8th International Conference
APLIMAT 2008
Faculty of Mechanical Engineering - Slovak University of Technology in Bratislava
Section:

VISUALIZATION OF TRIVARIATE NURBS VOLUMES

SAMUELČÍK Martin (SK)

Abstract. In this paper we focus on one particular set of free-form objects and its visualization. We extended approach for modeling curves and surfaces and prepared trivariate volumes based on Bezier and B-spline expressions. Our main goal is to visualize given parametric volumes. The visualization is done by approximation of volumes by net of isoparametric curves or surfaces and by boundary evaluation of volumes. Rendering is driven by OpenGL platform and there are used several techniques for enhanced visual effects like transparency. We also present practical output based on our implementation of proposed solutions.

Key words. Geometric modelling, NURBS volumes, rendering, OpenGL

1 Introduction

Free-form curves and surfaces are well known and described part of mathematical background in the field of Computer-Aided Geometric Design [2], [3]. In many cases, the parametric expression with polynomial functions is used. For better representation of objects like sphere or cone, natural extension in the form of rational function can be applied [3]. There are also many works on visualization of these free form objects. One uses approximation of object by polylines or triangular meshes. Different visualization option is raytracing, where most important part is to find intersection of line and given objects. To find these of intersections, we can use several algorithms like root finding or subdivision (Bezier clipping).
In our research, we are trying to extend polynomial curves and surfaces into trivariate polynomial objects (volumes) based on Bezier expression [5]. We can also prepare extension in the form of rational volumes and rational splines. For splines, we will use well-known spline representation, B-splines. These volumes can be used in many parts of geometric design, for example as grid for three-dimensional deformations, for scattered data interpolation or for representation of solid objects [7]. In design, it is very important to visualize objects. We will propose some basic options how to show these objects [6]. For rendering 3D objects on 2D screen, we will use Open GL graphics library, because it is supported on many platforms and in many hardware solutions [4].

2 Trivariate volumes

General trivariate object is defined by trivariate function F and the domain D for that function. Precisely:

$$
\begin{aligned}
& TO : R^3 -> E^3 \\
& TO(u,v,w) = F(u,v,w) \\
& (u,v,w) \in D
\end{aligned}
$$

In the field of geometric modeling, polynomial or piecewise polynomial (spline) function is most commonly used. We will work with only this representation defined by net of control points.

The rational Bézier tetrahedral volume is defined with a degree, domain, control net of points, and for each point one real number (weight). The degree is a positive integer number n and means degree of polynomials used to blend given points, The domain is a nondegenerated tetrahedron $ABCD$ in E^3 which is used for addressing of barycentric coordinates and control net with weights is tetrahedral structure of points in E^3, that can be written following way:

$$
\begin{aligned}
& V_{\mathbf{i}} \in E^3; w_{\mathbf{i}} \in R \\
& \mathbf{i} = (i,j,k,l); |\mathbf{i}| = i + j + k + l = n; \\
& i.j,k,l \geq 0
\end{aligned}
$$

Let us have point U from the domain and let $\mathbf{u}=(u,v,w,t);u+v+w+t=1$ are barycentric coordinates of point U ($U=uA+vB+wC+tD$) with respect to $ABCD$. Now we can define point of rational Bézier tetrahedral volume $RB^n(\mathbf{u})$ with recursive de Casteljau algorithm:

$$
\begin{aligned}
& V_{\mathbf{i}}^0(\mathbf{u}) = V_{\mathbf{i}}; w_{\mathbf{i}}^0(\mathbf{u}) = w_{\mathbf{i}}; \\
& w_{\mathbf{i}}^r(\mathbf{u}) = uw_{\mathbf{i}-\mathbf{e}_1}^{r-1}(\mathbf{u}) + vw_{\mathbf{i}-\mathbf{e}_2}^{r-1}(\mathbf{u}) + ww_{\mathbf{i}-\mathbf{e}_3}^{r-1}(\mathbf{u}) + tw_{\mathbf{i}-\mathbf{e}_4}^{r-1}(\mathbf{u});
\end{aligned}
$$

