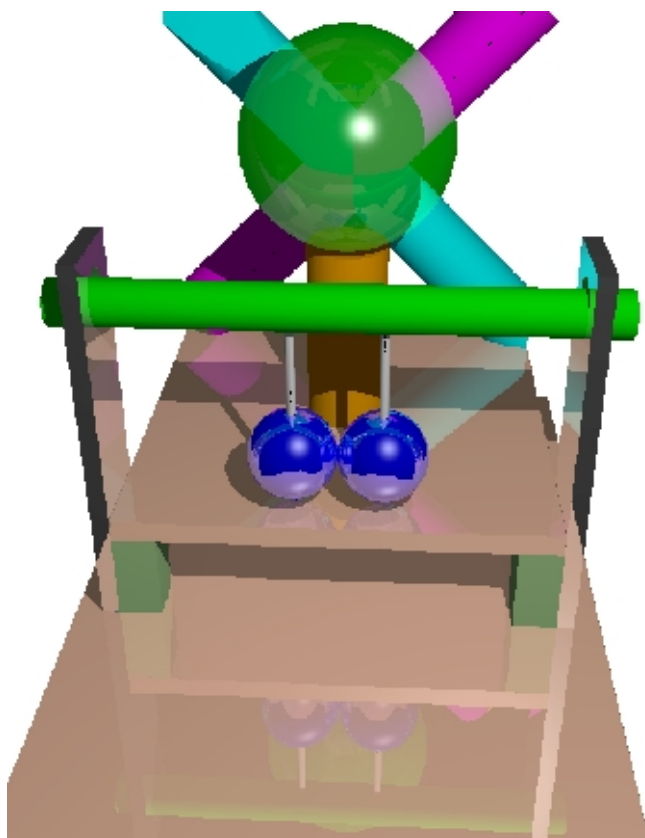

7

Modelos de Iluminación y Sombreado



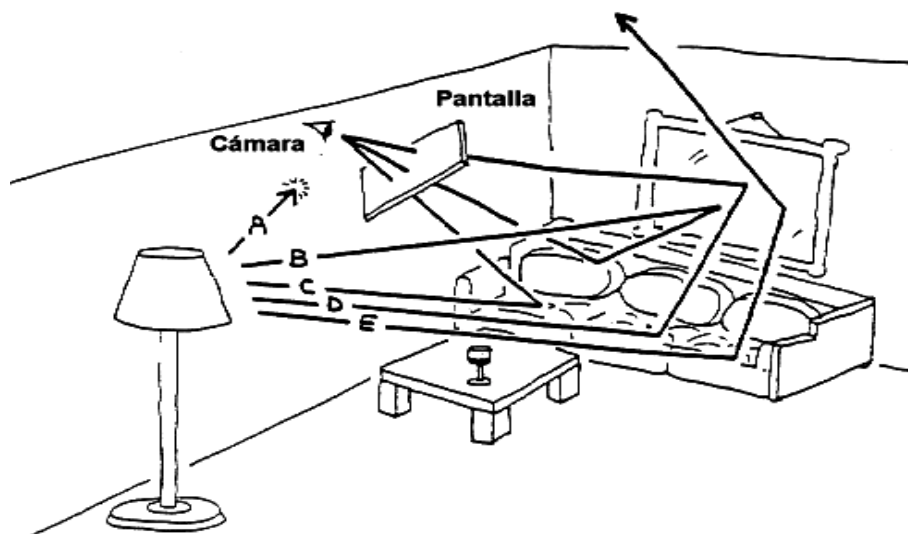


Figura 7.1 Una escena sintética y sus condiciones de iluminación

7.1 El Realismo en Computación Gráfica

Como ya mencionáramos en Capítulos anteriores, la búsqueda del realismo constituyó un cambio de paradigma en los objetivos de la Computación Gráfica durante la década del 70. El objetivo pasó a ser la creación de imágenes de escenas tridimensionales que parecieran *reales*. Una definición de realismo podría desembocar en un debate. El sentido que se le da en nuestra área está relacionado con la mayor fidelidad de la representación de atributos en los modelos computacionales subyacentes al proceso de generación de la imagen.

Por lo tanto es posible hablar de modelos “más” o “menos” realistas en función de la mayor o menor cantidad de características representadas, fundamentalmente en los modelos de interacción entre la luz y los objetos, y de la fidelidad con la que se representan dichas características. En un extremo están los modelos *fotorrealistas*, que buscan competir en fidelidad con los resultados fotográficos (ver [2]), mientras que en el otro están los modelos tan sencillos como la representación del *wireframe* o “armazón de alambre” que estamos manejando hasta ahora en este libro, modelos en los cuales los objetos son representados solamente con algunos vértices seleccionados y las aristas que los unen.

El realismo de una imagen tridimensional depende de la correcta simulación de los efectos de iluminación y sombreado. Se utiliza un modelo de iluminación para calcular las intensidades y colores de cada pixel de la imagen, mientras que las técnicas de sombreado permiten reducir la cantidad de pixels para los que se debe computar el modelo de iluminación. El modelo de iluminación representa una simulación de los fenómenos que, modificando o filtrando la distribución de energía de la luz incidente, dan lugar a los colores en las imágenes. Por ejemplo, en la Figura 7.1 se ilustra el efecto de querer simular una determinada condición de iluminación para una escena sintética, vista desde un determinado lugar. Los seis fenómenos fundamentales derivados de la interacción entre luces y objetos son reflexión, transmisión, absorción, difracción, refracción e interferencia. Aunque todos ellos se han modelado en computación gráfica, el más importante visualmente es la reflexión.

Los modelos de reflexión que se estudiaron en la década del 70 no buscaron simular las ecuaciones integrales que reproducen exactamente muchos efectos visuales (basadas en la teoría de ondas electromagnéticas y en las ecuaciones termodinámicas), sino que buscaron modelos empíricos que fueran un buen compromiso entre las restricciones en tiempo y equipo disponible y la simulación del comportamiento de la luz en la realidad. Ya aparecía el sombreado de objetos a fines de los '60 (en [87] y [83]) computándose con funciones inversamente proporcionales a la distancia al foco luminoso. Ahora se ignora casi completamente el efecto de la distancia en el cálculo de iluminación excepto en la simulación de niebla. Probablemente el primer modelo empírico se debe a Bouknight (en [16]), donde se evaluaba la intensidad de cada faceta de un objeto de acuerdo a su orientación respecto a la fuente de luz, en base a la ley de Lambert, y también incluía un término de luz ambiente para iluminar las partes no enfrentadas con fuentes de luz.

El modelo de Phong [69], que con el tiempo llegó a convertirse en el estándar, divide la reflectividad en componentes difusa y especular (brillo). Este es un modelo de reflexión especular local, es decir, no tiene en cuenta las interreflexiones entre objetos de la escena. Sin embargo, es elegante y produce resultados satisfactorios a la vista. El paso más importante en esta línea fue dado por Whitted [85], quien utiliza el modelo de Phong localmente pero extiende el modelo por medio de la técnica de trazado de rayos (ray tracing) utilizada por Appel en 1968 para determinar la cara oculta (ver Sección 4.6). Dichos rayos permiten computar la interreflexión entre objetos, y por lo tanto un modelo de iluminación no local.

Desde entonces, el tema del realismo ha experimentado un desarrollo notable. Al mismo tiempo han surgido nuevos modelos que retoman la idea de simular los fenómenos ópticos desde el punto de vista del electromagnetismo (por ejemplo, la ecuación del rendering de Kajiya) o del equilibrio termodinámico (por ejemplo, la radiosidad, de Greenberg et. al.). En este Capítulo presentaremos los elementos fundamentales de los modelos de locales de iluminación y de las técnicas de sombreado. Esto permite

concluir con el modelo de rendering denominado *scan-line*, el cual es la base de este texto. También describiremos brevemente los modelos de iluminación no locales, los cuales dan origen a dos modelos de rendering: el trazado de rayos o ray tracing, y la radiosidad. Por último, describiremos algunas técnicas que permiten mejorar la representación de diversas cualidades por medio de mapas de atributos.

7.2 Modelos de Iluminación

El objetivo de un modelo de iluminación es representar los fenómenos óptico-físicos que determinan el color con que debe graficarse una superficie en un punto dado. Pueden tenerse en cuenta distintos factores, y darle diferente importancia a cada uno. Por lo tanto existen varios modelos de iluminación, cada uno expresado mediante a una ecuación de iluminación que determina el color de un punto en función de la geometría, posición y orientación de la superficie a la que pertenece, el material que la compone, las fuentes de luz que la iluminan, y la interacción luminosa con otros objetos.

Es importante tener en cuenta que estas ecuaciones normalmente no provienen de soluciones simplificadas a las ecuaciones de la óptica o de la propagación de la radiación electromagnética. Por el contrario, tienen más bien una base empírica que la justifica por los resultados obtenidos en la práctica. Esto es así porque los modelos físico-matemáticos de la propagación luminosa son de una complejidad en principio inmanejable, aún para escenas sencillas. Por otro lado, la computación de los caminos luminosos posibles en una escena es un problema similar pero mucho más complejo que el de la eliminación de las partes ocultas al observador, que ya estudiáramos en Capítulos anteriores. Por lo tanto, es necesario contar con un modelo de iluminación, aunque sea una aproximación muy cruda, que permita comenzar a recorrer el camino ascendente del realismo.

Una de las primeras simplificaciones consiste en omitir del modelo la interacción luminosa entre objetos, la cual, sin duda, es la parte más compleja del cálculo de una ecuación de iluminación. Este tipo de modelos se denominan *locales*, dado que resuelven la ecuación de iluminación de un punto solamente a través de la información geométrica local a dicho punto. Los modelos que estudiaremos en esta Sección son de estas características, dejando para la Sección 7.4 una breve reseña de las ecuaciones de iluminación basadas en modelos físicos, para la Sección 7.5 el tratamiento de modelos no locales basados en aplicaciones recursivas ray tracing y para el próximo Capítulo una introducción a los modelos de refracción.

Los modelos de iluminación que presentaremos en esta Sección representan la combinación de diversos factores, como por ejemplo la radiación luminosa de cada objeto, la reflexión de luz proveniente de fuentes de iluminación, etc. Una caracterización fiel de las mismas desde el punto de vista de la teoría del color significaría representar completamente la interacción para cada frecuencia visible del espectro. En la práctica esto es innecesario, porque como vimos en el Capítulo 5, el ojo transforma dicha información espectral a una representación cromática simplificada. Por lo tanto, es solo necesario computar el modelo de iluminación en ciertas frecuencias espectrales, por ejemplo en el modelo RGB.

