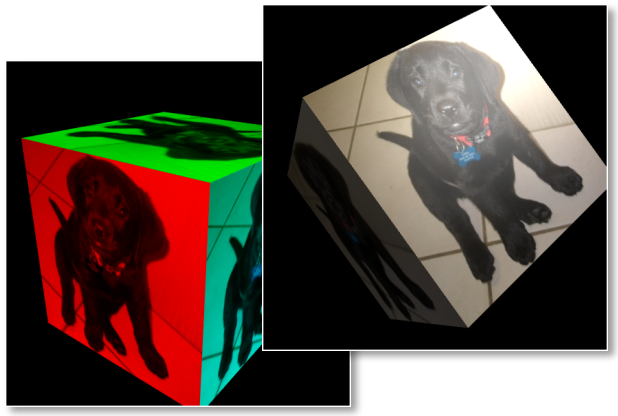
Getting fancy with texture mapping (Part 1)

CS559 – Fall 2017

4 Apr 2017

Texturing and lighting

jsbin.com/pafugaqoho/edit
jsbin.com/hejizelule/edit



fragmentShader

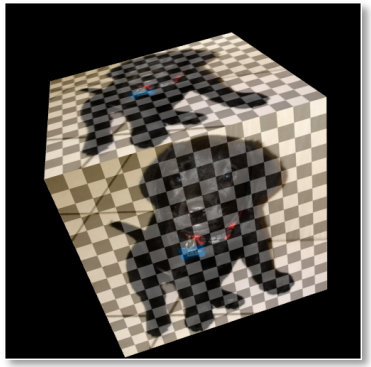
```
varying vec3 fNormal;
varying vec2 fTexCoord;
uniform sampler2D texSampler;

vec3 blinnPhongDir(...)

void main(void) {
  vec3 texColor=texture2D(texSampler1,fTexCoord).xyz;
  vec3 n = (uMVn * vec4(fNormal, 0.0)).xyz;
  vec3 ColorS  = blinnPhongDir(lightV,fNormal,....specular coeffs....).y*lightCol;
  vec3 ColorAD = blinnPhongDir(lightV,fNormal,..ambient/diff coeffs..).x*texColor;
  gl_FragColor = vec4(ColorAD+ColorS,1.0); }
```

jsbin.com/weluvebodu/edit

Multiple textures



fragmentShader

```
varying vec2 fTexCoord;
uniform sampler2D texSampler1;
uniform sampler2D texSampler2;

void main(void) {
   vec4 texColor1 = texture2D(texSampler1,fTexCoord);
   vec4 texColor2 = texture2D(texSampler2,fTexCoord);
   gl_FragColor = vec4(0.6*texColor1.xyz+0.4*texColor2.xyz,texColor1.a);}
```

.js

```
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```

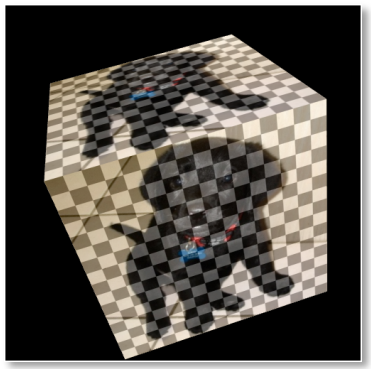
```
shaderProgram.texSampler1 = gl.getUniformLocation(shaderProgram, "texSampler1");
gl.uniform1i(shaderProgram.texSampler1, 0);
shaderProgram.texSampler2 = gl.getUniformLocation(shaderProgram, "texSampler2");
gl.uniform1i(shaderProgram.texSampler2, 1);
```

