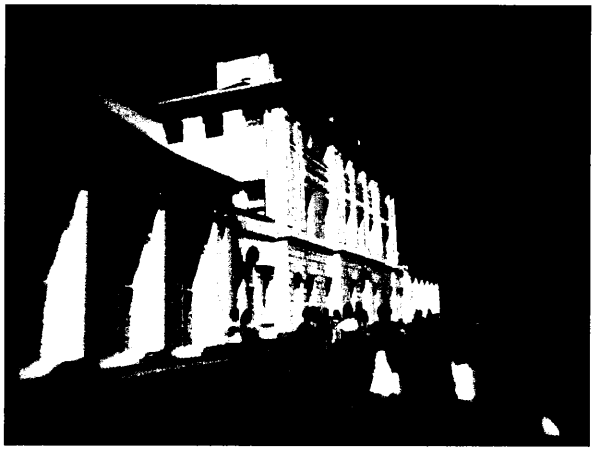
OPTI 415/515 Homework 5

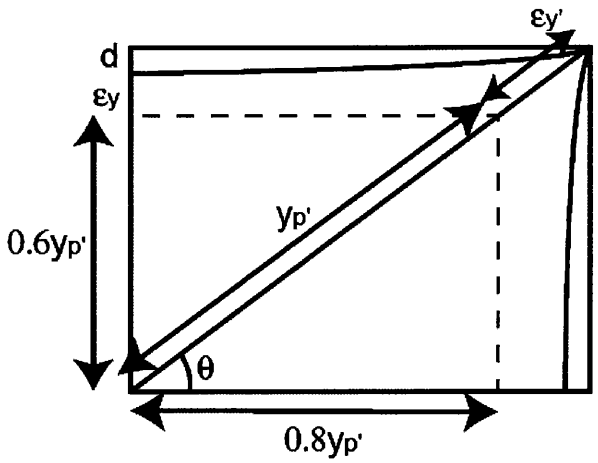
Undergraduates do questions 1-3, and Graduate Students do all four problems.

1. Calculate the Percent Distortion in the image below (original available on line). HINT: In class, we defined %Distortion=$\frac{y' - y'_p}{y'_p}$, where $y' - y'_p = \varepsilon_y$ (i.e. the transverse ray error for distortion) and y'_p = paraxial image height. This calculation must be done along the diagonal of the image.



Consider the upper right quadrant of the image as illustrated in the figure below. The dashed lines are the boundaries of the undistorted image. The distance d is the difference in height between the center top of the picture and the corner of the picture. I measured it in Photoshop and d=15 pixels. The distance y'_p *is unknown, but we know that* $y' = \varepsilon_y + y'_p = 400$ pixels *since the original image is 640x480 making the full diagonal 800 pixels. Furthermore, the image has a 4:3 aspect ratio due to these pixel values. The full field (h = 1) occurs along the diagonal and* ε_y *is the transverse ray error for h = 1. The transverse ray error* ε_y *occurs along the vertical direction for h = 0.6. For the illustration, we see that*

$$\sin\theta = \frac{0.6y'_p}{y'_p} = \frac{d+\varepsilon_y}{\varepsilon_{y'}}.$$



The expressions for transverse ray error for distortion give

$$\varepsilon_y = -\frac{R}{r_{max}} W_{311}(0.6)^3 \text{ and}$$

$$\varepsilon_{y'} = -\frac{R}{r_{max}} W_{311}(1)^3 \Rightarrow \varepsilon_y = 0.216\varepsilon_{y'}$$

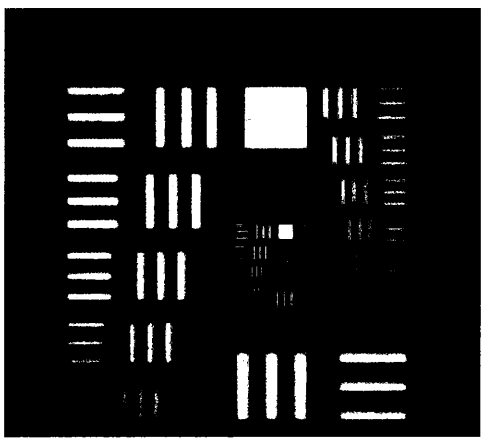
Plugging in to the expressions above leads to

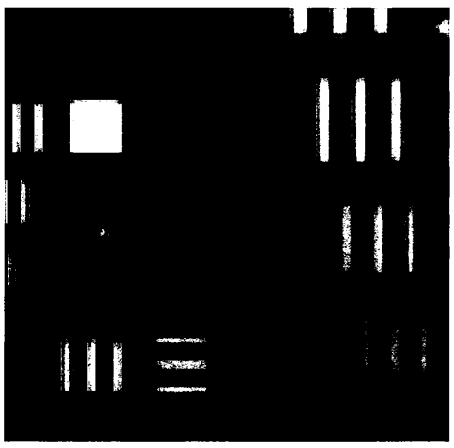
$$\varepsilon_{y'} = \frac{d}{0.6-0.216} = 39.06 \text{ pixels and } y'_p = 360.96 \text{ pixels}.$$

Finally,

$$\%\text{Distortion} = \frac{y' - y'_p}{y'_p} = \frac{39.06}{360.96} = 0.0108 = 10.8\%$$

2. Assume that an imaging system with a magnification m = -0.5 is used to capture an image of a 1951 USAF target (original on-line). What is the resolution limit of the system in cyc/mm?





If we zoom in on the image, the contrast of the bars appears to go to zero for Element 5 in Group 5. This target corresponds to 50.8 cyc/mm in object space. We need to scale by 1/m to convert to image space spatial frequency, so the cutoff frequency of this system is 101.6 cyc/mm.

3. Write a program to remove the distortion from the image in question 1. Show your resulting image.

From problem 1, we saw that

$$\varepsilon_{y'} = 39.06 = -\frac{R}{r_{max}} W_{311}$$

Which means that the transverse ray error anywhere in the field is $\varepsilon_y = 39.06h^3$. *The basic algorithm for removing the distortion is as follows:*

1. Create a new array to hold the undistorted image.

2. For each pixel in the new array calculate the Cartesian coordinates (x,y) and the polar coordinates (r, θ) *relative to the center of the array.*