$$
\begin{aligned}
& w_{\mathbf{i}}^r(\mathbf{u})V_{\mathbf{i}}^r(\mathbf{u}) = uw_{\mathbf{i}-\mathbf{e}_1}^{r-1}(\mathbf{u})V_{\mathbf{i}-\mathbf{e}_1}^{r-1}(\mathbf{u}) + vw_{\mathbf{i}-\mathbf{e}_2}^{r-1}(\mathbf{u})V_{\mathbf{i}-\mathbf{e}_2}^{r-1}(\mathbf{u}) + ww_{\mathbf{i}-\mathbf{e}_3}^{r-1}(\mathbf{u})V_{\mathbf{i}-\mathbf{e}_3}^{r-1}(\mathbf{u}) + +tw_{\mathbf{i}-\mathbf{e}_4}^{r-1}(\mathbf{u})V_{\mathbf{i}-\mathbf{e}_4}^{r-1}(\mathbf{u}); \\
& RB^n(\mathbf{u}) = V_{\mathbf{0}}^n(\mathbf{u})
\end{aligned}
$$

where $r=1,...,n$; $|\mathbf{i}|=n-r$ and $e_1=(1,0,0,0)$; $e_2=(0,1,0,0)$; $e_3=(0,0,1,0)$; $e_4=(0,0,0,1)$. From this definition the analytical expression of rational Bézier tetrahedra can be evaluated. So, for barycentric coordinates $\mathbf{u}$ of any point U from the domain we have:

$$
RB^n(\mathbf{u}) = \frac{\sum_{|\mathbf{i}|=n} w_{\mathbf{i}} V_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u})}{\sum_{|\mathbf{i}|=n} w_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u})}
$$

where

$$
B_{\mathbf{i}}^n(\mathbf{u}) = \frac{n!}{i!\,j!\,k!\,l!} u^i v^j w^k t^l
$$

are generalized Bernstein polynomials.

Rational Bézier tensor volume is defined with three degrees *n, m, o* and a control net of points and for each point a real number (weight). The degrees are positive integers and represent the degree of blending polynomials in each direction, the domain is nondegenerated box *ABCDEFGH* in E^3 and the control net with weights is box structure of points in E^3 that can be written following way:

$$V_{\mathbf{i}} \in E^3; w_{\mathbf{i}} \in R$$
$$\mathbf{i} = (i, j, k); n \geq i \geq 0; m \geq j \geq 0; o \geq k \geq 0$$

Assume that we have point *U* from domain and let **u** are coordinates of U with respect to *ABCDEFGH*, so $\mathbf{u}=(u,v,w);0\leq u,v,w\leq 1$. Now we can define point of Bézier tensor volume $RB^{n,m,o}(\mathbf{u})$ with analytical expression:

$$RB^{n,m,o}(\mathbf{u}) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{o} w_{(i,j,k)} V_{(i,j,k)} B_i^n(u) B_j^m(v) B_k^o(w)}{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{o} w_{(i,j,k)} B_i^n(u) B_j^m(v) B_k^o(w)}$$

where

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

are simple Bernstein polynomials. For this type of volume also exists de Casteljau algorithm, but it is less generalizable.

Non-uniform rational B-spline (NURBS) volumes are a natural extension of NURBS curves and surfaces used in geometric modeling. Parameters that define NURBS volume are similar to the surface case, but there are new parameters due to new added parameter (direction). NURBS volume is defined with :

- three degrees d_u, d_v, d_w,
- three non-decreasing knot vectors $(u_0, u_1, \ldots u_{mu})$, $(v_0, v_1, \ldots, v_{mv})$, $(w_0, w_1, \ldots, w_{mw})$,
- three-dimensional net of control points $V_{i,j,k}$ in E^3; $0<=i<=n_u$; $0<=j<=n_v$; $0<=k<=n_w$;
- for each control point $V_{i,j,k}$ real number (weight) $p_{i,j,k}$
- domain $<u_{du},u_{nu}+1)$x$< v_{dv},v_{nv}+1)$x$< w_{dw},w_{nw}+1)$
- $m_u=n_u+d_u+1$, $m_v=n_v+d_v+1$, $m_w=n_w+d_w+1$

