Cube Maps

David Schedl

Environment mapping

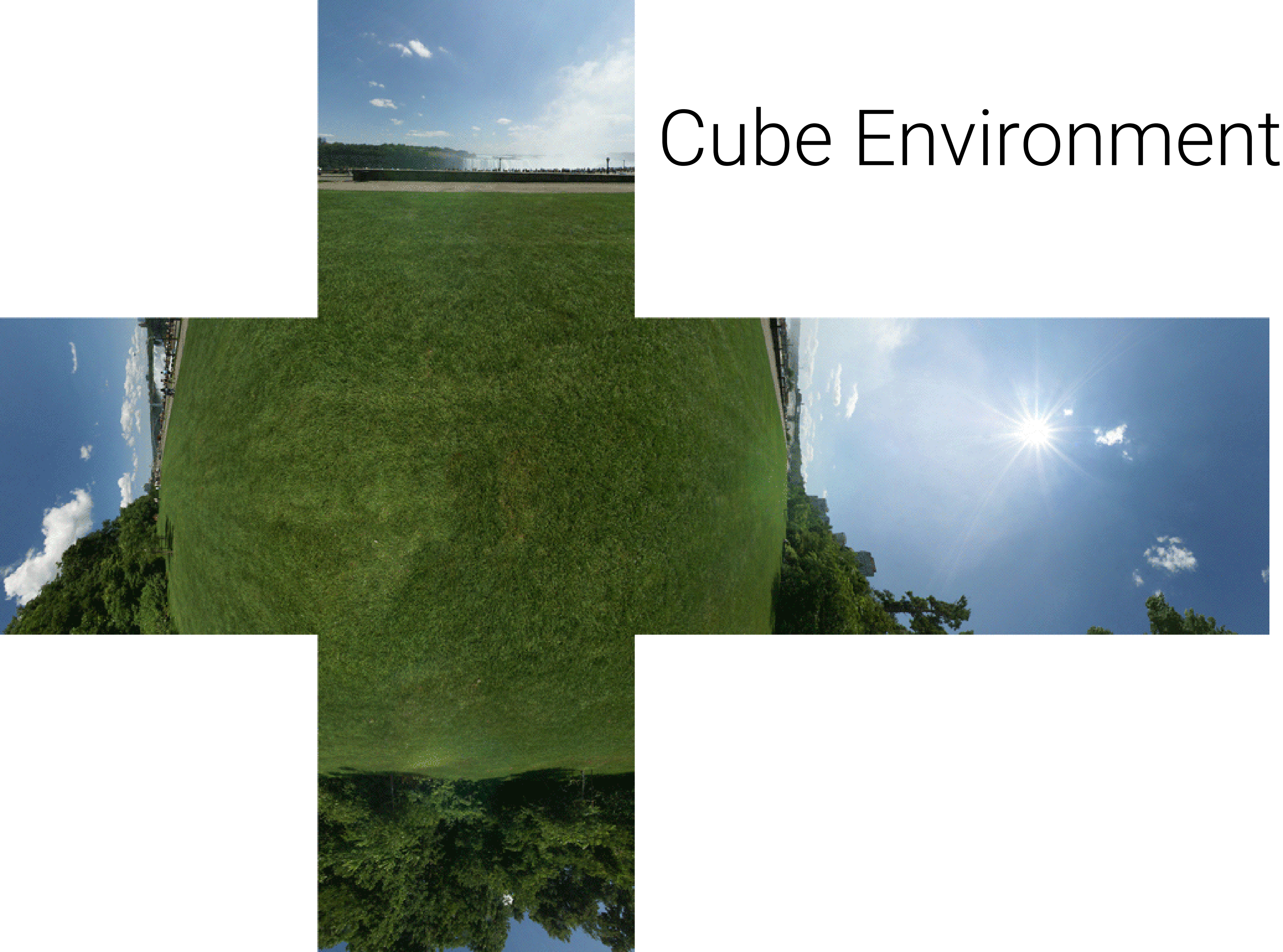
Environment Mapping

- Why?
  - Realistic (specular) reflections
  - Background (skybox)
  - Illumination (IBL)
- Where is it used?
  - Games
  - Movies

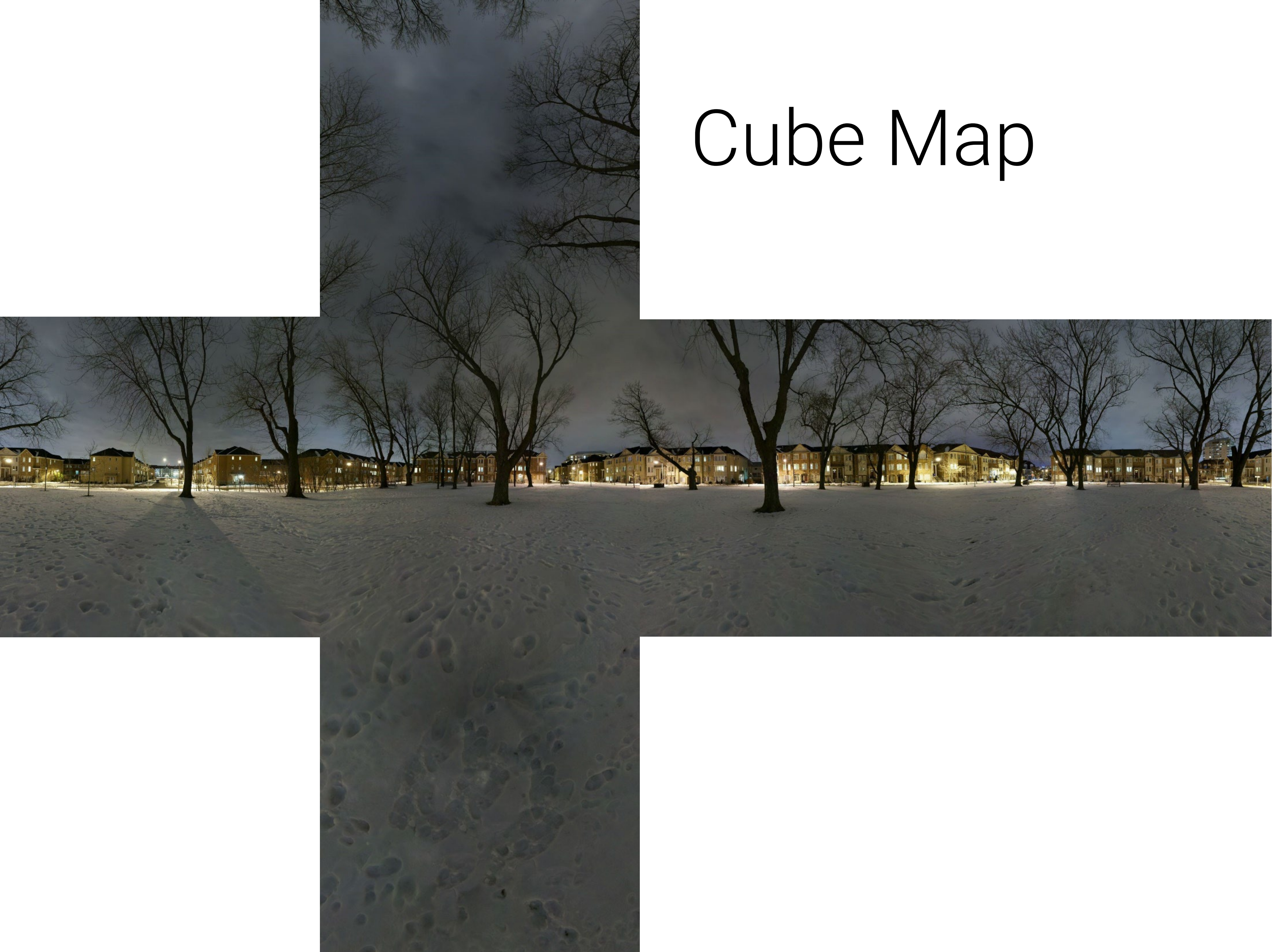
Jim Blinn

How to capture the environment?

