Cubemapping

CGT 520

Cubemapping

- A texture that uses 3D unit vectors as texture coordinates
- Can use world-space reflection vector as tex coords to simulate the appearance of reflections
- Can texture a cube to implement a skybox
- Fast hardware support in OpenGL, unlike some other approaches (e.g. Blinn and Newell environment maps and sphere maps)

Environment mapping 1976

- Blinn and Newell's original idea

Blinn, J. F. and Newell, M. E. Texture and reflection in computer generated images. Communications of the ACM Vol. 19, No. 10 (October 1976), 542-547.

  - Simply use the spherical coordinates ρ, Φ as texture coordinates
  - Expensive to compute
  - High distortion at poles
  - Not commonly used

$$u = \rho = \arccos(-r_x)$$

$$v = \phi = \arctan\left(\frac{r_y}{r_z}\right)$$

Sphere Map Textures

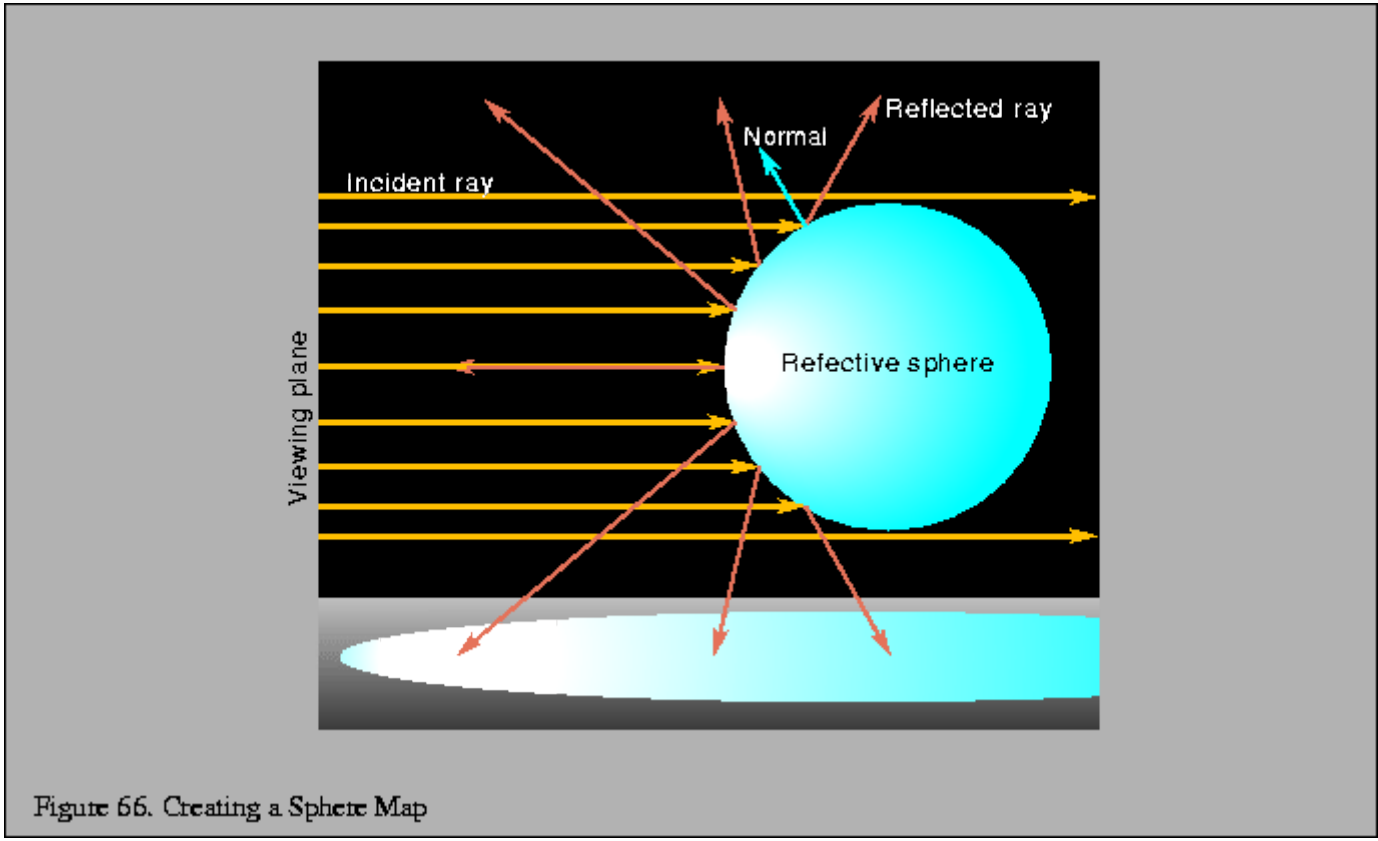


Figure 66. Creating a Sphere Map

$$p = \sqrt{r_x^2 + r_y^2 + (r_z + 1)^2}$$

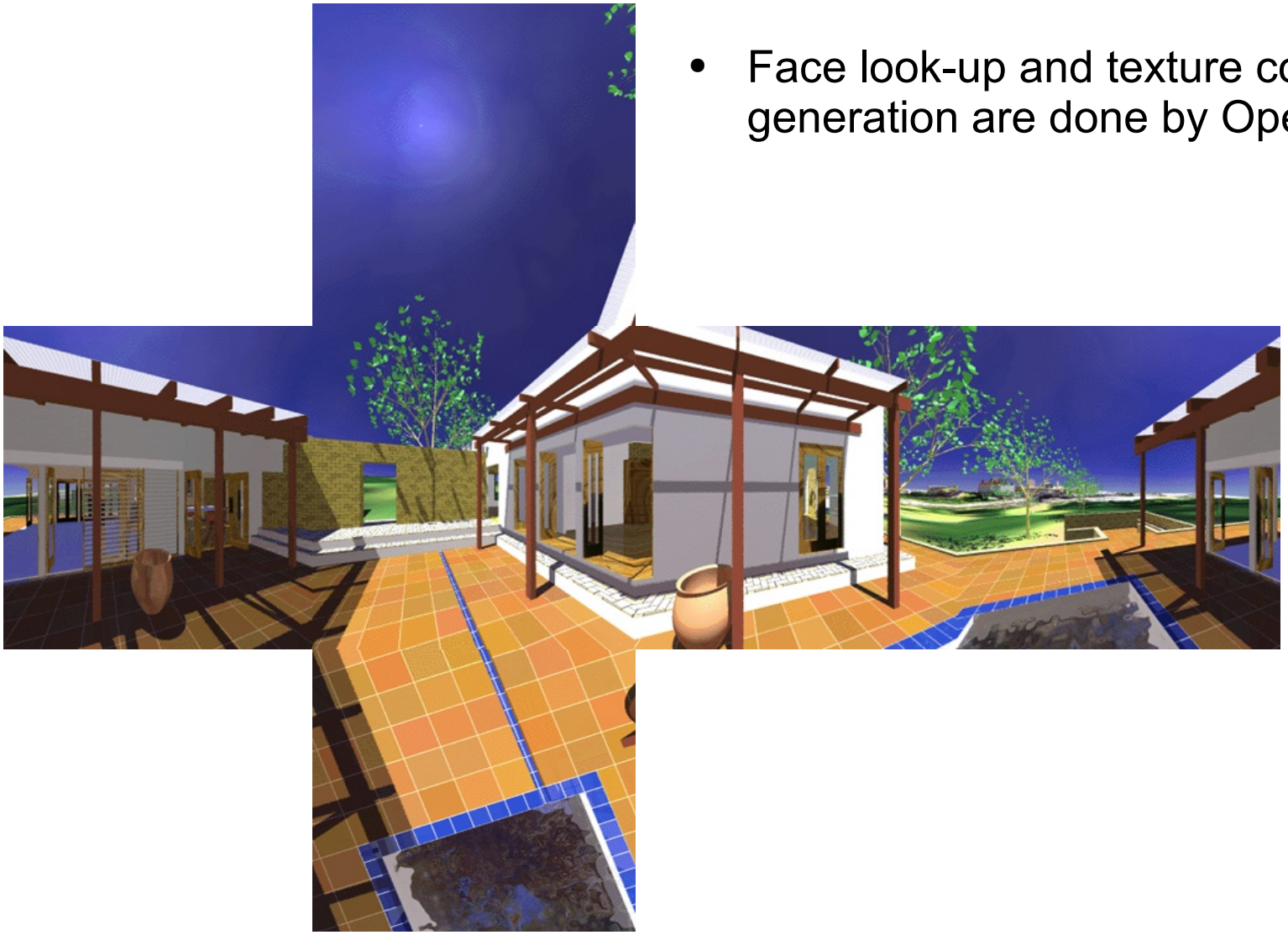
$$u = \frac{r_x}{2p} + \frac{1}{2}$$

$$v = \frac{r_y}{2p} + \frac{1}{2}$$

- Some pixels unmapped
- High distortion near edges of sphere
- Only accurate for a single view point

Cube Map Textures

One texture image for each face of a cube.



