Department of Algebra, Geometry and Didactics of Mathematics
Faculty of Mathemathics, Physics and Informatics
Comenius University, Bratislava

Bézier and B-spline volumes

Project of Dissertation

Martin Samuelčík

Bratislava, 2005

Contents

List of Figures

Abstract

This thesis presents basic definitions and properties of parametric curves, surfaces and volumes. We focus on the spline objects in Bézier and B-spline form. The project describes our aims in extending approaches from curves and surfaces to volumes. We work also with subdivision curves, surfaces and volumes that create bridge between continuous and discrete representations.

Abstrakt

V tejto práci predkladáme základné definície a vlastnosti parametrických kriviek, plôch a telies. Zameriavame sa na splajnové objekty v Bézierovej a B-splajnovej forme. Projekt opisuje naše ciele, kde sa pokúsime rozšíriť prístupy známe z kriviek a plôch na telesá. Opíšeme aj prerozdelovacie krivky, plochy a telesá, ktoré tvoria spojivo medzi spojitou a diskrétnou reprezentáciou.

Introduction

This work presents basics of parametric objects in geometric modelling. The main goal is to use known properties about parametric curves and surfaces to construct and describe parametric volumes based on Bézier and B-spline construction. To this description belong:

- *definitions and properties of such volumes*
- *volumes visualization*
- *modelling techniques*
- *subdivision volumes based on parametric volumes*

To achieve this goal, first we show how this modelling parts work for parametric curves and surfaces. Most parts about parametric volumes will be covered in final dissertation work.

The first chapter gives an overview of the most used 3D object representations that are used in computer graphics and geometric modelling. Presented are basic definitions, advantages and disadvantages.

The next chapter describes curves as basic objects in geometric modelling. The curves are used in many areas, such as paths description, contours representation and are basic elements for constructing surfaces and volumes. Presented are curves based on Bézier and B-spline form, its properties, properties of blending functions and also subdivision techniques for curves.

The third chapter presents parametric surfaces. Used are constructions for the parametric curves to make surfaces based on bivariate Bézier and B-spline functions. Described are Bézier triangles, Bézier and B-spline tensor product surfaces as well as its different modelling and visualization techniques. End of chapter belongs to the subdivision algorithms for surfaces.

In the fourth chapter, the knowledge from the previous chapter is used to construct Bézier, B-spline and subdivision volumes. Only basic terms are given, the whole theme will be covered in the dissertation work. All topics that will be in final dissertation work are described in the last chapter.

All definitions and computations will be held in three dimensional euclidean space, marked as E^3.

Chapter 1

Object representations

The basic element in 3D computer graphics and geometry is a three-dimensional object. In computer graphics and geometric modelling, there exist many ways to represent three-dimensional objects. Because of diversity and number of these representations, each one has its own advantages and disadvantages. The areas that we consider when comparing different representations are:

- *Data effectiveness for modelling.*
- *Occupied space in memory.*
- *Visualization issues using different techniques.*
- *Possibility to represent arbitrary objects (e.g. conics).*
- *Possibility of conversion to another representation.*

In this work, we will try to prepare framework for parametric representation of objects based on observations from the approaches of lower degree. So first we will give introduction to most used types of parametric curves and surfaces and then its extensions for parametric 3D objects. From now on, we will call objects represented this way as parametric solids or volumes. Some volumetric representations are presented in [3]. An overall introduction to object representations is given in [13].

1.1 Boundary representation (B-Rep)

Boundary representation describes object using just its boundary. This approach has its own advantages and disadvantages. Because only border is given, we don't have information on inside the object, on the other hand, size of the data storage is lower, computation times are faster and visualization is simpler with the help of graphics hardware. Border can be described using a few approaches:

- *Wireframe* - model is represented only as boundary edges. It takes only small amount of memory and visualization of object in wireframe is very fast and can be used for the preview. But we don't have information about faces of the object.

- *Polygonal representation* - The border of object is represented as the set of polygons. There are many data structures for storing objects in this representation. Used structures are winged edge, half edge, quad edge, indexed polygons. This representation is natively supported in graphics hardware, so its visualization is fast.

- *Parametric surfaces* - Border is described by bivariate function. Many forms of these functions are used. Later in this work, two usable forms are presented. Using this approach, complex borders can be described using small amount of data (control points), but for visualization, surface must be approximated using set of polygons. Here, fast work on surface can be performed, but relation of point in space and surface is hard to find.

- *Subdivision surfaces* - This is bridge between polygonal and parametric representation. Initial control mesh is refined repeatedly to get finer and smoother representation. Also this will be presented later.

- *Implicit surfaces* - Boundary surface is represented by function $F : R^3 \rightarrow R$ and object is a set $O = \{(x, y, z) \in E^3; F(x, y, z) = 0\}$. This representation is good for well known Boolean operations (union, intersection and difference), its easy to find if point in space is on surface. Visualization is again performed using approximation by set of polygons.

1.2 Functional representation (F-Rep)

This approach is similar to implicit surfaces, but instead of equality, inequality is used. For given function $F : R^3 \rightarrow R$, the object is the set $O = \{(x, y, z) \in E^3; F(x, y, z) \leq 0\}$. Very good advantage of this approach for modelling purposes is that the Boolean operations are performed very fast and easy. Also a very complex object can be described using one function.

1.3 Constructive solid geometry (CSG)

In this representation, object is constructed from basic objects using defined set of operations such as Boolean operations. Object is then represented as a graph or a tree with basic objects (box, cone, sphere) in leaves of tree and with inner nodes as operations (union, intersection, difference). Many operations and modelling techniques are performed by traversing this construction tree.

1.4 Discrete volume representation

Object is represented as a finite set of volume elements called voxels, These voxels are usually boxes or tetrahedrons. Can be arranged in uniform grid, in tree structure (octree) or in non-uniform grid glued together with neighborhood information in each voxel. Also here Boolean operations are performed easily. Due to discrete representation of the object, many artifacts (like alias) can arise.

1.5 Parametric representation

An object is described as a set of points represented by a trivariate function. More about this representation will be presented later. This approach uses more storage space for parameters (control points, weights, knots), also approximation by discretization uses more memory and computational time, but modern hardware can handle this. Here we have many degrees of freedom, that can be lowered using additional constraints. Representation is good for deformations and other modelling parts. Also there exists bridge between parametric volumes and discrete representation called subdivision volumes.

Chapter 2

Parametric curves

Curves are objects that are widely used in computer graphics and geometric modelling. They are used, for example, for defining shape of characters in fonts, for describing paths in various types of simulations, as a basic element for defining, describing, building and modelling surfaces and volumes. As in the case of three dimensional objects, we know many types of representation for curves. But for our purposes, we will focus on one of the most used representation of curves, parametric representation.
Parametric curves can be treated as motion of point during some time interval. So in definition of such curve, curve is set of position based on continuing time variable.

Definition: Parametric curve in E^3 is a set of points $C = \{X \in E^3; X = f(u); u \in \langle a, b\rangle\}$, where $f : R \rightarrow E^3$ is function describing curve, u is (time) parameter and $\langle a, b\rangle$ is curve domain.

It is obvious, that shape and properties of the curve are related to function f and to interval $\langle a, b\rangle$. But also different functions can lead to same set of points. One example of such different functions is $f_1 : u \rightarrow (t, t, t)$ and $f_2 : u \rightarrow (t^2, t^2, t^2)$, both on interval $\langle 0, 1\rangle$. These two representations of the same set of points have also some different properties.
Parametrically defined curve is described using one function. But for modelling purposes, many functions don't have properties that we need. So, only some classes of functions are used. We will use only polygonal, rational functions and its piecewise version, such curves are then called splines. Also form of the function will be in some way special, we will use form $f(u) = \sum P_i\theta_i(u)$. Here P_i are control points that gives overall shape to the curve and $\theta_i(u)$ are blending functions that define how particular control point affects shape of curve. Control points form a polygon $P_0P_1...P_n$ called control polygon. Because this is a barycentric combination

of control points, sum of blending functions must be equal to 1, i.e. $\sum \theta_i(u) = 1$. This type and form of function describing curve give us good properties modelling and also for easy evaluation of many properties of curve.
Following basics are described in many publications. There are works focusing on algorithms for implementation of Bézier and NURBS curves [12], theoretic aspects of spline curves [4] or general introduction of curves in CAGD ([5] or [6]).

2.1 Bézier curves

In this section, we focus on the polynomial parametrical curves. The curve is described using function which coordinate functions are polynomial functions in some form. The basic and simplest form of this function can be

$$f(t) = \begin{bmatrix} a_n u^n + ... + a_1 u + a_0 \\ b_n u^n + ... + b_1 u + b_0 \\ c_n u^n + ... + c_1 u + c \end{bmatrix} = \begin{bmatrix} a_n \\ b_n \\ c_n \end{bmatrix} u^n + ... + \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} u^1 + \begin{bmatrix} a_0 \\ b_0 \\ c_0 \end{bmatrix} \quad \begin{bmatrix} a_n \\ b_n \\ c_n \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Here, n is polynomial degree, blending functions are $\theta_i(u) = u^i$ and control points are $P_i = [a_{n-i}, b_{n-i}, c_{n-i}]$. This form isn't usually practical for modelling, we don't know the good relations between shape of control polygon and shape of the curve. For this reason, other blending functions are used that are more suitable for modelling polynomial curves.

Definition: Given polynomial degree $n \in Z, n \geq 0$, we define $(n+1)$ Bernstein polynomials (functions) as $B_i^n : R \rightarrow R; i = 0, 1, ..., n$:

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

If $i < 0$ or $i > n$, then $B_i^n(u) = 0$. There exist also generalized form of Bernstein functions, we will work with it later. Such defined polynomials have many useful properties.

Theorem: Properties of Bernstein polynomials:

- if $u \in \langle 0, 1 \rangle$, then $B_i^n(u) \geq 0$
- $B_0^n(0) = 1$ and $B_i^n(0) = 0$ for $i = 1, ..., n$.
- $B_n^n(1) = 1$ and $B_i^n(1) = 0$ for $i = 0, ..., n-1$.
- $\sum_{i=0}^{n} B_i^n(u) = 1$

- Derivation of Bernstein function $B_i^n(u)$ is equal to $\binom{n}{i}t^{i-1}(1-t)^{n-1-i}(i-nt) = n[B_{i-1}^{n-1}(u) - B_i^{n-1}(u)]$. This leads to observation that Bernstein polynomial $B_i^n(u)$ has maximum at $u = i/n$.
- $B_i^n(u) = uB_{i-1}^{n-1}(u) + (1-u)B_i^{n-1}(u)$ for $n > 0, i = 0, ..., n$.
- The Bernstein functions of degree n form a basis of polynomials of degree n, i.e. each polynomial of degree n can be written as linear combination of $B_i^n(u); i = 0, 1, ..., n$.

Now blending functions are defined, we can start with definition of curve based on these functions. Presented are two equivalent definitions.

Definition: For given polynomial degree $n \in Z, n \geq 0$, $(n+1)$ points $P_i \in E^3, i = 0, ..., n$ and parameter $u \in \langle 0, 1\rangle$, we define points $P_i^k(u), k = 0, ..., n, i = 0, ..., n-k$ in following way:

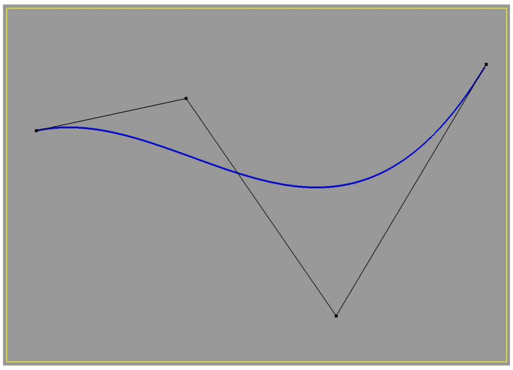
$$P_i^0(u) = P_i, i = 0, ..., n$$
$$P_i^k(u) = (1-u)P_{i-1}^{k-1}(u) + uP_i^{k-1}(u), k = 1, ..., n, i = 0, ..., n-k$$

Then point of n-th degree Bézier curve given by control points P_i for parameter u is point $P_0^n(u)$. This point will be marked as $B^n(u)$, i.e. $B^n(u) = P_0^n(u)$. Presented algorithmic definition is called de Casteljau algorithm.

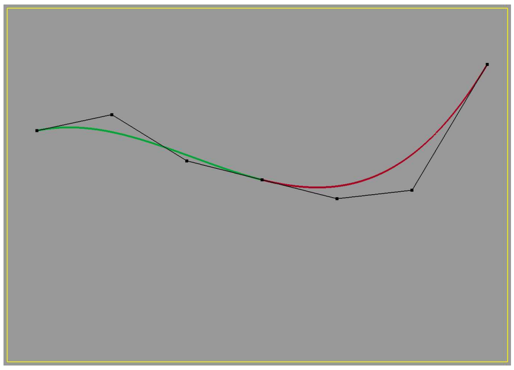
Definition: For given polynomial degree $n \in Z, n \geq 0$, $(n+1)$ points $P_i \in E^3, i = 0, ..., n$ and parameter $u \in \langle 0, 1\rangle >$, we define point of n-th degree Bézier curve given by control points P_i for parameter u as $B^n(u) = \sum_{i=0}^n P_i B_i^n(u)$. This is also called analytical evaluation of Bézier curve.

Theorem: Bézier curve has many properties suitable for modelling:

- The Bézier curve is invariant to affine transformation. If $A : E^3 \rightarrow E^3$ is affine transformation, then $A(\sum_{i=0}^n P_i B_i^n(u)) = \sum_{i=0}^n A(P_i)B_i^n(u)$. So when we want to transform Bézier curve, all we need is to transform control points of that curve.
- The Bézier curve can be defined on any non-empty interval $\langle a, b\rangle$, then curve is given as $Q^n(u) = B^n(\frac{u-a}{b-a})$. This property is called invariation under affine transformation of parameter.
- The Bézier curve lies in the convex hull of its control points.
- The Bézier curve approximates shape of its control polygon.
- The Bézier curve has property of variation diminishing, i.e. straight line can intersect Bézier curve no more often than it intersects control polygon.



(a) Bézier curve with control points.



(b) Bézier curve decomposed into two Bézier curves using de Casteljau algorithm.

Figure 2.1: Bézier curve of degree 3.

