LAB REPORT: LAB 4

TNM079, MODELING AND ANIMATION

Linnea Bergman
linbe810@student.liu.se

Saturday 21st July, 2018

Abstract

This lab implemented a simple yet effective framework for implicit geometries. The framework contains operators for Constructive Solid Geometry, abbreviated CSG. Operators such as union, intersection and difference was implemented. The system were restricted to analytical surfaces such as quadrics and torii.

1 Introduction

The functionality and how to implement different Constructive Solid Geometry, abbreviated CSG, operators is a part of this lab, as well as quadric surfaces, gradients and curvature. CSG operators simplifies the work for many people across the world since they can use these operators to get the disired result instead of doing it by hand. The same goes for the quadric surfaces. Quadric surfaces is one way to easily get the desired surface instead of modelling it by themselves.

2 Assignments

The files used in the following sections are Implicit.cpp, CSG.h, Sphere.cpp, Quadric.cpp and FrameMain.cpp. It will cover the implementation of CSG operators, quadric surfaces and the discrete gradient operator for implicit.

2.1 Assignment Implement CSG operators

To start with the GetValue-function for the Union, Intersection and Difference found in file CSG.h was modified. All of these use Boolean operators.

$$\textbf{Union}(A, B) = A \cup B \qquad = min(A, B) \tag{1}$$
$$\textbf{Intersection}(A, B) = A \cap B \qquad = max(A, B) \tag{2}$$
$$\textbf{Difference}(A, B) = A - B \qquad = max(A, -B) \tag{3}$$

A and B represent the two objects that are operated with. Union was implemented and computed with the following Equation 1, intersection with 2 and difference with 3.

In the GetValue() in the Union, the coordinates (x, y, z) was the input parameters which was transformed from world space coordinates into object space. It was a necessary step to do since the CSG operators are implicit geometries and are therefore transformed the same way as the implicit surfaces. Then the values of the left and right child was used in the Boolean operation, Equation 1, which is the minimum value of the two. The value of the Boolean operators was returned.

The Intersection and Difference works the same way except for the Boolean operation. In Intersetcion the maximum value of the left and right child, Equation 2, was returned. In Difference the maximum value of the left and the negative right child, 3, was returned.

2.2 Assignment Implement the quadric surface

In this assignment the functions GetValue and GetGradient in Quadric.cpp was implemented with the use of Equation 4 and 5.

$$\nabla f(x,y,z) = 2\begin{bmatrix} \mathrm{Ax + By + Cz + D} \\ \mathrm{Bx + Ey + Fz + G} \\ \mathrm{Cx + Fy + Hz + I} \end{bmatrix} = 2\begin{bmatrix} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \\ \mathrm{B} & \mathrm{E} & \mathrm{F} & \mathrm{G} \\ \mathrm{C} & \mathrm{F} & \mathrm{H} & \mathrm{I} \end{bmatrix}\begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \\ 1 \end{bmatrix} \tag{4}$$

$$f(x,y,z) = f(\mathbf{p}) = \mathbf{p}^T\mathbf{Q}\mathbf{p} \tag{5}$$

The matrix represent the function for the $\mathbf{p}^T\mathbf{Q}\mathbf{p}$ surface and $\mathbf{p}$ the point for the given coordinates.

In GetValue the given x, y and z coordinates are transformed from world space into object space. To evaluate the matrix of the quadric Equation 5 was used and the value was returned from the function, were $\mathbf{p}$ consisted of the transformed coordinates.

The GetGradient-function uses the quadric matrix to evaluate the gradient. Given x, y and z, the coordinates were transformed from world space into object space. With Equation 4, which computes the normal of the given point, the return value was calculated. To visualize the gradient choose a mesh in the program and use the "Gradient" visualization mode.

Different shapes and meshes have different quadric matrices. In FrameMain.cpp the quadric matrix for planes, cylinder, spheres and ellipsoids, cones, paraboloids and hyperboloids was added. The matrices was derived from the analytical expression of each shape. Following are the unique cases of the analytical expression.

- Planes

$$f(x,y,z) = ax + by + cz = 0 \tag{6}$$

- Cylinders

$$f(x,y,z) = x^2 + y^2 - 1 = 0 \tag{7}$$

- Spheres and ellipsoids

$$f(x,y,z) = x^2 + y^2 + z^2 - 1 = 0 \tag{8}$$

$$f(x,y,z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0$$

- Cones

$$f(x,y,z) = x^2 + y^2 - z^2 = 0 \tag{9}$$

- Paraboloids

$$f(x,y,z) = x^2 \pm y^2 - z = 0 \tag{10}$$

- Hyperboloids

$$f(x,y,z) = x^2 + y^2 - z^2 \pm 1 = 0 \tag{11}$$

From these Equations the respective matrix component was computed.