- Face look-up and texture coordinate generation are done by OpenGL

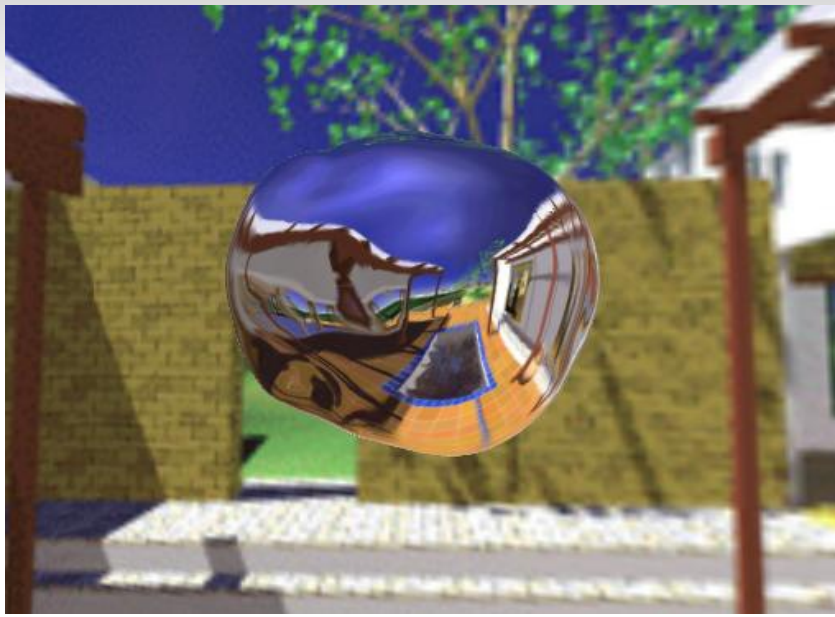
Cubemapping in OpenGL

- New texture object target:
  - **GL_TEXTURE_CUBE_MAP**
  - Use this target for `glBind, glTexParameter`
- Six new texture image targets:
  - **GL_TEXTURE_CUBE_MAP_POSITIVE_X**
  - **GL_TEXTURE_CUBE_MAP_NEGATIVE_X**
  - **GL_TEXTURE_CUBE_MAP_POSITIVE_Y**
  - **GL_TEXTURE_CUBE_MAP_NEGATIVE_Y**
  - **GL_TEXTURE_CUBE_MAP_POSITIVE_Z**
  - **GL_TEXTURE_CUBE_MAP_NEGATIVE_Z**
- Setting a cube face texture image
  - `glTexImage2D(`**`GL_TEXTURE_CUBE_MAP_POSITIVE_X`**`, level, internalformat, width, height, border, format, type, *data);`

Creating a Cube Map Texture in OpenGL

- Generate an unused ID
  - `glGenTextures(1, &CubeID);`
- Bind ID to Cube Map Target
  - `glBindTexture(GL_TEXTURE_CUBE_MAP, CubeID);`
- Load Images for each face
  - `glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_X, level, internalformat, width, height, border, format, type, *data);`
  - +5 more faces
- Set Parameters
  - Wrap mode
    - `glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);`
    - `glTexParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);`
  - Filtering

Cubemapping in OpenGL



To render cubemapped reflections:

- Bind Texture ID
  - `glBindTexture(GL_TEXTURE_CUBE_MAP, CubeID);`
- Draw geometry (needs normal vectors)

  - Declare samplerCube texture object in shader.

    ```
    uniform samplerCube cubetex;
    ```

  - You use a vector instead of tex coords.

    ```
    vec3 rv = reflect(-v, n); //reflected view vector
    vec4 reflectionColor = textureCube(cubetex, rv.xyz);
     //use this reflection color as specular lighting term
    ```

Another cubemap use: sky box

- Only distant environment should be represented in skybox.
- Simply apply texture to a box surrounding the camera