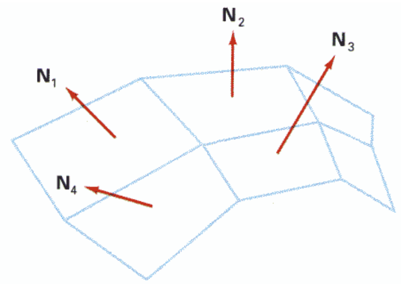
10.1 Types of Shading

In order to produce a realistic image with various kinds of reflection, there are 3 common shading methods which are mainly applied to polygons:

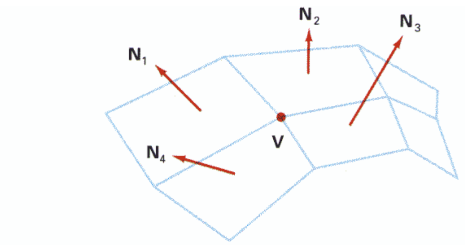
1. Constant Intensity Shading (Flag Shading)

- The intensity value for the whole polygon is calculated once and the whole polygon will be shaded with the same intensity value.
- Fast and simple but can not model specular reflection.
- Problem of intensity discontinuity between polygons (The light intensity is computed based on the normal of each polygon surface. Since different polygons have different normals, the light intensity computed may be quite different).



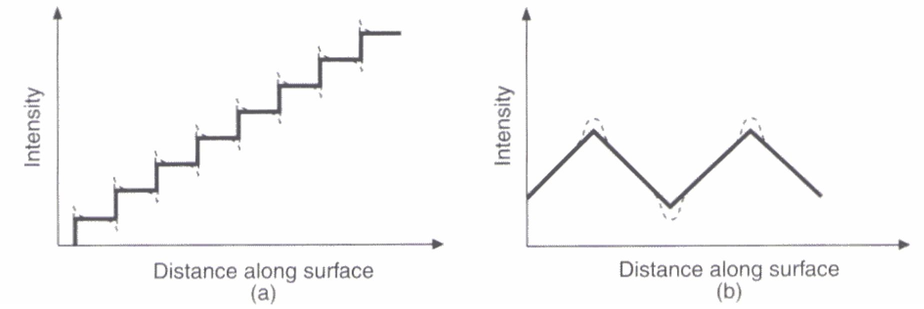
2. Gouraud Shading

- The intensity value is calculated once for each vertex of a polygon.
- The intensity values for the inside of the polygon are obtained by interpolating the vertex values.
- Eliminates the intensity discontinuity problem.
- Still not model the specular reflection correctly.

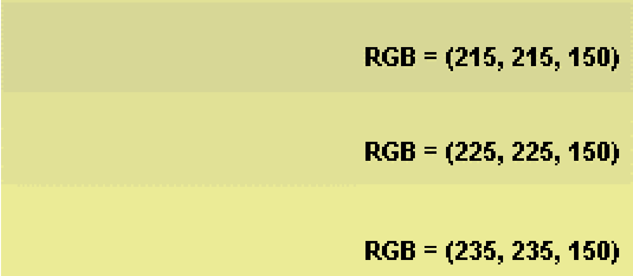


The normal vector at vertex **V** is calculated as the average of the surface normals for each polygon sharing that vertex.

- The interpolation of color values can cause bright or dark intensity streaks, called the **Mach-bands**, to appear on the surface.

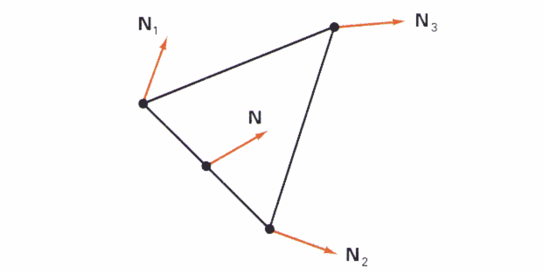


2 examples of Mach band effect.
Dashed lines: perceived intensity; solid lines: actual intensity



3. Phong Shading

- Instead of interpolating the intensity values, the normal vectores are being interpolated between the vertices.
- The intensity value is then calculated at each pixel using the interpolated normal vector.
- This method greatly reduces the Mach-band problem but it requires more computional time.

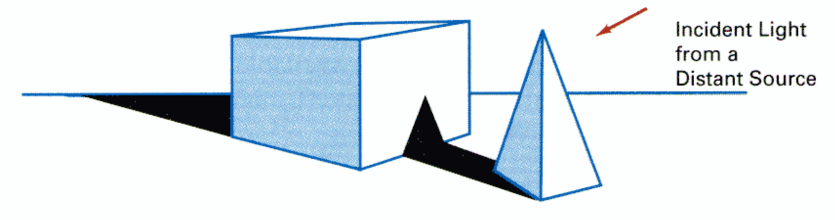


Interpolation of surface normals along a polygon edge.

