Supplement to Lecture 19

Environmental/Bump Maps

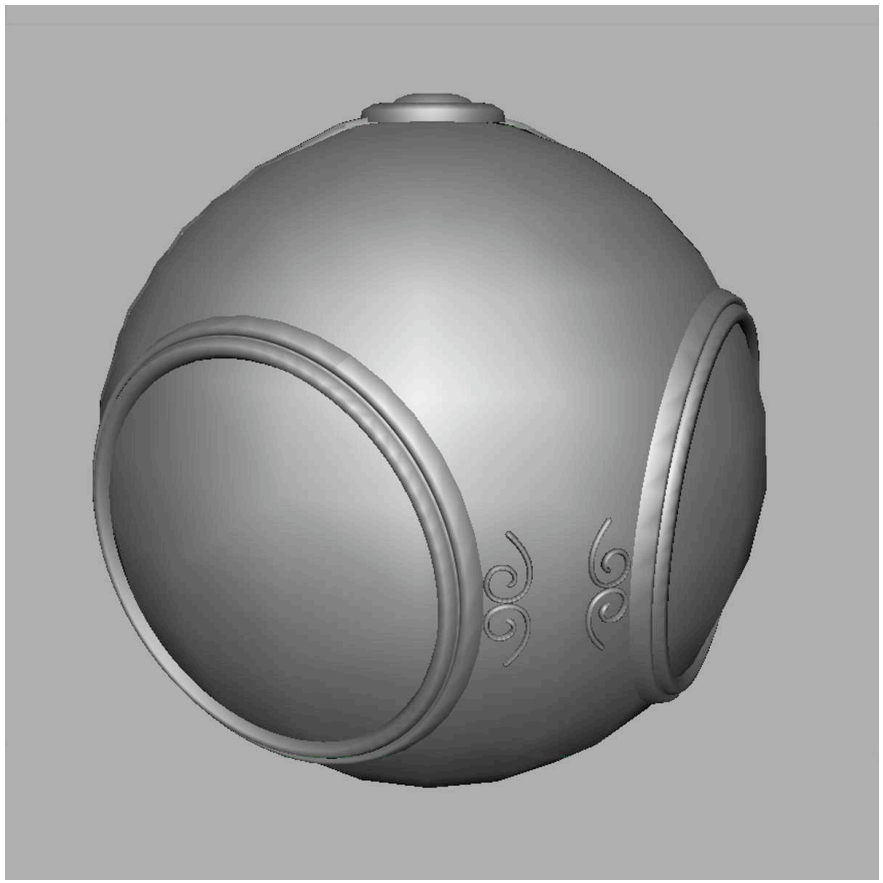
CS 354 Computer Graphics
http://www.cs.utexas.edu/~bajaj/
Department of Computer Science
University of Texas at Austin

Notes and figures from *Ed Angel: Interactive Computer Graphics, 6th Ed., 2012 © Addison Wesley*

