Applied geometric modelling

Daniel G. Razafimandimby

Master of Technology, 5 Data/IT,
Narvik University College, Narvik, Norway

December 15, 2014

Abstract

1 Introduction

1.1 GMlib

GMlib is an open source geometric modelling library developed and maintained at Narvik University College, NUC, by the college's teachers and students. The library contains several modules for creating and visualizing parametrized objects, such as curves, surfaces and volumes.

2 Method

2.1 Bezier and the evaluator

A bezier curve is a parametrized curve with knot vectors which range from 0 to 1 and can be defined for any degree n. Higher dimensions of bezier curves are called bezier surfaces and they are defined by a set of control points and the position of a point P is is given by the parametric coordinates u and v:

$$P(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u) B_j^m(v) k_{i,j} \tag{1}$$

There are three different types of evaluators used for Bezier curves: de Casteljau algorithm, a Cox/de'Boor algorithm and computing the Bernstein polynomials directly [1]. In this project, I have used a Cox/deBor algorithm: computing from the left. The algorithm starts by filling in the matrix with the Bernstein polynomial between first and the requested degree.

Listing 1: Assigning the top half of the matrix.

```
//Matrix is the incoming matrix, degree is the
//Bernstein polynomial degree, t is the parameter
//value 0 <= t <= 1 and scaling factor delta
function compute basis matrix(matrix, degree, t, delta)
  matrix[degree-1][0] = 1 - t
  matrix[degree-1][1] = 1
  for(i = rows - 2; i >= 0; i--)
    matrix[i][0] = (1 - t) matrix[i+1][0]
    for(j = 1; j < degree - i; j++)
      matrix[i][j] = t * matrix[i+1][j-1] + (1-t)
                   * matrix[i+1][j]
    end for
    matrix[i][degree-i] = t * matrix[i+1][degree-i-1]
  end for
```

Following this every row except the first one is multiplied with the delta and derivatives. Because the Bezier surface ranges from 0 to 1, a scaling parameter is introduced and multiplied into every row except the first.

Listing 2: Assigning the bottom half of the matrix

```
  matrix[degree][0] = -delta
  matrix[degree][1] = delta
  for(k = 2; k <= degree; k++)
    scale = k * delta
    for(i = degree; i > degree-k; i--)
      matrix[i][k] = scale * matrix[i][k-1]
      for(j = 1; j < degree - i; j++)
        matrix[i][j] = scale * (matrix[i][j-1]
                     - matrix[i][j])
      end for
      matrix[i][0] = -scale * matrix[i][0]
    end for
  end for
end function
```

2.2 B-spline

A b-spline is a curve which has knot definitions ranging from 0 to n.

$$c(t) = \begin{pmatrix} 1 - w_{1,i}(t) & w_{1,i}(t) \end{pmatrix} \begin{pmatrix} 1 - w_{2,i}(t) & w_{2,i}(t) & 0 \\ 0 & 1 - w_{2,i}(t) & w_{2,i}(t) \end{pmatrix} \begin{pmatrix} 1 - w_{3,i}(t) & w_{3,i}(t) & 0 & 0 \\ 0 & 1 - w_{3,i}(t) & w_{3,i}(t) & 0 \\ 0 & 0 & 1 - w_{3,i}(t) & w_{3,i}(t) \end{pmatrix} \begin{pmatrix} c_{i-3} \\ c_{i-2} \\ c_{i-1} \\ c_i \end{pmatrix} \quad (2)$$

Formula 2 shows the formula for a b-spline [1]. The evaluator for a B-spline is very similar to the Bezier algorithm, however a B-spline has multiple interior knots which makes it necessary to find the corresponding knot-index to t. Therefore instead of using t in the matrix, we introduce a function w which maps the value between 0 and 1. This function is dependant on the knot index, knot value and the Berstein polynomial degree.

2.3 Expressional rational B spline (ERBS), and ERBS surfaces

An expressional rational B-spline, ERBS, is a new type of B-spline, purposed by Arne Lakså, Børre Bang and Lubomir T. Dechevsky (Dechevsky, Lakså, Bang, 2005)(http://gtwavelet.bme.gatech.edu/wp/ExpoSplines.pdf). The concept is shortly explained in their article, Exploring Expo-Rational B-splines for Curves and Surfaces (Dechevsky, Lakså, Bang, SOME YEAR). This was created to cover some of the limitations found in regular B-splines.

An ERBS surface is a surface which consists of an amount of local patches which are created by either using bezier surfaces or sub surfaces. These local patches are created by deciding the size of the grid and if the surface is closed in u- and/or v-direction. This information is then used to construct the knot-vectors which will be the basis for the surface.

2.4 Tesselation

Tesselation is a technique where we till a plane into different shapes. It is not permitted to have overlapping edges or gaps between the lines which define the shape. Voronoi and Delaunay are two very common methods of doing this.

3 Observations

3.1 Help classes and containers

3.1.1 MyKnotVector

MyKnotVector is a simple data container which contains the information needed to create a knot vector.

3.1.2 Animation

Animation is a helper class which performs the animations of surfaces. The various types of animations are seperated into individual classes which inherit from a common class.

3.2 MySubsurface

3.3 MyBezierSurface

3.4 MyERBSSurface

MyERBSSurface is capable of receiving a surface to copy and convert it to an ERBS surface. It can currently use either subsurfaces or bezier surfaces to generate the surface.

3.5 My animation/effect

4 Discussion

4.1 The performance of MyERBSSurface

5 Conclusion

//

References

[1] A. Lakså. *Basic properties of Expo-Rational B-splines and practical use in Computer Aided Geometric Design.* 2007.