LAB REPORT: LAB 2

TNM079, MODELING AND ANIMATION

Johnny Elmér
jonel107@student.liu.se

Wednesday 27th April, 2022

Abstract

This lab report describes the basics of a surface simplification algorithm created by Garland and Heckbert which uses quadrics and pair contraction to simplify digital objects. Pair contraction and error quadrics are described more in depth, giving a brief overview on how they work and what their function is. The results show the difference between simple mesh decimation and mesh decimation using quadrics to showcase how well the algorithm works for preserving important details of a fairly complex 3D object, in this case a cow. Although the results show great promise there is also a lot of improvements possible, which could be achieved through, for example, implementing other cost heuristics as well.

1 Background

When using mesh decimation on a digital object a small error is always introduced. This error metric can be used for the sake of adjusting the algorithm with which the mesh decimation is performed to get the best visual result possible, while also helping to avoid taking away information which is important in the eyes of the viewer. When performing a simple decimation the results often lead to some rather strange results, so in this lab, a decimation mesh using quadrics is introduced to solve this issue. In the article [1], Garland and Heckbert give an overview of their algorithm for simplifying surface models while maintaining the most important information with the help of pair contraction and quadrics.

Pair contraction consists of moving the two vertices (v_1 and v_2) to a new location ($\bar{v}$) and connecting all incident edges to v_1 before deleting the v_2. By doing this it is possible to remove several faces at ones without destroying much of the general structure of the object. It is also possible to do the same process with more than two vertices if need be, without having to adjust the process much.

Error quadrics refers to a way of calculating the error at each vertex to be able to determine the cost of performing a pair contraction between all different possible pairs. First, a Q matrix which is associated with each vertex is calculated, thereafter all possible pairs that are valid for pair contraction are selected. When all the possible pairs are selected the optimal location $\bar{v}$ for each pair is calculated and the cost of that specific contraction is gotten with the help of Equation 1

$$\bar{v}^T(Q_1 + Q_2) * \bar{v} \tag{1}$$

where Q_1 and Q_2 is the Q matrix of v_1 and v_2 respectively. The final step is to simply perform the contraction with the least cost and then to update the new costs, before repeating the whole process again until all contractions necessary have been performed. It is important to make sure only one contraction is done at a time, as each contraction can have an affect on the costs for other contractions as well as the validity of certain pairs.

2 Results

Figure 1 shows the original unaltered cow object before having applied any mesh decimation to it. The original cow consisted of 5804 faces, 8706 edges and 2903 vertices. The reason for some areas being a bit brighter/whiter is because of the use of a "Collapse Cost" option in the GUI, this will be explained a bit further down.

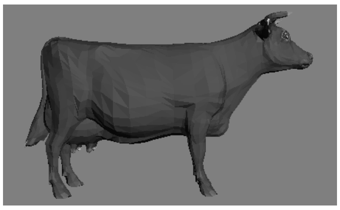


Figure 1: The original cow object without any mesh decimation.

If the model is decimated using a simple version of a decimation mesh the result is Figure 2, which has quite a few issues, mainly the fact that a lot of the important details are now missing. The identifiable features like the udders and horns are either hard to make out or missing completely, which is the opposite of what is wanted.

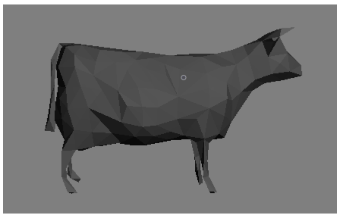
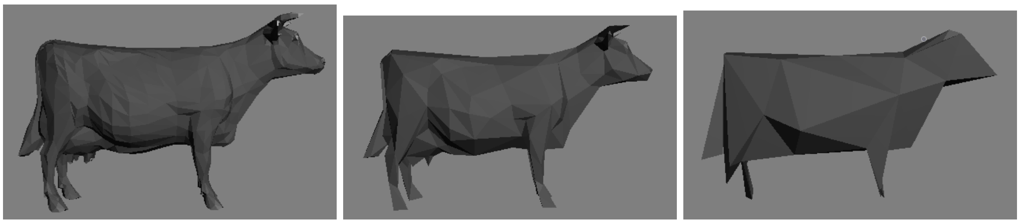


Figure 2: The cow object after a simplistic mesh decimation is applied.

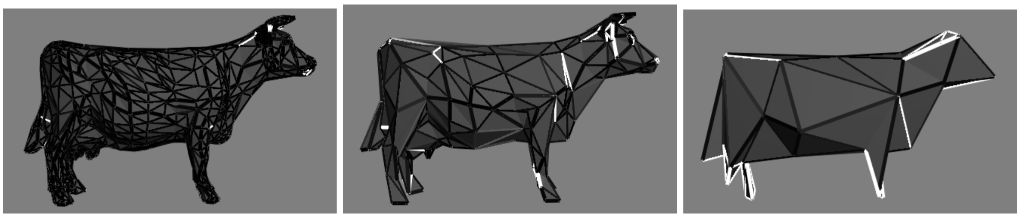
Figure 3 shows the same cow model as the one in 1 after different levels of mesh decimation using quadrics. Figure 3(a) shows the cow when decimated down to 2000 ..., 3(b) to 500 ... and 3(c) to 100 ... . The result is a mesh decimation which, unlike the simple mesh decimation shown in 2, doesn't remove those important identifiable features, yet still removes unnecessary ... .

Figure 4 shows the same object, decimated down to the same number of ... as in Figure 3 but with a wireframe representation to more clearly show what is going on. In this case a "Collapse Cost" option in the pre-implemented was used to visualise the different costs associated with collapsing the different edges. The lighter the colour of the edge, the higher the cost and therefore the less likely it is that the edge will be collapsed.



(a) (b) (c)

Figure 3: (a) Cow object decimated to 2000 ... (b) Cow object decimated to 500 ... (c) Cow object decimated to 100 ...



(a) (b) (c)

Figure 4: Wireframe representations of the objects shown in Figure 3(a) through 4(c).

3 Conclusion

When performing mesh decimation on objects it's important to be able to remove unnecessary information, to be able to for example have different levels of detail on a certain object, without without removing too much of the more crucial details. Using quadrics as an error metric helps to avoid removing information which is unnecessary or even unwanted to remove. By focusing on how much collapsing a certain edge might cost it is also possible to collapse one edge at a time in real time, making it much less noticeable to the viewer that there is a change in the number of triangles making up an object. Of course it is possible to get an even better result than the one presented in this report by, for example, implementing other cost heuristics.

4 Lab partner and grade

For the records: My lab partner was Felix Lindgren. All assignments for grade 3 were completed, hence, I should get grade 3.

References

[1] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. *SIGGRAPH 97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209 – 216.