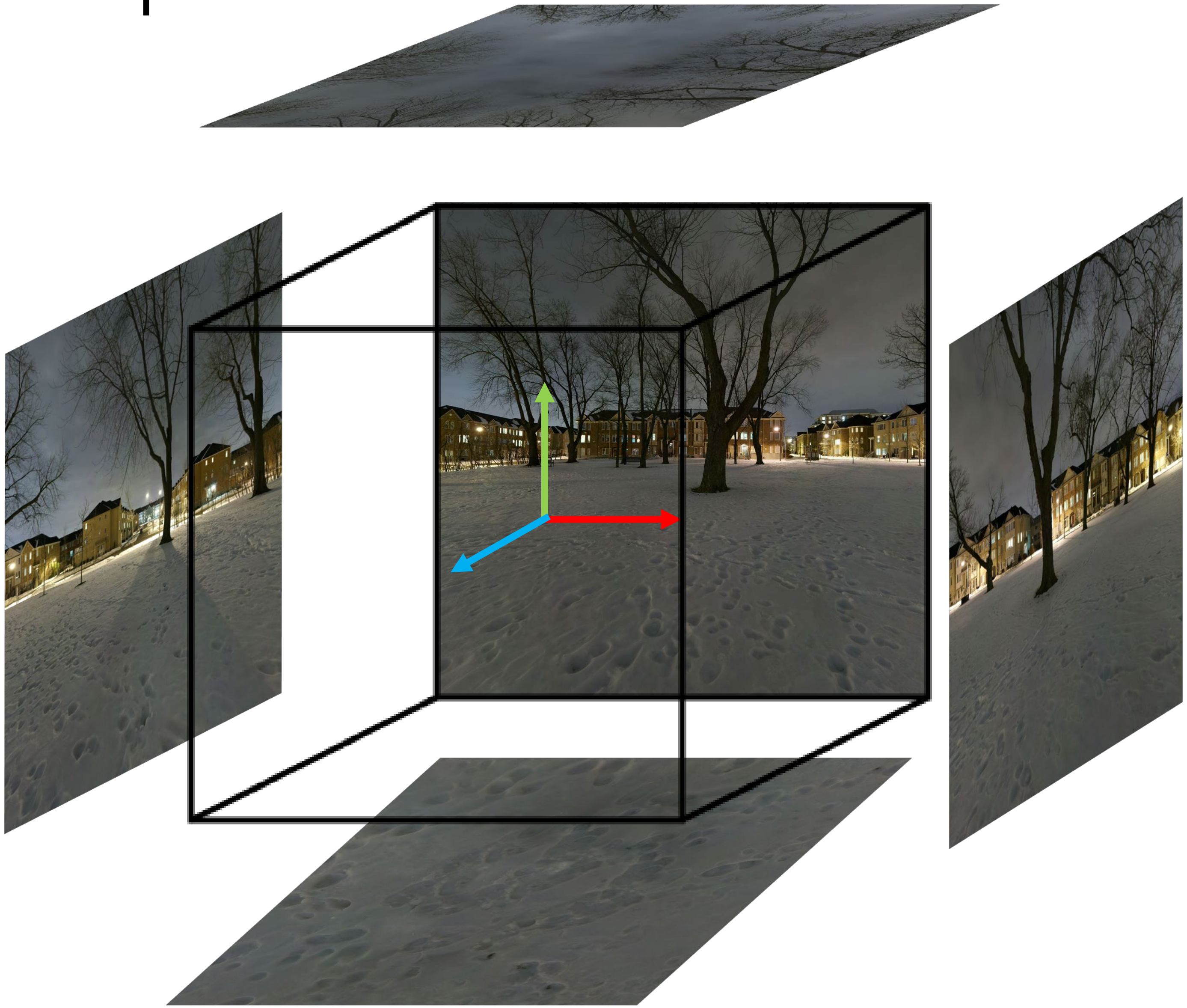
Cube Environment Mapping



Cube Map

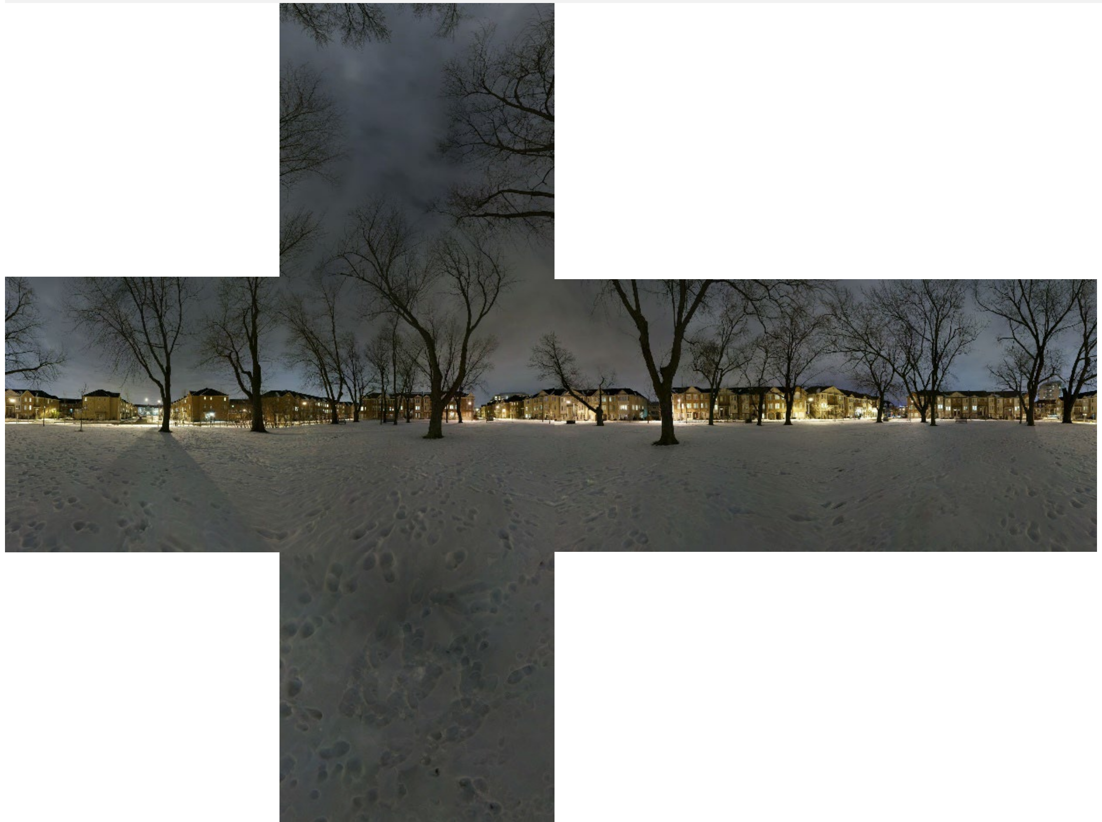


Cube Map



The Cube Map Texture

```
glGenTextures(1, &cubeTexture);
glBindTexture(GL_TEXTURE_CUBE_MAP, cubeTexture);

std::string faces[6] = { "right.jpg", "left.jpg","top.jpg","bottom.jpg", "front.jpg",
                         "back.jpg" };
```



The Cube Map Texture

```
glGenTextures(1, &cubeTexture);
glBindTexture(GL_TEXTURE_CUBE_MAP, cubeTexture);

std::string faces[6] = { "right.jpg", "left.jpg","top.jpg","bottom.jpg", "front.jpg",
                         "back.jpg" };

for (unsigned int i = 0; i < 6; i++) {
  unsigned char* image = load_image(faces[i].c_str(), &width, &height);

  if (image) {
    glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_X + i, 0, GL_RGB, width, height, 0,
                 GL_RGB, GL_UNSIGNED_BYTE, image);
  } else {
    std::cout << "Cubemap texture failed to load at path: " << faces[i] << std::endl;
  }
}
```

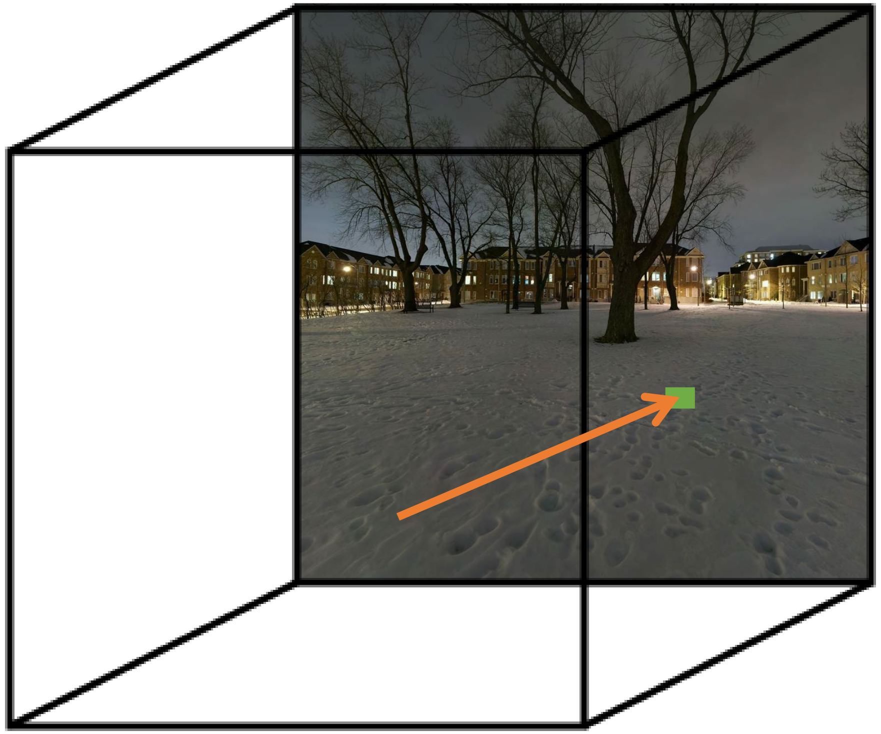
Cube Map

- Texture Lookup is 3D (x,y,z) instead of 2D (u,v).
- OpenGL code:

```
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_CUBE_MAP,
        cubeTexture);
```