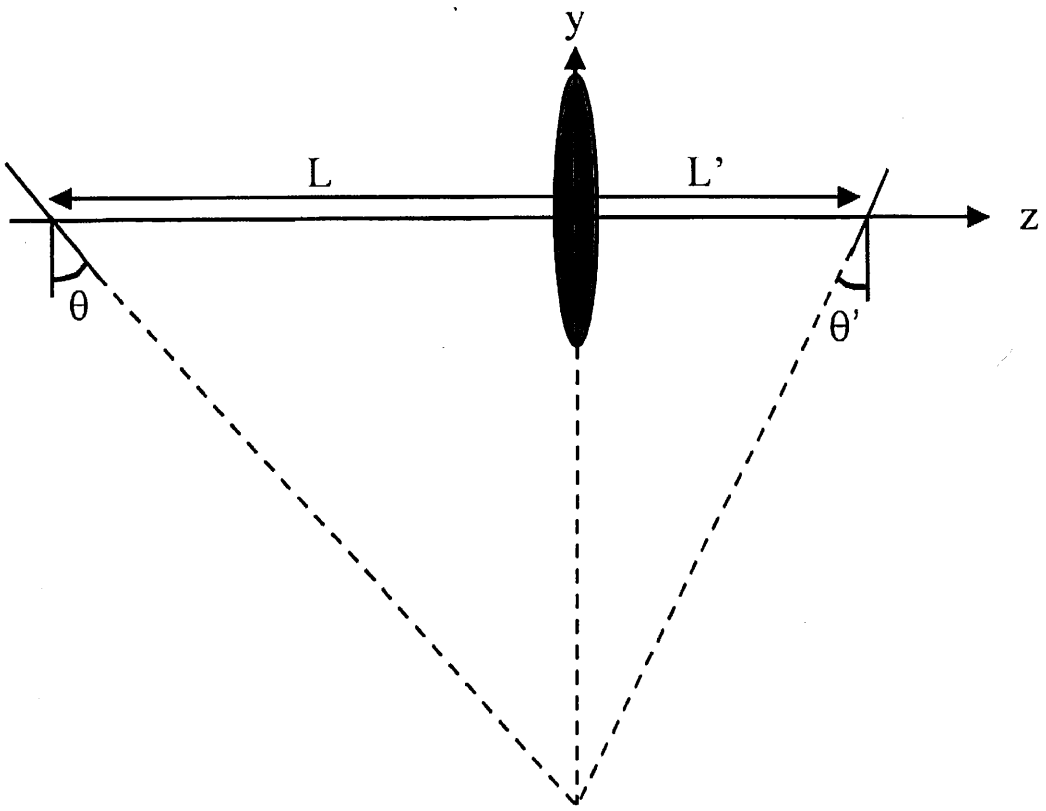
3. Calculate the normalized field coordinate $h = r / y'_p$, *where* $y'_p = 360.96$ pixels *from the first question.*

4. Calculate $\varepsilon_y = 39.06h^3$.

5. If the location $(x+\varepsilon_y \cos\theta, y+ \varepsilon_y \sin\theta)$ *lies within the distorted image, then get the pixel value at that location and set the pixel in the new array to this value.*

Here's my image

Scheimpflug imaging



The imaging formula is given by

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}. \quad (1)$$

In conventional imaging, the object and image planes are parallel to one another with $z = L$ (L is negative in the figure above) and $z' = L'$. If the object plane is tilted by an angle θ, then the Scheimpflug condition says the image plane is tilted as well. The tilted object and image planes become functions of y, so the Lensmaker's formula becomes

$$\frac{1}{z'(y)} - \frac{1}{z(y)} = \frac{1}{f}. \quad (2)$$

From the geometry in the image above, the object plane is described by a plane tilted about the x axis such that

$$z(y) = L - y \tan\theta. \quad (3)$$

where a counterclockwise rotation of the object plane corresponds to a positive value of θ. Plugging this expression (3) into equation (2) and solving for $z'(y)$ leads to

$$z'(y) = \frac{f(L - y\tan\theta)}{f + L - y\tan\theta} \cong \frac{f(L - y\tan\theta)}{f + L} \quad (4)$$

where the assumption that $L >> y \tan\theta$ has been made. Equation (4) also describes a plane tilted about the x axis.

Location of Image Plane

The location of the image plane can be found by evaluating z'(0).

$$z'(0) = \frac{fL}{f + L}$$
$$\frac{f + L}{fL} = \frac{1}{z'(0)} \quad (5)$$
$$\frac{1}{f} + \frac{1}{L} = \frac{1}{z'(0)}$$

Equation (5) is just a statement of the Lensmaker's formula, requiring z'(0) = L'.

Intersection of the Object and Image Planes

The object and image planes intersect when z(y) = z'(y). This intersection occurs when

$$y = \frac{L}{\tan\theta} \quad (6)$$

Plugging (6) back into the expressions for the object and image planes leads to

$$z\left(\frac{L}{\tan\theta}\right) = L - L = 0 \quad \text{and} \quad z'\left(\frac{L}{\tan\theta}\right) = \frac{fL}{L + f} - \frac{fL}{L + f} = 0 \quad (7)$$

In other words, the object and image plane intersect at the plane of the lens.

Image Plane Tilt

Equation (4) can be rewritten as

$$z'(y) = \frac{fL}{f + L} - \frac{f\tan\theta}{f + L}y = L' - y\tan\theta' \quad (8)$$

where

$$\tan\theta' = \frac{f\tan\theta}{L + f}. \quad (9)$$

Magnification

The magnification m_o for the axial object and image points is given by

$$m_o = \frac{L'}{L} = \frac{f}{L+f}. \quad (10)$$

To calculate the magnification m as a function of y for the tilted system, equation (4) without the approximation L >> y tan θ needs to be used.

$$z'(y) = \frac{f(L - y\tan\theta)}{f + L - y\tan\theta} = \frac{f\, z(y)}{f + L - y\tan\theta} \Rightarrow m = \frac{z'(y)}{z(y)} = \frac{f}{(f+L)}\left[\frac{1}{1 - \frac{y\tan\theta}{f+L}}\right] \quad (11)$$

$$m = \frac{m_o}{1 - \frac{\tan\theta}{f+L}y}. \quad (12)$$

Using a binomial expansion on equation (12) leads to

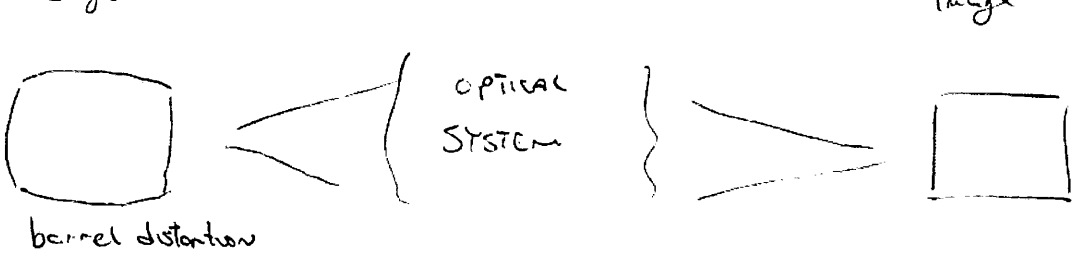
$$m = m_o\left[1 + \frac{\tan\theta}{f+L}y + \ldots\right]. \quad (13)$$

In other words, the magnification is linear in y or there is keystone distortion in the system (at where the truncated binomial expansion closely approximates equation (12)).

