Offsets of Rational Surfaces & Ray Tracing.

Andrzej Łukaszewski

Institute of Computer Science
University of Wrocław
ul. Przesmyckiego 20
51-151 Wrocław, Poland

http://www.tcs.uni.wroc.pl/~anl
email: anl@tcs.uni.wroc.pl

June 24, 1999

Andrzej Łukaszewski Wrocław

Abstract

Offsets or parallel surfaces were used since Leibniz and are important means in robotics, CAGD and tolerance analysis ([3]). Presented algorithm solves the problem of finding all intersections between a line and an offset of a rational parametric surface. The main idea is to eliminate the normal vector from the intersection calculations. The square root which is present in normal vector calculations is responsible for the complicated nature of the offset as well as for the high cost of direct solving of intersection equations. Instead of using the normal vector this algorithm uses extensively the distance function and it follows Hoffmann dimensionality paradigm [1].

For rational surface $S(u,v)$ of degree (m,n) the intersection problem is reformulated into the problem of solving an array of equations: $F(u,v,t) = 0$, $\partial_u F(u,v,t) = 0$, $\partial_v F(u,v,t) = 0$ where function F is polynomial in Bézier form of degree (2m,2n,2) (F is simplification of $f(u,v,t) = ||S(u,v) - R(t)||^2 - d^2$). Array of equations is solved by subdivision method and the solutions are classified because they can belong either to positive offset $S_d(u,v)$ or to the negative one $S_{-d}(u,v)$.

Simplicity of the algorithm makes it efficient and easy to implement. The complexity analysis and the experimental results show that for polynomial surfaces of degree 2–4 the method is at least 5 times faster than the one proposed by M.E.Vafiadou and N.M.Patrikalakis [4]. If we compare it with the obvious generalization of [4] to rational surfaces our method is at least 30 times faster and it is even faster for the surfaces of higher degree. The algorithm can be used in the ray tracing method ([5]) for offset visualization as well as in the offset shape interrogation.

Definitions & Problem Statement

Definition 1. *A parametric suface $S(u,v)$ is $regular$ if its partial derivatives are nowhere equal to zero and nowhere collinear.*

For the regular parametric surface $S(u,v)$ we can define the normal vector using vector product:

$$N(u,v) = \frac{\partial_u S(u,v) \times \partial_v S(u,v)}{||\partial_u S(u,v) \times \partial_v S(u,v)||}$$

Definition 2. *For a regular progenitor surface $S(u,v)$ we can now define the $offset$ surface at the distance d as:*

$$S_d(u,v) = S(u,v) + dN(u,v)$$

It is a displacement of the original surface in the direction of the normal vector.

We will use Bézier form of rational surfaces and volumes (generalization to three variables) as tensor products.

