LAB REPORT: LAB 3

TNM079, MODELING AND ANIMATION

Linnea Bergman
linbe810@student.liu.se

Monday 2nd July, 2018

Abstract

In computer graphics one important feature is to be able to make surfaces and curves smooth. By subdividing the curves and surfaces with an approximation they can become smoother and get more vertices. The scheme that was used for the subdivision, to create smooth curves ans surfaces, was the Loop subdivision scheme.

1 Introduction

Subdivision curves and surfaces are fast to compute and are widely used in the area of applied computer graphics. The splines used in the lab were uniform cubic B-spline curves. Parametric representations were also used and the curves are rewritten to a stable basis. The curve subdivision and the mesh subdivision was implemented. An evaluation of the spline was also done.

2 Assignments

This section covers the assignments done during the lab session. All code that was modified and used lies in the files UniformCubicSplineSubdivisionCurve, LoopSubdivisionMesh and UniformCubicSpline.

2.1 Assignment Implement curve subdivision

The class UniformQubicSplineSubdivisionCurve was updated such that it approximates a spline by subdividing each line segment. More specifically, the function Subdivide was updated with the subdivision scheme for a natural cubic spline. To begin with, space was allocated for the new coefficients, which represented the new vertices and points. All of the coefficients was then multiplied with the subdivision matrix, S Equation **??**, with boundary constraints,

$$c_{i+1} = S * c_i \tag{1}$$

where c_{i+1} was the new curve, S the matrix and c_i the old curve. The whole matrix was not needed since there are only two rules except for the boundaries seen in Equation 2 and 3.

$$\mathbf{c}_{i-1} = \frac{1}{8}(1\mathbf{c}_{i-1} + 6\mathbf{c}_i + 1\mathbf{c}_{i+1}) \tag{2}$$

$$\mathbf{c}_{i+\frac{1}{2}} = \frac{1}{8}(4\mathbf{c}_i + 4\mathbf{c}_{i+\frac{1}{2}}) \tag{3}$$

The old coefficients, c_i, are re-weighted and a new coefficient, c_{i+1} were inserted. Values are added in the allocated space. Lastly the new coefficients were saved by overwriting the old ones.

2.2 Assignment Implement mesh subdivision

The class LoopSubdivisionMesh had a simple subdivision scheme which was improved. More specifically, the functionality in the functions VertexRule and EdgeRule was modified to provide better placements of the additional vertices by the Loop subdivision scheme. The new triangles created from the added vertices generates smoothness if placed correctly.

In VertexRule, given a vertex, the neighbouring vertices was saved in a temporary variable. Multiply the coordinates of the given vertex with the center. For each vertex the coordinates were summarized and multiplied with β according to Equation 5.

$$\beta = \begin{cases} \frac{3}{8k}, & k > 3 \\ \frac{3}{16}, & k = 3 \end{cases} \tag{4}$$

$$v_{new} = v * (1 - k * \beta) \tag{5}$$

v is the given vertex, v_{new} the new vertex, k the valence (number of incident edges) and β is given from Equation 4. The summation gave the new position and was returned from the function VertexRule.

In EdgeRule the new edge vertex should be placed halfway along a given edge. The edge pair, its previous edge and the pairs previous edge vertices was each saved in a temporary variable. These was used for the computation of the Loop's algorithm weighting mask, Equation 6.

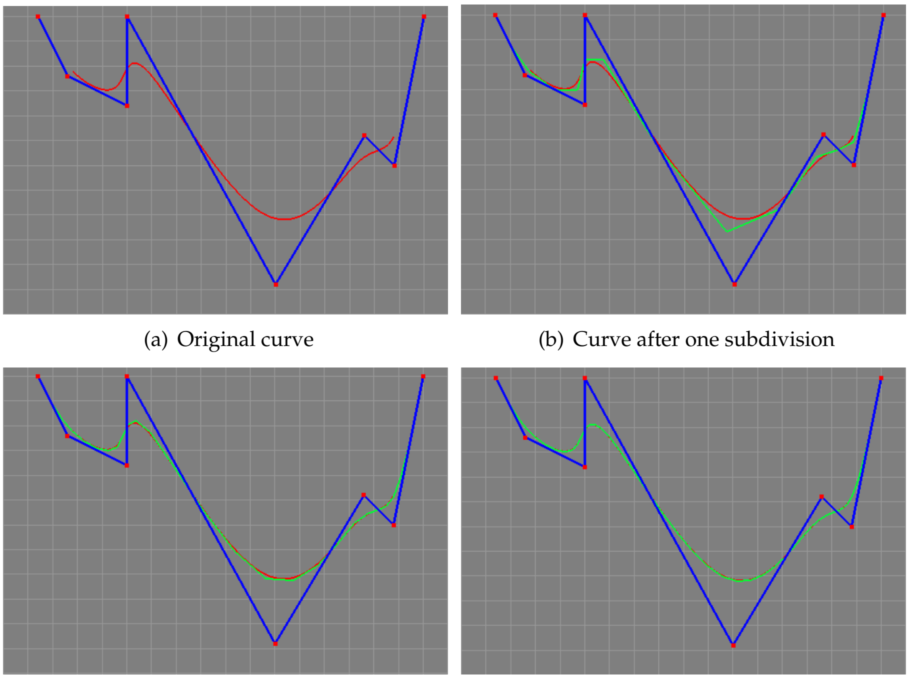
$$V_N = \frac{3}{8}V_L + \frac{3}{8}V_R + \frac{1}{8}V_B + \frac{1}{8}V_U \tag{6}$$

where V_N is the new vertex coordinate, V_L the vertex of the given edge, V_R the pair vertex, V_B the vertex of the given previous edge and V_U the pairs previous edge vertex, which is calculated in a function called Beta. The weighting mask gives the coordinate of the new vertex. The weight mask was then returned from EdgeRule.

