Shading Algorithms in MATLAB

MATLAB supports three different algorithms for lighting calculations, selected by setting the **`FaceLighting`** and **`EdgeLighting`** properties of each patch and surface object in the scene. Each algorithm produces somewhat different results:

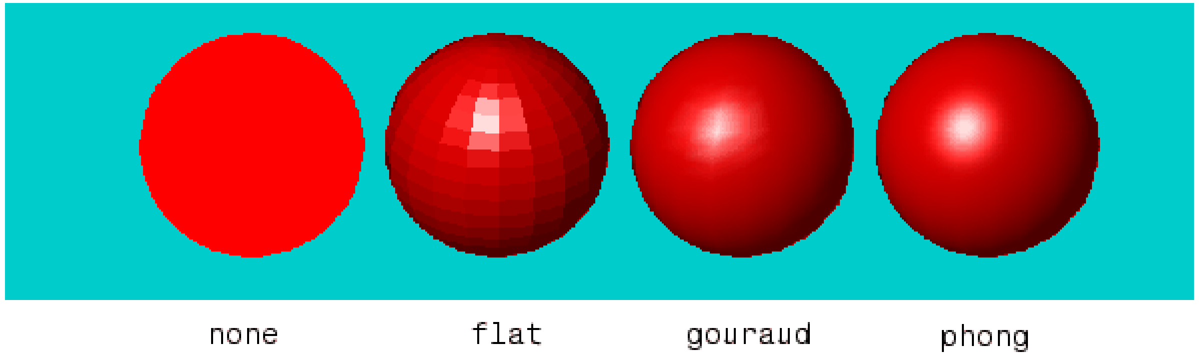
Flat lighting – Produces uniform color across each of the faces of the object. Select this method to view faceted objects.

Gouraud lighting – Calculates the colors at the vertices and then interpolates colors across the faces. Select this method to view curved surfaces.

Phong lighting – Interpolates the vertex normals across each face and calculates the reflectance at each pixel. Select this choice to view curved surfaces.

Shading Example

Phong lighting generally produces better results than Gouraud lighting, but takes longer to render. This illustration shows how a red sphere looks using each of the lighting methods with one white light source. The **`lighting`** command (as opposed to the **`light`** function) provides a convenient way to set the lighting method. (see next slide)



MATLAB **`lighting`** Function

`lighting` selects the algorithm used to calculate the effects of light objects on all surface and patch objects in the current axes.

`lighting flat` selects flat lighting.
`lighting gouraud` selects gouraud lighting.
`lighting phong` selects phong lighting.
`lighting none` turns off lighting

Light Objects

You create a light object using the light function. Three important light object properties are

`Color` – Color of the light cast by the light object

`Style` – Either infinitely far away (the default) or local

`Position` – Direction (for infinite light sources) or the location (for local light sources)

The **`Color`** property determines the color of the directional light from the light source. The color of an object in a scene is determined by the color of the object and the light source. The **`Style`** property determines whether the light source is a point source (Style set to local), which radiates from the specified position in all directions, or a light source placed at infinity (Style set to infinite), which shines from the direction of the specified position with parallel rays. The **`Position`** property specifies the location of the light source in axes data units. In the case of a light source at infinity, Position specifies the direction to the light source. Lights affect surface and patch objects that are in the same axes as the light. These objects have a number of properties that alter the way they look when illuminated by lights.

MATLAB **`light`** Function

`light` creates a light object in the current axes. Lights affect only patch and surface objects.

`light('PropertyName',PropertyValue,...)` creates a light object using the specified values for the named properties. MATLAB parents the light to the current axes unless you specify another axes with the Parent property.

`handle = light(...)` returns the handle of the light object created.

You cannot see a light object per se, but you can see the effects of the light source on patch and surface objects. You can also specify an axes-wide ambient light color that illuminates these objects. However, ambient light is visible only when at least one light object is present and visible in the axes. You can specify properties as property name/property value pairs, structure arrays, and cell arrays.