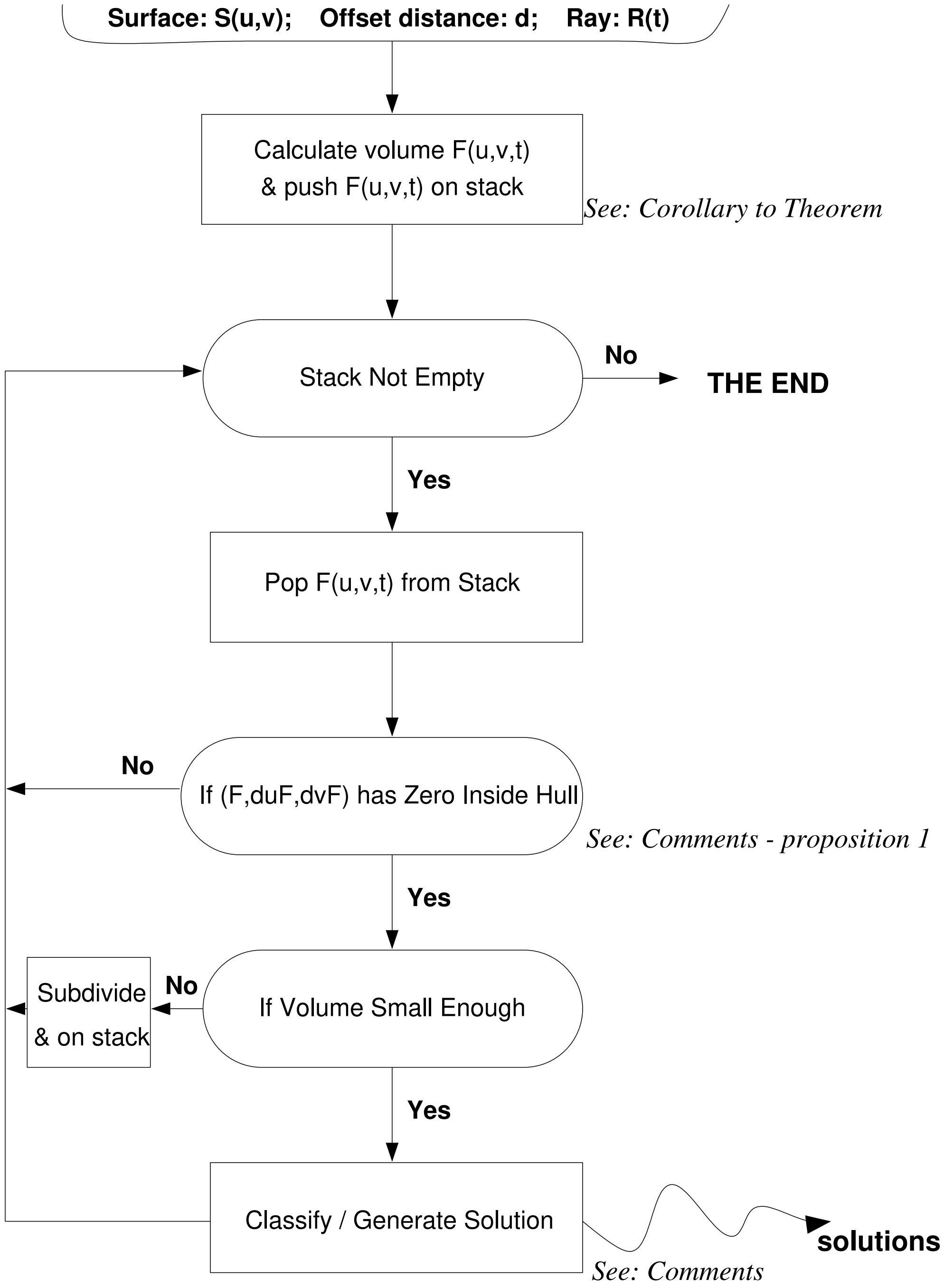
Even for simple polynomial surfaces their offsets are inherently more complex.

Andrzej Łukaszewski Wrocław

Offsets to Sweep of Parabola

The picture shows sweep of parabola with two offsets at distances $d = -0.5$ (below) and $d = 0.7$ (above). Even for such simple shape the offset surfaces at distance greater than the curvature radius can have self intersections.

Algorithm


Surface: S(u,v); Offset distance: d; Ray: R(t)
Calculate volume F(u,v,t)
& push F(u,v,t) on stack
See: Corollary to Theorem
Stack Not Empty
No
THE END
Yes
Pop F(u,v,t) from Stack
No
If (F,duF,dvF) has Zero Inside Hull
See: Comments - proposition 1
Yes
Subdivide
& on stack
No
If Volume Small Enough
Yes
Classify / Generate Solution
solutions
See: Comments

Comments on the Algorithm

- **Calculation of** F (F is defined in corollary to the theorem 1)
  We calculate control points of F in Bézier form using simple control points arithmetic (e.g. proposition 1) to have F ready for subdivision.

- **Subdivision & Convex Hull Checking**
  The cost of the subdivision method is determined by the cost of splitting the volume by de Casteljau algorithm. Here we can do it only for the function F since the control points of the partial derivatives of F are easy to calculate using the control points of F. More exactly we will only calculate these points while checking the convex hull, according to the following fact:

  Proposition 1. [Derivatives] *If F is a Bézier volume of degree (m, n) based on points P_{ijk} then $\partial_u F(u, v, t)$ is a Bézier volume of degree $(m - 1, n)$ based on the points $R_{ijk} = m(P_{i+1,j,k} - P_{ijk})$ and $\partial_v F(u, v, t)$ is a Bézier volume of degree $(m, n - 1)$ based on the points $T_{ijk} = n(P_{i,j+1,k} - P_{ijk})$.*

  To make it less expensive we check the bounding box instead of convex hull. In our case we have to check if there are control points with different signs for the function F and for its two partial derivatives. So the problem is simplified to the subdivision method for one function F with modified convex hull checking.

