LAB REPORT: LAB 3

TNM079, MODELING AND ANIMATION

Julian Voith
julvo429@student.liu.se

Sunday 24th September, 2023

Abstract

This lab report describes the tasks and results from lab 3 of the course TNM079, Modeling and Animation at Linköping University. The lab explained the theory behind uniform B-splines and how they can be used to subdivide curves. Additionally, Loop's subdivision scheme for triangle meshes were explained and implemented including an alternative simple adaptive scheme which preserves sharp edges while smoothing the surface. Overall the subdivision schemes described delivered good and expected results. The optimizing of the B-spline evaluation decreased the number of calculations significantly while providing the same results.

1 Background

Surfaces of models are not always flat and sharp. It is necessary to mathematically represent curves. In 3D graphics this is usually done by splines. In the lab a parametric representation of the spline was used which maps a real number to a vector output.

$$f : \mathbb{R} \to \mathbb{R}^n \tag{1}$$

The most prominent splines in computer graphics is the B-spline, especially the uniform cubic B-spline, which is the most commonly applied one. This lab discusses cardinal B-splines which are using equally spaced knots along the curve *C(t)* in the two dimensional plane. The curve parameter, t, is subdivided into intervals between the knots which are given by $t_{i+1} = t_i + h$ with constant h. The points on the curve are then calculated by

$$p(t) = \sum_i c_i N_i^n(t) \tag{2}$$

where c_i are the control points and N_i^n are the B-splines between them, and n is the polynomial order of the B-spline.

First order B-spline is a piece-wise constant function which is defined as

$$N_t^1(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & otherwise \end{cases} . \tag{3}$$

For curves higher order B-splines are needed which can be recursively constructed by using the **Cox & de Boor** algorithm:

$$N_i^n(t) = \frac{t - t_i}{t_{i+n-1} - t_i} N_i^{n-1}(t) + \frac{t_{i+n} - t}{t_{i+n} - t_{i+1}} N_{i+1}^{n-1}(t) \tag{4}$$

which defines the third order with i = 0 and interval [-2,2] like following:

$$N_t^3(t) = \frac{1}{6}\begin{cases} (t+2)^3, & -2 \leq t < -1 \\ -3(t+1)^3 + 3(t+1)^2 + 3(t+1) + 1, & -1 \leq -1 \leq t < 0 \\ 3t^3 - 6t^1 + 4, & 0 \leq t < 1 \\ -(t-1)^3 + 3(t-1)^2 - 3(t-1) + 1, & 1 \leq t \leq 2 \\ 0, & otherwise \end{cases} . \tag{5}$$

B-splines are subdividable according to Zorin and Schröder [4] by the following formula:

$$N_i^n(t) = \frac{1}{2^{n-1}}\left(\sum_{m=0}^{n} \frac{n!}{m!(n-m)!} N_i^n(2t-m)\right) \tag{6}$$

which states that the B-spline can be written as a linear combination of shifted and compressed copies of itself.

Since the basis changes, the control points or coefficients, also change. To find new ones, the 2 can be rewritten in matrix form as

$$p(t) = N^n(t)C, \tag{7}$$

with **C** and $N^t(t)$ as

$$C = (c_0, c_1, ..., c_k), N^n(t) = [N_0^n(t), N_1^n(t) \dots (N_k^n(t)] \tag{8}$$

Using this form, the refinement coefficients can be placed in a matrix **S** by

$$N^n(t) = N^n(2t)S \tag{9}$$

and rearranging the terms of (7)

$$p(t) = N^n(t)C = N^n(2t)SC. \tag{10}$$

and by extending

$$C_j + 1 = SC_j \tag{11}$$

where the subscript on the coefficients shows the number of refinements. The subdivision matrix can be obtained from the coefficients of equation 6 except for the boundaries. Therefore, rules need to be defined for subdivitions near to a boundary. These were defined by Lane and Riesenfeld [2] as

$$s_{0,0} = 1, s_{1,0} = 0.5, s_{1,1} = 0.5, s_{n-1,m-1} = 0.5, s_{n-1,m} = 0.5, s_{n,m} = 1 \tag{12}$$