Then NURBS volume is given analytically as

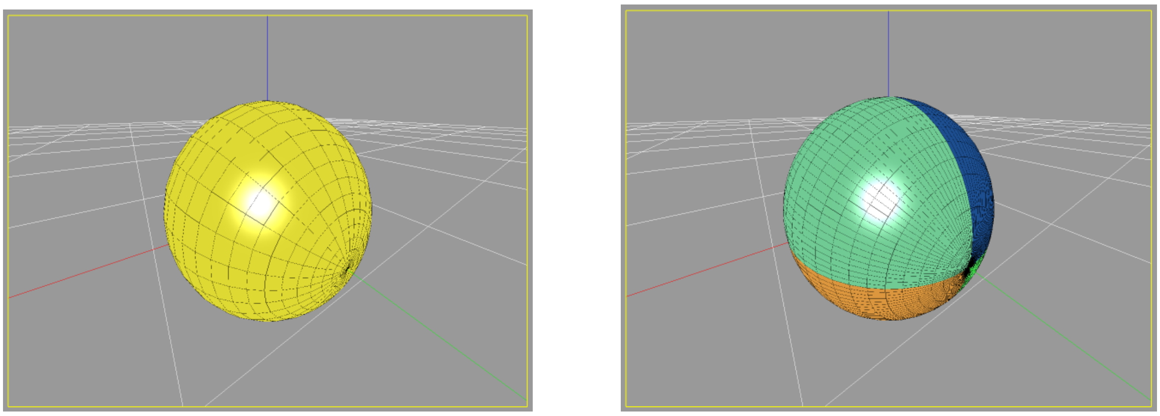
$$S(u,v,w) = \frac{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \sum_{k=0}^{n_w} p_{i,j,k} V_{i,j,k} N_i^{d_u}(u) N_j^{d_v}(v) N_k^{d_w}(w)}{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \sum_{k=0}^{n_w} p_{i,j,k} N_i^{d_u}(u) N_j^{d_v}(v) N_k^{d_w}(w)}$$

where $N_i^{d_u}(u), N_j^{d_v}(v), N_k^{d_w}(w)$ are B-spline blending functions defined on particular knot vectors with given degrees. Parameters u,v,w are from domain, $u \in <u_{d_u}, u_{n_u+1}), v \in <v_{d_v}, v_{n_v+1}), w \in <w_{d_w}, w_{n_w+1})$.

3 Visualization

In this section, we will describe basic solutions for visualization of trivariate free-form objects. Because we are dealing with parametric description of objects, many algorithms can be used in polynomial, rational or spline cases. There also exists simple algorithm for converting NURBS volume into set of Bezier volumes. Conversion algorithm is based on well-known algorithm of knot insertion for NURBS objects. This process is illustrated in Figure 1b.

Because trivariate volumes represent also interior of the object, we have to create some insight into interior. This leads to some approximations or loss of visualization on the boundary. In some algorithms, only boundary of volume will be determined.



a) b)

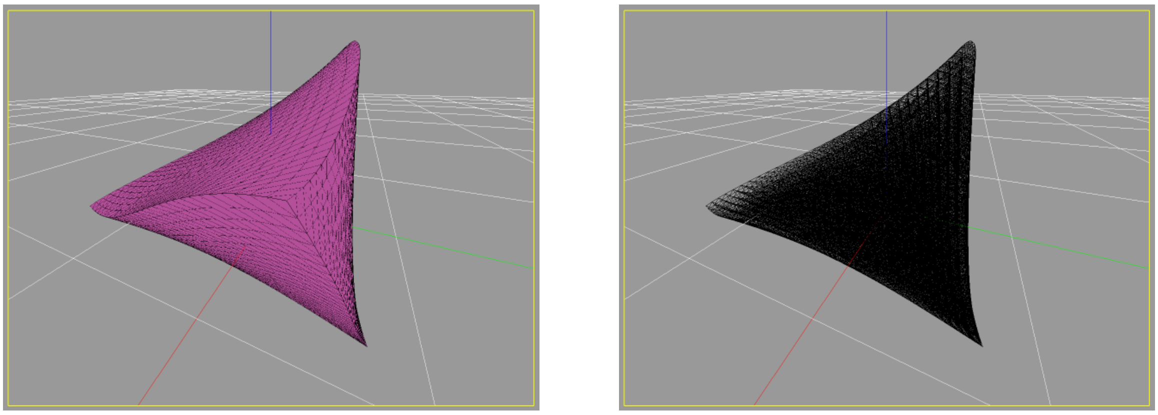
Figure 1: Sphere displayed with isoparametric curves and surfaces: a) Simple NURBS volume, b) Volume decomposed to four Bezier volumes.