Step 1 KEYSTONE AND SCHEIMPFLUG SLIDES

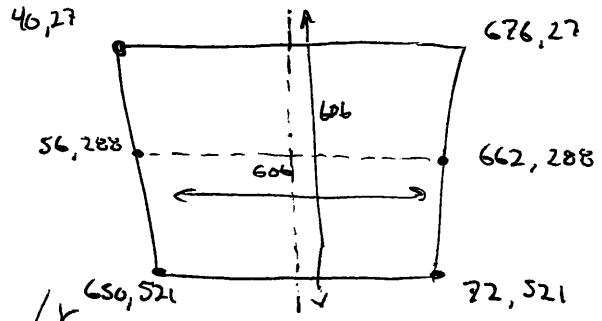
Measurement of Distortion

Place a target with known levels of distortions and examine image



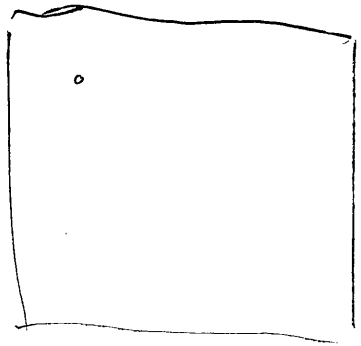
	27			-15
40		56		
	27			-15
676		662		
650	521	662		591
			591	
72	521	56		

662	288	303
56	303	288
606	591	-15



Do

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -X_x & -X_y \\ 0 & 0 & 0 & x & y & 1 & -Y_x & -Y_y \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix}$$



view. For this reason, you can use the UP and RIGHT vectors for aspect ratio correction if required and use the magnitude of the DISTANCE vector to control field of view.

You can also produce other effects by manipulating the UP and RIGHT vectors. To turn an image completely upside down, the UP vector, which is usually specified as <0 1 0>, is changed to <0 −1 0>. Along the same vein, image left and right can be reversed by changing the RIGHT vector, usually <1 0 0> (or <1.333 0 0>) to <−1 0 0> (or <−1.333 0 0>).

Much of the time you spend designing ray-traced imagery will be spent fine tuning the viewing geometry to get the exact view you desire. You will have to go through much experimentation before you will feel completely comfortable with the interaction of the viewing parameters. After a time, however, it will become intuitive. We will take up viewing geometry again in the next chapter and in Part Two of this book.

Parametric Equations of Rays

As the name ray tracing suggests, the ray is the basis for this technique. A ray is an extension of the vector discussed earlier. A ray has both a direction and an origin. As you will recall, a positional vector had an implied origin at 0,0,0 of our three-dimensional coordinate system. A ray has a stated origin not necessarily at the coordinate system origin. A ray is made up of two vectors: one describing its direction and one describing its origin. Given a starting point (the ray's origin) and the ray's direction, it is possible to calculate the trajectory of a ray given only how long the ray has been moving in the specified direction. If we indicate time as "t," the trajectory of a ray becomes:

Trajectory of Ray R = direction * t + origin

which more mathematically is described as:

$$R(t) = R_d * t + R_o \text{ for } t > 0$$

with R(t) describing a set of points that make up the ray's trajectory. Of course, this equation must be stated in three dimensions to be of interest to us. Thus, the equation of a point through which a ray passes in three dimensions as a function of t becomes:

$$P_X = X_d * t + X_o$$
$$P_Y = Y_d * t + Y_o$$
$$P_Z = Z_d * t + Z_o$$

where the ray's direction is described by the vector $\langle X_d, Y_d, Z_d \rangle$ and its origin by the vector $\langle X_o, Y_o, Z_o \rangle$. We will use this explicit form of the parametric ray equation (parametric because it is a function of parameter t) throughout this text. *Note*: You should use unit vectors for the direction of a ray during all calculations.

As you might have surmised, ray tracing involves a lot of checks to determine if rays and objects intersect. Using the above parametric equation of a ray it is relatively simple to determine if an intersection occurs between simple shapes (shapes defined by quadratic equations, for example) and a ray. In the ray tracer presented in the next chapter, intersection checks between rays and spheres and rays and planes will be required. For that reason, we present those intersection calculations below. Following that, we present a general solution for ray/quadric surface intersection, even though it is not utilized in the example ray tracer. Any surface whose equation is a quadratic in variables X, Y, and Z is called a quadric surface. For other ray/shapes intersection calculations, see the book *An Introduction to Ray Tracing*. Information about this book appears in the "Further Reading" section of Part Three.

Ray/Sphere Intersection Calculations

To determine if a ray intersects a sphere, the parametric ray equation is substituted into the equation of a sphere and then solved. This results in a second order quadratic equation that can easily be solved. To see how this is done, consider an equation for a sphere S as:

$$(X_s - X_c)^2 + (Y_s - Y_c)^2 + (Z_s - Z_c)^2 = \text{Radius}^2$$

which describes a sphere as a collection of surface points X_s, Y_s, Z_s centered at X_c, Y_c, Z_c with radius of Radius. What we need to determine is if our ray intersects this sphere at its surface. To do this, let us substitute the explicit form of the ray equation into the sphere equation as follows:

$$(X_d * t + X_o - X_c)^2 + (Y_d * t + Y_o - Y_c)^2 + (Z_d * t + Z_o - Z_c)^2 = \text{Radius}^2$$

A half page of algebra can easily prove that the above equation reduces to:

$$A * t^2 + B * t + C = 0$$

where:

A= $X_d^2 + Y_d^2 + Z_d^2$ which equals one, because the ray's direction vector was normalized (i.e., it was a unit vector).

B= $2 * (X_d * (X_o - X_c) + Y_d * (Y_o - Y_c) + Z_d * (Z_o - Z_c))$

C= $(X_o - X_c)^2 + (Y_o - Y_c)^2 + (Z_o - Z_c)^2 - \text{Radius}^2$

You probably recognize the above equation as a quadratic equation with roots that can be determined with the formula:

$$\frac{-B \pm \sqrt{B^2 - 4 * A * C}}{2 * A}$$

Given the fact that A equals one, we can simplify and solve the above equation as follows:

$$t_1 = \frac{-B + \sqrt{B^2 - 4 * C}}{2}$$

$$t_2 = \frac{-B - \sqrt{B^2 - 4 * C}}{2}$$

where the quantity:

$$B^2 - 4 * C$$

is referred to as the determinant. If the determinant is negative, denoting imaginary roots, the ray is presumed to miss the sphere completely. If real roots (positive roots) are returned, the

PRACTICAL RAY TRACING IN C - CRAIG A. LINDLEY
WILEY, NEW YORK 1992

smallest positive root determines the closest intersection between the ray and the sphere—the intersection closest to the eye's LOCATION. This is the intersection we are interested in. If you think about it, only three possible ray/sphere scenarios exist:

1. The ray misses the sphere completely.
2. The ray intersects the sphere tangentially which results in only a single root.
3. The ray passes through the sphere. That is, it enters and then exits. This results in two positive roots, one smaller than the other.

