LAB REPORT: LAB 1

TNM079, MODELING AND ANIMATION

Anna Wästling
annwa917@student.liu.se

Thursday 26th May, 2022

Abstract

This report describes the implementation of the half edge data structure which is one of many data structures used in computer graphics. The result of using the half edge data is discussed and physical attributes and artifacts are inspected.

1 Introduction

Data structures in computer graphics need to be able to handle polygon models efficiently. The most common structure to handle polygon models are by using vertices to build triangles. This makes fast rendering, however it give no easy access to information about neighbouring triangles. By adding more information to the data structure, by introducing edges, this can be achieved. There are many different data structures that considers this, the one implemented in this lab is the half-edge data structure.

2 Background

The half-edge data structure is built on the fact that every edge has two connected faces. Therefore the left face can access the right face through its *pair* as seen in figure 1 where the bold blue arrow is the current edge. In the half edge structure all the edges and vertices can be accessed from an edge through the next, previous and pair edge and the vertices connected to them. Each face has a connection to one of the edges.

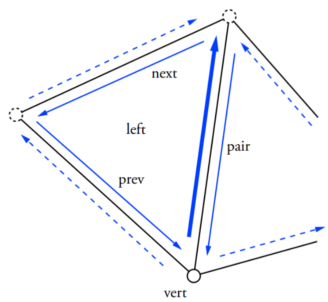

Figure 1: The half edge data structure, from [1]

Neighbor access was achieved by using the structure of half edge mesh and connecting every edges *pair*. One important feature is the 1 - ring neighbourhood of a vertex which lets us walk between vertices and edges. As seen in figure 2 all the vertices of a 1 - ring neighbourhood of v can be accessed by going through the edges. It is important to have easy access to the vertices for example to be able to calculate vertex normals. Vertex normals are used to avoid linear properties in the mesh when doing light calculations. This makes it possible to use for example *Phong shading* instead of flat shading.

The surface area was calculated for the mesh as the sum of the area of every face. Which was calculated by half the value of the cross product between two edges of the face.The mesh volume was calculated by using each face midpoint, normal and area.

One of the most important features of a mesh is the curvature. The curvature de-

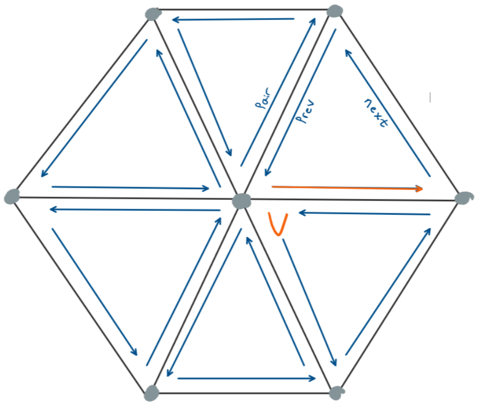

Figure 2: The 1 - ring neighbourhood of a vertex with the edges arranged counter clockwise

scribes how smooth a surface is. This is seen by how much the normal changes at a point when the points is moved along the surface. In this lab *Gaussian curvature* was implemented and visualised.

3 Tasks

3.1 Implement neighbor access

Implementing the 1 - ring neighbourhood was done by iterating through the edges to the faces connected to the vertex v. As seen in figure 2 one could visit all the vertices around v by going the path *prev* to *pair* to *prev* until the circuit was complete and all the vertices was visited. A 1 - ring neighbourhood was defined for each vertex in the mesh.

3.2 Calculate vertex normals

The normals of the vertices could then be calculated by averaging. The technique *Mean weighted equally (MWE)* and is defined as

$$n_{vi} = \widehat{\sum_{j \in N_1(i)}^{n} n_{fi}} \quad (1)$$

where n_{vi} is the sum of the normals in the 1-ring neighbourhood of vertex v_i, n_{vi} is then normalized.

3.3 Calculate surface area of a mesh

The surface area was computed by the sum of the areas of each face in the mesh.

$$A = \sum_{i \in S} \frac{1}{2} |(v_1 - v_2)_{fi} \times (v_3 - v_2)_{fi}| \quad (2)$$

Where v_1, v_2 and v_3 are vertex points for the i-th face in the mesh.

3.4 Calculate volume of a mesh

The volume was calculated by the dot product of the i-th face midpoint and the product of the normal and the area for the i-th face. This was computed for all the faces in the mesh and the sum computed the mesh volume as followed.

$$V = \frac{1}{3} (\sum_{i \in S} \frac{(v_1 + v_2 + v_3)_{fi}}{3} \cdot n(f_i) A(f_i)) \quad (3)$$

Where v_1, v_2 and v_3 are vertex points for the i-th face in the mesh, $n(f_i)$ is the normal for the face and $A(f_i)$ is the area of the face.

