LAB REPORT: LAB 3

TNM079, MODELING AND ANIMATION

Anna Wästling
annwa917@student.liu.se

Thursday 26th May, 2022

Abstract

This report describes the theory and implementation of curve and mesh subdivision. Repeated subdivision, where new points are inserted between old points, is used to get the wanted result. Special consideration was taken for the boundaries of the curves. Loop´s subdivision scheme for triangle meshes was used for subdividing mesh and the weights for the new vertex position was calculated using the average of the old ones. The result of the subdivision is presented and discussed both for curves and meshes.

1 Introduction

Smooth curves and surfaces can be achieved through subdivision which is often used in computer graphics for different applications. Because of the nature of subdivision, it is fast to compute, numerically stable and operations only affect the curve locally.

2 Background

By using basis functions a parametric curve *p(t)* can be described as

$$p(t) = \sum_{i=0}^{n} c_i t^i \tag{1}$$

Where t^i is the i:th polynomial basis function and c_i is the i:th coefficient. A linear interpolation between two control points p_0 and p_1 is

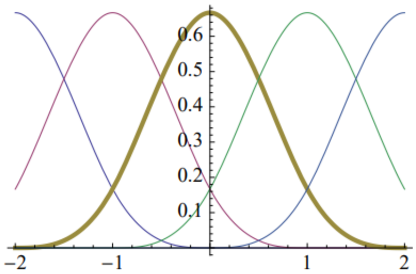


Figure 1: A cubic B-spline basis function (yellow) with translated copies, from [1]

given by

$$p_1 = (1-t)c_0 + tc_1 \tag{2}$$

where the basis function is

$$\{t, 1-t\} \tag{3}$$

for each coefficient in the interval [0,1]. By including more control points more interesting curves can be produced.

In this report the *uniform cubic B-spline* is used. The basis function of the uniform cubic B-spline in the interval [-2,2] is described by

$$B_3(t) = \frac{1}{6} \begin{cases} (t+2)^3 & , -2 \le t < -1 \\ -3(t+1)^3 + 3(t+2)^2 + 3(t+1) + 1 & , -1 \le t < 0 \\ 3t^3 - 6t^2 + 4 & , 0 \le t < 1 \\ -(t-1)^3 + 3(t-1)^2 - 3(t-1) + 1 & , 1 \le t \le 2 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The B-spline can be constructed as a linear combination of dilated and translated copies of itself as seen in figure 1. This is called that the spline is *refineable* and is an important property. Using the refinement equation

$$B_d(t) = \frac{1}{2^d} \sum_{i=0}^{d+1} \binom{d+1}{i} B_d(2t-i) \tag{5}$$

The refinement for a cubic B-spline ca be described as

$$\begin{aligned} B_3(t) = \frac{1}{8}(1B_3(2t) \\ +4B_3(2t-1) \\ +6B_3(2t-2) \\ +4B_3(2t-3) \\ +1B_3(2t-4)) \end{aligned} \tag{6}$$

When the basis functions change it is also needed to find new coefficients. These can be calculated by rewriting the equation 1 into vector form as

$$C = \begin{pmatrix} c_0 \\ c_1 \\ . \\ . \\ . \\ c_n \end{pmatrix} \tag{7}$$

and

$$B(t) = \begin{pmatrix} B_n(t) & B_n(t-1) & ... & B_n(t-n) \end{pmatrix} \tag{8}$$

which gives

$$p(t) = B(t)C \tag{9}$$

After dilating as described in equation 6 the refinement coefficients can be described in a matrix S with values from equation 6. This gives

$$p(t) = B(2t)SC \tag{10}$$

This process is then repeated to place the new coefficients, update the basis by re-weight old coefficients and thereby refine the curve. An example of this can be seen in figure 2

Some important notes is that the cubic B-spline is a convex combination where all coefficients sums up to 1 and therefore is inside the convex hull and forms a stable basis.

Figure 2: curve subdivision, from [1]

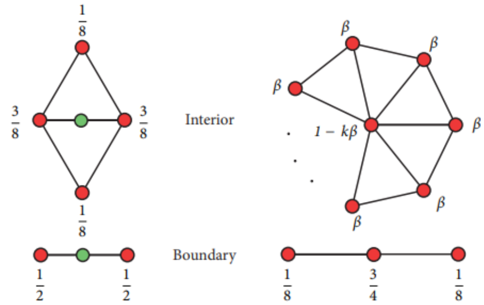


Figure 3: Mesh subdivision from [1]

Furthermore when using the matrix S boundary constraints need to be consider since the boundary values can not be weighted in the same way as the rest. In this report the boundary values are set by using the first and last value and setting these as first and last value in the coefficient vector, thereby clamping the boundaries to the spline.

To calculate the mesh subdivision the Loop subdivision scheme was used. Each face is split into four new faces and the new vertex position are weighted averages of the old vertex position. The weights used are presented in figure 3. The value β was calculated by

$$\beta = \begin{cases} \frac{3}{8k} & k > 3 \\ \frac{3}{16} & k = 3 \end{cases} \tag{11}$$

where k is the number of vertices in a one-ring neighbourhood of a vertex.