- Because $B^n(0) = P_0$ and $B^n(1) = P_n$, Bézier curve always interpolates first and last control points.
- For derivative of Bézier curve, we have $\frac{\partial}{\partial u}B^n(u) = \sum_{i=0}^{n-1} Q_i B_i^n(u)$, where $Q_i = n(P_{i+1} - P_i)$. In this case, the result is vector, because it is linear combination of vectors Q_i.
- i-th control point affects mostly region around parameter $u = i/n$, but generally affects whole curve.
- In the de Casteljau algorithm, points $P_0^0 P_0^1 ... P_0^n$ and $P_n^0 P_{n-1}^1 ... P_0^n$ form two control polygons of two Bézier curves. These two curves together generate given curve. So using de Casteljau algorithm, we can split Bézier curve into two Bézier curves. This process is illustrated in Figure 2.1(b).
- $\sum_{i=0}^{n} P_i B_i^n(u) = \sum_{i=0}^{n+1} Q_i B_i^{n+1}(u)$, where $Q_i = \frac{iP_{i-1}+(n+1-i)P_i}{n+1}, i = 0, ..., n+1, P_{-1} = P_0, P_{n+1} = P_n$. So we can replace control polygon with another control polygon with increased number of points without changes in shape of the curve. This proces is called degree elevation.
- For $u \in (0, 1)$, Bézier curve $B^n(u)$ is C^∞.

From these properties we can create many useful modelling properties. The shape of Bézier curve based on the shape of a control polygon is rendered in Figure 2.1(a).

2.2 B-spline curves

Bézier curves has important disadvantage, higer number of control points leads to the higher degree. This relation between number of control points and degree of the Bézier curve has mainly computational issues and also high order curves are not needed for modelling. So we need independence of degree and number of control points, this can be done using piecewise polynomial curve. First, we need the interval of parameter for whole curve, then we must split this interval on smaller parts so each Bézier curve has its own interval of definition. We must also give control points for each Bézier curve. Whole definition of piecewise Bézier curve is as follows.

Definition: Lets have two non-negative integer numbers d, n, a non-empty interval $\langle a, b\rangle$, real numbers $u_0 = a < u_1 < u_2 < ... < u_n = b$, and control points $P_{i,j} \in E^3; i = 0, ..., d; j = 1, ..., n$. Then piecewise Bézier curve PB^d for parameter $u \in \langle a, b\rangle >$ is given by:

$$u \in \langle u_j, u_{j+1}\rangle,\ PB^d(u) = \sum_{i=0}^{d} P_{i,j+1} B_i^n(\frac{u - u_j}{u_{j+1} - u_j})$$

Usually, we want this curve to be at least C^0, in this case $P_{d,i} = P_{0,i+1}; i = 1, ..., n-1$. We call d a degree of curve, n is number of segments and $(u_0, u_1, u_2, ..., u_n)$ is knot vector, its components are knots.

When joining together Bézier curves, we also want some higher order continuity, usually C^1 or C^2. For example, Figure 2.2(b) shows C^1 piecewise Bézier curve with control points. This goal can be achieved using restriction for control points as in next theorem.

Theorem: A piecewise Bézier curve is C^1 continuous in knot $u_j; j = 1, ..., n-1$ if and only if is C^0 in that knot and $\frac{1}{u_j - u_{j-1}}(P_{d,j} - P_{d-1,j}) = \frac{1}{u_{j+1} - u_j}(P_{1,j+1} - P_{0,j+1})$. Piecewise Bézier curve is C^2 continuous in knot $u_j; j = 1, ..., n-1$ if and only if is C^1 in that knot and $\frac{1}{(u_j - u_{j-1})^2}(P_{d,j} - 2P_{d-1,j} + P_{d-2,j}) = \frac{1}{(u_{j+1} - u_j)^2}(P_{0,j+1} - 2P_{1,j+1} + P_{2,j+1})$.

Higher order continuity gives lesser degree of freedom for placing control points. We need to input lesser number of control points. This then leads to a new set of control points with decreased number of control points and a new set of blending functions. This also leads to extended knot vector with multiplied knots. Based on these observations, a new set of blending functions can be introduced.

Definition: Let $u_0 \leq u_1 \leq ... \leq u_m$ is sequence of real numbers. For $k = 0, 1, ..., m$, and $i = 0, ..., m-k-1$, define i-th B-spline blending function of degree k as

$$N_i^0(u) = \begin{cases} 1 & \text{for } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and for $k > 0$

$$N_i^k(u) = \begin{cases} \frac{u-u_i}{u_{i+k}-u_i} N_i^{k-1}(u) + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1}^{k-1}(u) & u_{i+k} \neq u_i,\ u_{i+k+1} \neq u_{i+1} \\ \frac{u-u_i}{u_{i+k}-u_i} N_i^{k-1}(u) & u_{i+k} \neq u_i,\ u_{i+k+1} = u_{i+1} \\ \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1}^{k-1}(u) & u_{i+k} = u_i,\ u_{i+k+1} \neq u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Theorem: Properties of B-spline blending functions:

- If $u \geq u_{i+k+1}$ or $u < u_i$, then $N_i^k(u) = 0$, i.e. $N_i^k(u)$ is nonzero only in the interval $\langle u_i, u_{i+k+1})$. In that interval, $N_i^k(u) > 0$ for $u \in (u_i, u_{i+k+1})$.
- If $u \in (u_j, u_{j+1})$ and $i = j-k, j-k+1, ..., j$, then $N_i^k(u) > 0$.
- $\sum_{i=0}^{m-k-1} N_i^k(u) = 1$.

After the definition of blending functions, we can define curve based on these functions. Just like for Bézier curves, we have two equivalent definitions, one algorithmic and one analytical expression of B-spline curves. The algorithmic definition is also called de Boor algorithm.

Definition: Lets have real numbers $u_0 \leq u_1 \leq ... \leq u_m$, degree $d \in N, d \geq 1$ and control points $P_0, P_1, ..., P_n \in E^3$, where $n = m - d - 1$. Then B-spline curve N^d of degree d over knot vector $(u_0, u_1, ..., u_m)$ is defined over interval $\langle u_d, u_{n+1})$ in two equivalent ways as

- $N^d(u) = \sum_{i=0}^{n} P_i N_i^d(u) \quad u \in \langle u_d, u_{n+1})$
- Step 1: $u \in \langle u_d, u_{n+1})$, find J such that $u \in \langle u_J, u_{J+1})$.
  Step 2: Define $P_i^0(u) = P_i, i = 0, 1, ..., n$.
  Step 3: For $k = 1, ..., d$ and $i = J - d + k, ..., J$, let
  $$P_i^k(u) = \frac{u - u_i}{u_{i+d-k+1} - u_i} P_i^{k-1}(u) + \frac{u_{i+d-k+1} - u}{u_{i+d-k+1} - u_i} P_{i-1}^{k-1}(u)$$
  Step 4: $N^d(u) = P_0^d(u)$.

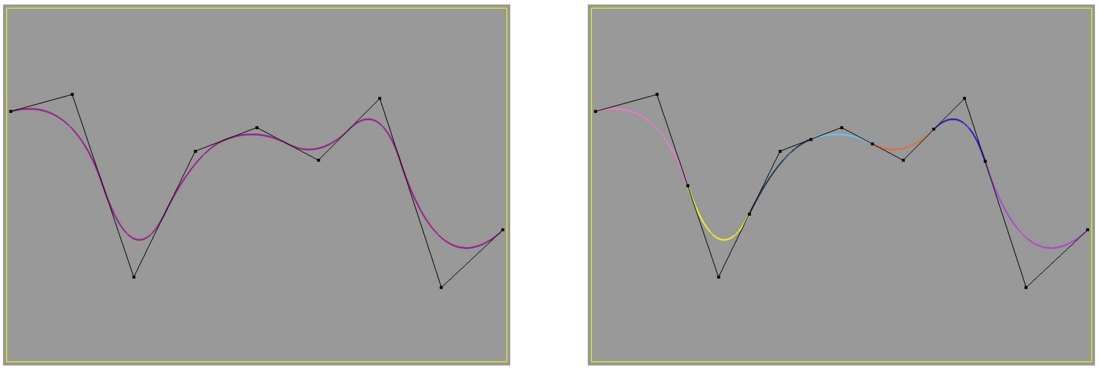
B-spline curve is another form of spline (or piecewise polynomial) curve, one example of such curve is in Figure 2.2(a). The following properties show how knot vector is related to shape of curve and how approximate control polygon. It also shows another modelling properties similar to properties of Bézier curves as well as algorithm for conversion from B-spline to piecewise Bézier form.

Theorem: Properties of B-spline curve N^d:

- B-spline curve is invariant under an affine transformation. So if $A : E^3 \to E^3$ is affine transformation, then $A(\sum_{i=0}^{n} P_i N_i^d(u)) = \sum_{i=0}^{n} A(P_i) N_i^d(u)$
- B-spline curve lies in the convex hull of its control points.
- B-spline curve approximates shape of its control polygon.
- B-spline curve has property of variation diminishing, i.e. straight line can intersect B-spline curve no more often than it intersect control polygon.
- If $u_0 = u_1 = ... = u_d$, then B-spline curve interpolates first control point. If $u_{m-d} = u_{m-d+1} = ... = u_m$, then B-spline curve interpolates last control point.
- If $n = d + 1$, $u_0 = u_1 = ... = u_d = 0$ and $u_{m-d} = u_{m-d+1} = ... = u_m = 1$, then B-spline curve is Bézier curve with the same control polygons and de Boor algorithm is simply de Casteljau algorithm.
- If knot u_j has multiplicity m_j in knot vector, then in this knot, B-spline is at least C^{d-m_j}.
- If each knot has multiplicity $d+1$ in knot vector, then B-spline curve with control polygon becomes piecewise Bézier curve. In Figure 2.2, B-spline curve and its piecewise Bézier version is illustrated.
- When the derivative of B-spline curve exists, it is given by $\frac{\partial}{\partial u} N^d(u) = \sum_{i=0}^{n-1} Q_i N_i^{d-1}(u)$, where $Q_i = d \frac{P_i - P_{i-1}}{u_{i+d} - u_i}$. In this case, the result is vector, because it is a linear combination of vectors Q_i.
- Lets have B-spline curve $N^d(u) = \sum_{i=0}^{n} P_i N_i^d(u)$ with control points P_i and knot vector $\mathbf{u} = (u_0, u_1, ..., u_m)$. Let $\overline{u}$ be a real value such that $u_J \leq \overline{u} < u_{J+1}$, and form a new knot vector $\mathbf{u} \cup \{\overline{u}\}$ with corresponding B-spline functions $\overline{N}_i^d(u)$. Then we can write the B-spline curve as another B-spline curve with increased number of control points and knots as $\sum_{i=0}^{n} P_i N_i^d(u) = \sum_{j=0}^{n+1} Q_j \overline{N}_j^d(u)$, where

$$N_i^d(u) = \begin{cases} \overline{N}_i^d & i \leq J - d + 1 \\ \frac{\overline{u} - u_i}{u_{i+d+1} - u_i} \overline{N}_i^d(u) + \frac{u_{i+d+2} - \overline{u}}{u_{i+d+2} - u_{i+1}} \overline{N}_{i+1}^d(u) & J - d \leq i \leq J \\ \overline{N}_{i+1}^d & J + 1 \leq i \end{cases}$$

$$Q_j = \begin{cases} P_j & j \leq J - d \\ \frac{\overline{u} - u_j}{u_{j+d+1} - u_j} P_j + \frac{u_{j+d+1} - \overline{u}}{u_{j+d+1} - u_j} P_{j-1} & J - d + 1 \leq j \leq J \\ P_{j-1} & J + 1 \leq j \end{cases}$$



(a) Spline curve in B-spline form with displayed control points and knot vector $(0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 7, 7)$.

(b) Piecewise Bézier spline curve with control points of each segment.

Figure 2.2: Spline curve of degree 2.

This algorithm is called also simple knot insertion or Bohm algorithm, because it changes knot vector and control polygon without changing shape of the curve.

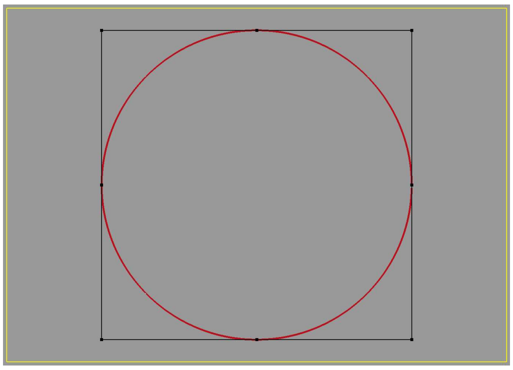
2.3 Rational curves

Polynomial and piecewise polynomials represent a wide set of curves, but not all simple curves are part of this set. There are still curves that we want to represent with help of Bézier or B-spline blending functions. For example, some conics like circle or ellipse can't be described using polynomial or piecewise polynomial functions. A circle with the center in the origin of coordinate system and radius R lying in xy coordinate plane can be written as $\{X \in E^3; X = [\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}, 0]\}$, i.e. as ratio of two polynomials. Such functions are called rational functions. We introduce rational blending functions based on Bernstein and B-spline blending functions. In the case of curves, this also introduces a new set of real parameters called weights, one weight for each control point. Usually weights are non-negative numbers.

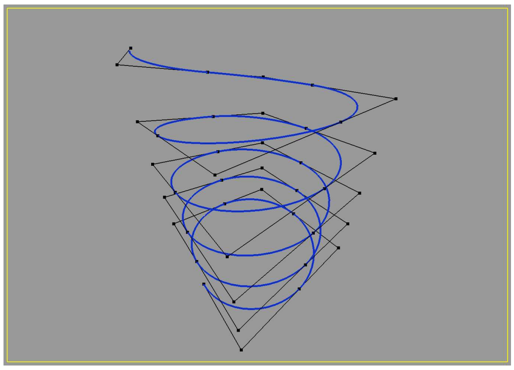
Definition: Assume given degree $n \in N$, control points $P_i \in E^3; i = 0, 1, ..., n$ and for each control point given weight $w_i \in R; w_i \geq 0; i = 0, 1, ..., n$. Not all w_i can be equal to 0. Then rational the Bézier curve RB^n is defined as

$$RB^n(u) = \frac{\sum_{i=0}^n w_i P_i B_i^n(u)}{\sum_{i=0}^n w_i B_i^n(u)} = \sum_{i=0}^n P_i \frac{w_i B_i^n(u)}{\sum_{i=0}^n w_i B_i^n(u)} \quad u \in \langle 0, 1 \rangle$$

Definition: Assume given degree $d \in N$, knot vector $u_0 \leq u_1 \leq ... \leq u_m$, control points $P_i \in E^3; i = 0, 1, ..., n$, $n = m - d - 1$ and for each control point given weight



(a) Circle as NURBS curve of degree 2 with 9 control points and knot vector $(0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1)$.