El modelo de iluminación más sencillo es considerar que los objetos radian su propia energía luminosa, sin interactuar con fuentes de luz. Por lo tanto, cada objeto se muestra con un color constante propio

$$I = k_i,$$

donde I es la intensidad resultante, y k_i es la intensidad luminosa propia del objeto i . En el modelo RGB debemos computar tres veces esta ecuación, una para cada primario, donde tendríamos tres intensidades resultantes y tres intensidades propias de cada objeto, que determinan específicamente su color. Este modelo es obviamente muy simplificado, pero tiene la ventaja de que se evalúa solo una vez por objeto.

Para poder comprender de qué manera se representan los demás factores en un modelo de iluminación en su correspondiente ecuación, necesitamos previamente conocer y clasificar los mecanismos por medio de los cuales es transportada la energía luminosa. En particular, nos concentraremos primero en la reflexión.

Físicamente existen dos posibilidades cuando un fotón incide sobre un objeto: el mismo es absorbido y por lo tanto su energía se disipa en forma de calor, o bien es vuelto a emitir. Este último fenómeno reconoce otras dos posibilidades: o bien el fotón no encuentra un lugar en el espacio de energías del objeto, y por lo tanto “rebota” balísticamente, o bien es absorbido temporalmente excitando la órbita de un electrón en un centro de color, para luego ser reemitido.

Estos tres fenómenos se denominan, respectivamente, absorción, reflexión especular y reflexión difusa, y la energía luminosa incidente total debe repartirse entre los tres. Los adjetivos “especular” y “difuso” se refieren a que en el primer caso los fotones literalmente “rebotan” y por lo tanto la geometría del rayo luminoso reflejado puede conocerse a partir de la geometría del rayo incidente y de la superficie del objeto, mientras que en el segundo caso dicha relación se pierde, debiendo utilizarse factores estadísticos para simularlo.

Por su parte, las fuentes de luz pueden producir un haz luminoso geoméricamente coherente, en cuyo caso hablamos de “iluminación puntual” dado que la dirección de un rayo luminoso en un punto puede computarse como la dirección entre la fuente luminosa y dicho punto. Sin embargo, al existir fenómenos de reflexión difusa, es posible que la coherencia geométrica se rompa, y que por lo tanto la energía luminosa en un punto deba computarse con una función más compleja. Una aproximación a este fenómeno es considerar que en todo punto y en toda dirección existe una intensidad luminosa determinada, a la cual se denomina “iluminación ambiente”.

Tenemos por lo tanto que considerar cuatro mecanismos de transporte: la reflexión especular de la iluminación puntual, la reflexión difusa de la iluminación puntual, la reflexión especular de la iluminación ambiente, y la reflexión difusa de la iluminación ambiente. Normalmente los dos primeros mecanismos son fácilmente calculables, mientras que los otros dos no, y por lo tanto se los aproxima con un único “factor ambiente”. De esa manera tenemos

$$I = I_a k_a,$$

donde I_a es la intensidad de la iluminación ambiente (en cada primario) y $k_a \in [0, 1]$ es el coeficiente de reflexión ambiente del material (también en cada primario).

Este modelo es también muy sencillo, aunque es una grosera simplificación de los fenómenos de iluminación ambiente. Produce escenas donde los objetos se ven antinaturalmente uniformes en brillo, pero sensibles a los cambios de iluminación ambiente. Por ejemplo, un objeto amarillo ($k_a = \langle 0.8, 0.8, 0 \rangle$) se verá rojo si es iluminado con luz ambiente roja ($I_a = \langle 1, 0, 0 \rangle$), amarillo si la luz es blanca ($I_a = \langle 1, 1, 1 \rangle$), y negro si la luz es azul ($I_a = \langle 0, 0, 1 \rangle$).

Uno de los primeros modelos de iluminación implementados fue el de Wylie, en 1967. El modelo sombreaba los puntos de la superficie de un objeto usando una intensidad (monocromática) inversamente proporcional a la distancia del punto a la fuente de luz. Superficies planas, entonces, reciben distintas intensidades. Otro desarrollo fue la consideración de la orientación de los puntos en la superficie con respecto a la fuente de luz. Bouknight [16] agregó en 1970 un término difuso Lambertiano (dependencia por el coseno del ángulo) para evaluar la intensidad de objetos construidos con polígonos o caras planas.

Consideremos ahora qué sucede cuando un objeto reflector difuso es iluminado con luz puntual. Esto es lo que sucede con objetos como la tiza, los cuales reflejan la luz incidente en todas las direcciones por igual. Por lo tanto, la intensidad luminosa reflejada depende exclusivamente de la intensidad luminosa incidente. Sea el punto p el lugar de incidencia, y sea N el versor normal al plano tangente de la superficie del objeto en p (ver Figura 7.2).

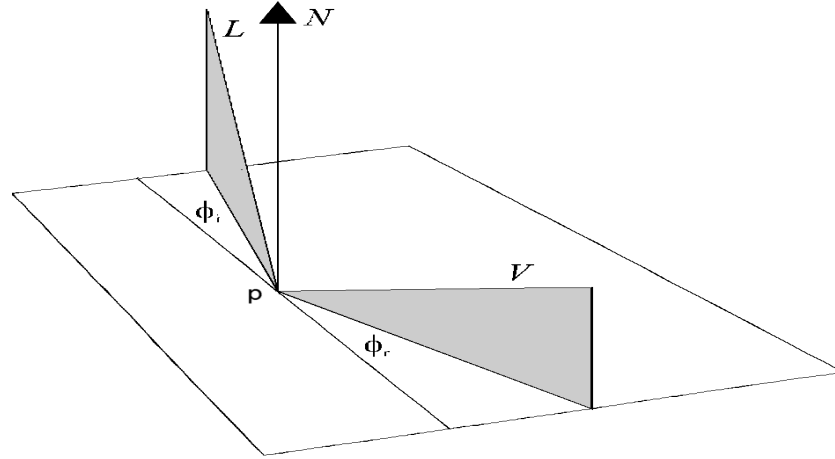


Figura 7.2 Vectores asociados al modelo de reflexión difusa.

Para el punto p , la intensidad luminosa reflejada es un factor constante por la energía incidente. Esta última sólo depende del ángulo θ que forman la dirección L que va de p a la fuente de luz (llamado vector de incidencia) y el normal N a la superficie en p (ecuación de Lambert):

$$I = I_p k_d (\cos \theta),$$

donde I_p es la intensidad de la luz incidente en el punto p , k_d es el coeficiente de reflexión difusa para el material que constituye el objeto, y $\theta \in [-\frac{\pi}{2}.. \frac{\pi}{2}]$ es el ángulo entre el vector normal y el vector de incidencia. Si $|N| = |L| = |V| = 1$, entonces es posible utilizar las identidades vectoriales (ver Apéndice 2) y expresar la ecuación de Lambert como

$$I = I_L k_d (N \cdot L).$$

Si el iluminante es lejano, entonces el vector de incidencia L puede considerarse constante, lo cual se denomina también “luz direccional”.

También puede agregarse al modelo una función de atenuación similar a la propuesta por Wylie. En general el modelo de inversa al cuadrado de la distancia da variaciones muy abruptas, por lo que se utiliza una función

$$f_{att} = \min \left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1 \right),$$

donde las constantes c_1, c_2, c_3 , determinadas por el usuario, simulan algún efecto atmosférico. La ecuación de iluminación es, entonces:

$$I = I_a k_a + f_{att} I_L k_d (N \cdot L).$$

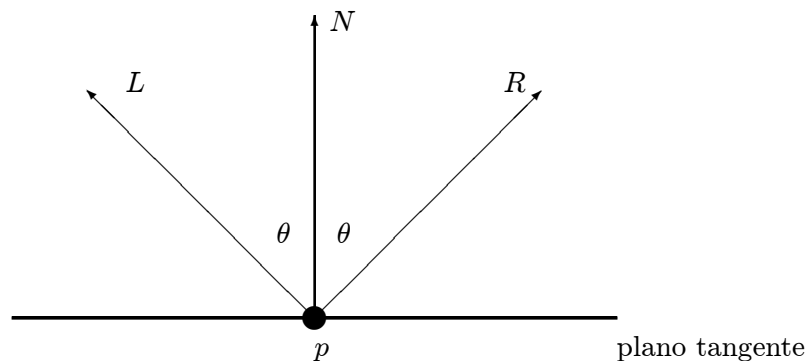


Figura 7.3 Geometría de la reflexión especular.

Este modelo resultó ser poco satisfactorio, porque las caras “invisibles” desde la fuente de luz no reflejan energía luminosa, lo que le da una apariencia muy artificial a las escenas. Por ello, Bouknight también incluyó un término ambiente para modelar la iluminación de las partes de un objeto que son visibles al observador pero invisibles para la fuente de luz.

Si el modelo de iluminación fuese el de un objeto especular puro, el 100% de la energía luminosa incidente se reflejaría en la dirección R (ver Figura 7.3). En cambio, si el objeto es reflector Lambertiano puro, el coseno del ángulo θ entre el normal N y la dirección V hacia la cual se encuentra el observador no se aplica para la energía reflejada. Es decir, la *intensidad* luminosa de un reflector Lambertiano puro, vista desde cualquier dirección, es constante. Esto se debe a que, si bien la *energía* luminosa radiada hacia el observador depende de la ubicación del observador, también el ángulo sólido subtendido por un diferencial de área alrededor del observador depende de dicha dirección. Como la energía es la integral de la intensidad por unidad de área, al variar la dirección del observador, varía la energía pero también varía el área en forma idéntica, con lo cual ambos factores se cancelan y la intensidad permanece constante.

En la práctica es fácil ver que ningún reflector es Lambertiano puro. Los objetos tienden a reflejar mayor intensidad hacia el ángulo de la reflexión especular predicha por la ley de Snell (ver Figura 7.4). Phong Bui-Tuong propuso un modelo de reflexión en el cual es posible separar una componente difusa (Lambertiana) de una especular, cada una con su coeficiente específico. De esa manera, es posible determinar una ecuación de iluminación en la cual interviene también el ángulo que forman el normal y la dirección del observador.

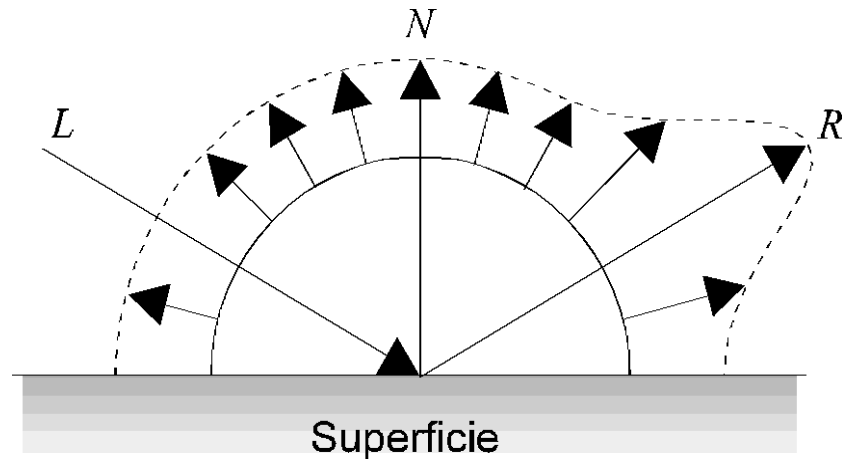


Figura 7.4 Una superficie normalmente refleja una combinación difusa y especular.