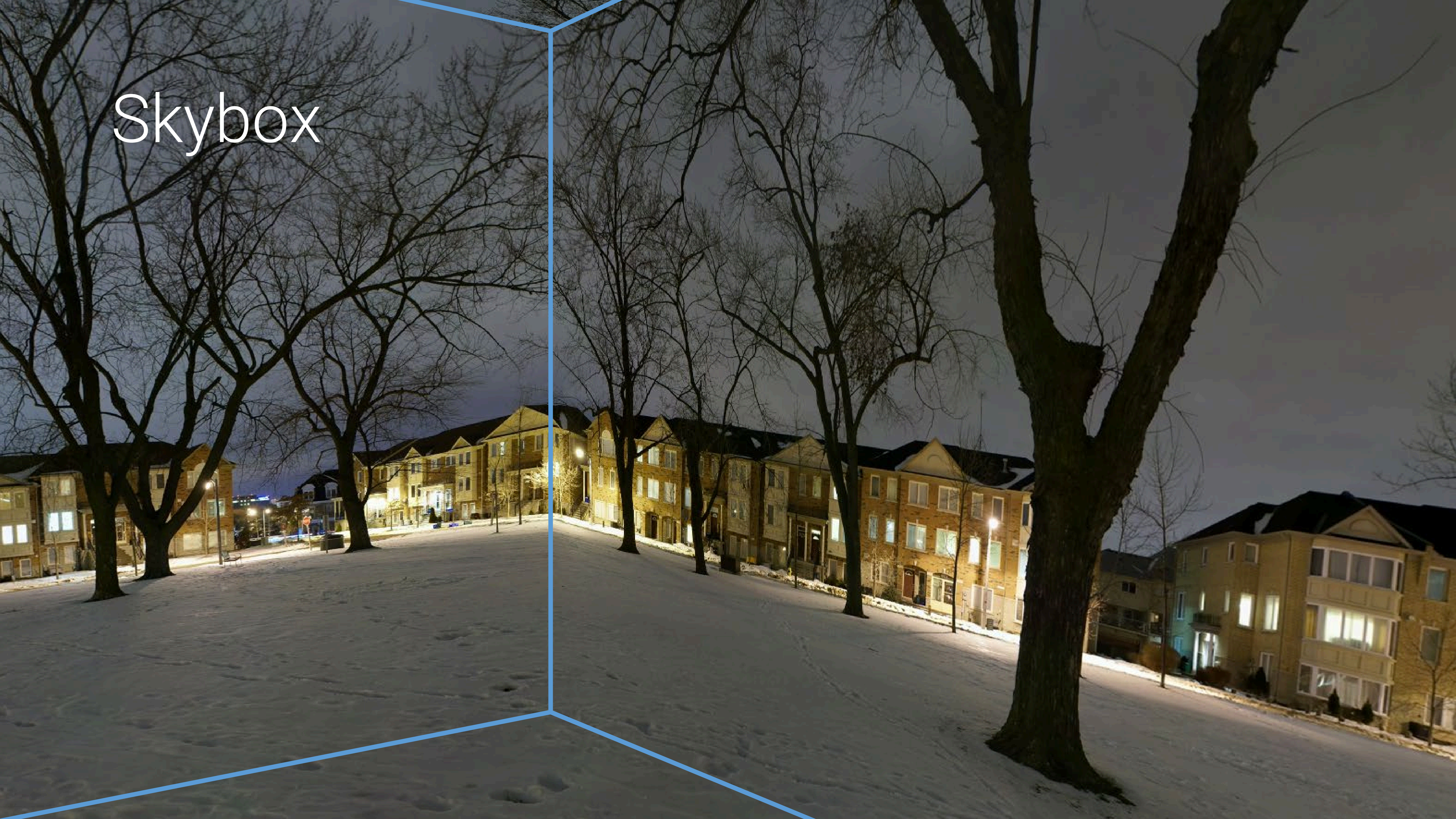
GLSL code:

```
in vec3 texCoords;
uniform samplerCube skyboxTex;
void main() {
    color = texture(skyboxTex, texCoords);
}
```



Skybox

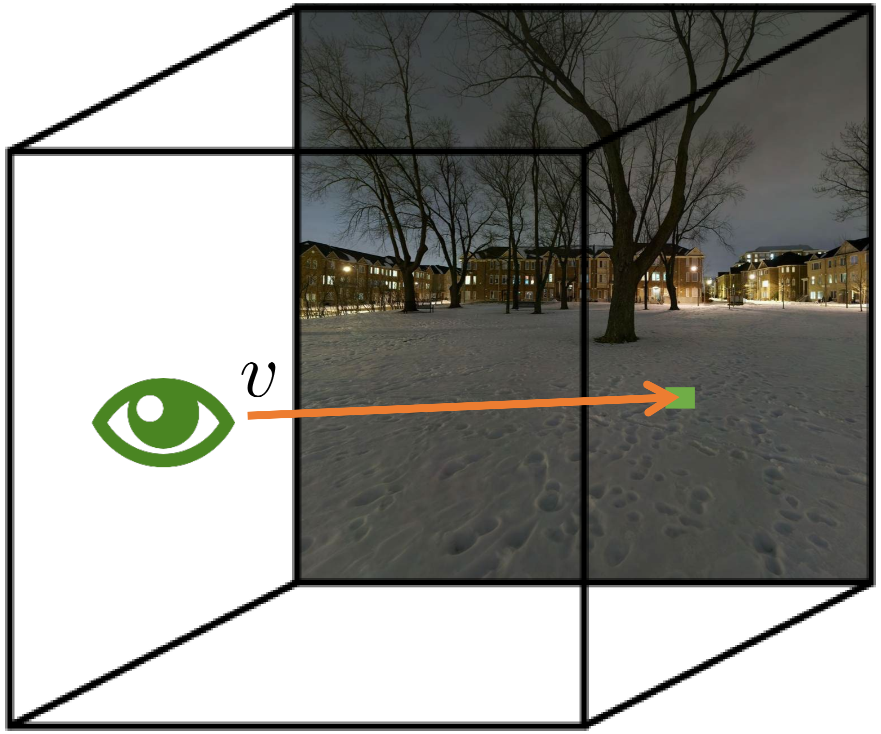
Skybox



Skybox

- Viewer in the center of environment.
- Box is background
  (e.g. sky, clouds, ...).
- A cube for rendering.

- Needs: view, projection matrix
  Note: no model matrix!



The Vertex Shader

```
in vec3 aPos;
uniform mat4 projection;
uniform mat4 view;
out vec3 WorldPos;

void main()
{
   WorldPos = aPos;

   mat4 rotView = mat4(mat3(view));
   vec4 clipPos = projection * rotView * vec4(WorldPos, 1.0);

   gl_Position = clipPos.xyww;
}
```

The Vertex Shader

skybox.vert

```
in vec3 aPos;
uniform mat4 projection;
uniform mat4 view;
out vec3 WorldPos;

void main()
{
   WorldPos = aPos;

   mat4 rotView = mat4(mat3(view));
```

Removes the translation from the view matrix.
We assume the viewer is always at the center of the box.

The Vertex Shader

skybox.vert

```
in vec3 aPos;
uniform mat4 projection;
uniform mat4 view;
out vec3 WorldPos;

void main()
{
   WorldPos = aPos;

   mat4 rotView = mat4(mat3(view));
   vec4 clipPos = projection * rotView * vec4(WorldPos, 1.0);

   gl_Position = clipPos.xyww;
```

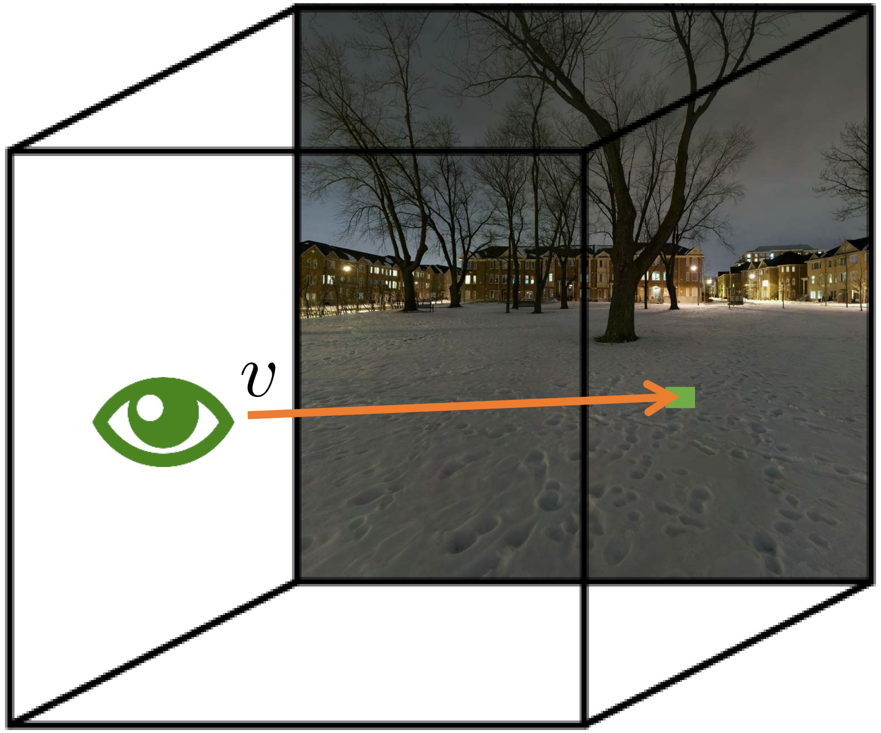
The skybox is overdrawn by all other objects (1 in the depthbuffer).
Note: adjust the depth test to ≤ instead of <

