LAB REPORT: LAB 3

TNM079, MODELING AND ANIMATION

Algot Sandahl
algsa119@student.liu.se

Wednesday 16th June, 2021

Abstract

This report covers the theory behind uniform B-splines and how they are used to derive subdivision curves. Furthermore, Loop's subdivision scheme for triangle meshes is covered. A simple scheme for keeping sharp edges when refining objects is also presented. The subdivision schemes described refined the curves and surfaces with satisfactory result. The optimization of the B-spline evaluation decreased the number of calculations significantly while keeping the result intact.

1 Background

Mathematically representing smooth curves and surfaces is highly interesting for many applications, for example when modeling surfaces. This is typically done using splines.

A spline curve in n dimensions is represented as a parametric function

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

One spline that is highly interesting for the field of computer graphics is the B-spline. This lab narrows the scope and covers the cardinal B-splines: B-splines with equidistant knots. The curve parameter, t, is subdivided into intervals between the knots given by $t_i = ih$ where h is the constant step size. The points on the curve are then given by

$$\mathbf{P}(t) = \sum_i \mathbf{c}_i N_i^n(t) \tag{2}$$

where $\mathbf{c}_i$ are the control points, N_i^n are the B-splines, and n is the B-spline order and is one higher than the polynomial order.

The first order B-spline is a piecewise constant function with limited support:

$$N_i^1(t) = \begin{cases} 1, & \text{if } i \leq t/h < i+1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Higher order B-splines can be constructed recursively from lower order ones using the Cox–de Boor algorithm:

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

For example, the fourth order B-spline with $i = 0$ is given by

$$N_0^4 = \frac{1}{6} \begin{cases} (t+2)^3, & -2 < t < -1 \\ -3(t+1)^3 + 3(t+1)^2 + 3(t+1) + 1, & -1 \leq t < 0 \\ 3t^2 - 6t^2 + 4, & 0 \leq t < 1 \\ -(t-1)^3 + 3(t-1)^2 - 3(t-1) + t, & 1 \leq t < 2 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The sum in (2) can be evaluated for the entire range of i, but when evaluating curve points in practice this is ineffective. Since N_i^n is only non-zero in a small range, the sum only needs to be evaluated for a small range of i. For N_i^4 the evaluation is done for i between

$$\begin{aligned} \text{start} &= \lfloor t \rfloor + 1 \\ \text{end} &= \lfloor t \rfloor + 2 \end{aligned} \tag{6}$$

The floor operation can be simply performed as a `static_cast<size_t>` of the floating point number t.

As shown by Zorin and Schröder [1], B-splines are subdividable according to the following formula

$$N_i^n(t) = \frac{1}{2^{n-1}} \sum_{l=0}^{n} \binom{n}{l} N_i^n(2t-l) \tag{7}$$

where

$$\binom{k}{m} = \frac{k!}{m!(k-m)!}. \tag{8}$$

What this means is that the a B-spline can be constructed as a linear combination of shifted and compressed copies of itself.

Since the basis changes, the control points, $\mathbf{c}_i$, does as well. To find the new ones, the sum in (2) can be rewritten in vector form:

$$\begin{aligned} \mathbf{P}(t) &= \mathbf{N}^n(t)\mathbf{C}, \\ \mathbf{N}^n(t) &= \begin{bmatrix} N_0^n(t) & N_1^n(t) & \cdots & N_k^n(t) \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_k \end{bmatrix}. \end{aligned} \tag{9}$$

Using this form, the refinement coefficients can be put in a matrix, $\mathbf{S}$, giving

$$\mathbf{N}^n(t) = \mathbf{N}^n(2t)\mathbf{S} \tag{10}$$

and with (9):

$$\mathbf{P}(t) = \mathbf{N}^n(t)\mathbf{C} = \mathbf{N}^n(2t)\mathbf{S}\mathbf{C} \tag{11}$$

and by extension,

$$\mathbf{C}_{j+1} = \mathbf{S}\mathbf{C}_j \tag{12}$$

where the subscript denotes the number of iterations. The subdivision matrix for B-splines of order 4 (and for others as well) can be obtained from the coefficients given by (7), except at the boundaries which are given by

$$\begin{bmatrix} 1 & & & & & \\ 0.5 & 0.5 & & & & \\ & & \ddots & & & \\ & & & & 0.5 & 0.5 \\ & & & & & 1 \end{bmatrix} \tag{13}$$

as described by [2]. So the subdivision matrix for B-splines of order 4 for a curve with five points, takes the form

$$\mathbf{S} = \frac{1}{8} \begin{bmatrix} 8 & 0 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 & 0 \\ 1 & 6 & 1 & 0 & 0 \\ 0 & 4 & 4 & 0 & 0 \\ 0 & 1 & 6 & 1 & 0 \\ 0 & 0 & 4 & 4 & 0 \\ 0 & 0 & 1 & 6 & 1 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 8 \end{bmatrix}. \tag{14}$$