El modelo de Phong es el que más se usa en Computación Gráfica. En él la reflectividad de una superficie está dada por la contribución de dos factores. El primero es esencialmente un factor Lambertiano, que depende de un coeficiente definido para el material, y del coseno del ángulo θ de incidencia. El segundo es un factor que tiene en cuenta la reflexión especular, y depende de un coeficiente del material y del coseno del ángulo α entre la dirección de reflexión ideal R y la dirección V al observador.

Pero la observación crucial de Phong es que en la reflexión especular, el “brillo” que se produce aparece del mismo color que la luz (con excepción de los metales), y en general no depende del color de la superficie. Por lo tanto nuestras ecuaciones deben ahora tener en cuenta la distribución espectral (el color) del iluminante y del objeto por separado. Designaremos, entonces, con I_λ a la intensidad resultante del modelo a cierta longitud de onda λ , (que normalmente es alguno de los primarios R, G, B), $I_{a\lambda}$ es la intensidad de la iluminación ambiente a dicha longitud de onda, y $I_{p\lambda}$ es la intensidad del iluminante puntual. También designaremos con O_λ al color del objeto, de modo que $k_a O_\lambda$ corresponde al coeficiente de reflexión ambiente a una determinada longitud de onda.

Otro problema es que la ecuación de la reflexión especular predicha por la ley de Snell implica que toda la energía se refleja especularmente en una única dirección (ver Figura 7.3):

$$R = 2N(N \cdot L) - L.$$

Phong propone entonces una función empírica para determinar en forma efectiva y sencilla la distribución geométrica de la energía luminosa reflejada en forma especular.

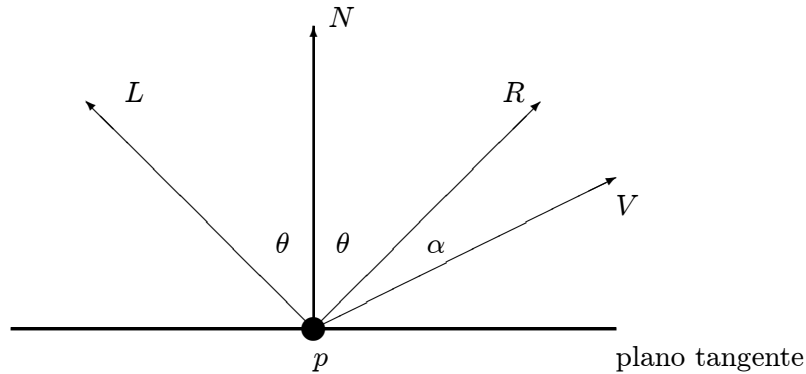


Figura 7.5 El ángulo entre el rayo reflejado ideal y la dirección del observador.

Dicha función parte de considerar el ángulo α entre el rayo reflejado ideal R y la dirección del observador V (ver Figura 7.5).

Utilizar el coseno de dicho ángulo determina funciones de reflexión especular que “decaen” lentamente a medida que la dirección de observación se aparta de la del rayo reflejado. Esto es característico de superficies poco pulidas, como el papel común, la madera, y algunas telas. Pero en superficies más pulidas, dicho decaimiento es más abrupto, o dicho de otra forma, la influencia de la reflexión especular es intensa pero muy localizada. Por dicha razón, la función de *glossiness* o pulimiento propuesta por Phong es $\cos^n \alpha$, donde n puede ser un entero entre 1 y 100:

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} \cos \theta + W(\theta) \cos^n \alpha],$$

donde

- $I_{a\lambda} k_a O_{d\lambda}$: Iluminación ambiente.
- $f_{att} I_{p\lambda} k_d O_{d\lambda} \cos \theta$: Término de reflexión difusa.
- $f_{att} I_{p\lambda} W(\theta) \cos^n \alpha$: Término de Phong (iluminación especular no ideal).
- n : factor de *glossiness*.

Como antes, si L , V , R y N están normalizados, entonces $\cos \theta = (N \cdot L)$ y $\cos \alpha = (R \cdot V)$. $W(\theta)$ es utilizada para modelar funciones de reflectividad que varían con el

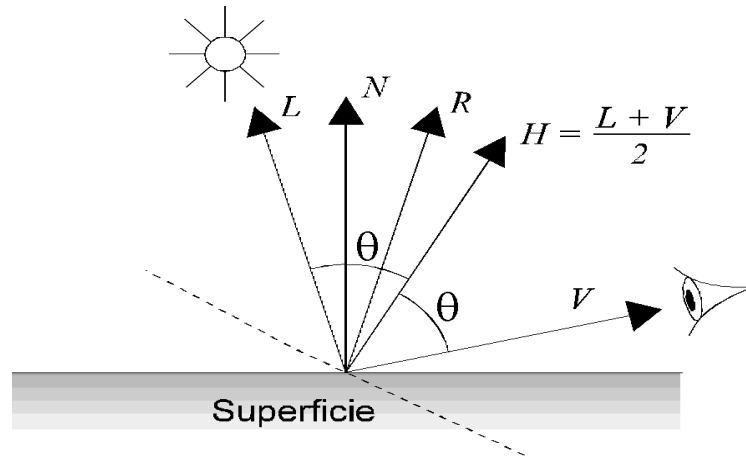


Figura 7.6 Interpretación geométrica del vector H .

ángulo de incidencia, aunque normalmente una constante k_s (coeficiente de reflexión especular) $k_s \in [0..1]$ da resultados aceptables.

De esa manera, la ecuación de iluminación de Phong, en función de los coeficientes (cromáticos) de reflexión ambiente, difusa y especular, y de la intensidad (cromática) de la iluminación ambiente y puntual, es

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} (N \cdot L) + k_s (R \cdot V)^n] \quad (7.1)$$

Es importante observar que k_s no depende de la longitud de onda, haciendo que el color del brillo sea el mismo que el de la fuente de luz (dado por $I_{p\lambda}$). Además, según se propuso, el exponente n controla el “brillo” del material. Sin embargo, el efecto que resulta al variar n es simular que la fuente de luz varía su tamaño. Reduciendo n hace que la fuente luminosa parezca mayor; pero **no** reduce el brillo del objeto (ver Figura 7.8 [84]).

Blinn [8] propone una manera de simplificar el costo computacional de la ecuación 7.1. Básicamente propone un vector H el cual está a mitad de camino entre R y V (ver Figura 7.6):

$$H = \frac{(L + V)}{|L + V|}.$$

De esa manera, la componente especular de la ecuación de Phong es

$$(N \cdot H)^n.$$

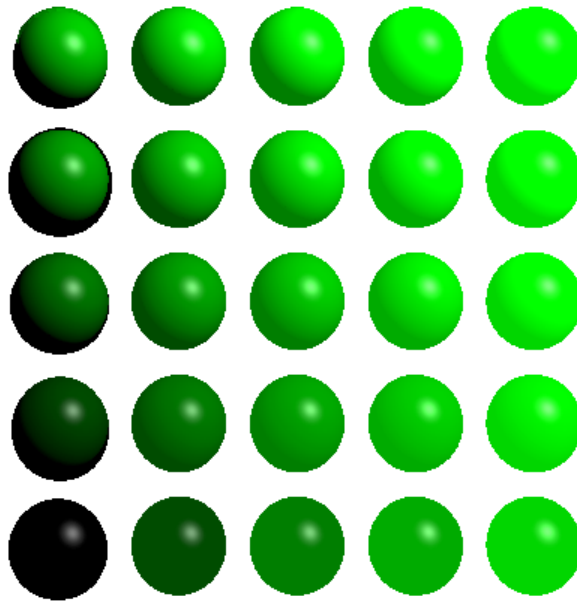


Figura 7.7 Modelo de iluminación de Phong. Se varía de izquierda a derecha el coeficiente de iluminación ambiente de 0 a 1, y de arriba abajo el coeficiente de reflexión difusa de 0 a 1.

Como el cálculo de dicha potencia no debe realizarse necesariamente con gran precisión, es posible aproximarlos satisfactoriamente con tablas o funciones polinomiales de bajo orden. De esa manera, el modelo de Phong y su ecuación 7.1, si bien parte de una base empírica, logra resultados bastante adecuados en tiempos razonables.

En la Figura 7.7 podemos observar los diferentes efectos que produce, para un iluminante y un objeto dado, el cambio de los coeficientes de iluminación ambiente y reflexión difusa, para un factor de glossiness y un factor de reflexión especular fijos. En la Figura 7.8 podemos observar los diferentes efectos que produce en el mismo objeto el cambio de los coeficientes de reflexión especular y factor de glossiness, para factores de reflexión difusa e iluminación ambiente fijos. Una última mejora que mencionamos al modelo de Phong proviene de considerar que las fuentes luminosas pueden no radiar energía uniformemente en todas las direcciones. Para ello puede elevarse el coseno del ángulo *de incidencia* a una cierta potencia, simulando luces reflectoras, o bien utilizar una función umbral para simular haces muy concentrados y *spots*.

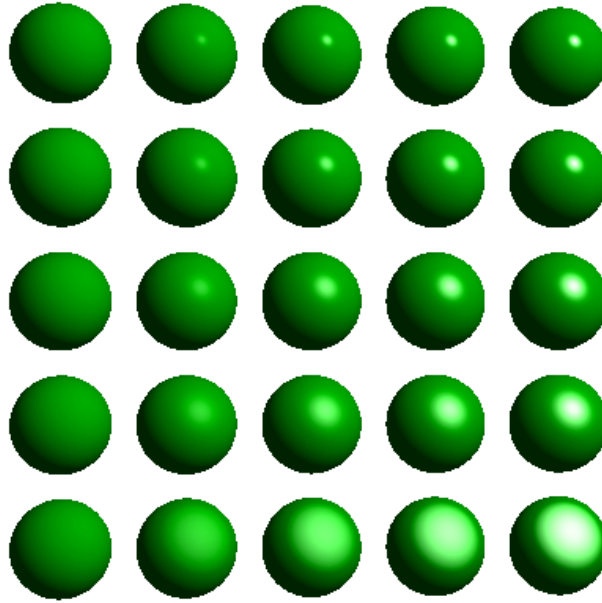


Figura 7.8 Modelo de iluminación de Phong. Se varía de izquierda a derecha el coeficiente de reflexión especular de 0 a 1, y de arriba abajo el coeficiente de glossiness de 1 a 50.