See also the patch and surface **`AmbientStrength`**, **`DiffuseStrength`**, **`SpecularStrength`**, **`SpecularExponent`**, **`SpecularColorReflectance`**, and **`VertexNormals`** properties. Also see the **`lighting`** and **`material`** commands.

Light **`style`** Property

```
Style                    {infinite} | local
```

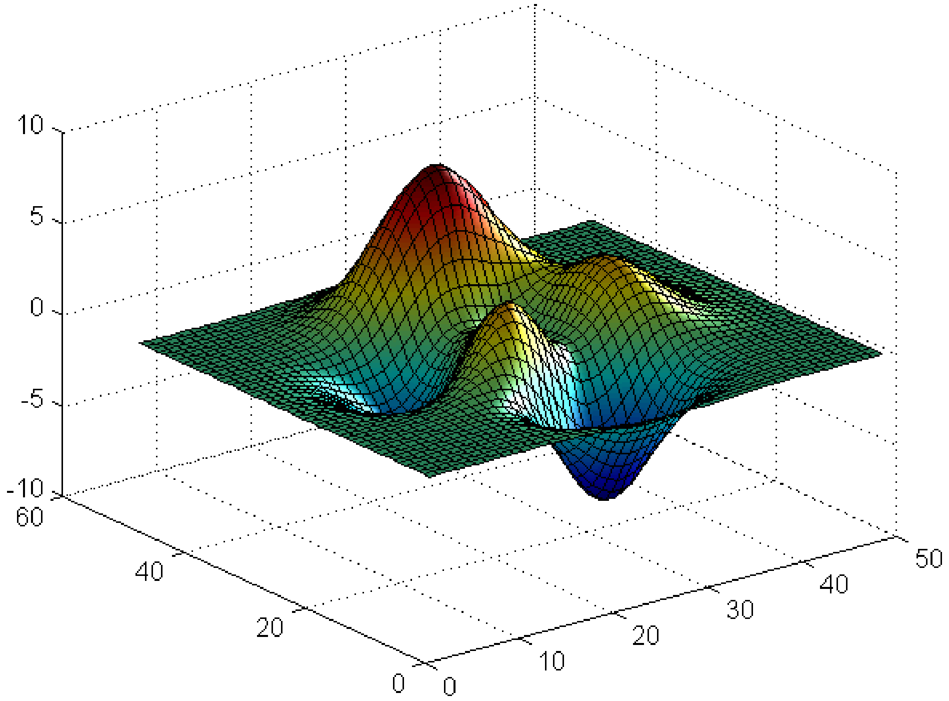
The style property defines a parallel or divergent light source. This property determines whether MATLAB places the light object at infinity, in which case the light rays are parallel, or at the location specified by the **`Position`** property, in which case the light rays diverge in all directions.

If the **`Style`** property is set to **`local`**, **`Position`** specifies the actual location of the light (which is then a point source that radiates from the location in all directions).

If the **`Style`** property is set to **`infinite`**, **`Position`** specifies the direction from which the light shines in parallel rays.