10.2 Other Effects

Shadow

- Shadow can help to create realism. Without it, a cup, eg., on a table may look as if the cup is floating in the air above the table.
- By applying hidden-surface methods with pretending that the position of a light source is the viewing position, we can find which surface sections cannot be "seen" from the light source => shadow areas.
- We usually display shadow areas with ambient-light intensity only.



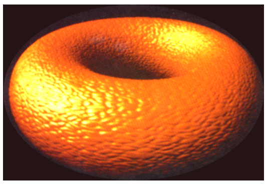
Texture Mapping

Since it is still very difficult for the computer to generate realistic textures, a method called texture mapping is developed in which a photograph of real texture is input into the computer and mapped onto the object surface to create the texture for the object.

- Texture pattern is defined in a MxN array of *texels* or a *texture map* indiced by (u,v) coordinates.
- For each pixel in the display:
  - Map the 4 corners of pixel back to the object surface (for cureved surfaces, these 4 points define a surface patch)
  - Map the surface patch onto the texture map (this mapping computes the source area in the texture map)
  - The pixel values is modified by weighted sum of the texels' color.

Bump Mapping

- To simulate surface roughness within the geometric description of the object, surface roughness can be generated by perturbing surface normals.



Fog Effect

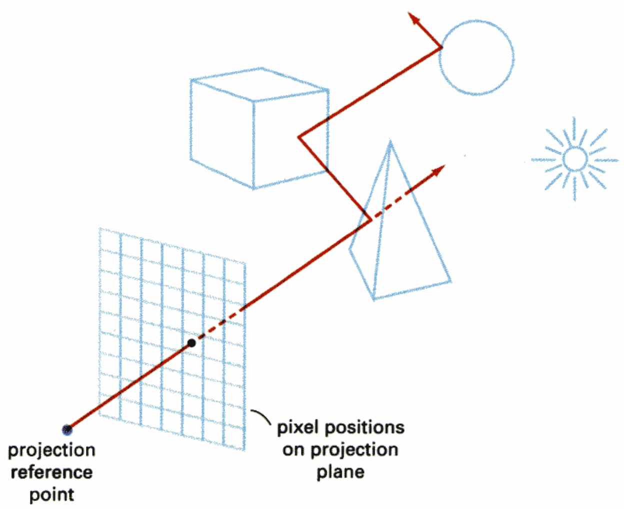
- As an object is further away from the observer, the color of the object fades.
- Fog is a general term that describes similar forms of atmospheric effects. It can be used to simulate haze, mist, smoke, or pollution.

10.3 More Accurate Rendering Methods

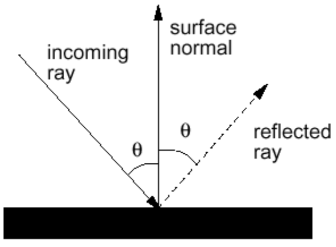
The previously mentioned shading methods have a fundamental limitation that they do not model light reflection / refraction well. The following discussed methods, ray tracing and radiosity, are able to generate realistic light reflection and refraction behavior.

Ray tracing

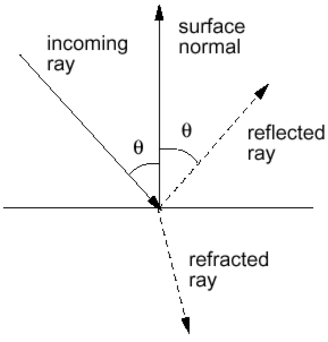
- For each pixel on the image plane, a ray is projected from the center of projection through the pixel into the scene.
- The first object that the ray intersects is determined.
- The point at which the ray hits the object (ie. the ray intersection point) is also determined. The color value is then calculated according to the directions of the light sources to the surface normal.
- However, if there is another object located between a particular light source and the ray intersection point, then the point is in shadow and the light contribution from that particular light source is not considered.


pixel positions on projection plane
projection reference point

- The ray is then reflected and projected from the object until it intersects with another object. (If the surface is a transparent surface, the ray is refracted as well as reflected.)



Normal surface

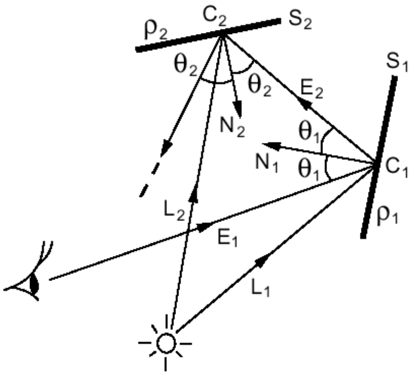


Transparent surface

- The point at which the reflected ray hits the second object are determined and the color value is again calculated in a similar way as the first object.
- The reflected ray is then reflected again from the second object. This process will continue until the color contribution of an intersected object is too small to be considered.
- In practical situation, se would specify the maximum number of reflections that a pixel can make to prevent spending too much processing at a particular pixel.
- All the color values calculated from the intersected objects will be weighted by the attenuation factors (which depend on the surface properties of the objects) and added up to produce a single color value. This color value becomes the pixel value.

