Introduction to Computer Graphics (CS360A)

Instructor: Soumya Dutta

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur (IITK)

email: soumyad@cse.iitk.ac.in

New Sphere Drawing Code

- HelloIITK → Resources → New Sphere Drawing Routine
- Try it out if the previous code was giving you trouble

Texture Map Code Demo

IITK CS360A

- 3DTextureMapExample.js, 3DTextureMapExample.html

Reflection Mapping

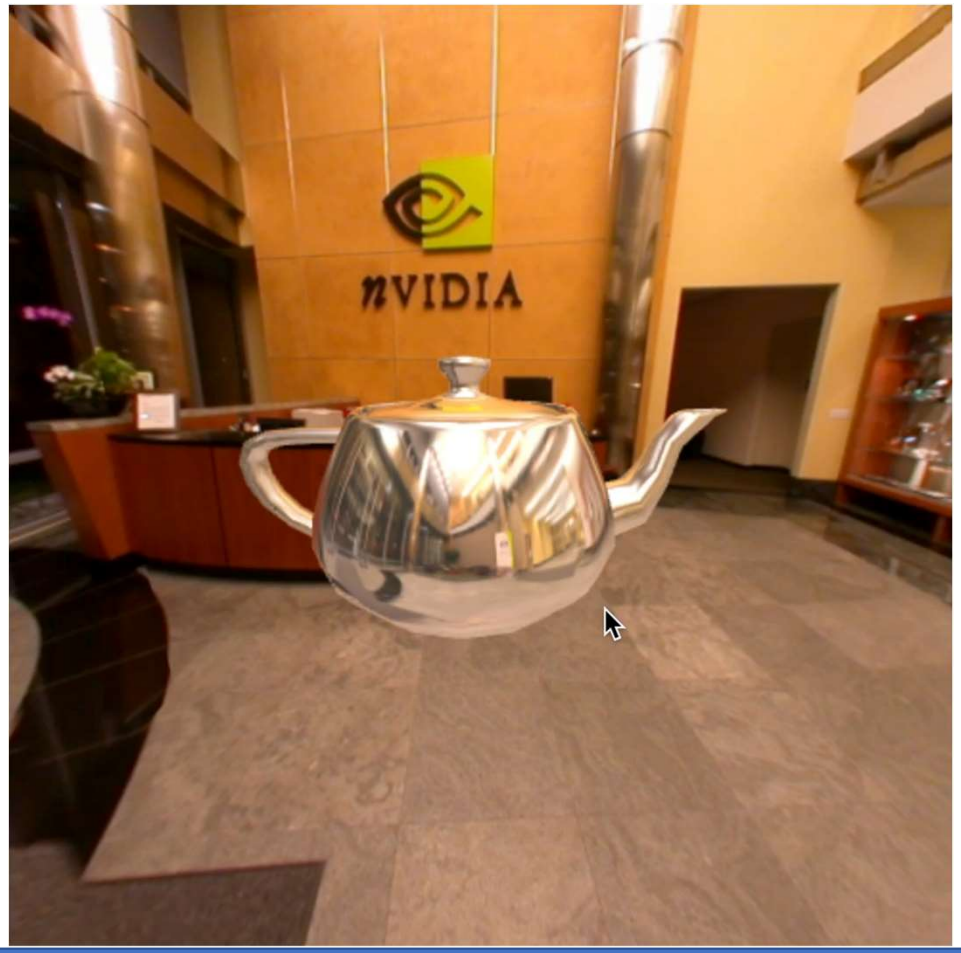
A Rough Teapot

What If the Teapot is Shiny?

Is It Realistic? What is Missing?

What If My Teapot is Very Very Shiny?

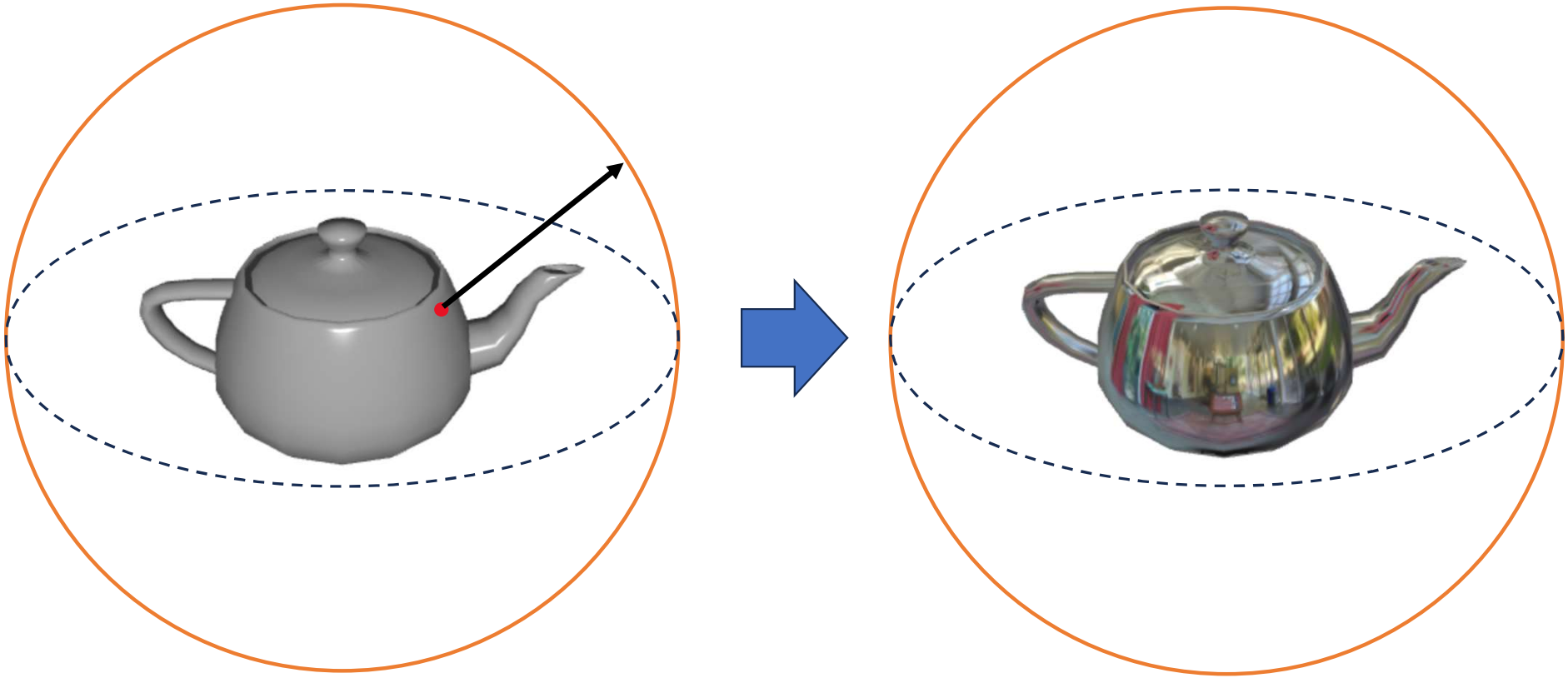
Environment Reflection



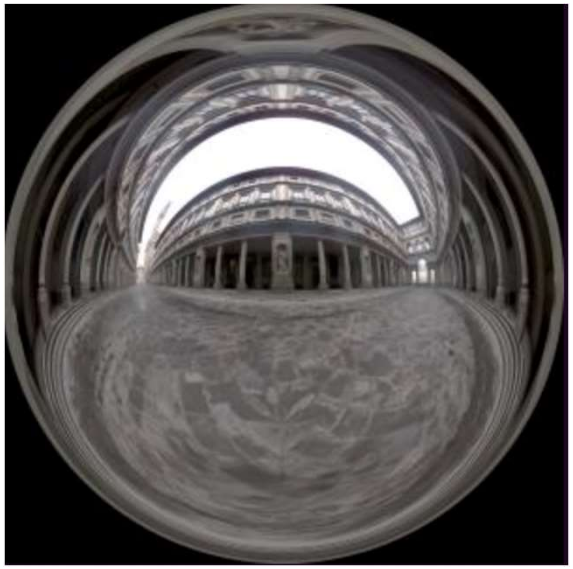
How Do We Get Such Environment Reflections?