The subdivision matrix for B-splines at order 3 becomes following form:

$$S = \frac{1}{8}\begin{pmatrix} 8 & 0 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 & 0 \\ 1 & 6 & 1 & 0 & 0 \\ 8 & 4 & 4 & 0 & 0 \\ 8 & 1 & 6 & 1 & 0 \\ 8 & 0 & 4 & 4 & 0 \\ 8 & 0 & 1 & 6 & 1 \\ 8 & 0 & 0 & 4 & 4 \\ 8 & 0 & 0 & 0 & 8 \end{pmatrix} \tag{13}$$

In practice the whole matrix does not need to be stored fully. Except for the boundaries there are only following rules for the matrix:

$$c_i' = \frac{1}{8}(1c_{i-1} + 6c_i + 1c_{i+1}) \tag{14}$$

$$c_{i+\frac{1}{2}}' = \frac{1}{8}(4c_i + 4c_{i+1}) \tag{15}$$

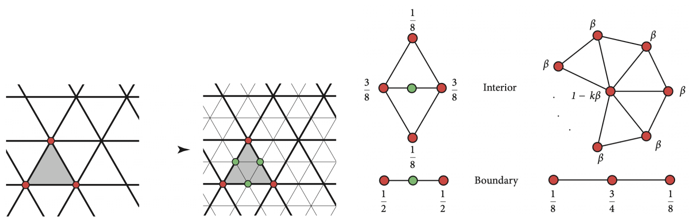
where each old coefficient c_i is being re-weighted and a new coefficient is being inserted with $c_{i+\frac{1}{2}}$ between two old ones. The boundary refinements are given by

$$c_0' = c_0 \tag{16}$$

$$c_e'nd = c_end. \tag{17}$$

It can be shown that the subdivision is convergent, smooth, and invariant under affine transformations.

In many occasion in 3D graphics it is necessary to create coarser or smoother models automatically. One way to generate a smoother surface is the Loop subdivision scheme [3], which was described in the lab. This algorithm splits each triangle of the original mesh into four new triangles like shown in Figure 1. Due to the fact that new vertices and faces are added, the



(a) Triangle subdivision (b) Weights for new positions

Figure 1: Subdivision of one triangle with Loop subdivision and the used weights

positions of the new and old vertices need to be recalculated. The positions of the vertices are calculated as the weighted averages of the old vertices. The weights can be found in figure 1. To calculate the weights β the formula proposed by Hoppe er al. [1] is used

$$\beta = \begin{cases} \frac{3}{8k'}, k > 3 \\ \frac{3}{16'}, K = 3 \end{cases} . \tag{18}$$

where k is the valence (number of incident edges).

It is not always necessary to subdivide every single triangle in a mesh. The level of refinements can differ between locations of the model, depending of the use case. One use case to create a adaptive subdivision scheme, is the subdivision of a mesh which contains smooth surfaces but also sharp edges. Here its desirable to keep the sharp edges, while smoothing the surfaces. A simple way to archive this is to first fetch the neighboring faces of the currently selected face. Next step is to iterate through all the faces and calculate the **maximum angle** between the current face and its neighboring faces with the glm function `glm::angle`. Is the angle α rather big then it is assumed that the face is part of a sharp edge. Which angle is considered as sharp edge needs to be defined by a threshold t. In the lab the value for t was $\frac{\pi}{3}$ or 60°. This leads to following rule:

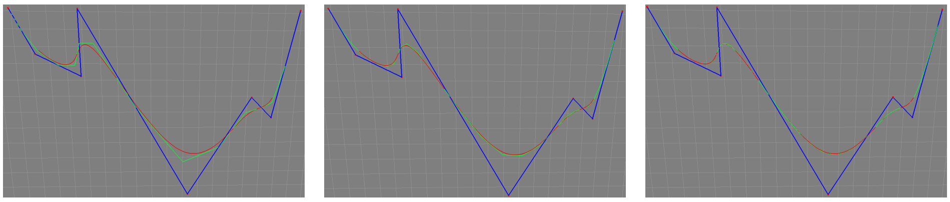
$$subdividable = \begin{cases} yes, \alpha <= t \\ no, \alpha > t \end{cases} . \tag{19}$$

2 Results

The results of the lab implementations are presented in following sub chapters.

