LAB REPORT: LAB 4

TNM079, MODELING AND ANIMATION

Lucas Hägg
lucha567@student.liu.se

Monday 10th June, 2024

Abstract

In this lab, the focus was on understanding and using implicit surfaces in modeling and animation, particularly through quadrics. The lab used basic geometry shapes like spheres and cylinders, manipulating them with boolean operations like union and intersection. A framework was developed to handle these shapes and their interactions. The main insight from the lab is the advantages of implicit surfaces, such as their ability to model without leaving holes or overlaps. A quadric method to compute the surface gradient were also performed, improving the understanding of the surfaces geometrical aspects.

1 Introduction

The main reason for doing this lab is to learn about implicit surfaces in computer graphics, especially for modeling and animation. Implicit surfaces are different from other 3D objects because they use math functions to describe their shape. This means that they do not have any risk for holes or parts that cross over each other. This lab uses tools to combine and change these shapes in different ways, showing a different way to make complex models using implicit surfaces.

2 Background

The first part of the lab was the implementation of Constructive Solid Geometry (CSG) operations for implicit surfaces. This included the tasks of setting up Union, Intersection, and Difference operations, each of which manipulates the spatial relationships between forms.

1. **Union:**
   The union function combines two surfaces such that the resulting surface covers all the space occupied by either.
2. **Intersection:**
   This function results in a surface representing the common volume occupied by both operands.
3. **Difference:**
   This operation models the subtraction of one surface that crosses another.

The second part of the lab was to implement mathematical functions for implicit surfaces.

1. **Value calculation:**
   This involves evaluating a point $p = (x, y, z)$ to determine if it lies on the surface, inside, or outside.
2. **Gradient calculation:**
   Computes the gradient at a point on the surface, which is crucial for rendering and physics calculations in graphics applications. With a quadric, the gradient is calculated as:

$$\nabla f = 2Qp \tag{1}$$

The last part of the lab was implementing quadric matrices for various quadric objects like planes, cylinders, ellipsoids, cones, and paraboloids to the scene. Each type of quadric surface is created by defining its specific quadric matrix and then used in the rendering process.

The equation defining the quadric surface can then be represented as:

$$\mathbf{p}^T Q \mathbf{p} = 0 \tag{2}$$

Where $\mathbf{p}$ are the coordinates:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3}$$

The quadric matrix Q is defined as:

$$Q = \begin{bmatrix} A & B & C & D \\ B & E & F & G \\ C & F & H & I \\ D & G & I & J \end{bmatrix} \tag{4}$$

The values in the matrix Q control the curvature, orientation, and position of the surface in 3D space. Different values lead to different types of quadric surfaces, making this representation flexible for geometric modeling.

3 Tasks

This chapter goes through how the tasks were implemented during the lab.

3.1 CSG operations

For a point $p = (x, y, z)$, the function value $f(p)$ determines whether the point lies inside, outside, or on the implicit surface:

1. Inside the surface: $f(p) < 0$
2. Outside the surface: $f(p) > 0$
3. On the surface: $f(p) = 0$

Given two implicit surfaces, the first step was to transform the point p to object coordinates. After that, the value of the point p was evaluated for the two implicit surfaces. For the returned values left and right, the CSG operations were implemented as following:

1. Union: $min(left, right)$.
2. Intersection: $max(left, right)$
3. Difference: $max(left, -right)$.

If the returned value is close to zero, the point lies on the surface.

3.2 Value Calculation

The **Quadric::GetValue()** function was implemented as following:
Given a point $p = (x, y, z)$:

1. Converted the world coordinates to object coordinates
2. Applied equation 2, using the dot product with $glm :: dot(p, Q * p)$ where Q is the quadric matrix.

If the returned value is close to zero, the point lies on the surface.

3.3 Gradient Calculation

The **Quadric::GetGradient()** function was implemented as following:
Given a point $p = (x, y, z)$:

1. Converted the world coordinates to object coordinates
2. Applied equation 1 using the quadric matrix Q and the point p.
3. Returned the values x, y, z from the result of the equation.

If the returned value is close to zero, the point lies on the surface.

3.4 Quadric surface matrices

Matrices were implemented for planes, cylinders, spheres, ellipsoids, cones, paraboloids and hyperboloids. For example, the plane equation:

$$Ax + By + Cz + D = 0 \tag{5}$$

used with the quadric in 4, and equation 2, gives:

$$Q = \begin{bmatrix} 0 & 0 & 0 & \frac{a}{2} \\ 0 & 0 & 0 & \frac{b}{2} \\ 0 & 0 & 0 & \frac{c}{2} \\ \frac{a}{2} & \frac{b}{2} & \frac{c}{2} & d \end{bmatrix} \tag{6}$$

In code, this was applied by creating an empty 4D matrix and updating corresponding values manually.

4 Results

The result of the CSG operations are demonstrated using two implicit spheres.

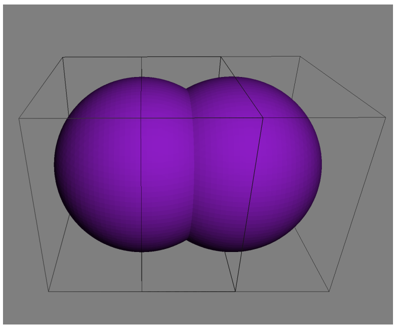
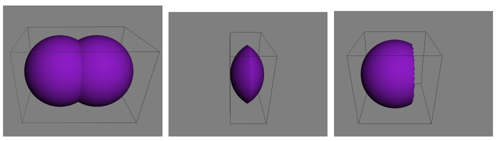


Figure 1: Two implicit spheres, overlapping each other.

Below are the result of the three operations:



(a) (b) (c)

Figure 2: (a) Union, (b) Intersection, (c) Difference

Next are the result of the quadric surface matrices, as well as the gradiant calculation, demonstrated with a plane.

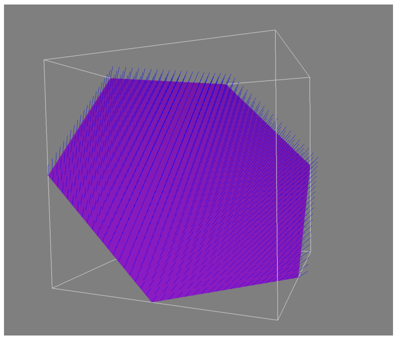


Figure 3: The resulting plane of the quadric Q from equation 6, with visualized gradients in the color blue.

5 Conclusion

The results from applying the CSG operations to two spheres demonstrate a practical application of implicit surface manipulations that was explored in this lab. Each operation visually show the theoretical expectations and confirms the mathematical principles.

6 Lab partner and grade

Lab partner was Kenton Larsson and the grade I am for is 3.

References

[1] Mark Eric Dieckmann. *MESH DATA STRUCTURES*. Linkoping university, 2023.