- This happens because global reflection and lighting
- Simulating global lighting and reflection is very expensive!
- If we assume the environment is very far away (at infinity) from the object, then there is a way we can simulate it cheaply in real time
- We use texture mapping technique smartly to get such effects
  - The computation is done in hardware in GPU, hence very fast!
- Two well-known methods
  - Sphere reflection mapping
  - Cube environment (reflection) mapping

General Idea: Sphere Mapping

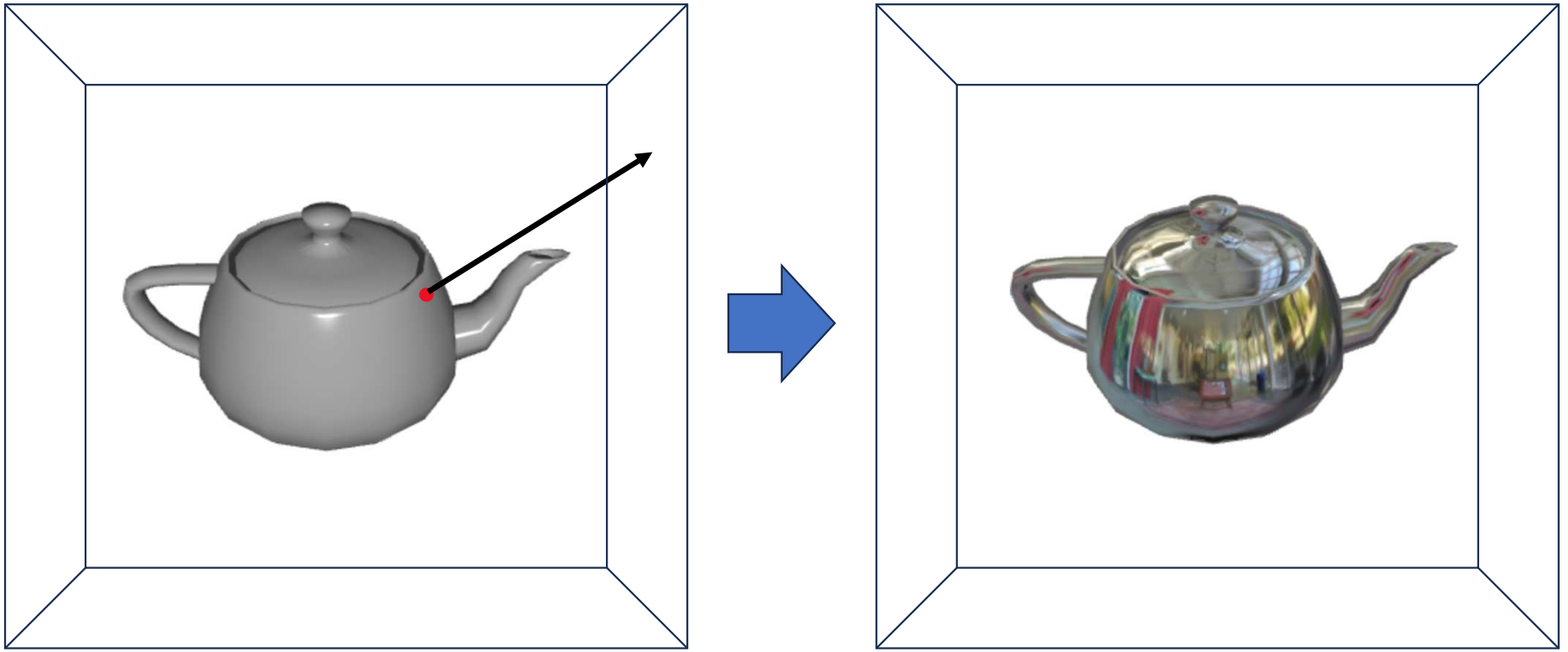


Sphere Mapping Texture

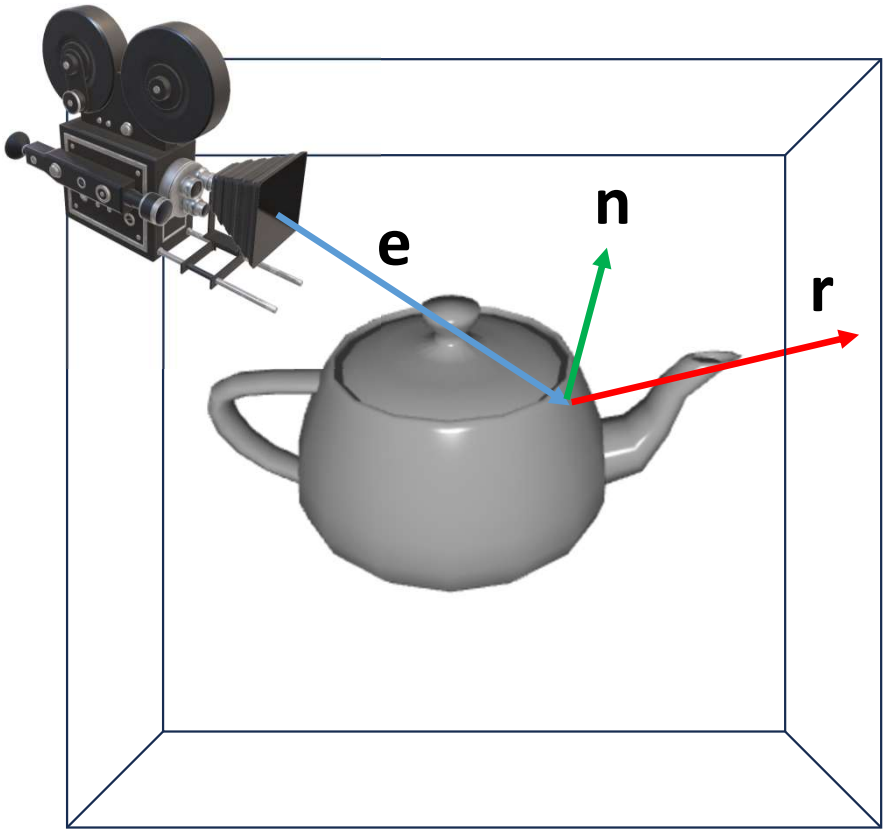


A light probe image is an omnidirectional, HDR image that records the incident illumination conditions at a particular point in space.

General Idea: Cube Environment Mapping

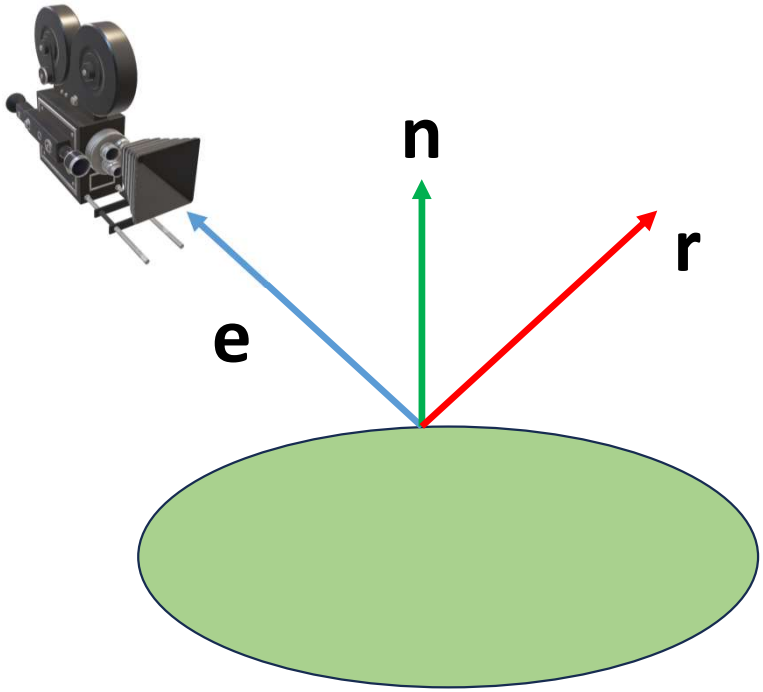


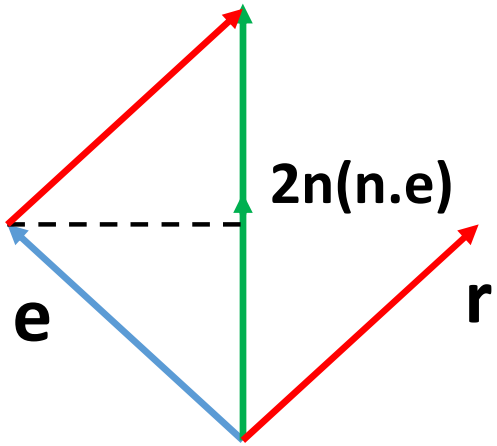
General Idea: Cube Environment Mapping



- From the camera/eye position, we are looking at the object
- We want to look up using the reflection vector (**r**) as shown
- All we need is the reflection vector from each point of the object