Remember, the solution to this equation is the time parameter t in the ray equation. When t is known, it can be substituted back into the ray equation to find the coordinates of point P, the point of intersection between the ray and the sphere. Once you have established the point of intersection, another important item you must determine is the surface normal at the point of intersection. A normal is a vector that generally points away from the surface in such a manner as to be perpendicular to some point on the surface. A surface normal is used to help determine the optical properties at the point of intersection. How a surface normal is used as part of the ray-tracing process will be discussed later. For now, we will concentrate on how the surface normal for a sphere is calculated.

Consider the ray/sphere intersection shown in Figure 2.4. It shows a ray striking a sphere at point P. To calculate the normal N at P we simply subtract the coordinates of point P on the surface from the coordinates of the sphere's center. With this done, we have created a new ray with its origin at the center of the sphere and extending to the sphere's surface at point P. Since only the direction of this normal is what we are interested in, we will convert the direction portion of this ray to a unit vector. We can do this by dividing each component of this vector by its magnitude. We already know the magnitude of the vector because we have the radius of the sphere. Thus, the normal calculations are as follows:

$$X_N = \frac{X_P - X_C}{\text{Radius}}$$

$$Y_N = \frac{Y_P - Y_C}{\text{Radius}}$$

$$Z_N = \frac{Z_P - Z_C}{\text{Radius}}$$

You may be wondering why our normal vector does not originate at point P and point perpendicularly out into space instead of originating at the sphere's center and just reaching the sphere's surface. While it is helpful to visualize a normal that points from the surface outward, the subsequent calculation we will perform depends only on the direction of the normal, not its origin. For this reason it is unnecessary to translate the normal's origin from the center of the sphere to the surface at point P. It is important, however, that the normal points the way it does—out of the sphere. This direction is assumed during all further calculations.

As we have seen, only the specifications of the ray (its origin and direction) and the specifications of the sphere (its location and radius) are required to determine ray/sphere intersection. Given that an intersection exists, the surface normal at the point of intersection is easily calculated.

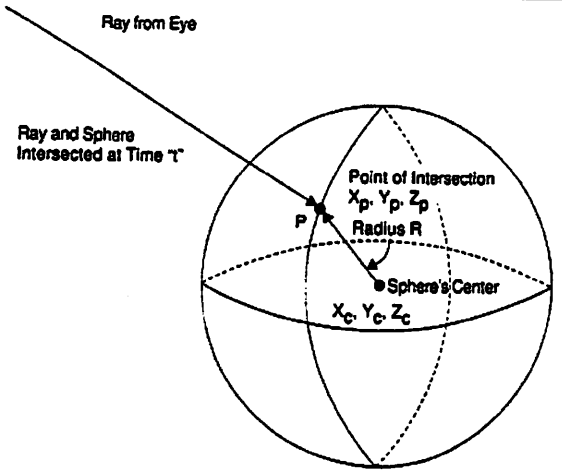


Figure 2.4 Ray/Sphere Intersection

Ray/Plane Intersection Calculations

The next primitive quadric shape we shall consider is the plane. A plane is a geometric entity that divides the space through which it passes in two. A plane, unlike a sphere, continues on infinitely; that is, it is unbounded. A sphere in contrast is bounded. The method used to calculate ray/plane intersection is very similar to that used for ray/sphere intersection. We begin with the equation of a plane, which is defined as:

$$AX + BY + CZ + D = 0$$

In total, four numbers define a plane: A, B, C, and D. The first three of these numbers, A, B, and C, define a unit vector normal to the plane. By definition, therefore, $A^2 + B^2 + C^2$ must equal one. *Note*: A vector that is normal to a plane at one point is normal to the plane at all points. The factor D in the above equation defines the distance of the plane from the coordinate system origin at 0,0,0. To calculate the intersection we again substitute the explicit form of the ray parametric equation into the plane equation and solve again for t as follows:

$$A*(X_d*t + X_o) + B*(Y_d*t + Y_o) + C*(Z_d*t + Z_o) + D = 0$$

then:

$$t = \frac{-(A*X_o + B*Y_o + C*Z_o + D)}{A*X_d + B*Y_d + C*Z_d}$$

If the denominator of this equation is equal to zero, the ray and the plane are parallel or lie in the same plane, and no intersection occurs; calculations can stop at this point. If the denominator is nonzero, calculations must continue. If the denominator is greater than zero,

the normal of the plane is pointing in the same direction as the ray and may have to be reversed for later use.

The solution to this equation, the parameter t, is calculated in a straightforward manner as it is a linear equation with all variables known. A, B, C, and D define the plane, whereas X_d, Y_d, Z_d, X_o, Y_o, and Z_o are defined by the ray. If t is less than zero, no intersection occurs and calculations can again be halted. If t is a positive number, an intersection has been found. The point of intersection is then calculated by substituting the value of t back into the ray equation and solving for P. As mentioned above, the surface normal for the plane may not be pointing in the appropriate direction. Usually we want the normal to point back toward the origin of the ray. The backwards normal condition is indicated when the denominator of the equation above evaluates to a number greater than zero. To reverse the direction of the normal it must be negated. Vector negation was discussed previously.

General Ray/Quadric Surface Intersection Calculations

Even though both the sphere and the plane are examples of quadric surfaces, the solutions given above did not treat them in a general manner as such. Instead, each solution was unique and somewhat optimized for the shape involved; not general purpose at all. In this section, we present a generalized solution to the complete family of quadric surfaces. Using this approach it is possible to ray trace any of the quadric surfaces, including the sphere, the plane, the ellipsoid, the cylinder, the cone, the hyperboloid, and the paraboloid. Due to the generality of this approach, you can handle sphere and plane intersections using the quadric solution presented, although the previously presented solutions would probably execute faster because they are less complex.

The generalized quadratic formula describing the quadric family of surfaces is given as:

$$A*X^2 + B*Y^2 + C*Z^2 + D*X*Y + E*X*Z + F*Y*Z + G*X + H*Y + I*Z + J = 0$$

This formula implies that all possible quadric surfaces at all possible locations in space and all possible orientations can be defined with just ten numbers (defining parameters), A through J. As you shall see shortly, this is a very powerful concept.

To proceed, we again substitute the parametric equation of a ray into the generalized quadratic equation above. After a couple of pages of algebra, we can reduce the equation to the form:

$$A_c*t^2 + B_c*t + C_c = 0$$

where:

$$A_c = A*X_d^2 + B*Y_d^2 + C*Z_d^2 + D*X_d*Y_d + E*X_d*Z_d + F*Y_d*Z_d$$

$$B_c = 2*A*X_o*X_d + 2*B*Y_o*Y_d + 2*C*Z_o*Z_d + D*X_o*Y_d + D*Y_o*X_d + E*X_o*Z_d + E*Z_o*X_d + F*Y_o*Z_d + F*Z_o*Y_d + G*X_d + H*Y_d + I*Z_d$$

$$C_c = A*X_o^2 + B*Y_o^2 + C*Z_o^2 + D*X_o*Y_o + E*X_o*Z_o + F*Y_o*Z_o + G*X_o + H*Y_o + I*Z_o + J$$

We then use the determinant $B_c^2 - 4*A_c*C_c$ to determine if an intersection exists. Simply, if this value is less than zero, no intersection exists between the specified ray and the quadric surface and the calculations can be terminated at this point. If a solution does exist, it is found by solving the quadratic formula for the two possible roots t_1 and t_2.

$$t_1 = \frac{-B_c - \sqrt{B_c^2 - 4*A_c*C_c}}{2*A_c}$$

$$t_2 = \frac{-B_c + \sqrt{B_c^2 - 4*A_c*C_c}}{2*A_c}$$

The second root, t_2, needs to be solved only if t_1 is less than zero. The root we are interested in is the smallest positive one.