(b) Spiral as three dimensional NURBS curve with displayed control polygon.

Figure 2.3: Examples of rational curves.

$w_i \in R; i = 0, 1, ..., n$. Not all w_i can be equal to 0. Then rational B-spline curve RN^d is defined as

$$RN^d(u) = \frac{\sum_{i=0}^{n} w_i P_i N_i^d(u)}{\sum_{i=0}^{n} w_i N_i^d(u)} = \sum_{i=0}^{n} P_i \frac{w_i N_i^d(u)}{\sum_{i=0}^{n} w_i N_i^d(u)} \quad u \in \langle u_d, u_{n+1})$$

Rational B-spline curve is called NURBS (non-uniform rational B-spline) curve, examples of NURBS curve are in Figure 2.3.
This definition gives a new class of rational and piecewise rational curves with many similar properties as for polynomial curves. We get these curves also by projection of polynomial and piecewise polynomial curves from four dimensional space E^4 to three dimensional space E^3. This technique is simple, first take given control points $P_i = [x_i, y_i, z_i]$ and weights w_i and create four dimensional points $P_i^w = [w_i x_i, w_i y_i, w_i z_i, w_i]$. Then compute point of Bézier or B-spline curve with control points P_i^w and then make projection of result into E^3. This projection is done when the first three coordinates of point in E^4 are divided by the fourth coordinate. Using this technique, many properties of rational version of curves can be proven with properties of polynomial versions. For example, it is easy to create rational version of de Casteljau or de Boor algorithms.

2.4 Subdivision curves

Subdivision curves are not exactly parametric curves, they generate its own special class, but are related to parametric curves. In the limit process, they converge to parametric curves. At the beginning, there is given polygon $\mathbf{P_0} = P_1 P_2 ... P_n$ called control polygon. Subdivision process is then set of rules, that from the polygon $\mathbf{P_i}$ produces a new, finer polygon $\mathbf{P_{i+1}}$ with more points. This iteration process is stopped when a good approximation is achieved or new steps of process will not present any significant changes. One similar process was introduced with de Casteljau algorithm for Bézier curves, where control polygon was replaced with two control polygons of two new Bézier curves that together created the old curve. Also the degree elevation creates new, finer control polygon and converges to Bézier curve. Here, we will present two another subdivision processes for curves.

Scheme: Lets have control polygon $\mathbf{P_0}$. Then Chainkin subdivision process or Chainkin curve is defined with these rules: If $\mathbf{P_i} = P_1P_2...P_n$ is polygon after i-th step of process, then polygon after one step is $\mathbf{P_{i+1}} = Q_1Q_2...Q_m$, where $m = 2*n$ if polygon $\mathbf{P_i}$ is closed (when there is segment between P_n and P_1) or $m = 2*n-2$ otherwise. Points of new control polygon are

- If polygon $\mathbf{P_i}$ is closed, then $\mathbf{P_{i+1}}$ is closed and

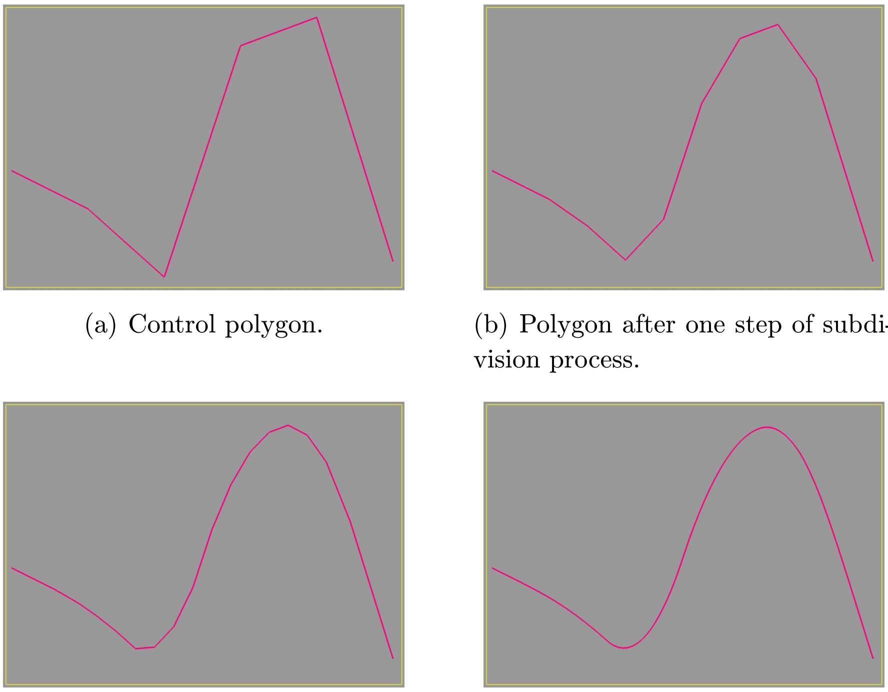
$$\begin{aligned} Q_1 &= \frac{3}{4}P_1 + \frac{1}{4}P_n \\ Q_{2i} &= \frac{3}{4}P_i + \frac{1}{4}P_{i+1} \quad i = 1, ..., n-1 \\ Q_{2i+1} &= \frac{1}{4}P_i + \frac{3}{4}P_{i+1} \quad i = 1, ..., n-1 \\ Q_{2n} &= \frac{1}{4}P_1 + \frac{3}{4}P_n \end{aligned}$$

- If polygon $\mathbf{P_i}$ is not closed, then $\mathbf{P_{i+1}}$ is not closed and

$$\begin{aligned} Q_1 &= \frac{3}{4}P_1 + \frac{1}{4}P_2 \\ Q_{2i} &= \frac{1}{4}P_i + \frac{3}{4}P_{i+1} \quad i = 1, ..., n-1 \\ Q_{2i+1} &= \frac{3}{4}P_{i+1} + \frac{1}{4}P_{i+2} \quad i = 1, ..., n-2 \\ Q_{2n-2} &= \frac{1}{4}P_{n-1} + \frac{3}{4}P_n \end{aligned}$$

This subdivision algorithm is called the corner cutting algorithm because new polygon is created from the old one by cutting the old polygon near its points. Process converges to quadratic uniform B-spline. A few steps of this subdivision algorithm are in Figure 2.4.

Scheme: For control polygon $\mathbf{P_0}$, Catmull-Clark subdivision process is defined with rules: If $\mathbf{P_i} = P_1P_2...P_n$ is polygon after i-th step of process, then polygon after one next step is $\mathbf{P_{i+1}} = Q_1Q_2...Q_m$, where $m = np + ns$, np is number of points and ns is number of segments of $\mathbf{P_i}$. The points of new polygon $\mathbf{P_{i+1}}$ are created from points of previous polygon $\mathbf{P_i}$:



(a) Control polygon.

(b) Polygon after one step of subdivision process.

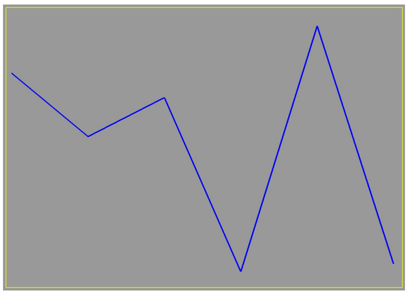
(c) Polygon after two steps of subdivision process.

(d) Polygon after four steps of subdivision process.

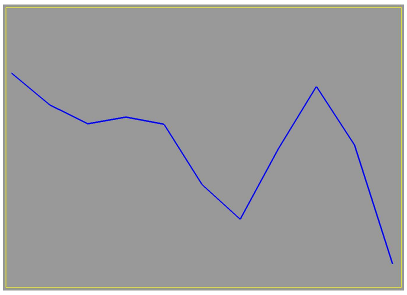
Figure 2.4: Chainkin subdivision process.

- For each segment of polygon $\mathbf{P_i}$ create edge point E as average of end points of segment.
- For each point P of polygon $\mathbf{P_i}$, create a new vertex point $V_j = \frac{1}{2}P + \frac{1}{4}A + \frac{1}{4}B$, where A, B are previously created edge points of segments that are neighbors to point P (if particular segment does not exist, such edge point is ignored).
- New polygon $\mathbf{P_{i+1}}$ is created by alternating vertex and edge points, $\mathbf{P_{i+1}} = V_1E_1V_2E_2$....... If $\mathbf{P_i}$ was closed, then also $\mathbf{P_{i+1}}$ must be closed.

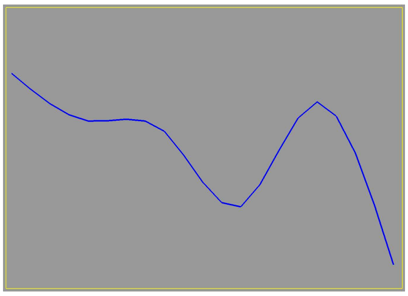
This process converges to a cubic uniform B-spline. Process is illustrated in Figure 2.5.



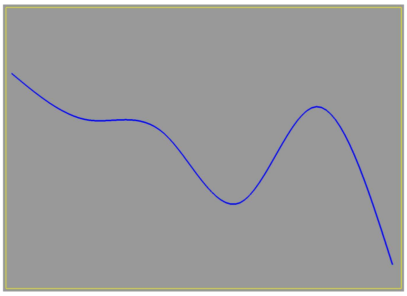
(a) Control polygon.



(b) Polygon after one step of subdivision process.



(c) Polygon after two steps of subdivision process.



(d) Polygon after five steps of subdivision process.

Figure 2.5: Catmull-Clark subdivision process for curves.

Chapter 3

Parametric surfaces

Surfaces are widely used in computer graphics like in description of object's boundary, fitting data or in approximation theory. We know many types of representation for surfaces, there are implicit surfaces, meshes (surface consists of closed filled polygons) or parametric surfaces. Parametric surfaces are a natural extension of parametric curves. One real parameter is added, the surfaces are described by bivariate functions, or functions with two dimensional domain. Again, we discuss only about polynomial, piecewise polynomial and rational functions.

Definition: Parametric surface in E^3 is set of points $C = \{X \in E^3; X = f(\mathbf{u}); \mathbf{u} \in U, U \subset R^2\}$, where $f : R^2 \to E^3$ is function describing surface, $\mathbf{u}$ is parameter, U is domain of surface.

Function f will be described by control points and blending functions, but there are more forms for it. One form produces tensor product surfaces, other one (similar to form for curves) will be introduced for Bézier triangles.

Definition: Consider two sets of univariate functions, $\{f_i(u)\}_{i=0}^n; \sum_{i=0}^n f_i(u) = 1$ and $\{g_j(v)\}_{j=0}^m; \sum_{j=0}^m g_j(v) = 1$, with interval domains U and V, respectively. For given control points $P_{i,j} \in E^3; i = 0, 1, ..., n; j = 0, 1, ..., m$, a surface formed by $h(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} f_i(u) g_j(v)$ is called tensor product parametric surface with domain $U \times V$.

Because tensor product surface can be written also as

$$h(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{i,j} f_i(u) g_j(v) = \sum_{i=0}^{n} f_i(u) [\sum_{j=0}^{m} P_{i,j} g_j(v)]$$

, many properties and evaluations can be prepared using properties for curves with given blending functions, first for curve with blending functions $\{g_j(v)\}_{j=0}^m$ and original control points $P_{i,j}$, then use the same approach for curve with evaluated

points and blending functions $\{f_i(u)\}_{i=0}^n$.
All basic definition and properties of parametric spline surfaces are treated together with parametric curves in many publications or web articles. Just like for curves, good books to start with are [12], [4] or [5].

3.1 Bézier surfaces

Two types of Bézier surfaces are known, each with its own configuration of control points, these control points are blended using polynomial functions. These two forms can be extended from two ways how Bézier curve can be written

$$B^n(u) = \sum_{i=0}^{n} P_i \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

$$B^n(\mathbf{u}) = \sum_{|\mathbf{i}|=i+j=n} P_{\mathbf{i}} \frac{n!}{i!j!} u^i v^j \qquad \mathbf{u} = (u,v); u+v = 1$$

Extension of these two forms with another parameter leads to a Bézier tensor product surface and a Bézier triangle. As a blending function, Bernstein polynomials and generalized Bernstein polynomials are used again.

Definition: Given points $P_{i,j} \in E^3; i = 0,1,...,n; j = 0,1,...,m$, Bézier tensor product surface $B^{n,m}$ is given analytically as

$$B^{n,m}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{i,j} B_i^n(u) B_j^m(v) \qquad u,v \in \langle 0,1 \rangle$$

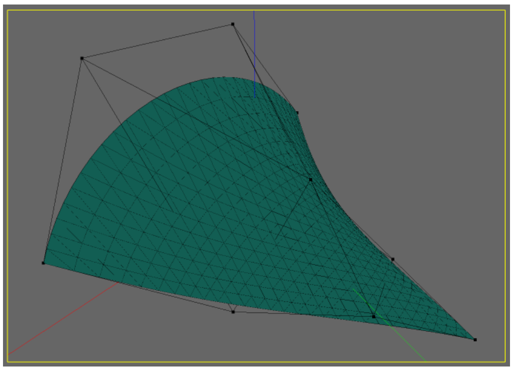
where n,m are degrees in u resp. v direction, $\langle 0,1\rangle \times \langle 0,1\rangle$ is domain of surface and $P_{i,j}$ are control points.

