7K-5

Geometric method to find the intersection of a general parametric curve with the natural quadrics

Aidan G. O'Neill, Teiji Takamura

RICOH Co., Ltd. Software Research Center

1 Introduction

The calculation of intersection curves between surfaces is one of the major current challenges in computer aided design. A class of design system called a solid or geometric modeler, which uses a unified data structure, is increasingly acquiring the capability to represent objects with sculptured or free-form surfaces. Subsequently, the intersection problem becomes difficult when such surfaces are involved. The requirements are that such surface intersection calculations be robust, accurate and fast. This paper will concentrate on the intersection of a general parametric curve with a special quadratic surface type, that of the natural quadrics, i.e. sphere, cylinder and cone. The method that we propose consists of two stages. The first is to get rough intersecting points which are used as starting points for the second step. These points are then refined by our geometric intersection method, known hereafter as the Geometric Newton-Raphson algorithm. The starting points are calculated using a sub-division technique.

2 Current techniques used

The following techniques are used in calculating intersections in CAD:

Algebraic approach Solve algebraic equations based on the curve and surface.

Iterative/numerical techniques Find the intersections by using an iterative method like recursive subdivision.

Geometric methods Calculate the intersections based on the geometric relationship between the curve and the surface.

The main idea in the **Algebraic approach** in computing the intersection of a curve and a surface is to solve by analytic techniques the real roots of a rational polynomial representing the curve and the surface (as in the Elimination method proposed by Sederberg). This approach is effective as analysts have developed efficient algorithms to solve the equation if the degree is low [1], but usually there is a quick growth in the degree of the resultant polynomial when free-form surfaces are involved. Efficient algorithms have been developed to solve intersections with Steiner patches [2], but the patches themselves lack good representation.

Iterative techniques are based on the *divide and conquer* concept, and they are useful when Bézier or B-spline curve formats are used. These techniques are accurate [4], but there are problems with touch point cases and extensive computations. It is also easy to miss intersection points [5]. An extension of the elimination method mentioned above has been proposed by Chandru and Kochar, which uses analytic techniques in conjunction with numerical techniques known as a semi-analytic approach, but that idea is not yet developed [6].

The **Geometric approach** has major advantages over the above methods. This method consists of giving a surface a special type code plus type dependent variables [3]. Numerical errors are minimized and small changes in data induce equally small changes in the location of points. The problems with this method are increased amounts of code for special case handling and different algorithms for different surface types. The method that we propose solves the problem of requiring different code for different surfaces by using a tangent plane of the surface concerned, so that the method can be used universally.

3 The Geometric Newton-Raphson Algorithm

Our *Geometric Newton-Raphson* algorithm is based on an enhanced form of the general geometric approach. We first calculate rough intersecting points between the curve and the surface, and then we try to converge those points into accurate intersection points using the geometric information of the curve and surface. All starting points between the intersecting curve and the quadric surface are calculated using a recursive subdivision method as follows:

1. A rough bounding box defined by the control points of the intersecting curve is checked for intersection with the surface. If no intersection, then exit.
2. If an extended line defined from the end points of the curve intersects the quadric surface, that intersection point(s) becomes a new starting point, and exit.
3. If it does not intersect, the curve is split into two

自然二次曲面と一般パラメトリック曲線との干渉計算方法
エイドン=オニール, 高村禎二
(株)リコー

pieces at a parameter value $t = 0.5$, and each piece is defined in terms of a new smaller polygon whose points lie closer to the surface. The splitting of the curve is achieved by using the de Casteljau algorithm. For each of the split curves, go to step 1.

Once all the initial starting points are calculated, call the Geometric Newton-Raphson method to converge the points. The algorithm is as follows:

1. Get a starting point parameter t_0 from the above method.
2. Calculate the point on the intersection curve $\mathbf{C}(t_0)$ and the derivative vector of the curve $\mathbf{C}'(t_0)$ at the specified parameter t_0. (See fig. 1).
3. Project $\mathbf{C}(t_0)$ onto the quadric surface, and get the projected point on the surface $\mathbf{Q}$. $\mathbf{Q}$ is the nearest point on the surface from the point $\mathbf{C}(t_0)$.
4. If the distance between $\mathbf{C}(t_0)$ and $\mathbf{Q}$ is below a specified tolerance , $\mathbf{C}(t_0)$ is an accurate intersection point, and exit.
5. Get the intersection point between the tangent plane which passes $\mathbf{Q}$ and the tangent line which passes $\mathbf{C}(t_0)$. Call this point $\mathbf{T}$.
6. $\mathbf{T} - \mathbf{C}(t_0) = \Delta t \cdot \mathbf{C}'(t_0)$ therefore by multiplying $(\mathbf{T} - \mathbf{C}(t_0) + \mathbf{C}'(t_0))$ to both sides, we get a scaler equation. By solving we get Δt.
7. Set the new t_0 to $t_0 + \Delta t$, and go to step 2.