The calculation of the normal for the generalized quadric surface requires a bit of calculus to prove. Since this is outside the scope of this discussion we will not describe it here. Suffice it to say that the normal is the partial derivative of the generalized quadratic formula given at the start of this section with respect to X, Y, and Z. In other words:

$$N_X = 2*A*X + D*Y + E*Z + G$$
$$N_Y = 2*B*Y + D*X + F*Z + H$$
$$N_Z = 2*C*Z + E*X + F*Y + I$$

To calculate the normal, we must know the coordinates of the point of intersection between the ray and the quadric surface. Since we now have the solution for the parameter t (either t_1 or t_2), we can substitute it back into the parametric ray equation and solve as follows:

$$P_X = X_d*t + X_o$$
$$P_Y = Y_d*t + Y_o$$
$$P_Z = Z_d*t + Z_o$$

By substituting the values of P_X, P_Y, and P_Z for X, Y, and Z in the normal equation above and providing the values of A, B, C, D, E, F, G, H, and I from the equation of the quadric surface being used, we can calculate the surface normal. *Please note*: The calculated surface normal will not be normalized. It must be converted to a unit vector before it is used in any subsequent calculations. Also, as was cautioned above, the calculated surface normal may be pointing in the direction of the ray instead of back toward the origin of the ray as is usually required. If the dot product of the unit normal and the ray's direction vector is greater than zero, this is the case. Under these conditions, the normal should be negated to point in the correct direction just as before.

Quadric Shape Definitions

Ten numbers, A through J, are used to define all of the quadric surfaces. Even fewer numbers are required if we make a few simplifying assumptions about the quadric shapes. If we assume, for example, that all surfaces are created at the origin (0,0,0) and that all are of size one (sphere's radius is one, cylinder's radius is one, a cone's a, b, and c axes' lengths are one, etc.), the quadric surfaces can be defined as shown in Figure 2.5.

Surface	Implemented Equation	Parameters A	B	C	D	E	F	G	H	I	J
Sphere	$X^2 + Y^2 + Z^2 - 1 = 0$	1	1	1	0	0	0	0	0	0	-1
Cylinder along											
X	$Y^2 + Z^2 - 1 = 0$	0	1	1	0	0	0	0	0	0	-1
Y	$X^2 + Z^2 - 1 = 0$	1	0	1	0	0	0	0	0	0	-1
Z	$X^2 + Y^2 - 1 = 0$	1	1	0	0	0	0	0	0	0	-1
Cone along											
X	$-X^2 + Y^2 + Z^2 = 0$	-1	1	1	0	0	0	0	0	0	0
Y	$X^2 - Y^2 + Z^2 = 0$	1	-1	1	0	0	0	0	0	0	0
Z	$X^2 + Y^2 - Z^2 = 0$	1	1	-1	0	0	0	0	0	0	0
Plane in											
YZ axis	$X = 0$	0	0	0	0	0	0	1	0	0	0
XZ axis	$Y = 0$	0	0	0	0	0	0	0	1	0	0
XY axis	$Z = 0$	0	0	0	0	0	0	0	0	1	0
Paraboloid along											
X	$Y^2 + Z^2 - X = 0$	0	1	1	0	0	0	-1	0	0	0
Y	$X^2 + Z^2 - Y = 0$	1	0	1	0	0	0	0	-1	0	0
Z	$X^2 + Y^2 - Z = 0$	1	1	0	0	0	0	0	0	-1	0
(one sheet) Hyperboloid along											
X	$-X^2 + Y^2 + Z^2 - 1 = 0$	-1	1	1	0	0	0	0	0	0	-1
Y	$X^2 - Y^2 + Z^2 - 1 = 0$	1	-1	1	0	0	0	0	0	0	-1
Z	$X^2 + Y^2 - Z^2 - 1 = 0$	1	1	-1	0	0	0	0	0	0	-1

Figure 2.5 Quadric Surfaces—Equations and Parameters

Sometimes it is desirable to define a quadric surface (an ellipsoid or hyperboloid of two sheets, for example) with specific location and specific geometrical properties instead of defining it with the assumptions listed above. The process of determining the ten defining parameters, A through J, is a bit more complex, but it is still possible to do with the procedure outlined below. The following steps illustrate how this is done.

1. From a math reference book, get the defining equation of the quadric shape you desire to model. For example, the equation for an ellipsoid centered at the origin is:

$$\frac{X^2}{a^2} + \frac{Y^2}{b^2} + \frac{Z^2}{c^2} = 1$$

This equation was taken from *Calculus and Analytic Geometry* by George B. Thomas, Jr. The letters a, b, and c represent the axes of the ellipsoid in the X, Y, and Z directions, respectively. To displace the ellipsoid from the origin, the defining equation is modified as follows:

$$\frac{(X-X_c)^2}{a^2} + \frac{(Y-Y_c)^2}{b^2} + \frac{(Z-Z_c)^2}{c^2} = 1$$

2. Substitute into the above equation your specific values for X_c, Y_c, and Z_c, the location of the ellipsoid's center, and values for the axes a, b, and c.
3. Simplify the resulting equation as far as possible. For the ellipsoid example, the simplified result will look something like:

$$4*X^2 + Y^2 + 9*Z^2 - 48*X - 18*Y + 36*Z - 315 = 0$$

4. Match the coefficients in the equation directly above to the defining parameters in the generalized quadratic equation listed previously. This pairing results in A = 4, B = 1, C = 9, D = 0, E = 0, F = 0, G = -48, H = -18, I = 36, and J = -315 for the ellipsoid example.
5. Use these defining parameters to define your ellipsoid quadric shape. By the way, these ten numbers define an ellipsoid positioned at 6, 9, -2 with axes of 12, 24, and 8, respectively.

Other quadric shapes can be defined just as easily.

Shading

In the previous sections of this chapter, we have discussed view geometry (which allows us to control what is within the field of view of the eye), eye ray generation (which uses the view geometry to construct rays originating at the eye's position that pierce the view plane and extend into the three-dimensional object space), and ray/object intersections (which allow us to determine what, if anything, our eye rays intersect). The final missing piece of the ray-tracing puzzle is "shading." Shading is the process by which color is assigned to the objects the eye ray intersects and how that color information is interpreted for display on a computer monitor.