The Fragment Shader

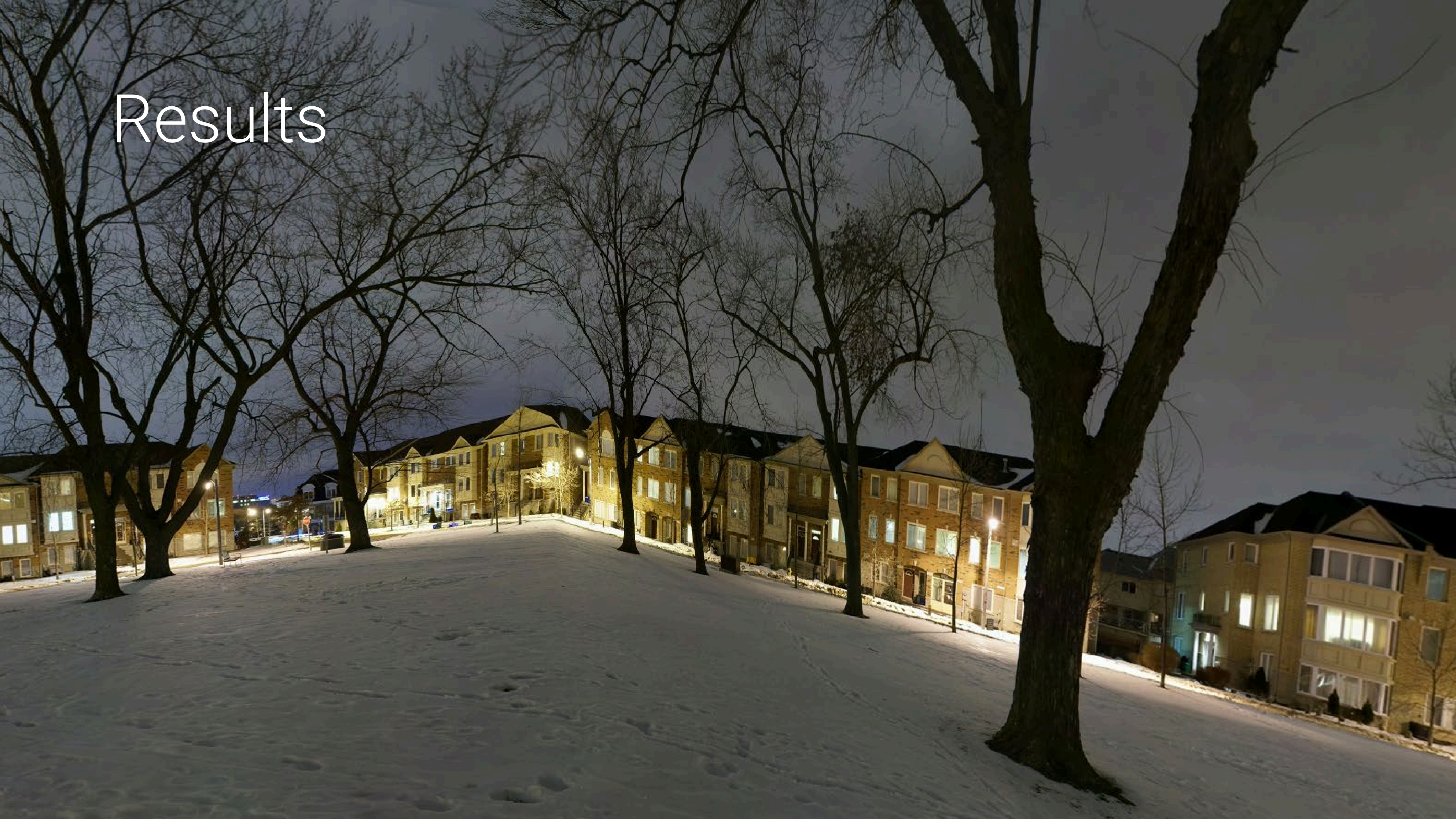
skybox.frag

```
out vec4 FragColor;
in vec3 WorldPos;

uniform samplerCube skybox;

void main()
{
    FragColor = texture(skybox, WorldPos);
}
```

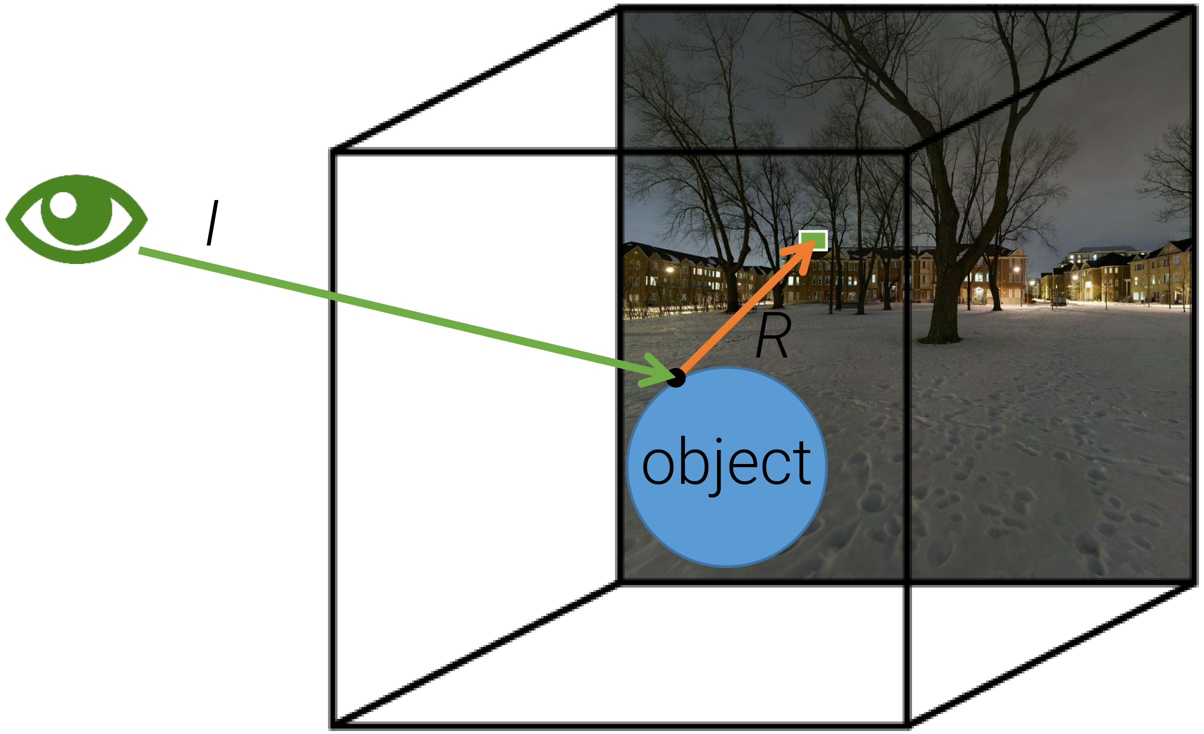
We use the non-transformed cube vertices to access the cubemap texels (3D lookup). Remember: viewer is at origin (0,0,0).