```
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture1);
gl.activeTexture(gl.TEXTURE1);
gl.bindTexture(gl.TEXTURE_2D, texture2);
gl.drawElements(gl.TRIANGLES, triangleIndices.length, gl.UNSIGNED_BYTE, 0);
```

jsbin.com/weluvebodu/edit

Multiple textures

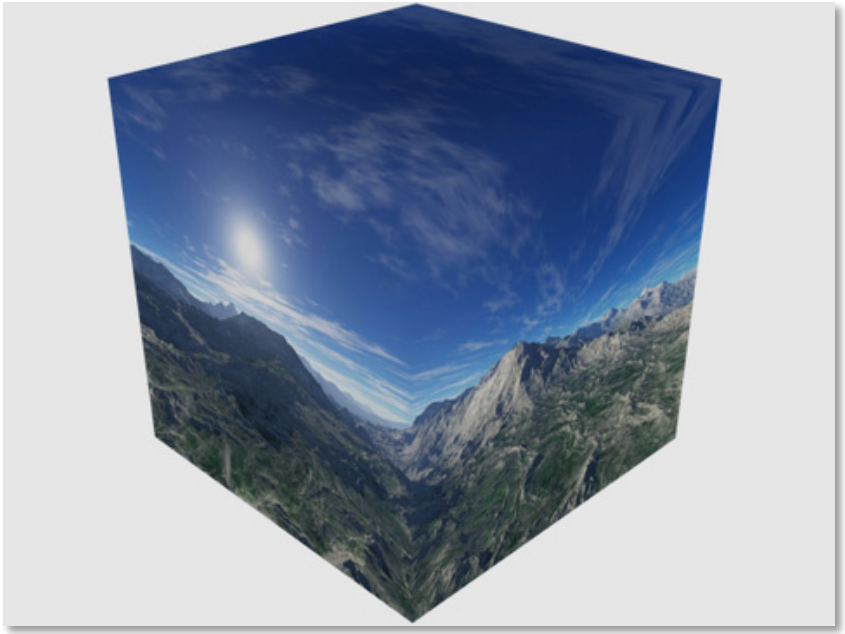
```
.js
start(){
  initShaders();
  sendData();
  initTextures();
  draw()
}
```



```
var texture1 = gl.createTexture();
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture1);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
var image1 = new Image();
var texture2 = gl.createTexture();
gl.activeTexture(gl.TEXTURE1);
gl.bindTexture(gl.TEXTURE_2D, texture2);
gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, null);
var image2 = new Image();