La ecuación de Phong es también adecuada para modelar varios iluminantes, debiendo computarse separadamente la contribución especular y difusa de cada uno:

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + \sum_{i=\text{luces}} f_{att}^i I_{p\lambda}^i \left[k_d O_{d\lambda} (N \cdot L^i) + W(\theta) (R^i \cdot V)^n \right].$$

Podemos ver en la Figura 7.9 el resultado de computar un objeto con varias fuentes de luz. Sin embargo, la interreflexión no puede determinarse, por lo cual el modelo sigue siendo *local*.

7.3 Sombreado de Polígonos

Si bien los resultados de la Sección anterior pueden ser satisfactorios, en la práctica se demuestra que no es necesario tomar una muestra del modelo de iluminación local para cada pixel de la escena. Normalmente este cómputo es redundante y muy ineficiente. Por lo tanto, en esta Sección estudiaremos algunas técnicas que permiten acelerar el

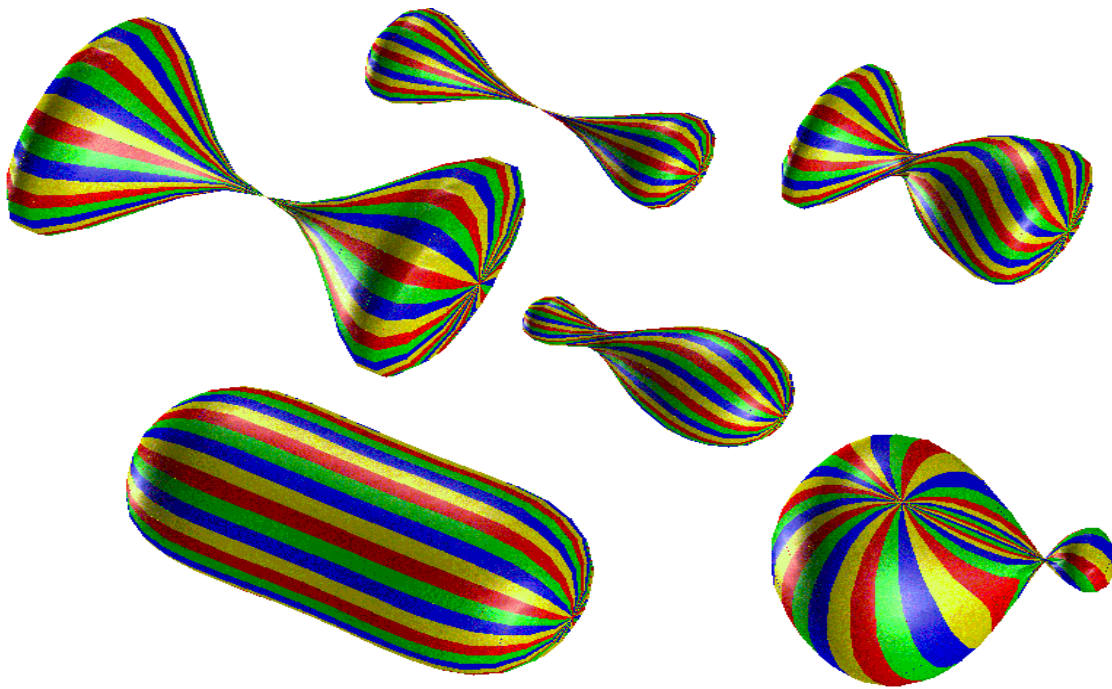


Figura 7.9 Modelo de iluminación de Phong con iluminantes múltiples.

dibujado de escenas con realismo por medio de técnicas de *sombreado*, es decir, por medio de interpolaciones entre los valores de las muestras.

La manera más rápida de pintar un polígono, es por medio de la técnica de “sombreado constante”. El modelo de iluminación es utilizado para calcular una única intensidad, con la cual se “pinta” todo el polígono. Si la fuente luz está muy alejada, entonces $N \cdot L$ es constante en toda la cara del polígono. Si además el observador está alejado, entonces también $N \cdot V$ es constante. Por lo tanto, muestreamos la ecuación para todo el polígono: (en el centro o en un vértice). La intensidad resultante es utilizada durante la conversión scan del polígono para setear cada uno de sus pixels (ver Figura 7.11). El algoritmo para graficar una escena con este método puede verse en la Figura 7.10.

Si alguna de las dos suposiciones deja de ser verdadera, es decir, o bien la fuente de luz o el observador está próximo al polígono, entonces el esquema comienza a ser más y más artificial. El color de cada polígono dependerá de dónde se efectúe la muestra, dado que con un mismo normal, en distintos puntos de la cara el modelo de iluminación arrojará colores diferentes. Puede tomarse una muestra en cada vértice,

```
procedure sombreado_constante(e:escena);  
...  
begin  
  para cada objeto o en e  
    para cada cara de o  
      computar un normal n  
      computar el modelo de iluminacion para n  
      efectuar la conversion-scan de o  
end
```

Figura 7.10 Algoritmo de sombreado constante de polígonos.

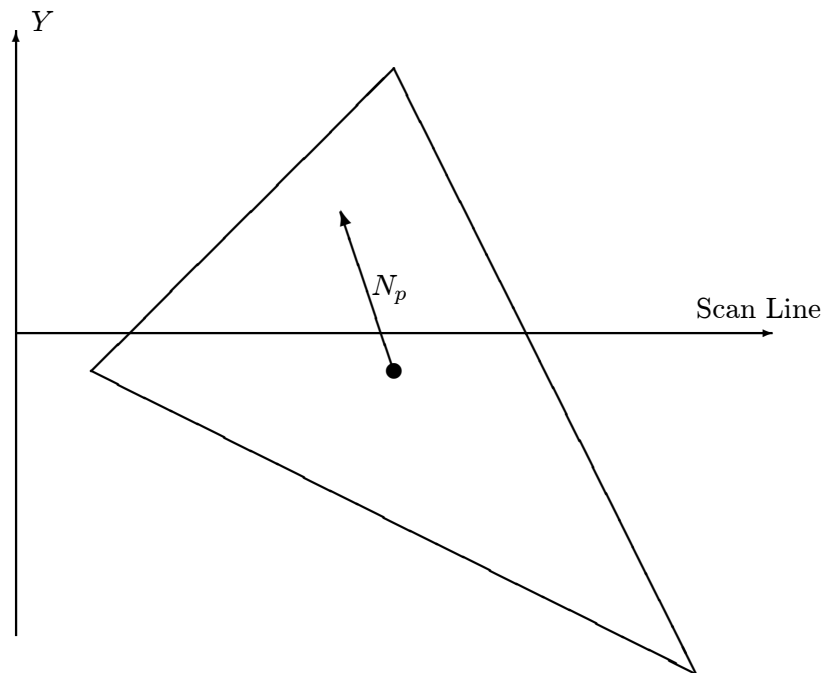


Figura 7.11 Sombreado constante de polígonos.

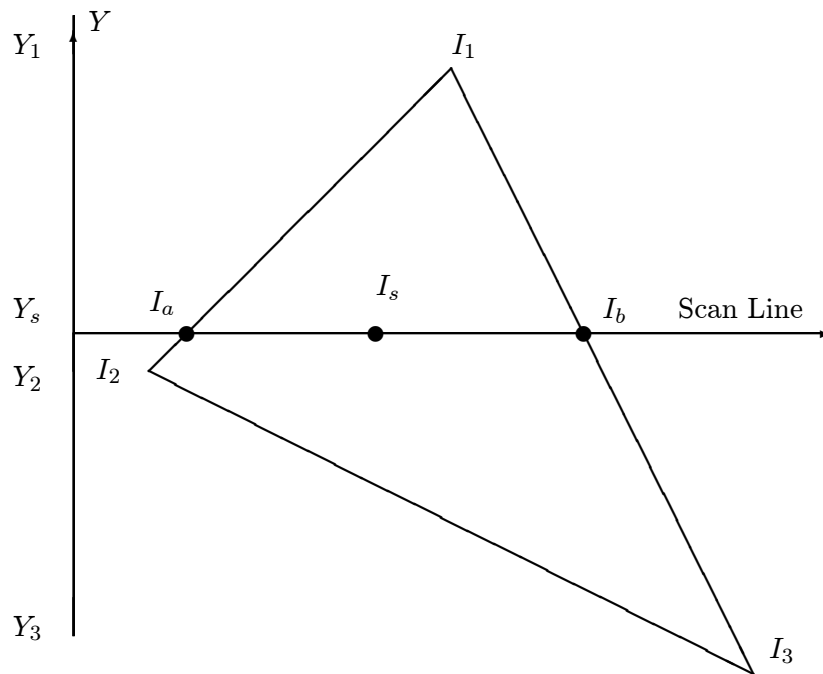


Figura 7.12 Técnica de sombreado de Gouraud.

y pintar cada polígono con el promedio de los colores de los vértices. Sin embargo, se seguirá obteniendo el desagradable facetado resultante.

La primer técnica de sombreado suave, que soluciona este problema sin aumentar la cantidad de muestras, se debe a Gouraud [29]. Como en el sombreado constante, se calcula el modelo de iluminación en cada vértice (es decir, en promedio una evaluación por polígono). Durante la conversión scan del polígono, se interpola el valor de iluminación a lo largo de las aristas. Por ejemplo, dado un triángulo con vértices 1, 2, 3, se computa el modelo de iluminación en dichos vértices, obteniéndose I_1, I_2, I_3 (ver Figura 7.12). Para una línea de scan, interpolando entre I_1 e I_2 podemos obtener I_a , y entre I_1 e I_3 podemos obtener I_b . Luego, para todo pixel en la línea de barrido, se interpola nuevamente entre I_a e I_b para obtener el valor de I_s .

$$\begin{aligned}
I_a &= I_1 - (I_1 - I_2) \frac{Y_1 - Y_a}{Y_1 - Y_2}, \\
I_b &= I_1 - (I_1 - I_3) \frac{Y_1 - Y_a}{Y_1 - Y_3}, \\
I_s &= I_b - (I_b - I_a) \frac{Y_b - Y_s}{Y_b - Y_a}.
\end{aligned}$$

Normalmente los normales en los vértices están mal definidos, porque dependerán con respecto a qué cara se los considere. Sin embargo, es posible promediar los normales definidos según cada cara, lo cual dará una medida consistente. Si los polígonos fueron contruidos de manera disciplinada, tal cual se sugiere en la Sección 6.4, entonces encontrar los normales es bastante directo. Es muy importante destacar que todas estas operaciones que se realizan durante la conversión scan del polígono, pueden efectuarse con técnicas aritméticas que posibilitan su implementación por hardware. Tal es así que en la actualidad las PC poseen tarjetas gráficas que reciben directamente los puntos e iluminaciones de los polígonos, y efectúan la conversión scan y sombreado de Gouraud, *miles* de polígonos por segundo!

