LAB REPORT: LAB 2

TNM079, MODELING AND ANIMATION

Iris Kotsinas
iriko934@student.liu.se

Monday 13th September, 2021

Abstract

This is the second lab in a series of six labs in the course TNM079 Modeling and Animation at Linköping University. The report discusses how decimation of a mesh works and the improvements as well as difficulties that come with removing data from a mesh. In this lab, the quadric error metric was used as a way to implement edge collapse for manifold meshes represented by the half-edge data structure. It uses quadric matrices to determine the edges in the mesh which can be collapsed. The implemented method resulted in a working decimation algorithm, which provided the option to reduce the number of faces in the original model.

1 Introduction

In computer graphics, many applications require complex and highly detailed models to increase the realism. However, the computational cost for working with a highly detailed model can be considerably high. Therefore, it is often desirable to instead use approximations of the models in certain situations where the full model isn't necessary. For this purpose, decimation can be used. Decimation is used for collapsing edges from the mesh to reduce the resolution. In this lab, the quadric error metric was used as a way to implement edge collapse for manifold meshes represented by the half-edge data structure.

2 Background

Garland and Heckbert introduces a general decimation algorithm based on quadric error metrics in [3], where both edge and non-edge contractions are allowed. However, in this lab, only edge contractions are handled to simplify the implementation. In this lab, as previously mentioned, the quadric error metric was used as a way to implement edge collapse for manifold meshes represented by the half-edge data structure [2].

Edge collapse is performed by defining the cost for removal of each edge in the mesh. The edges with the lowest cost are then removed. The calculation of the cost is performed by implementing the quadric error metric. For each face, the fundamental error quadric is calculated. The vertex quadric is a sum of all the error quadrics of the faces in its 1-ring. The face quadric is the distance of a vertex from a plane. The error metric can be written in quadratic form as Equation 1 [1].

$$\mathbf{K}_p = \mathbf{p}\mathbf{p}^T = \begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix} \tag{1}$$

where $\mathbf{p} = [a\ b\ c\ d]^T$ represents the plane defined by equation $ax + by + cz + d = 0$, where $a^2 + b^2 + c^2 = 1$. (x, y, z) is the position of the face. The fundamental error metric $\mathbf{K}_p$ can be used to find the squared distance of any point to plane $\mathbf{p}$. The sum of the fundamental quadrics can represent a full set of planes by a single matrix $\mathbf{Q}$. The cost for a half-edge between vertex v_1 and v_2 has the quadric $\mathbf{Q} = \mathbf{Q}_1 + \mathbf{Q}_2$.

The error estimate $\nabla(\vec{\mathbf{v}})$ of collapsing an edge can then be calculated (see Equation 2).

$$\nabla(\vec{\mathbf{v}}) = \vec{\mathbf{v}}^T \mathbf{Q} \vec{\mathbf{v}} \tag{2}$$

In Equation 2 $\vec{\mathbf{v}}$ is the position of the new vertex, which will replace the edge. By deriving $\nabla(\vec{\mathbf{v}})$ by x, y and z, three equations are acquired. The result will be similar to the original $\mathbf{Q}$ but the last row is set to be $[0, 0, 0, 1]$. If the matrix is non-singular, $\mathbf{Q}$ is inverted and the most optimal vertex position is calculated and the cost returned (see Equation 3). Matrix $\mathbf{Q}$ is non-singular and invertible if the determinant is nonzero.

$$\vec{\mathbf{v}} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{24} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3}$$

If $\mathbf{Q}$ is singular, the optimal vertex placement has to be calculated in a different way. In this lab there is three positions considered. The edge is subdivided along v_1 and v_2, resulting in the position $(v_1 + v_2)/2$. The subdivided position is considered, as well as v_1 and v_2. Then the position that provides the lowest cost is picked. Once the best position for the new vertex has been found, Equation 2 can be used to calculate the cost.

To maintain all possible edge collapses, and efficiently pick the collapse with the least cost, a heap is used. In other words, when all the calculations for the edges have been done, the mesh will be sorted with a heap data structure. This allows the decimation function to iteratively collapse and remove the lowest cost edges from the heap to achieve a new resolution.

3 Results

In this section detailed information is provided regarding the output of the lab, where mesh decimation using quadrics was implemented. More specifically, three functions were implemented; *createQuadricForVert*, *createQuadricForFace* and *computeCollapse*.

The function *createQuadricForVert* creates the quadric of each vertex used for the calculation of the cost. It iterates over the faces and calls on *createQuadricForFace* for each separate face. That function will in turn return a quadric, which is added to all the other face quadrics.