2.1 Curve subdivision

The subdivision curve is demonstrated in figure 2. The curve converges each iteration more and more towards the analytical curve since the number of subdivisions grows exponentially with the number of subdivision.

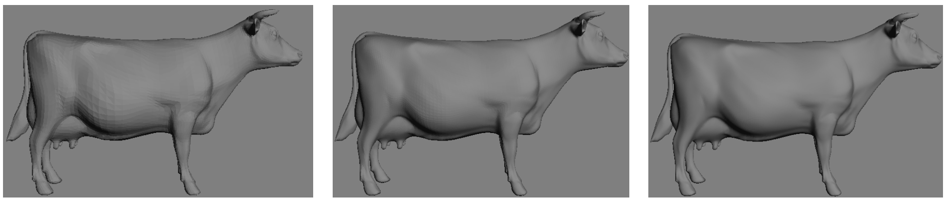


(a) 1 subdivision iteration (b) 2 subdivision iteration (c) 3 subdivision iteration

Figure 2: Demonstration of the subdivision curve. The initial curve is blue, the subdivided curve green, and the analytical curve is red.

2.2 Mesh subdivision

A demonstration of Loop's subdivision scheme can be found in figure 3.



(a) 1 mesh subdivision (b) 2 mesh subdivision (c) 3 mesh subdivision

Figure 3: Demonstration of Loop's subdivision scheme applied to the cow model.

2.3 Localization of the spline evaluation

The localization of the B-spline curve evaluation was tested on the curve shown in 2. Before the implementation 4000 B-spline evaluations were performed. The localization decreased the number to 1999.

2.4 Adaptive mesh subdivision

A demonstration of the adaptive mesh subdivision can be found in figure 4. The custom scheme can be especially seen at the ears throughout the iterations because the faces are arranged in a angle witch is bigger than the threshold t. The rest of the face is getting smoother after each iteration.

(a) 1 mesh subdivision (b) 2 mesh subdivision (c) 3 mesh subdivision

Figure 4: Demonstration of adaptive mesh subdivision on the cow model.

3 Conclusion

Overall B-splines provide a good foundation to draw curves as well as surfaces in the 3D environment.

The curve subdevision estabilishes a good aproximation of the analytical B-spline as well as the Loop subdivision which generates a quite smooth surface after some iterations. One consequence of the surface subdivision is, that it increases the amount of faces drastically which results in a high memory demand. A sophisticated adaptive subdivision scheme can be a counter measure for that. In the lab only a very simple one has been implemented.

The evaluation of the curve can be established more efficiently by localizing the evaluation of the curve. The localization provides identical results

4 Lab partner and grade

The lab and all tasks were solved together with William Toft(wilto938). This report aims for grade 5.

References

[1] T. Cuhamp M. Halstead H. Jin J. Mcdonald J. Schweitzer H. Hoppe, T. DeRose and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1994. ACM Press.

[2] J.M. Lane and R.F. Riesenfeld. A theoretical development for the computer display and generation of piecewise polynomial surfaces. In *IEEE Transactions PAMI2*, 1980.

[3] C. Loop. *Smooth subdivision surfaces based on triangles*. PhD thesis, Unversity of Utah, August 1987.

[4] Denis Zorin and Peter Schröder. Subdivision for modelin and animation: Siggraph 2000 course notes. In *SIGGRAPH '00: ACM SIGGRAPH 2000 Courses*, New York, NY, USA, 2000. ACM.