LAB REPORT: LAB 2

TNM079, MODELING AND ANIMATION

Julian Voith
julvo429@student.liu.se

Thursday 13th April, 2023

Abstract

This lab report describes the tasks and results from lab 2 of the course TNM079, Modeling and Animation at Linköping University. The lab explained the theory and implementation of an algorithm which uses a quadric error metric to decimate a high detail mesh to a lower detail mesh while keeping details in the model as much as possible. Additionally, this metric was visualized and an alteration of the quadric heuristic was implemented. The alteration decimates the sides of a model more heavily but leaves more detail on top of it. The quadric error metric and the associated algorithm is a good method for mesh decimation with acceptable results and the alteration lead to the wished outcomes.

1 Background

Computer Graphics are getting more detailed and complex which leads to more realism. Unfortunately, this usually goes along with higher computational cost. Hence, sometimes it is useful and demanded to scale down meshes and only show complex and detailed meshes when necessary. For example objects in further distance don't need to be as high detailed as objects in front of the camera where quality differences are more noticeable. Such objects with low detail can, of course, be created manually by hand but then a degree of flexibility is lost. It is more useful and flexible to create them automatically. The aim of this lab was to implement an automatic decimation of the mesh based on a quadric error metric which creates a low detail mesh out of a high detail mesh.

The algorithm used in the lab decides based on the quadric error metric which edges should be collapsed. It iteratively collapses edges that have the smallest cost/error, until the set number of remaining faces is reached.

In order to calculate the cost of collapsing or distance to the new vertex $\bar{v}$ the *quadric error metric* calculation based on the method presented by Garland and Heckbert[1] was used. This method extends a infinite plane for each adjacent faces of a vertex $\mathbf{v}$. The distance between $\mathbf{v}$ and the three planes is zero: $\Delta(v) = 0$. Shifting the vertex $\mathbf{v}$ to a new position $\bar{v}$ should yield $\Delta(v) > 0$. By letting the distance from a new position, $\bar{v}$, to each of those planes be d, the error metric of moving $\mathbf{v}$ to a new position $\bar{v}$ can be defined using the squared distance:

$$\Delta(\bar{v}) = \sum_{i=1}^{N_t} d_i^2 \tag{1}$$

where N_t is the number of triangles in the 1-ring of v. To find the distances for **d**, the normal form of the plane **p**

$$ax + by + cz + d = 0 \tag{2}$$

is used, where a, b, and c is given by the face normal **n**, which is already known. Hence, we need to find **d**, which can be obtained by

$$d = -(ax_o + by_o + cz_0) = -v_0 * n \tag{3}$$

where **v** is a point on the plane (e.g. one of the triangle vertices). By using this information the term 1 can be reformulated to

$$\Delta(\bar{v}) = \sum_{p \in planes(v)} (p_i^T \bar{v})^2 \tag{4}$$

where $p = (a, b, c, d)^T$ for every plane i. Since $\bar{v}$ does not depend on **p**, it can be factored out of the sum:

$$\Delta(\bar{v}) = \bar{v}^T (\sum_{p \in planes(v)} p_i p_i^T) \bar{v}) = \bar{v}^T (\sum_{p \in planes(v)} K_p) \bar{v} \tag{5}$$

$\sum_i p_i p_i^T$ gives a 4x4 matrix K_{pi} which is the fundamental error quadric for the plane **p**:

$$K_{pi} = \begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix} \tag{6}$$

The matrix for the contraction $(v_1, v_2) \to \bar{v}$ is symmetric and approximated by Garland and Heckbert as

$$\bar{Q} = Q_1 + Q_2 = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{pmatrix} \tag{7}$$

where Q_1 and Q_2 are the error matrices for v_1 and v_2.

To find the optimal position for $\bar{v}$ can be found in the extreme of $\Delta(\bar{v})$ with

$$\frac{\partial \delta(\bar{v})}{\partial x} = \frac{\partial \delta(\bar{v})}{\partial y} = \frac{\partial \delta(\bar{v})}{\partial z} = 0 \tag{8}$$

which then can be formulated as a matrix

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \bar{v} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{9}$$

If the matrix is invertable - i.e. the determinant is not zero - the optimal point $\bar{v}$ can be found like following:

$$\bar{v} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{10}$$

Hence, the matrices are not always invertable, alternatives are needed. Gareland and Heckbert describe three different alternatives but only one is used in the lab. Alternative positions for $\bar{v}$ are v_1, v_2 and $(v_1 - v_2)/2$. The alternatives are chosen by the lowest error or cost for the new position. To calculate the cost following function is used:

$$\Delta(\bar{v}) = \bar{v}^T \bar{Q} \bar{v} \tag{11}$$

A simple modification of the cost heuristic was also implemented and tested. In case a scene uses a top-down view, only details which are on top of a model are necessary to render in higher quality. This means regions facing upwards maintain more detail than the rest. In order to implement this simple heuristic, the normal of the face which is associated to the half-edge of the vertex v_1 is fetched. The *y-value* of the face normal ranges between -1 and +1. If the normal has a positive y-value, they point in an upwards direction. Of course, not all positive y-values should be categorized as upwards because they still can point mostly horizontally. Therefore, a threshold t needs to be set. In the lab a threshold t of **0.3** was set. Once the threshold is met, a weight w is set. The weight is calculated by the *y-value* of the face times a constant c. This constant can be varied depending on the intended results. In the lab the constant c had a value of **10**. The cost calculated with the cost function 11 is then multiplied by the calculated weight.