2.3 Assignment Implement the discrete gradient operator for implicit

In the file Implicit.cpp, three Get-functions exists and in the assignment GetGradient was modified. Input to the function was the x, y and z coordinates, given in world space.

$$D_x \equiv \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h} \approx \frac{f(x_0 + \epsilon) - f(x_0)}{\epsilon} \tag{12}$$

Equation 12 is an approximation of the gradient, where ϵ represents a sufficiently small value.
In GetGradient the x, y and z coordinates are given. With the use of these coordiantes Equation 12 was computed three times, one for each axis. The returned value was a vector of type "Vector3" that contained the computed result for each axis. To visualize it, use "Gradients" visualization mode in the GUI to view the gradient at the surface.

2.4 Assignment Implement the discrete curvature operator for implicit

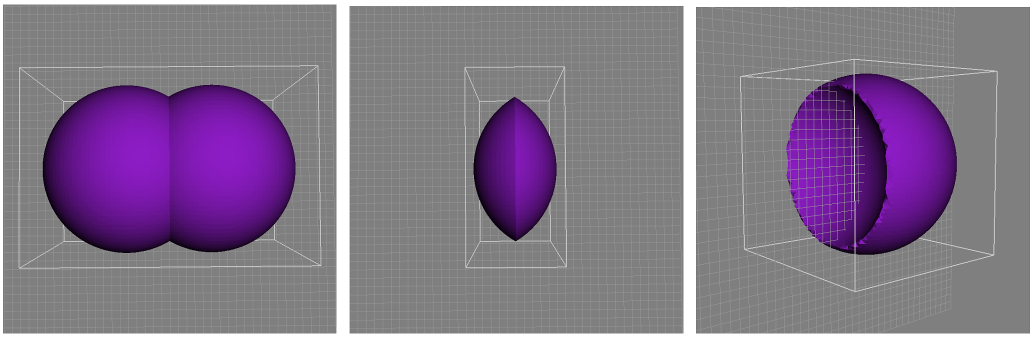
The function GetCurvature in Implicit.cpp was implemented with an approximation equation.

$$D_x \equiv \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h} \approx \frac{f(x_0 + \epsilon) - f(x_0 - \epsilon)}{2\epsilon} \tag{13}$$

Equation 13 is the central difference approximation of computing the curvature. ϵ is a sufficiently small value.
GetCurvature uses Equation 13 three times to compute the curvature in each direction, one for $x-, y-$ and $z-$axis. The result for each axis were returned as a vector of type "Vector3".
To visualize the curvature use the "Curvature" visualization mode with "Jet" or "HSV" colourmap on.

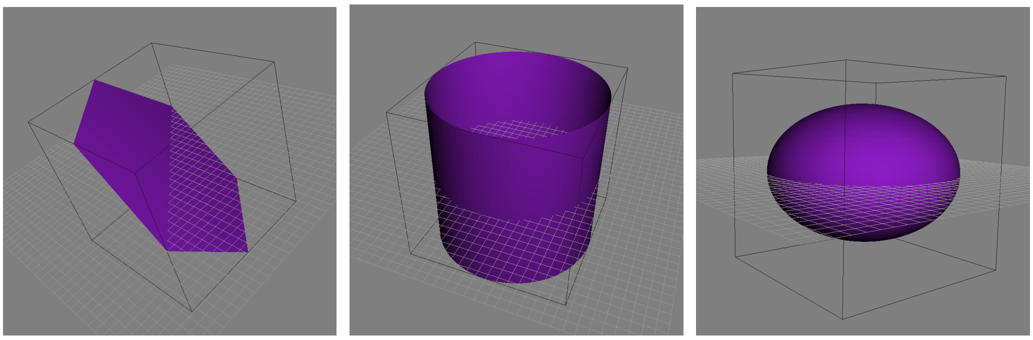
3 Results

The effect of the three Constructive Solid Geometry (CSG) operators is shown in Figure 3. To the far left 1(a) the union between two spheres can be seen, in the middle 1(b) is the intersection and to the far right 1(c) the difference in shown.

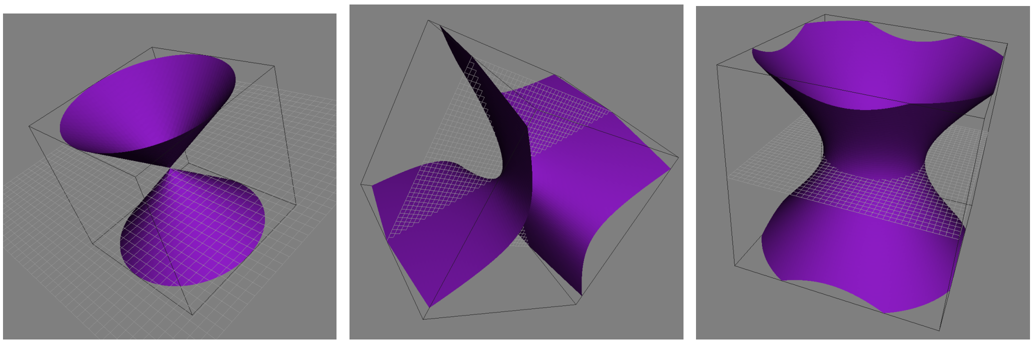


(a) Union between two spheres (b) Intersection between two spheres (c) Difference between two spheres