Definition: For a given degree $n \in N$, points $P_{(i,j,k)} \in E^3; i,j,k \in N; i+j+k = n$, Bézier triangle BT^n is defined analytically as

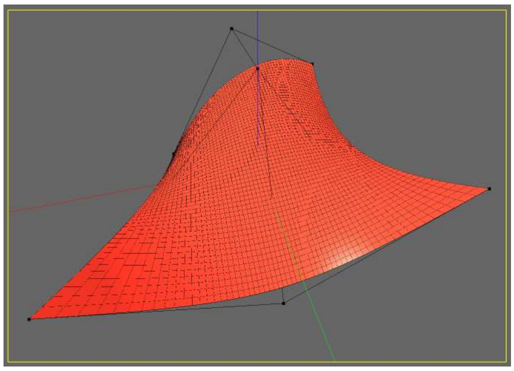
$$BT^n(\mathbf{u}) = \sum_{|\mathbf{i}|=i+j+k=n} P_{\mathbf{i}} \frac{n!}{i!j!k!} u^i v^j w^k = \sum_{|\mathbf{i}|=i+j+k=n} P_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u})$$

where $\mathbf{u} = (u,v,w); 0 \le u \le 1; 0 \le v \le 1; 0 \le w \le 1; u+v+w = 1$. The parameter n is called degree, $P_{\mathbf{i}}$ are control points, domain is triangle $T = \{(u,v); 0 \le u \le 1; 0 \le v \le 1; u+v \le 1\}$. $B_{\mathbf{i}}^n(\mathbf{u}) = \frac{n!}{i!j!k!} u^i v^j w^k$ is called trivariate Bernstein function.

For these surfaces, shape of domain, net of control points and patch are similar. For Bézier tensor product surface, this shape is quadrilateral, for Bézier triangle, the shape is triangle. There are definitions of Bézier triangle where domain is a



(a) Bézier triangle of degree 3.



(b) Bézier tensor product surface of degrees $(2, 2)$.

Figure 3.1: Bézier surfaces with nets of control points.

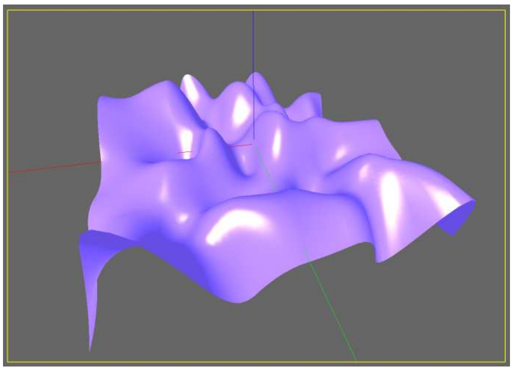
non-degenerative triangle. Figure 3.1 shows examples of two types of Bézier surfaces.
The properties of Bézier surfaces are very similar to properties of Bézier curves, almost every evaluations and proofs are done with similarity to curve case. For example, corner control points always interpolate Bézier patches.
Bézier surfaces are used as base for piecewise polynomial surfaces, sometimes splines are decomposed to its Bézier segments and task is performed on these segments. Also can be used form smoothing triangular or quadrilateral meshes, e.g. PN triangles [20].

3.2 B-spline surfaces

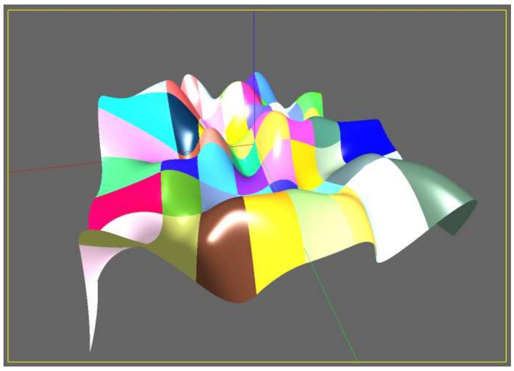
Only one type is presented, B-spline tensor product surface. It is defined with all attributes as B-spline curve, but every parameter is doubled, for u and v direction. At the end, this form is general form for describing piecewise polynomial surfaces. It is also, with some specific configuration, piecewise Bézier form. All comes from similar properties for curves. We will present definition using analytical expression.

Definition: Assume given real numbers $u_0 \leq u_1 \leq ... \leq u_{m_u}$, $v_0 \leq v_1 \leq ... \leq v_{m_v}$, degrees $d_u, d_v \in N; d_u, d_v \geq 1$ and control points $P_{i,j} \in E^3; i = 0, 1, ..., n_u; j = 0, 1, ..., n_v$, where $n_u = m_u - d_u - 1, n_v = m_v - d_v - 1$. Then B-spline surface N^{d_u,d_v} of degrees d_u, d_v over knot vectors $(u_0, u_1, ..., u_{m_u})$ and $(v_0, v_1, ..., v_{m_v})$ is defined over interval $\langle u_{d_u}, u_{n_u+1}\rangle \times \langle v_{d_v}, v_{n_v+1}\langle$ as

$$N^{d_u,d_v}(u, v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} P_{i,j} N_i^{d_u}(u) N_i^{d_v} j(v) \quad u \in \langle u_{d_u}, u_{n_u+1}\rangle, v \in \langle v_{d_v}, v_{n_v+1}\rangle$$



(a) B-spline surface with 10 × 10 control points.



(b) Decomposition to Bézier segments.

Figure 3.2: B-spline tensor product surfaces of degrees (2, 2).

where $N_i^{d_u}(u)N_j^{d_v}(v)$ are B-spline blending functions.

Properties of B-spline surface are similar to properties of B-spline curve, for each (u and v) direction separately. For example, a simple knot insertion process in u direction must be performed on each row of control points and leads to increased number of control points in u direction. Figure 3.2 shows example of B-spline tensor product surface and its decomposition to Bézier tensor product surfaces by inserting knots in u and v direction.

3.3 Rational surfaces

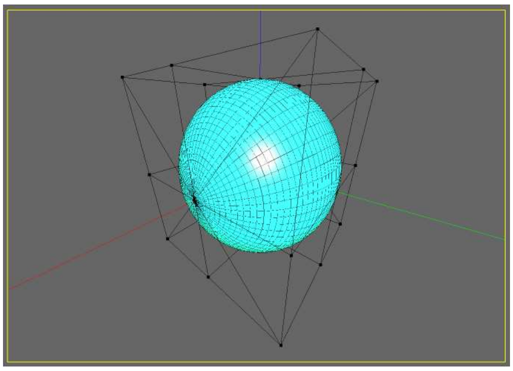
Rational surfaces extend class of polynomial surfaces, so some useful patches like sphere can be represented using this form. Rational surfaces bring a new set of real parameters, for each control point one real non-negative parameter called weight.

Definition: For given degrees $n, m \in N$, points $P_{i,j} \in E^3; i = 0, 1, ..., n; j = 0, 1, ..., m$, real numbers $w_{i,j} \in R; w_{i,j} \geq 0; i = 0, 1, ..., n; j = 0, 1, ..., m$ that are not all zero, then rational Bézier tensor product surface $RB^{n,m}$ is given analytically as

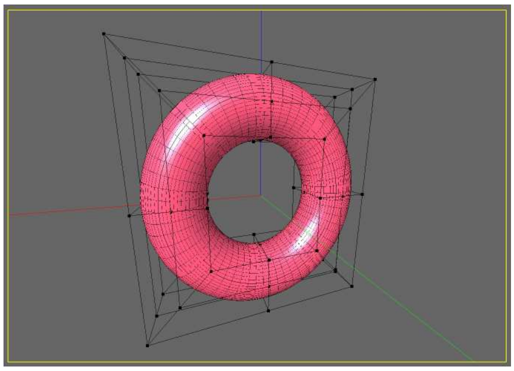
$$RB^{n,m}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j}P_{i,j}B_i^n(u)B_j^m(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{i,j}B_i^n(u)B_j^m(v)} \qquad u, v \in \langle 0, 1\rangle$$

where n, m are degrees in u resp. v direction, $\langle 0, 1\rangle \times \langle 0, 1\rangle$ is domain of surface, $P_{i,j}$ are control points and $w_{i,j}$ are weights.

Definition: For a given degree $n \in N$, points $P_{\mathbf{i}} = P_{(i,j,k)} \in E^3; i, j, k \in N; i + j + k = n$, real numbers $w_{\mathbf{i}} = w_{(i,j,k)} \in E^3; w_{(i,j,k)} \geq 0; i, j, k \in N; i + j + k = n$, define



(a) Sphere.



(b) Torus.

Figure 3.3: Examples of rational B-spline surfaces (NURBS surfaces).

rational Bézier triangle RBT^n analytically as

$$RBT^n(\mathbf{u}) = \frac{\sum_{|\mathbf{i}|=i+j+k=n} w_{\mathbf{i}} P_{\mathbf{i}} B^n_{\mathbf{i}}(\mathbf{u})}{\sum_{|\mathbf{i}|=i+j+k=n} w_{\mathbf{i}} B^n_{\mathbf{i}}(\mathbf{u})}$$

where $\mathbf{u} = (u, v, w); 0 \leq u \leq 1; 0 \leq v \leq 1; 0 \leq w \leq 1; u + v + w = 1$ and $B^n_{\mathbf{i}}(\mathbf{u})$ are trivariate Bernstein polynomials. Parameter n is called degree, $P_{\mathbf{i}}$ are control points, $w_{\mathbf{i}}$ are weights, domain is triangle $T = \{(u, v) \in E^2; 0 \leq u \leq 1; 0 \leq v \leq 1; u + v \leq 1\}$.

Definition: For given real numbers $u_0 \leq u_1 \leq ... \leq u_{m_u}$, $v_0 \leq v_1 \leq ... \leq v_{m_v}$, degrees $d_u, d_v \in N; d_u, d_v > 1$ and control points $P_{i,j} \in E^3; i = 0, 1, ..., n_u; j = 0, 1, ..., n_v$, real numbers $w_{i,j} \in R; i = 0, 1, ..., n_u; j = 0, 1, ..., n_v$, where $n_u = m_u - d_u - 1, n_u = m_v - d_v - 1$. Then rational B-spline surface RN^{d_u,d_v} of degrees d_u, d_v over knot vectors $(u_0, u_1, ..., u_{m_u})$ and $(v_0, v_1, ..., v_{m_v})$ is defined over interval $\langle u_{d_u}, u_{n_u+1}) \times \langle v_{d_v}, v_{n_v+1})$ as

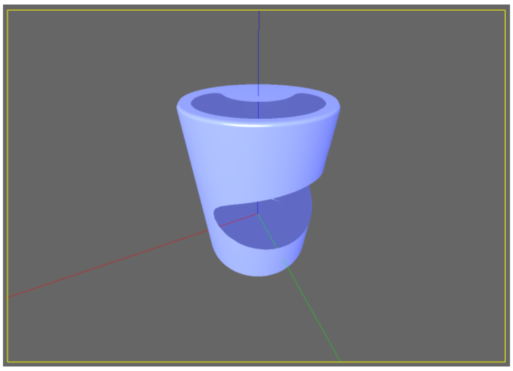
$$RN^{d_u,d_v}(u, v) = \frac{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} w_{i,j} P_{i,j} N_i^{d_u}(u) N_i^{d_v}(v)}{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} w_{i,j} N_i^{d_u}(u) N_i^{d_v}(v)}$$

$$u \in \langle u_{d_u}, u_{n_u+1}), v \in \langle v_{d_v}, v_{n_v+1})$$

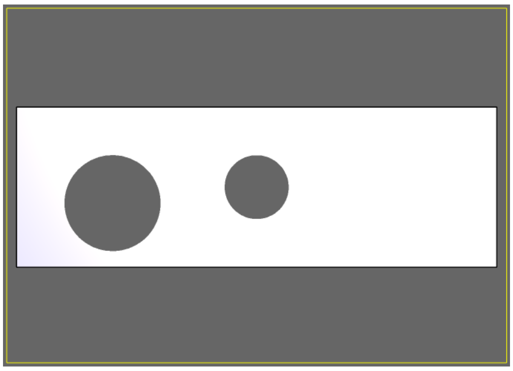
Rational B-spline surfaces (NURBS surfaces) are one of the most used representations of smooth curtved surfaces in academic and commercial products, now it is industrial standard. Some basic objects described as NURBS surfaces are in Figure 3.3.

3.4 Trimmed surfaces

Trimming of the parametric surfaces is an easy but a very powerful technique for extending class of rational surfaces with patches that consist just part of original



(a) Trimmed cylinder with holes.



(b) Domain of cylinder with trimming curves creating holes.

Figure 3.4: Trimmed surface.

surface. This approach is useful for representing result of Boolean operation with two parametric surfaces. Analytical solution of such operation is often too difficult to solve. Main idea is to divide domain of the surface into two parts. One part becomes valid domain, the other is invalid. Then the parameters from valid part only are taken and evaluated.

Definition: Lets have parametric surface $\{X \in E^3; X = f(\mathbf{u}); \mathbf{u} \in U, U \subset R^2\}$, where U is domain of this surface. Take subset $\overline{U} \subset U$. Then trimmed parametric surface is set $\{X \in E^3; X = f(\mathbf{u}); \mathbf{u} \in \overline{U}\}$.

For Bézier or B-spline surfaces, the domain is triangle or rectangle. For marking just part of these domains, we insert curves into domain space, these curves will present border between valid and invalid regions of domain. These curves are often in NURBS form. To evaluate if some point in domain is valid or not, an arbitrary ray is shot from that point, and number of intersections of ray and border curves is computed. If number of intersections is even, point is valid, otherwise it is invalid. One such domain with two trimming curves is in Figure 3.4 with corresponding trimmed NURBS surface.

3.5 Subdivision surfaces

Subdivision surfaces create bridge between meshes and smooth parametric surfaces. Like for curves, each type of subdivision surface is represented by set of rules that describe how to create new points and how to create topology over these new points (or how to connect these new points with edges and faces). These rules are called also stencils and in one scheme, there are different rules for regular vertices or irregular (extraordinary) vertices, boundary vertices, vertices on defined creases

and corners. These rules often work in the close neighborhood of vertex, edge or face. The regularity of vertices depends on topology of mesh, for triangular meshes, regular vertices have six neighbors, otherwise the vertex is irregular. We know many subdivision schemes, there are schemes that work only on triangular meshes, quadrilateral meshes or meshes with arbitrary topology. Subdivision schemes can approximate or interpolate given control mesh. Also some schemes can change rules during subdivision process. We give basic subdivision surfaces that extends described subdivision curves, and add some schemes for triangular meshes.
Subdivision surfaces are very popular in current stage of computer graphics. This popularity comes from easy implementation of such subdivision algorithms and it works on polygonal meshes so it is also easy to render it. Basic introduction to subdivision surfaces can be found in [1] or [21]. Description of basic schemes on triangular meshes was given also in [15].

