

New Features implemented

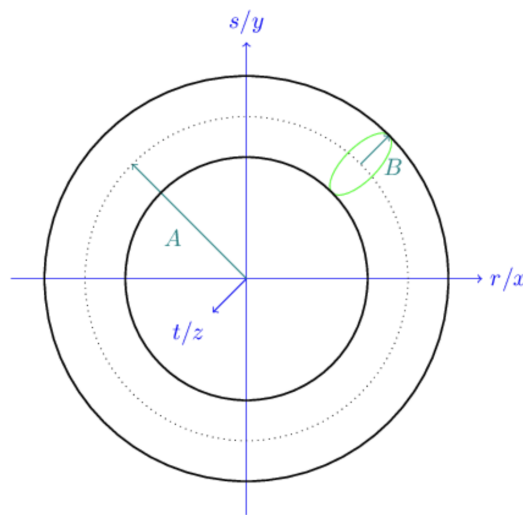
1. New Primitive Shape: Cone, Torus

a. Cone implementation:

- i. Initialization: For cones, we define a cone as its center(which is the tip of the cone); the center axis of the cone; the cosa, which is the cosine of the angle between the center axis of the cone and the outer cone side; and the height of the cone; and the material.
- ii. Intersection of the ray and the cone: We identify the smallest t of the intersection by applying the cone function and solving the quadratic equation like spheres but with different parameters. And then we determine whether the ray has a smallest t or is no hit by comparing its determinant and the ray.start and ray.end. One thing to note here is that we also have to make sure that the angle a is also less than 90 degrees or else that wouldn't yield a correct cone.

b. Torus implementation:

- i. Initialization: For torus, we define a torus as its center; its torus radius, which is denoted as A in the picture; and its outer circle radius, which is denoted as B in the picture.
- ii. Intersection: Torus is an implicit surface. Therefore, we need to solve the implicit function in order to get the intersection of the ray and the torus. A ray comes and intersects the torus four times at most so it is a quartic equation. Therefore we do some math here and solve the equation giving four solutions which some might be imaginary numbers. So we process the four output and identify the minimum t that we need.



2. CSG of sphere intersection: We implemented the intersection of spheres where it is embodied in our ufo plate. The intersection is basically implemented by comparing the t in the Scene intersect function and checking whether the t that we had was the t that is embodied in the intersection and not just the minimum t of every surface.
3. Refraction of transparent shading:
 - a. The refraction

- i. Refraction could be separated into Fresnel effects and refraction itself. I defined the function `refraction`, `fresnel`, and used it within the shading with adding colors. I used this paper and the textbook for reference to make the transparent material:

https://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf

4. Texture mapping

Step 1: Load texture image as SGB

Step 2: Map 3D image coordinates to 2D texture coordinates using texture coordinate functions for each shape

- a. Spherical mapping (For planets)

$$\phi(x, y, z) = ([\pi + \text{atan2}(y, x)]/2\pi, [\pi - \text{acos}(z/\|x\|)]/\pi).$$

- b. Triangle mapping (For starry sky background)

$$\phi(x, y, z) = (\tilde{u}/w, \tilde{v}/w) \quad \text{where} \quad \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ * \\ w \end{bmatrix} = P_t \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

Step 3: Lookup texture color given 2D texture coordinates

This function figures out the location, in the coordinate system of the texture image, that corresponds to the shading point, and it reads out the color at that point in the image, resulting in the texture sample.

Step 4: Edit shade function to set diffuse coefficient of hit material to the texture color

That color is then used in shading, and since the texture lookup happens at a different place in the texture for every pixel that sees the floor, a pattern of different colors shows up in the image.

- 5. Antialiasing: We also wrote antialiasing which took a mean of the surrounding pixels in the rendering function to make the image look smoother.

Scene design

Our work is inspired by the bagel multi-universe from the movie *Everything Everywhere All at Once*.

The main, largest object is a black-colored bagel made with an implicit-shape primitive torus, placed in the center of the image.

On the right, there is a flying UFO with transparent glass shell and CSG-made base, with a beam of cone-shaped light shining from the bottom of the UFO.

On the left, there are some planets being pulled into the hole of bagel by its magnetic force, which are implemented with texture mapping. These planets include the Earth, violet planet, and orange planet.

There are also some grey reflective triangle debris and spherical shiny white particles dragged into and through the hole of the bagel as well.

After trials, there are point lights placed at the center of the space shining from back to front, and a less intense slightly yellow light at the right front, and a slightly blue light shining from bottom of the UFO.

In order to render the image, run `python three_spheres.py`