A bit of warning may be necessary here. Shading concepts are the most complex part of ray tracing. The shading model used to describe the interaction of light with surfaces of various materials and attributes can be very complex indeed. This is especially true if the shading model takes into consideration the frequency of the light and the angles of intersection. The complexity of shading points out how hard it is to simulate (model) nature within a computer. Light/surface intersection has many subtle properties that must be simulated accurately for an image to appear real. The human eye/brain system uses these subtle cues to make determinations about what it is looking at. When the cues are missing, the brain can sometimes fill in but a loss of realism is perceived. Luckily, some very clever means of simulating light surface interactions have been developed that are much less costly in computer terms to implement than a full simulation. Some of these techniques are based on geometric optics and physics while others were arrived at empirically. The Phong specular reflection model is an example of a shading technique arrived at empirically. We will discuss Phong shading shortly.

Please keep in mind what shading is and what it is not. The purpose of shading is to calculate the color and intensity of light leaving a surface and traveling back to the eye for subse-

quent viewing. The color of the light perceived by the eye is a combination of the object's surface color, the color of any light reflected by the object, and the color of any light transmitted through the object. Shading is not hidden surface removal. The fact that an eye ray always intersects the object closest to it means that shading will be applied only to surfaces guaranteed to be in the foreground and visible to the viewer. Shading is never applied to obscured objects or obscured portions of objects—obscured, of course, from the viewer's perspective.

As if it isn't bad enough that calculating shading for a scene is difficult and imprecise, after the calculations are done, we find that current PC computer graphics systems don't have the full gamut of colors necessary to display what was calculated. We must make trade-offs in the shading process and even more in the display process. We can partially remedy the limited color availability during display with color quantization, which is discussed in depth in Chapter 4. (See Chapter 4 for all further discussions of the display side of the shading problem.)

To make ray-tracing programs independent of display hardware, colors are usually specified in RGB (Red, Green, and Blue) coordinates with each color component expressed as a floating-point number in the range zero to one. This allows a near infinite number of color possibilities. A color component value of zero indicates darkness (no intensity), whereas a value of one indicates maximum intensity (independent of the actual values used by a graphics display system for minimum and maximum intensity levels). The RGB color system is an additive system such that the absence of color, as indicated by R=G=B=0, indicates black, and colors are built by adding color component values. For example, R=1 G=B=0 specifies pure red of maximum intensity. All colors that have equal color component values represent a shade of gray. Maximum and equal color component values of R=G=B=1 represent white. An RGB color value of I*R,I*G,I*B when I is between zero and one is considered a shade of the color RGB of intensity I. Therefore, shades of gray can also be considered shades of the color white. The additive nature of the RGB color system is utilized throughout our description of the shading process.

As we discuss the shading process, keep in mind that as we trace light rays within a scene to arrive at a representative pixel color and intensity for display, both the direction of the rays and the surface properties must be taken into consideration. The meaning and significance of this statement will hopefully become clear as the discussion proceeds.

When light rays (photons of light, actually) interact with a surface of an object, shading tells us how much light is passed or "propagated" from the surface back toward the viewer. Propagation can be broken down into two major components: specular propagation and diffuse propagation. Specularly propagated light is directionally oriented, whereas diffusely propagated light is light that goes equally in all directions with no relationship to the incident light's direction. Both specular and diffuse propagation must be thought about in terms of reflection and transmission. Reflection is what happens when you look in the mirror or admire your face reflected in a newly waxed automobile. Transmission is what happens when you look through a transparent or semi-transparent surface. The light that arrives at your eye is transmitted through the surface of the object. Looking through a glass window or staring at a fish in a clear stream are examples of transmission of light. In total, therefore, there are four light propagation possibilities to consider in a complete shading model. These are sometimes referred to in the graphics literature as the four light transport modes.

To complicate matters further, all four of these transport mechanisms should be considered both for light coming directly from a light source and for light coming from other objects

within a scene. In total then, eight sources of light could be considered when trying to decide the color of each and every pixel in a ray-traced image. As you might expect, this would be exceedingly time-consuming. The Hall shading model described in *An Introduction to Ray Tracing* attempts to take most of the eight light propagation modes into consideration and is therefore very complex. Most ray-tracing programs make simplifications to this type of shading model. These simplifications trade off computation time for accuracy. The visual result of these simplifications is an image that is not exact, but close enough for general use. Some of the simplifications typically made to complex shading models are:

1. *The removal of frequency-dependent terms.* In real light/surface interaction, the frequency of the light factors into how the light plays on an object. A prism, for example, works because light bends differing amounts, depending on its frequency components. The omission of frequency-dependent terms from the shader equation means prisms cannot be modeled and other subtle lighting effects will be lost.
2. *Interobject reflections and transmissions ignored.* Interobject reflections and transmissions aid in the illumination of most scenes. Because many objects reflect light diffusely, they contribute some undirected light to other objects in a scene. Some of this light finds its way back to the eye and is therefore visible to the viewer. How best to handle this contribution to scene illumination is still a matter of much debate. One way to handle the calculations for indirect illumination within a scene is through a technique called "radiosity." Radiosity uses the laws of conservation of energy to figure out how much energy in terms of light would be radiated from a surface. The techniques of radiosity are beyond the scope of this book and are still for the most part research issues.

   Because of the difficulty involved in calculating indirect illumination from other objects within a scene, most basic shading models replace the interobject contributions to lighting with an "ambient light" term. Without something to replace the indirect illumination naturally occurring in a scene, all objects not directly illuminated by a light source would be black and generally invisible. The incorporation of an ambient term provides a small amount of light within a scene, which makes obscured objects visible and therefore simulates, very crudely, light reflected and transmitted by other objects.
3. *Distance.* As most everyone (technical people, anyway) is aware, light traveling through space is attenuated as the square of the distance. That is, light traveling twice as far is one-quarter as intense. Some shading models do not take the distance between the light source and objects nor the distance between objects that provided reflected light to another object into consideration. While this sounds like a serious omission, it really doesn't matter as long as all objects within a scene are placed a similar distance from the light sources. When distance is taken into consideration by dividing the calculated light intensities by the distance squared, the lighting effects seem somewhat harsh. For this reason, the $1/d^2$ is often softened to $1/(d+d_o)$, where d_o is a suitably chosen constant less than d. The visual effect of this minor change to reality is much more pleasing to look at. However, the shaders used in most basic ray-tracing programs ignore distance completely.

With all of these items left out of the shading model, you may be asking yourself what remains. Of course, this varies from implementation to implementation but in general most basic shaders consider:

- Ambient lighting
- Diffuse reflection
- Specular reflection
- Specular transmission (refraction)

We discuss the contribution of each of these light transport mechanisms below.