3.1 Isoparametric curves

Isoparametric curves are curves generated from parametric function that defines parametric volume. This curve is univariate free-form object derived from trivariate function by making two of three parameters as constants. For example in the case of Bezier tensor volume, we have following set of curves in *u* direction:

$$\{C_{v,w}(u), v,w \in <0,1>; C_{v,w}(u) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{o} w_{(i,j,k)} V_{(i,j,k)} B_i^n(u) B_j^m(v) B_k^o(w)}{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{o} w_{(i,j,k)} B_i^n(u) B_j^m(v) B_k^o(w)}\}$$

This set contains infinite number of curves, so we have to pick some finite number of sample curves. The basic choices are curves when we uniformly sample interval <0,1> for values v and w. This process of generating isoparametric curves can be repeated in v and w direction. Figure 2b shows result of visualization of several curves for all directions in the case of Bezier tetrahedral volume.



a) b)

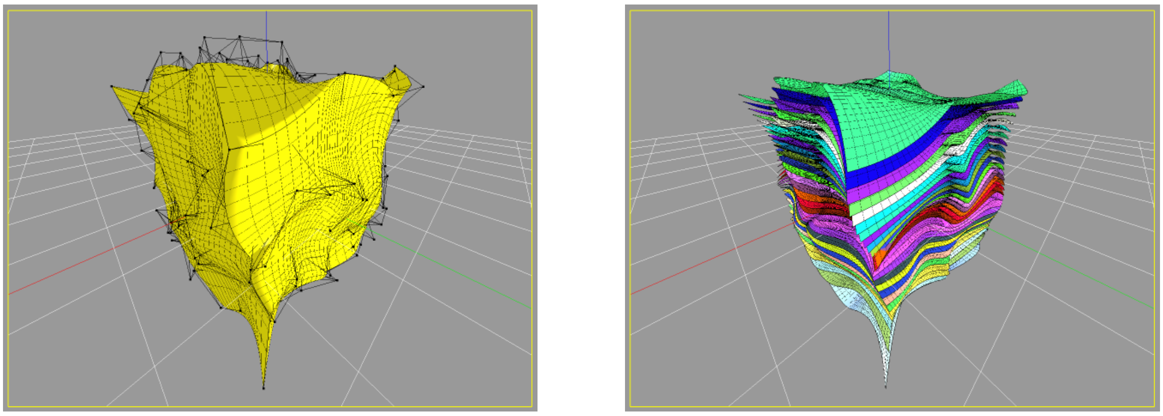
Figure 2: a) Bezier tetrahedra, b) Bezier tetrahedra with displayed isoparametrical curves

3.2 Isoparametric surfaces

Just like the case of isoparametric curves, isoparametric surfaces are generated from volume parametric expression by putting one of the parameters as constant. Because this constant parameter is from continuous interval, we will get infinite number of surfaces. We need to choose some finite number of representative surfaces, this is done by simple uniform sampling of a definition interval. Precisely for rational Bezier tensor volume we will choose and visualize following surfaces for *u* direction (for *v*, *w* direction, surfaces can be determined similarly):

$$\{C_u(v,w), u \in <0,1>; C_u(v,w) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{o} w_{(i,j,k)} V_{(i,j,k)} B_i^n(u) B_j^m(v) B_k^o(w)}{\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{o} w_{(i,j,k)} B_i^n(u) B_j^m(v) B_k^o(w)}\}$$

For whole volume rendering it is not necessary to render isoparametric surfaces in all directions, it is often sufficient to choose only one direction.



a) b)

Figure 3: a) NURBS volume with random control points, b) NURBS volume displayed using isoparametric surafces