The function *computeCollapse* recieves an edge and its connecting vertex quadrics. Each edge has pointers to **cost** and **position**. The quadrics are summed up to then create the quadric Q. A copy of Q is created, where the last row is is set to be $[0, 0, 0, 1]$. If the quadric is non-singular, the determinant nonzero, the matrix can be inversed. Therefore, the quadric can be calculated with Equation 3. If the quadric is singular, the optimal vertex placement is calculated by evaluating which position is the most optimal among the two vertices and their midpoint. Lastly, the cost is calculated with Equation 2.

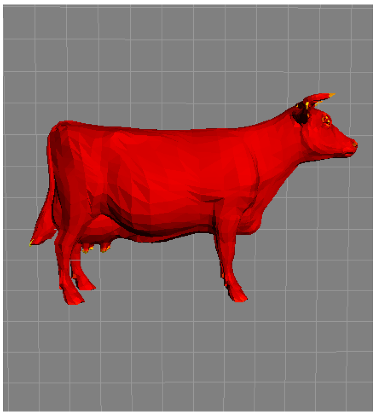

Figure 1: The original model with 5,804 faces.

In Figure 1 the original model with 5,804 faces can be seen. In Figures 2-5 a sequence of approximations can be noted. It can be noted that the approximation in Figure 2 with 994 faces is quite similar to the original model 1 with 5,804 faces, although the resolution is massively reduced. However, as the number of faces decrease even more, the resemblance to the original model decreases as well. The number of faces in Figure 4 are 25% of the faces in Figure 2, and it can be noted that further reduction of the number of faces ruins the topology due to complexity of the model.

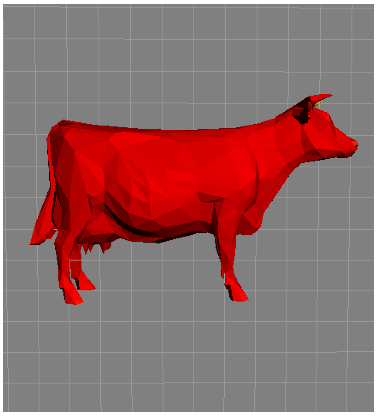

Figure 2: Approximation of the model, with 994 faces.

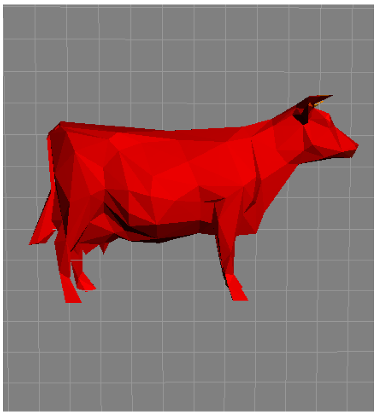

Figure 3: Approximation of the model, with 532 faces.

4 Lab partner and grade

My lab partner was Viktor Tholén, and I am aiming for grade 3.

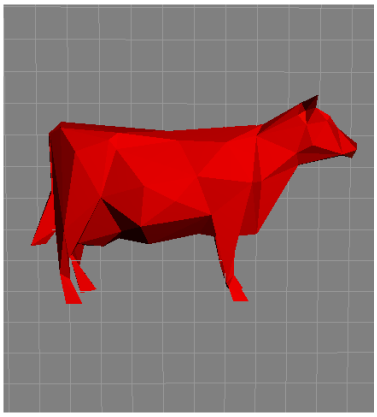

Figure 4: Approximation of the model, with 248 faces.

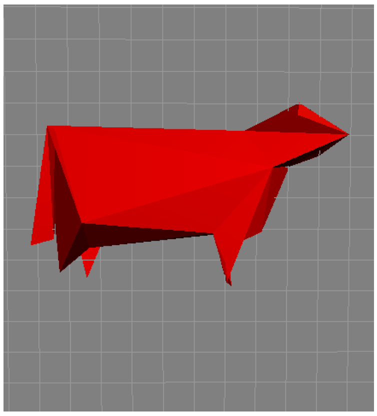

Figure 5: Approximation of the model, with 64 faces.

References

[1] Michael Garland. *Quadric-Based Polygonal Surface Simplification*. 1999.

[2] Emma Broman Mark Eric Dieckmann, Robin Skånberg. *Mesh Decimation*. Modeling and animation, lab 2, 2021.

[3] Paul S. Heckbert Michael Garland. Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, page 209–216. ACM Press/Addison-Wesley Publishing Co, New York, NY, USA, 1997.