POSTER: An Efficient Approach to Direct NURBS Surface Rendering for Ray Tracing

Aleksandrs Sisojevs
Riga Technical University, Department of images processing and computer graphics
Meza Str. 1/3 – 304
LV-1048, Riga, Latvia
alexiv@inbox.lv

Aleksandrs Glazs
Riga Technical University, Department of images processing and computer graphics
Meza Str. 1/3 – 338
LV-1048, Riga, Latvia
glaz@egle.cs.rtu.lv

ABSTRACT

Ray tracing is a popular method for generating high quality images. NURBS surface is a popular tool for 3D modeling. There are different approaches to NURBS surface visualization by ray tracing methods. But these methods of computer graphics do not solve this task at an expected level. Therefore in this paper the authors propose to use a new effective approach for NURBS surface visualization by ray tracing method to render objects with high accuracy and quality.

Keywords

NURBS, rendering, direct ray tracing.

1. INTRODUCTION

Most of the modern ray tracing based applications only deals with triangles as basic primitives. NURBS surface representation is common for most of 3D modeling tools because of its compactness and the useful geometric properties of NURBS surfaces. Using the direct raytracing of NURBS surfaces, one can achieve better quality of rendered images [Efr05a].

Martin et al. [Mar00a] describes a framework for ray-tracing of trimmed NURBS using Newton's method. A nonlinear equation system is used for finding the intersection point. The Newton's method is used for solving this equation system.

Nishita et al. [Nis90a] describes an iterative algorithm called Bézier Clipping, used to compute intersections between a ray and a Bézier patch by identifying and cutting away regions of the patch that are known not to intersect with the ray. There are several disadvantages in this approach, like numerical unstable work for some surfaces and problems in solving the case of multiple intersection points.

Efremov et al. [Efr05a] proposed some modification to Nishita's algorithm for Bézier and NURBS surfaces. But in this case, the number of patches and calculations increases.

Schollmeyer and Froehlich [Sch09a] describe an approach for NURBS surface ray tracing, where surface trimming is used to set of monotonic Bézier curves. For finding of intersection point the bisection method is used. But in this case, the number of calculations increases too.

This paper presents an effective approach for finding ray – NURBS surface intersection points, which are used for high-quality visualization of NURBS surfaces.

2. PROPOSED APPROACH

The mathematical task of finding an intersection point between the ray and a parametric surface can be described as a nonlinear equations system [Mar00a, Sis08a]:

$$\begin{cases} S'_X(u,v) = C_X(t) \\ S'_Y(u,v) = C_Y(t) \\ S'_Z(u,v) = C_Z(t) \end{cases}, \quad (1)$$

where: S'_X, S'_Y, S'_Z – surface equations,

$C_X(t)$, $C_Y(t)$, $C_Z(t)$ – ray equations.

A NURBS surface patch in Cartesian 3D space can be formulated as [Rog90b, Hea04b]:

$$S'(u,v)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}P'_{i,j}\cdot w_{i,j}\cdot N_{i,p}(u)\cdot N_{j,q}(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m}w_{i,j}\cdot N_{i,p}(u)\cdot N_{j,q}(v)}, \quad (2)$$

where: $P'_{i,j}$ – control points; $w_{i,j}$ – weights; $N_{i,p}(u)$, $N_{j,q}(v)$ – the B-spline polynomials; p, q - the B-spline polynomials degree; u, v – parameters.

Projection to R^2

Typically calculation is performed in R^4 for rational patches [Nis90a]. Woodward [Woo89a] shows how the problem can be projected to R^2. This means that the number of calculations is reduced by 25%. This approach is used in [Nis90a] for rational Bezier patch subdivision. But this approach is good for NURBS surface calculation too. In this case the task of ray-surface intersection point search is transformed to the problem of non-linear equations system solving:

$$\begin{cases} S_X(u,v)-x_R=0 \\ S_Y(u,v)-y_R=0 \end{cases}, \quad (3)$$

where: $S_X(u,v)$, $S_Y(u,v)$ – are the surface equations on the projection plane, x_R, y_R – is the ray projection.

The system (3) solving task is divided into two parts: preprocessing and iterative root finding.

Preprocessing

Preliminary searches for NURBS patch surface is equivalent to procedure for B-spline patch surfaces what is described in work [Sis09a]. In order to obtain the preliminary values of parameters u and v in every pixel, we propose to create a map of preliminary values. The map consists of the patch data and is coded in RGB channels. The Red channel contains the number of the patch. The Green and Blue channels contain the uniform gradient texture fill of the patches. The map is coded using the OpenGL graphics library. In this case the preliminary values of parameters can be described as follows:

$$\begin{cases} u_0=(u_{\max}-u_{\min})\cdot\frac{g}{255}+u_{\min} \\ v_0=(v_{\max}-v_{\min})\cdot\frac{b}{255}+v_{\min} \end{cases}, \quad (4)$$

where: u_0, v_0 – parameters preliminary value in pixel; $u_{\min}$, $u_{\max}$, $v_{\min}$, $v_{\max}$ – parameters u and v minimum and maximum; g, b – colour value in green and blue channels.

