Applied Geometry and Special Effects (DTE-3604-1)

J. H. Klæboe (jkl019)

UiT - The Arctic University of Norway, P.O. Box 385, N-8505 Narvik, Norway

Submitted 2022-12-10

Abstract

This paper is about the work done throughout the project given in the course *Applied Geometry and Special Effects*. The project has revolved around implementing different geometric objects where the focus has been on B-spline, closed subdivision curve and Blending spline curve/surfaces. The paper goes through the theory of the objects and how the geometry has been implemented and shows the resulting geometry made.

Keywords: Geometric modelling; B-splines; Blending spline; GMlib;

1 Introduction

This paper is about the work done in the course *Applied Geometry and Special Effects*. Throughout the course we have implemented various geometric objects. The main focus of the project was to use affine transformation of local curves to dynamically change the shape of the curves.

In section 2 we will go through the implementation of the different objects that has been implemented. Further in section 3 the resulting objects created is illustrated in the form of screenshots of the simulation. Finally, in section 4 we will give an assessment of the results and a brief conclusion.

Further we will go through some of the theory for the geometric objects.

1.1 B-spline

The first geometric object that was implemented was the B-spline curve. B-splines consists of a set of basis functions which are connected to one point. Together they form a spline space which is defined by a knot vector and the polynomial degree (Lakså, 2022, p. 75).

1.2 Subdivision curve

Subdivision is a form of corner-cutting where a set of points is replaced with larger set of points. The new points is placed between the previous points (Lakså, 2022, p. 106).

1.3 Blending spline curve and surface

The Blending spline curve/surface is built upon the same principall as the B-spline. However, instead of using control points to create the objects, the Blending spline curve is created using local curves and the Blending spline surface is created using local surfaces.

1.4 Related work

Implementation of the geometric objects is based on the math and algorithms from Arne Lakså's book, Blending techniques in Curve and Surface constructions (Lakså, 2022).

2 Method

The implementation of the curves and geometry are done using the programming language C++ and the geometric modelling library GMlib (Lakså et al., 2022). The repository with the source code with the implementation of the different objects can be found at (Klæboe, 2022).

2.1 B-spline

The B-spline was implemented with two different methods. In the first method the B-spline was created by using control points and in the second the B-spline was made by sampling points along a curve and using least square method to create the control points. We have used polynomial degree 2 for both methods. Further we will describe the two different implementation of the B-spline.

2.1.1 B-spline with control points

In the first implementation of the B-spline curve we used 6 control points. The size of the knot vector then becomes $n+k$ where n is the number of control points and k is the order. The order is defined by the *degree + 1*, meaning we get order 3. The number of basis functions at each interval corresponds to the order, k.

When the knot vector has been calculated we can use the formula 1 for the B-spline curve from Laksås book (Lakså, 2022, p. 75).

$$c(t) = \sum_{i=0}^{n-1} c_i b_{k,i}(t) \tag{1}$$

Where c_i is the control point at the given interval in the knot vector and $b_{k,i}(t)$ is the basis functions. Since the order of our B-spline is 3, we get 3 basis functions at each interval which are found by equation 2.

$$\begin{aligned} b_{k,i} &= w_{1,i}(t)w_{2,i}(t) \\ b_{k-1,i} &= (1 - w_{1,i}(t)) * w_{2,i-1}(t) + w_{1,i}(t) * (1 - w_{2,i}(t)) \\ b_{k-2,i} &= (1 - w_{1,i}(t)) * (1 - w_{2,i-1}(t)) \end{aligned} \tag{2}$$

The $w_{d,i}(t)$ is a scaling function which maps the parameter t to a value in the in the range of the knot vector.

2.1.2 B-spline with least square method

In the second method we are creating a B-spline by using the *least square method*. The constructor created for this method takes a set of points p and the number of control points to be made n as input. Then we create a matrix A with dimension m x n, where m is the number of points given. Least square approximation values of the curve are then added to A. In order to make the control points we follow the steps in equation 3 and end up with a vector with the control points c.

$$\begin{aligned} &1. b = A^T p \\ &2. B = A^T A \\ &3. c = B^{-1} b \end{aligned} \tag{3}$$

2.2 Subdivision curve

There exist different algorithms for subdivision. We have implemented Lane-Riesenfeld's closed subdivision algorithm. The algorithm is divided into two parts, where the first part is doubling the points and the second part is smoothing the curve.