How to Compute the Reflection Vector

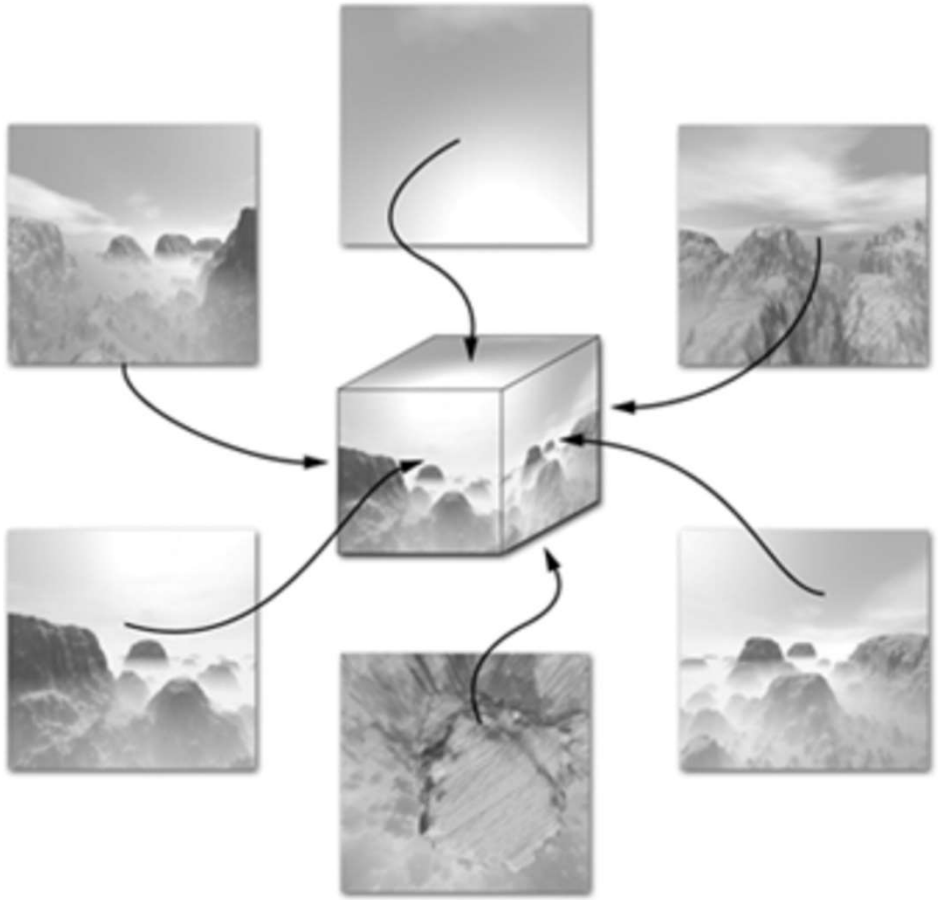




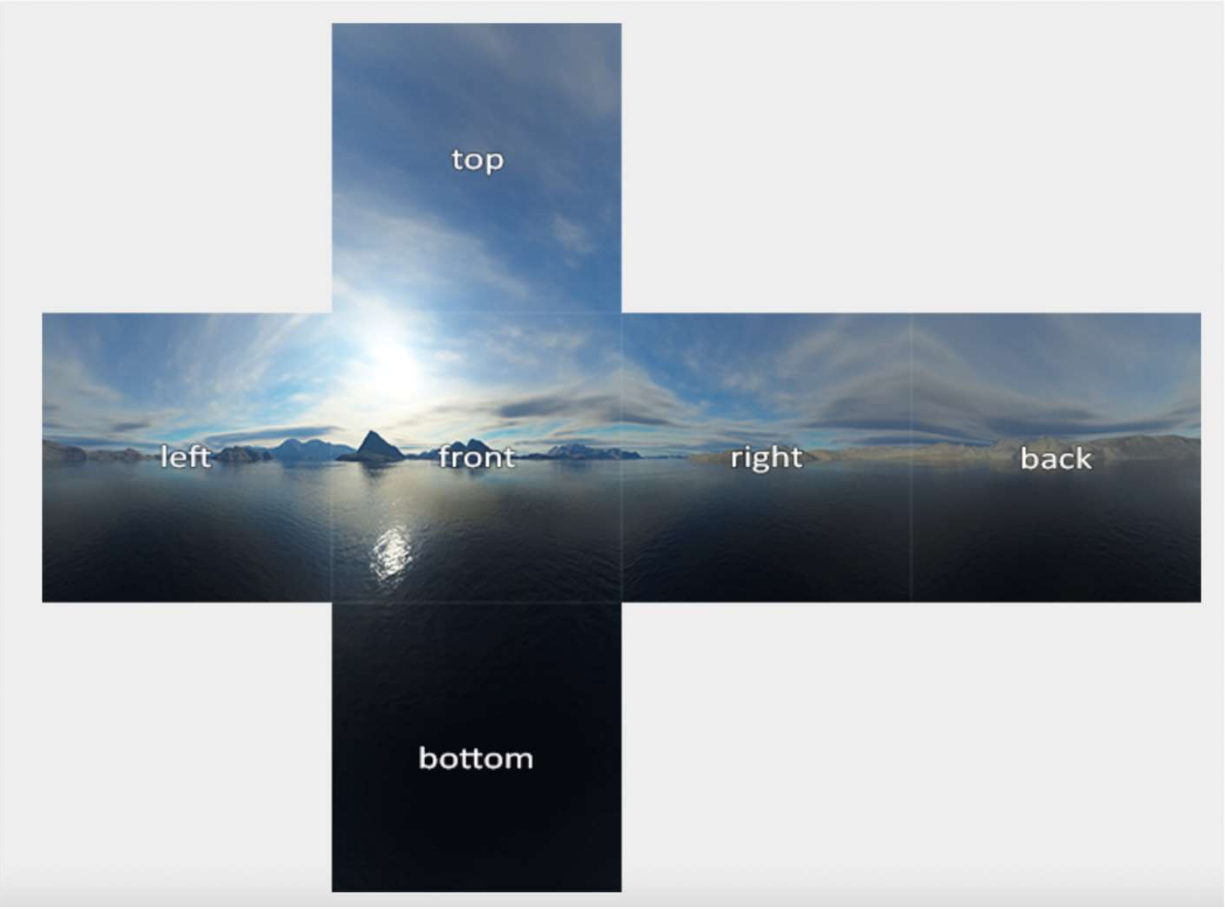
r = 2 n (n.e) - e

n and **e** should be normalized

How Exactly Textures are Used?