3.3 Boundary evaluation

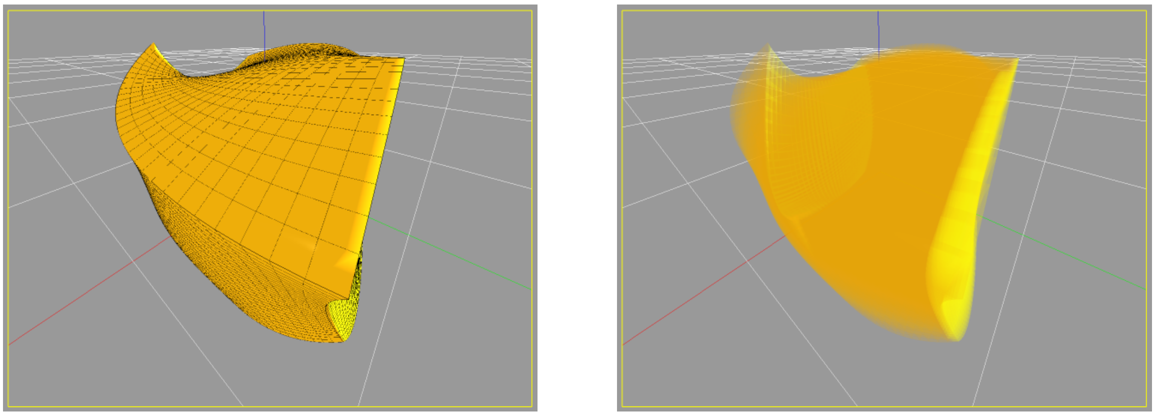
In many applications, visualization of volume interior is not necessary. The boundary of volume is sufficient visualization element. The boundary of trivariate object consist of boundary isoparametric curves (for extremal parameters) together with part of solids where Jacobi determinant is zero:

$$J(U) = \begin{vmatrix} \frac{\partial RB}{\partial u}(U) & \frac{\partial RB}{\partial u}(U) & \frac{\partial RB}{\partial u}(U) \\ \frac{\partial RB}{\partial v}(U) & \frac{\partial RB}{\partial v}(U) & \frac{\partial RB}{\partial v}(U) \\ \frac{\partial RB}{\partial w}(U) & \frac{\partial RB}{\partial w}(U) & \frac{\partial RB}{\partial w}(U) \end{vmatrix} = 0$$

When combined these two options, we can render whole boundary of trivariate object, as in Figures 1, 3a.

3.4 Transparency

It is possible to render all isoparametric surfaces for three sampled parameters presented in section 3.2. In that case, only approximated boundary of whole volume will be visible (Figure X). For insight into interior of object, we can use transparency of generated isoparametric surfaces. Rendering of transparency is done using basic features of OpenGL rendering engine. Figure 4 shows NURBS volume rendered using enabled transparency.



a) b)

Figure 4: a) NURBS volume with random control points, b) NURBS volume displayed using isoparametric surafces

4 Results & Future Work

We have implemented all proposed visualization options of trivariate objects. Our own system called GeomForge can model and visualize volumes in easy and efficient way. All figures presented in this paper were renderer using this system.

We have tested frame rates of visualization on AMD Sempron system with 1.61GHz and 1GB size of memory. Used graphics card was ATI Radeon X700 with 128MB of graphics memory. Table 1 shows these rates for various objects and settings.

In the future, we want to focus on raytracing methods using approximation methods and Bezier clipping algorithm. We have also some unsolved features like exact sorting of geometry for proper transparency of NURBS volumes.

Model	Isoparametric curves	Isoparametric surfaces	Transparency
Sphere	59 FPS	161 FPS	51 FPS
Torus	60 FPS	140 FPS	51 FPS
Random	23 FPS	80 FPS	19 FPS

Table 1: Frame rates of visualization options for several NURBS volumes.

Acknowledgement
The paper was supported by grant from VEGA Grant Agency under contract no. 1/3083/06 with title “Complexity of Geometric Algorithms for Realtime Rendering in Virtual Reality”.

References

[1] Joy K., Duchaineau, M., Boundary determination for trivariate solids, Proc. of Pacific Graphics '99, pp 82-91, Oct 1999.
[2] Farin G. Curves and Surfaces for Computer-Aided Geometric Design - A Practical Guide. ISBN 0-12-249052-6, Academic Press, Boston, 1993.
[3] Piegl L., Tiller W. The NURBS Book. ISBN 3-540-61545-8, Springer-Verlag, New York, 1997.

[4] Various. OpenGL homepage, http://www.opengl.org.
[5] Dieter Lasser. Bernstein - Bézier representation of volumes. Computer Aided Geometric Design, 2(1-3):145-149, 1985.
[6] Dieter Lasser. Visualization of free-form volumes. In IEEE Visualization, pages 379-387, 1990.
[7] Ron MacCracken and Kenneth I. Joy. Free-form deformations with lattices of arbitrary topology. In SIGGRAPH '96: Proceedings of the 23-rd annual conference on Computer graphics and interactive techniques, pages 181-188, New York, NY, USA, 1996. ACM Press.

RNDr. Martin Samuelčík
Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynska dolina, Bratislava, 842 48
e-mail: samuelcik@fmph.uniba.sk