

ENVIRONMENT MAPPING

OUTLINE

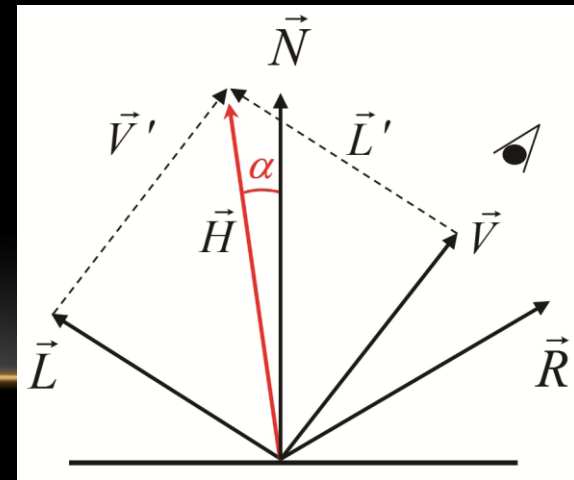
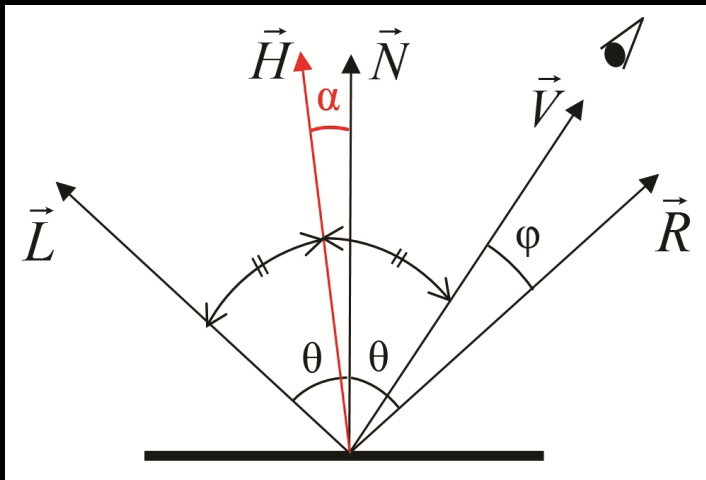
- Why?
- How?
 - Relationship to Blinn-Phong Lighting
- Limitations

WHY?

