Name : Sanketh Karuturi

Email ID: karutusa@oregonstate.edus

CS 557 *Computer Graphics Shaders*

Project #4: Cube Mapping Reflective and Refractive Bump-mapped Surfaces

1) What you did and explaining why it worked this way

Cube Mapping for Reflective and Refractive Surfaces

- I Implemented cube mapping to simulate reflective and refractive properties on a bump-mapped surface.
- This approach works effectively for simulating realistic environmental reflections and refractions because it uses pre-rendered images of the environment (the cube map) to mimic how light interacts with surfaces. By mapping these images onto the faces of a cube surrounding the scene, you can calculate reflections and refractions based on the surface's orientation and the viewer's perspective, creating a realistic effect without the need for complex real-time calculations.

Bump Mapping with Mathematical Function

- I utilized a mathematical function for bump mapping, carried over from Project #3.
- The use of a mathematical function for bump mapping enables the simulation of complex surface textures with relatively little computational cost. When combined with cube mapping, the altered normals influence the reflection and refraction calculations, allowing for detailed and dynamic surface appearances that change with the viewing angle and lighting conditions.

Parameter Effects

- Parameters such as **uA** (ripple amplitude), **uB** (ripple frequency), **uNoiseAmp** (noise amplitude), **uNoiseFreq** (noise frequency), and **uMix** (mix of reflection and refraction) were varied to demonstrate their effects on the surface's appearance.

- Varying these parameters changes the surface's visual properties dynamically. For example, adjusting the ripple amplitude and frequency alters the perceived depth and spacing of the bumps, affecting how light reflects off or refracts through the surface. The noise parameters introduce randomness, creating a more natural and less uniform appearance. The **uMix** parameter enables blending between reflective and refractive effects, offering control over the balance between the two, which is crucial for achieving realistic visual results.

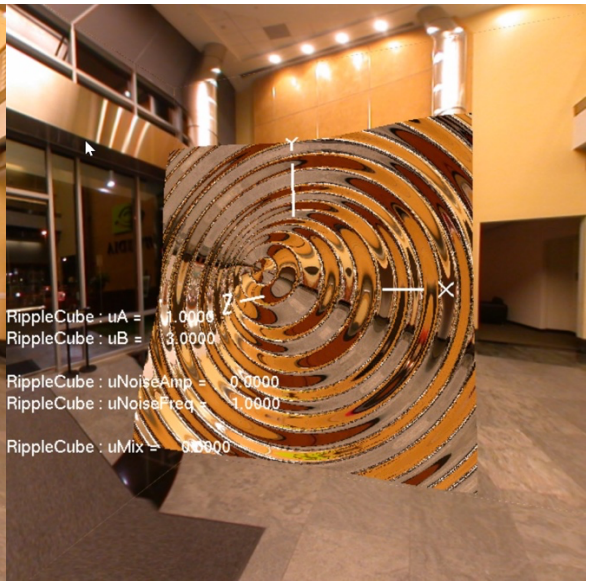
Hard-coded Parameters

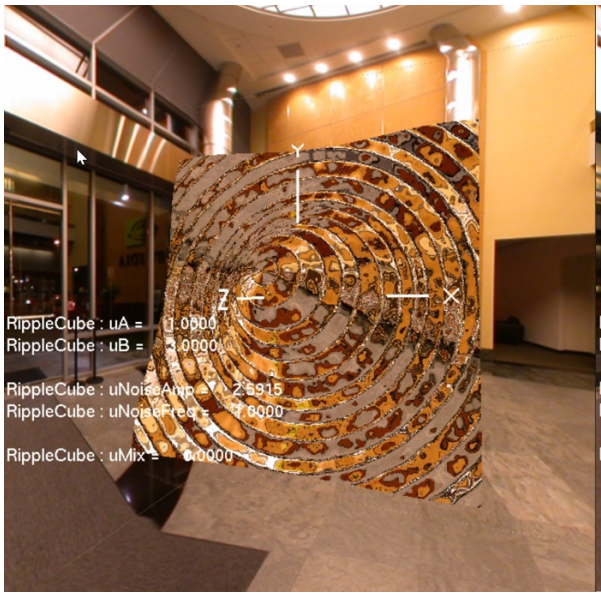
- Some parameters, like **uEta** (index of refraction), were hard-coded. The index of refraction determines how much the light bends when passing through the material, a key component in simulating refraction.

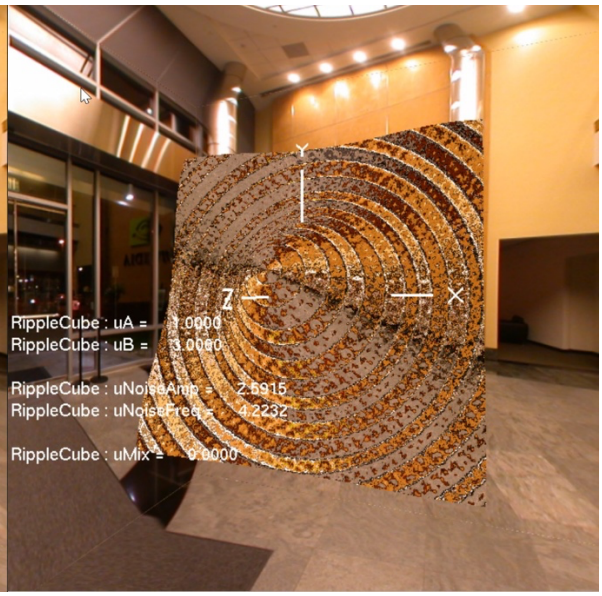
- Hard-coding **uEta** simplifies the implementation while still providing a realistic approximation of light bending. Since the focus is on demonstrating cube mapping and bump mapping effects, keeping some parameters fixed allows for a clearer observation of how varying other parameters impacts the visual outcome.

2) Side-by-side images showing different values for the input parameters


RippleCube : uA = 1.0000
RippleCube : uB = 1.0195
RippleCube : uNoiseAmp = 0.0000
RippleCube : uNoiseFreq = 1.0000
RippleCube : uMix = 0.0000


RippleCube : uA = 1.0000
RippleCube : uB = 3.0000
RippleCube : uNoiseAmp = 0.0000
RippleCube : uNoiseFreq = 1.0000
RippleCube : uMix = 0.0000


RippleCube : uA = 1.0000
RippleCube : uB = 3.0000
RippleCube : uNoiseAmp = 2.5915
RippleCube : uNoiseFreq = 1.0000
RippleCube : uMix = 0.0000


RippleCube : uA = 1.0000
RippleCube : uB = 3.0000
RippleCube : uNoiseAmp = 2.5915
RippleCube : uNoiseFreq = 4.2232
RippleCube : uMix = 0.0000


RippleCube : uA = 1.0000
RippleCube : uB = 3.0000
RippleCube : uNoiseAmp = 2.5915
RippleCube : uNoiseFreq = 0.6183
RippleCube : uMix = 1.0000

3) **Image(s) showing that your refraction is correct.**

4) **Image(s) showing that your reflection is correct.**

4) **Media Link :** https://media.oregonstate.edu/media/t/1_nabchbfc