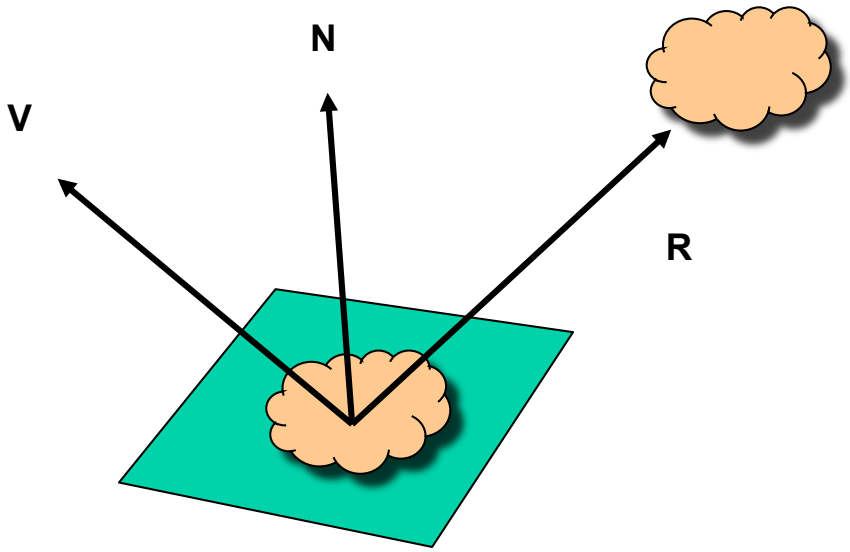
Environment Mapping

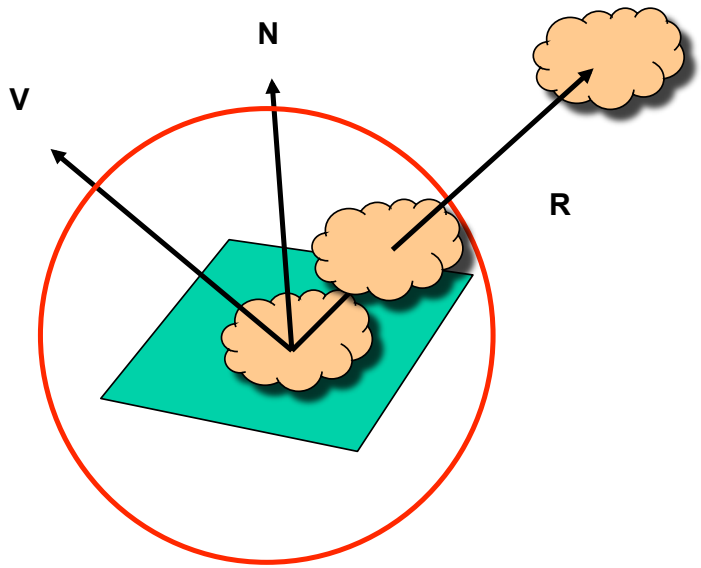
- Environmental mapping is way to create the appearance of highly reflective surfaces without ray tracing which requires global calculations
- Examples: The Abyss, Terminator 2
- Is a form of texture mapping
  - Supported by OpenGL and Cg

Example



Reflecting the Environment





Mapping to a Sphere

Hemisphere Map as a Texture

- If we map all objects to hemisphere, we cannot tell if they are on the sphere or anywhere else along the reflector
- Use the map on the sphere as a texture that can be mapped onto the object
- Can use other surfaces as the intermediate
  - Cube maps
  - Cylinder maps

