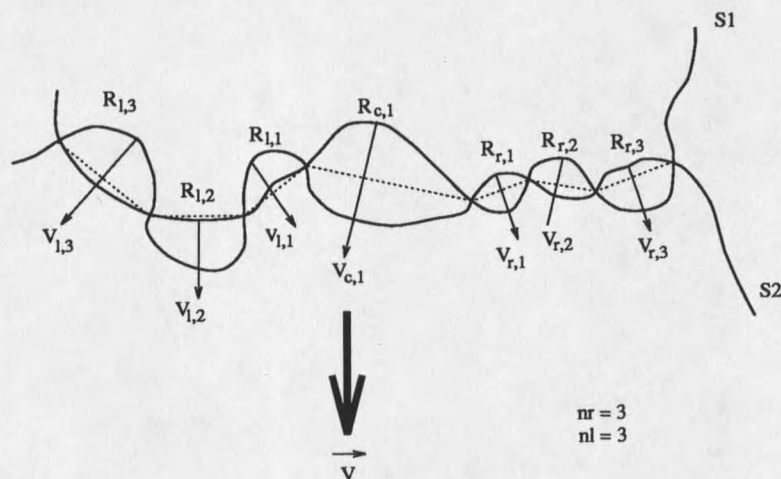


Figure 5: Weighted Average of Perpendicular Vectors

an easy way to find a good direction of movement is to find their centers of mass and move along the line joining them (Figure 6). If $\vec{V} = i\vec{V}_x + j\vec{V}_y + k\vec{V}_z$ is the unit vector along this line, then $mv_x = \vec{V}_x$, $mv_y = \vec{V}_y$, and $mv_z = \vec{V}_z$. $M(x)$, $M(y)$ and $M(z)$ would be calculated as defined in the *Perpendicular Vector* method for isolation.

Weighted Average of CM and the Weighted Average Perpendicular Vector

Surfaces will not be evenly distributed around the regions of intersection in many cases. But we can still combine the above two methods by taking the weighted average of the two vectors (directions) calculated using the above two methods to get an improved direction of movement. So the final vector $\vec{V}$ would be given as:

$$\vec{V} = \frac{\vec{V}'}{|\vec{V}'|}$$

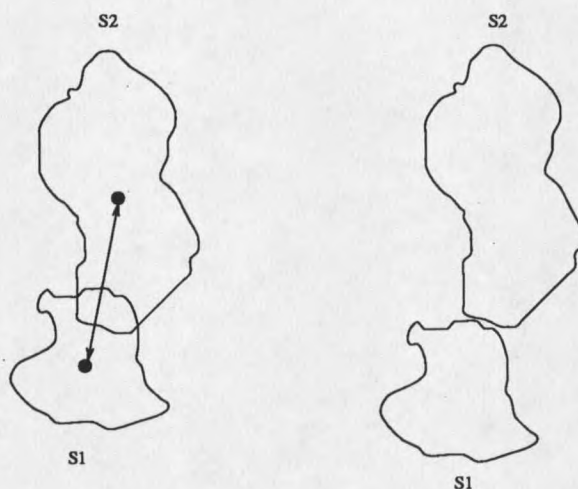


Figure 6: The Center of Mass Method

$$\vec{V}' = a\vec{V}_1 + b\vec{V}_2,$$

where $\vec{V}_1$ and $\vec{V}_2$ are the directions of movement as calculated in methods 4 and 5 respectively and a and b are the weights of the two vectors that we want to consider for our final direction of movement. $M(x)$, $M(y)$ and $M(z)$ would be calculated as explained in the third method.

RESULTS

Six different heuristics have been presented in this thesis. I have not yet developed an analytical method for the performance analysis of these methods and there is no existing method for measuring the performance of such heuristics. Infact, it would take another thesis to actually develop and implement a measure of performance. So the results presented here are based on the visual observations of the test runs.

Visual observations of the test runs show that the sixth heuristic is the most efficient one. The weighted average of perpendicular vectors method is superior to the other four methods in most of the cases. In the cases where the two surfaces are very evenly distributed around the regions of intersections, the center of mass method performs slightly better than the weighted average of perpendicular vectors method. The first three methods do not take into account the distribution of all the regions of intersection and their relative distribution around the areas of intersections. So they do not work well in most of the cases, although the cases where direction of movement largely depends upon the area of intersection containing the maximum distance vector, the perpendicular vector method performs satisfactorily. The last method accounts for all the regions of intersection and the relative distribution of the two surfaces, by taking a partial weight of the direction vector obtained in the center of mass method, and so works successfully and performs better than all the other methods in all the cases.

REFERENCES

- [1] Richard H. Bartels, John C. Beatty, and Brian A. Barsky. *An Introduction to Splines for Use in Computer Graphics & Geometric Modeling*. Morgan Kaufmann, 1987.
- [2] Rajeev Chawla, Denbigh Starkey, Dan Wessol, and Ray Babcock. A heuristic to isolate interfering bodies in the BNCT treatment planning system. In *Seventh Annual IEEE Symposium on Computer-Based Medical Systems*, Winston-Salem, NC, June 1994.
- [3] M.Ġ. Cox. The numerical evaluation of b-Splines. *National Physics Laboratory DNAC*, 4.
- [4] C. deBoor. On calculation with b-Splines. *Journal of Approximation Theory*, 6:50-62, 1972.
- [5] Gerald Farin. From conics to NURBS: A tutorial and survey. *IEEE Computer Graphics & Applications*, pages 78-86, September 1992.
- [6] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, second edition, 1990.
- [7] Frederick M. Hawthorne. The role of chemistry in the development of boron neutron capture therapy of cancer. *Angew. Chem. Int. Ed. Engl.*, pages 950-984, January 1993.
- [8] James T. Kajiya. New techniques for ray tracing procedurally defined objects. *ACM Transactions on Graphics*, pages 161-181, July 1983.

- [9] Tomoyuki Nishita, Thomas W. Sederburg, and Masanori Ka kimoto. Ray tracing trimmed rational surface patches. pages 337-345, 1990.
- [10] Les Piegl. On NURBS: A survey. *IEEE Computer Graphics & Applications*, pages 55-71, January 1991.
- [11] David F. Rogers and J. Alan Adams. *Mathematical Elements for Computer Graphics*. McGraw-Hill, second edition, 1990.
- [12] S.M. Rubin and T. Whitted. A 3-dimensional representation for fast rendering of complex scenes. In *Computer Graphics(Proceedings of the SIGGRAPH)*, pages 110-116, 1980.
- [13] Michael A.J. Sweeney and Richard H. Bartles. Ray tracing free-form B-Spline surfaces. *IEEE Computer Graphics & Applications*, pages 41-49, February 1986.
- [14] Alan Watt and Mark Watt. *3D Computer Graphics*. Addison-Wesley, second edition, 1993.
- [15] Daniel E. Wessol and Floyd J. Wheeler. Creating and using a type of free-form geometry in monte carlo particle transport. *Nuclear Science and Engineering*, pages 314-323, August 1992.

