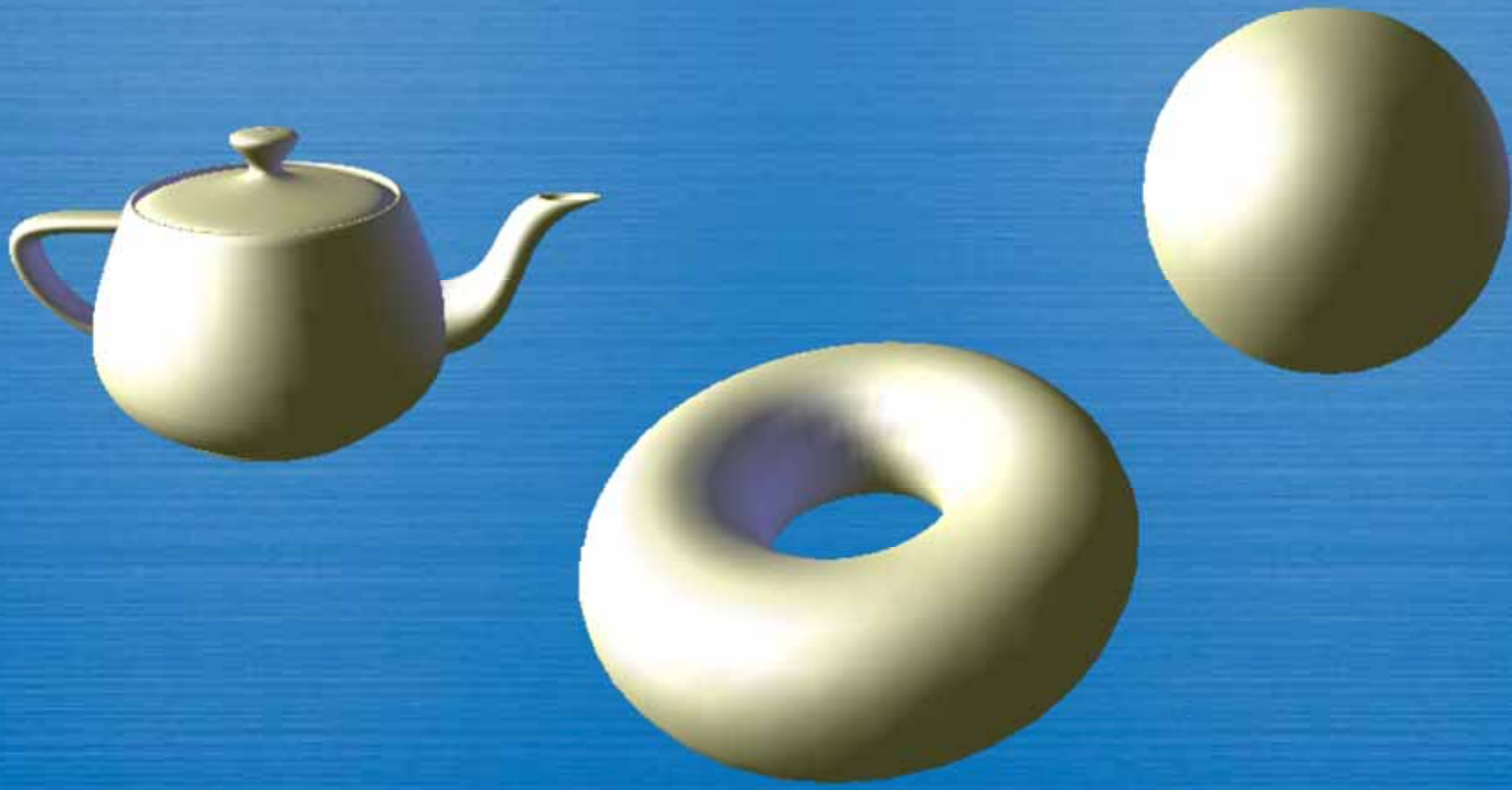
CMSC 491G/691G

Computer Graphics for Games

Marc Olano

OpenGL State Demo: Vertex Lighting



Lighting Vectors in Eye Space

```
void main() {
  // convert shading-related vectors to eye space
  vec4 P = gl_ModelViewMatrix*gl_Vertex;
  vec4 E = gl_ProjectionMatrixInverse*vec4(0,0,-1,0);
  vec3 V = normalize(E.xyz*P.w-P.xyz*E.w);
  vec3 N = normalize(gl_NormalMatrix*gl_Normal);
  …
```

Accumulate Each Light

```
…
// accumulate contribution from each light
gl_FrontColor = vec4(0);
for(int i=0; i<gl_MaxLights; i++) {
  vec3 L = normalize(gl_LightSource[i].position.xyz*P.w
                     - P.xyz*gl_LightSource[i].position.w);
  vec3 H = normalize(L+V);
  float diff = dot(N,L);

  gl_FrontColor += gl_LightSource[i].ambient;
  if (diff > 0.) {
    gl_FrontColor += gl_LightSource[i].diffuse * diff;
    gl_FrontColor += gl_LightSource[i].specular *
       max(pow(dot(N,H), gl_FrontMaterialShininess),0.);
  }
}
…
```

Standard Vertex Shader Stuff

```
  …
  // standard texture coordinate and position stuff
  gl_TexCoord[0] = gl_TextureMatrix[0]*gl_MultiTexCoord0;
  gl_Position = ftransform();
}
```