Oregon State University

Project #4: Cube Mapping Reflective and Refractive Surfaces

Hyuntaek Oh

ohhyun@oregonstate.edu

Due: Feb. 12, 2025

1 Description

1.1 Set up

GLSL and glman are used to create a reflective and refractive display of a **bump-mapped** 3D object with cube-mapping in the project #4.

1.2 Program Description

Six images such as KEC or NVIDIA for cube map are required to build the background wall, and a vase is used as a 3D object. For reflective and refractive display, $uMix$ is used to show the effect of blending the reflective and refractive versions of the scene. $uEta$ is an uniform variable to represent an index of refraction. Given the function PerturbNormal3() implement the bump-mapping on the 3D object.
In the fragment shader, vNormal and vEyeDir, vector from the eye position to the point, are assigned to variable Normal and Eye respectively:

$$\text{vec3 Normal} = \text{vNormal};$$

$$\text{vec3 Eye} = \text{vEyeDir};$$

For reflectVector and reflectColor, we use reflect and texture functions:

$$\text{vec3 reflectVector} = \text{reflect(vEyeDir, Normal)};$$

$$\text{vec3 reflectColor} = \text{texture(uReflectUnit, reflectVector).rgb};$$

Unlike the reflect vector, refract vector needs one more parameter $uEta$, an index of refraction, and refract function can be used to show the direction of the refract vector:

$$\text{vec3 refractVector} = \text{refract(vEyeDir, Normal, uEta)}$$

After that, gl_FragColor mix the color of reflectColor and refractColor. So, it would be:

$$\text{vec3 color} = \text{mix(refracColor, reflectColor, uMix)};$$

$$\text{color} = \text{mix(color, WHITE, uWhiteMix)};$$

$$\text{gl\_FragColor} = \text{vec4(color, 1.)};$$

1.3 URL

Video Link(bitly): `https://bit.ly/4aTBDwQ`

Video Link(original):
`https://oregonstate.zoom.us/rec/share/L8HTgWMJbYLHLr--EnL32CRw4oytwyFega1HUHHsPMQMHqGm-nz4vG-oVmfdTu7f7SueLYNwkM6?startTime=1739039908000`

1.4 Test Result

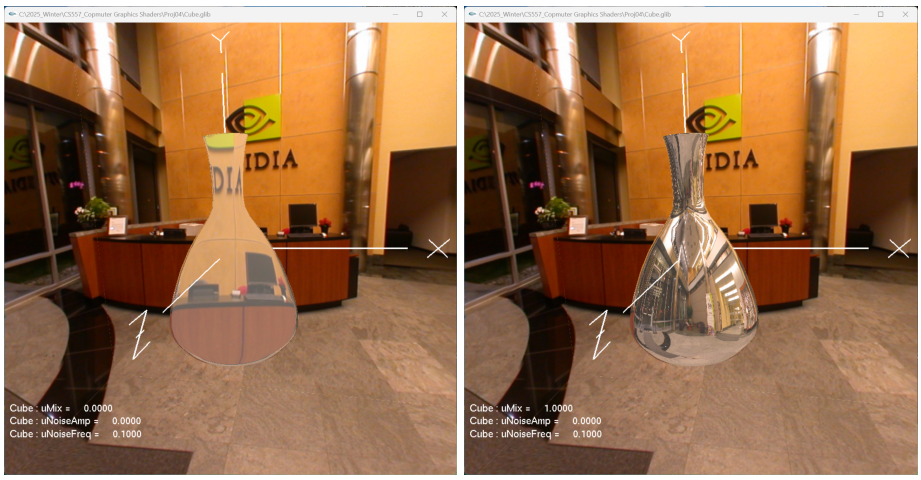

Figure 1: Refract(left) and Reflect(right)

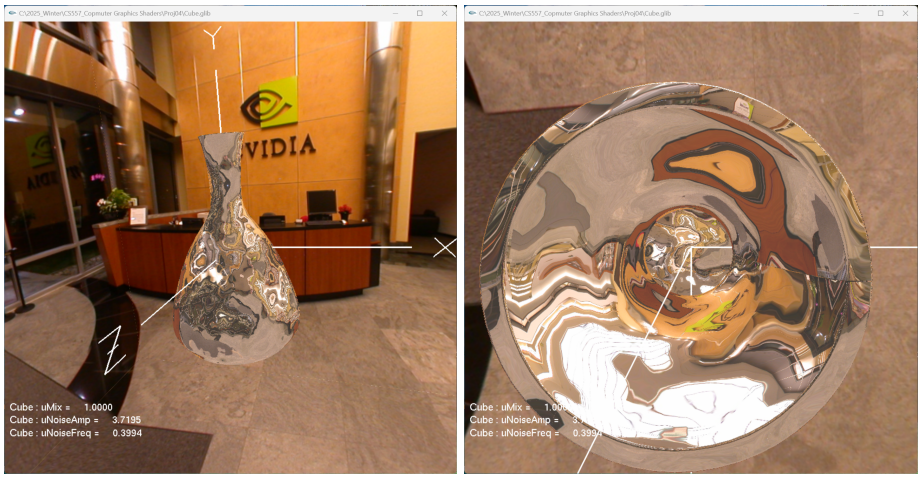

Figure 2: Bump-maps