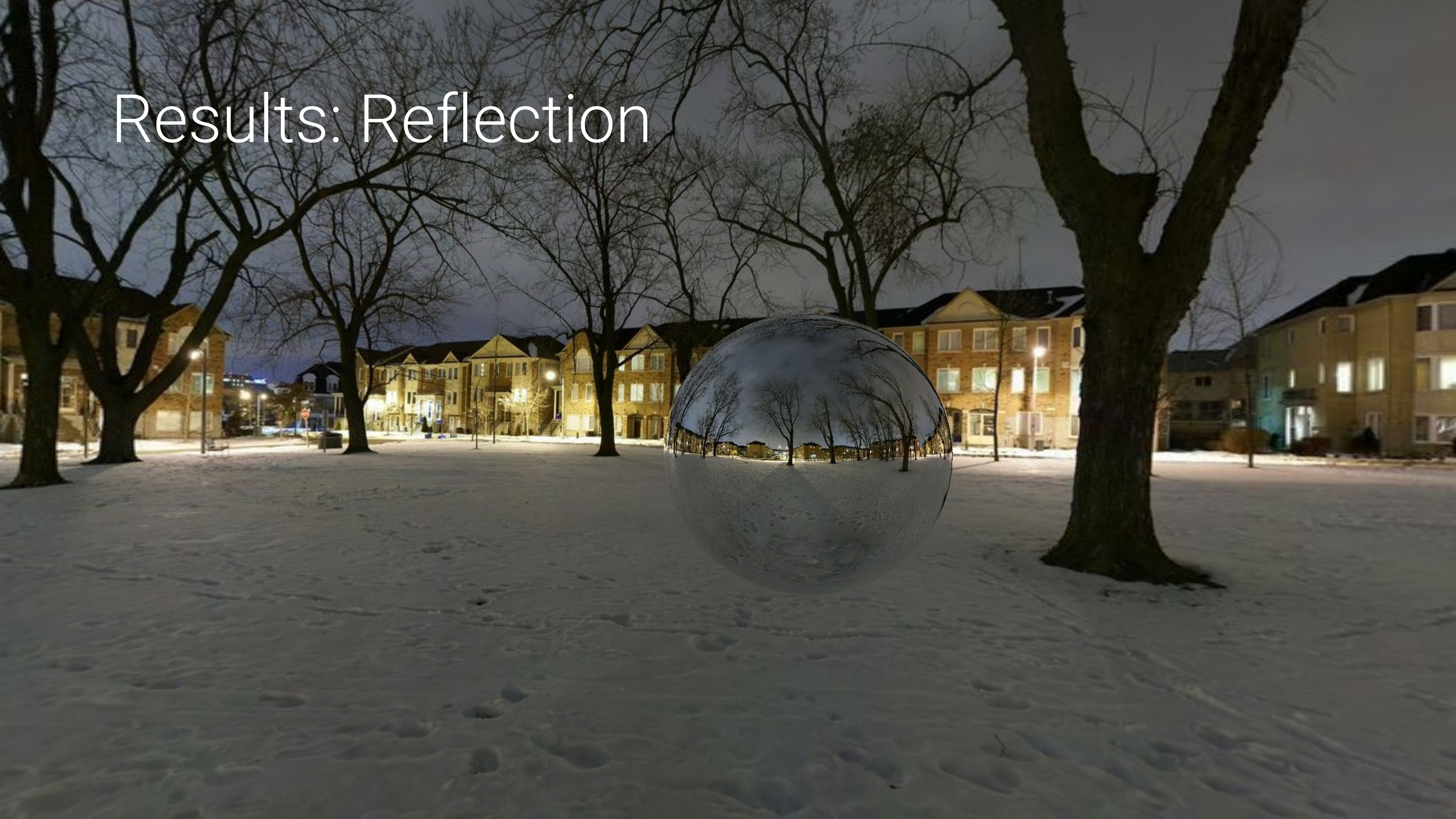
Results



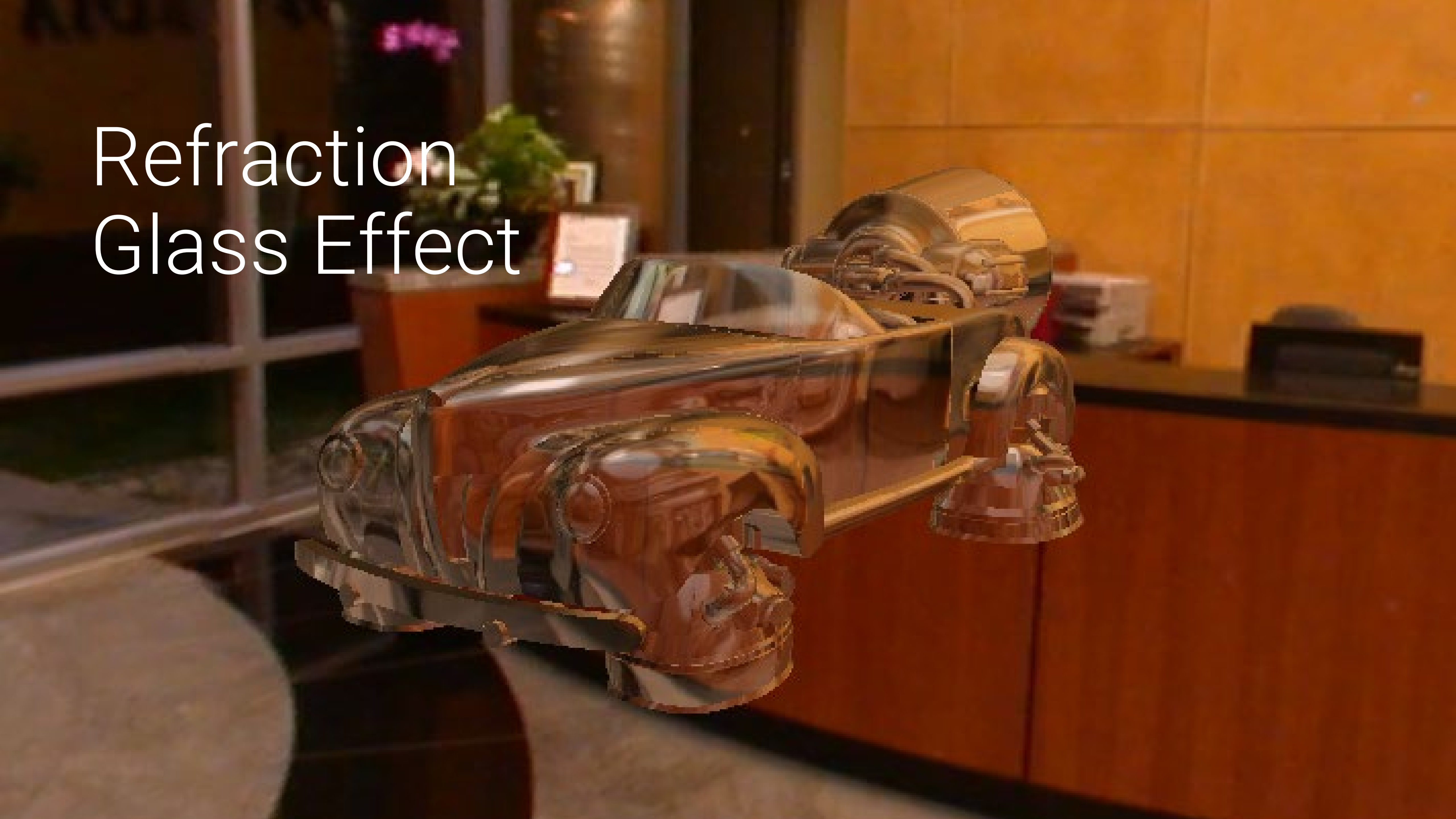
Cube Environment Mapping



Results: Reflection

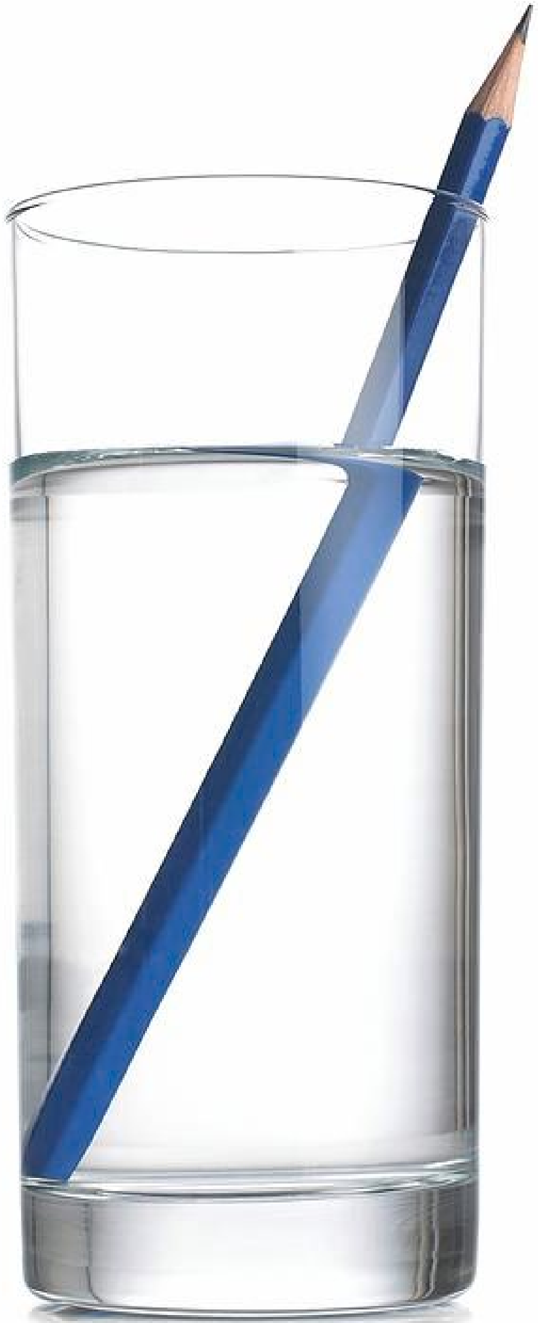


Refraction Glass Effect