Esta técnica realiza una aproximación muy grande, y por lo tanto puede tener resultados insatisfactorios. El ejemplo característico es el sombreado de un polígono iluminado por una fuente de luz cercana, ubicada sobre su centro. La evaluación de la iluminación en cada vértice será aproximadamente similar, por lo que todo el polígono será pintado de un color constante! Una solución para este problema y para varios otros consiste en subdividir los polígonos. Practicamente en todos los métodos derivados del scan-line, la solución por “fuerza bruta” es aumentar la cantidad de polígonos.

Consistentemente con la técnica de Gouraud, Phong elaboró otra técnica denominada *sombreado de Phong* (que no debe confundirse con el modelo de iluminación de Phong). La técnica consiste en interpolar los normales en cada punto de la conversión scan del triángulo, a partir de los normales en cada vértice del mismo. Esto es geométricamente mucho más preciso, y restaura la curvatura a las caras. En el mismo, la interpolación se realiza durante la conversión scan, como en Gouraud, primero entre vértices, y luego entre líneas de barrido (ver Figura 7.13):

$$\begin{aligned}
N_a &= N_1 - (N_1 - N_2) \frac{Y_1 - Y_a}{Y_1 - Y_2}, \\
N_b &= N_1 - (N_1 - N_3) \frac{Y_1 - Y_a}{Y_1 - Y_3}, \\
N_s &= N_b - (N_b - N_a) \frac{Y_b - Y_s}{Y_b - Y_a}.
\end{aligned}$$

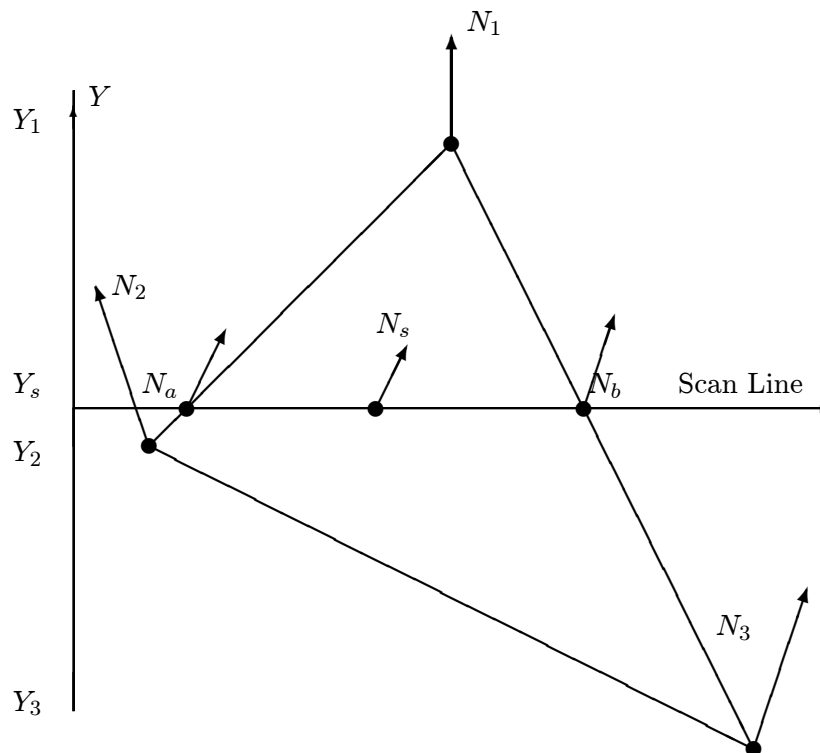


Figura 7.13 Sombreado de Phong.

El sombreado de Phong es más preciso que el de Gouraud, reproduce mejor la reflexión especular, inclusive la interior a un polígono, y restaura la curvatura en los objetos poligonizados. Sin embargo, es más costoso computacionalmente que el de Gouraud, dado que para cada pixel debe calcularse el modelo de iluminación. Esto, además, hace que no sea posible su implementación por hardware, al menos por ahora.

Todas las técnicas de sombreado que vimos tienen problemas en general. Básicamente, los objetos aparecen curvados, pero sus siluetas siguen siendo poligonales. También es necesario tener en cuenta que el recorrido de un polígono por scan-line no representa una recorrida uniforme en el espacio del objeto, debido al efecto de la perspectiva. De esa manera, un sombreado con interpolación lineal como Gouraud o Phong produce una distorsión en el efecto de la perspectiva. Esto también sucede si el objeto es rotado. Es decir, en general la interpolación no es invariante frente a transformaciones.

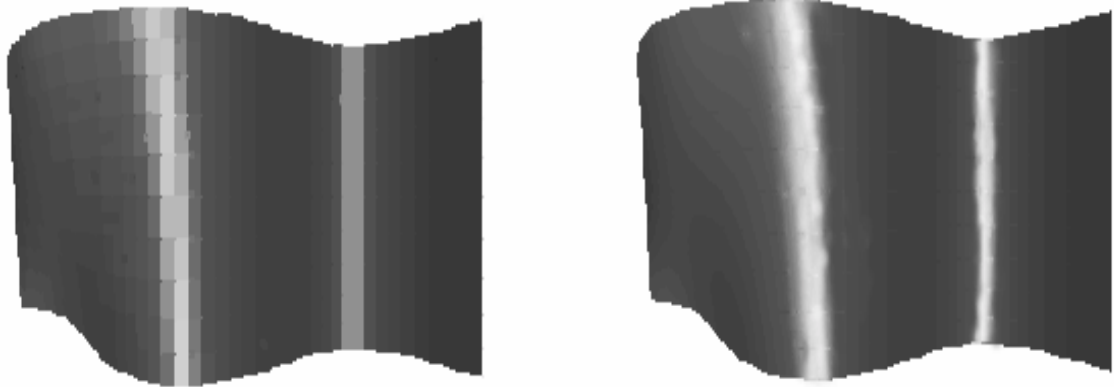


Figura 7.14 *Sombreado constante y sombreado de Phong.*

También pueden ocurrir problemas geométricos diversos. Por ejemplo, si los polígonos no conforman una red regular, entonces pueden existir polígonos vecinos que no compartan un vértice. De esa manera, el valor encontrado para un punto vecino a dicho vértice puede ser muy diferente al del vértice mismo. Por último, promediar los normales para encontrar los normales en los vértices puede producir una uniformidad artificial.

7.4 Modelos de Iluminación Avanzados

Como ya mencionáramos, el modelo de iluminación de Phong es una solución empírica al problema, sin una verdadera base física. En esta Sección presentaremos algunos resultados importantes en el desarrollo de modelos de iluminación de mayor realismo, basados en consideraciones de la física aplicada, en particular para reemplazar el término de reflexión especular del modelo de Phong. El punto de partida es el modelo de superficie de Torrance-Sparrow, adaptado por Blinn para su uso en Computación Gráfica [8].

En este modelo, se asume que la superficie de un objeto está compuesta por pequeñas microfacetas, cada una de las cuales es un reflector ideal, pero cuya geometría y orientación es imperfecta y sigue una determinada distribución estadística. Los parámetros de dicha distribución determinan la rugosidad y orientación de la misma. De esa manera, se supone que solamente una fracción D de las microfacetas totales está orientada de

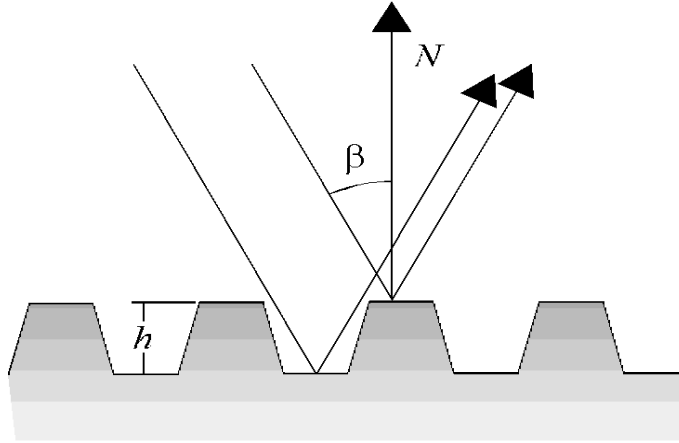


Figura 7.15 Modelo geométrico de Torrance-Sparrow para un objeto rugoso.

manera que la iluminación puntual se refleje en forma especular hacia el observador. Si β es el ángulo entre el normal N y el vector H (ver Figura 7.15), y h es la altura media de la rugosidad de la superficie (ver Figura 7.15), entonces una distribución estadística Gaussiana de dicho coeficiente es:

$$D(h, \lambda, \beta) = e^{-(4\pi \frac{h}{\lambda} \cos \beta)^2}$$

el cual modifica la reflectividad especular.

Pero Torrance-Sparrow y Blinn muestran que otras distribuciones producen efectos más adecuados, además de tener justificaciones físicas adecuadas [8]. En particular se utiliza una distribución denominada distribución de Beckmann:

$$D(h, \lambda, \beta) = \frac{1}{4m^2 \cos^4 \beta} e^{-\frac{\tan^2 \beta}{m^2}}.$$

Se definen también dos nuevos factores. G_s es un factor de “sombreado” que describe la interferencia de otras microfacetas sobre el rayo que incide en la superficie, y G_m es un factor de “enmascaramiento” que pesa la interferencia de otras microfacetas sobre un rayo que ya rebotó en la superficie. Se calculan como:

$$G_s = \frac{2(N \cdot H)(N \cdot L)}{V \cdot H},$$

$$G_m = \frac{2(N \cdot H)(N \cdot V)}{V \cdot H}.$$

La atenuación por enmascaramiento G es el mínimo entre G_m y G_s , excepto si ambos se hacen mayores que 1 (en ángulos de incidencia casi perpendiculares), en cuyo caso se toma $G = 1$.

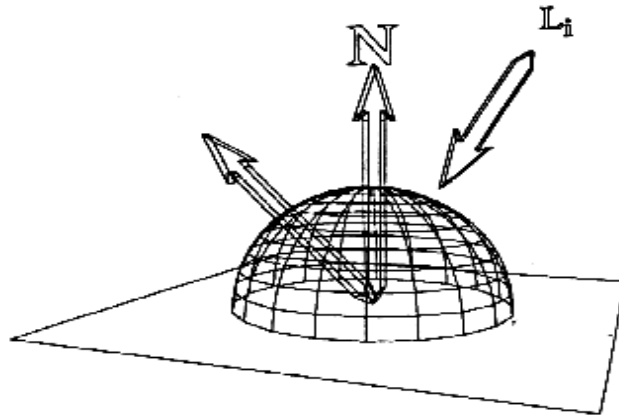


Figura 7.16 Geometría de la ecuación del rendering para un punto dado.