Scheme: Let us have given control mesh $\mathbf{P_0}$ with arbitrary topology. Then Doo-Sabin subdivision surfaces over control mesh $\mathbf{P_0}$ is defined: If $\mathbf{P_i}$ is mesh after i-th step of process, then mesh after one additional step is $\mathbf{P_{i+1}}$ and it is created using the following rules:

- For each face $F = (V_0V_1...V_K)$ of mesh $\mathbf{P_i}$ and for each vertex $V_i; i = 0, 1, ..., K$ of that face, create new face-vertex point F_i. If $K = 3$, face F is quad and $F_i = \frac{9}{16}V_i + \frac{3}{16}V_{(i-1)mod4} + \frac{3}{16}V_{(i+1)mod4} + \frac{1}{16}V_{(i+2)mod4}$. If $K \neq 3$, then $F_j = \sum_{k=0}^{K} \alpha_k V_{(i+k)mod(K+1)}$, where $\alpha_0 = \frac{1+5K}{4}$ and $\alpha_k = \frac{1}{K}(3 + 2cos\frac{2\pi k}{K})$ for $k = 1, 2, ..., K$. These points become points of mesh $\mathbf{P_{i+1}}$.

- For each vertex of mesh $\mathbf{P_i}$, create new face from all face-vertex points belonging to given vertex.

- For each edge of mesh $\mathbf{P_i}$, create new face from all face-vertex points that were created from faces adjacent to edge and end vertices of edge.

- For each face of mesh $\mathbf{P_i}$, create new face from all face-vertex belonging to given face.

- All newly created faces become faces of mesh $\mathbf{P_{i+1}}$.

This subdivision algorithm is called corner cutting algorithm because the new mesh is created from the old one by cutting the old polygon near its vertices and edges. The process was created for quadrilateral meshes and extended for meshes on arbitrary topology and converges to a quadratic uniform B-spline surface and approximates given control polygon.

Scheme: For a given control mesh $\mathbf{P_0}$ with arbitrary topology, Catmull-Clark subdivision surfaces over control mesh $\mathbf{P_0}$ is defined: If $\mathbf{P_i}$ is mesh after i-th step of process, then the next step mesh is $\mathbf{P_{i+1}}$ that is created using following rules:

- For each face of mesh $\mathbf{P_i}$, create new face point F as average of that face vertices.
- For each edge of mesh $\mathbf{P_i}$, create new edge point E as the average of that edge end vertices and face points of adjacent faces. If edge is on boundary, E is just average of end vertices of edge.
- For each vertex V of mesh $\mathbf{P_i}$, create a new vertex point $\overline{V}$ as $\overline{V} = \frac{1}{n}Q + \frac{2}{n}R + \frac{n-3}{n}V$, where Q is the average of the new face points surrounding the vertex V, R is the average of the midpoints of the edges that share the vertex V and n is the number of edges that share the vertex V.
- All new face, edge and point vertices are vertices of mesh $\mathbf{P_{i+1}}$.
- For each face and each vertex V of mesh $\mathbf{P_i}$, create quad $ABCD$. A is new vertex point from V, B and D are edge points of edges adjacent to V and C is face point of given face. All faces (quads) created this way are faces (polygons) of mesh $\mathbf{P_{i+1}}$

This works on meshes of arbitrary topology, but after first step of process, all meshes contains only quads. Process produces meshes that approximate control mesh, example of process is in Figure 3.8. It can be proven that this process converges to quadratic uniform B-spline surface almost everywhere.

Scheme: Let us have triangular control mesh $\mathbf{P_0}$. Then Loop approximative subdivision surfaces over control mesh $\mathbf{P_0}$ is defined: If $\mathbf{P_i}$ is mesh after i-th step of process, then the mesh after one additional step is $\mathbf{P_{i+1}}$ that is created using following rules:

- For each edge of mesh $\mathbf{P_i}$, create new edge point E as $E = \frac{3}{8}A + \frac{3}{8}B + \frac{1}{8}C + \frac{1}{8}D$, where A and B are end points of edge and C, D are third points of faces adjacent to edge.
- For each vertex V of mesh $\mathbf{P_i}$, create new vertex point $\overline{V}$ as $\overline{V} = (1 - k\beta(k))V + \sum_{j=1}^{k} \beta(k)N_j$, where $N_1, ..., N_k$ are neighbors of vertex V in mesh $\mathbf{P_i}$. Parameter β can be chosen in different ways, acceptable values are $\beta(k) = \frac{1}{k}(\frac{5}{8} - \frac{(3+2cos(2\pi/k))^2}{64})$, $\beta(k) = \frac{3}{k(k+2)}$, $\beta(k) = \frac{3}{16}$ for $k = 3$ or $\beta(k) = \frac{3}{8k}$ otherwise.
- For each face (triangle) ABC create four new triangles from three vertex points and three edge points like in Figure 3.5.

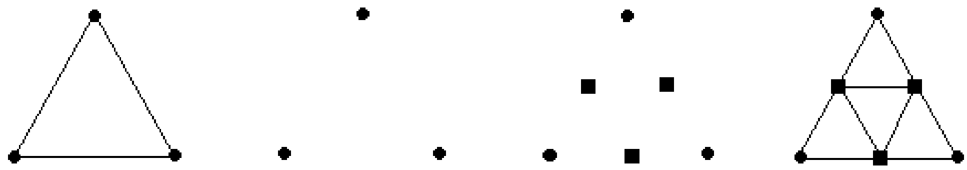


Figure 3.5: Creation of new triangles from edge and vertex points for Loop and modified butterfly subdivision schemes.

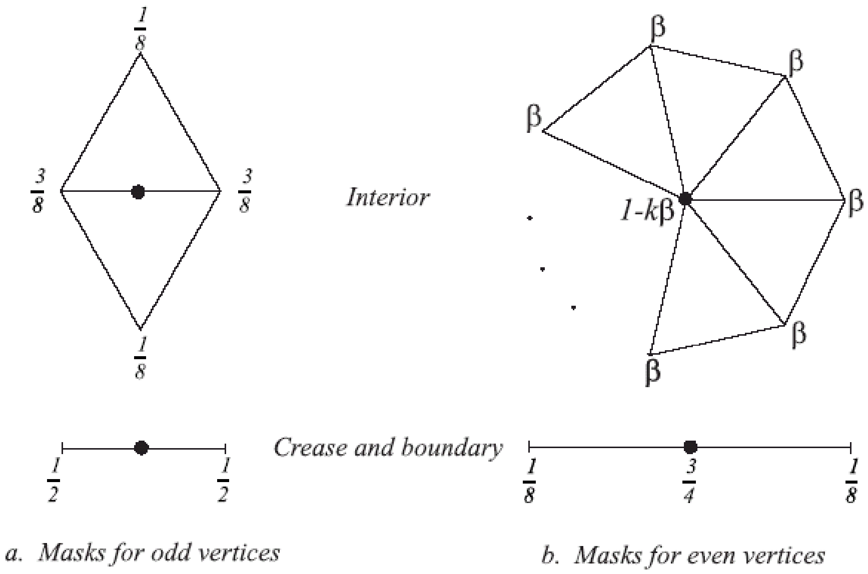


Figure 3.6: Masks for creation of new vertices when performing one step of loop subdivision process. Figure from [21].

- All new faces and vertices generate new mesh $\mathbf{P_{i+1}}$.

These rules are only for the basic mesh without a boundary or creases, complete rules can be found in [21]. Rules how to create new vertices can be given graphically, they are called masks or stencils. Masks for Loop scheme are in Figure 3.6. Loop scheme is an approximative scheme that produces C^1 smooth surface near extraordinary vertices, otherwise it produces C^2 surfaces.

Scheme: For a given triangular control mesh $\mathbf{P_0}$, modified Butterfly interpolation subdivision surface over control mesh $\mathbf{P_0}$ is defined: If $\mathbf{P_i}$ is mesh after i-th step of process, then mesh for next step is $\mathbf{P_{i+1}}$ and is created using following rules:

- For each edge of mesh $\mathbf{P_i}$, create new edge vertex E between end vertices of edge. We have three situations here:

a. Both end vertices are regular. Then we use mask in Figure 3.7.

b. If only one of the end vertices is an extraordinary vertex, then we use mask on the right side of Figure 3.7. We use for creation of the new vertex only the vertices from neighborhood of the extraordinary vertex plus that vertex with weight $\frac{3}{4}$. The coefficients are $s_i = \frac{1}{k}(\frac{1}{4}+cos(\frac{2i\pi}{k})+\frac{1}{2}cos(\frac{4i\pi}{k}))$ for $k > 4$. For k=3, $s_0 = \frac{5}{12}, s_1 = s_2 = -\frac{1}{12}$ and for $k = 4$ is $s_0 = \frac{3}{8}, s_2 = -\frac{1}{8}, s_1 = s_3 = 0$.

c. If both end vertices are extraordinary, we use the mask for both end vertices separately as in case b. and we get the final result E as barycenter of these two computed vertices.

- For each vertex V of mesh $\mathbf{P_i}$, just create a new vertex at the same location as V.
- For each face (triangle) ABC create four new triangles from three vertex points and three edge points.
- All new faces and vertices are parts of new mesh $\mathbf{P_{i+1}}$.

The modified butterfly scheme is an interpolation scheme that produces C^1 surfaces.

3.6 Modelling

For curves, the modelling was simply presented as adding, removing or changing position of control points, changing parameters like degree, knot vector or weights and some ways how to create usable curves like conics. Modelling of surfaces is much wider area that introduces new ways how to create new surfaces from curves. Algorithms presented here are for modelling of NURBS surfaces, because they consist of Bézier surfaces. We know several ways how to accomplish this task.

Definition: Given two curves, a sweep surface is surface that is created by moving beginning of first curve by second curve.

Definition: Given curve, line and angle, a surface of revolution is surface that is created by rotating curve around line by angle.

Definition: Given two curves, a ruled surface is surface that is created by sweeping first curve to second curve along line.

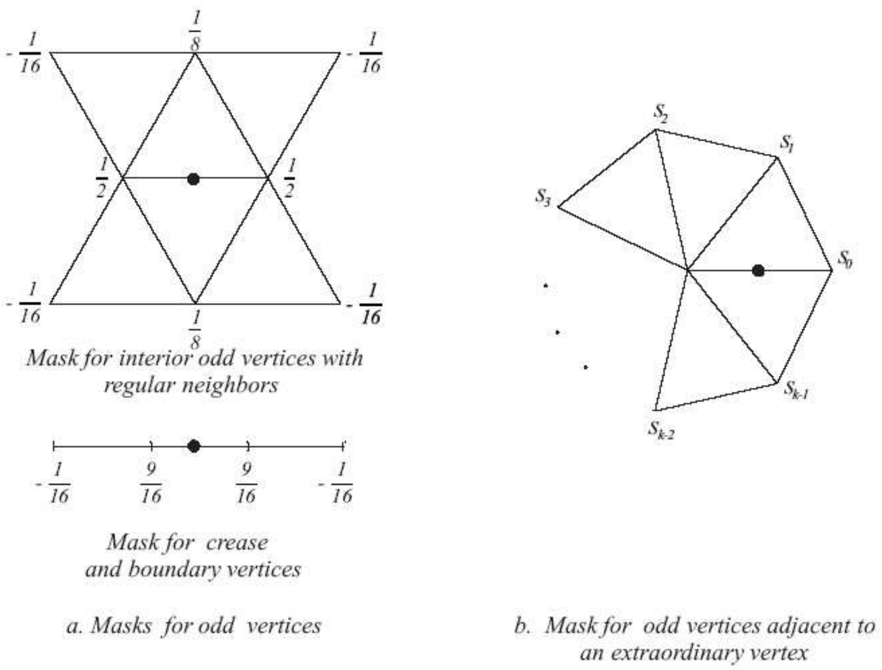
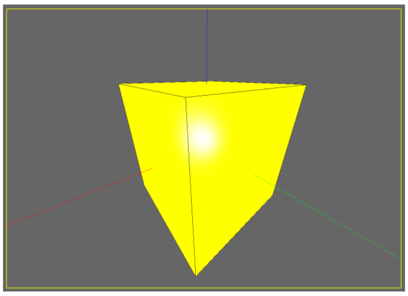
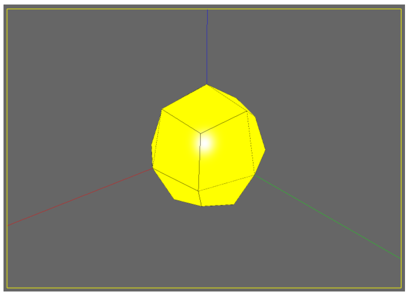


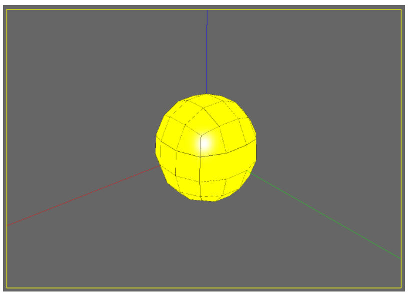
Figure 3.7: Masks for creation of new vertices when performing one step of loop subdivision process. Figure from [21].



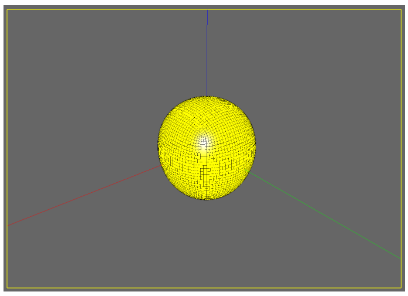
(a) Control mesh - cube.



(b) Mesh after one step of subdivision process.



(c) Mesh after two steps of subdivision process.



(d) Mesh after five steps of subdivision process.

Figure 3.8: Catmull-Clark subdivision process for surface.

When given curves are expressed as a NURBS curves, such modelling techniques can be easily achieved. For example, we can write circular arc in NURBS form and then create the surface of revolution by sweeping curve by that circular arc. Usually, the axis is the z-axis.

Theorem: For given NURBS curve C of degree d_1 with control points $P_i; i = 0, 1, ..., n_1$, weights $w_i; i = 0, 1, ..., n_1$, knot vector $(u_0, u_1, ..., u_{m_1})$ and NURBS curve D with degree d_2, control points $Q_i; i = 0, 1, ..., n_2$, weights $s_i; i = 0, 1, ..., n_2$ and knot vector $(v_0, v_1, ..., v_{m_2})$, the sweep surface that is created by sweeping curve C by curve D is a NURBS surface with degrees (d_1, d_2), control points $P_{i,j} = P_i + Q_j - Q_0$, weights $w_{i,j} = w_i * s_j$; and knot vectors $U = (u_0, u_1, ..., u_{m_1})$ and $V = (v_0, v_1, ..., v_{m_2})$.