Cube Map Textures



- Take photographs of 6 sides of the environment
- Paste them on a cube's 6 faces
- Use this as a texture to look up colors using the reflection vector
- Shade the reflective object

Cube Environment Mapping in WebGL

- How does WebGL determine which face of the cube texture to look up and how does it look up the texture?
  - Assume object at origin
  - Largest magnitude component of **r** determines face of cube
  - Other two components are used to compute the texture coordinates for color look up

Cube Environment Mapping in WebGL


```
function initCubeMap() {
  const faceImages = [
    {
      target: gl.TEXTURE_CUBE_MAP_POSITIVE_X,
      url: cubeMapPath.concat("posx.jpg"),
    },
    {
      target: gl.TEXTURE_CUBE_MAP_NEGATIVE_X,
      url: cubeMapPath.concat("negx.jpg"),
    },
    {
      target: gl.TEXTURE_CUBE_MAP_POSITIVE_Y,
      url: cubeMapPath.concat("posy.jpg"),
    },
```

-
-

Cube Environment Mapping in WebGL


```
cubemapTexture = gl.createTexture();
gl.bindTexture(gl.TEXTURE_CUBE_MAP, cubemapTexture);

faceImages.forEach((face) => {
  const { target, url } = face;

  // Upload the canvas to the cubemap face.
  const level = 0;
  const internalFormat = gl.RGBA;
  const width = 512;
  const height = 512;
  const format = gl.RGBA;
  const type = gl.UNSIGNED_BYTE;
  // setup each face so it's immediately renderable
  gl.texImage2D(target, level, internalFormat,
                width, height, 0, format, type, null
  );
```

```
  // load images
  const image = new Image();
  image.src = url;
  image.addEventListener("load", function () {
    gl.bindTexture(gl.TEXTURE_CUBE_MAP, cubemapTexture);
    gl.texImage2D(target, level, internalFormat, format, type, image);
    gl.generateMipmap(gl.TEXTURE_CUBE_MAP);
    drawScene();
  });
});

// uses mipmap for texturing
gl.generateMipmap(gl.TEXTURE_CUBE_MAP);
gl.texParameteri(
  gl.TEXTURE_CUBE_MAP,
  gl.TEXTURE_MIN_FILTER,
  gl.LINEAR_MIPMAP_LINEAR
);
```

Cube Environment Mapping in WebGL: Shader

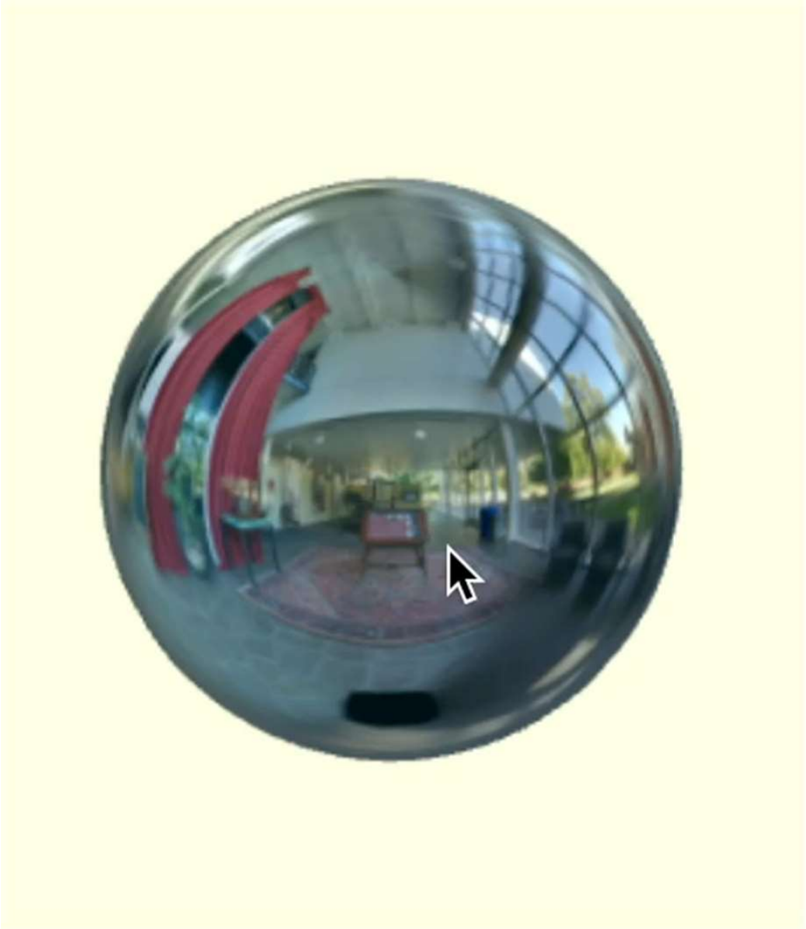
- *uniform samplerCube cubeMap;*
- *directionReflection = reflect(eyeToSurfaceDir,worldNormal);*
- *fragColor = texture(cubeMap, directionReflection);*

Cube Environment Mapping in World Space

- Environment mapping is done in World coordinate space
- Need to send world space vertex position and world space normal for computation of reflection vector in world space
  - How to do this?
  - Need world space normal matrix

```
// normal matrix in world space.. needed for cubemap
wnMatrix = mat4.transpose(mat4.inverse(mMatrix));
```

Cubemap Rendering

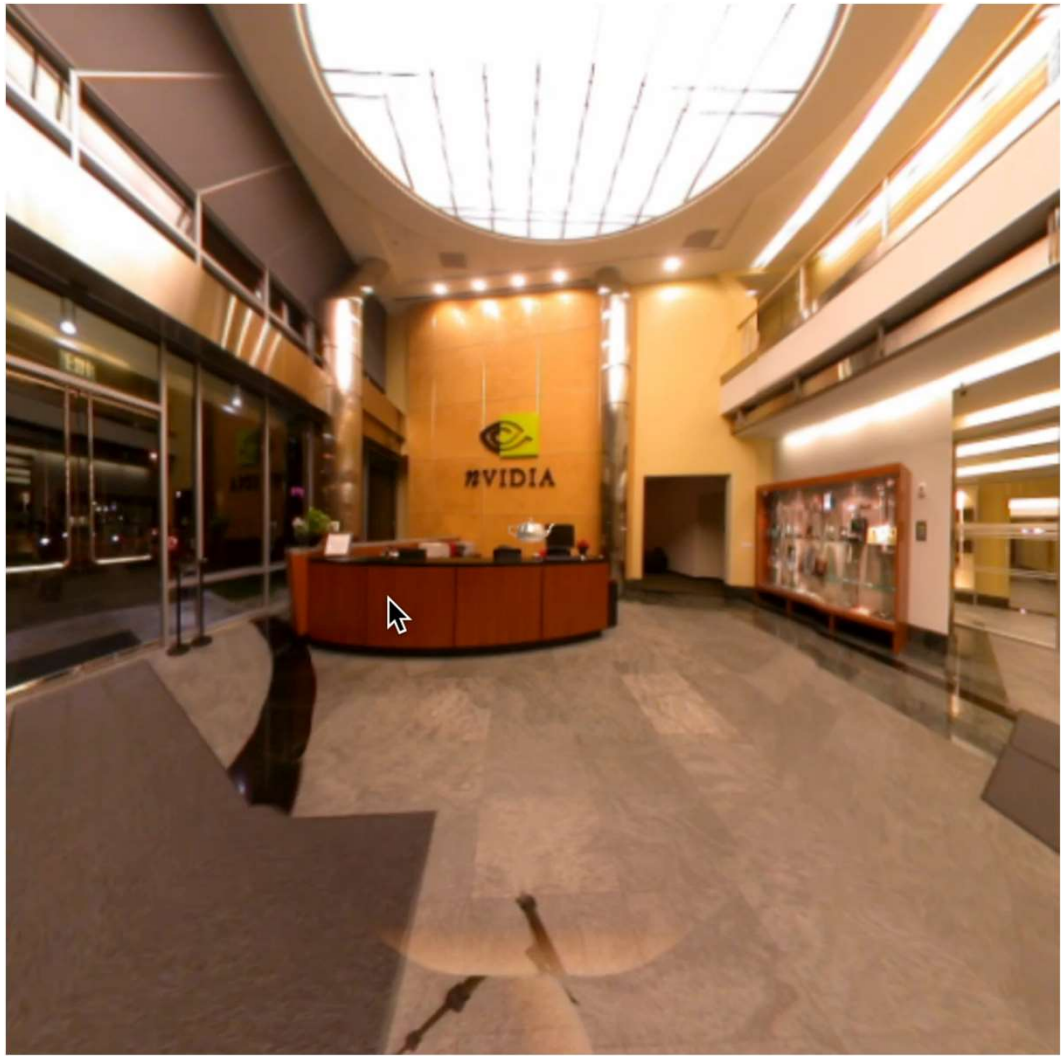


Something missing?
Where is the environment?