Reemplazando el nuevo modelo de reflexión especular en el modelo de Phong, pero manteniendo el término Lambertiano y el ambiente, obtenemos el denominado modelo de Phong-Blinn:

$$I_{\lambda} = I_{a,\lambda}k_aO_{d,\lambda} + f_{att}I_{p,\lambda}[k_dO_{d,\lambda}(L \cdot N) + k_s \frac{DG\rho_{\lambda}(\beta)}{N \cdot V}].$$

La diferencia entre este modelo y el de Phong es notable con ángulos de incidencia de 70 grados o mayores, en los cuales la cantidad de energía reflejada es mucho mayor y en un ángulo más cercano al perpendicular. Estos efectos se observan en la realidad.

El modelo de Phong-Blinn no reproduce correctamente los metales pulidos, porque dan la impresión de ser superficies plásticas. Una mejora ulterior a este aspecto fue propuesta por Cook y Torrance [21], en el cual el color de la reflexión especular depende del color del material según una función de la longitud de onda y del ángulo de incidencia. Esta función es una aproximación del factor de Fresnel, el cual depende de que la superficie reflectante sea dieléctrica o conductora. De esa manera, en el modelo de Cook y Torrance el coeficiente de reflexión especular se desdobra en un coeficiente dieléctrico (plástico) y en otro conductor (metálico). Este modelo sigue manteniéndose dentro de los límites computacionalmente aceptables, pero produce imágenes de mucho mayor realismo que con el modelo de Phong, y permite una mayor versatilidad en la representación de las propiedades de las superficies de los materiales.

Otro de los enfoques consiste en plantear directamente la solución integral del equilibrio de energías en cada punto de la escena. De esa manera se propone que cada punto x, y, z de la escena tiene una función de reflectancia bidireccional (BRDF) para

la que puede plantearse la intensidad radiada (ver Figura 7.16), denominada *ecuación del rendering* [54]:

$$I_r(x, y, z) = \int_{t=-\infty}^{+\infty} \int_{\lambda=400nm}^{700nm} \int_{\varphi=0}^{\frac{\pi}{2}} \int_{\theta=0}^{\pi} L(t, x, y, z, \varphi, \theta, \lambda) R(t, \varphi, \theta, \lambda) d\theta d\varphi d\lambda dt,$$

donde L es una función de iluminación,

R es la función de reflectancia bidireccional (BRDF) del material,

x, y, z, t coordenadas y tiempo,

φ es el ángulo azimutal,

θ es el ángulo con respecto al eje z .

La solución simultánea de la ecuación del rendering para cada punto de la escena debe entonces predecir con exactitud la condición de iluminación buscada. Tal como está planteada, esta ecuación no puede resolverse computacionalmente, ni siquiera en forma muy aproximada. Sin embargo, sirve como modelo matemático adecuado para caracterizar otros algoritmos de rendering como los que veremos en las siguientes Secciones.

7.5 Ray Tracing

En los modelos de iluminación presentados en las Secciones anteriores se llegó paulatinamente a una ecuación que contempla en forma físicamente adecuada la reflexión difusa (Lambertiana) y la reflexión especular de iluminación proveniente de fuentes de luz puntual. Sin embargo, la interreflexión de luz entre objetos de la escena quedó representada en un término de iluminación ambiente, único para todos los objetos de la escena, lo cual es indudablemente insatisfactorio para expresar un modelo de iluminación global. Fue en 1980 cuando Whitted [85] propuso un modelo de iluminación recursivo que integra reflexión, refracción, cara oculta y sombras en un único procedimiento.

El modelo más simple consiste en trazar rayos (de ahí su nombre) a través de los caminos de reflexión y refracción entre los objetos de la escena. Cada rayo se traza desde el observador, pasando a través de un pixel de la pantalla, hasta encontrar el primer objeto de la escena. Si el rayo no intersecta ningún objeto (Figura 7.17(a)), entonces se pinta el pixel de un color de fondo. Cuando intersecta un objeto (Figura 7.17(b)), se plantea un modelo de iluminación local (por ejemplo, Phong) para determinar el color, pero se computa además un rayo hacia la fuente de luz (Figura 7.17(c)), para garantizar que otros objetos de la escena no estén “haciendo sombra” sobre el punto de la escena que estamos iluminando.

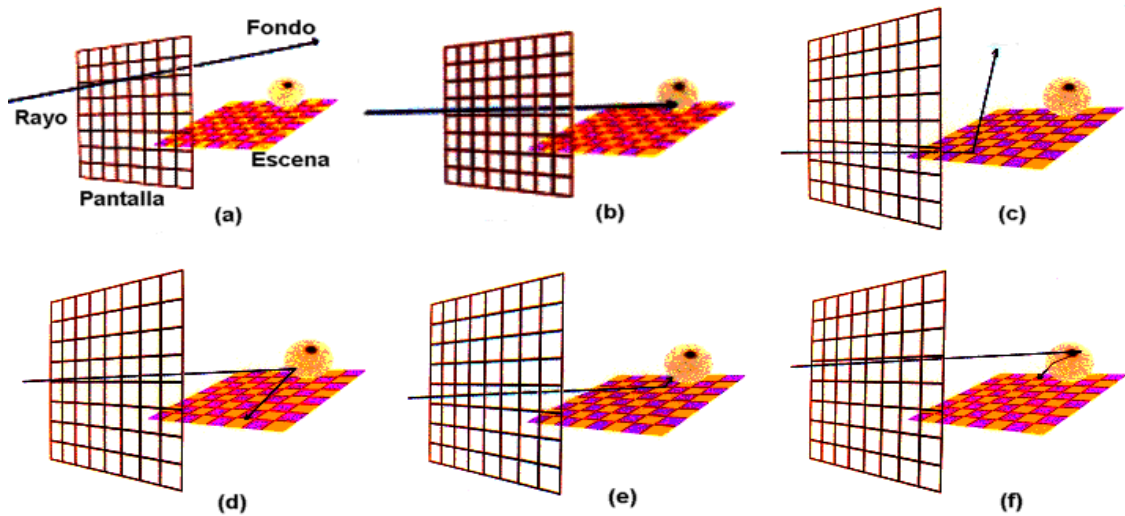


Figura 7.17 Distintas posibilidades en el cómputo de un rayo.

Si los objetos de la escena tienen, además, un coeficiente de reflexión especular o de refracción, entonces es necesario computar un factor proveniente de los rayos idealmente reflejados y refractados. La intensidad de cada uno de dichos rayos es computada recursivamente, para lo cual se envían nuevos rayos que experimentan el mismo procedimiento (Figura 7.17(d), (e) y (f)), lo cual configura un árbol de rayos, cada uno de los cuales ejerce una influencia en el color del pixel (ver Figura 7.18).

El costo computacional del trazado de rayos es muy alto, porque para cada pixel de la pantalla es necesario elaborar todo un árbol completo de iluminación. Es posible disminuir el costo limitando la profundidad de la recursión. Por ejemplo, en la Figura 7.20 podemos ver una escena compuesta por tres esferas rodeadas por tres espejos. Trazando los árboles de ray tracing con profundidad 1 se computa solamente el modelo de iluminación local, por lo que los espejos aparecen como superficies oscuras y pulidas. Con profundidad 2, aparecen las reflexiones directas de las esferas en los espejos del costado, y con profundidad 3 aparecen las reflexiones (indirectas) de dichas reflexiones en el espejo de abajo.

Los rayos se consideran infinitesimalmente delgados, y la reflexión y refracción son ideales, sin dispersión, como si se tratara de superficies idealmente suaves. Por dicha razón, las imágenes sintetizadas con ray tracing tienen una característica especial, siempre se trata de escenas con objetos geoméricamente simples y brillantes que exhiben reflexiones múltiples de una agudeza levemente sobrenatural (ver Figura 7.21).

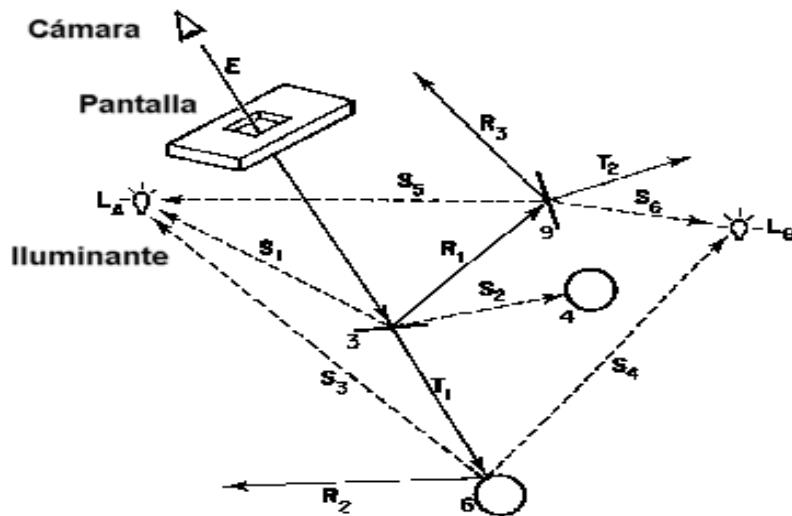


Figura 7.18 Árbol de ray tracing.

Un prototipo de trazador de rayos recursivos sigue aproximadamente los siguientes pasos. Dado un punto p de donde parte el rayo, en una dirección d , y una profundidad de recursión r , se deben encontrar todos los objetos de la escena que intersectan al rayo, calcular el punto de intersección en cada caso, y elegir el más cercano de los puntos de intersección. En dicho punto se encuentra un modelo de iluminación local, y se calculan las direcciones del rayo reflejado y del rayo refractado. Luego se computa recursivamente un ray tracing sobre dichos rayos (con un nivel menos de recursión), y con el color resultante de los dos rayos más el color resultante del modelo de iluminación local se computa el color final del pixel (ver Figura 7.19).

Pese al costo computacional, es una característica muy atractiva que se combinen tantos efectos de iluminación en un único algoritmo. Por dicha razón es que el tema se ha desarrollado de una manera impresionante en los últimos 10 años. En particular, podemos mencionar que se han estudiado una gran cantidad de métodos para acelerar la ejecución del ray tracing, o para permitir una representación más versátil de objetos.

