Tema IV

Ray casting avanzado

Ricardo Ramos

Colaboradores: Joaquin Villanueva Balsera, Mª del Carmen Suárez Torrente, Mª Sandra García Peláez, Francisco J. Barroso Alfambra y Mª Félix de la Corte Martínez

Hemos visto en el tema anterior las características de un proceso de ray casting básico. Toca ahora estudiar otro algoritmo de intersección importante, concretamente el de intersección rayo-polígono, y mejorar la calidad de las imágenes generadas por el algoritmo de ray casting.

1.1 Introducción

Veamos las cuestiones que quedaron pendientes en el tema anterior, al no ser utilizadas por el algoritmo simplificado de ray casting.

Para su estudio, organizaremos el tema de forma similar al anterior, es decir, en función de las fases del algoritmo. Por tanto, ampliaremos la primera fase viendo un nuevo e importante algoritmo de intersección rayo-objeto. En la segunda fase se estudiará lo relacionado con los **modelos de fuentes emisoras**, **sombreado** y nuevos modelos de color. Por último, ampliaremos la tercera fase viendo el modelo de intensidad de **Cook y Torrance**, que basa el cálculo de la componente especular en conceptos físicos.

1.2 Primera fase: ¿Cuál es el punto de procedencia?

1.2.1 Intersección rayo-polígono

El cálculo del punto de intersección entre una recta y un polígono se realiza en tres pasos, claramente diferenciados:

- Definir el plano al que pertenece el polígono.
- Buscar la intersección entre el rayo y el plano.
- Verificar si el punto de intersección rayo-plano pertenece al polígono.

Los dos primeros pasos son evidentes: *si el rayo interseca con el polígono, necesariamente ha de hacerlo con el plano que lo contiene.*

El último paso es necesario ya que *no todos los rayos que intersecan el plano intersecan también el polígono*, como muestra la figura 1.

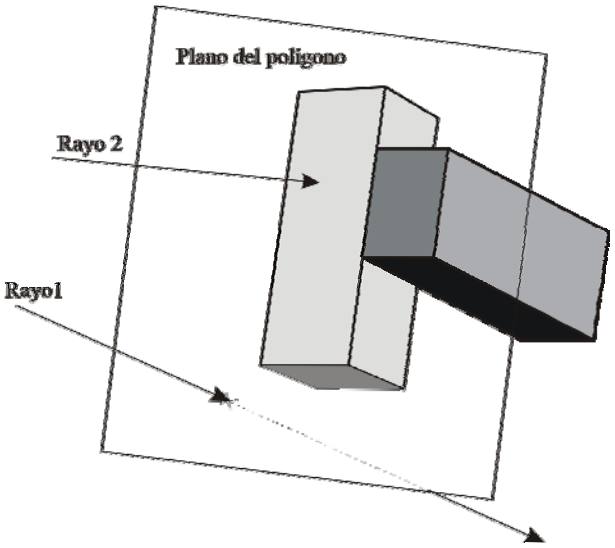

figura 1: no todos los rayos que intersecan el plano lo hacen con el polígono

Veamos primero cómo se define el plano al que pertenece un polígono.

1.2.2 Definición del plano

A la hora de definir los polígonos, la información geométrica más significativa que puede registrarse de ellos son sus vértices. Entonces, dado un polígono, tomando tres de sus vértices (P_0, P_1 y P_2) es posible encontrar los vectores $V_1 = P_1 - P_0$ y $V_2 = P_2 - P_0$, como muestra la figura 2.

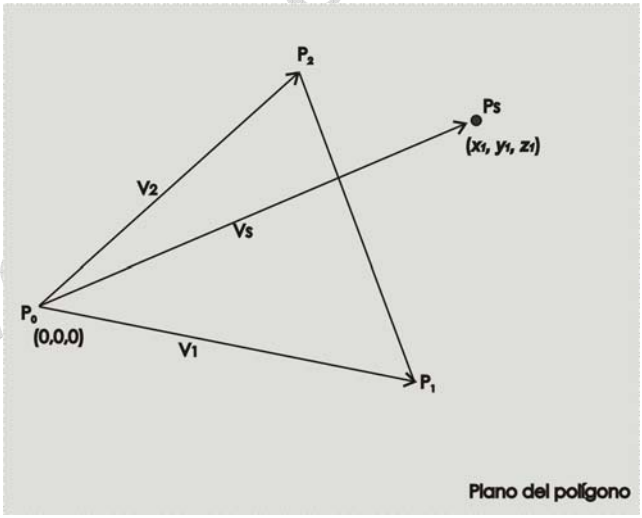

figura 2: definición del plano

Si se realiza el producto vectorial $\mathbf{V_1}$ x $\mathbf{V_2}$, se obtiene un vector perpendicular al polígono, y por tanto al plano que lo contiene. *Suponiendo que* P_0 *se encuentra en el origen de un sistema de coordenadas local 3D* (P_0 = (0, 0, 0)), y siendo $P_1 = (P_{1x}, P_{1y}, P_{1z})$ y $P_2 = (P_{2x}, P_{2y}, P_{2z})$ las coordenadas de los otros vértices del polígono en dicho sistema, podemos expresar el producto vectorial de $\mathbf{V_1}$ y $\mathbf{V_2}$ como:

[1]

$$\mathbf{V_p} = \mathbf{V_1} \times \mathbf{V_2}$$

siendo

$$\mathbf{V_p}(V_{px}, V_{py}, V_{pz}) = (P_{1y}P_{2z} - P_{1z}P_{2y},\ P_{1z}P_{2x} - P_{1x}P_{2z},\ P_{1x}P_{2y} - P_{1y}P_{2x})$$

De cara al algoritmo de intersección, conviene que el vector normal al plano sea un vector unidad, es decir, que se encuentre normalizado. Como es más que probable que $\mathbf{V_p}$ no quede normalizado al multiplicar vectorialmente, se procede como sigue:

Siendo $\left|V_p\right| = \sqrt{V_{px}^2 + V_{py}^2 + V_{pz}^2}$, las coordenadas (A, B, C) del vector normal al plano ($\mathbf{V_n}$), vienen dadas en el sistema local por:

[2]

$$A = \frac{V_{px}}{\left|V_p\right|},\ B = \frac{V_{py}}{\left|V_p\right|},\ C = \frac{V_{pz}}{\left|V_p\right|}$$

siendo $A^2 + B^2 + C^2 = 1$.

Resumiendo,

[3]

$$\mathbf{V_{normal}} = \mathbf{V_n} = [A, B, C]$$

Por otro lado, *el producto escalar entre* $\mathbf{V_n}$ *y cualquier otro vector del plano, dará como resultado 0 ya que son perpendiculares*. Por ejemplo, supongamos un punto del plano $(P_s) = (x_1, y_1, z_1)$, con coordenadas dadas en el sistema de referencia local, como muestra la figura 2. Si se multiplican escalarmente los vectores $\mathbf{V_n}$ y $\mathbf{V_s}$, ha de ocurrir que

[4]

$$\mathbf{V_n} \cdot \mathbf{V_s} = |V_n|\,|V_s| \cos\theta = 0$$

ya que cos 90°= 0.