Garland[2] also described a method to visualize the quadric error metric with isosurfaces. Let ϵ be an error value, then the isosurface can be described by

$$x^T Q x = \epsilon. \tag{12}$$

By using the Cholesky decomposition, **Q** can be factored into a symmetric positive definite matrix as

$$Q = R^T R \tag{13}$$

where **R** is an upper triangular matrix. **Q** is symmetric positive definite if and only if its isosurfaces are ellipsoids. Once this is the case, it is vizualized. It also then can be reformulated as following:

$$x^T Q x = x^T (R^T R) x = (Rx)^T (Rx) = \epsilon \tag{14}$$

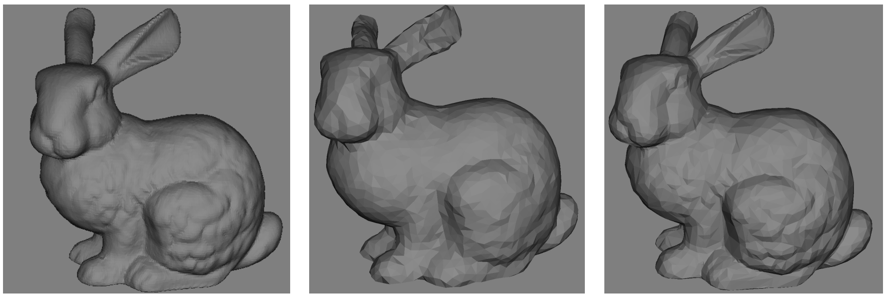
By letting **y = Rx** it can be noted that $y^T y = \epsilon$ is the equation of a sphere. Hence, **R** transforms the ellipsoid into a sphere of radius ϵ. The sphere then can be transformed into the appropriate isosurface by applying the transform R^{-1}.

2 Results

The following results will be shown based on different sample objects which were provided for the lab.

2.1 Decimation using the Quadric Error Metric

A demonstration of the decimation can be found in Figure 1. It includes a comparison between the decimation implemented in `SimpleDecimationMesh`, which always places the contracted vertex halfway between two contracted vertices based on the distance between the new position and the contracted vertex, and the newly implemented decimation `QuadricDecimationMesh`.

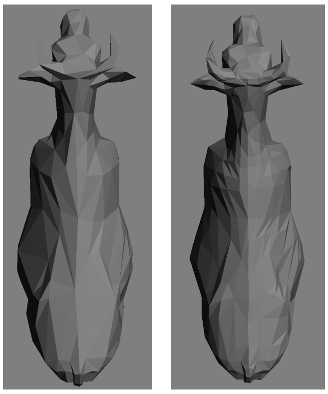


(a) Rabbit with 70064 faces (b) Rabbit with 5000 faces (simple decimation) (c) Rabbit with 5000 faces (quadric decimation)

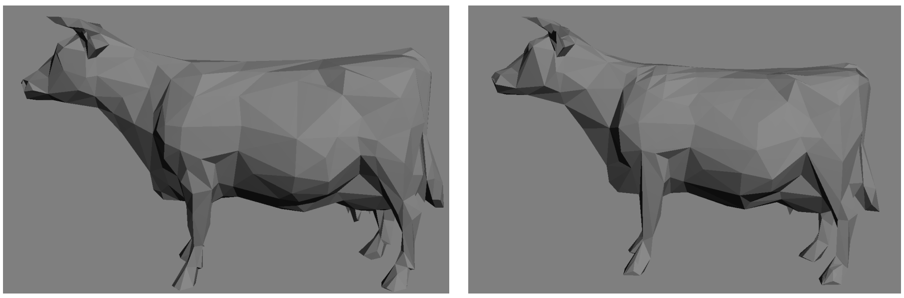
Figure 1: Decimation of the original model with the simple implementation, and with the quadric metric implementation

2.2 Alternative cost heuristic

The differences between the quadric heuristic and the altered heuristic can be observed in figure 1. Both the top-view and side-view is being shown with a decimation down to 1000 faces.



(a) Cow 1000 faces top-view (b) Cow 1000 faces top-view (alt. heuristic)



(c) Cow 1000 faces side-view (d) Cow 1000 faces side-view (alt. heuristic)

Figure 2: Decimation with an alternative heuristic (threshold *t = 0.3* and weight *w = 20*)

2.3 Alternative cost heuristic

The visualization of the quadrics calculated for the vertices in the cow model can be found in figure 3.

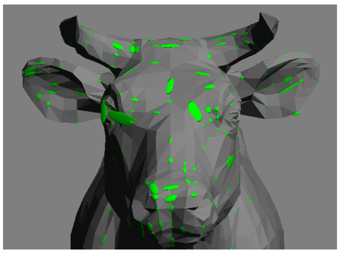


Figure 3: Visualization of the error quadric isosurfaces on the cow model

3 Conclusion

In general the quadric error is a good cost metric and heuristic for mesh decimation. If you compare the `SimpleDecimationMesh` implementation with the `QuadricDecimationMesh` implementation, you can see that it keeps more overall details when decimating. This is due to the adaptive heuristic. `SimpleDecimationMesh` uses a uniform heurisitc which treats each vertex the same based on a simple metric. This difference can be especially seen in figure 1.

The alternative cost heuristic introduced a simple way to focus the details on the upwards facing parts of the model rather than on the other parts. When comparing the pictures in figure 2 it can bee seen that the decimation down to 1000 faces keeps more details on the top/back of the cow. This simple heuristic achieves the desired effect but it might only be useful in some specific cases and also might need to be designed more sophisticated. However, it clearly shows how the metric can be adapted to create efficient mesh decimation based on specific use cases.

The visualization of the quadrics shows that points on flatter surfaces can be moved further than on pointy surfaces like in figure 3. This means movements on flatter surfaces costs less than in regions with more detail or points. Hence, the algorithm paired with the metric leaves more detail in regions where detail should be kept. It is also important to note that vertices can only be moved within the isosurfaces, otherwise the model will be destroyed.

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

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