Figure 1
Result from assignment 2.1 showing the three CSG operators between two identical spheres, with the Union 1(a), Intersection 1(b) and Difference 1(c) in that order from left to right.

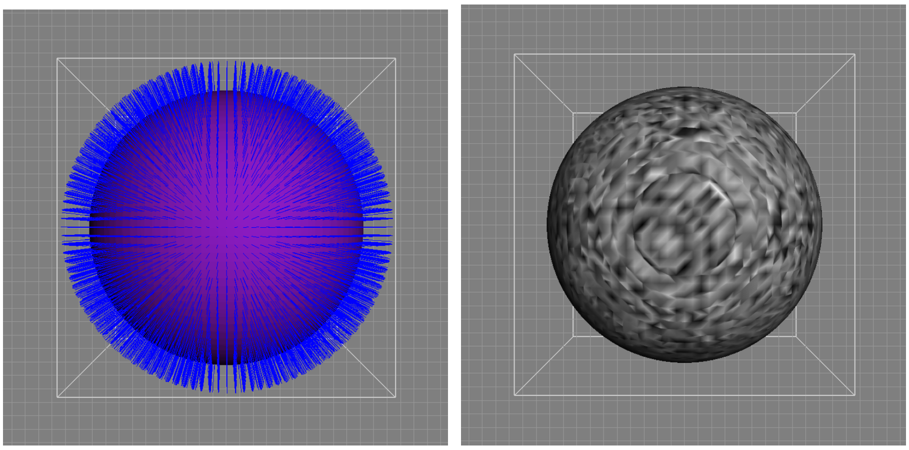


(a) Implicit surface: Plane (b) Implicit surface: Cylinder (c) Implicit surface: Ellipsoid



(d) Implicit surface: Cone (e) Implicit surface: Paraboloid (f) Implicit surface: Hyperboloid

Figure 2
Result from assignment 2.2 showing a variety of implicit surfaces. Top row: Plane to the left 2(a), cylinder in the middle 2(b) and ellipsoid to the right 2(c). Bottom row: Cone to the left 2(d), paraboloid in the middle 2(e) and hyperboloid to the right 2(f).



(a) Gradient of sphere (b) Curvarute of sphere

Figure 3
Result from assignment 2.3 is shown to the left in 3(a) and the result for assignment 2.4 to the right 3(b).

For assignment 2.2 the various quadric surfaces can be seen in Figure 2. Tbe surfaces are the following; plane 2(a), cylinder 2(b), ellipsoid 2(c), cone 2(d), paraboloid 2(e) and hyperboloid 2(f).

Figure 3(a) displays the discrete gradient for the implicit surface. The blue lines on the sphere shows the magnitude and direction of the gradient of the different points on the surface.

Lastly, the result for the discrete curvature operator is visualized in Figure 3(b). The grayscale gradient represents the steepness of the curvature on that specific point and the pattern the direction.

4 Conclusion

The CSG operators are a good tool for manipulating or managing two or more objects at the same time. There is room for improving the difference operator since the new edges are not very smooth. For a smoother surfacse a smaller sampling difference could be used. A sampling distance gives a higher accuracy when triangulating the mesh and changing the shape.

For the quadric surface all meshes in Figure 2 uses the same basis but varying functions $f(x, y, z)$ to define how the surface should behave. The size and shape depends on the coefficients used in the quadric matrix.

Figure 3(a) shows the gradient for every point on the surface. If ϵ changes in Equation 12 then the length of the gradient changes as well. The maximum length that the gradient can be are one. When ϵ is significantly small then the gradients goes towards one. A larger ϵ results in shorter length of the gradient lines.

The curvature for every point on the surface can be seen in Figure 3(a). Similar to the gradient, the curvature behaves differently if ϵ is altered in Equation 13. It is the same principle as the gradient but instead of the length being altered it is the colour and pattern that visualize the curvature that changes. For a small ϵ the colours will be brighter and a larger ϵ results in darker colours.

5 Lab partner and grade

This lab was done together with Rebecca Cedermalm, rebca973. We did all the assignments for grade 3 and 4 plus half of the assignments for grade 5. Since all tasks for grade 4 and task 5a was completed I think grade 5 is fair enough, according to what is written in the lab instructions for grades.