LAB REPORT: LAB 4

TNM079, MODELING AND ANIMATION

Viktor Sjögren
viksj950@student.liu.se

Sunday 23rd May, 2021

Abstract

We discuss surfaces in their implicit form. This implies an indirect approach that requires computations of an equation that describes the surface to be solved to approximate points on a mesh surface. The advantage of this representation is mainly the physical accuracy that it offers, making it suitable for simulations. Implementation of three boolean operators, union, intersection and difference using simple min and max operations have been done and is discussed. These operations allows for simple implicit surfaces to be combined and morphed together to create more complex surfaces, while keeping the representation on essentially the same complexity level. Six fundamental quadric surfaces has been constructed and added to a scene to show some of the potential that implicit representation offers.

1 Background

Implicit surface representation are surfaces that are represented by an indirect approach in comparison to explicit surfaces. In the explicit representation, each polygon defines the surface. For implicit surfaces, equations that describes the surface layout is instead used. This means that the equation has to be solved in order to get the points in which the surface is located. A straight forward example, like a circle, would be defined implicitly as:

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

From the implicit representation, like seen in equation 1, a *level-set* can be defined as $f(x,y) = C$. The C represents then all the possible contours of the implicit mesh. Typically, where $C = 0$ is the most interesting to look at. It also possible from the properties of implicit surfaces to identify where points are located, based on the iso-value C. The properties are defined as:

$$\begin{aligned} Inside : f(x) &< C \\ Outside : f(x) &> C \\ On : f(x) &= C \end{aligned} \tag{2}$$

Implementation of the first task was done in the class *CSG.h*. The idea was to implement Boolean operations that allows for a combination of simple implicit meshes to construct more complex meshes that retains the implicit representation. The operations *union, intersection* and *difference* were implemented. By the property declared in (2), the way these operations can be defined is quite simple. Given two implicit meshes, A and B, the operations could be implemented using the following equations:

$$\begin{aligned} Union(A,B) &= A \cup B = min(A,B) \\ Intersection(A,B) &= A \cap B = max(A,B) \\ Diff(A,B) &= A - B = max(A,-B) \end{aligned} \tag{3}$$

As mentioned, the principal of why these operations are defined this way is by the prop-

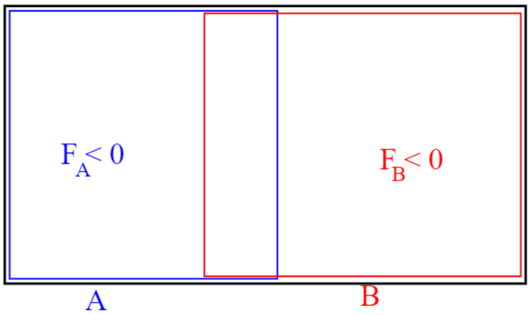


Figure 1: Union operator with two primitives, A and B.

erty in (2). A visualisation of the union operation can be seen in Figure 1, where everything that is within object A, has a function value less than zero and likewise for object B.

The second task aimed at constructing quadric surfaces, which was implemented in *Quadric.cpp* and *FrameMain.cpp*. In *Quadric.cpp* a function responsible for obtaining the values on a quadric in object space. As the quadric is defined by a 4x4 coefficient matrix, the value can be computed as:

$$p^T Q p = [\, x\ y\ z\ 1] \begin{bmatrix} A & B & C & D \\ B & E & F & G \\ C & F & H & I \\ D & G & I & J \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4)$$

where p is the point which has been transformed from world space to object space.

To compute the normal for points on the quadric surface, we know that we can use the gradient as it is parallel to the normal. With the same matrix principle as in equation 4, the gradient is obtained through:

$$2Q_{sub} p = 2 \begin{bmatrix} A & B & C & D \\ B & E & F & G \\ C & F & H & I \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (5)$$

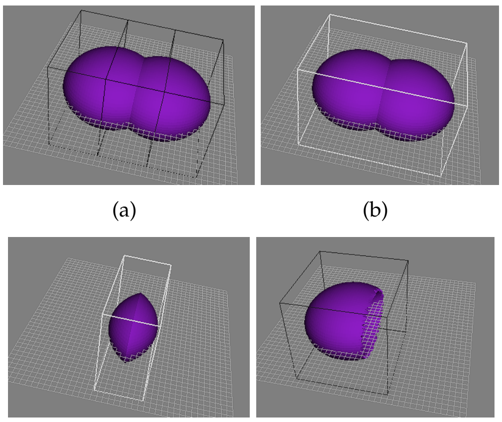
The quadric surfaces themselves was then implemented in *FrameMain.cpp*. Implementing for example the plane we use equation 4 to obtain the implicit function to construct the plane. In that case the quadric plane is defined as:

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

With the coefficient matrix, this is fulfilled if $D = G = I = 1$. Note that this means only the G and I in the last row and the D in first row.