Because ray-tracing method calculates the intensity value for each pixel independently, we can consider specular reflection and refraction in the calculation. Hence the method can generate very realistic images. The major problem of this method, however, is that it requires a lot of computations and therefore this method is slow.

To calculate the color of a pixel, consider the following diagram:



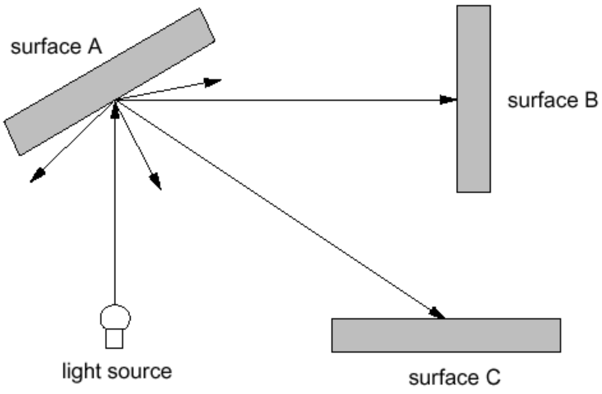
- Assume that surfaces S_1 and S_2 have reflective indices ρ_1 and ρ_2 respectively.
- Given the surface normal vector, N_1, and the light vector, L_1, of surface S_1, we can calculate the color, C_1, of the surface at the point where the eye ray, E_1 intersects the surface.
- Similarly, given the surface normal vector, N_2, and the light vector, L_2, of surface S_2, we can calculate the color, C_2, of the surface at the point where the eye ray, E_2 intersects the surface.
- The color for the pixel is then calculated as: $C_p = \rho_1 C_1 + \rho_1 \rho_2 C_2 + \ldots$

Radiosity

Ray-tracing can model specular reflection very well, but not diffuse reflection.

Why?

> Recall that in diffuse reflection, although light is reflected with equal intensity in all directions, the amount of light energy received by another surface depends on the orientation of the surface relative to the source. Hence surfaces of different orientations may receive different amount of light.
>
> Consider the diagram, assuming that all 3 surfaces are matte. Although A reflects light with equal intensity to B and C, B would receive more light energy than C because B has a smaller angle of incident (ie. higher cosθ).



Radiosity is developed to model this kind of light interaction between surfaces.

- The algorithm is based on the theory of the conservation of energy.

- In a closed environment such as a room, the rate at which energy leaves a surface, called its radiosity, is the sum of the rates at which the surface emits energy and it reflects (or transmits) energy from other surfaces.

- To simplify the calculation, all surfaces in the scene are broken into small patches. Each of which is assumed to be of finite size, emitting and reflecting light uniformly over its entire area.

- The radiosity of a patch i, B_i, can be calculated as follows:

$$B_i = E_i + \rho_i \sum_{j \in all\ \ patches} L_{j \to i}$$

where $L_{j \to i}$ is the amount of light from patch j reaching patch i,

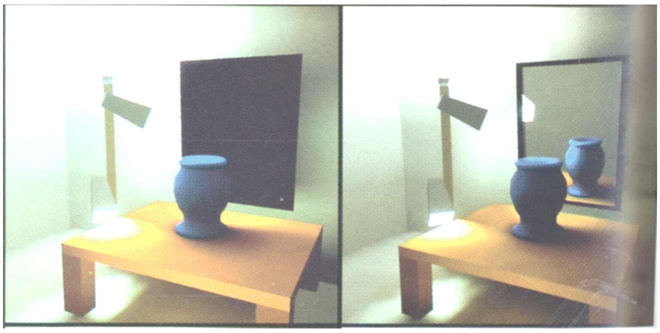
E_i is the light emitted from patch i, and

ρ_i is the reflectivity of patch i. (For an incident light, what ratio of energy is reflected)

- Once we have obtained a radiosity for each patch, we can render the scene with a scan-conversion method using the calculated radiosities as the intensities of the patches.

- Note that the radiosities calculated are view-independent. Hence, the radiosity method although can deal with diffuse reflection well, cannot deal with specular reflection. (Specular reflection is view-dependent).

- To be able to deal with both, we may combine the radiosity method with the ray-tracing method. The prices are the added complexity of the algorithm and the increase in computational time. An example is a 2-pass approach that includes a view-independent radiosity process executed in the first pass, followed by a view-dependent ray-tracing approach in the second pass.



Left: radiosity. Right: Diffuse first pass and ray-traing second pass.