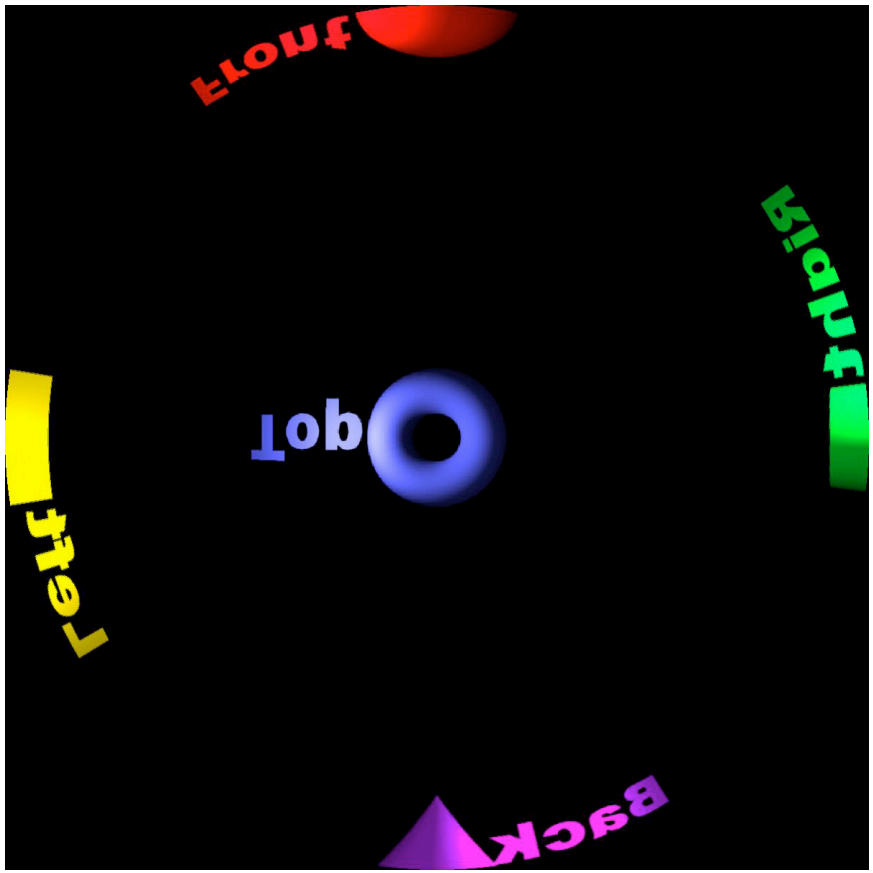
Issues with Sphere Maps

- Must assume environment is very far from object (equivalent to the difference between near and distant lights)
- Object cannot be concave (no self reflections possible)
- No reflections between objects
- Need a reflection map for each object
- Need a new map if viewer moves

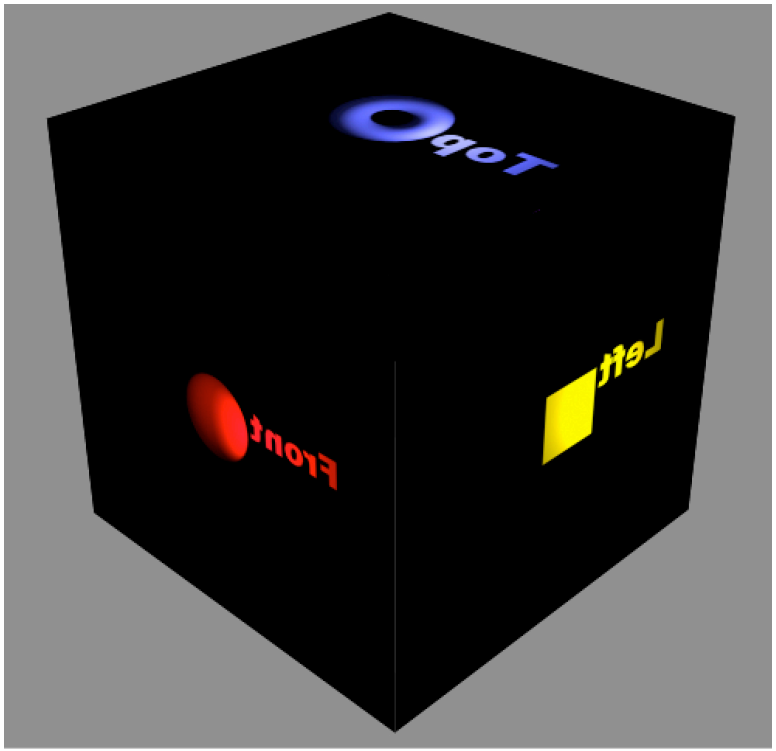
OpenGL Implementation

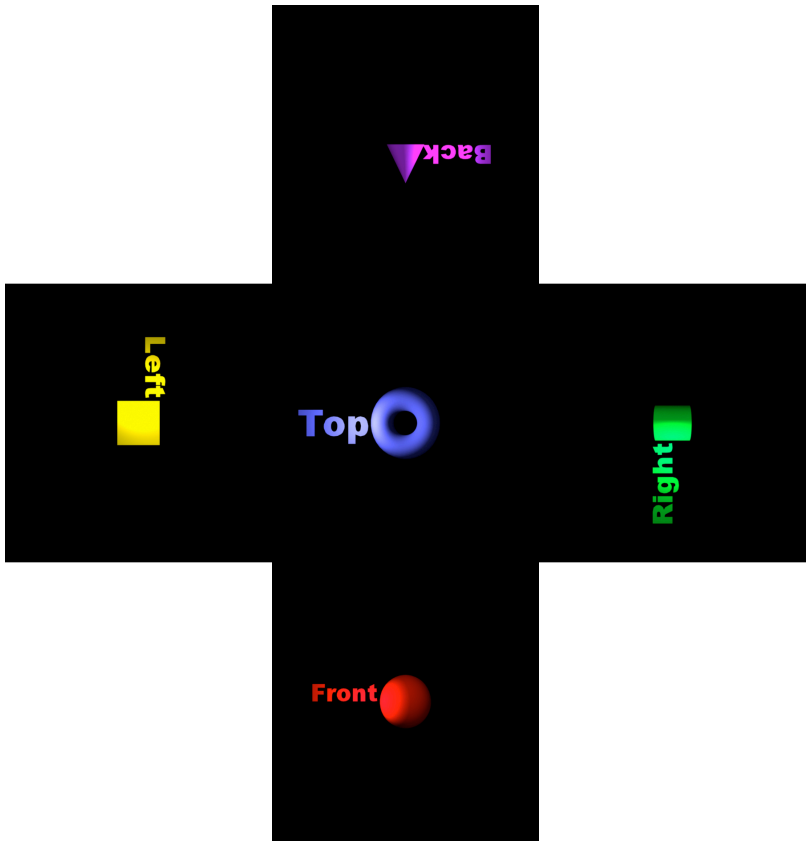
- OpenGL supports spherical and cube maps
- First must form map
  - Use images from a real camera
  - Form images with OpenGL
- Texture map it to object