2 Results

The results of the boolean operators can be seen working in Figure 2. In 2(a) two implicit meshes in the form of ellipsoids are added to the scene, no operations have been made. In Figure 2(b) the two meshes are unified using the union operator. As the shape of the meshes are identical the result looks essentially the same, however it is classified as one singular implicit object after the union operation. The bottom-left image, 2(c) shows the result after intersection between the two ellipsoids. Lastly, the difference operation can be seen in Figure 2(d), where the left ellipsoid was selected first.



(a) (b) (c) (d)

Figure 2: Results when performing the union, intersection and difference operator on two implicit ellipsoids. CSG blending was not used to generate these results.

By using blended CSG (Construcitve solid geometry) the result is the same but the edges between the relevant objects are smoother and we obtain a more pleasing result. In Figure 3 the same operation with the union operator as in 2(b) but with CSG blending turned on is seen.

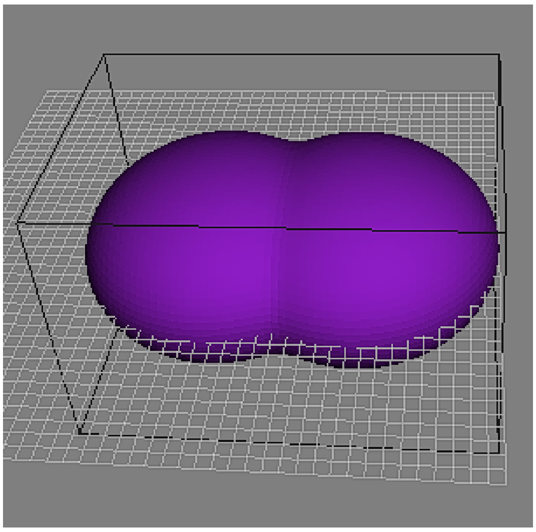
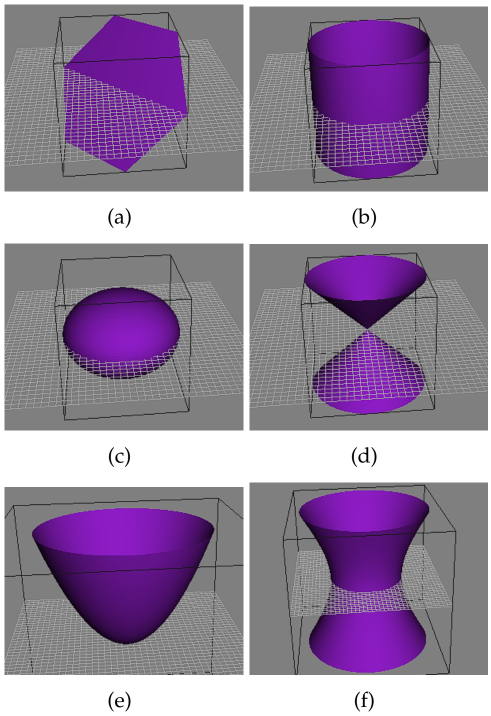


Figure 3: Union operator with CSG blending enabled.

The six implemented implicit quadric meshes *Plane, Cylinder, Ellipsoid, Cone, Paraboloid* and *Hyperboloid* can be seen added into a scene in Figure 4. Verification of the gradient computation was done by visualising the gradient vectors that extends from the implicit meshes, an example can be seen in Figure 5. For both the *GetValue* and *GetGradient* computations the transformation of the world coordinates to object space coordinates was done each time the function was called. Another option is to transform the coefficient matrix only once instead.



(a) (b)

(c) (d)

(e) (f)

Figure 4: Resulting implcit quadric meshes, determined coefficeints using equation **??**

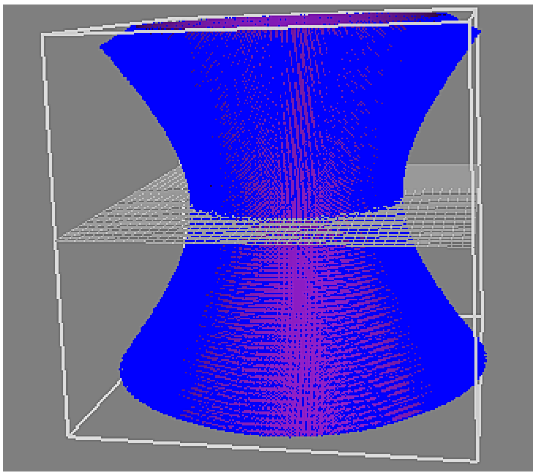


Figure 5: Quadric hyperboloid with gradients visualised

3 Lab partner and grade

The lab was made in collaboration with Algot Sandahl. The report scope and content is aimed at grade **3**.

References

[1] M.E. Dieckmann. *Lecture 9* , tnm079, 2021.