This can then be reduced to this simple procedure: For each subdivision step, a new point is inserted in between each of the old points; the new points and the new positions of the old points are given by

$$\begin{aligned} \mathbf{c}_i' &= \frac{1}{8}(\mathbf{c}_{i-1} + 6\mathbf{c}_i + \mathbf{c}_{i+1}) \\ \mathbf{c}_{i+\frac{1}{2}}' &= \frac{1}{8}(4\mathbf{c}_i + 4\mathbf{c}_{i+1}) \end{aligned} \tag{15}$$

where $\mathbf{c}_i$ are the old points. The boundary conditions are given by

$$\begin{aligned} \mathbf{c}_0' &= \mathbf{c}_0 \\ \mathbf{c}_{\text{end}}' &= \mathbf{c}_{\text{end}} \end{aligned} \tag{16}$$

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

Generating a smooth model from a coarse one is useful in many contexts, for example when creating models. One way to do so for triangle meshes is Loop's subdivision algorithm [3] which is partly based on the properties of B-spline curves. The algorithm works by subdividing all triangles into four new ones, as seen to the left in Figure 1. The position of the new vertices as well as the pre-existing ones are calculated as a weighed average of the nearby vertices according to the weights in Figure 1. As proposed by Hoppe et al. [4], the weights β are calculated as

$$\beta = \begin{cases} \dfrac{3}{8k}, & k > 3 \\ \dfrac{3}{16}, & k = 3 \end{cases} \tag{17}$$

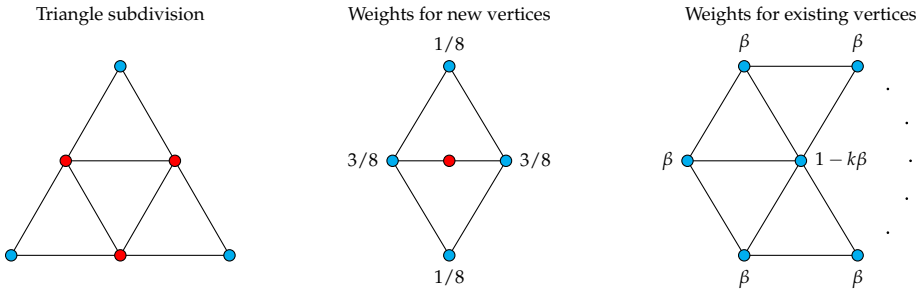


Figure 1: Description of how triangles are subdivided in Loop's subdivision algorithm. Red vertices are new, and blue vertices are pre-existing.

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

It is not always necessary or even desirable to subdivide every single triangle in a mesh; some parts might for example be less visible and thus not need a resolution as high as other parts of the mesh.

Another situation that can motivate an adaptive subdivision scheme, is the subdivision of meshes with sharp edges mixed with smoother surfaces. It is most likely desirable to keep the sharp edges, while letting the smoother surfaces be subdivided. A simple way to achieve this is to begin with calculating the cosine of the angle between the current face and all of the neighboring faces (indexed by j),

$$\cos\theta_j = \hat{\mathbf{n}} \cdot \hat{\mathbf{n}}_j \tag{18}$$

where $\hat{\mathbf{n}}$ is the normal of the current face, and $\hat{\mathbf{n}}_j$ is the normal of the neighboring face. To decide if the face is subdividable, the following rule is used

$$\text{subdividable} = \begin{cases} \text{no}, & \text{if } \cos\theta_{\max} < \alpha \\ \text{yes}, & \text{otherwise} \end{cases} \tag{19}$$

where α is some threshold value chosen such that edges are preserved without preventing the subdivision of other parts. We used the threshold $\alpha = 0.2$.

It should be noted that when not all faces are subdivided, the neighboring faces can not be subdivided in the same way as before.