El control adaptativo (dependiente de la escena) del nivel de recursión es la optimización más directa. Hall y Greenberg [47] señalaron que el porcentaje de una escena que contiene superficies muy transparentes y reflexivas es, en general, pequeño y es por lo tanto ineficiente llegar con cada rayo hasta la profundidad máxima. Sugieren entonces usar un control de profundidad adaptativo que dependa de las propiedades de los materiales con los cuales intersectan los rayos, hasta una profundidad máxima pre-

```

procedure trazarrayo(p:punto;d:vector;r:integer;var col:color);
...
begin
  if d < MAXRECURS then begin                {no llego al limite}
    r:=r-1;                                   {un nivel menos de recursion}
    intersecta(p,d,objeto,intersec);          {primer objeto que intersecta}
    if objeto=nil then col:= COLORFONDO;      {no intersecta ningun objeto}
    else begin
      phong(objeto,intersec,collocal);        {ilum. local del objeto}
      reflejado(p,d,objeto,intersec,refl);    {direc. reflejado}
      refractado(p,d,objeto,intersec,refr);    {direc. refractado}
      trazarrayo(intersec,refl,r,colrefl);     {color rayo reflejado}
      trazarrayo(intersec,refl,r,colrefr);     {color rayo refractado}
      colorglobal(objeto,collocal,colrefl,colrefr,col); {modelo hibrido}
    end;
  else col.r:=col.g:=col.b:=0                 {supero recursion -> negro}
  end;
end;

```

Figura 7.19 Prototipo de un trazador de rayos.

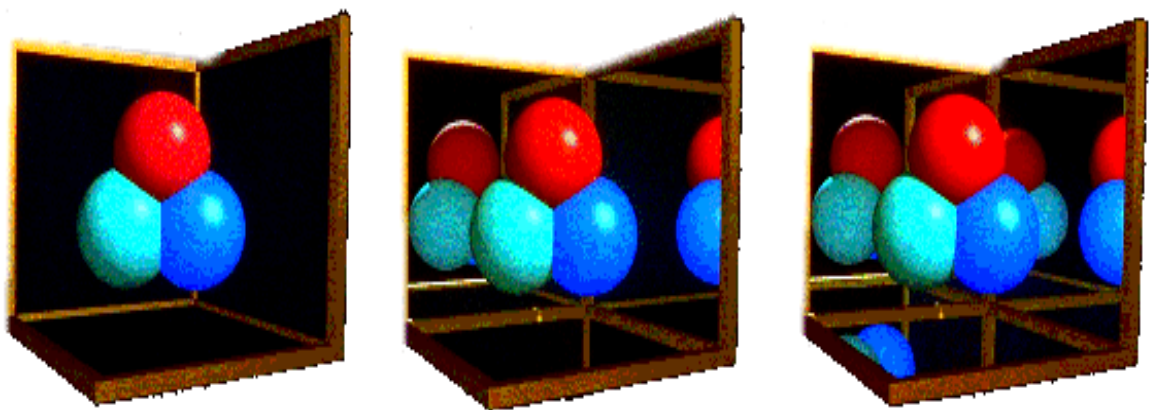


Figura 7.20 Efecto de modificar la profundidad de la recursión.

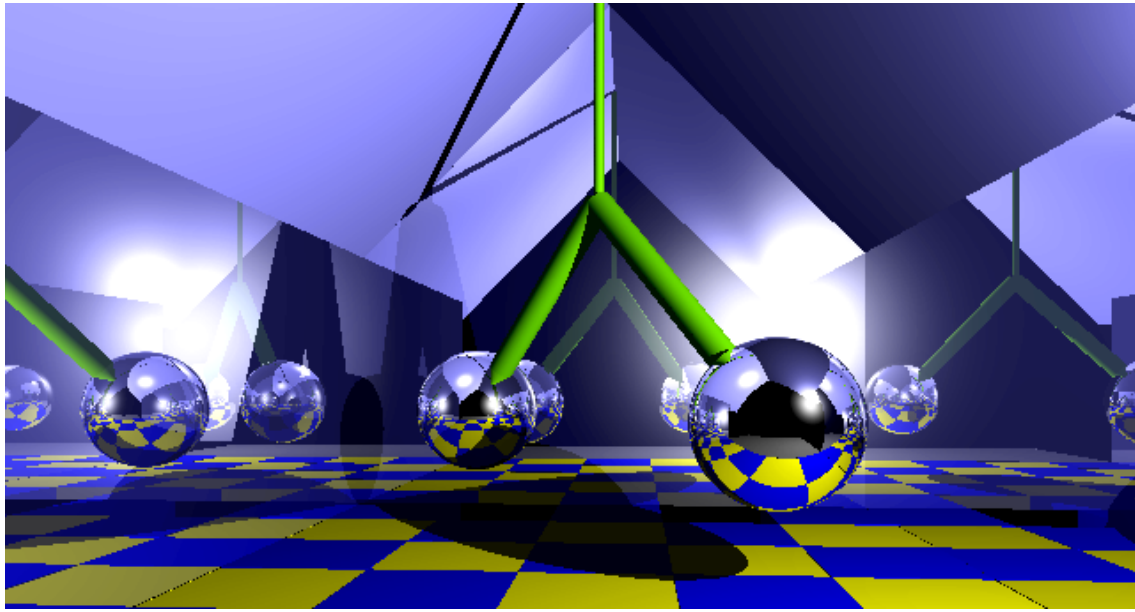


Figura 7.21 Una escena compleja elaborada con ray tracing.

fijada. Cuando un rayo se refleja en una superficie se atenúa de acuerdo al coeficiente de reflexión especular de ésta. De la misma forma, cuando es refractado es atenuado por el coeficiente de transmisión global de la superficie. La atenuación final del rayo será el producto de las atenuaciones en cada intersección. Si este valor cae por debajo de un cierto nivel no tendrá sentido seguir propagando el rayo.

Otra estrategia es reducir el número de objetos con los que debe intentarse la intersección con un rayo. De esa manera, cada objeto (especialmente los muy complejos) se consideran encerrados por un volumen sencillo, por ejemplo un cubo, con el cual se chequea la intersección. Para un objeto representado por un gran número de polígonos esto evita el cálculo de intersección con cada uno de ellos. Las esferas han sido usadas casi siempre como volúmenes debido a la simplicidad de cálculo de intersecciones entre esferas y rayos. Sin embargo su conveniencia depende mucho de la naturaleza del objeto. Por ejemplo, para objetos largos, angostos y curvados resultará en una situación donde la mayor parte de los rayos que intersectan la esfera no intersectan el objeto.

Otra estrategia es el uso de coherencia espacial. El espacio ocupado por la escena se subdivide en regiones. En lugar de intersectar el rayo con todos los objetos o conjuntos de ellos, determinamos qué regiones del espacio atraviesa el rayo y consideramos solamente los objetos que ocupan esas regiones. Esta aproximación ha sido desarrolla-

da independientemente por muchos autores. Todas las aproximaciones involucran un preprocesamiento del espacio para preparar estructuras de datos auxiliares. Entre las soluciones que se basan en la coherencia espacial encontramos:

- Dividir el espacio en una grilla fija de regiones iguales. La dificultad consiste en determinar el tamaño de las regiones y guardar la gran estructura resultante.
- Dividir el espacio en forma binaria, método conocido como BSP (Binary Space Partitioning).
- Dividir el espacio en octrees. El *octree* es una estructura de datos representada por un árbol en el que cada nodo, llamado *voxel*, tiene 8 hijos, asociados a particiones de un cubo por tres planos ortogonales que pasan por su centro. La profundidad de una rama del árbol puede variar.

También se puede considerar el trazado de rayos de un grosor dado o de un conjunto de rayos paralelos, porque en la mayoría de las escenas, muchos grupos de rayos adyacentes siguen prácticamente las mismas trayectorias. El algoritmo descrito en [50] se denomina *Beam Tracing* y está basado en este tipo de coherencia. Es indicado para escenas que solo posean superficies poligonales, ya que en ellas las reflexiones son lineales. Es esencialmente un algoritmo de superficie oculta recursivo que encuentra todos los polígonos visibles dentro de una pirámide arbitraria. Para cada pirámide de rayos, comenzando por el plano de proyección de la imagen completa, se crea una lista ordenada por profundidad de los objetos de la escena. Estos objetos se prueban uno a uno con la pirámide. En el caso que la intersecten, la pirámide se divide en la porción que hizo la intersección y la que no intersectó, que seguirá su recorrido contra los objetos restantes. Se simula reflexión y refracción lanzando recursivamente nuevas pirámides, que son deformaciones de la original por medio de transformaciones lineales.

Podemos mencionar, por último, que en la graficación de la escena es posible encontrar zonas completas de la pantalla en la cual los pixels adyacentes están fuertemente correlacionados, tienen intensidades similares e historias muy semejantes de intersección con objetos. En [1] se aprovecha esta correlación al elaborar un algoritmo que, en regiones con pocas variaciones de iluminación y con la misma historia de rayos, reduce el tiempo de cálculo lanzando solamente algunos rayos e interpolando el resto. Esta técnica se conoce como sub-muestreo. El algoritmo procede recursivamente de una manera similar al de Warnock. Se empieza calculando las intensidades en una malla de pixels separados entre sí. Luego los clasifica según la diferencia de intensidad entre rayos vecinos y sus historias de rebotes en la escena. Los pixels intermedios se calculan por interpolación.

7.6 Radiosidad

Así como el ray tracing parte del modelo de iluminación local de Phong y busca mejorar la interreflexión especular, sombreado y refracción, el método de radiosidad busca una solución global a los términos de iluminación e interreflexión difusa. La creación de imágenes con reflexión difusa requiere el cálculo de la iluminación de todas las superficies para un conjunto de posiciones y direcciones discretas del entorno que están determinadas por criterios independientes de la ubicación del observador. La discretización del entorno tridimensional (las superficies a visualizar) reduce en gran medida el número de puntos muestreados independientemente de la resolución de la imagen final. Para superficies con alto gradiente de intensidad, en los bordes de las sombras por ejemplo, no es suficiente una discretización uniforme, por lo que se usa una discretización que se ajusta adaptativamente por el gradiente de intensidad [20].

Ray tracing selecciona aspectos particulares de la interacción luz-objeto: reflexión especular y transmisión; aproxima estos y excluye otras consideraciones. Radiosidad, toma en algún sentido el punto de vista opuesto, favorece la interacción de superficies con reflexión difusa y excluye la reflexión especular. Más aún, el método tradicional de radiosidad asume comportamiento difuso ideal en todas las superficies y expresa las transferencias entre superficies por medio de un factor de forma [43]. El algoritmo de radiosidad convencional comprende las siguientes fases:

1. Computar los factores de forma: determinando los polígonos visibles (o porciones de ellos) desde cada otro polígono, usando algún artefacto auxiliar como por ejemplo un semi-cubo.
2. Resolver la ecuación de la matriz de radiosidad usando algún método numérico (por ejemplo por Gauss-Seidel). Como la matriz es diagonal dominante converge rápidamente. Se hace para cada banda de color por separado.
3. Mostrar los resultados: se determinan las superficies ocultas y se interpolan los valores de radiosidad. Para ello se puede usar z-buffer y sombreado de Gouraud respectivamente.