The constructor takes the set of points as input and sets the number of intervals n to be the amount of points. When *sample* is called with the parameters m and d the new points get generated and the smoothing of the curve is done. The parameters represent the number of refinements m and d represent the degree of the curve. The implementation is based on the Lane-Riesenfeld's closed algorithm described in Laksâ's book (Laksâ, 2022, p. 112).

2.3 Model curve

The model curve for the project was chosen based on its look and being symmetric. The curve is in the domain $t \in [0, 2\pi]$ and the parametric function for the curve can be observed in equation 4.

$$\begin{aligned} x &= r * cos(nt) * cos(t) \\ y &= r * cos(nt) * sin(t) \\ z &= r * sin(nt) \end{aligned} \tag{4}$$

The parameters are set to be $r = 1$ and $n = 4$.

2.4 Blending Spline curve

The constructor for the Blending Spline curve takes three parameters as input. The chosen model curve *mc*, the number of local curves to be made n and the degree d. In order to blend the curves, there is added a *B-function*. The chosen *B-function* for the Blending Spline is a trigonometric *B-function* of order 2 which can be observed in equation 5.

$$B_2(t) = t - \frac{1}{2\pi} sin(2\pi t) \tag{5}$$

2.4.1 Affine transformation

The implementation of the animation for the Blending Spline curve is done in *localSimulate*-function. This takes the variable *dt* as an input which is further used to *translate* and *rotate* the local curves. In addition, there is a variable *_dt* which alternates between 1 and -1 and is used to *translate* the curves.

2.5 Blending Spline surface

Blending Spline surface is similar to the Blending Spline curve. However, instead of creating local curves there is generated local surfaces. Meaning that we are operating in two dimensions and the knot vector t is replaced with $\_tU$ and $\_tV$ for the two directions. The methods from the Blending Spline curve has been modified to take the two dimensions into account. Additionally, there is implemented different versions of the knot vector, depending on if the surface is closed or open in the two directions. In the *eval*-function for the Blending Spline surface we calculate the derivatives, thus the amount of computation is more than for Blending Spline curve. The general formula for the Blending Spline surface found in Laksâ's book (Laksâ, 2022, p. 221) can be observed in equation 6.

$$S(u,v) = \sum_{j=0}^{n_v-1} \sum_{i=0}^{n_u-1} s_{i,j}(u,v) B(w_{1,i}(u)) B(w_{1,j}(v)) \tag{6}$$

Which can be formulated into equation 7.

$$S(u,v) = (1 - B_i(u) B_i(u)) \begin{pmatrix} s_{i-1,j-1}(u,v) & s_{i-1,j}(u,v) \\ s_{i,j-1}(u,v) & s_{i,j}(u,v) \end{pmatrix} (1 - B_j(v) B_j(v)) \tag{7}$$

Where $s_{i-1,j-1}(u,v)$ is the point, $s_{i-1,j}(u,v)$ is the derivative with respect to u, $s_{i,j-1}(u,v)$ is the derivative with respect to v and $s_{i,j}(u,v)$ is both derivatives.

3 Results

Results of the implemented geometric objects are further shown in the form of screenshots taken from the simulation. B-spline created by a set of given control points can be observed in figure 1 at the left side and a the one generated by creating the control points with the *least square method* on the right. The latter one shows the B-spline in red and the circle to be approximated in blue. In figure 2 the result of the closed subdivision curve can be observed with the initial points the constructor is getting as input. The chosen model curve and the Blending Spline's copy of the curve can be seen in figure 3. The Blending Spline surface it was created three different surfaces. A plane which is shown in figure 4, a torus shown in figure 5 and a cylinder figure 6. All three figures shows the surfaces before and after the transformation of the local surfaces.

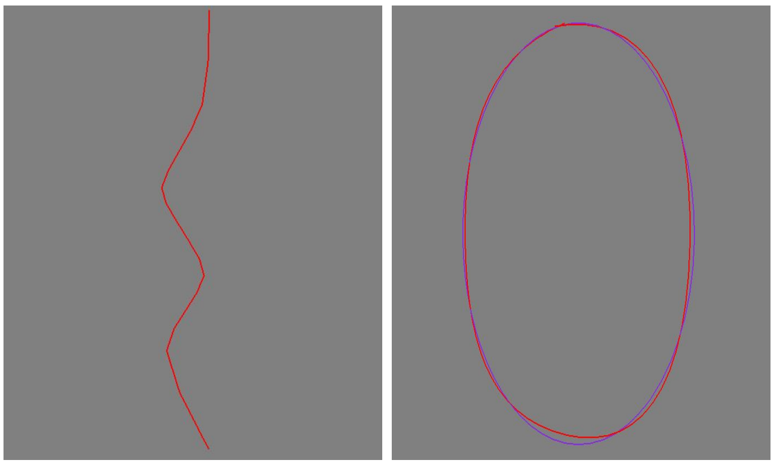

Figure 1. Illustration of B-spline created by using control points (left) and by using *least square method*(right)