2 Results

The results of the lab are presented here.

2.1 Curve Subdivision

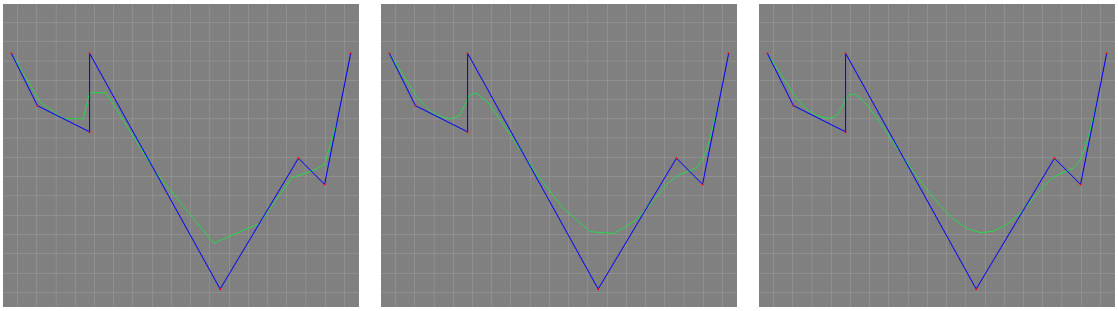
The subdivision curve is demonstrated in Figure 2. The curve converges quite quickly towards the analytical curve shown in Figure 4 since the number of subdivisions grows exponentially with the number of subdivisions.

2.2 Mesh Subdivision

Loop's subdivision scheme is demonstrated in Figure 5. Since each triangle is subdivided into four new ones every iteration, the number of triangles grow rapidly, increasing the smoothness as well as the memory requirements. As mentioned earlier, adaptive subdivision schemes could be used to decrease the number of faces in areas where they are unnecessary, but to what extent this can be done is highly dependent of the model as well as the context considered.

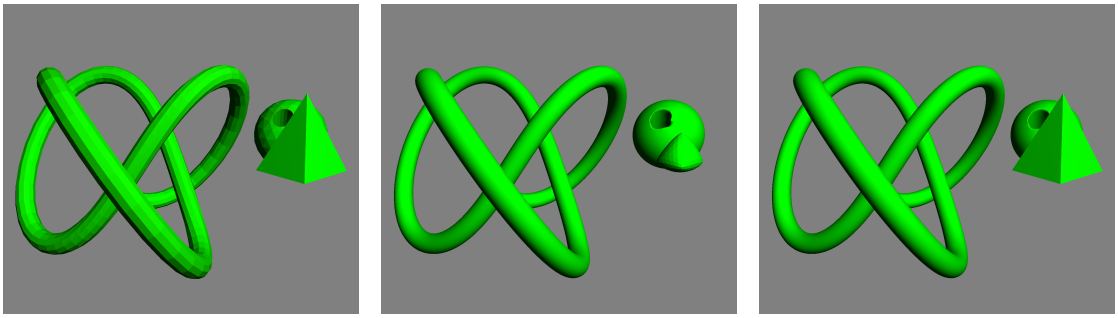
2.3 Localization of the B-Spline Curve Evaluation

The localization of the B-spline curve evaluation was tested on the curve shown in Figure 4. Before implementing the localization, 4000 B-spline evaluations were performed; with the localization, this number was halved, to 2000 evaluations.



(a) 1 subdivision. *(b)* 2 subdivisions. *(c)* 3 subdivisions.

Figure 2: Demonstration of the subdivision curve. The initial curve is blue.



(a) Original model. *(b)* 3 iterations without edge triangles. *(c)* 3 iterations with edge preservation.

Figure 3: Comparison between the edge preservation technique and no edge preservation.

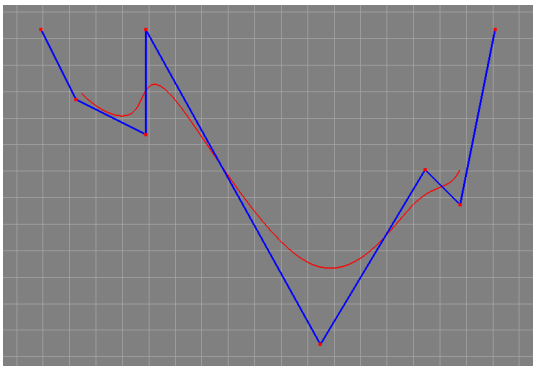


Figure 4: The corresponding analytical curve (red) to Figure 2.