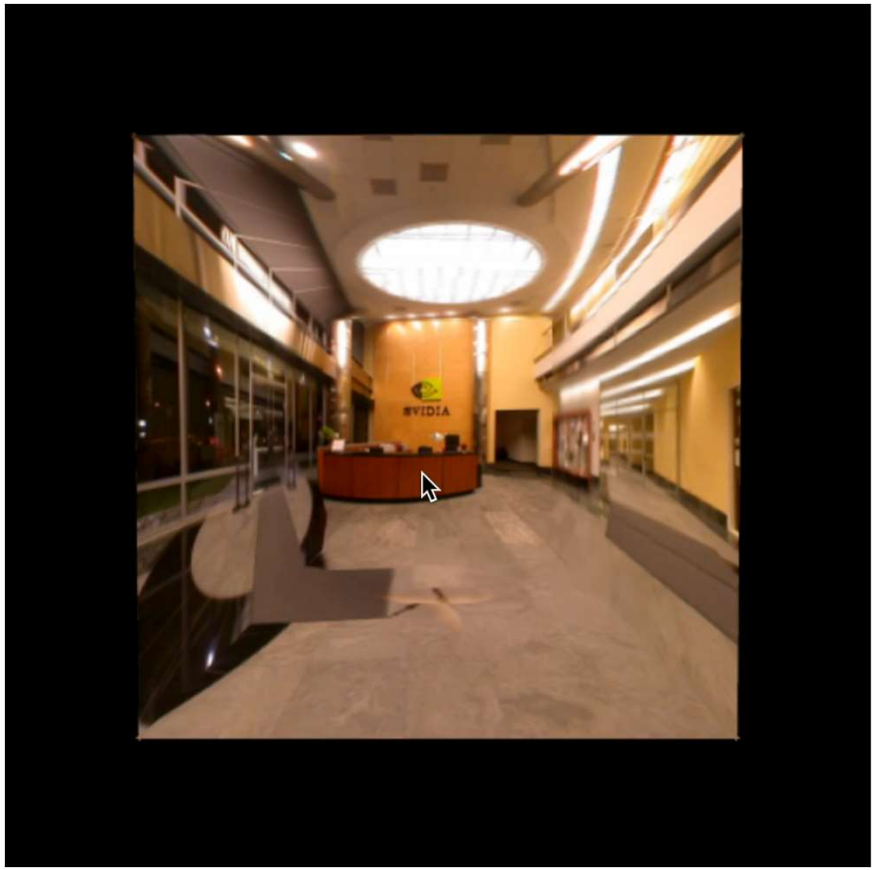
How to Add the Background Environment?

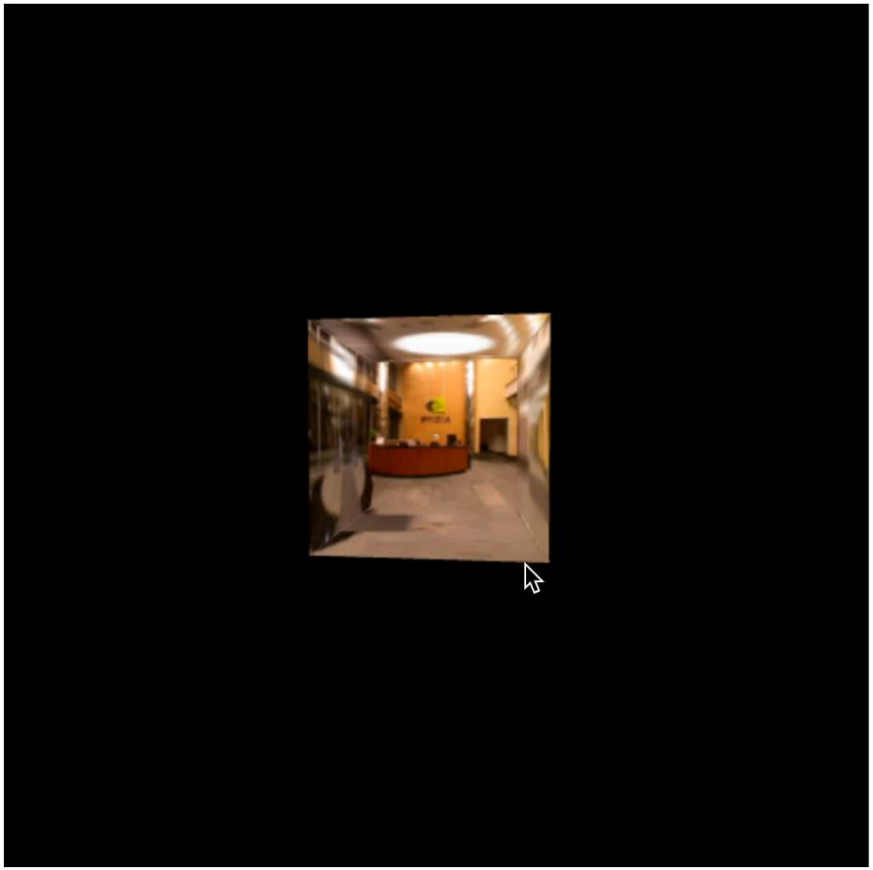
- Create a **SkyBox!**
- Easy trick: Create six planes (quads) and add the 6 images to each plane and then put them together to form a cube like structure by translating and scaling the planes
  - Plane can be simulated as a square/quad
  - Use regular texture mapping to map 6 textures to 6 squares
  - Make the squares very large so that the objects are small compared to the square