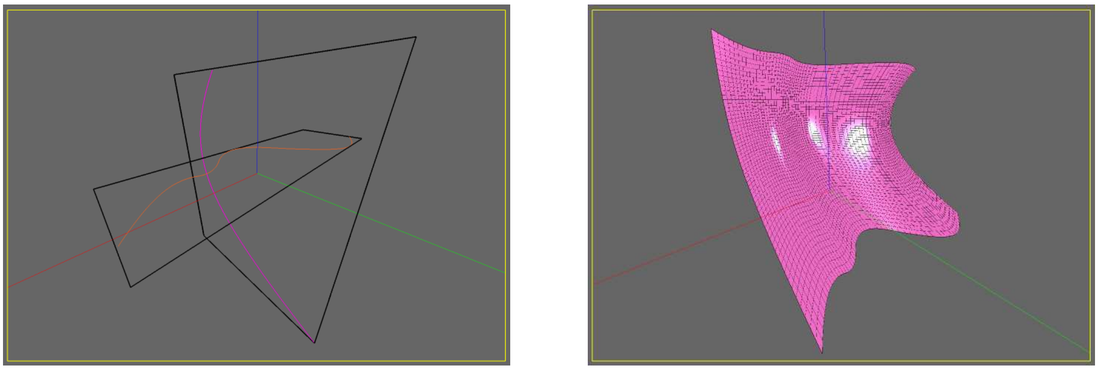
Theorem: Let us have NURBS curve C of degree d with control points $P_i; i = 0, 1, ..., n$, weights $w_i; i = 0, 1, ..., n$ and knot vector $(u_0, u_1, ..., u_m)$. Let NURBS curve A represents circular arc lying in xy plane with sweep angle ϕ, A has degree 2, control points $Q_i; i = 0, 1, ..., p$, weights $s_i; i = 0, 1, ..., p$ and knot vector $(v_0, v_1, ..., v_q)$. Then the surface of revolution that is created by rotation of curve C around z-axis by angle ϕ is NURBS surface with degrees $(d, 2)$, control points $P_{i,j} = P_i + Q_j - Q_0$, weights $w_{i,j} = w_i * s_j$ and knot vectors $U = (u_0, u_1, ..., u_m)$ and $V = (v_0, v_1, ..., v_q)$.

Theorem: Let us have NURBS curves C_1 of degree d with control points $P_i \in E^3; i = 0, 1, ..., n$, weights $w_i \in R; i = 0, 1, ..., n$ and knot vector $(u_0, u_1, ..., u_m)$ and NURBS curve C_2 of degree d with control points $Q_i \in E^3; i = 0, 1, ..., n$, weights $q_i \in R; i = 0, 1, ..., n$ and knot vector $(u_0, u_1, ..., u_m)$. Then the ruled surface created from curves C_1, C_2 can be described as a NURBS surface with degrees $(d, 1)$, control points $P_{i,j}$, weights $w_{i,j}$ and knot vectors $U = (u_0, u_1, ..., u_m)$ and $V = (0, 0, 1, 1)$, where $P_{i,0} = P_i, P_{i,1} = Q_i, w_{i,0} = w_i, w_{i,1} = q_i, i = 0, 1, ..., n$.

Examples of the creation the NURBS sweep surface and the surface of revolution from NURBS curves are in Figure 3.9 and Figure 3.10.

3.7 Visualization

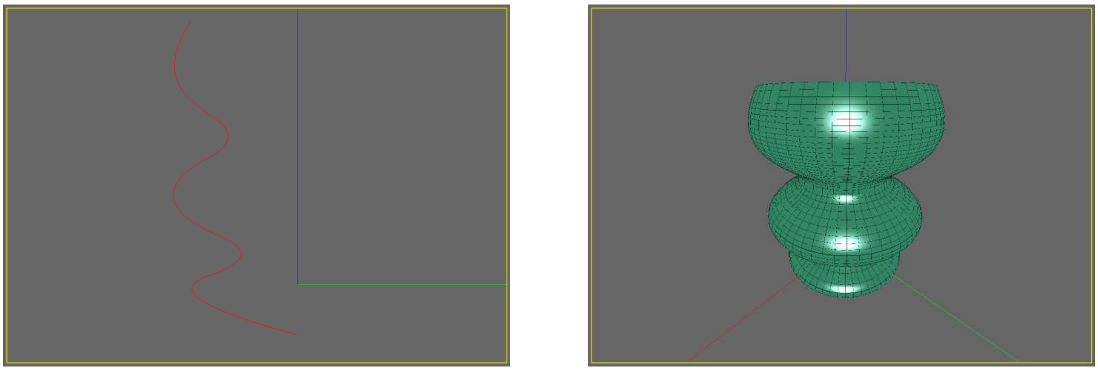
For visualization of continuous entities like parametric curves or surfaces, visualization of their simpler approximation is often used. For curves, this approximation is polygon, for surface it is mesh consisting of triangles or quads. To get such polygonal approximation of curve C, first some representatives $u_0 = a < u_1 < ... < u_n = b$ of parameter from domain of curve $\langle a, b \rangle$ are picked.



(a) Two NURBS curves ready for sweeping.

(b) Sweep surface from previous curves.

Figure 3.9: Creation of sweep surface.



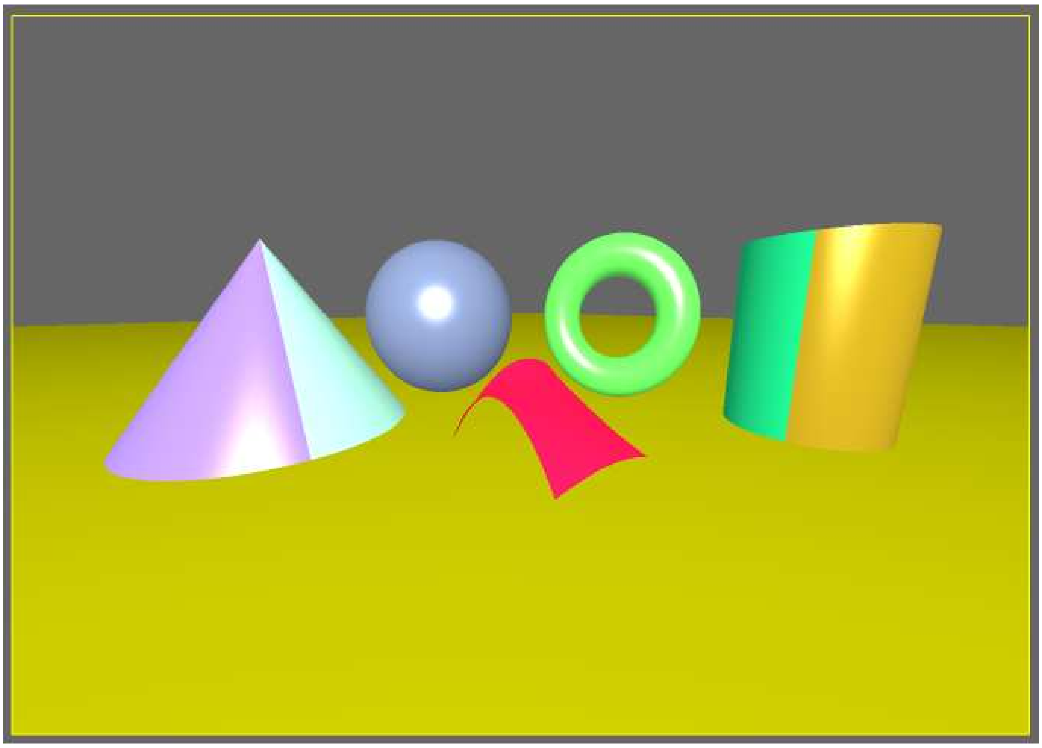
(a) NURBS curve (red) will be rotated arond z-axis (blue).

(b) Resulting surface of revolution.

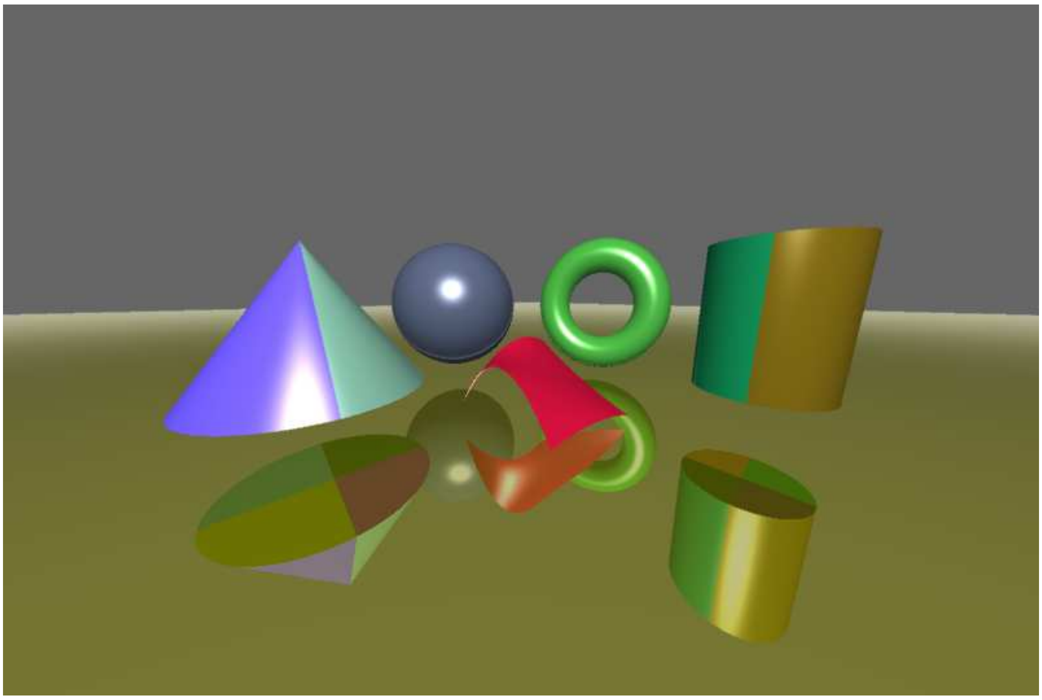
Figure 3.10: Surface of revolution.

These representatives of domain then create polygon $C(u_0)C(u_1)...C(u_n)$, that is approximation of curve C. Bigger n leads to better approximation, also types of picking representatives can influence it (usually representant are picking equidistantly). When we achieve good approximative polygon, that polygon is rendered. For surfaces, representatives are picked for both parameters $u_0 = a < u_1 < ... < u_n = b$;$v_0 = c < v_1 < ... < v_m = d$ from domain $\langle a, b\rangle \times \langle c, d\rangle$. Then quads $S(u_i, v_j)S(u_{i+1}, v_j)S(u_{i+1}, v_{j+1})S(u_i, v_{j+1})$;$i = 0, 1, ..., n-1; j = 0, 1, ..., m-1$ are created and rendered. These quads can be split into two triangles. Another option is to render only border of the quads, we then get the wireframe approximation of surface. Representatives can be picked in a few ways like equidistantly or based on curvature of the surface. Visualization using approximation by mesh is presented in Figure 3.11(a).

Another option how to visualize parametric surface is to use technique called raytracing. Here for each picture element, one ray is created that represents the pixel and intersections of this ray and objects are searched. The elementary algorithm in this technique is to found the intersection of the parametric surface and the ray. For low degree surfaces, the intersection can be found using direct computation. Generally, intersection can be found using numerical methods ([18]), methods that use properties of Bézier patches [11] or by decomposing surface into small Bézier patches and then searching for the intersection between the ray and the control net. This method creates reflections and shadows more precisely, this is shown in Figure 3.11(b).



(a) Visualization using mesh approximation of patches.



(b) Visualization using raytracing.

Figure 3.11: Visualization of scene with several NURBS patches.

Chapter 4

Parametric volumes

In the same way as we extended parametric curves into parametric surfaces, we can also extend parametric surfaces into parametric volumes. Here, we have parameter from three dimensional domain or three real parameters from such a domain. The parametric volumes can be used for representation of objects, where we want also representation of interior of object. Also deformations can be easily performed on these volumes, so we can call them free-form objects [17]. Presented are parametric volumes in tensor product and blending functions form, blending functions will be again only polynomial or rational functions with its piecewise extensions.

Definition: Parametric volume in E^3 is set of points $C = \{X \in E^3; X = f(\mathbf{u}); \mathbf{u} \in U, U \subset R^3\}$, where $f : R^3 \to E^3$ is function describing volume, $\mathbf{u}$ is parameter, U is domain of volume.

Definition: Consider three sets of univariate functions, $\{f_i(u)\}_{i=0}^{n}; \sum_{i=0}^{n} f_i(u) = 1$, $\{g_j(v)\}_{j=0}^{m}; \sum_{j=0}^{m} g_j(v) = 1$ and $\{h_k(w)\}_{k=0}^{l}; \sum_{k=0}^{l} h_k(w) = 1$, with interval domains U,V and W. For given control points $P_{i,j,k} \in E^3; i = 0, 1, ..., n; j = 0, 1, ..., m; k = 0, 1, ..., l$, a volume described by $h(u, v, w) = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=0}^{l} P_{i,j,k} f_i(u) g_j(v) h_k(w)$ is called a tensor product parametric volume with domain $U \times V \times W$.

Definition: Lets have set $U \subset R^3$ and functions $\phi_i(\mathbf{u}); i = 0, 1, ..., n; \mathbf{u} \in U$ such that $\sum_{i=0}^{n} \phi_i(\mathbf{u}) = 1$ for each $\mathbf{u} \in \mathbf{U}$. Then a parametric volume with blending functions ϕ_i is defined as $h(\mathbf{u}) = \sum_{i=0}^{n} P_i \phi_i(\mathbf{u})$. P_i are given points from E^3 called control points.

Tensor product volumes can be written as a sum of parametric curves or surfaces, so the evaluations and properties can be derived from curves and surfaces. Modelling of volumes can be then based on modelling of curves and surfaces so it is possible to create swept volumes, ruled volumes or volumes of revolution.

Volumes are used in geometric modelling, but not with such high intensity like curves or surfaces. But there is a lot of papers about this topic. Parametric volumes are basically described in a few chapters in [4] or [3], general framework of these volumes is presented in [10]. Focus on Bézier volumes were given in [7], [8] and [16], only Bézier tetrahedrons and its properties are described in [14].