Ambient Lighting

As mentioned, ambient lighting is a contrivance that is meant to compensate for interobject indirect illumination. This method appears natural without requiring excessive calculations. Rays of ambient light within a scene can be envisioned to strike an object's surface from all directions and reflect off in all directions. The intensity of the ambient light reflected to the eye is independent of the direction to the viewer or the direction to the light source(s). Ambient lighting contribution can be calculated in one of two ways. First, by application of the formula:

$$I_a = k_a * I_l$$

where I_l is the intensity of the ambient light source and k_a is the ambient absorbation constant. The constant k_a determines how much of the ambient light will be reflected from the surface. The problem with this method of calculation is that the ambient light reflected from an object's surface is a function of the light's color, not the object's color. You can get a more realistic effect by using a different ambient light calculation. In this case:

$$I_a = k_a * I_o$$

where I_o represents the color of the object's surface and k_a determines how much of the surface's color should be visible with ambient lighting. A typical value of k_a is 0.4. With this method, objects within a scene illuminated only by ambient light will show a darker intensity (or shade) of their true color.

The equation above is for monochromatic (single color) light. To make use of it and the other equations within this section, you must apply it to all three RGB color components separately. The color version of this formula would then become:

$$I_{aRed} = k_{aRed} * I_{oRed}$$
$$I_{aGreen} = k_{aGreen} * I_{oGreen}$$
$$I_{aBlue} = k_{aBlue} * I_{oBlue}$$

More simplistic shaders make the absorbation constants k_{aRed}, k_{aGreen}, and k_{aBlue} equal, even though it is well known that most surface materials absorb differing frequencies of light (different colors of light) at different levels. Reality notwithstanding, this simplification is generally made.

Diffuse Reflection

Diffusely reflected light reflects in all directions with equal intensity. The theory that explains this phenomena is that as a photon of light hits a surface with diffuse reflective properties, it is temporarily absorbed by the surface. The increase in energy experienced by the atoms in the surface is momentarily heightened but cannot be sustained. Eventually the surface gives up a photon to lower its energy level back to a stable state. The photon emitted from the surface takes off in a random direction unrelated to the angle of incidence of the incoming photon. The contribution of diffusely reflected light to a surface will appear the same regardless of the position of the viewer. For this reason, the direction to the viewer's eye does not enter into the calculations. What is important is the relationship between a ray from the surface to the light source and the surface normal. The amplitude of the diffusely reflected light is proportional to the cosine of the angle between the incident light and the normal. This relationship is referred to as "Lambert's Law." If the light ray L and the normal N are both unit vectors, the cosine of the angle between them is their dot product. Further, if N dot L is less than or equal to zero, the surface faces away from the light source and therefore receives no contribution of light from it.

Since not all of the light that impinges a surface is diffusely reflected, another absorbation constant, k_d, is introduced. Therefore, the monochromatic equation for the contribution of diffuse reflection to the total shading equation is:

$$I_d = I_l * k_d * \cos \text{Theta} = I_l * k_d * N \text{ dot } L$$

For scenes illuminated with multiple light sources, the sum of all diffuse contributions (from all light sources) should be used. Multiple light sources are treated as if each source were the only light source within a scene and the individual contributions from each are summed together.

Just as before, the monochromatic equation above needs to be solved three times to be used with the RGB color model. Also, k_d would have different values of absorbation for each of the three colors. In practice, many shaders will simulate the differing values of k_d by introducing the color of the surface being shaded into the equation as follows:

$$k_{dRed} = k_d' * I_{oRed}$$
$$k_{dGreen} = k_d' * I_{oGreen}$$
$$k_{dBlue} = k_d' * I_{oBlue}$$

where k_d' is a single absorbation constant used in all three cases and I_o is the color of an object's surface. Intuitively, this makes sense in that the amount of light absorbed by a surface depends on both the color of the light and the color of the surface. For example, if white light is diffusely reflected off a red surface, the reflected light will appear red because the blue and green components of the light will have been absorbed (filtered out) by the surface. By essentially multiplying the color of the light by the color of the surface in these calculations, the filtering effects work as would be expected.

Specular Reflection

Specular reflection is exhibited by smooth surfaces. If a surface is smooth enough, specular highlights will appear. A specular highlight appears on the surface of the object as a small

bright patch of light that is the color of the light being reflected. The smoother the surface, the tighter the highlight appears. The highlights do not take on the color of the surface because the photons of light that impinge the surface are not absorbed and re-emitted by the surface as they were in the case of diffuse reflection. Instead, they immediately bounce off the hard surface at an angle of reflection equal to the angle of incidence; it works just like a billiard ball bouncing off the side of a billiard table or a rock thrown at a shallow angle into a still lake.

Unlike diffuse reflections, the contribution of specular reflection to a surface's color as perceived by a viewer is highly directional. To understand how specular reflections work, you must consult Figure 2.6(a). Specular highlights are caused by a light source being reflected off a surface directly into the viewer's eye. The angle between the reflection of the light source (R) and the ray to the viewer (V) determines how much of a contribution to a surface's color specular reflections make. When the vectors R and V coincide exactly, the maximum effect occurs. As these vectors diverge in direction, the effect is diminished. As before, the dot product is used to determine how closely R and V coincide, because it is the angle between them that interests us. *Note*: For the dot product comparison to work, both R and V must be unit vectors. But how is R arrived at? For a given ray/object intersection we will have the point of intersection P and the view vector V. From these, we can calculate the surface normal N by knowing the type of object intersected. We then generate the ray L from the point of intersection to light source L. Given this information and the fact that vectors R, N, and L would all be in the same plane and the fact that the angle of incidence equals the angle of reflection, the reflected unit vector R can be calculated from (I'll spare you the algebra):

$$R = 2*N*(L \text{ dot } N) - L$$

assuming the vector orientations shown in the figure.

Now that we have both vectors V and R, we can determine the maximum intensity of the specular highlight for the point being shaded. But as mentioned before, the size of the highlight is a function of surface smoothness, not just the geometry of the viewer and the reflection angles. Typically, the Phong reflection model is used to adjust for this phenomena. With the Phong model, a new factor n is introduced, which characterizes the surface material. The larger the value of n, the higher polish a surface has. If n equaled infinity, the surface would be perfectly smooth. Incorporating the Phong model into the equation for specular reflection yields the following:

$$I_s = I_l * k_s * (R \text{ dot } V)^n$$

A large value of n makes the specular highlight fall off very quickly as the eye ray direction diverges from the reflected light. This causes tighter specular highlights on smoother surfaces, as we expect. Note that the Phong model is based not on physics but on empirical observations. The visual result is very close to what occurs in nature but requires much less computation to arrive at. The specular absorbation constant, k_s, is a function of the surface material and should be broken into three constants for each of the RGB calculations. In more complex shading models, these absorbation constants would vary with the angle of incidence of the light rays.