SkyBox

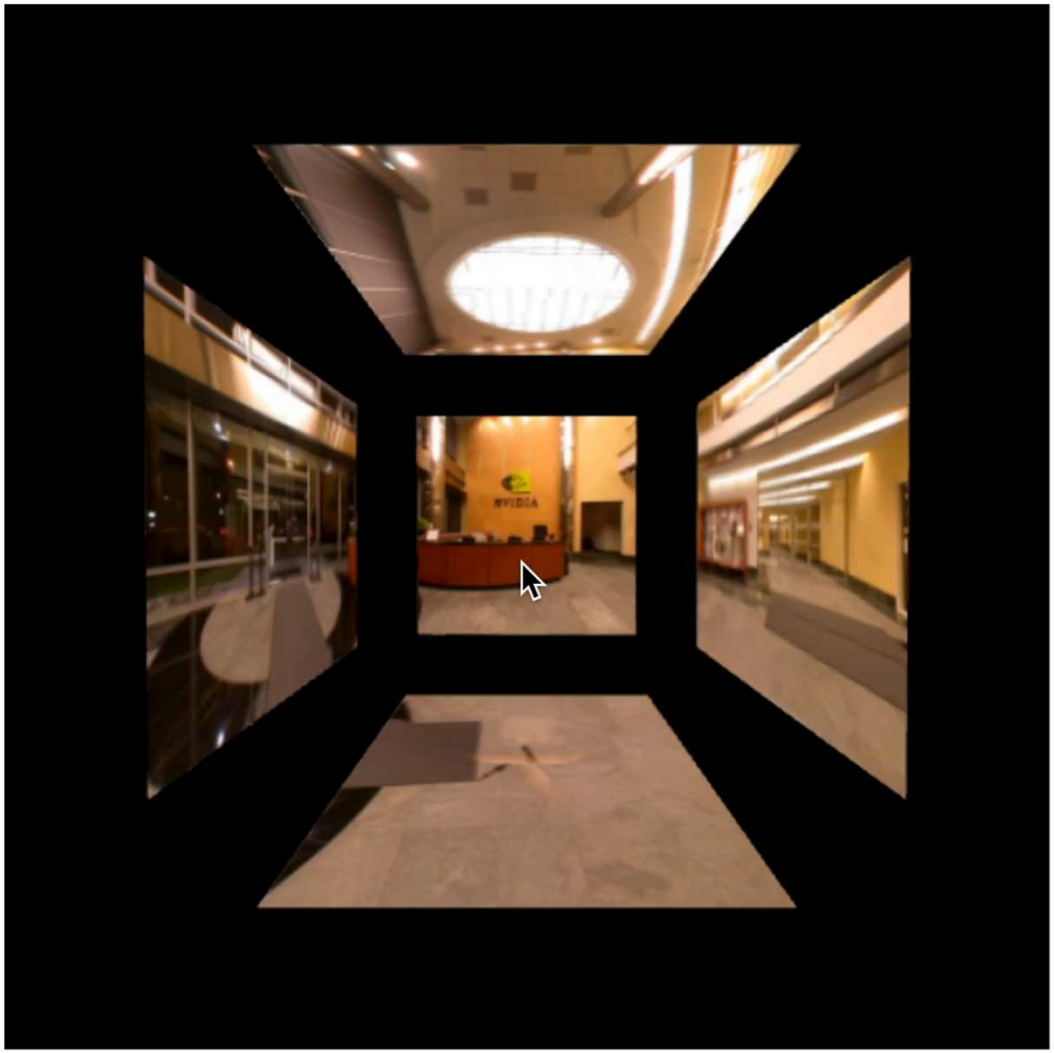


SkyBox





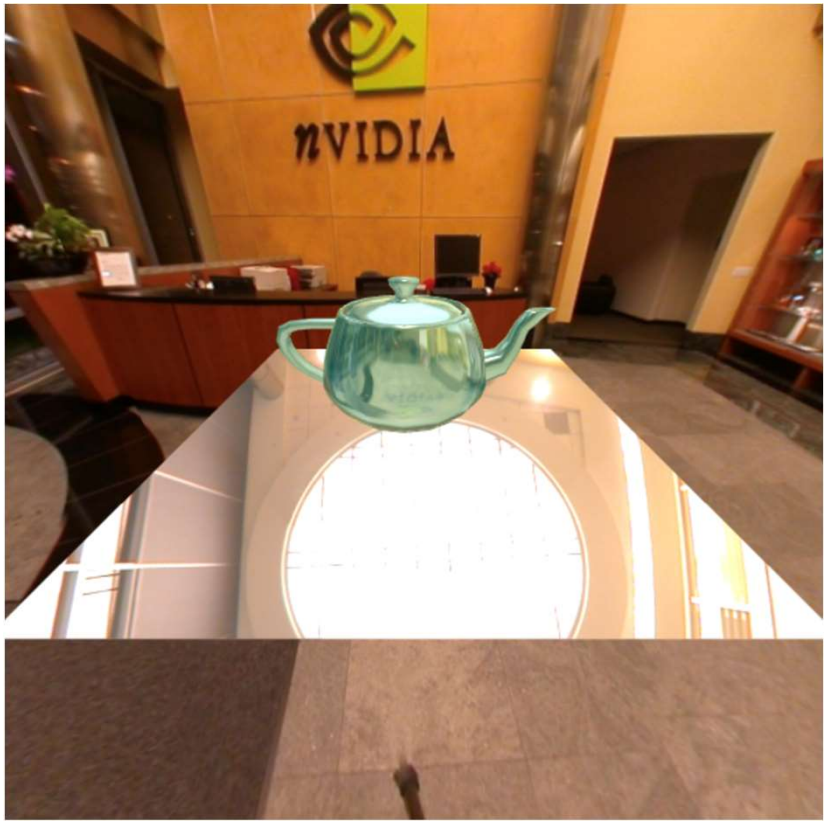
SkyBox

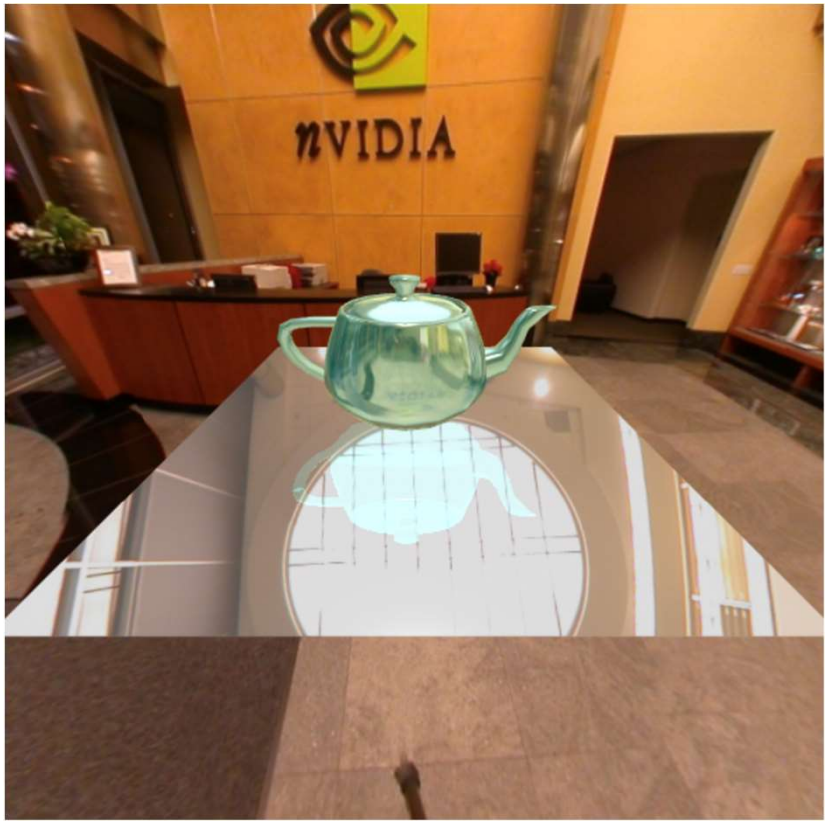


Is This Image Accurate?

IITK CS360A

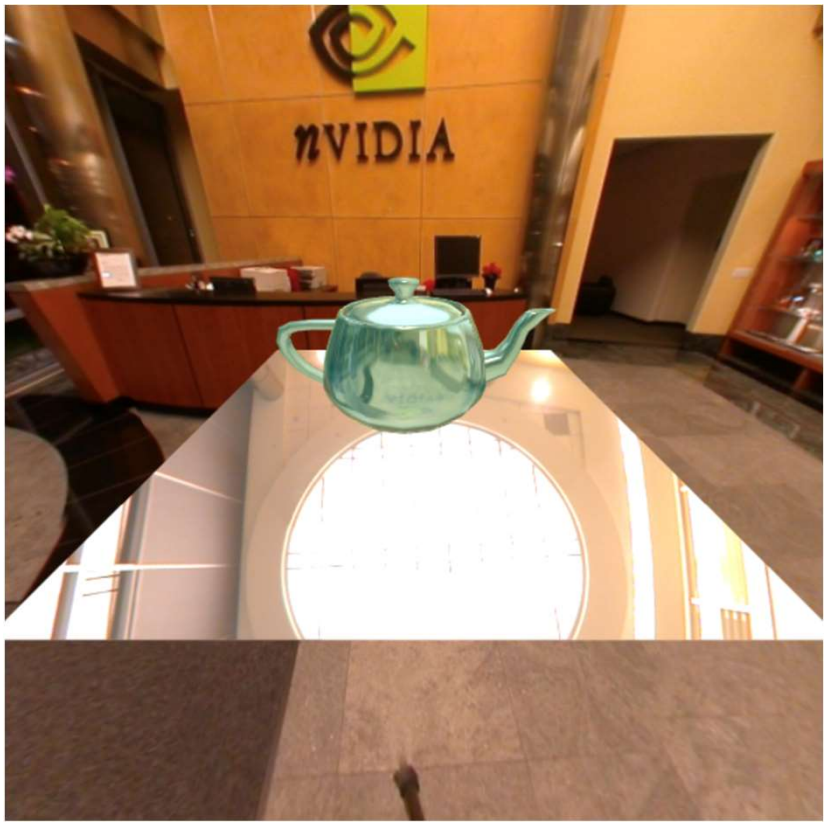
- Anything missing?