Dado que ambos vectores parten del origen del sistema de coordenadas local, podemos expresar su producto escalar en función de las coordenadas de ambos. Queda entonces que

[5]

$$\mathbf{V_n} \cdot \mathbf{V_s} = Ax_1 + By_1 + Cz_1$$

Igualando [4] y [5] queda

[6]

$$\mathbf{V_n} \cdot \mathbf{V_s} = Ax_1 + By_1 + Cz_1 = 0$$

Como $\mathbf{V_s}$ puede ser cualquier punto del plano, *la expresión anterior ha de ser cierta para todos ellos*. Por tanto, para todos los puntos (*x, y, z*) de un plano *que pase por el origen* de coordenadas se ha de cumplir que

[7]

$$Ax + By + Cz = 0$$

Dado que normalmente los planos a los que pertenecen los polígonos *no van a pasar por el centro de coordenadas* del Sistema de Referencia Universal (donde quedan ubicados los objetos para su visualización), es necesario generalizar la ecuación [7] de forma que sea válida para cualquier plano. Para ello, se supone que $P_0 = (a, b, c)$, es decir, que no se encuentra necesariamente en el origen de coordenadas. Operando de modo similar a como hemos visto, se llaga a la ecuación general del plano

[8]

$$Ax + By + Cz + D = 0$$

siendo "D" el módulo del vector que pasa por el origen del SUR, y que es normal al plano; en otras palabras, la distancia más corta desde el origen de coordenadas hasta el plano. El signo de D indica qué cara del plano está orientada hacia el origen del sistema de coordenadas.

La ecuación implícita del plano, al igual que la de la esfera, no proporciona las coordenadas de los puntos del plano sino que, dado un punto, permite averiguar si pertenece a él o no.

Para calcular D, como se conocen las componentes A, B y C de $\mathbf{V_n}$ y las coordenadas de tres (o más) puntos que cumplen la ecuación [8] (P_0, P_1 y P_2), basta con sustituir cualquiera de ellos en la ecuación, y despejar D. Por ejemplo, si elegimos $P_0 = (a, b, c)$, entonces

$$D = -(Aa + Bb + Cc)$$

1.2.3 Intersección rayo-plano

Continuando con el algoritmo de intersección rayo-polígono, en el siguiente paso hay que encontrar el punto de intersección entre el rayo y el plano. Para ello, se ha de ver primero cómo queda definido el rayo.

La recta (rayo) va a quedar definida de modo similar a como vimos en el algoritmo de intersección rayo-esfera. Así,

$$R_0 = [x_0, y_0, z_0]: \text{punto de origen del rayo}$$

$$R_d = [X_d, Y_d, Z_d]: \text{vector de dirección del rayo}$$

con $X_d^2 + Y_d^2 + Z_d^2 = 1$, es decir R_d se encuentra normalizado.

Las coordenadas de todo punto [X, Y, Z] del rayo se pueden expresar en función de "t" como sigue:

[9]

$$X = X_0 + X_d \cdot t$$

$$Y = Y_0 + Y_d \cdot t$$

$$Z = Z_0 + Z_d \cdot t$$

dando valores creciente a "t", para t > 0.

Las expresiones anteriores se pueden escribir en formato vectorial, obteniéndose la **ecuación explícita del rayo,** es decir,

[10]

$$\mathbf{R(t)} = \mathbf{R_0} + \mathbf{R_d} * t\,, \qquad \text{siendo } t > 0$$

A) Cálculo del punto de intersección

Si el rayo cruza el plano, como muestra el caso de la figura 3, *ha de existir un valor real y positivo de "t"* que, aplicado en las ecuaciones [9], nos proporcione las coordenadas del punto de intersección rayo-plano.

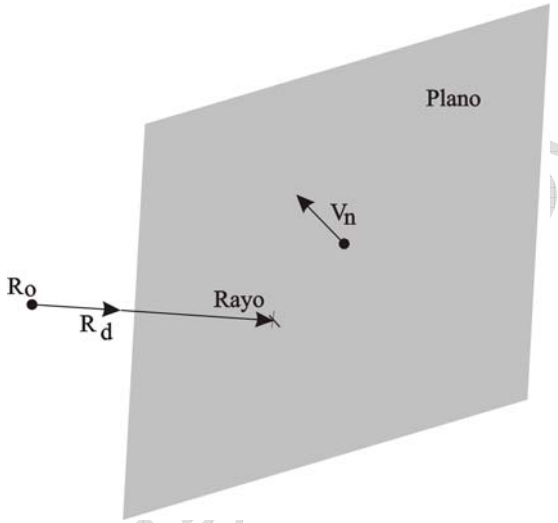

figura 3: el rayo interseca el plano

Dado que el punto de intersección pertenece tanto al rayo como al plano, se ha de verificar que

[11]

$$A \cdot (x_0 + X_d \cdot t) + B \cdot (y_0 + Y_d \cdot t) + C \cdot (z_0 + Z_d \cdot t) + D = 0$$

Despejando t obtendremos

[12]

$$t = \frac{-(A \cdot x_0 + B \cdot y_0 + C \cdot z_0 + D)}{A \cdot X_d + B \cdot Y_d + C \cdot Z_d}$$

que en notación vectorial queda

[13]

$$t = \frac{-(\mathbf{V_n} \cdot \mathbf{R_0} + D)}{\mathbf{V_n} \cdot \mathbf{R_d}}$$

En el numerador tenemos el producto escalar de la normal al plano y el origen del rayo y en el denominador, el producto escalar de la normal al plano y el vector de dirección del rayo.

Para hacer un cálculo más eficiente de la ecuación [12] comenzamos primero por calcular el producto escalar del denominador

[14]

$$v_d = \mathbf{V_n} \cdot \mathbf{R_d} = A \cdot X_d + B \cdot Y_d + C \cdot Z_d$$

Dependiendo del valor que tome v_d (menor, igual o mayor que cero), estaremos ante diferentes situaciones:

* **Si $v_d = 0$**, el rayo es paralelo al plano (forma un ángulo de 90° con la normal) y por tanto *no existirá intersección entre ambos*. Al ser paralelos, puede ocurrir que el rayo avance a lo largo del plano (figura 4), aunque este caso no tiene importancia ya que, en la visualización de los polígonos, no se prevé la posibilidad de que éstos aparezcan de canto.

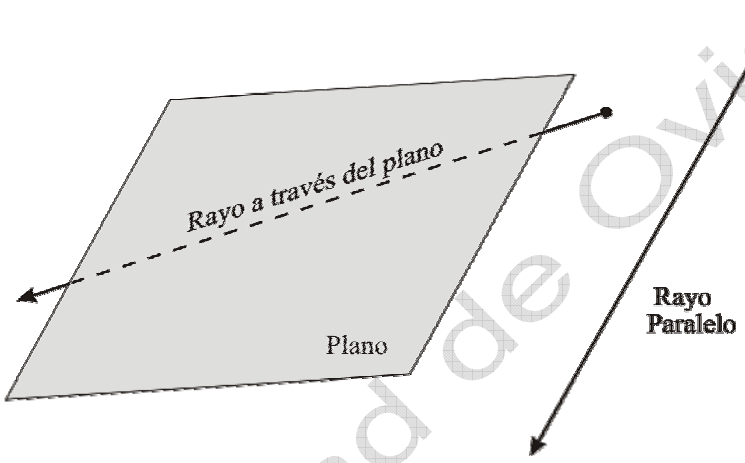

figura 4: casos en que v_d es cero y por tanto el rayo paralelo al plano.

* **Si $v_d > 0$**, *el rayo apunta en la misma dirección que la normal al plano*, como puede verse gráficamente en la figura 5. Si los objetos han sido modelados utilizando una sola cara de los polígonos, *se puede considerar que no se produce intersección rayo-polígono*, ya que el rayo incide sobre la cara interior (no visible) del polígono. En cambio, si se ha decidido que las dos caras de los polígonos sean visibles, lo que se hace es cambiar el signo de la normal, para que apunte en sentido contrario al del rayo. Queda pues que:

Si $V_d > 0$, $\mathbf{V'_n} = -\mathbf{V_n}$, sino $\mathbf{V'_n} = \mathbf{V_n}$

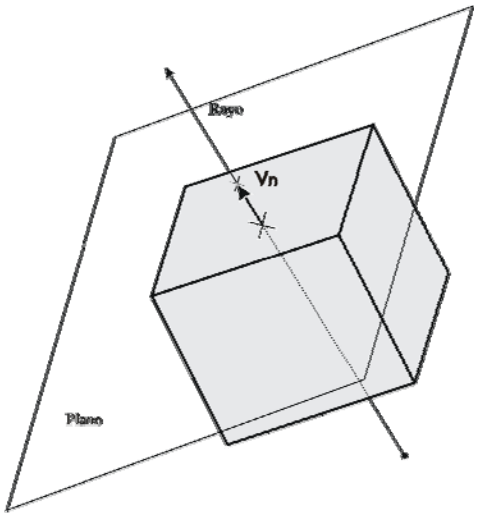

figura 5: Normal al plano apunta en la misma dirección que el plano.

