1 Phong Shading Model Demo

This demo highlights the interaction of white light with two spherical objects with different chromatic and material properties. The sphere that appears in the front has a copper tone with RGB = $(1.0, 0.63, 0.4)$ and the one in the back is in red in color. Variables used in the code given below follow the Phong shading formula:

$$R(\lambda, \vec{x}_p, \vec{V}) = k_a r(\lambda) + k_d r(\lambda)[\vec{N} \cdot \vec{L}]I(\lambda) + k_s S(\lambda)(\vec{M} \cdot \vec{V})^{k_e}$$

Matlab code:**phongDemo.m**

```
% clean up workspace and close all figures
clear all;
close all;

% white light shines on 2 spheres
lightColor = [1 1 1];

%surface material = 'metal'
surfaceType = 'metal';
ka  = 0.1; % ambient  reflection coefficient
kd  = 0.1; % diffuse  reflection coefficient
ks  = 1.0; % specular reflection coefficient
ke  = 5.0; % spectral exponent
scr = 0.5; % reflected light a combination of
           % illuminant and surface color
phongShade(surfaceType, lightColor, ka, kd, ks, ke, scr);

% surfaceType = shiny
surfaceType = 'shiny';
ka  = 0.1;
kd  = 0.6;
ks  = 0.7;
ke  = 5.0;
scr = 1.0; % reflected light pure illuminant color
phongShade(surfaceType, lightColor, ka, kd, ks, ke, scr);

% surfaceType = diffuse
surfaceType = 'diffuse';
ka  = 0.1;
kd  = 0.7;
ks  = 0.0;
ke  = 1.0; % since ks = 0, the exact values for
scr = 1.0; % ke and scr do not matter
phongShade(surfaceType, lightColor, ka, kd, ks, ke, scr);

% surfaceType = ambient, observe complete lack of
% 3D information from the spheres. why is that?
surfaceType = 'ambient';
ka  = 1.0;
kd  = 0.0;
ks  = 0.0;
ke  = 1.0; % since ks = 0, the exact values for
scr = 1.0; % ke and scr do not matter
phongShade(surfaceType, lightColor, ka, kd, ks, ke, scr);
```

Check the following in matlab:

```
help material
t = material('metal')
t = material('shiny')
t = material('dull')
```

QUESTIONS:

- Why is there no inter-reflection?
- How are $r(\lambda)$, $I(\lambda)$ computed?

Matlab code:**phongShade.m**

```
function phongShade(surfaceType, lightColor, ka, kd, ks, ke, scr)
% function phongShade(surfaceType, lightColor, ka, kd, ks, ke, scr)
%   surfaceType: calling program denotes the material used
%   lightColor : illuminant color. 2 lights are defined at infinity.
%          ka  : ambient  reflection coefficient [0, 1]
%          kd  : diffuse  reflection coefficient [0, 1]
%          ks  : specular reflection coefficient [0, 1]
%          ke  : spectral exponent
%          scr : reflected light a combination of illuminant
%                and surface color
%%%  Author: ADJ, Fall 2001


  % copper color map
  copperCM   = copper(64);
  copperRGB  = copperCM(52, :);

  figure;

  % graphics rendering set up
  set(gcf, 'RendererMode', 'manual');
  set(gcf, 'Renderer',    'zbuffer');
  %set(gcf, 'Renderer', 'OpenGL');

  % define a spherical object
  [xSphere, ySphere, zSphere] = sphere(180);

  % object handle for sphere 1
  hSphere1 = surf(xSphere, ySphere, zSphere);
  hold on;
  % object handle for sphere 2
  hSphere2 = surf(xSphere -2, ySphere+2, zSphere);

  % light 1
  hL1 = light('Position', [1 -1 1], 'Color', lightColor);
  % light 2
  hL2 = light('Position', [-3 0 3], 'Color', lightColor);


  set(hSphere1, 'FaceLighting', 'phong',...
```

```
'FaceColor', copperRGB,...
'EdgeColor', 'none',...
'AmbientStrength',  ka, ...
'DiffuseStrength',  kd, ...
'SpecularStrength', ks,...
'SpecularExponent', ke, ...
'SpecularColorReflectance', scr, ...
'BackFaceLighting', 'unlit');

  set(hSphere2, 'FaceLighting', 'phong',...
'FaceColor', 'r',...
'EdgeColor', 'none',...
'AmbientStrength',  ka, ...
'DiffuseStrength',  kd, ...
'SpecularStrength', ks,...
'SpecularExponent', ke, ...
'SpecularColorReflectance', scr, ...
'BackFaceLighting', 'lit');

  axis equal vis3d;
  view([20 25]);
  axis off;

  set(gca,'Fontsize',20);
  title(sprintf('Surfaces: %s \n ka=%1.2f kd=%1.2f ks=%1.2f ke=%1.2f
                scr=%1.2f',surfaceType, ka, kd, ks, ke, scr));
```

2 See the Light

In the figure, **pepper.gif** (from the course webpage):

- Identify image regions that exhibit, predominantly, one or more of the following reflectance properties: ambient, diffuse and specular.
- When does specular reflection become a highlight?
- Can you infer the illuminant direction from the position of the highlights?
- Is the illuminant a point source or a distributed source?
- What is the color of the illuminant?
- Not all highlights seem to have the same intensity. Why is that?
- BRDF's for the peppers: Shiny/Metal/Matte?
- Where do you see interreflections?