Intersection Test

The Newton iteration [Tah03a] can be used for solving system (3) solving. In this case, iteration steps take the form:

$$\begin{bmatrix} u_{iter+1} \\ v_{iter+1} \end{bmatrix}=\begin{bmatrix} u_{iter} \\ v_{iter} \end{bmatrix}+\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}, \quad (5)$$

where:

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}=-[J]^{-1}\cdot[F], \quad (6)$$

where:

$$[J]=\begin{bmatrix} \frac{\partial S_X}{\partial u} & \frac{\partial S_X}{\partial v} \\ \frac{\partial S_Y}{\partial u} & \frac{\partial S_Y}{\partial v} \end{bmatrix};\ [F]=\begin{bmatrix} S_X-x_R \\ S_Y-y_R \end{bmatrix}, \quad (7)$$

$[J]$ is Jacobian matrix.

Proposed method

The NURBS patch surface equation on the projection plane can be described as follows:

$$S(u,v)=\left(\frac{x}{w}\quad\frac{y}{w}\right), \quad (8)$$

where: x, y, and w – are numerator and denominator of NURBS surface equations on the projection plane.

Using NURBS surface partial derivatives [Rog90b], after transformation, the equation (6) takes the form:

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}=\frac{w}{M}\cdot\begin{bmatrix} M_U \\ M_V \end{bmatrix}, \quad (9)$$

where:

$$M=\begin{vmatrix} x & \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ y & \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ w & \frac{\partial w}{\partial u} & \frac{\partial w}{\partial v} \end{vmatrix};\ M_U=\begin{vmatrix} x & x_R & \frac{\partial x}{\partial v} \\ y & y_R & \frac{\partial y}{\partial v} \\ w & 1 & \frac{\partial w}{\partial v} \end{vmatrix}, \quad (10)$$

and

$$M_V=\begin{vmatrix} x & \frac{\partial x}{\partial u} & x_R \\ y & \frac{\partial y}{\partial u} & y_R \\ w & \frac{\partial w}{\partial u} & 1 \end{vmatrix}, \quad (11)$$

In contradistinction to method Martin et al. [Mar00a], where both the rational and non-rational equations partials derivatives were calculated, in the proposed method only the non-rational equations partials derivatives was calculate. This approach allows to decrease number of calculations and speed-up rendering time.

Termination Criteria

In this work, three criteria are used to decide when to terminate the Newton iteration. There are drawn [Yang87a]. The first condition is success criterion: if we are closer to the root than some predetermined ε_1:

$$\left\| \left[F\left(u_{iter}, v_{iter}\right) \right] \right\|_2 < \varepsilon_1, \quad (12)$$

In this case, we report an intersection to. Otherwise, we continue the iteration. There is no allowing the new (u_{iter}, v_{iter}) estimate to take us farther from the root than the previous one:

$$\left\| \left[F\left(u_{iter}, v_{iter}\right) \right] \right\|_2 > \left\| \left[F\left(u_{iter-1}, v_{iter-1}\right) \right] \right\|_2, \quad (13)$$

In this work check is made to assure that the matrix [*M*] is not singular:

$$\left| \det\left(M\right) \right| < \varepsilon_2, \quad (14)$$

3. EXPERIMENTAL RESULTS

In this work the proposed method, as well as the methods suggested by Martin et al. were implemented. In order to visualize a scene the 1 ray/pixel approach was used. The size of the images in experiment is 512*512 pixels. The experiments were carried out on a computer with CPU Intel Xeon 3,2 GHz, RAM 2 GB. The received images are shown in Figures 1-6. Image rendering time is shown in Table 1. For comparison we shall consider the time of visualization in percentage, by taking earlier known method (Martin et al.) for 100%.

Objects	Proposed method, sec.	Martin et al. Method, sec	Speed-up of rendering, %
"Mobile phone"	4,297	5,422	20,75
"Machine component"	1,890	2,203	14,21
"27 Ducks"	9,0	11,86	24,11

Table 1. Image rendering time

4. CONCLUSION

In this work an efficient approach to direct NURBS surface rendering for ray tracing is proposed. The results (in Figures 1 – 6) shows, that:

- As it is possible to see from these figures the proposed method gives an advantage on quality of the images (there's no distortion on the borders of patches).
- As seen from the table, the proposed method gives stable results in the fastest rendering time in our experiments (speed-up is 14% – 24% than the method of Martin et al.).