* Por último, **si** v_d **< 0**, se procede a calcular el numerador de [12], para continuar verificando las condiciones de intersección rayo-plano. Tenemos pues que:

[15]

$$v_0 = -(\mathbf{V_n} \cdot \mathbf{R_0} + D) = -(Ax_0 + By_0 + Cz_0 + D)$$

Una vez que se conoce v_0, se halla el valor de "t". Según [13], queda que

$$t = \frac{v_0}{v_d}$$

En la relación anterior, **si t < 0**, la trayectoria del rayo interseca con el plano detrás del origen del rayo, es decir, éste nunca llega a intersecar con el plano (ver figura 6).

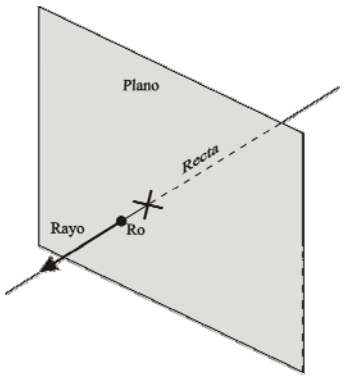

figura 6: caso en que el origen del rayo esta delante del plano.

En caso contrario, se procede a calcular el punto de intersección entre el plano y el rayo:

[16]

$$r_i = [x_i, y_i, z_i] = [x_0 + X_d \cdot t,\ y_0 + Y_d \cdot t,\ z_0 + Z_d \cdot t]$$

B) Algoritmo de intersección

Veamos cómo queda el pseudocódigo del algoritmo de intersección rayo-plano:

```
interseccion_rayo_plano()
{
   V'n = Vn
   - Calcular Vd
   Si (Vd = 0) /* rayo paralelo al plano */
      - devolver(falso)
   sino
   {
      Si (Vd > 0)
      {
         Si (doble cara) /* Se cambia el signo a la normal */
            V'n = -Vn
         sino
            - devolver(falso)
      }
      - Calcular t /* Utilizando Vn */
      Si (t < 0) /* Origen del rayo detrás del plano */
         - devolver(falso)
      sino
      {
         - Calcular Vi /* Se calcula el punto de intersección */
         - devolver(verdadero)
      }
   }
} /* Fin interseccion_rayo_plano() */
```

Al finalizar el algoritmo, en V'_n queda la normal al plano que ha de utilizarse en el modelo de intensidad. Según [Glas89, pg. 52], el algoritmo anterior requiere, en el peor de los casos, 8 sumas/restas, 9 multiplicaciones y 3 comparaciones. Veamos un ejemplo numérico, tomado del mismo libro, para entender mejor este algoritmo.

C) Ejemplo numérico sobre el algoritmo de intersección rayo-plano

Dado un plano [1, 0, 0, -7] y un rayo cuyo origen está en [2, 3, 4] y su dirección [0.577, 0.577, 0.577], se ha de encontrar la intersección del rayo con el plano. Se supone que el plano tiene doble cara.

* Como indica el algoritmo, el primer cálculo es v_d. Entonces, como

$v_d = \mathbf{V_n} \cdot \mathbf{R_d} = A \cdot X_d + B \cdot Y_d + C \cdot Z_d$, queda:

$$v_d = 1 \cdot 0.577 + 0 \cdot 0.577 + 0 \cdot 0.577 = 0.577$$

* Vemos que $V_d > 0 \Rightarrow$ se ha de cambiar el signo a la normal.

Como V_n= (1, 0, 0), la nueva normal será V'_n= [–1, 0, 0]

* A continuación se calcula el valor de v_0, utilizando V_n.

Como $v_0 = -(\mathbf{V_n} \cdot \mathbf{R_0} + D) = -(Ax_0 + By_0 + Cz_0 + D)$, queda

$$v_0 = -(1 \cdot 2 + 0 \cdot 3 + 0 \cdot 4 + (-7)) = 5$$

* Ahora se calcula t:

$$t = \frac{5}{0.577} = 8.66$$

Como t es positivo, representa la distancia (o el tiempo) desde el origen del rayo a la intersección.

* Se calculan las coordenadas del punto de intersección

$$x = 2 + 0.577 \cdot 8.66 = 7$$

$$y = 3 + 0.577 \cdot 8.66 = 8$$

$$z = 4 + 0.577 \cdot 8.66 = 9$$

por tanto el punto de intersección es $r_i = [7, 8, 9]$ y $V'_n = [-1, 0, 0]$

La figura 7 muestra gráficamente una aproximación al caso del ejemplo anterior, con el origen del rayo detrás del plano, por lo que es necesario cambiar la normal al plano para que la visualización se haga de forma correcta.

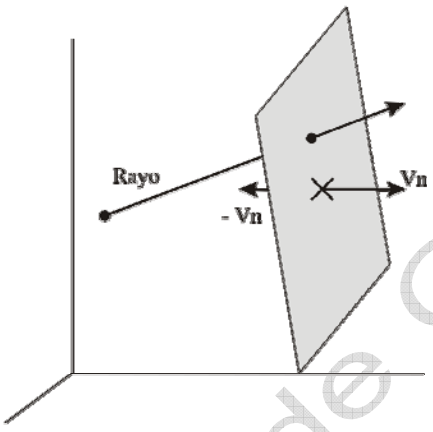

figura 7: intersección de la cara interior

1.2.4 Intersección con el polígono

Acabamos de ver cómo se calcula el punto de intersección entre el rayo y el plano. El próximo paso consiste en verificar si el punto se encuentra dentro o fuera del polígono.

Existen varias formas de conocer si el punto de intersección entre el plano y el rayo se encuentra dentro del polígono. Un método conceptualmente simple, aunque costoso en tiempo de cálculo, consiste en sumar los ángulos que forman los vectores que van desde el punto de intersección r_i a cada vértice del polígono, como muestra la figura 8.

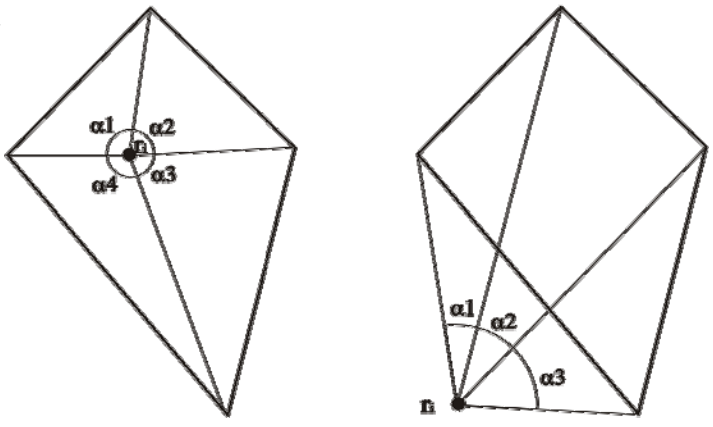

figura 8: test de intersección con el polígono

Si la suma de los ángulos es igual a 360º indica que el punto de inter-

sección r_i está dentro del polígono. En caso contrario, se encuentra fuera del polígono.

Existen métodos más económicos para averiguar si un punto se encuentra en el interior o fuera de un polígono basados en el *teorema de la curva de Jordan*. Éste viene a decir que:

Si un segmento corta a un polígono un numero par de veces entonces el origen de segmento se encuentra fuera; en cambio, si lo cruza un numero impar de veces, el origen del segmento esta dentro del polígono.