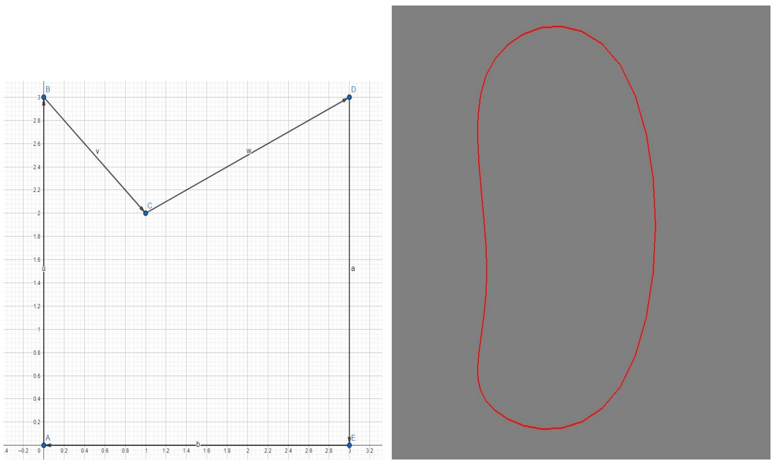

Figure 2. Illustration of the closed subdivision curve (right) with the initial points (left)

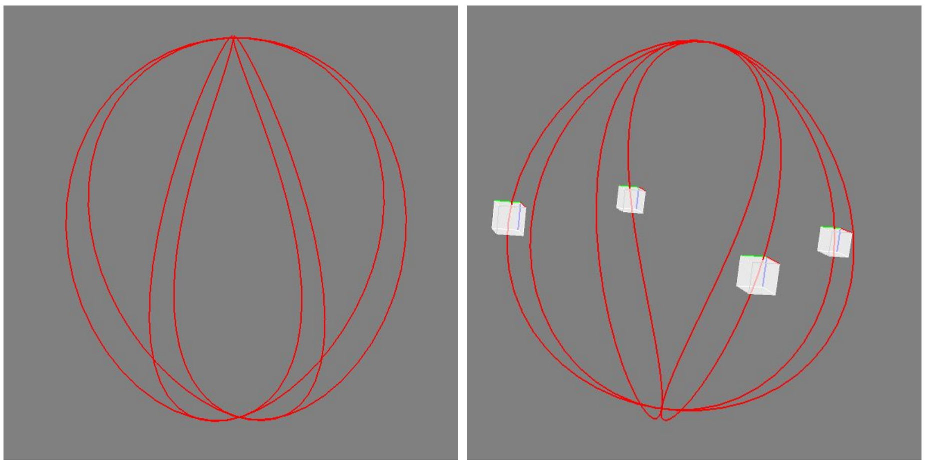

Figure 3. Illustration of the chosen model curve (left) and copied by the Blending Spline curve (right)

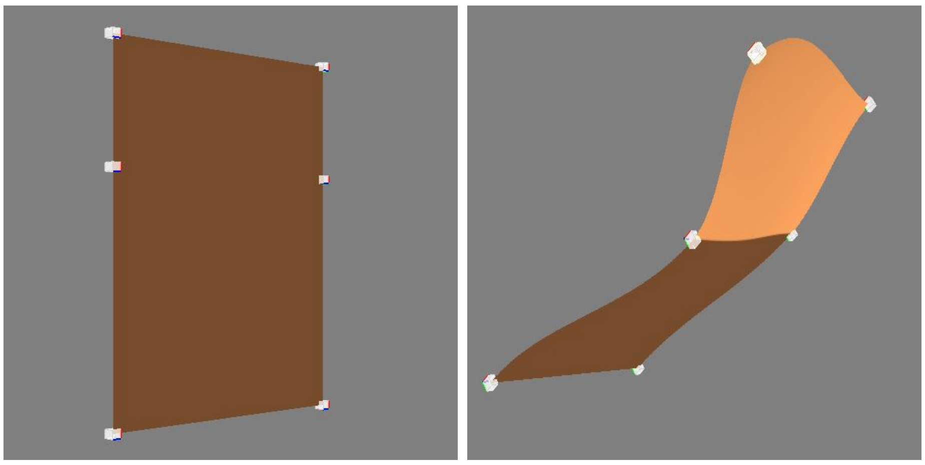

Figure 4. Illustration of the Blending Spline surface with a plane surface before (left) and after (right) it has been tranformed