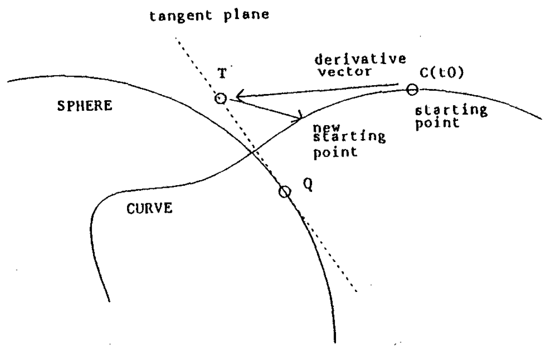

fig 1. The Geometric Newton Raphson algorithm converging.

This algorithm works efficiently as little time is spent in calculating the initial starting points, and the Geometric Newton-Raphson method converges quickly to an accurate intersection point. The algorithm is robust because enough initial points are generated to obtain all intersection points. Also the equation got in step 6 is very stable as it is a function of one variable. Touch point cases are taken care of by using a bounding box for the intersecting curve segment and also for the quadric surface. The disadvantage of this is that more starting points than necessary are generated but its advantage is that no points will be missed, and also the number of iterations is low. A special form of this algorithm which is used in planar surface intersections can be seen in [7]

Figures 2 and 3 show examples of an intersection curve and a cone. The intersection points are marked.

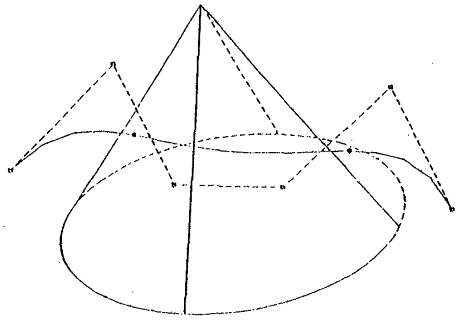

fig 2. Intersection of a rational Bézier curve and a cone.

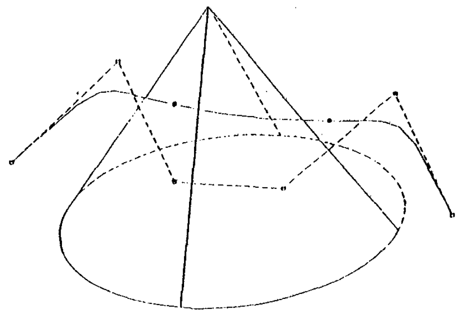

fig 3. Intersection of a rational Bézier curve and a cone. The weight at some control points has been changed.

4 Conclusion

A geometric method to find the intersections between a curve and the natural quadrics is presented. We show how all intersection points can be accurately found with reasonable computing effort. The main advantages of this method include its reliability and speed in finding accurate intersecting points. This method is implemented as part of the set operations in the solid modeler DESIGN-BASE, developed at the Ricoh Software Research Center.

References

[1] Sederberg T.W., "Implicit and parametric curves and surfaces for computer aided geometric design", *Ph.D. thesis, Purdue Univ.*, 1983.

[2] Sederberg T.W., "Steiner Surface Patches", *IEEE Computer Graphics and Applications*, Vol. 5, No. 5, May 1985, pp. 23-36.

[3] Miller, J. R. "Geometric Approaches to Nonplanar Quadric Surface Intersection Curves" *ACM Transaction on Graphics*, Vol. 6, No. 4, October 1987, pages 274-307.

[4] Barnhill R.E. and Farin G. and Jordan M. and Piper B.R., "Surface/surface intersection", *Computer-Aided Geometric Design*, Vol. 4, No. 1, 1987, pp. 3-16.

[5] Trimmer H. G., "Analytic background for computation of surface intersections", Douglas Aircraft Technical Memo ,April 1977.

[6] Chandru V. and Kochar B. S., "Analytic Techniques for Geometric Intersection Problems", *Geometric Modeling:Algorithms and New Trends*, 1987.

[7] Fujisawa N. and Takamura T., "任意の次数の自由曲線と平面の干渉計算方法" 情報処理学会第 39 回全国大会, 1989.

[8] Chiyokura H., "Solid Modeling with DESIGNBASE", Addison-Wesley, 1988.