El ejemplo de la figura 9 muestra claramente el teorema anterior. Desde luego, pueden darse situaciones bastante más complicadas que las mostradas aquí, pero en todos los casos el teorema anterior es cierto.

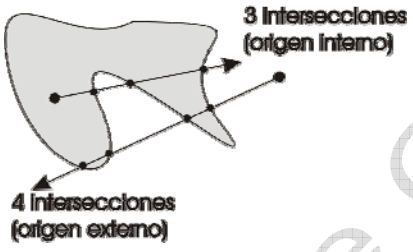

figura 9: test basado en el *teorema de la curva de Jordan*

Veamos en detalle uno de los métodos basados en el teorema de Jordan, el cual podemos encontrarlo en [Glas89, pg. 53].

* Supongamos que tenemos un polígono definido como un conjunto de N vértices, es decir,

$$\text{polígono} \equiv G_n = [x_n\ y_n\ z_n], \text{ siendo } n = 0, 1, \ldots N\text{-}1$$

Por otro lado está el plano, el cual queda definido por tres vértices del polígono. Su ecuación será

[17]

$$\text{plano} \equiv Ax + By + Cz + D$$

y por tanto, la normal viene dada por

$$\mathbf{V}_n = [A, B, C]$$

Por último, el punto de intersección del rayo con el plano está dado por

$$r_i = [x_i,\ y_i,\ z_i]$$

A partir de los datos anteriores, para realizar el test de pertenencia al polígono aplicando el teorema de la curva de Jordan, lo primero que se ha de hacer es proyectar el polígono, situado en 3D, sobre un plano; a partir de dicha proyección será fácil averiguar el total de intersecciones entre un segmento, con origen en el punto de intersección (r_i), y el polígono. Para efectuar la proyección del polígono sobre un plano, nada mejor que utilizar cualquiera de los planos ortogonales del sistema de referencia.

En definitiva, para cada vértice $[x_i, y_i, z_i]$ del polígono lo que se ha de conseguir son las coordenadas (U, V), en uno de los planos ortogonales. Además, también será preciso conocer las coordenadas (U, V) correspondientes al punto de intersección.

Una manera de conseguir la parejas (U, V) sería similar a la mostrada en la figura 10, que consiste en girar el polígono hasta lograr que la normal quede paralela a alguno de los ejes. Una vez conseguido esto, los ejes restantes podrán ser utilizados para encontrar los pares (U, V).

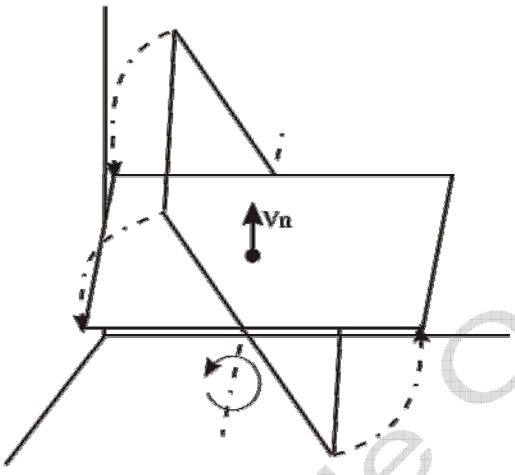

figura 10: proyección costosa del polígono

El problema del método anterior está en que el tiempo de cálculo para girar los polígonos es elevado, por lo que no es aconsejable.

Una elegante solución al problema anterior consiste en efectuar la proyección paralela del polígono, sin efectuar giro alguno, procurando simplemente elegir el plano apropiado. Operando de esta forma*, el polígono proyectado y el que se proyecta serán diferentes* en la forma y el área (ver la figura 11), a no ser que los planos de ambos polígonos sean paralelos (caso anterior). Sin embargo, *la topología de ambos polígonos será la misma*, lo que significa que si el punto de intersección se encuentra dentro o fuera del polígono que se proyecta, entonces *también se encontrará dentro o fuera del polígono proyectado*. Por lo tanto, dado que lo único que interesa conocer es la ubicación del punto de intersección con respecto al polígono, el polígono proyectado es tan válido para aplicar el test de pertenencia como el polígono que se proyecta.

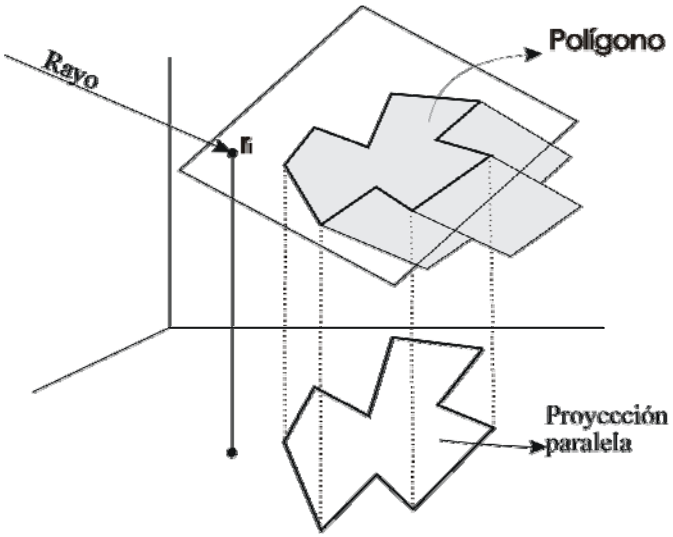

figura 11: proyección que no conserva las proporciones

Para realizar la proyección paralela de un polígono sobre un plano ortogonal, simplemente se ha de eliminar una de las coordenadas en cada vértice. En este caso, para decidir qué coordenada es la que se ha de quitar, se ha de comprobar cuál de las componentes (A, B o C) de la normal al polígono tiene mayor valor absoluto (coordenada dominante): *El eje ortogonal que corresponda a la coordenada dominante de* $\boldsymbol{V_n}$ *es el que se ha de eliminar en las coordenadas de los vértices*. Por ejemplo, si V_n = (-10, 2, 6), el eje que desaparece será el X, siendo por tanto (U, V) = (Y, Z), es decir, el polígono proyectado estará formado por N vértices, de forma que

$$Gp_n = [y_n, z_n] = [u_n, v_n], \text{ con } n = 0, 1, ..., N-1$$

Al eliminar la coordenada de mayor magnitud se evitan las proyecciones sobre planos que sean perpendiculares a los polígonos, ya que sobre ellos la proyección sería una línea; en otras palabras, de esta forma se garantiza que los polígonos proyectados posean el área suficiente para realizar sin problemas el test de pertenencia.

Una vez que tenemos el polígono Gp en el sistema UV (como se puede ver en la figura 12-a) es necesario trasladar el polígono y el punto de intersección de forma que éste quede situado sobre el origen de coordenadas del sistema UV, lo que equivale a restar las coordenadas del punto de intersección de cada uno de los vértices del polígono. El resultado será el que se puede ver en la figura 12-b.

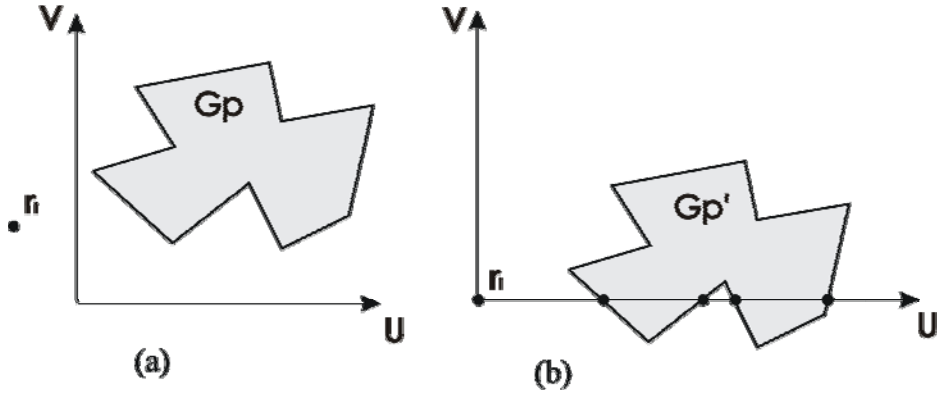

figura 12: (a) Proyección en el sistema UV. (b) Translación de r_i hacia el origen.