image1.onload = function() { loadTexture(image1,texture1); };
image1.src = "http://myurl.com/spirit.jpg";
image2.onload = function() { loadTexture(image2,texture2); };
image2.src = "http://myurl.com/checkerboard.jpg";
```

Flash preview ...



Skyboxes as backdrops

Credits : Flipmode 3D

Textures beyond color?

Surface roughness?

(bump maps, displacement maps)

Reflected light?

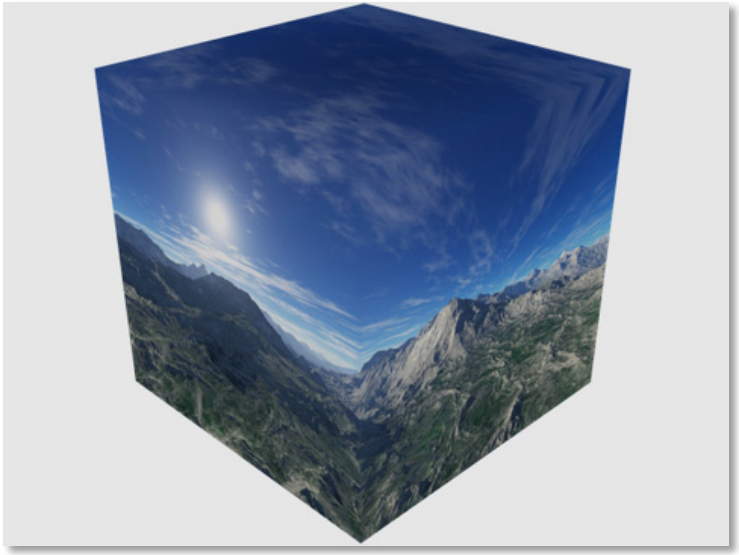
(environment maps)

Shininess?

Tranparency?

Shadows?

Skybox



Very simple effect

Exposes interesting tools
(cube maps)

Stepping stone for more complex effects
(environment/reflection mapping)

Emulates appearance of **faraway** surrounding environment

Skybox/Cubemaps

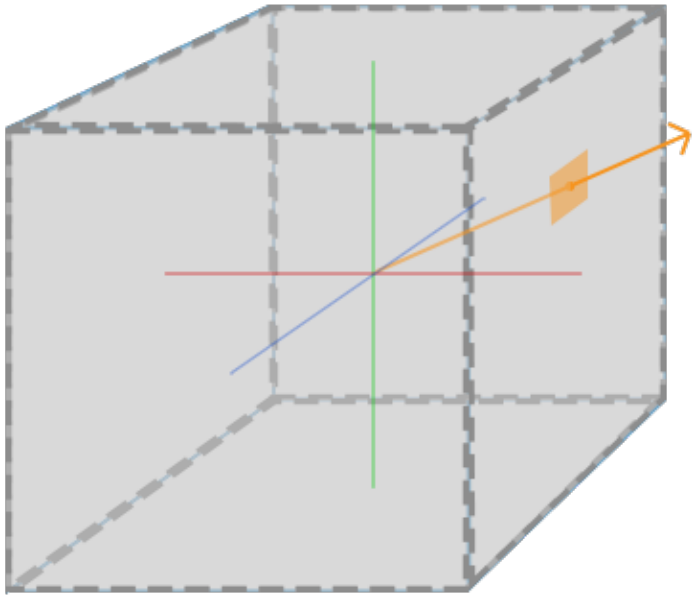
Camera surrounded by large, textured cube

(could be sphere/cylinder, but cube is much easier)

Hardware helps with lookups

Can query texture with just a **vector** originating at the center of the cube

Proper side is determined automatically



Skybox/Cubemaps

Texture seamlessly covers cube boundary

Skybox/Cubemaps

Downsides:

- No parallax
- Not animated (?)
- Objects always far away (?)
- Looks recognizably “flat” (?)

[Example] [Parallax]

Cubemaps (implementation)