4.1 Bézier volumes

We know two types of Bézier volumes derived from Bézier tensor product surfaces and Bézier triangles, Their nets of control points and domains have shape of cuboid resp. tetrahedron. One type is tensor product volume, other uses generalized Bernstein blending functions.

Definition: For given $n, m, l \in N$ and points $P_{i,j,k} \in E^3; i = 0, 1, ..., n; j = 0, 1, ..., m; k = 0, 1, ..., l$, a Bézier tensor product volume $B^{n,m,l}$ is analytically defined as

$$B^{n,m,l}(u, v, w) = \sum_{i=0}^{n} \sum_{j=0}^{m} \sum_{k=0}^{l} P_{i,j,k} B_i^n(u) B_j^m(v) B_k^l(w) \qquad u, v, w \in \langle 0, 1\rangle$$

where n, m, l are degrees in u, v resp. w direction, domain of volume is $\langle 0, 1\rangle \times \langle 0, 1\rangle \times \langle 0, 1\rangle$ and $P_{i,j,k}$ are control points that create control net.

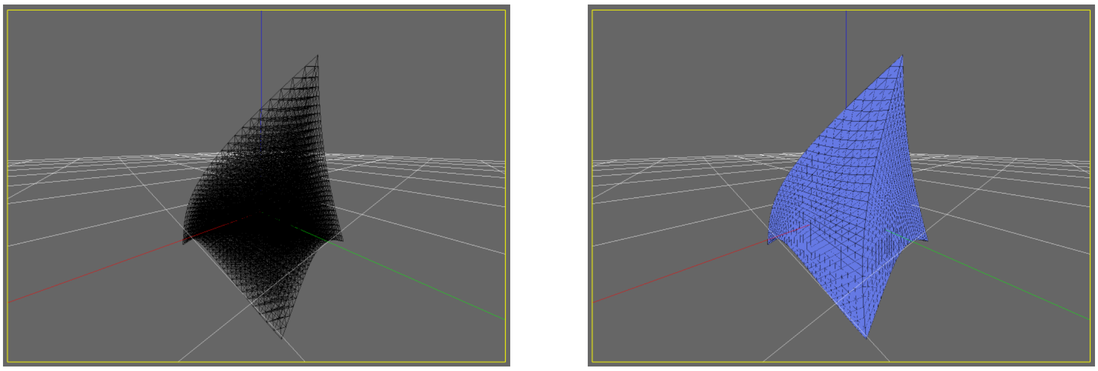
Definition: Let us have degree $n \in N$, points $P_{(i,j,k,l)} \in E^3; i, j, k, j \in Z_0; i + j + k + l = n$, then a Bézier tetrahedron BTH^n is given analytically as

$$BTH^n(\mathbf{u}) = \sum_{|\mathbf{i}|=i+j+k+l=n} P_{\mathbf{i}} \frac{n!}{i!j!k!l!} u^i v^j w^k t^l = \sum_{|\mathbf{i}|=i+j+k+l=n} P_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u})$$

where $\mathbf{u} = (u, v, w, t); 0 \leq u \leq 1; 0 \leq v \leq 1; 0 \leq w \leq 1; 0 \leq t \leq 1; u+v+w+t = 1$. The parameter n is called degree, $P_{\mathbf{i}}$ are control points that create tetrahedral control net, domain is tetrahedron $T = \{(u, v, w); 0 \leq u \leq 1; 0 \leq v \leq 1; 0 \leq w \leq 1; ; u+v+w \leq 1\}$. $B_{\mathbf{i}}^n(\mathbf{u}) = \frac{n!}{i!j!k!l!} u^i v^j w^k t^l$ is called tetravariate Bernstein function.

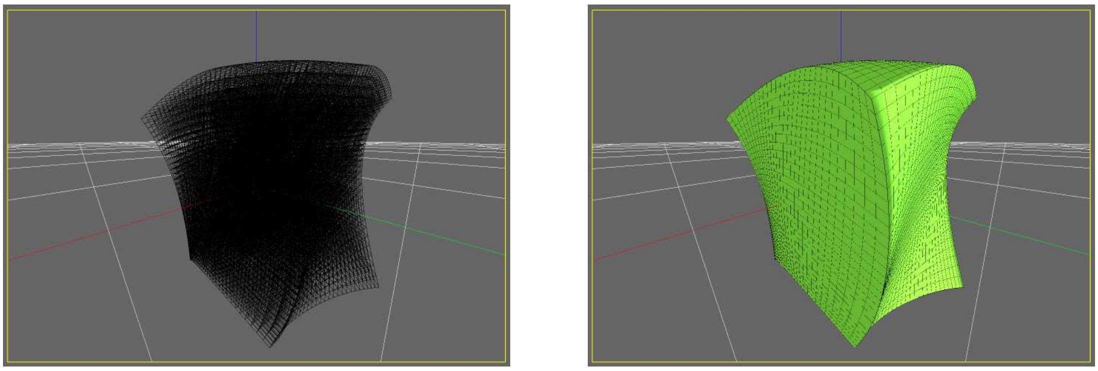
Theorem: Properties of Bézier volume:

- The Bézier volume is invariant under affine transformation
- The Bézier volume approximates shape of its control net.
- The Bézier volume interpolates corner points of its control net. For Bézier tensor product volume, it interpolates control points $V_{0,0,0}, V_{n,0,0}, V_{0,m,0}, V_{n,m,0}, V_{0,0,l}$, $V_{n,0,l}, V_{0,m,l}, V_{n,m,l}$, for Bézier tetrahedron, it interpolates control points $V_{(n,0,0,0)}, V_{(0,n,0,0)}, V_{(0,0,n,0)}, V_{(0,0,0,n)}$.



(a) Wireframe visualization. (b) Wireframe and solid visualization.

Figure 4.1: Bézier tetrahedron.



(a) Wireframe visualization. (b) Wireframe and solid visualization.

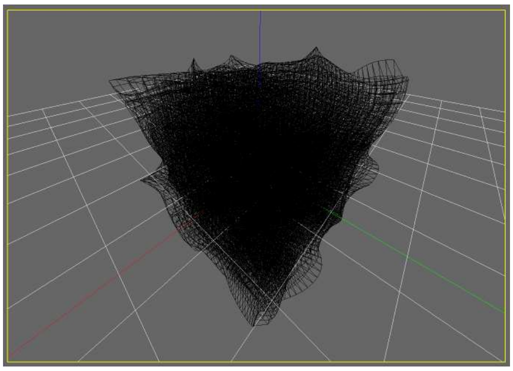
Figure 4.2: Bézier tensor product volume.

- Boundary patches are Bézier surfaces, boundary curves are Bézier curves.
- It is possible to elevate degree of Bézier tetrahedron or elevate degrees of Bézier tensor product volume. For the Bézier tensor product volume, the degree can be elevated independently in each direction like in the curve case. The degree elevation of the Bézier tetrahedron is based on equation $\sum_{|\mathbf{i}|=n} P_{\mathbf{i}} B_{\mathbf{i}}^{n}(\mathbf{u}) = \sum_{|\mathbf{i}|=n+1} Q_{\mathbf{i}} B_{\mathbf{i}}^{n+1}(\mathbf{u})$, where

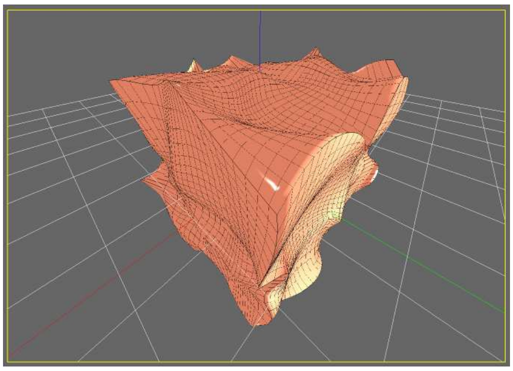
$$Q_{\mathbf{i}} = Q_{(i,j,k,l)} = \frac{iP_{(i-1,j,k,l)} + jP_{(i,j-1,k,l)} + kP_{(i,j,k-1,l)} + lP_{(i,j,k,l-1)}}{n+1}$$

  $i + j + k + l = n + 1$.

The basic visualizations of Bézier volumes and its control points are in Figure 4.1 and Figure 4.2.



(a) Wireframe visualization.



(b) Wireframe and solid visualization.

Figure 4.3: B-spline volume with degrees $(3, 3, 3)$ and $8 \times 8 \times 8$ control points.

4.2 B-spline volumes

From the Bézier form of polynomial volumes, we can move to the piecewise polynomial volumes. We focus on the volumes in B-spline form based on B-spline basic functions. It is defined with all attributes as B-spline curve, here we have three parameters u, v, w, definition is similar to case for curves and surfaces.

Definition: Let us have real numbers $u_0 \leq u_1 \leq ... \leq u_{m_u}$, $v_0 \leq v_1 \leq ... \leq v_{m_v}$, $w_0 \leq w_1 \leq ... \leq w_{m_w}$ degrees $d_u, d_v, d_w \in N; d_u, d_v, d_w \geq 1$ and control points $P_{i,j,k} \in E^3; i = 0, 1, ..., n_u; j = 0, 1, ..., n_v; k = 0, 1, ..., n_w$, where $n_u = m_u - d_u - 1, n_v = m_v - d_v - 1, n_w = m_w - d_w - 1$. Then B-spline volume N^{d_u,d_v,d_w} of degrees d_u, d_v, d_w over knot vectors $(u_0, u_1, ..., u_{m_u})$, $(v_0, v_1, ..., v_{m_v})$ and $(w_0, w_1, ..., w_{m_w})$ is defined over interval $\langle u_{d_u}, u_{n_u+1}) \times \langle v_{d_v}, v_{n_v+1}) \times \langle w_{d_w}, w_{n_w+1})$ as

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

$$u \in \langle u_{d_u}, u_{n_u+1}), v \in \langle v_{d_v}, v_{n_v+1}), w \in \langle w_{d_w}, w_{n_w+1})$$

where $N_i^{d_u}(u), N_j^{d_v}(v), N_k^{d_w}(w)$ are B-spline blending functions.

The properties of B-spline volumes are similar to the properties of B-spline curves and surfaces. B-spline volume over randomly picked control points can be seen in Figure 4.3.

4.3 Rational volumes

Also parametric polynomial volumes can be extended into rational and piecewise rational volumes. This give us opportunity to represent and model a wider set of volumes. New degrees of freedom are added by introducing real number called weights, for each control point one weight

Definition: Assume given points $P_{i,j,k} \in E^3; i = 0, 1, ..., n; j = 0, 1, ..., m, k = 0, 1, ..., l$ and real numbers $w_{i,j,k} \in R; w_{i,j,k} \geq 0; i = 0, 1, ..., n; j = 0, 1, ..., m, k = 0, 1, ..., l$, we define rational Bézier tensor product volume $RB^{n,m,l}$ analytically as

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

$$u, v, w \in \langle 0, 1 \rangle$$

where n, m, l are degrees in u resp. v resp. w direction, $\langle 0, 1 \rangle \times \langle 0, 1 \rangle \times \langle 0, 1 \rangle$ is domain of volume, $P_{i,j,k}$ are control points and $w_{i,j,k}$ are weights.

Definition: Let us have degree $n \in N$, points $P_{\mathbf{i}} \in E^3; \mathbf{i} = (i, j, k, l); i, j, k, l \in N_0; i + j + k + l = n$ and real numbers $w_{\mathbf{i}} \in E^3; w_{\mathbf{i}} \geq 0; \mathbf{i} = (i, j, k, l); i, j, k, l \in N_0; i + j + k + l = n$. Not all $w_{\mathbf{i}}$ can be equal to 0. Then the rational Bézier tetrahedra RBT^n is given analytically as

$$RBT^n(\mathbf{u}) = \frac{\sum_{|\mathbf{i}|=i+j+k+l=n} w_{\mathbf{i}} P_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u})}{\sum_{|\mathbf{i}|=i+j+k+l=n} w_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u})}$$

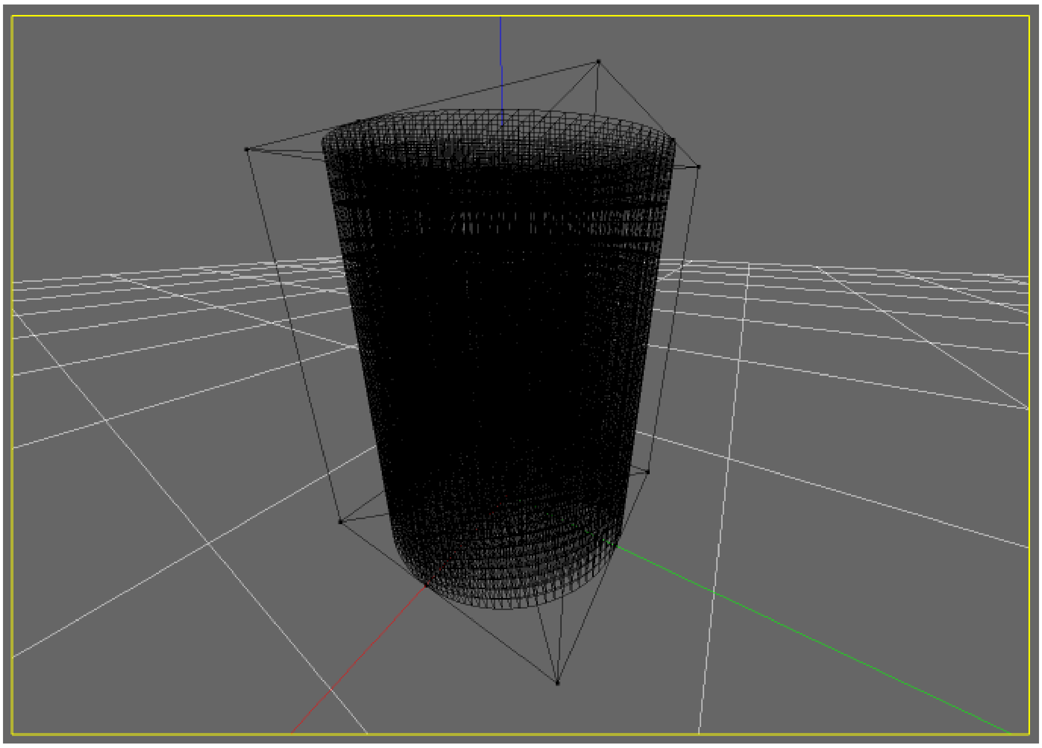
where $\mathbf{u} = (u, v, w, t); 0 \leq u, v, w, t \leq 1; u + v + w + t = 1$. Parameter n is called degree, $P_{\mathbf{i}}$ are control points, $w_{\mathbf{i}}$ are weights, and tetrahedron $T = \{(u, v, w); 0 \leq u, v, w \leq 1; u + v + w \leq 1\}$ is domain of the volume RBT^n.