Para evitar confusiones, al polígono Gp trasladado lo definiremos como

$$Gp'_n = [u'_n, v'_n], \text{ con } n = 0, 1, ..., N-1$$

Cuando se encuentra el punto de intersección (r_i) en el origen de UV, por cada arista del polígono proyectado (Gp'_n) se comprueba si el eje U interseca con ella, anotando las aristas que son cruzadas por dicho eje. Como para verificar empíricamente el teorema de Jordan sirve cualquier recta (sea cual sea su dirección), el eje U es tan válido como cualquier otra recta para realizar esta prueba. Al finalizar con todas las aristas del polígono sabremos si el punto de intersección se encuentra fuera o dentro del polígono, según sea par o impar el numero de cruces (figura 12-b).

En el proceso anterior, si algún vértice se encuentra justo en el eje U puede haber problemas, como el que se muestra en la figura 13.

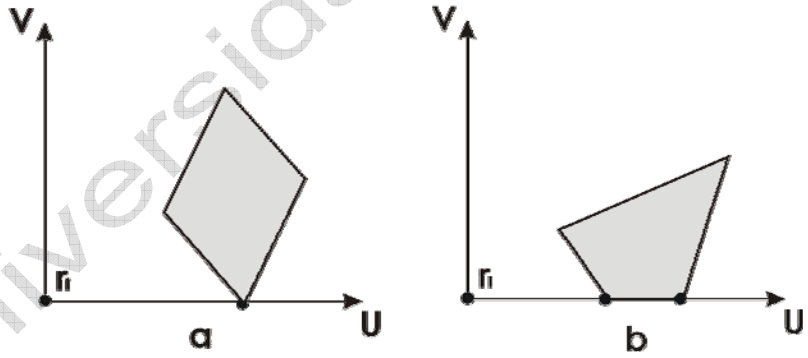

figura 13: caso en el que algún vértice se encuentra en el eje U

Vemos en este ejemplo (figura a) que si contabiliza el paso por el vértice como una intersección, se llegaría a la conclusión errónea de que el punto de intersección se encuentra dentro del polígono. En la figura b, al encontrarse una de las aristas del polígono en el eje U, pueden surgir problemas a la hora de contabilizar el número de intersecciones del eje U con dicha arista.

Existen varias posibilidades para solucionar este problema, aunque posiblemente la más efectiva sea la apuntada por Eric Haines [Glas89, pg. 56], la cual simplemente consiste en que, de cara a la realización de los cálculos,

no permitir que existan vértices en el eje U. Para ello, dado que el eje U divide el plano en dos zonas (V+ y V–), si un vértice cae en el eje U (o sea, V = 0), en los cálculos se le considera ubicado en la zona V+ del plano.

A) Test de pertenencia al polígono

Veamos finalmente cómo es el algoritmo que indica si el punto de intersección pertenece o no al polígono.

Sean $[x_n\ y_n\ z_n]$ las coordenadas de los vértices que definen el polígono que se proyecta y $[u_n, v_n]$ los vértices del polígono proyectado sobre el plano de coordenada dominante, con n = 0, 1, ..., N-1 en ambos casos. Una vez realizada la traslación de los vértices $[u_n, v_n]$ hacia el origen de coordenadas de UV, las nuevas coordenadas quedarán indicadas por $[u'_n, v'_n]$.

Para una arista cualquiera del polígono proyectado y trasladado, $P_a = [u'_a, v'_a]$ serán las coordenadas del vértice inicial y $P_b = [u'_b, v'_b]$ las del vértice final.

Según lo anterior, veamos cómo queda el algoritmo

```
test_pertenencia()
{
   para(i = 0 hasta i < N)
   {
      /* Se elimina del vértice "i" el eje correspondiente a la coor-
      denada dominante. Se obtiene (u_i, v_i) */
      proyectar_vertice(i)
      /* Se traslada el vértice hacia el origen de UV:(u_i, v_i) - r_i */
      trasladar_origen(i)
   }
   /* Se inicializa el contador de cruces */
   cruces = 0
   para(cada P_a, P_b) /* Para cada arista del polígono */
   {
      /* Se comparan los signos de v'_a y v'_b para averiguar si la
      arista cruza el eje U. Si son opuestos cruza el eje U */
      si(signo(v'_a) ≠ signo(v'_b))
         /* Se comprueba si el cruce está en U+ */
         si((u'_a > 0) y (u'_b > 0))
            cruces++
         sino
            si((u'_a > 0) o (u'_b > 0))
               si(u'_a - v'_a·(u'_b - u'_a)/(v'_b - v'_a) > 0)
                  cruces++
   }
   si(cruces es impar)
      la intersección está dentro del polígono
   sino
      la intersección está fuera del polígono
}/* Fin test_pertenencia() */
```

La primera comprobación que efectúa el algoritmo es ver si la arista cruza el eje U; esto se sabe comparando los signos de v'_a y v'_b ya que si son opuestos, necesariamente ha de cruzar U (figura 14-a). Si la arista no cruza el eje U puede ser ignorada, ya que no influirá en el número de cruces final.

Que la arista cruce el eje U no significa que el rayo (segmento) que se traza desde r_i cruce con ella, puesto que podría ser que el cruce entre U y la

arista se produzca en U–, como muestra la figura 14-b

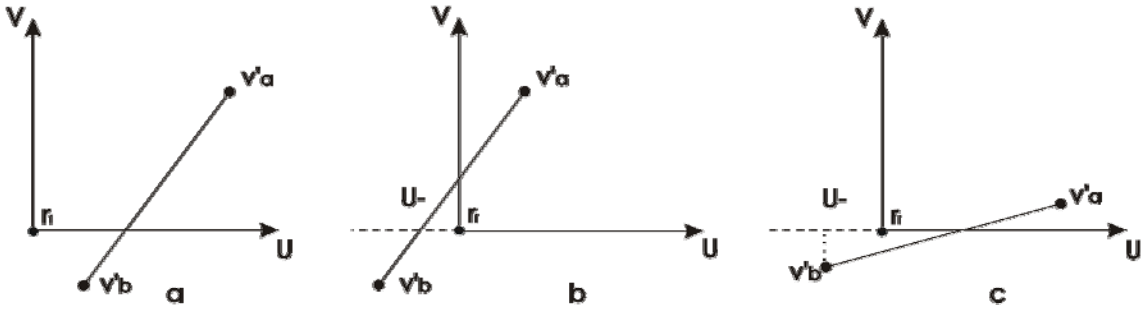

figura 14: casos posibles de cruce del eje U

Por tanto, además de comprobar que la arista cruza el eje U, se ha de verificar si lo hace por la parte positiva, pues en caso contrario no habrá intersección entre la arista y el rayo trazado. Este test se lleva a cabo mediante la condición del algoritmo "si (($u'_a > 0$) y ($u'_b > 0$))". Si esta condición es cierta, el cruce de la arista con U ha de ser necesariamente por U+. De igual forma, si los dos extremos se encontrasen en U–, no habría cruce rayo-arista.

¿Qué ocurre si un extremo de la arista está en U+ y el otro en U–?. En este caso, previsto en el algoritmo por la condición "si(($u'_a > 0$) o ($u'_b > 0$))" se calcula el punto de corte de la arista en U. Si es mayor que cero, evidentemente se encuentra en U+, y por tanto se producirá un cruce rayo-arista.