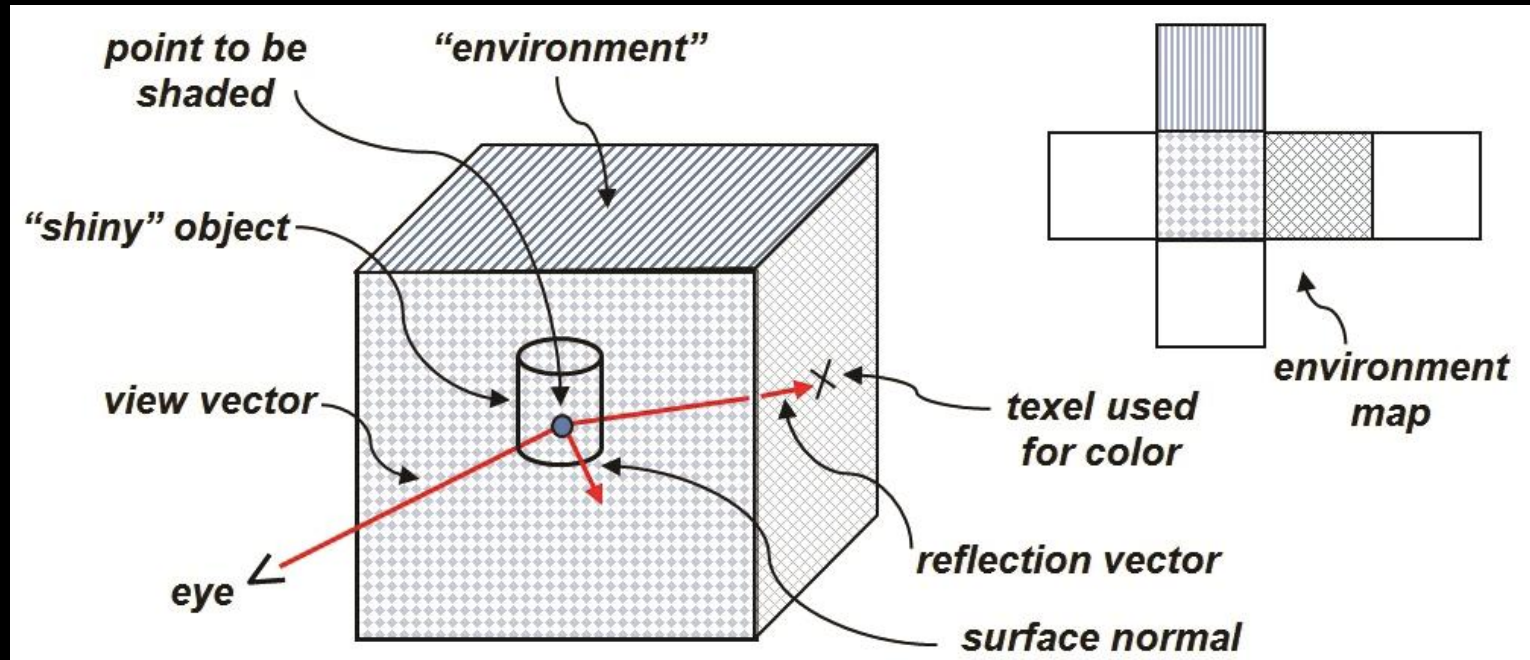
- For very shiny objects
 - Create more than just a specular highlight
 - Reflects the environment around it

HOW?

- Use the same concepts as in Blinn-Phong lighting
 - If your scene already incorporates lighting, much of the work is already done
 - Instead of finding a reflection vector based on light position, find it based on eye position
 - Use this vector to choose what pixel to reflect from the skybox cube map
 - Only need to change some code in the fragment shader

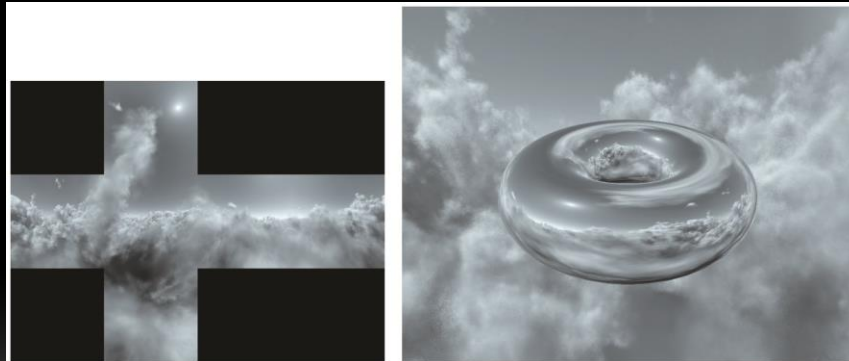


HOW?



ENVIRONMENT MAPPING EXAMPLE

- In `setupVertices()`, calculate normals for the torus
 - If not combining with a texture on the torus, don't need to texture it
- In `display()`:
 - Link normal matrix to a uniform variable and activate it
 - Activate texture cube map as texture for torus
- In vertex shader, treat as if using lighting
- In fragment shader, retrieve output color from cube map texture using the reflection vector to get the texture coordinate



LIMITATIONS

- Object only reflects skybox cube map
 - Other objects in the scene are not reflected
- If it is important to your application, can use the OpenGL stencil buffer

SUMMARY

- Why?
- How?
 - Relationship to Blinn-Phong Lighting
- Limitations