Definition: Let us have real numbers $u_0 \leq u_1 \leq ... \leq u_{m_u}$, $v_0 \leq v_1 \leq ... \leq v_{m_v}$ and $w_0 \leq w_1 \leq ... \leq w_{m_w}$, degrees $d_u, d_v, d_w \in N$ and control points $P_{i,j,k} \in E^3; i = 0, 1, ..., n_u; j = 0, 1, ..., n_v; k = 0, 1, ..., n_w$, real numbers $w_{i,j,k} \in R; i = 0, 1, ..., n_u; j = 0, 1, ..., n_v; k = 0, 1, ..., n_w$, where $n_u = m_u - d_u - 1, n_v = m_v - d_v - 1, n_w = m_w - d_w - 1$. Not all $w_{(i,j,k)}$ can be equal to 0. Then the rational B-spline volume (NURBS volume) RN^{d_u,d_v,d_w} of degrees d_u, d_v, d_w over knot vectors $(u_0, u_1, ..., u_{m_u})$, $(v_0, v_1, ..., v_{m_v})$ and $(w_0, w_1, ..., w_{m_w})$ is defined over interval $\langle u_{d_u}, u_{n_u+1}) \times \langle v_{d_v}, v_{n_v+1}) \times \langle w_{d_w}, w_{n_w+1})$ as

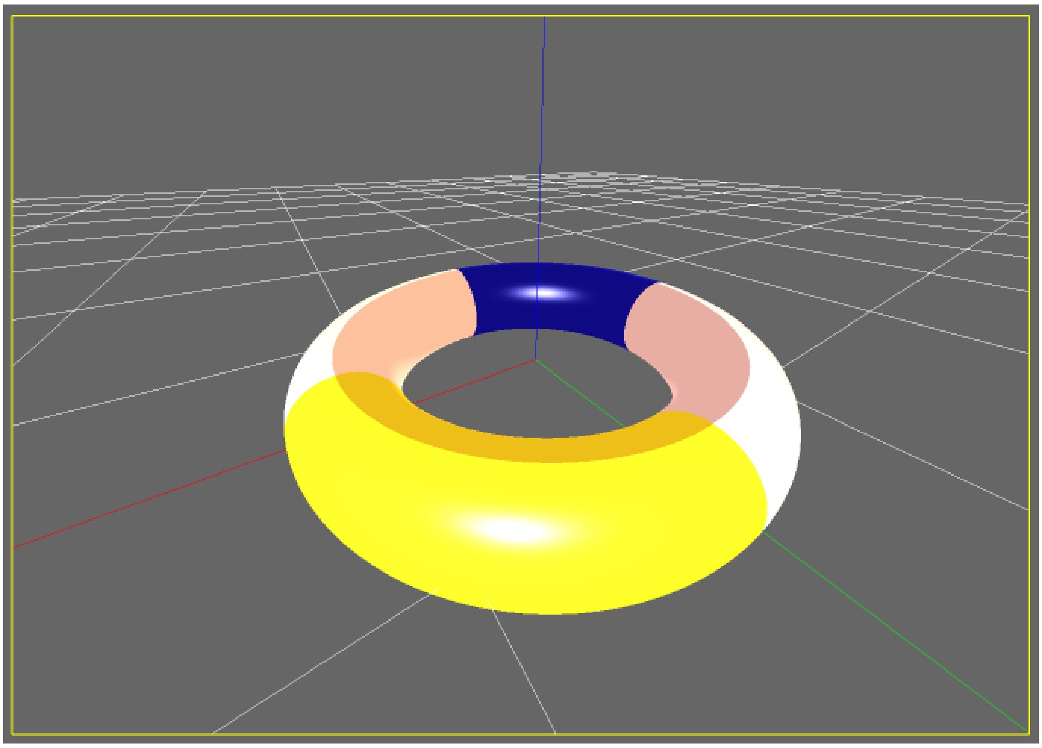
$$RN^{d_u,d_v,d_v}(u, v, w) = \frac{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \sum_{k=0}^{n_w} w_{i,j,k} P_{i,j,k} N_i^{d_u}(u) N_j^{d_v}(v) N_k^{d_w}(w)}{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \sum_{k=0}^{n_w} w_{i,j,k} N_i^{d_u}(u) N_j^{d_v}(v) N_k^{d_w}(w)}$$

$$u \in \langle u_{d_u}, u_{n_u+1}), v \in \langle v_{d_v}, v_{n_v+1}), w \in \langle w_{d_w}, w_{n_w+1})$$

Using these rational volumes, we can describe some useful objects, this is illustrated in Figure 4.4.



(a) Cylinder in wireframe.



(b) Torus decomposed to rational Bézier volumes.

Figure 4.4: Rational B-spline volumes (NURBS volumes).

Chapter 5

Project of Dissertation

In the fourth chapter, we give a basic introduction to parametric volumes in Bézier and B-spline form. Definitions and basic properties are given . For better understanding and better usability of these volumes, more properties, techniques, tools and algorithms must be described. In the next section are presented parts that we want to extend. These parts come from observation of similar methods in the case of curves and surfaces.

5.1 Modelling of volumes

The aims in this section are straightforward. We want to create and describe techniques to create ruled volumes, swept volumes or volumes of revolution from curves and surfaces. Descriptions of filled conics are needed. Another goal in this section will be the computation or approximation of intersections between volumes or between volumes and other objects like lines, planes, surfaces. Trimmed volumes can help with this problem.

5.2 Visualization of volumes

We will present many ways of volumes visualization. We want to improve basic wireframe and solid visualization based on approximation of volume by net of points. Other types of visualization can be done using parametric isocurves or isosurfaces or sections of volume by planes or surfaces. Also the raytracing technique can be used here, we need to intersect ray with volume, this intersection can be found using numerical (Newton) or geometrical (Bézier clipping) methods. We also want to focus on usability of recent GPU advancements on better visualization of volumes.

5.3 Subdivision volumes

In the last time, there are papers about performing subdivision algorithms on three dimensional meshes. Significant works in this area are [9] or [2]. We plan to implement them and also we will try to extend and implement additional schemes from surfaces to solids represented as tetrahedral meshes or other types of 3D object discrete representation. We will start with observation of subdivision surfaces inserted in 2D space. Then we will try to extend subdivision rules for volumes (3D varieties) in 4D space and then insert them in 3D space.

5.4 Trimmed volumes

Our plan is also to extend class of B-spline and Bézier volumes by trimming domain of such volume. This will help with some modelling purposes, especially with Boolean operations on volumes. The exact type of trimming objects in domain space of volume must be evaluated. Then visualization of such trimmed volumes must be done.

5.5 Conversion to another representations

There are many used representations of three dimensional objects. To make our 3D object representation useful, we must present also conversion algorithm between these representations and our parametric volume representation, specially boundary and discrete volume representations.

5.6 GeomForge system

To illustrate the solutions described in final dissertation work, we started with design and implementation of our own complex system for creating, modelling and visualization of curves, surfaces and volumes. This system has now implemented many basic numerical, geometric and visualization algorithms. Also all topics that will be covered in final thesis will be implemented in this system. For visualization purposes, OpenGL geometric library is used [19]. All figures of curves, surfaces and solids in this work were created using an early version of this system.

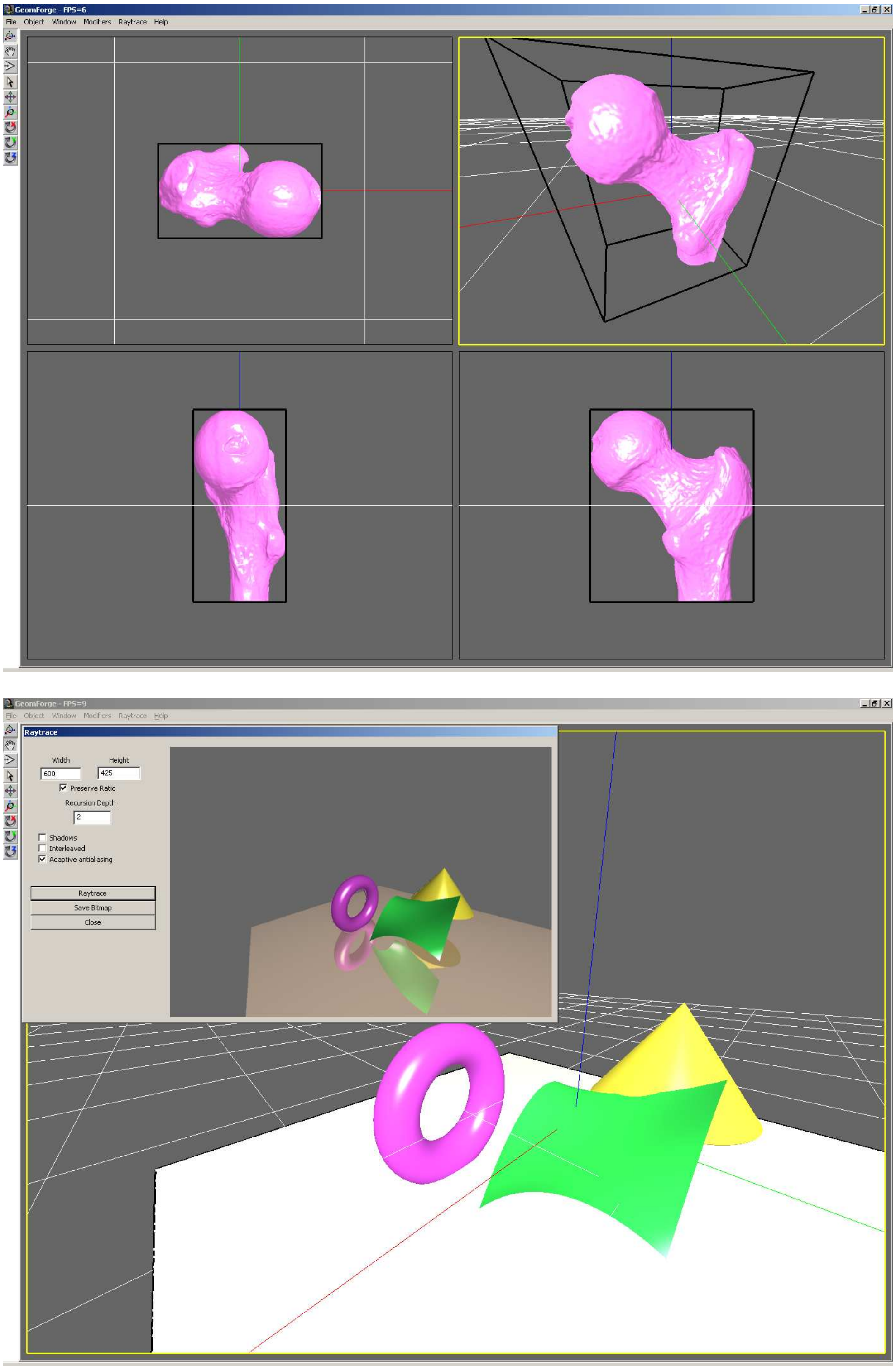


Figure 5.1: Screenshots of GeomForge system.

Bibliography

[1] Tomas Akenine-Möller and Eric Haines. *Real-Time Rendering.* A.K. Peters, Natick, MA, USA, second edition, 2002.

[2] Chandrajit L. Bajaj, Scott Schaefer, Joe D. Warren, and Guoliang Xu. A subdivision scheme for hexahedral meshes. *The Visual Computer*, 18(5-6):343–356, 2002.

[3] Min Chen, Arie Kaufman, and Roni Yagel, editors. *Volume Graphics.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[4] Elaine Cohen, Richard F. Riesenfeld, and Gershon Elber. *Geometric Modeling with Splines.* A.K. Peters, Natick, MA, USA, 2001.

[5] Gerald Farin. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide.* Academic Press, New York, NY, USA, third edition, 1993.

[6] Ron Goldman. Introduction to geometric modelling, *http //www.msri.org/local/sottile_gm/rag_gm_talks/goldman.pdf*.

[7] Dieter Lasser. Bernstein - Bézier representation of volumes. *Computer Aided Geometric Design*, 2(1-3):145–149, 1985.

[8] Dieter Lasser. Visualization of free-form volumes. In *IEEE Visualization*, pages 379–387, 1990.

[9] 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.

[10] William Martin and Elaine Cohen. Representation and extraction of volumetric attributes using trivariate splines: a mathematical framework. In *SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 234–240, New York, NY, USA, 2001. ACM Press.

[11] Tomoyuki Nishita, Thomas W. Sederberg, and Masanori Kakimoto. Ray tracing trimmed rational surface patches. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 337–345, New York, NY, USA, 1990. ACM Press.

[12] Les Piegl and Wayne Tiller. *The NURBS book*. Springer-Verlag, London, UK, 1995.

[13] Aristides G. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Comput. Surv.*, 12(4):437–464, 1980.

[14] Martin Samuelcik. Bézier tetrahedras in e^4 and its applications. (in slovak). Master's thesis, FMPI, Comenius University, Bratislava, 2002.

[15] Martin Samuelcik. Basic subdivision schemes on triangular meshes. In *Symposium on Computer Geometry*, pages 113–118, Kocovce, Slovakia, 2004.

[16] Martin Samuelcik. Modeling with rational bézier solids. (poster). In *WSCG 2005*, pages 67–68, Plzen, Czech Republic, 2005.

[17] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, New York, NY, USA, 1986. ACM Press.

[18] Daniel L. Toth. On ray tracing parametric surfaces. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 171–179, New York, NY, USA, 1985. ACM Press.

[19] Various. OpenGL homepage, *http://www.opengl.org*.

[20] Alex Vlachos, Jörg Peters, Chas Boyd, and Jason L. Mitchell. Curved PN triangles. In *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 159–166, New York, NY, USA, 2001. ACM Press.

[21] Denis Zorin and Peter Schröder. Subdivision for Modeling and Animation. Technical report, SIGGRAPH 2000, 2000. Course Notes.