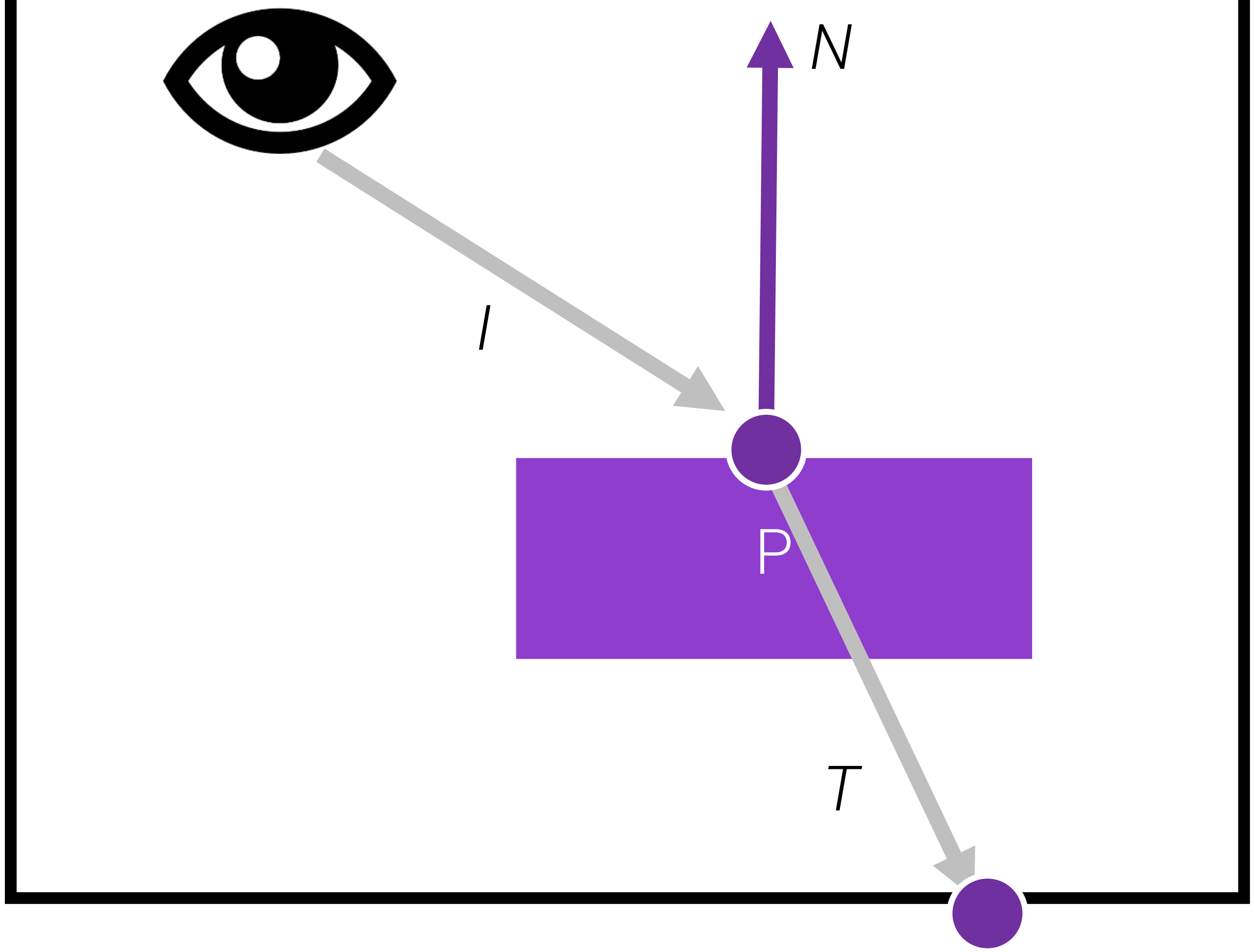


Refraction

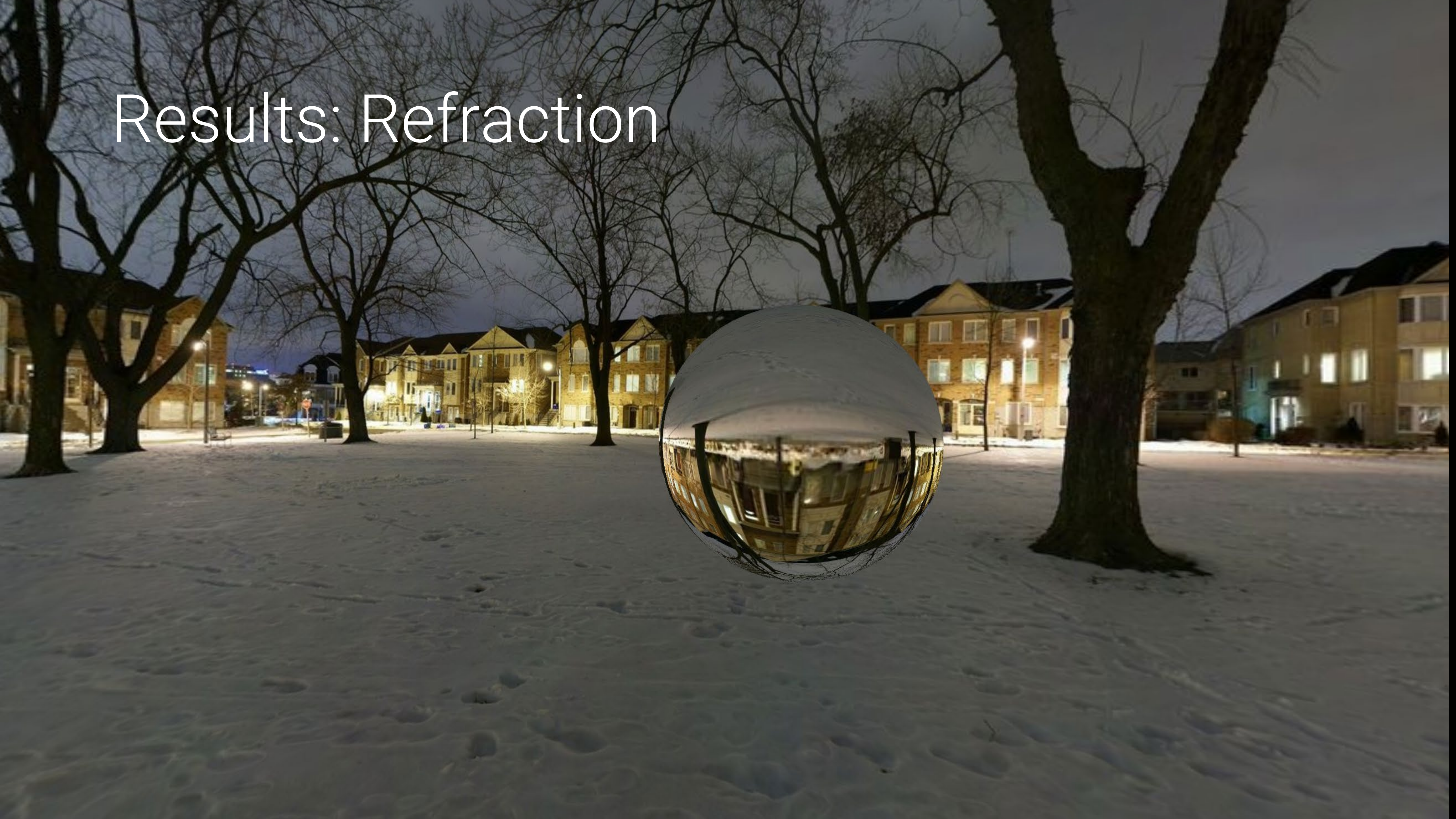
When light passes through a boundary between two materials of different density (air, water, etc.) the light‘s direction changes