Las imágenes realizadas con radiosidad exhiben un tratamiento adecuado para problemas geométricos enormemente complejos, como las sombras suaves (como las que ocurren con la luz difusa), y la interreflexión difusa y el trasvasamiento de color. Estos resultados son notablemente verosímiles, no por el nivel de detalle (como en ray tracing) sino por la calidez y armonía que exhiben. Lamentablemente, radiosidad no permite un tratamiento adecuado de la reflexión especular, y hasta la fecha no se ha podido encontrar un método que combine las ventajas de ambos.

7.7 Comparación de los métodos de rendering

Podemos dedicar en este punto unos párrafos a comparar los tres grandes métodos de rendering vistos hasta ahora: scan-line, ray tracing y radiosidad.

Velocidad: Los métodos scan-line tienen siempre la ventaja con respecto a la velocidad de ejecución. El tiempo de ejecución de ray tracing puede acelerarse con alguna de las técnicas vistas en Secciones anteriores, pero siempre será un orden de magnitud mayor que scan-line. Los tiempos de radiosidad son aún más lentos, dado que el cómputo de los factores de forma y la subdivisión adaptativa involucra algoritmos de complejidad potencialmente exponencial.

Resultados: Tanto radiosidad como ray tracing producen resultados superiores a scan-line. El modelo de iluminación de radiosidad es físicamente correcto, y por lo tanto las imágenes son las más verosímiles realizadas hasta la fecha. La falta de iluminación puntual y reflexión especular es una limitación. Ray tracing, por su parte, produce resultados que impresionan pero no son siempre creíbles. El modelo de iluminación es incorrecto porque considera separadamente la componente local y la recursiva.

Algoritmos: Los algoritmos para ray tracing son sin duda los más sencillos y adaptados a una metodología de desarrollo. Scan line utiliza un modelo conceptual uniforme (la tubería de procesos), pero para determinados problemas (por ejemplo, cara oculta) este modelo se interrumpe y es necesario intercambiar información entre distintos bloques del programa, con lo cual las implementaciones se van tornando más y más desordenadas. Radiosidad necesita recurrir a algoritmos para computar el factor de forma y subdividir adaptativamente la escena, los cuales son enormemente complejos y problemáticos.

Primitivas geométricas: En scan-line es posible representar objetos de una forma arbitraria, o bien se puede trabajar con poligonizaciones. En ray tracing es indispensable contar con objetos cuya intersección con rayos sea fácilmente computable. Poligonizar objetos puede tener un costo prohibitivo porque incrementa enormemente la cantidad de tests de intersección. Radiosidad debe trabajar con poligonizaciones, por lo que en principio tampoco está limitado en la geometría de los objetos.

Aliasing: Es un problema derivado del muestreo de la escena. En scan-line, un objeto que al proyectarse se vuelve pequeño o delgado, siempre es dibujado porque su discretización encuentra por lo menos un pixel. En cambio en ray tracing puede suceder que un objeto (o parte de un objeto) pequeño se encuentre entre dos rayos trazados y por lo tanto no se dibuje.

7.8 Mapas de atributos

La síntesis de imágenes con realismo ha seguido en este Capítulo el camino de agregar a un modelo sencillo y bien conocido (aunque de base empírica) los elementos de mayor complejidad que permiten simular mejor los efectos físicos en un modelo de iluminación. De esa manera partimos de los modelos de iluminación locales, para mejorar con ray tracing la simulación de reflexiones especulares, y con Radiosidad la simulación de iluminación y reflexiones difusas. Pero otro camino para mejorar el realismo es dotar a las superficies de los objetos de propiedades y atributos que los vuelvan verosímiles, dado que hasta ahora solamente se trataban de superficies lisas, de un determinado color y rugosidad.

Las técnicas de mapeo de texturas en general constituyen una de las formas más utilizadas y de mejores resultados para realzar las cualidades de los objetos representados. De esa manera, se obtiene mayor realismo en las escenas con un costo adicional relativamente aceptable. La idea básica del mapeo de atributos consiste en mapear o proyectar un patrón bidimensional para un atributo dado sobre la superficie de un objeto. Por ejemplo, podemos tener almacenada la textura de una madera como una perturbación local de los colores (coeficientes O_d en el modelo de Phong en la ecuación 7.1). Al graficar un polígono cuya superficie debe parecer de madera, se recorre dicho patrón utilizando alguna función asociada a las coordenadas generadas durante la conversión scan del polígono, y se utiliza el coeficiente en dicha coordenada como perturbación local en el color de la superficie.

En general, los atributos que pueden mapearse a una superficie pueden ser varios:

Mapa de Texturas: Como ya describimos, consiste en alterar localmente el color de la superficie modificando los coeficientes en el modelo de reflexión difusa del objeto.

Mapa de Entorno: Consiste en almacenar en un buffer especial los objetos de la escena tal como son visibles desde la superficie del objeto que se está considerando. Dicho buffer es luego utilizado como coeficiente de iluminación especular en superficies lisas, dando la impresión de que la escena “se refleja” en ellas. Si bien el planteo es geoméricamente incorrecto [7], los resultados que produce son satisfactorios, y es actualmente una técnica muy popular.

Mapa de Desplazamientos: También denominada *bump mapping*. Se perturba la posición local de los puntos del polígono en dirección de la normal, según una distancia especificada en el patrón bidimensional. De esa manera es posible darle a un objeto una apariencia arrugada o abollada (ver Figura 5.17).

Mapa de Normales: Dado que el modelo de iluminación de Phong es muy sensible a la orientación de los normales, en muchas situaciones no es necesario desplazar la figura para que aparezca una textura espacial, sino que basta con desplazar los normales. Por ejemplo, para modelar una pared de ladrillos, además del mapa de texturas, es necesario que el modelo de iluminación genere la sensación de que en la superficie hay hendiduras.

Mapa de Transparencias: Permite modificar localmente el coeficiente de transmisión de un objeto translúcido, de manera que permite simular efectos de transparencia como por ejemplo nubes como elipsoides de transparencia aleatoriamente distribuida.

Mapa de Refracciones: Altera localmente el índice de refracción, por lo que permite producir efectos de ondas similares a los visibles en la superficie del agua.

En la Figura 7.22 podemos ver el uso de mapas para dar texturas a una escena graficada con ray tracing. Se utiliza un mapa de textura para el piso, lo cual permite simular las baldosas del piso. Además, las esferas tienen un mapa de normales que modifica localmente la reflexión especular, por lo que aparece abollada sin estarlo (no podría aplicarse un mapa de desplazamientos porque el ray tracing no puede encontrar la intersección rayo-esfera abollada).

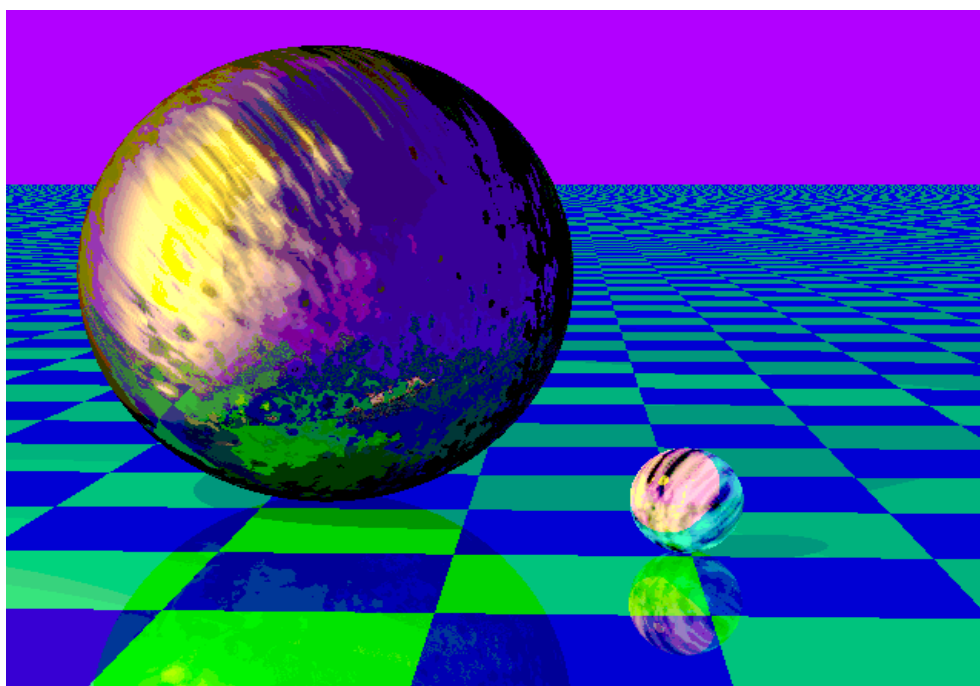


Figura 7.22 *Escena con mapas de atributos.*

7.9 Ejercicios

1. Implementar el sombreado plano de polígonos y el de Gouraud. Testarlos con valores cualquiera de iluminación en los vértices.
2. Generar una escena con iluminantes, encontrar los valores de iluminación por medio del modelo de Bouknight (ambiente+Lambertiana) y sombreadar con la técnica de Gouraud.
3. Repetir el ejercicio anterior con el modelo de iluminación de Phong y el sombreado de Phong. Comparar tiempos y resultados.
4. Implementar el Ray Tracing recursivo y graficar una escena sencilla en la que ocurran interreflexiones.
5. Agregar a los ejercicios 3 y 4 un mapa de texturas y de normales.
6. Discutir los pasos necesarios para encontrar un mapa de entorno (pauta: utilizar las caras de un cubo que rodea al objeto).

7.10 Bibliografía recomendada

La mayor parte de las referencias originales a los trabajos se fueron dando a lo largo del Capítulo. Sin embargo, como las mejoras en los modelos han sido objeto de estudio intensivo, siempre es preferible recurrir a un texto moderno que presente los resultados en forma ordenada y con una notación sistemática. Pocos son los libros en los que se aborda este tema como único asunto. Entre ellos recomendamos leer el libro de Hall [46]. Recomendamos consultar también [9, 54, 48] para mayores detalles.

Los modelos básicos de iluminación y sombreado se pueden consultar en el Capítulo 25 del libro de Newman-Sproull [66] y en el Capítulo 16 del libro de Foley et. al. [33]. Los lectores interesados en una presentación rigurosa y concisa pueden consultar el Capítulo 2 del libro de Watt y Watt [84]. Los algoritmos de Ray Tracing han generado una copiosa bibliografía. Es preferible acercarse a través de un enfoque gradual, como el presentado en el libro de Glassner [41], aunque la introducción al tema en los Capítulos 8, 9 y 10 del libro de Watt y Watt es bastante satisfactoria. Una presentación de los detalles de implementación de radiosidad puede consultarse en [20] y en el Capítulo 11 del libro de Watt y Watt. Las técnicas de mapeo de atributos están descriptas en varios artículos. Aquí recomendamos consultar [44, 51].