For a curve with more knots, the improvement would be even more significant. This can be shown by realizing that after the optimization, the number of evaluations needed for each point at the curve no longer depends on the number of knots—this is in contrast with the naïve implementation where it grows linearly with the number of knots.

2.4 Edge Preservation

The edge preservation scheme is shown in Figure 3. As seen, the pyramid changes shape drastically without the edge preservation. With the edge preservation, however the the shape is preserved while still smoothing the rest of the elements.

Some artifacts can however occur. Furthermore, flat surfaces are subdivided using this scheme, to no visual benefit.

3 Conclusion

B-splines provide a useful mathematical foundation for curves as well as surfaces.

The curve subdivision provides a good approximation of the analytical B-spline after only a few iteration.

The Loop subdivision scheme smoothens a surface with good result, but the number of faces increases rapidly as a consequence.

By localizing the evaluation of the curve, less computer resources are needed, yet the resulting curve is identical.

The simple edge preservation scheme used works, but more sophisticated schemes could probably provide better results.

Lab Partner and Grade

The lab was done together with Viktor Sjögren. All lab tasks were finished and the report aims for grade 5.

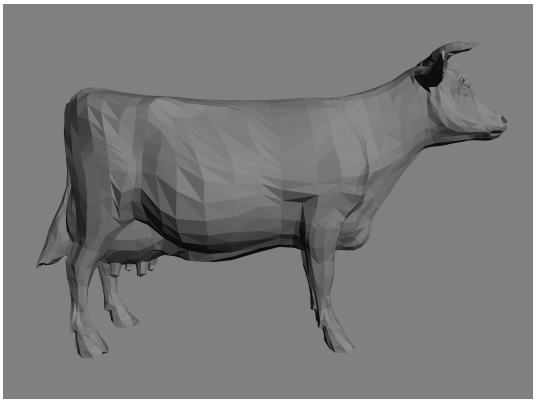
References

[1] D. Zorin, P. Schröder, A. Derose, L. Kobbelt, A. Levin, and W. Sweldens, "Subdivision for modeling and animation," 2000.

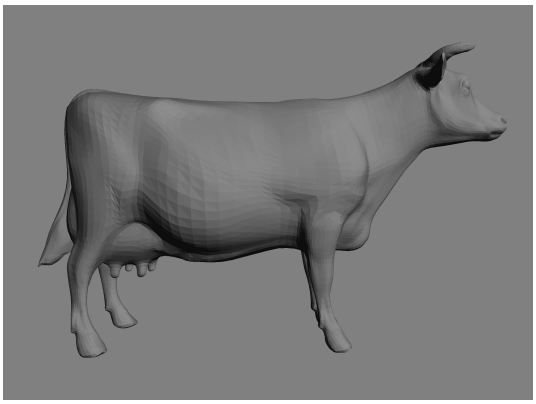
[2] J. M. Lane and R. F. Riesenfeld, "A theoretical development for the computer generation and display of piecewise polynomial surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 1, pp. 35–46, 1980.

[3] C. Loop, "Smooth subdivision surfaces based on triangles," Ph.D. dissertation, January 1987.

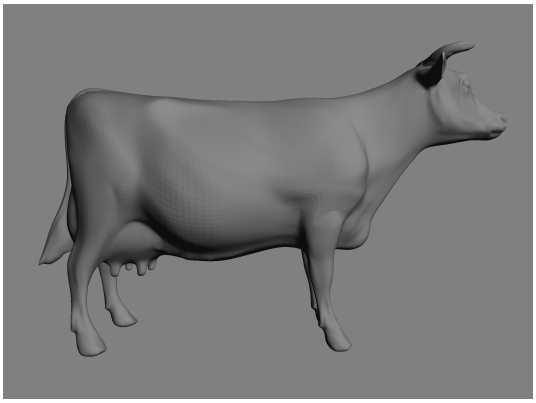
[4] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, "Piecewise smooth surface reconstruction," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '94. New York, NY, USA: Association for Computing Machinery, 1994, p. 295–302. [Online]. Available: https://doi.org/10.1145/192161.192233



(a) Original model.



(b) 1 iteration.



(c) 2 iterations.

Figure 5: Loop's subdivision scheme applied to a model of a cow.