- **Classification of Solutions** Using theorem 1 we obtain solutions for two offsets at distances $+d$ and $-d$. If we are interested in only one offset $S_d(u, v)$ then we have to classify all solutions we obtain. The solution (u, v, t) corresponds to $S_d(u, v)$ if the following holds:

$$Sign(d) = Sign(< N(u, v), R(t) - S(u, v) >)$$

Main Theorem

Theorem 1. *Let $S(u,v)$ denote the regular parametric surface, $R(t)$ is a curve, d is a positive number and the function $f(u,v,t)$ is defined by:*

$$f(u,v,t) := ||S(u,v) - R(t)||^2 - d^2 \tag{1}$$

Then the following :

$$R(t_0) = S_i(u_0, v_0) \ \ \textit{for} \ \ i = -d \ \ \textit{or} \ \ i = +d$$

is equivalent to the system of equations:

$$\begin{aligned} f(u_0, v_0, t_0) &= 0 \qquad (2) \\ \partial_u f(u_0, v_0, t_0) &= 0 \qquad (3) \\ \partial_v f(u_0, v_0, t_0) &= 0 \qquad (4) \end{aligned}$$

Lemma 1. *Let $C(t)$ denote a parametric curve such that $C'(t) \neq 0$ for all t and let R be a point. Then the nonzero vector $C(t_0) - R$ is perpendicular to the curve $C(t)$ at the point $C(t_0)$ iff the derivative of the square distance function $f(t)$ defined by $f(t) := ||C(t) - R||^2$ vanishes: $f'(t_0) = 0$.*

Proof. Using the following rule for calculating the derivative of the scalar product:

$$< A(t), B(t) >' = < A'(t), B(t) > + < A(t), B'(t) >$$

we obtain:

$$f'(t_0) = < C(t_0) - R, C(t_0) - R >' = 2 < C(t_0) - R, C'(t_0) >$$

□

Proof. If $R(t_0) = S_i(u_0, v_0)$ for $i = -d$ or $i = +d$ then the distance from the point $R(t_0)$ to the point on the surface $S(u_0, v_0)$ is equal to d and the equation (2) holds. From the definition of the offset we know that the vector $S(u_0, v_0) - R(t_0)$ is perpendicular to any curve on the surface at the point $S(u_0, v_0)$ and in particular for the curves $S(t, v_0)$ and $S(u_0, t)$. From the lemma applied to these curves we obtain the equations (3) and (4).

To prove reverse implication if the equations (3) and (4) hold then by the lemma the vector $S(u_0, v_0) - R(t_0)$ is orthogonal to the linearly independent vectors $\partial_u S(u_0, v_0)$ and $\partial_v S(u_0, v_0)$. This means that the point $R(t_0)$ lies on the line orthogonal to the surface at the point $S(u_0, v_0)$. From the equation (2) follows that the distance of the points $R(t_0)$ and $S(u_0, v_0)$ is equal to d. Therefore $R(t_0) = S_{-d}(u_0, v_0)$ or $R(t_0) = S_{+d}(u_0, v_0)$. $\square$

Corollary. *For a regular rational Bézier surface defined by:*

$$S(u, v) = \left(\frac{X(u, v)}{W(u, v)}, \frac{Y(u, v)}{W(u, v)}, \frac{Z(u, v)}{W(u, v)} \right)$$

where $X(u, v)$, $Y(u, v)$, $Z(u, v)$, $W(u, v)$ are polynomial functions such that $W(u, v) > 0$ and for $R(t) = (x(t), y(t), z(t))$ a curve in R^3, if d is a positive number and the function $F(u, v, t)$ is defined by:

$$F(u, v, t) := (X - xW)^2 + (Y - yW)^2 + (Z - zW))^2 - d^2W^2 \quad (5)$$

then the following :

$$R(t_0) = S_i(u_0, v_0) \quad \text{for} \quad i = -d \quad \text{or} \quad i = +d$$

is equivalent to the system of equations:

$$F(u_0, v_0, t_0) = 0, \quad \partial_u F(u_0, v_0, t_0) = 0, \quad \partial_v F(u_0, v_0, t_0) = 0$$

Complexity Analysis

Time for calulating F function can be neglected, solving the equation system is the most time consuming part of the algorithm. The number of steps in subdivision method is a function of desired accuracy ε and it belongs to the class $O(\log(1/\varepsilon))$.