3 Tasks

Implementation of the curve and mesh subdivision is explained in this section.

3.1 Implementing curve subdivision

For calculating the curve subdivision of a cubic B-spline the property refineable was used and the refinement was calculated as in equation 6 to create the S matrix used in equation 10. If the boundaries was taken out of consideration the S matrix could be transformed into two rules.

$$c_i^t = \frac{1}{8}(1c_{i-1} + 6c_i + 1c_{i+1}) \tag{12}$$

$$c_{i+\frac{1}{2}}^t = \frac{1}{8}(4c_i + 4c_{i+1}) \tag{13}$$

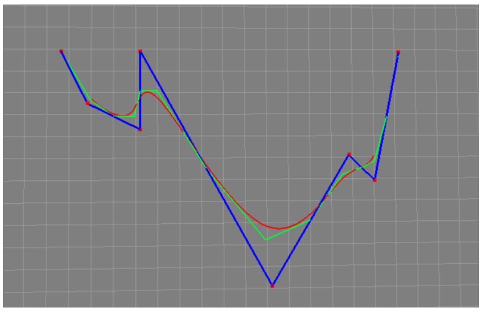
These was used when iterating through the vector containing all the old coefficients, to re-weight their position and add more coefficients positioned in between the old ones. The boundaries was as previously stated handled by using the splines first and last value to clamp the boundaries to the spline.

3.2 Implementing mesh subdivision

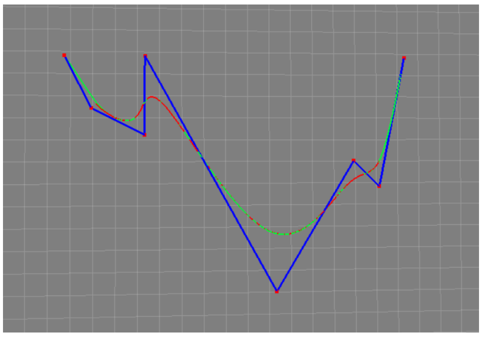
One task was to provide better placements of the vertices with weights by using the Loop subdivision scheme. This was achieved by, for each vertex, use its one-ring neighbourhood and weight the vertices in the neighbourhood by using equation 11 and weight the vertex with 1-kβ as seen in figure 3. The other task was to divide the faces into four new faces and was done by weight the new vertices position from the old ones by following

$$v = \frac{3}{8}(v_0 + v_1) + \frac{1}{8}(v_2 + v_3) \tag{14}$$

as seen in figure 3 where v is the green point.



(a)

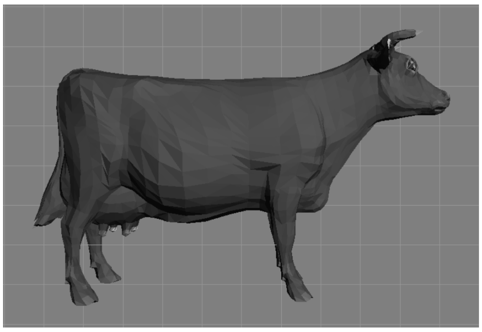


(b)

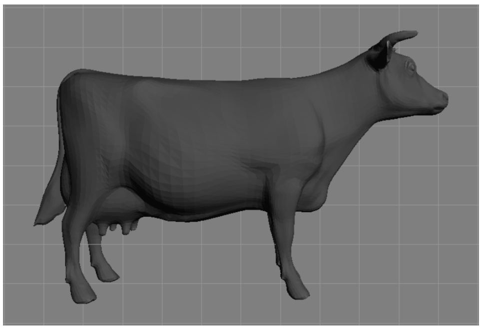
Figure 4: Result from task one showing (a) one iteration of subdivision (green curve) compared to three iterations in (b) (green curve). The red points are the control points and the red curve is the analytical cubic B-spline

4 Results

The curve subdivision gives a satisfactory result and only needs a few subdivisions before the curve approximates the analytical spline. After three iterations the curves are very similar. The result of the curve subdivision is presented in figure 4. The implementation for mesh subdivision also gives a satisfactory result and produces more smooth meshes for every subdivision. However as can be imagined the number of faces quickly increases and made the program slow after only two subdivisions. The result of the mesh subdivision is presented in figure 5 where one clearly see a difference between the original model and the subdivided model that is more smooth.



(a)



(b)

Figure 5: Result from task two showing (a) the original model compared to one iteration of Loop´s subdivision in (b).

5 Conclusion

The curve subdivision was rather straight forward to implement when knowing the theory and created a satisfactory result with few iterations.

Using Loops subdivision scheme was a nice choice for the data structure half-edge mesh structure used in the lab since it provides easy and fast access to a vertex neighbours, making it relatively fast and easy to re-weight the positions of old vertices. The result is satisfactory and by weighting the new vertex positions and re-weighting the old ones the result is smoother and better formed overall.

6 Lab partner and grade

The lab was done together with Tim Olsson and aims for grade 3.

References

[1] Robin Skånberg Mark E. Dieckmann and Emma Broman. “splines and subdivision”. 2021.