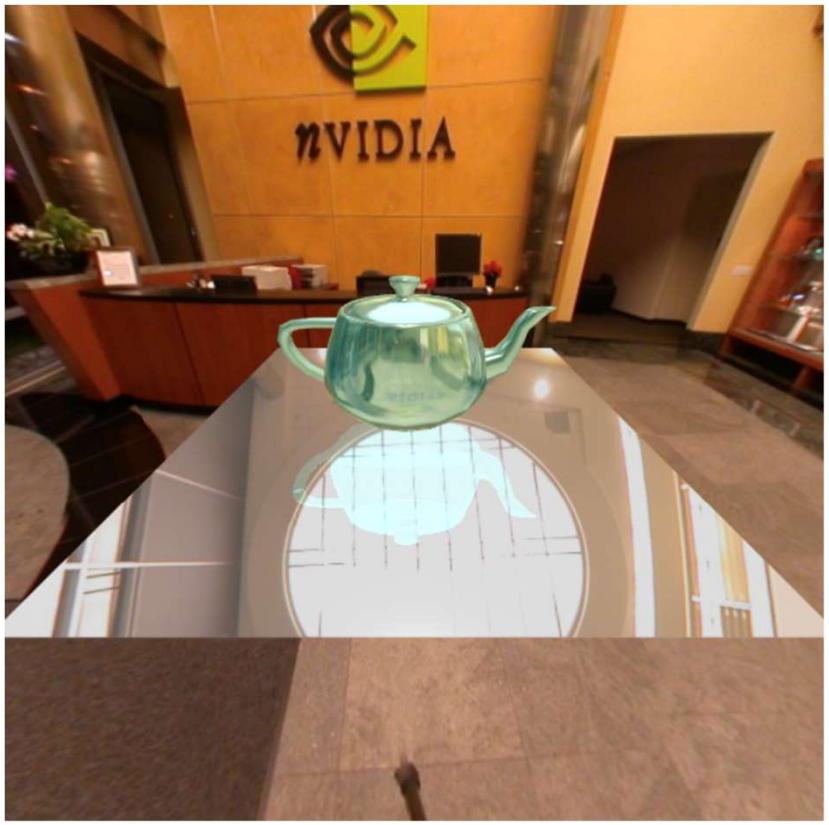




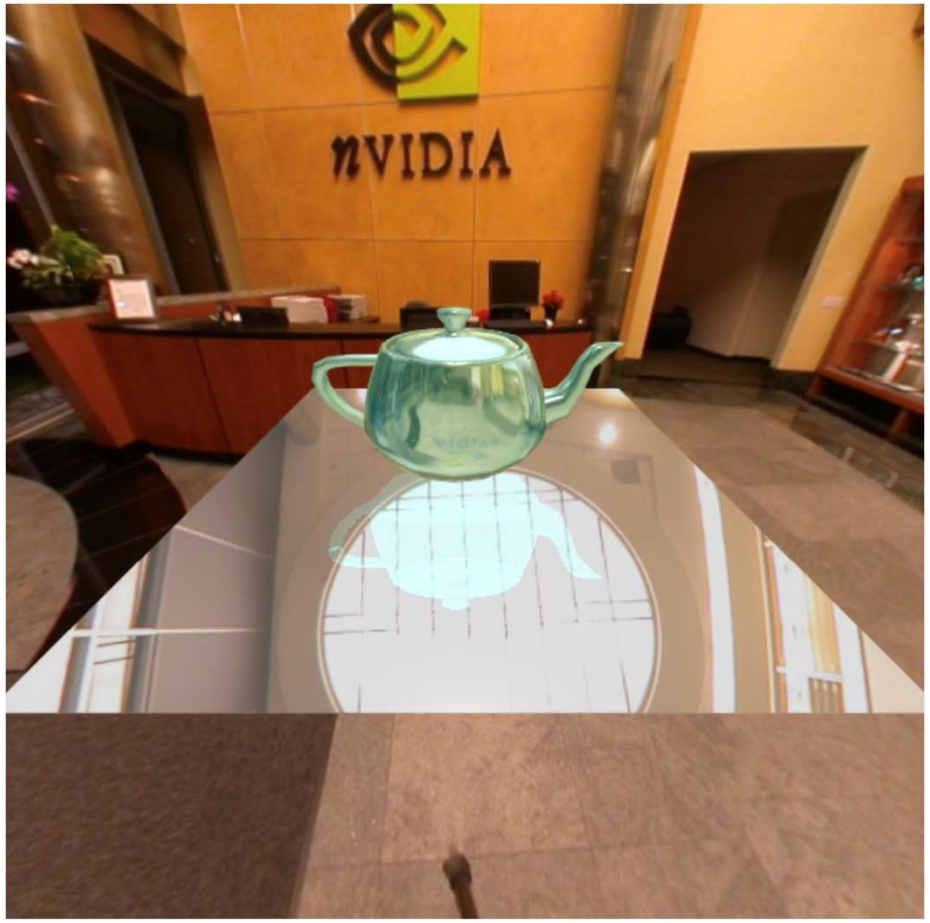
Is This Image Accurate?

- Reflection of the teapot on the shiny surface!