Con este algoritmo muchas de las aristas del polígono pueden ser rechazadas o aceptadas de una manera rápida y trivial. Tan sólo el último caso es cuando se realizan cálculos para poder decidir si corta el eje o no.

B) Ejemplo numérico sobre el test de pertenencia

Veamos un ejemplo sacado de [Glas89, pg. 57].

Sea un triángulo cuyos vértices son

$$G_0 = [-3, -3, 7]$$
$$G_1 = [3, -4, 3]$$
$$G_2 = [4, -5, 4]$$

y el punto de intersección con el plano $r_i = [-2, -2, 4]$.

Veamos si r_i se encuentra dentro o fuera del triángulo.

La ecuación del plano es $P = [1, 2, 1, 2]$, y por tanto la coordenada dominante de la normal es la del eje Y.

Eliminando este eje de las coordenadas de los vértices y del punto de intersección, se obtiene la proyección paralela de cada punto en el plano XZ, con las coordenadas siguientes:

$$G_{uv0} = [-3,\ 7]$$
$$G_{uv1} = [\ 3,\ 3]$$
$$G_{uv2} = [\ 4,\ 4]$$
$$r_{uvi} = [-2,\ 4]$$

El siguiente paso indicado por el algoritmo, consiste en trasladar los puntos anteriores hacia el origen de coordenadas del sistema UV, de forma que r_i quede situado en el origen, es decir, $r'_i = (0, 0)$. Para ello, se resta de cada vértice el punto de intersección con el plano [–2, 4].

El triángulo proyectado y trasladado será

$$G'_{uv0} = [-1\ \ 3]$$
$$G'_{uv1} = [\ 5\ -1]$$
$$G'_{uv2} = [\ 6\ \ 0]$$

Supongamos ahora que la primera arista que se verifica va de $(u'_a, v'_a) = (–1, 3)$ a $(u'_b, v'_b) = (5, –1)$.

Como los signos de v'_a y v'_b son distintos, esta arista no es rechazada pues cruza el eje U. Sin embargo, también son distintos los signos de u'_a y u'_b, por lo que se ha de calcular la posición del punto de corte de la arista y el eje U

$$u'_a - v'_a \cdot (u'_b - u'_a) / (v'_b - v'_a) \equiv -1 - 3 \cdot (5 - (-1)) / (-1 - 3) = 3.5$$

Tenemos entonces que el corte es en U+, por lo que se produce la intersección rayo-arista, luego se incrementa el contador "cruces" en 1.

Veamos ahora qué ocurre con la segunda arista, que va desde $(u'_a, v'_a) = (5, -1)$ hasta $(u'_b, v'_b) = (6, 0)$.

En este caso la arista también cruza U, pues v'_a y v'_b tienen signos opuestos y además se produce la intersección rayo-arista, ya que sus dos extremos se encuentran en U+ al ser u'_a y u'_b ambos positivos. Por tanto incrementamos a 2 el contador de cruces.

La tercera arista está definida por $(u'_a, v'_a) = (6, 0)$, $(u'_b, v'_b) = (-1, 3)$.

En este ocasión los signos son iguales, lo que implica que la arista no corta a U y por tanto no puede haber intersección rayo-arista. No se incrementa el contador.

Como el algoritmo finaliza con el contador de cruces igual a 2, significa que el punto de intersección rayo-plano se encuentra fuera del polígono.

Ver que el vértice (u', v') = (6, 0), que cae justo en el eje U, no ha supuesto ningún problema al haber sido tratado como un punto más del eje positivo de V.

1.3 Segunda fase: ¿Llega luz? ¿Cuánta y de qué color?

1.3.1 Modelado de las fuentes.

Las características físicas de las fuentes de iluminación es uno de los aspectos más simplificados en la Síntesis de Imágenes. La mayor parte de los sistemas gráficos asumen que las fuentes de iluminación son puntuales, evitando así las consideraciones geométricas que complican y ralentizan los cálculos.

Sin embargo, en escenas complejas (p. e. interiores de habitaciones) la única forma de conseguir realismo es teniendo en cuenta las características de las fuentes de iluminación.

Una descripción adecuada de las características físicas de las fuentes debería incluir los tres atributos siguientes:

- *Geometría de la fuente de iluminación*: una bombilla tiene diferente forma que un fluorescente o una ventana.
- *Distribución de la intensidad luminosa*, dada en función de la geometría de la fuente.
- *Distribución espectral emitida*: las barras fluorescentes tienen una distribución espectral diferente, tanto de las luces incandescentes como de la luz natural.

El principal obstáculo para modelar correctamente las fuentes de iluminación es el tiempo de proceso: *se ha de calcular un vector **L** y una intensidad emitida I diferentes para cada punto de la superficie de la fuente.*

Hay algunos métodos que tratan de modelar las características indicadas, como por ejemplo el modelo de Warn, que tiene en cuenta las dos primeras, o el de Verbeck-Greenberg que considera todas. Veamos el modelo de Warn, que es uno de los más sencillos.

A) Modelo de Warn

Warn desarrolló su modelo basándose en los controles básicos ejercidos sobre las luces utilizadas en los estudios fotográficos: *la dirección y la concentración*.

Para ello, el modelo de Warn define las fuentes como puntos de reflexión especular ubicados en superficies reflectantes hipotéticas, iluminadas por una fuente puntual, según muestra la figura 15. **L** es el vector de la luz incidente, y el vector **L'** el vector normal a la supuesta superficie de reflexión, con origen en la fuente puntual.

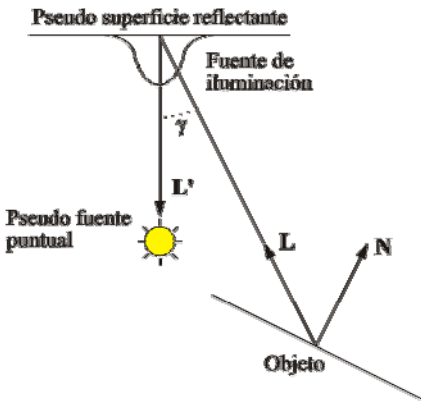

figura 15: en el Modelo de iluminación de Warn

La intensidad de luz que se recibe en el objeto depende del ángulo entre **L'** y **L** (γ), la cual se calcula mediante la componente especular de Phong, suponiendo que el coeficiente de reflexión especular de la superficie hipotética es 1 ($K_e = 1$). Queda entonces que

[18]

$$I_{\mathbf{L}} = I_{\mathbf{L}'} \cos^{p} \gamma$$

donde $I_{\mathbf{L}'}$ es la intensidad de la fuente puntual, $I_{\mathbf{L}}$ es la intensidad de la fuente de Warn y p es un exponente que determina la concentración de la luz que se refleja en superficie ficticia (similar a n en la ecuación de Phong).

Si en la ecuación anterior se sustituye $\cos^{p} \gamma$ por el producto escalar de los vectores correspondientes, se obtiene

[19]

$$I_{\mathbf{L}} = I_{\mathbf{L}'}(-\mathbf{L} \cdot \mathbf{L}')^{p}$$

Sustituyendo esta ecuación por I_{ij} en la ecuación de Phong, quedarían incorporadas las fuentes de iluminación de Warn en el modelo de intensidad Phong.

Dado que $I_{\mathbf{L}}$ depende exclusivamente de la reflexión especular, cuanto mayor sea el valor de p, más concentrada estará la luz a lo largo de **L**. Así, un valor grande de p puede simular una luz muy densa dirigida hacia un punto, mientras que un valor pequeño simulará una luz más difusa. Si p es 0, la fuente de Warn actúa como una fuente puntual con radiación uniforme, es decir, que la luz tiene la máxima intensidad en todas las direcciones.