```
var texture=gl.createTexture();
gl.bindTexture(gl.TEXTURE_CUBE_MAP, texture);
gl.texImage2D(gl.TEXTURE_CUBE_MAP_POSITIVE_X,
   0,gl.RGBA,gl.RGBA, gl.UNSIGNED_BYTE, image);
gl.texParameteri(gl.TEXTURE_CUBE_MAP,gl.TEXTURE_WRAP_S,GL_CLAMP_TO_EDGE);
```

Fill skybox with six images

Mipmaps/clamping as usual

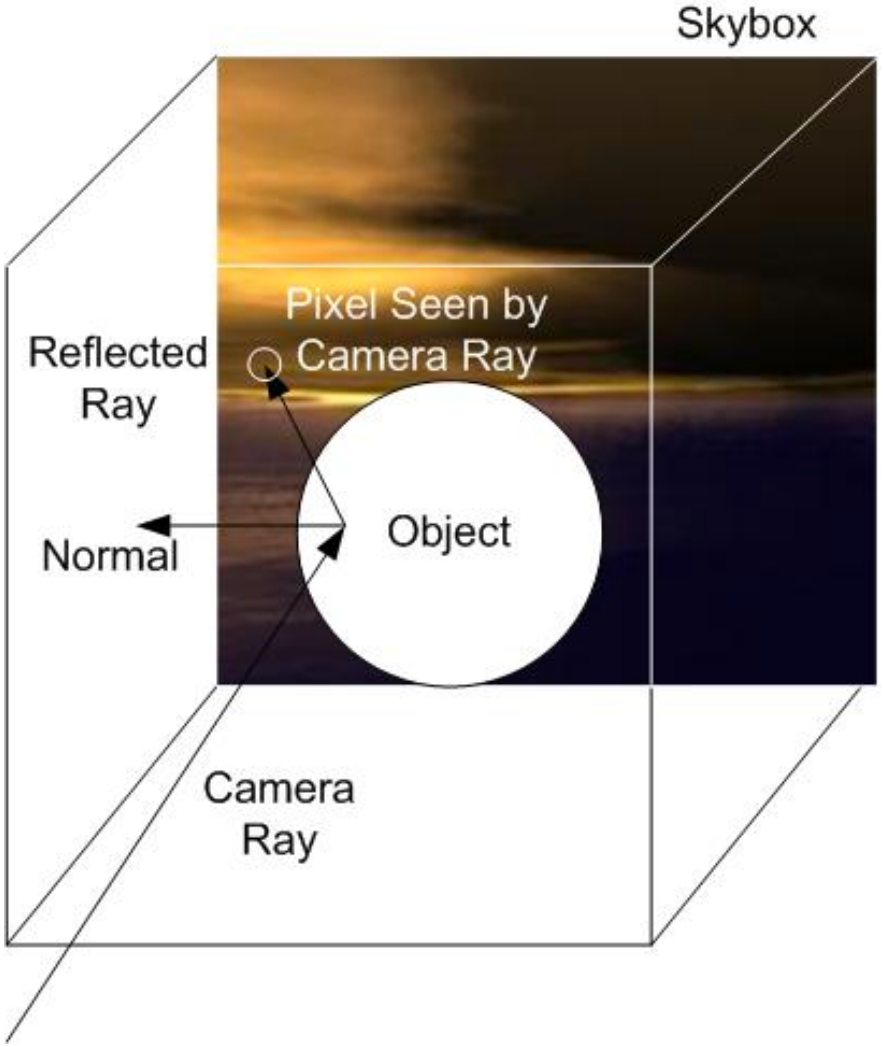
Options
gl.TEXTURE_CUBE_MAP_POSITIVE_{X|Y|Z}
gl.TEXTURE_CUBE_MAP_NEGATIVE_{X|Y|Z}