Computing Refraction Vector

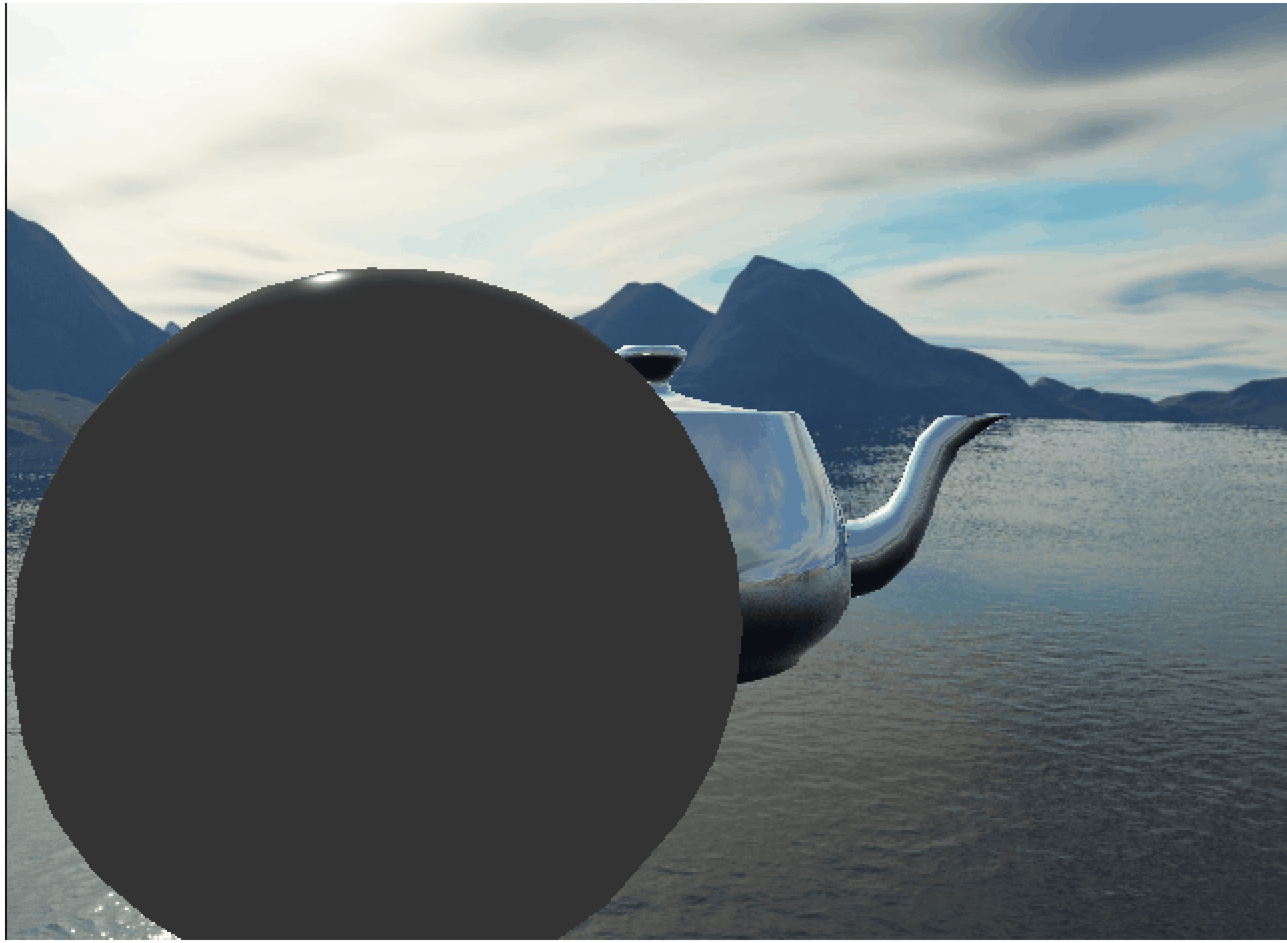


Results: Refraction



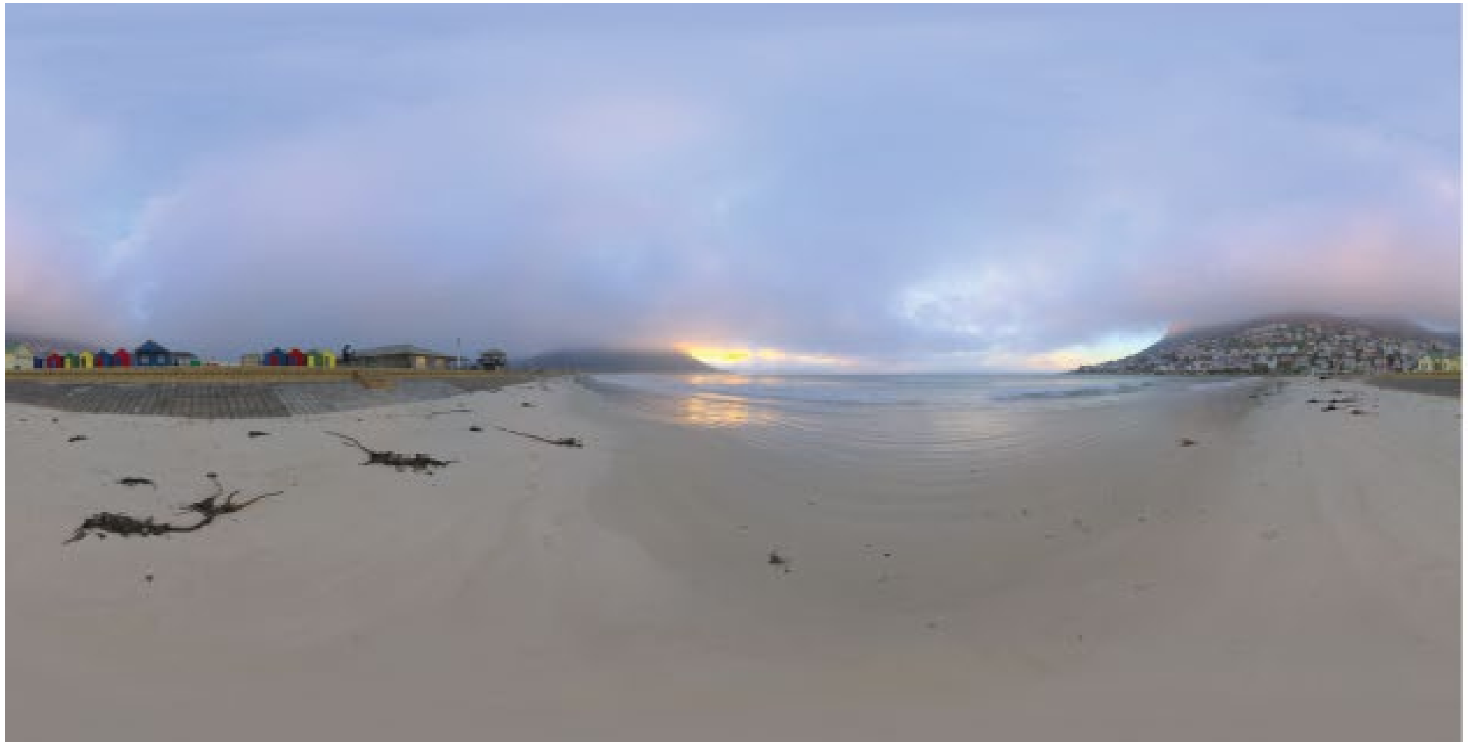
Dynamic Cube Maps

- Faces of a cube map are only six views of a scene.
- So we can generate them dynamically, too.
- Effect: reflections show scene objects (not only background).