Para delimitar los efectos de la luz a un área determinada del escenario, Warn implementó las **aletas** (flaps) y los **conos**. Las aletas funcionan de forma similar a las utilizadas por los focos en los estudios de fotografía, restringiendo los efectos de la luz dentro de un rango de coordenadas x, y o z (figura 16-a). Si un punto del objeto está dentro del rango de las coordenadas de las aletas, p.e. $x_{\min} \le x_{\text{objeto}} \le x_{\max}$, se evalúa la contribución de la luz de la fuente. Si no, esta contribución se ignora.

Los conos se utilizan para producir iluminación muy marcada, similar a la creada por los cañones de luz (figura 16-b). Así, suponiendo un cono de revolución, con la fuente en su vértice, el vector **L'** como eje de giro y δ como

ángulo generador, hay dos planteamientos para calcular la intensidad de la fuente:

* Evaluar el modelo de iluminación sólo cuando $\gamma < \delta$ [Fole90, pg. 733], y

* Aplicar la intensidad máxima cuando $\gamma < \delta$ y evaluar normalmente el modelo cuando $\gamma \geq \delta$ [Roge85].

Lógicamente, el primer planteamiento genera círculos de luz más marcados que el segundo.

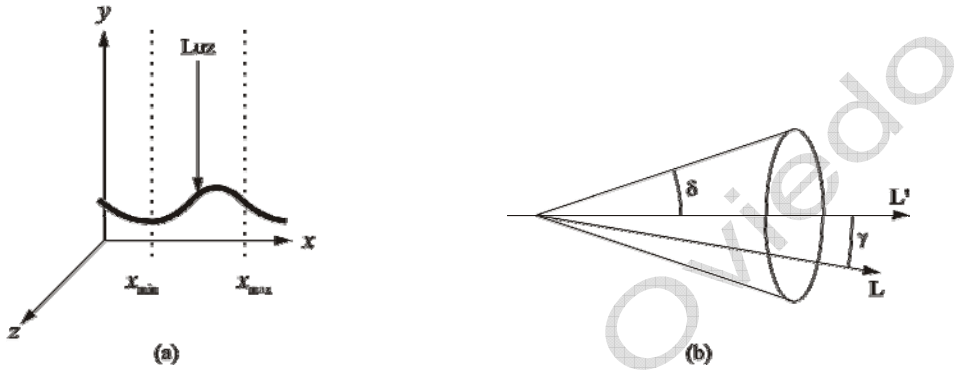

figura 16: la distribución de intensidad de Warn se puede limitar con (a) aletas y (b) conos.

B) Diagramas goniométricos.

Los diagramas goniométricos son *una forma de representar la distribución de la intensidad de la luz alrededor de su vector de dirección*, en coordenadas polares.

La figura 17 muestra las distribuciones de intensidad para una fuente puntual con radiación uniforme y una fuente de iluminación de Warn con diferentes valores de p. Vemos que cuanto mayor es el valor de p, más concentrada se encuentra la luz a lo largo del vector **L**.

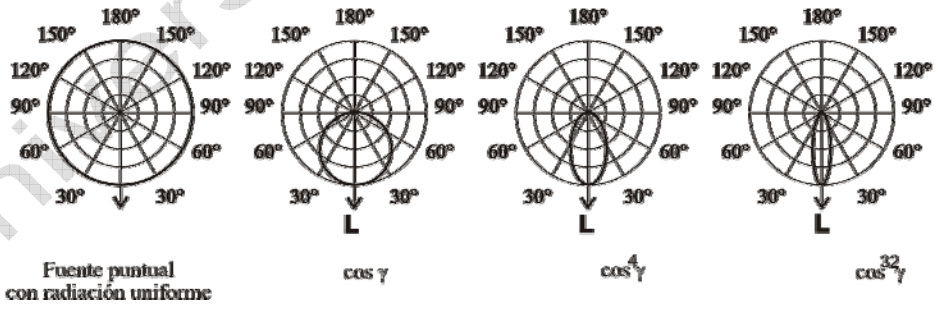

figura 17: diagramas goniométricos

A partir de estos diagramas se puede calcular la intensidad que tiene la luz que se dirige hacia un punto del objeto sencillamente analizando el punto de corte del vector **L** con la curva del diagrama. Algunos modelos de iluminación más complejos, como el de Verbeck-Greenberg, utilizan los diagramas goniométricos de las fuentes en el cálculo de las intensidades.

1.3.2 Sombreado

A la hora de calcular la intensidad de luz que llega a las superficies desde las fuentes, no solo hay que considerar las características de éstas, sino también la posibilidad de que su luz sea absorbida o desviada por objetos opacos o semitransparentes, lo que produciría el efecto de *sombra* o bien el de atenuación de la intensidad.

En principio, por cada punto de intersección de un rayo primario con un objeto, el algoritmo de ray casting ha de averiguar si el punto está en sombra o no. Para ello se utilizan los denominados **rayos de sombras** (*shadow rays*) o **sensores de sombras** (*shadow feelers*) [Glas89]. Estos rayos son trazados desde el punto de intersección hacia cada una de las fuentes de luz que haya en la escena, uno por fuente.

Al trazar un rayo hacia una fuente, dependiendo de los objetos en la escena, pueden darse tres situaciones:

- El rayo de sombra alcanza la fuente de luz sin intersecar con ningún objeto en su camino, lo que implica que la fuente de luz es visible desde el punto de intersección. Por lo tanto, la contribución de la fuente de luz deberá ser considerada a la hora de obtener la intensidad directa de las fuentes (I_i) en ese punto.

- El rayo alcanza algún objeto opaco antes de llegar a la fuente de luz. En este caso, el punto que consideramos en la superficie estará en sombra con respecto a la fuente. Si ninguna otra fuente iluminase el punto de intersección rayo-objeto, entonces solamente recibiría la intensidad de la componente ambiental.

- Finalmente, si el rayo de sombra interseca un objeto semitransparente, la I_i se atenúa. Al intersecar con un objeto transmisor, el rayo de sombra debería refractarse; en ese caso el rayo no alcanzaría la fuente de luz. En la mayoría de los sistemas gráficos, simplemente se calcula la atenuación debida al objeto, sin alterar la trayectoria del rayo de sombra.

Un aspecto importante del sombreado a tener en cuenta es que el número de fuentes de luz puede afectar en gran medida al tiempo de trazado de una imagen. En una escena con n fuentes se han de trazar n rayos de sombra por cada punto de intersección. Si se incrementa el número de fuentes, se incrementará en gran medida el tiempo de ejecución, ya que el coste del cálculo de la intersección de un objeto con el rayo de sombra es igual al de un objeto con un rayo primario. Más adelante veremos las técnicas que permiten acelerar el proceso de sombreado.

1.3.3 Modelos de color de interface de usuario.

En el tema anterior vimos cómo eran los *modelos de color hardware*. Ampliaremos ahora este campo con el estudio de los *modelos de color de interfaz*, y más en concreto el modelo **HSV** y el **HLS**.

Como ya sabemos, una de las desventajas de los modelos RGB y CYM es que *resulta difícil para el usuario variar las características de un color cualquiera*, si no efectúa cálculos previos sobre el color resultante.

Los modelos de interface de usuario permiten obtener la gama de colores de forma más intuitiva, aplicando un método similar al utilizado por los pintores en sus paletas. Éstos consiguen las tonalidades que buscan añadiendo color blanco y/o negro a los colores puros, para aclarar y oscurecer dichos colores.

A) Modelo HSV

En este modelo, los parámetros a disposición del usuario para el control del color son los de Matiz (**H**ue), **S**aturación y **V**alor (HSV). El conseguir un color más brillante, más pálido o más oscuro es mucho más fácil manipulando estas variables, que averiguar los valores RGB apropiados.

El HSV se representa gráficamente mediante una pirámide hexagonal invertida, la cual deriva del cubo de representación del modelo RGB. Si imaginamos una vista a lo largo de la diagonal que une el blanco con el negro (origen) en dicho cubo, vemos el perfil del cubo que tiene forma hexagonal, como puede verse en la figura 18.