A model for specular reflection that is based on theory and not empirical data is the "Torrance-Sparrow" model. This model is based on the concept that all surfaces are made up of

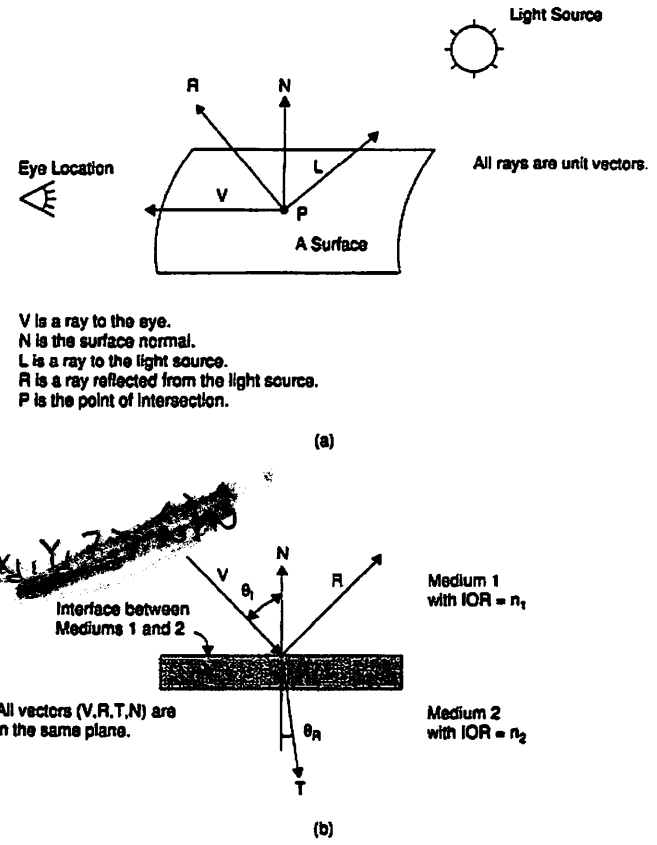


Figure 2.6 Shading Rays

"microfacets." These microfacets are tiny, flat, perfect reflectors. In rough surfaces, these microfacets are positioned randomly such that incident light would bounce around on the surface for quite a while before being reflected off. This would result in the reflected light taking on more of the object's surface color. Highly polished surfaces would have the microfacets aligned such that incident light would reflect immediately off the surface and its color would remain unaffected by the surface color. Intuitively, you can understand how calculating these effects would be more time-consuming then the Phong technique just presented.

Specular Transmission

The final light transport mechanism that we will examine is specular transmission, which is the phenomena whereby light arrives at a point on the surface of an object by passing through the object. For this to happen, of course, the object cannot be opaque to light; it must be somewhat transparent for the light to travel through it. When we look through transparent objects we find that in general the light rays will have bent. This phenomena is called "refraction" and is observed wherever light passes between mediums of differing density. The light bends because the speed of light through the differing media changes. The amount of bend is related to the difference in density of the media involved. This is the effect you see when you view fish in water. The fish are never really where you think they are because the light rays are bent when you see them.

To describe refraction mathematically, we must assign to each medium an "index of refraction," or IOR. The IOR measures the speed of light through the media in relation to the speed through an empty vacuum. Air has an IOR of almost exactly one. Water has an IOR of 1.333 and glass can have an IOR in the range of approximately 1.46 to 1.66, depending on its formulation. The bending of the light happens at the interface or junction between the two media involved. See Figure 2.6(b). "Snell's Law" relates the angle of incidence to the angle of refraction as follows:

$$\frac{\sin \text{Theta}_I}{\sin \text{Theta}_R} = \frac{n_2}{n_1} = n_{21}$$

The direction of the transmitted ray T can then be defined as:

$$T = n_{12} * V + (n_{12} * C - \sqrt{(1 + (n_{12}^2 * (C^2 - 1)))} * N$$

where C = V dot N. Note that T would not be a unit vector and would have to be normalized before being used. If the quantity $(1 + n_{12}^2 * (C^2 - 1))$ is less than zero, it indicates a condition referred to as "total internal reflection," or TIR. TIR occurs when light passes from a dense medium to one that is less dense at a shallow angle. If the angle is under a certain threshold, the incident light rays do not bend and pass through the media but reflect off the internal surfaces and back out the way they came. In this case, ray T does not exist and does not contribute to the point being shaded.

In terms of the overall shading equation, the lighting contribution of specular transmission is of the same form as specular reflection and can be expressed as:

$$I_{st} = I_l * k_{st} * (T \text{ dot } V)^{s'}$$

where all of the terms in this equation are analogous to their specular counterparts. This equation shows that transmitted light can cause highlights just as reflected light can.

Combining the lighting contribution of the four light transport mechanisms just discussed, we can produce a shading equation for use within a ray-tracing program. A shading equation for a single light source that incorporates ambient lighting, diffuse reflection, specular reflection, and specular transmission is of the form:

$$I = \frac{I_l}{d + d_o} * (k_d * (L \text{ dot } N) + k_s * (R \text{ dot } V)^s + k_{st} * (T \text{ dot } V)^{s'}) + k_a * I_a$$

where all terms have been previously identified. As mentioned, some shaders do not take distance into consideration, which would lead to a simplification of the equation above. You can also eliminate other terms in this equation if you don't need them for a specific application. You can apply the practical adjustments we discussed to the various terms in this equation as well.

Recursive Shading

The color assigned to a point on a surface struck by an eye ray is a combination of:

1. The interaction of light rays emanating from all light sources within a scene and the surface material. These interactions are governed by the shading equation and are referred to in this text as local shading.
2. Any contributions made by the reflection of light onto the eye ray.
3. Any contributions made by the transmission of light through a surface and into the eye ray.

Since the process begins at the eye and moves backwards through a scene, at any point in the overall shading computation there are always two unknowns, the reflected and transmitted contributions, which prohibit solving for the illumination of the point under consideration. To eliminate this Catch-22-like situation, we can calculate the shading of a point on a surface by the recursive application of the shading model which, as mentioned, builds a ray tree of contributing illumination sources. A ray tree, like other tree structures, contains a root node, intermediate nodes, branches, and leaves. The root node of this tree is the point being shaded and that viewable by the eye. Each branch contains contributions to the overall illumination of the surface, with the contributions growing less important percentage-wise as you go deeper into the ray tree. The leaf nodes of the tree contain the initial lighting contributions from which the surface shading can be derived. The shading to be applied to a point on the surface is the summation of the lighting contributions contained in the ray tree. The depth of the tree is a function of the recursion depth and must somehow be limited to allow the shading calculations to conclude. This is especially important in a scene that contains many reflective or transmissive surfaces, as the recursion could go on for quite a while taking all of the reflected and refracted rays into consideration. As mentioned, either a threshold of lighting contribution (this technique is referred to as adaptive tree depth) or a hard-coded recursion level is typically used to terminate recursion. When the recursion starts to unwind, the color (lighting) contributions calculated at the deepest levels bubble up toward the root of the tree and are combined with color (lighting) contributions calculated elsewhere in the tree. The final result is a color to be assigned to the first object intersected by the eye ray. At each juncture the light (or color) calculated by local shading is augmented by the contribution of reflected and refracted light. For reflected light:

$$\text{Color}_p = \text{Color}_s + \text{Reflection Coefficient} * \text{Color}_{\text{Reflected Ray}}$$