2.3 Assignment Localize evaluation of the spline

The existing implementation of the evaluation of the splines in the GetValue-function in UniformCubicSpline is an inefficient version. Evaluation of a spline is the summation over a set of control points supported by a basis function. GetValue was modified to restrict the summation to only those control points which are supported by the basis function at a given point. The only points that affect the given point were those in close relation to it. The rest gave no contribution to that point and could therefore be ignored. For each given point only the points two or three steps away were of importance and summarized. Before the summation a safety check was done to make sure that there were no negative indicies, that the index was not larger than the number of coefficients and that the summation was not larger than one. Return value of the function was the number of evaluations done.

3 Results



(a) Original curve

(b) Curve after one subdivision

(c) Curve after some subdivisions

(d) Curve after several subdivisions

Figure 1

Result from assignment 2.1 showing the original curve without any subdivisions, 2(a), and the result after one or more subdivisions performed on the curve, 2(b)-2(d). Red curve is the reference curve and blue the starting curve to be subdivided.

The subdivision curve can be seen in Figure 1. 1(a) is the original curve (blue) and the curve which the subdivided curve should resemble (red). 1(b), 1(c) and 1(d) shows the subdivision curve with different number of subdivisions where 1(b) had the lowest amount and 1(d) the highest. As seen, the approximation goes towards the reference curve (red).

For the mesh subdivision the different meshes can be seen in Figure 2. The original, unsubdivided, mesh is shown in 2(a). The different amount of subdivisions can be seen in 2(b), 2(c) and 2(d) where 2(b) have the least number of subdivisions and 2(d) the highest.

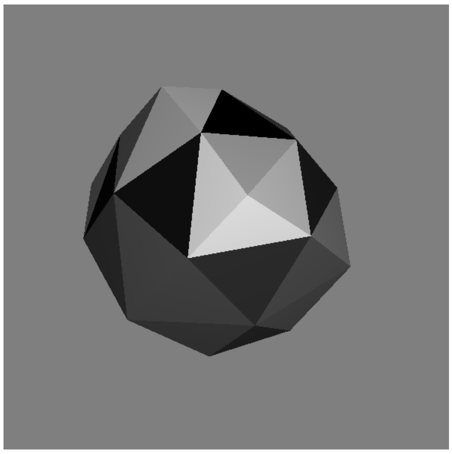
The spline evaluations went from approximately 4000 to 2499.

4 Conclusion

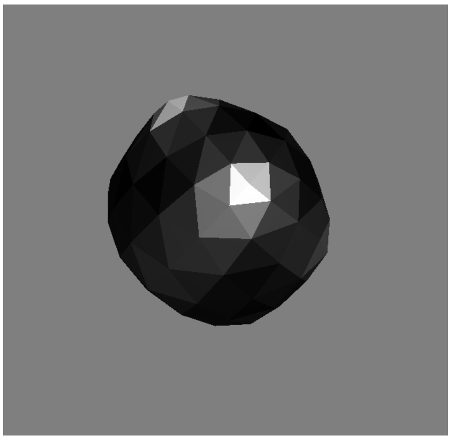
The subdivision uses approximation when subdividing and more iterations of the subdivision generates a smooth curve or mesh. The curve goes towards the wanted result (red curve in Figure 1). With enough iterations the two of them becomes hard to separate but the subdivision curve is not exactly the same as the wanted result, only really close to.

Meshes becomes smoother for every subdivision. The cube in Figure 2 looks more and more like a sphere after each iterations but will not become an exact sphere. From the first iteration the

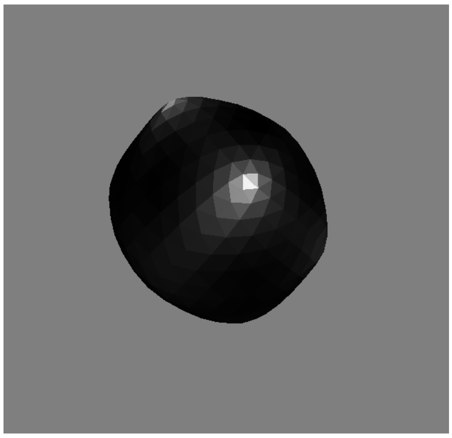
(a) Original mesh



(b) Mesh after one subdivision



(c) Mesh after some subdivisions



(d) Mesh after several subdivisions

Figure 2

Result from assignment 2.2 showing the original mesh without any subdivisions 2(a) and the result after one or more subdivisions performed on the mesh (2(b)-2(d)) cube gets a distorted shape (Figure 2(b)). The reason for that is because it have very few vertices and when a new vertex is added the rest reposition themselves to fit. Then when another vertex is added the rest are not in their original placement when they reposition themselves again. For the cube to become a perfect sphere then all new vertices have to be added and calculated before the rest of the vertices can reposition themselves. Otherwise the slight distortion will occur.

The number of necessary evaluations done for the spline could be reduced with a smart check/algorithm before the evaluation-function was called. The reduction was large because the checks only made sure that the wanted control points were evaluated and all points with a value out of bounds was skipped.

5 Lab partner and grade

This lab was done together with Rebecca Cedermalm, rebca973. We completed assignment ”Implement curve subdivision”, ”Implement mesh subdivision” and ”Localize evaluation of the spline” which are the assignments for grade 3 and 4. Therefore grade 4 seems fair.