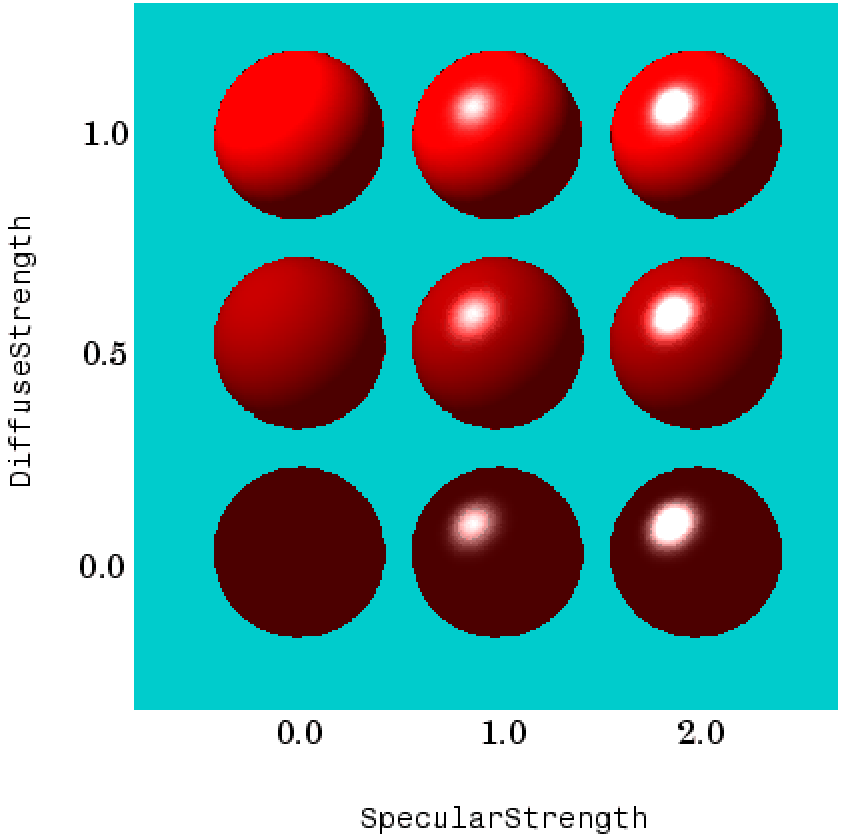
Example 2

```
h = surf(peaks);
set(h,'FaceLighting','phong','FaceColor','interp',...
      'AmbientStrength',0.5)
light('Position',[1 0 0],'Style','infinite');
```



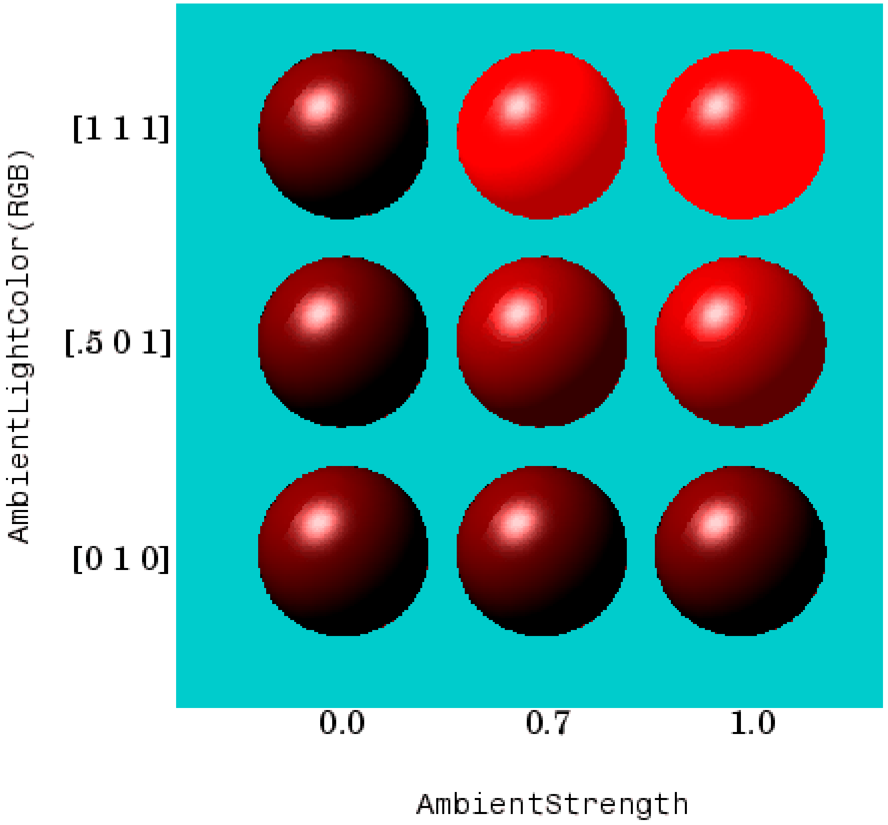
Specular and Diffuse Reflection

You can control the amount of specular and diffuse reflection from the surface of an object by setting the **`SpecularStrength`** and **`DiffuseStrength`** properties. This picture illustrates various settings.