5. ACKNOWLEDGMENTS

This work has been supported by the European Social Fund within the project «Support for the implementation of doctoral studies at Riga Technical University».

6. REFERENCES

[Efr05a] Efremov, A., Havran, V. & Seidel, H.-P. 'Robust and numerically stable Bézier clipping method for ray tracing NURBS surfaces.' In Proceedings of the 21st Spring Conference on Computer Graphics SCCG'05:127 - 135. 2005

[Hea04b] Hearn, D. and Baker, M.P. Computer Graphics with OpenGL, Prentice Hall, 2004.

[Him72b] Himmelblau, D. Applied Nonlinear Programming, – McGraw-Hill Book, 1972.

[Mar00a] Martin, W., Cohen, E., Fish, R., & Shirley, P. (2000). 'Practical Ray Tracing of Trimmed NURBS Surfaces.' JGT 5(1): 27-52. 2000.

[Nis90a] Nishita. T., Sederberg. T.W., & Kakimoto M. Ray tracing trimmed rational surface patches, Computer Graphics, vol.24, no. 4, pp. 337–345, 1990.

[Rog90b] Rogers, D.F. and Adams, J.A. Mathematical Elements for Computer Graphic, 2nd Ed., McGraw-Hill, Boston, MA, 1990.

[Sch09a] Schollmeyer A., Froehlich, B. Direct Trimming of NURBS Surfaces on the GPU, ACM Transactions on Graphics, Volume 28, Issue 3, Article No. 47, 2009.

[Sis08a] Sisojevs A., Glazs A An new approach of visualization of free-form surfaces by a ray tracing method, IEEE MELECON Proceedings, 2008, pp 872-875

[Sis09a] Sisojevs A., Glazs A. POSTER: Efficient approach to direct B–spline surface rendering by a ray tracing, The International Conference WSCG'2009 Poster proceedings, 2009. - 13- 16 p.

[Tah03b] Taha H.A. Operations Research: an Introduction, 7th Ed. – New Jersey: Prentice Hall, 2003. – 830 p.

[Woo89a] Woodward C. Ray tracing parametric surfaces by subdivision in viewing plane// Theory and practice of geometric modeling book contents, Springer-Verlag New York, Inc., New York, NY, USA. – 1989. pp. 273 – 287.

[Yan87a] Yang C.-G. On speeding up ray tracing of B-spline surfaces. Computer Aided Design, 19(3), April 1987.

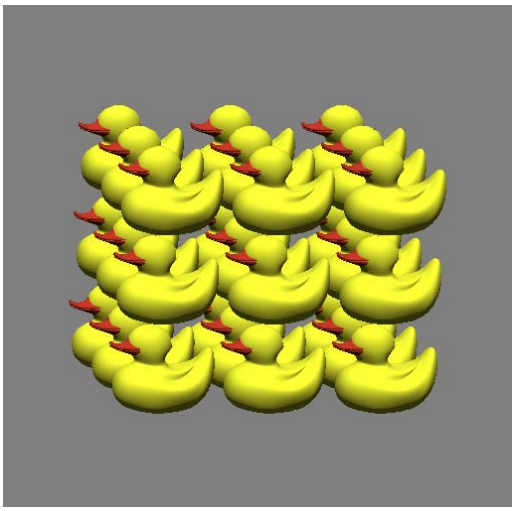

Figure 1. The image of a scene "27 Ducks" obtained using the proposed method.

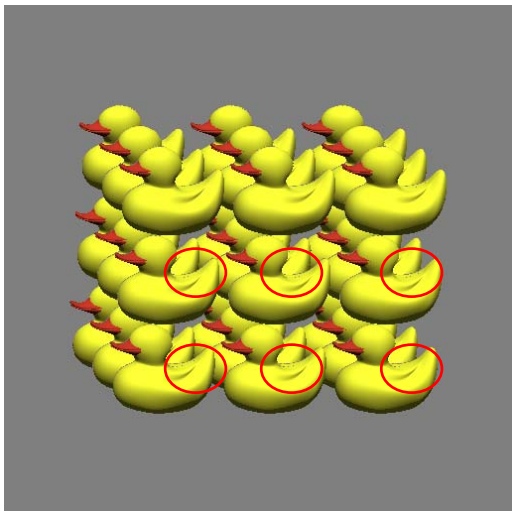

Figure 2. The image of a scene "27 Ducks" obtained using the method Martin et al.

Figure 3. The image of an object "Mobile phone" obtained using the proposed method.

Figure 4. The image of an object "Mobile phone" obtained using the method Martin et al.

Figure 5. The image of an object "Machine component" obtained using the proposed method.

Figure 6. The image of an object "Machine component" obtained using the method Martin et al.