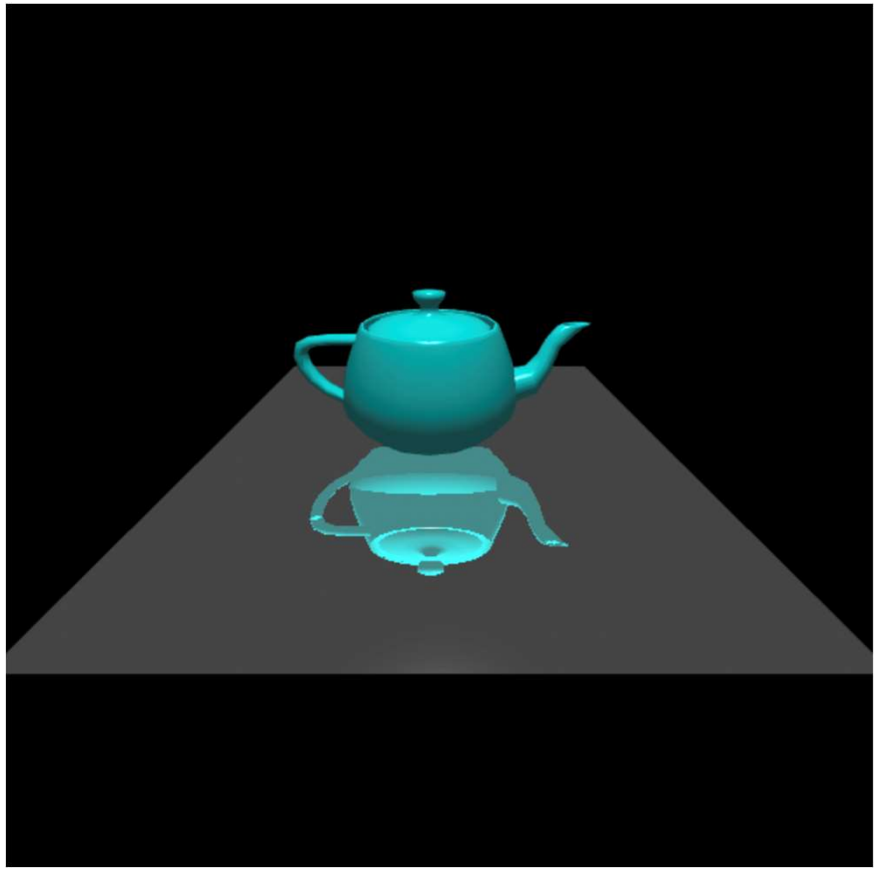


Planar Reflection

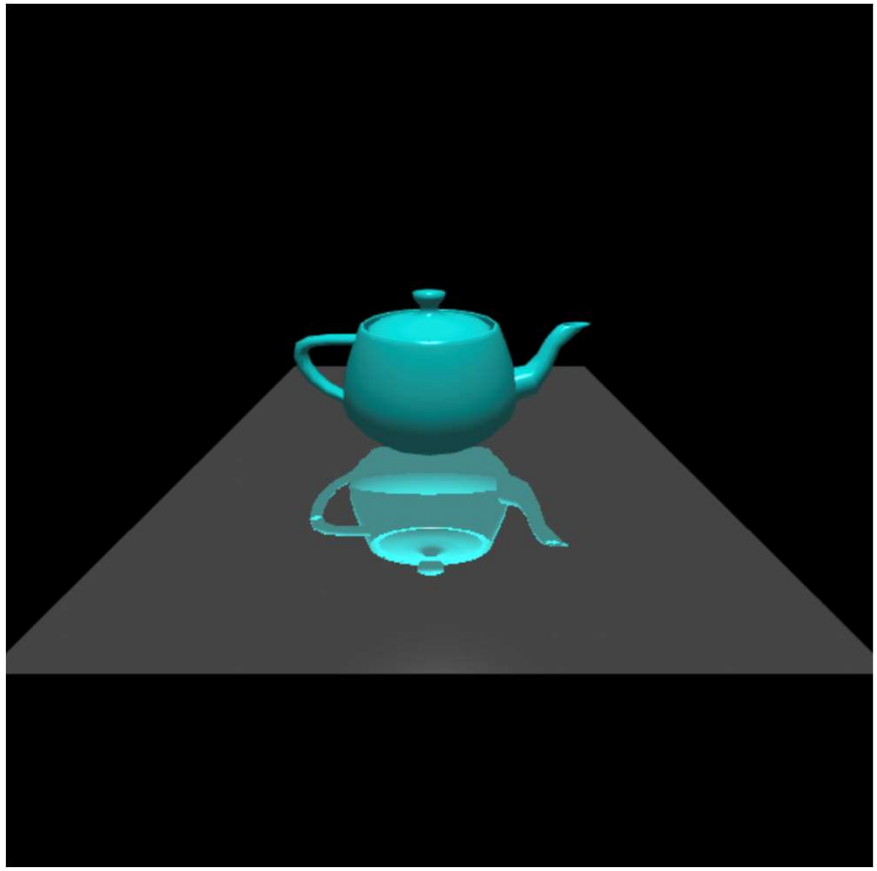


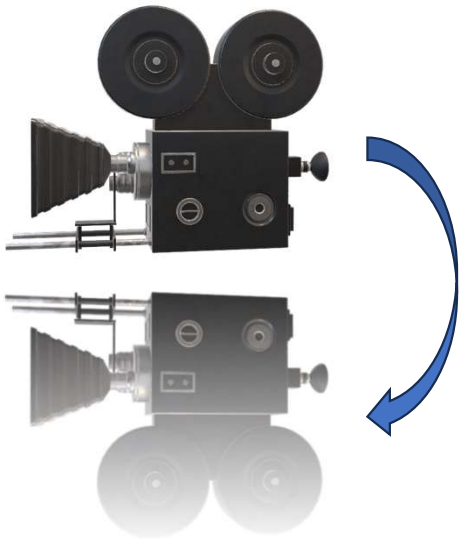
How do we get this planar reflection?

Planar Reflection

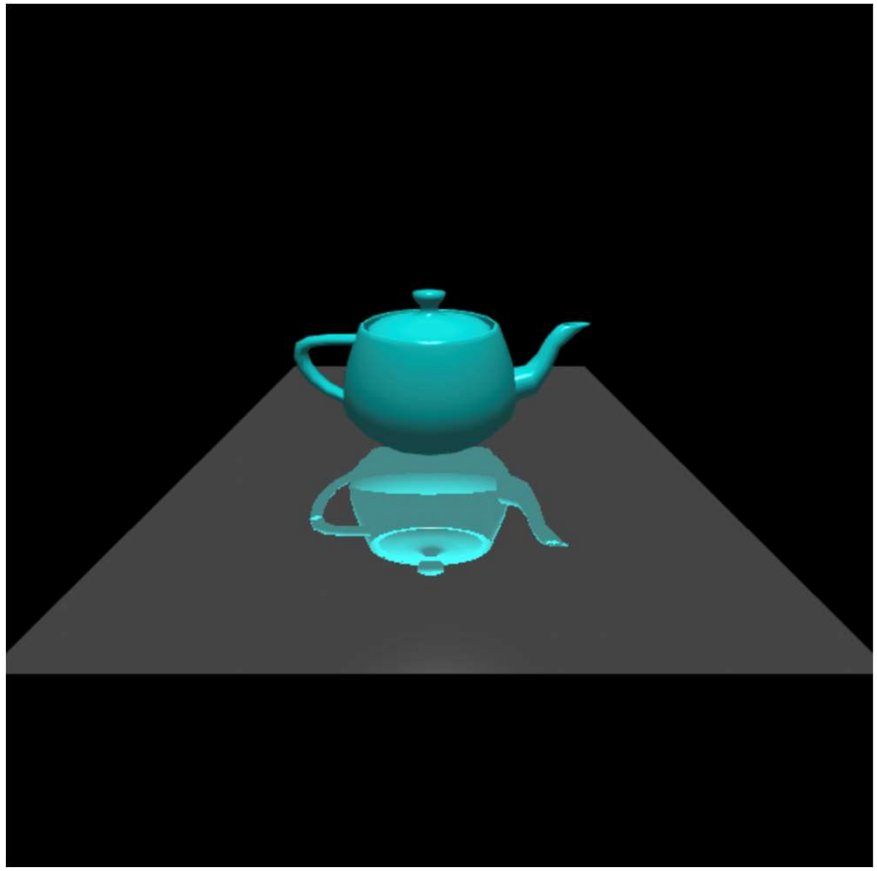


Planar Reflection



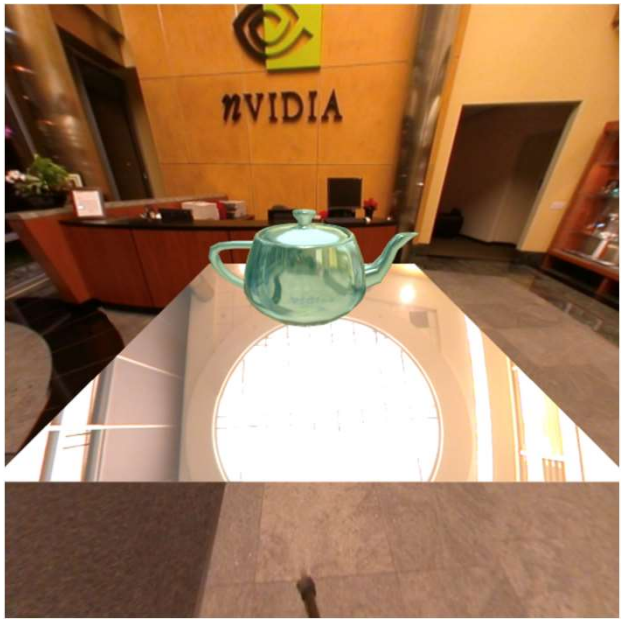


Planar Reflection



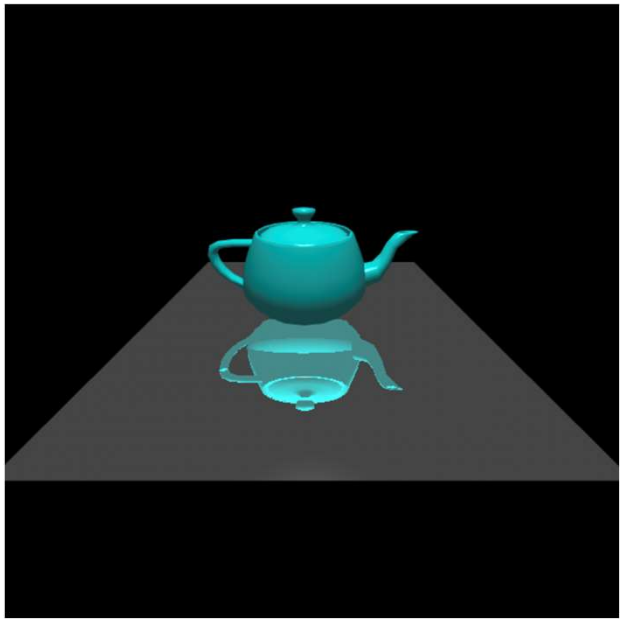
- Two pass rendering: Render the scene twice
- 1. Flip the camera and render the object flipped and store it into a buffer
- 2. Then during second pass, use the stored image as texture to add the reflection

Planar Reflection: Conceptual Idea



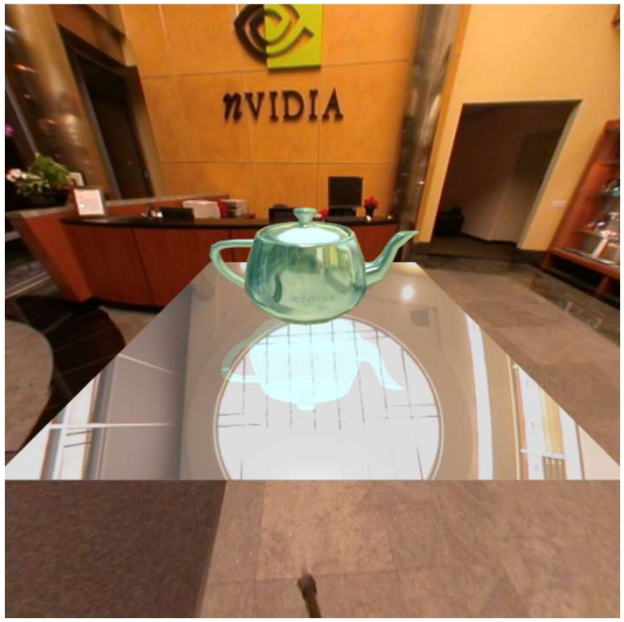
Scene with no reflection texture

+



Framebuffer (reflection) texture

=



Final rendered image with planar and cubemap reflection