Environment mapping

Credits: NVidia

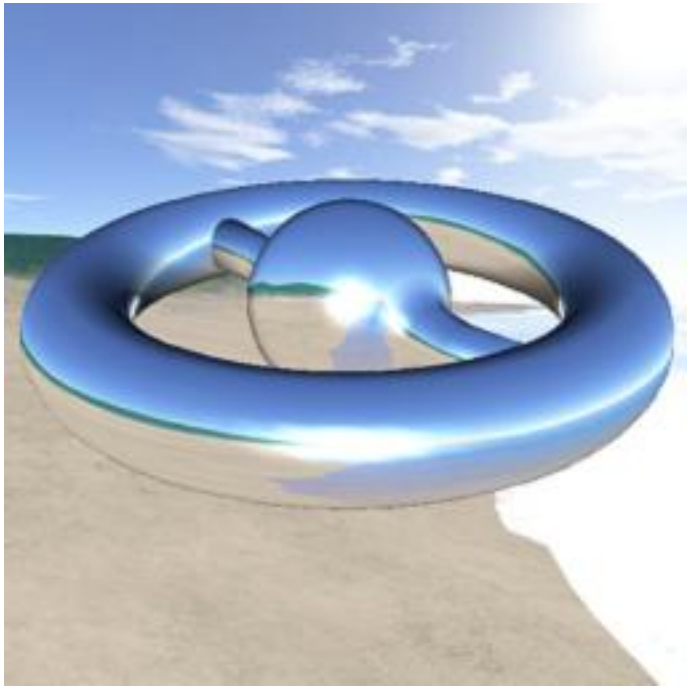
Environment mapping

Cubemap texture provides reflected color information on objects



Environment mapping

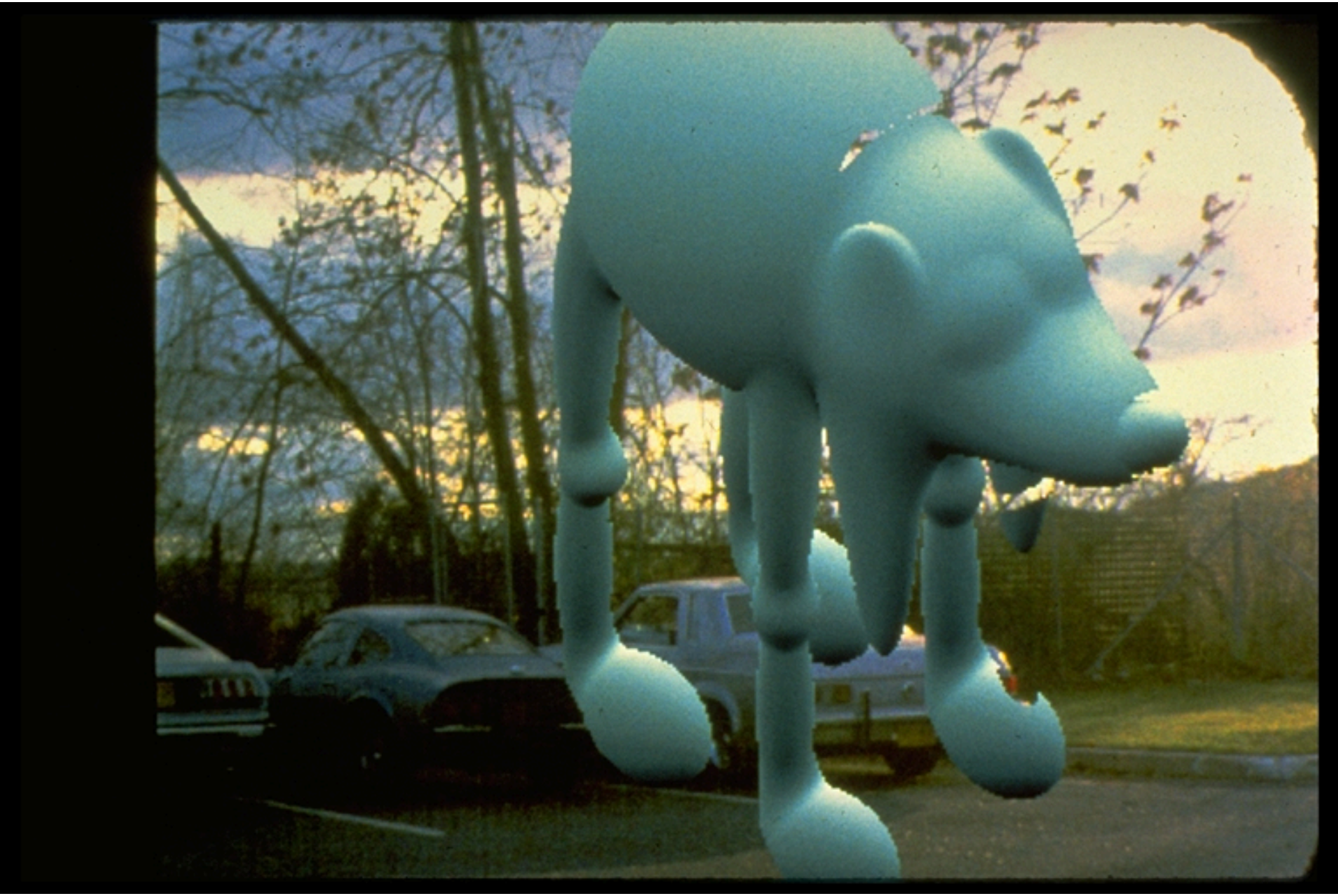
Cubemap texture provides reflected color information on objects



[Example]

Capturing reflection maps

Art by M.C. Escher



Credits: G. Miller/P. Debevec

Credits: G. Miller/P. Debevec