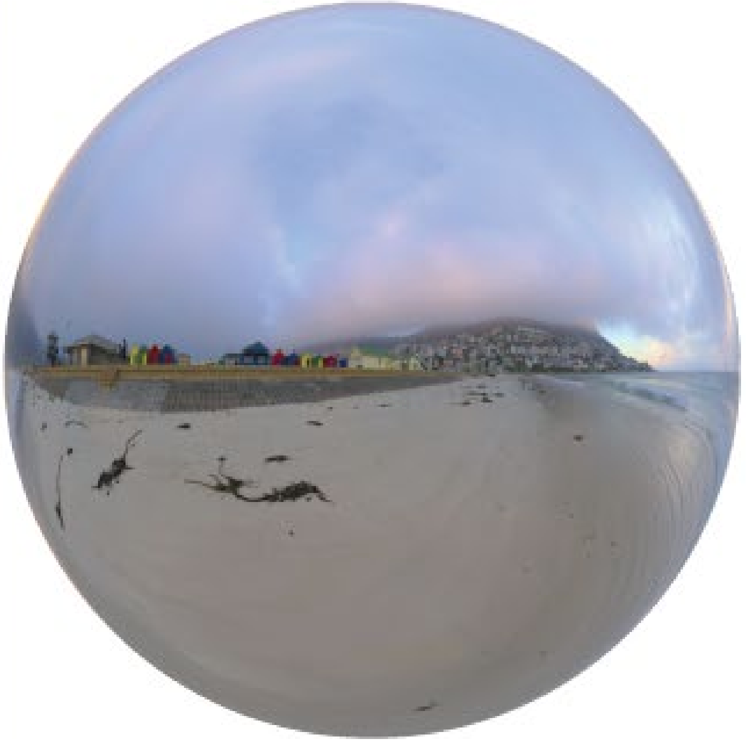


https://khongton.github.io/Dynamic-Cubemaps/

Other Formats

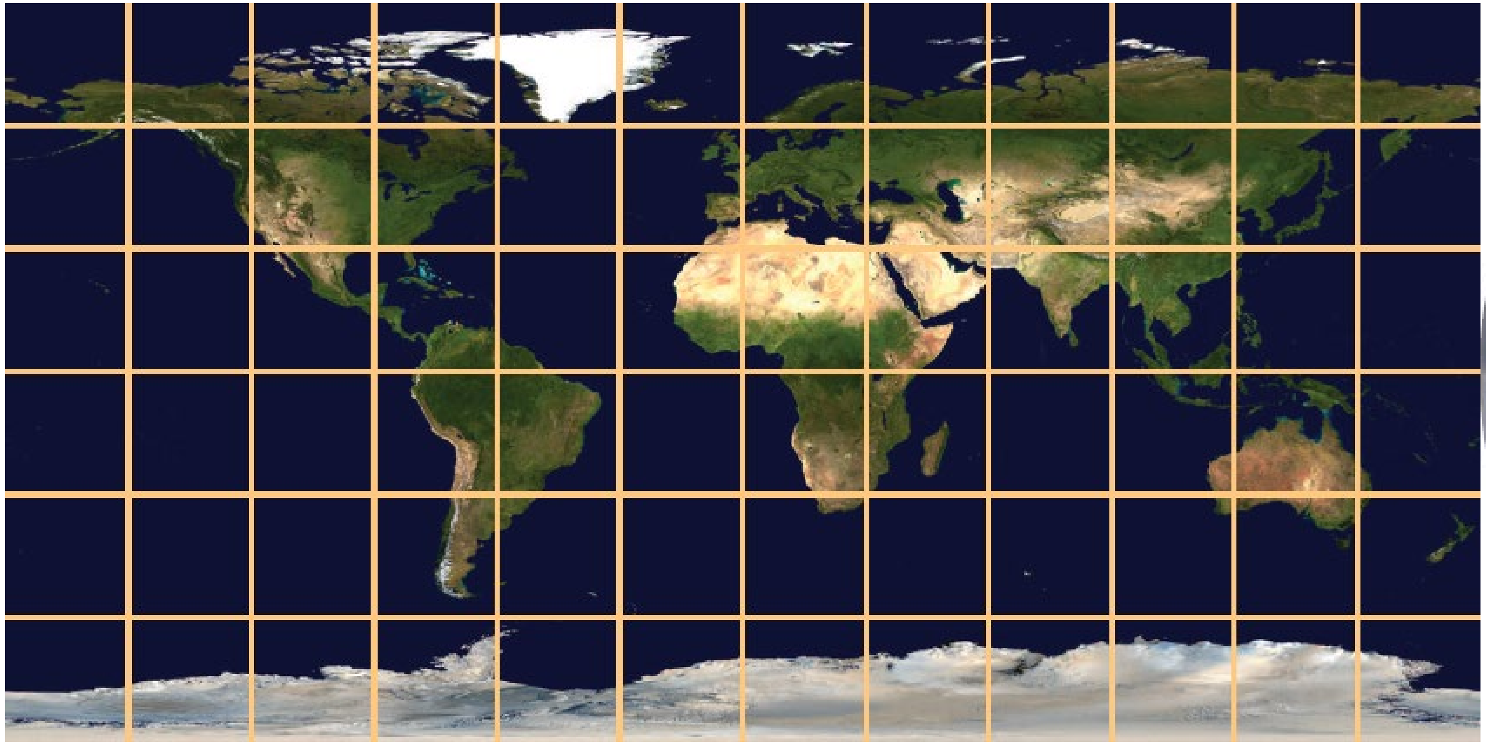


latitude-longitude format

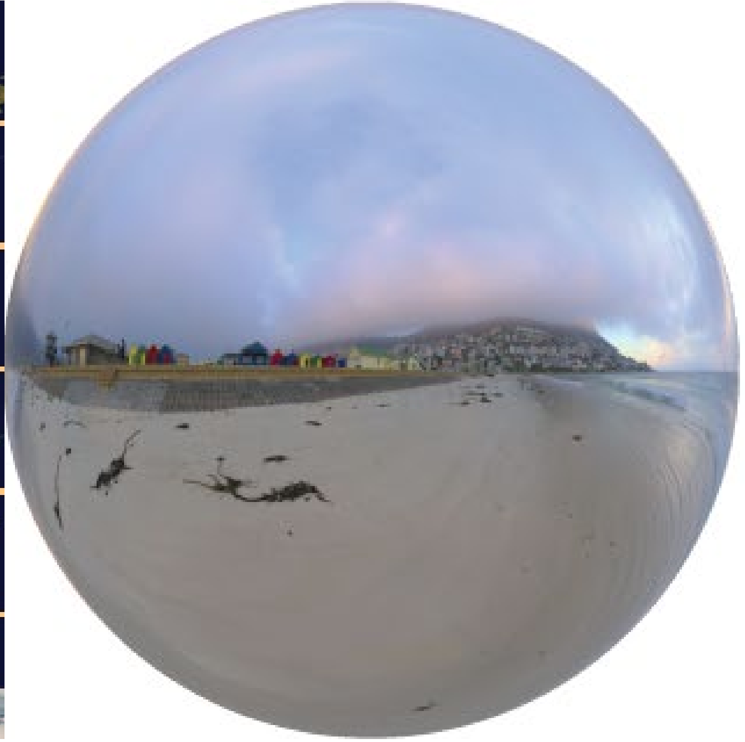


sphere map

Other Formats



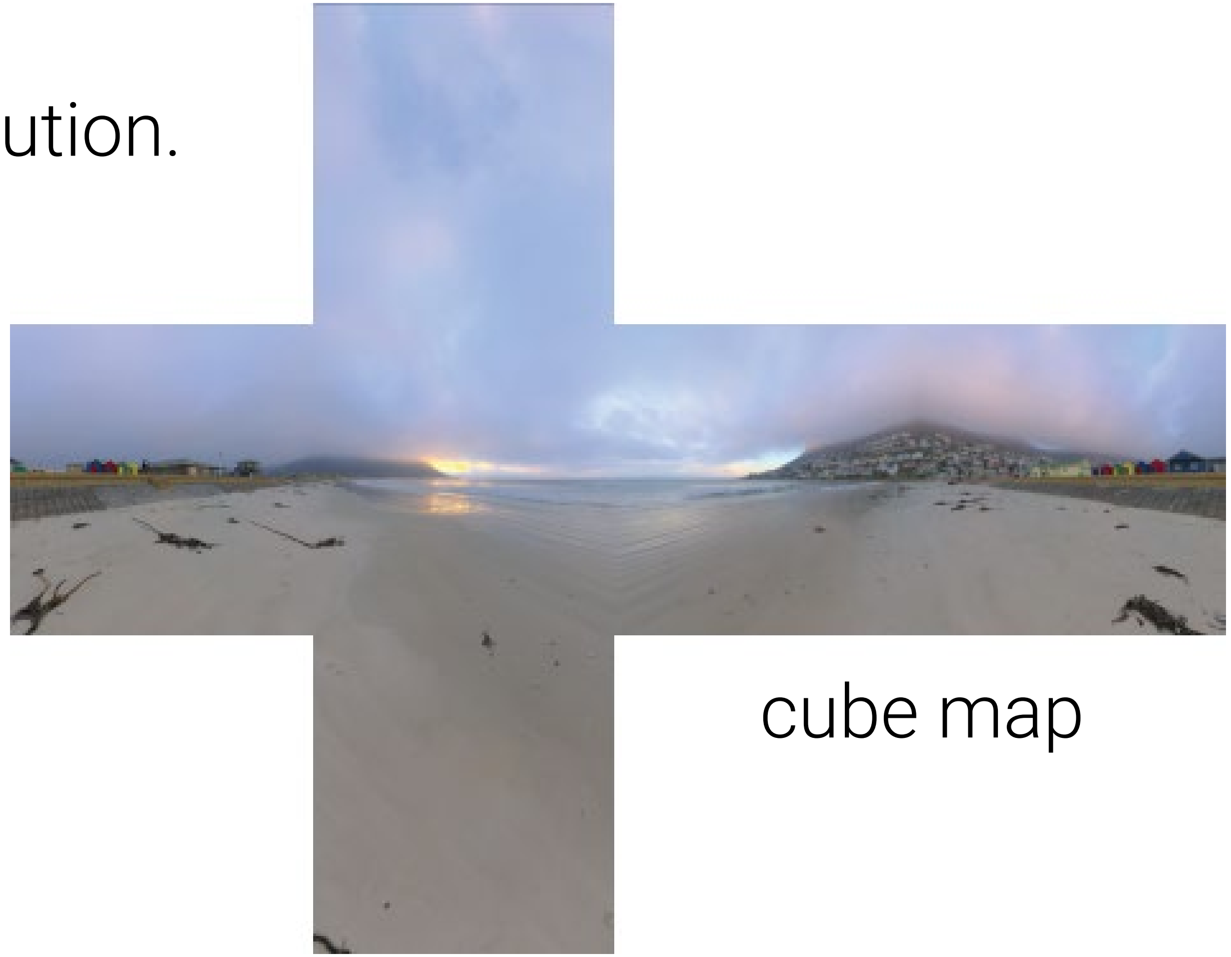
latitude-longitude of the earth



sphere map

Other Formats - Conversion

- Formats can be converted, but might lead to losses in resolution.
  - lat-lon maps have problems at the top and bottom. (we'll use them too)
  - sphere maps loose information at sphere edges (behind sphere)



cube map

Questions?

david.schedl@fh-hagenberg.at