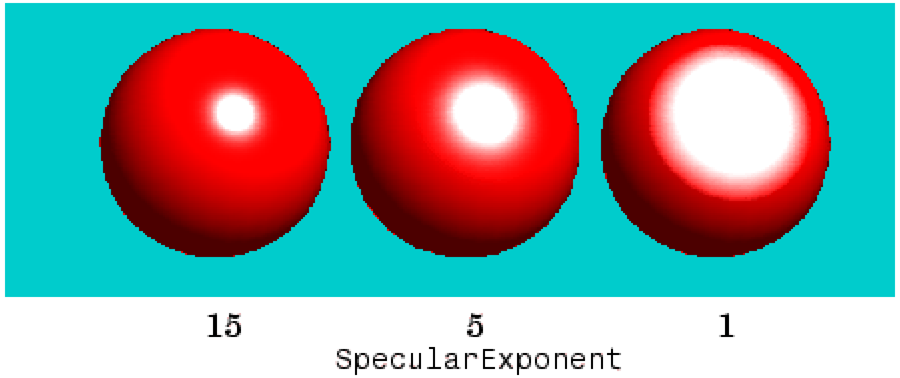
Ambient Light

Ambient light is a directionless light that shines uniformly on all objects in the scene. Ambient light is visible only when there are light objects in the axes. There are two properties that control ambient light -- **`AmbientLightColor`** is an axes property that sets the color, and **`AmbientStrength`** is a property of patch and surface objects that determines the intensity of the ambient light on the particular object. This illustration shows three different ambient light colors at various intensities. The sphere is red and there is a white light object present.



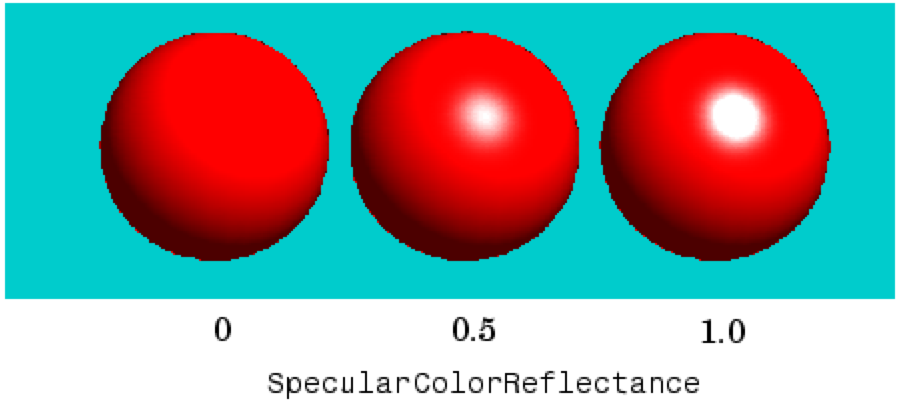
Specular Exponent

The size of the specular highlight spot depends on the value of the patch and surface object's **SpecularExponent** property. Typical values for this property range from 1 to 500, with normal objects having values in the range 5 to 20. This illustration shows a red sphere illuminated by a white light with three different values for the **SpecularExponent** property.



Specular Color Reflectance

The color of the specularly reflected light can range from a combination of the color of the object and the color of the light source to the color of the light source only. The patch and surface **`SpecularColorReflectance`** property controls this color. This illustration shows a red sphere illuminated by a white light. The values of the **`SpecularColorReflectance`** property range from 0 (object and light color) to 1 (light color).


0
0.5
1.0
SpecularColorReflectance