3.5 Implement and visualize Gaussian curvature

The Gaussian curvature is defined as

$$K = k_1 k_2 \quad (4)$$

Where k_1 and k_2 are two principal curvatures. To implement the Gaussian curvature the following formula was used

$$K = \frac{1}{A} (2\pi - \sum_{j \in N_1(i)} \theta_j) s \quad (5)$$

Where θ_j is the sum of angles calculated in the 1-ring neighbourhood at the vertex, N1(i) is the 1-ring neighborhood and A is the area of the faces in the 1-ring neighbourhood.

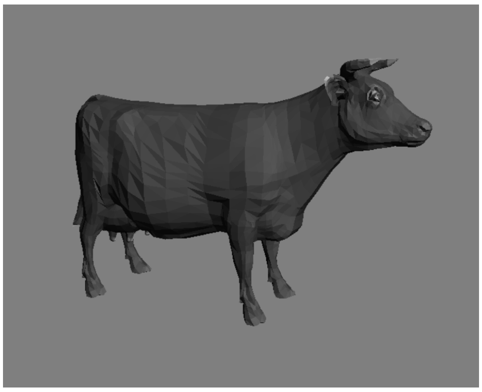

Figure 3: Face curvature

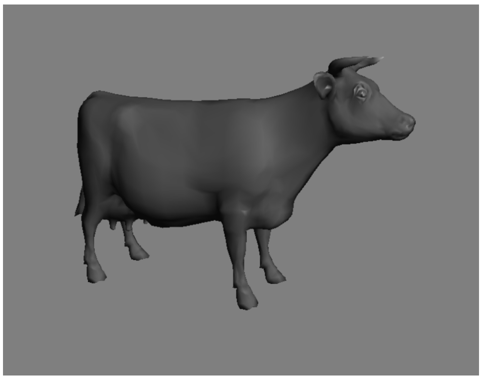

Figure 4: Vertex curvature

4 Results

The result of the lab is presented in this section. A sphere with the radius r=1.0 has an area of A=12.56637 and a volume of V=4.18879. Meanwhile the result of the lab gave an area of A=12.511 and a volume of V=4.15192. The difference between face curvature, where normals are specified for each face and using vertex curvature where the normals are specified for each vertex, can be seen in figure 3 and 4 respectively. The vertex and face normals can be visualised in figure 5 for a cube. The result of using Gaussian curvature can be seen in figure 6.

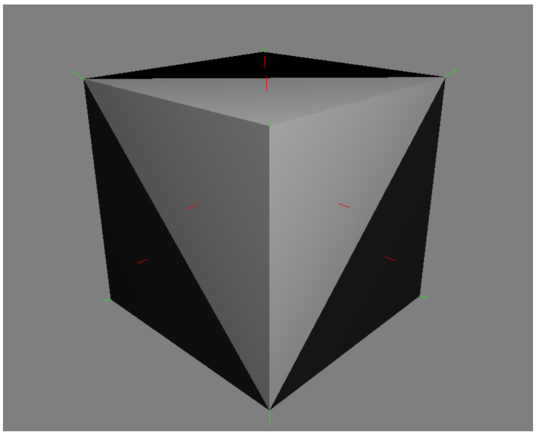

Figure 5: Visualisation of vertex normals (green) and face normals (red)

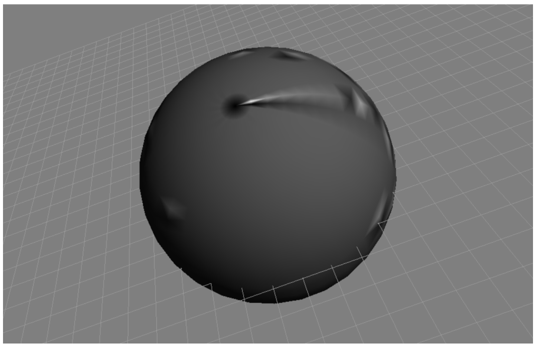

Figure 6: Gaussian curvature with mapping between colors in the interval [0.252891, 0.504374]

5 Conclusion

The artifacts seen in the figure 6 is the result of the sphere being built by triangles.

The half-edge mesh is very efficient when it comes to finding the neighboring vertices or faces which is noticeable in the execution time to calculate vertex normals. In comparison to a ordinary data structure, that needs to go through all vertices for each vertex, having access to neighbouring vertices seems to speed up the process.

The calculations of the area and the volume resulted in not the same as for a unit sphere. This is, again, the result of the sphere being built by triangles.

Using vertex normals instead of face normals allows for smoother shading while the model itself still contains the same amount of faces. This is a useful trick in computer graphics.

In the Gaussian curvature the mapping between the colors was in the interval [0.252891, 0.504374] for a sphere with the radii 1 where the curvature should be 1. This means that Gaussian curvature does not do an good job in estimating the curvature.

6 Lab partner and grade

The lab was done together with Tim Olsson and aims for grade 3.

References

[1] Robin Skånberg Mark E. Dieckmann and Emma Broman. "mesh data structures". 2021.