The cost of one step is the cost of a subdivision of a Bézier surface or volume using de Casteljau algorithm. It can be represented as the number of elementary operations of midpoint calculations. The surface should be split into four subsurfaces and the volume should be split into eight subvolumes. This will guarantee that the parameter intervals will be twice smaller. De Casteljau algorithm for curves of degree n uses $n(n+1)/2$ midpoint calculations. The numbers of elementary operations are presented in table 1 for surfaces and volumes corresponding to:

- the algorithm presented here (volume of degree $(2n, 2n, 2)$),
- the one proposed in work [4] (two surfaces of degree $(5n-2, 5n-2)$),
- its straight extension for the rational case (two surfaces $(8n-2, 8n-2)$),

Table 1: Numbers of midpoint calculations.

Form of equation	Operations for $n =$	2	3	4
New Algorithm	$6n^2(6n+4)$	384	1188	2688
VP Polynomial	$(5n-2)^2(15n-3)$	1728	7098	18468
VP Rational	$(8n-2)^2(24n-3)$	8820	33396	83700

As it can be seen from the table for the Bézier surfaces of degrees: 2,3,4 the subdivision phase for the new algorithm is 4–7 times faster for polynomial surfaces and 22–31 times faster for rational ones. Comparisons are even better for the new method for surfaces of higher degree.

Experimental Results

The algorithm invented by Vafiadou and Patrikalakis ([4]) has been tested on a graphics workstation running at 25MHz. These tests results (in seconds) are recalled for the comparison purpose in Table 2. The tests of the new algorithm has been carried out on a slow 486DX2 66MHz processor based computer running the *Linux* operating system. The effeciency of numerical calculations of this machine is comparable to the efficiency of 25MHz Sun workstations.

Table 2: Performance of Vafiadou–Patrikalakis algorithm.

Surface description	Degree	Test results in seconds
Sweep of parabola	(1,2)	2.7, 1.5
Sweep of x^4	(1,4)	10.4, 6.7, 10.5

Table 3: Performance of new algorithm (times in seconds).

Surface description	Degree	d	Hit %	Average	Maximum
Sweep of parabola	(1,2)	0.5	31.3	0.03	0.16
Sweep of x^4	(1,4)	0.5	30.0	0.07	0.38
Rational Hill	(2,2)	0.2	25.0	0.06	0.82

The tests have been carried out by ray tracing images of offset surfaces. This way for one surface many rays have been generated and the run times of the algorithm finding all intersections have been obtained. Table 3 shows the average as well as the maximum run times in seconds (for accuracy set up to 0.0001). Since the algorithm runs faster when there is no intersection the information how many rays hit the offset is also presented there.

Rational Hills & its Offsets

First picture shows two surfaces of degree $(2,2)$: polynomial one (left) and rational one (right) with central weight stretched to 10 (other control points have weight 1). Second picture shows offsets of these surface with $d = 0.5$.

Rational Hills & its Offsets II

Offsets of rational hills with $d = -0.2$.

Closeup (with blue plane removed) :

Bibliography

References

[1] Hoffmann, C., A Dimensionality Paradigm for Surface Interrogations, *Computer Aided Geometric Design* **7** (1990) pp.517–532.

[2] Łukaszewski, A., Finding Ray–Offset Intersection For Rational Bézier Surfaces, Research Report of Institute of Computer Science, University of Wrocław, Nr 97/04 (`http://www.tcs.uni.wroc.pl/reports`).

[3] Rossignac, J.R., Requicha, A.A.G., Offsetting operations in solid modelling, *Computer Aided Geometric Design* **3** (1986) pp.129–148.

[4] Vafiadou, M.E., Patrikalakis, N.M., Interrogation of Offsets of Polynomial Surface Patches, *Eurographics '91 (Elsevier Science Publishers B.V.)* pp.247–259.

[5] Whitted, T., An Improved Illumination Model for Shaded Display, Comm. ACM, v.23, June '80, pp. 343–349.