

Lab 4 Report

Implicit Solids

SSR3

Yuhan Zhang | CZ2003 Computer Graphics and Visualization | 30 October 2019

Implicit Solids

|  |  |
| --- | --- |
| **Component 1** | **Notes** |
| Component 1 is created from bounding the y domain of a cylinder with radius 0.3 defined by the equation cyl=0.3^2-x^2-z^2. It is bound by two plane-bounded half-space defined by functions b2=-0.3-y and b3=y+0.95. The operation min is used to bound the cylinder. This is shown in the file ‘component\_1.wrl’ | Each implicit function is defined as f(x,y)>=0 or f(x,y,z)>=0. If the cylinder is not bounded by the half-spaces, the cylinder would stretch in the y direction until it hits the bounding box, where the rest of the cylinder is cut off. By using the min function, we are able to specify where the cylinder is bounded, thus manipulating its size and location. |
| **Component 2** | **Notes** |
| Component 2 is created from bounding the y domain of a cone defined by cone=  ((y-0.5)\*0.5)^2-(x)^2-(z)^2. It is displaced by 0.5 along the y axis. It is bound by two plane-bounded half-space defined by functions b1=0.5-y and b4=y+0.3. The operation min is used to bound the cone. This is shown in the file ‘component\_2.wrl’ | If the cone is not bounded, the base of the cone would expand as y decreases (from y=0.5) and the base of another cone opening towards the opposite direction as the bottom cone would expand as y increases (from 0.5). As observed the Component 2, the cone is bounded exactly at y=0.5, which is the distance the cone was displaced from the origin. This way, only one cone is shown. |
| **Component 3** | **Notes** |
| Component 3 is ellipsoid defined by the equation ellip=1-x^2/0.4^2-(y-0.6)^2/0.2^2-z^2/0.3^2. It is displaced by 0.6 along the y axis. This is shown in the file ‘component\_3.wrl’ | No plane-bounded half-spaces were necessary to create an ellipsoid. This is because the equation of an ellipsoid already determines all the necessary parameters to create the ellipsoid. Similarly, no min or max functions are used unless a different shape is desired (modifications to the ellipsoid). |
| **Component 4** | **Notes** |
| Component 4 is created by bounding the z domain of a cone defined by equation cone1=(z\*0.5)^2-(x)^2-(y+0.7)^2. It is displaced by 0.7 along the negative y axis. It is bound by two plane-bounded half-space defined by functions zb=z and zb1=0.3-z. The operation min is used to bound the cone. This is shown in the file ‘component\_4.wrl’ | Please refer to notes for component 2. |
| **Component 5** | **Notes** |
| Component 5 is a half-space defined by equation b0=1+x/0.9+y/0.8+z/0.9. This is shown in the file ‘component\_5.wrl’. | Component 5 illustrates a plane-bounded half-space. The corners of the half-space appears to be cut off because it reached the boundary of the bounding box of size [2 2 2]. |
| **Solid 1** | **Notes** |
| Solid 1 is created by using the max operation to join components 1,2,3 and min operation to bound it with components 5 and subtract component 4. This is shown in the file ‘CSGsolid\_original.wrl’. | The max function is used to join the first 3 components together and the min function bounds the solid to one side of the half-space defined in Component 4. The min function is also used to subtract component 5 from the solid. |

Implicit Solid: Bounding Box and Resolution

|  |  |
| --- | --- |
| **BB** | **Notes** |
| Above shows the smallest possible bounding box for Solid 1. The bboxCenter is set at 0 0 0 and bboxSize is set at 0.8 1.9 0.8. | Since the largest absolute value of the y value reached by the solid is -0.95, the y parameter for the bounding box is must be minimum 1.9 in order to not cut off any parts of the solid. Similarly, the largest absolute value of the x and z values reached by the solid are both 0.4, so the x and z parameters of the bounding box must be minimum 0.8 in order to not cut off any parts of the solid. |
| **Resolution 1** | **Notes** |
| Above shows Solid 1 at resolution set to [30]. This is shown is file ‘Resolution\_1.wrl’. | At resolution set to [30], there are evidently rough edges on the solid, especially on curves or slanted edges. This can easily be observed in Resolution 1. |
| **Resolution 2** | **Notes** |
| Above shows Solid 1 at resolution set to [100]. This is shown is file ‘Resolution\_1.wrl’. | With resolution set to [100], there are significant improvements to the illustration of the solid as compared to Resolution 1. This can be observed in Resolution 2. However, the are still some observable roughness on the ellipsoid component of the solid. |
| **Resolution 3** | **Notes** |
| Above shows Solid 1 at resolution set to [150]. This is shown is file ‘Resolution\_1.wrl’. | With resolution set to [150], the solid is very clearly illustrated. The roughness on the ellipsoid is barely observable as compared to Resolution 1 and 2. With an even higher resolution, the roughness would inevitably become unnoticeable. However, rendering the solid with resolution higher than 150 would cause the running time to exceed 5 seconds, thus it won’t be shown in this report. |

Implicit Solid: Color

|  |  |
| --- | --- |
| **Color 1** | **Notes** |
| Above shows a color pattern specified by 3 simple algebraic equations r=(u-v+1.35)/2.7, g=((v+w)\*0.6+1.5)/2.35, and b=(w\*0.9+0.9)/1.5. This is shown in file ‘CSGsolid\_color1.wrl’ | The x,y,z parameters of the solid are [-0.4 0.4], [-0.95,0.8], and [-0.4,0.4] respectively. The rgb values while defining the function for color must be in the range of [0 1]. Because of this, the algebraic functions for each rgb element is specifically chosen to never exceed the boundaries for Solid 1. As observed from the equations, and Color 1, the value for red increases as x increases and y decreases. Similarly, the value for green increases as y and z increases. Finally, the value for blue increases as z increases. |
| **Color 2** | **Notes** |
| Above shows a color pattern specified by equations r=abs(sin(4\*pi\*abs(u))), g=abs(sin(4\*pi\*abs(v))), and b=abs(sin(4\*pi\*abs(w))). This is shown in the file ‘CSGsolid\_color2.wrl’. | Instead of using algebraic functions, Color 2 uses sine functions to determine the colors. Since the range of sine functions are between -1 and 1, the absolute value of a sine function will always be between 0 and 1. As observed in Color 2, the sine functions oscillate between 0 and 1 in a fixed period, creating the checkboard type pattern as the colors cross over each other. |