Spherical Map

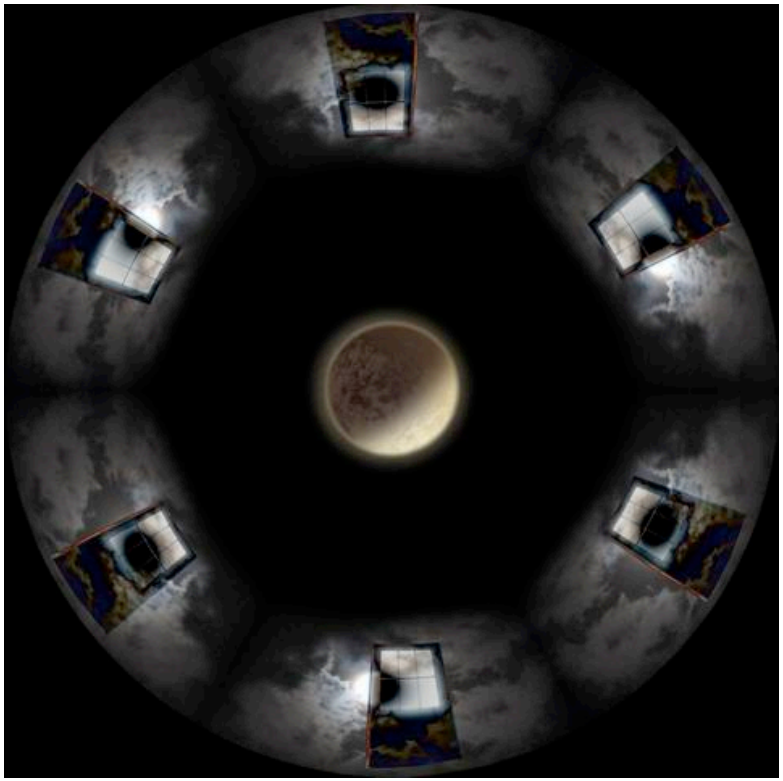


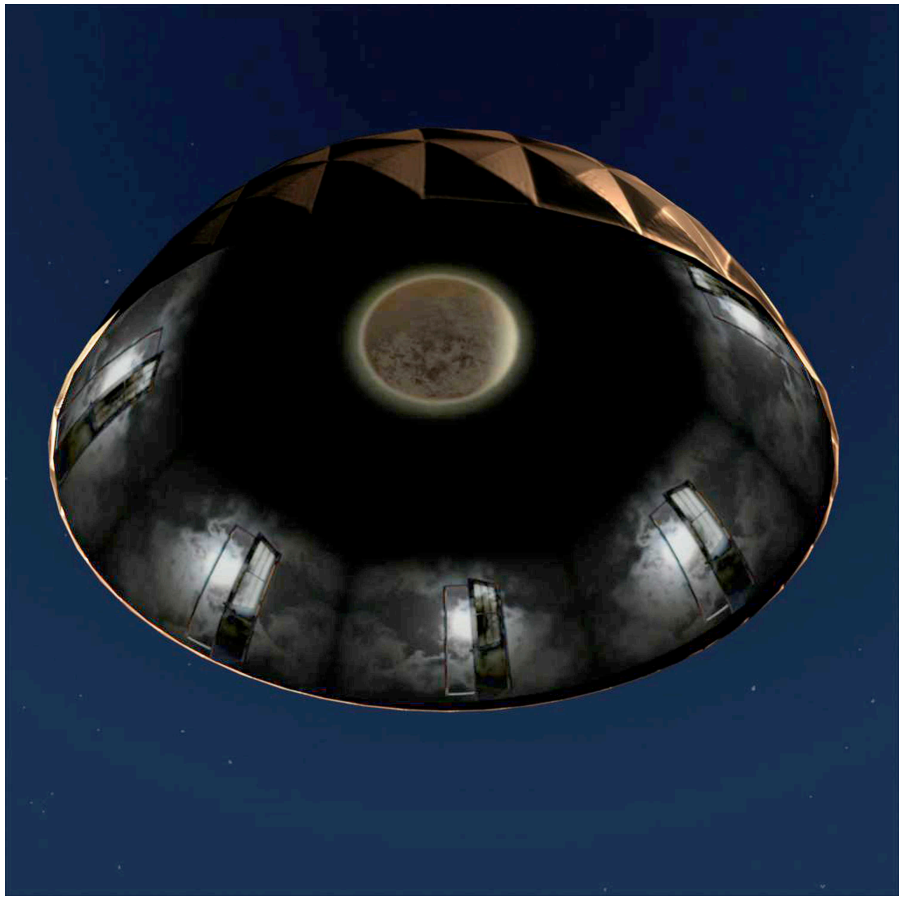
Cube Map





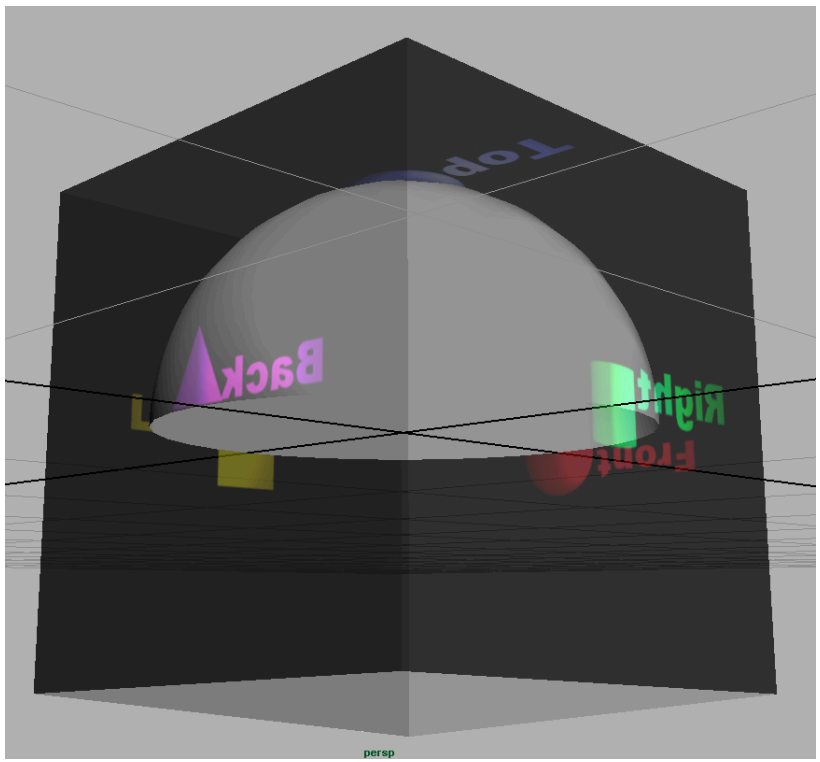
Hemisphere Map & Final Image

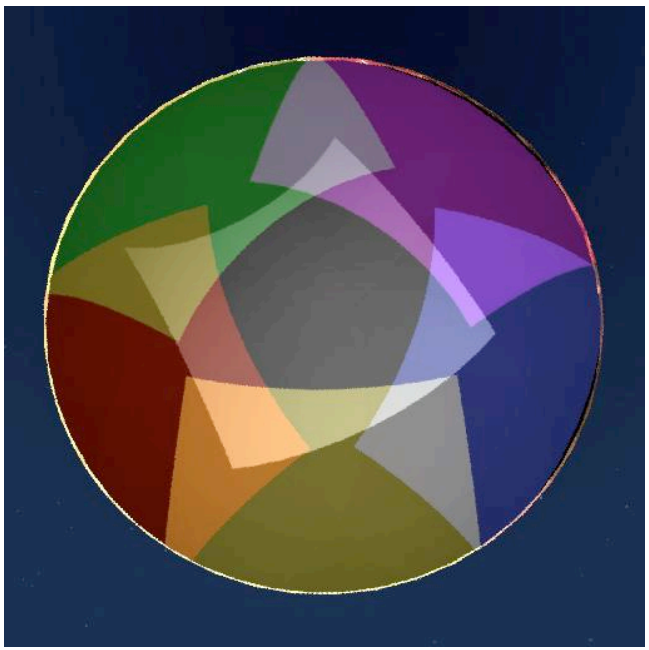




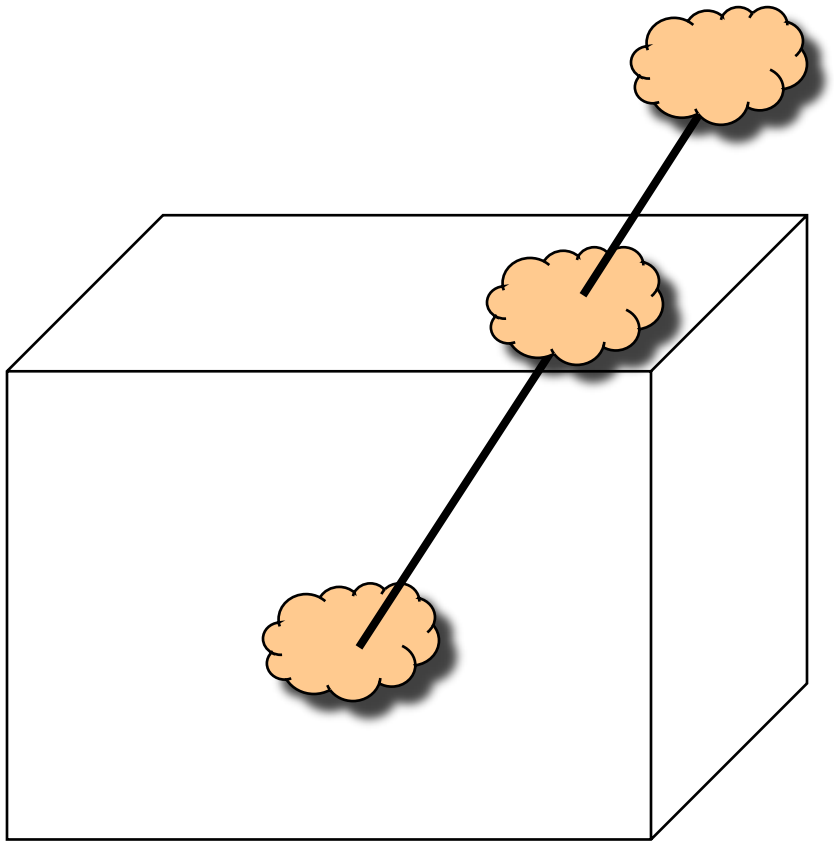
Cube/Slice Map & Final Image

- Dome Master sequence is sliced, feathered, and gamma corrected for 6 projector system





Cube Map



Doing it in OpenGL

- glTextureMap2D(GL_TEXTURE_CUBE_MAP_POSITIVE_X, level, GL_RGBA, rows, columns, border, GL_RGBA, GL_UNSIGNED_BYTE, image1)
- Same for other five images
- Make one texture object out of the six images

OpenGL cube map (contd)

- Parameters apply to all six images
- glTEXParameteri(GL_TEXTURE_CUBE_MAP, GL_TEXTURE_MAP_WRAP_S, GL_REPEAT)
- Same for t and r
- Note that texture coordinates are in 3D space (s, t, r)

OpenGL Cube Map (contd)

- Usually use automatic texture coordinate generation via glTexGen*()
- glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_REFLECTION_MAP);
- glEnable(GL_TEXTURE_GEN, S);
- Same for t and r
- glEnable(GL_TEXTURE_CUBE_MAP);

Normal Mapping

- Similar to texture mapping from cube
- glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_NORMAL_MAP);
- Idea is that we can store normals as textures on cube
- Provides fast normal access
- Works even if textures are stored at low precision (8 bits/component)

Bump Mapping