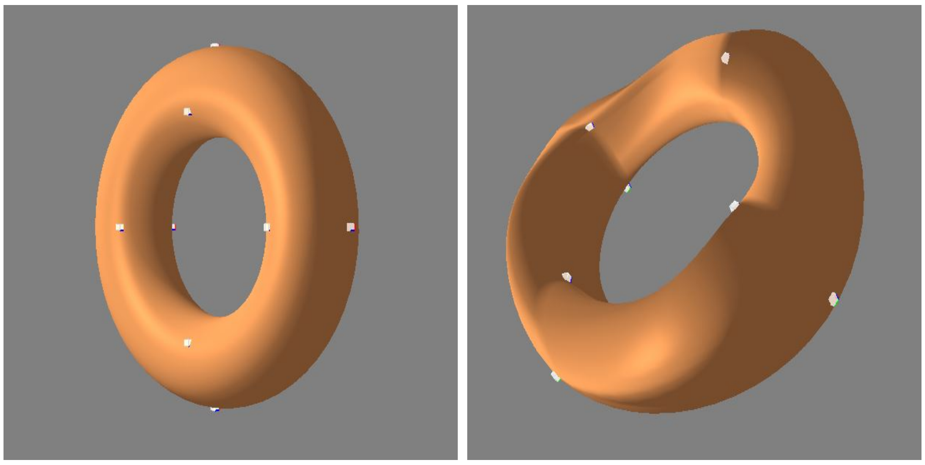

Figure 5. Illustration of the Blending Spline surface with a torus surface before (left) and after (right) it has been tranformed

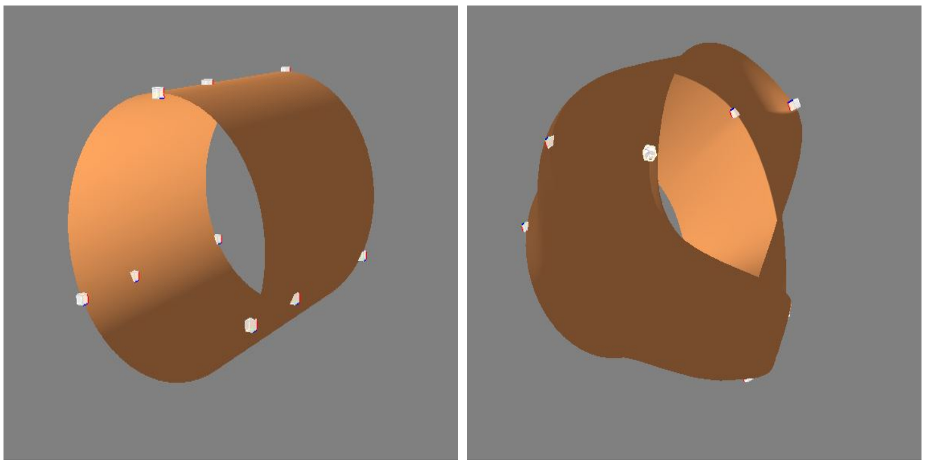

Figure 6. Illustration of the Blending Spline surface with a cylinder surface before (left) and after (right) it has been tranformed

4 Discussion & Conslusion

In summary we have managed to create all the geometric objects during the project. For future work we would like to further investigate the affine transformations of the Blending Spline curve described in section 2.4.1 due to the fact that the transformation is done similar for all the local curves. It would be interesting to explore the possibility to translate and rotate the local curves in order to transform the curve into a real-world object.

5 References

Klæboe, J. H. (2022). Applied geometry, code. https://github.com/jkl019/applied-geometry.

Lakså, A. (2022). *Blending techniques in Curve and Surface constructions.*

Lakså, A., Bratlie, J., Dalmo, R., and Bang, B. (2022). Gmlib. https://source.coderefinery.org/gmlib/gmlib1/gmlib