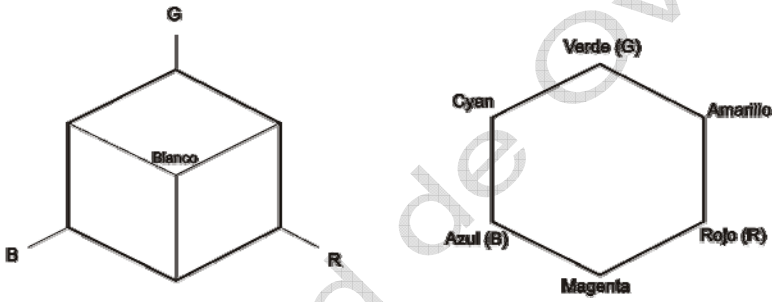

figura 18: base de la representación del modelo HSV

Utilizando el hexágono como base de una pirámide invertida, en cuyo perímetro se distribuyen los diferentes matices, se obtiene la representación gráfica del HSV, tal como muestra de la figura siguiente:

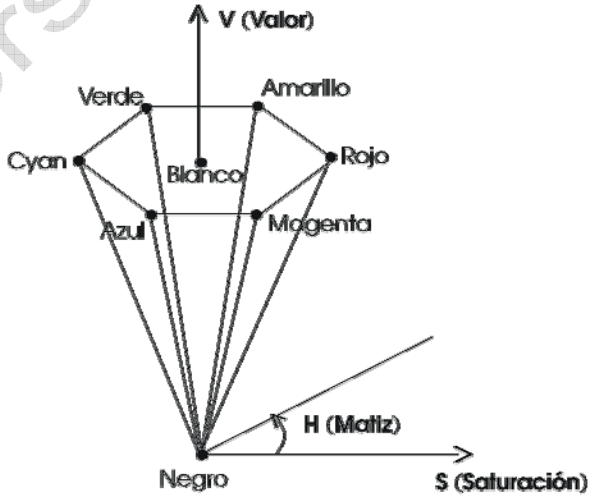

figura 19: representación espacial del modelo HSV

Los tres parámetros de este modelo se representan en la pirámide de la siguiente manera:

*** Matiz:**

Es representado mediante un ángulo, tomando como referencia el eje vertical, que varía desde los 0º del rojo hasta los 360º. Los vértices del hexágono están separados por intervalos de 60º. Así, dos colores que presenten una diferencia de 180º serán complementarios.

*** Saturación:**

La saturación mide la pureza relativa de un matiz (color). Se mide a lo largo del eje horizontal y su rango de valores está comprendido entre 0 y 1. Cuando en el modelo S = 1, un matiz alcanza su pureza máxima. Si S = 0, nos encontraremos en la escala de grises.

*** Valor:**

Se mide sobre el eje vertical desde el centro de la pirámide. Sus valores varían también entre 0 (en el vértice de la pirámide) y 1. En la base de la pirámide los colores presentan su intensidad máxima.

Un color que presente los parámetros V y S a 1 es un color puro. El blanco tiene de componentes V=1 y S=0.

Desde el punto de vista del usuario, el oscurecer un color añadiéndole negro, significará disminuir el valor del parámetro V dejando S constante. Aclarar un color añadiéndole blanco, se conseguirá disminuyendo la saturación del color (parámetro S) manteniendo el valor constante (parámetro V).

Por ejemplo, si se desea obtener un color amarillo oscuro, partiendo de V = S = 1 y H = 60º, puede conseguirse dando a V el valor 0.3. Si lo que se pretende es obtener un amarillo claro, entonces dejando V = 1 puede obtenerse haciendo S = 0.4.

Según [Hear97] el ojo humano puede distinguir 128 matices diferentes, y alrededor de 130 niveles de saturación (valores de S). El total de valores de V que pueden ser detectados dependen del matiz seleccionado. Así, para el color amarillo se pueden distinguir unos 23 valores diferentes de V (intensidades), mientras que para el azul solamente 16. Esto significa que somos capaces de percibir 128 x 130 x 23 = 382.720 colores diferentes.

Ya sean estos datos verdaderos o no, lo cierto es que en las aplicaciones gráficas comerciales el rango de posibilidades suele ser más amplio, del orden de los 3.600.000 colores, o más.

Los valores de un determinado color generado mediante los parámetros de este modelo son transformados a valores del modelo RGB ya que es el modelo utilizado en los monitores en color. Los algoritmos que realizan la conversión de un color en el modelo RGB al HSV y viceversa, están descritos en [Watt89] págs.321-322 , en [Hear97], pág. 578 y en [Fole90] pág.592.

B) Modelo HLS

El modelo HLS se trata de una variante del anterior. Los parámetros utilizados son el de matiz (**H**ue: color en estado puro), **L**uminosidad (brillo por unidad de área) y **S**aturación (proporción de blanco en el color). La representación gráfica de este modelo es la siguiente:

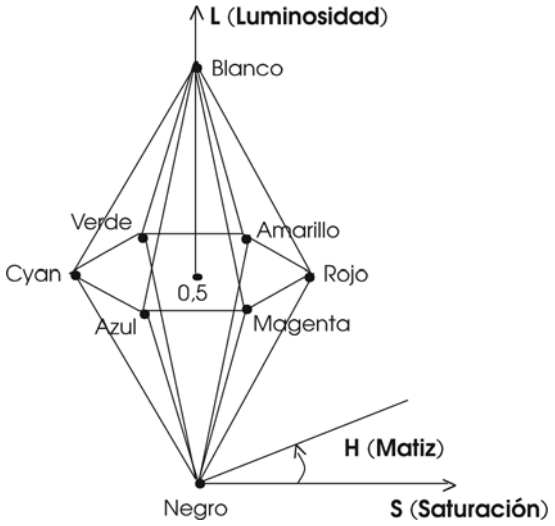

figura 20: representación espacial del modelo HLS

El **matiz** (H) se representa de igual manera que en el modelo anterior, por medio de un ángulo. Como en el HSV, el valor 0º se corresponde con el color rojo, aunque según preferencias y autores H = 0º puede corresponder al azul. El resto de los colores se representa a lo largo del perímetro de la pirámide en el mismo orden que el modelo HSV. Los colores complementarios están separados por 180º como ocurría en el modelo anterior.

La **luminosidad** en este modelo se representa mediante el eje vertical que atraviesa la pirámide. El negro corresponde a L = 0, y el blanco a L = 1.

La **saturación** también varía entre 0 y 1, y se consideran colores puros aquellos en los que se verifica que S = 1 y L = 0,5. A medida que el parámetro S va disminuyendo, los colores van siendo menos puros.

Vemos que conceptualmente el modelo HLS es similar al HSV. Los colores serán aclarados incrementando el parámetro L y oscurecidos decrementando dicho parámetro.

Los algoritmos de conversión entre el modelo RGB y el HLS están descritos en [Watt89] págs. 324-325 y en [Fole90] pág. 595.

1.3.4 Aplicaciones y utilización del color

A) La interpolación del color.

En un modelo de color, la interpolación del color consiste en calcular las componentes primarias a partir de dos colores conocidos, aplicando un determinado atributo (o peso). Dentro del espacio del modelo de color, el nuevo color se encontrará en la línea recta que une los colores de partida. Los resultados de la interpolación van a depender del modelo de color en el cual se lleve a cabo la interpolación, y por lo tanto es muy importante seleccionar un modelo de color apropiado.

En la conversión de un modelo de color a otro, si una línea recta de interpolación en el espacio del modelo de origen se convierte también en una línea recta en el modelo de destino, entonces los resultados de la interpolación lineal en ambos modelos será los mismos. Así ocurre, por ejemplo, en

los modelos RGB, CMY y CIE, los cuales están relacionados mediante transformaciones simples.

Sin embargo, esto no ocurre al pasar de alguno de estos modelos al modelo HSV o al HLS. Normalmente una línea recta de interpolación en cualquiera de los modelos hardware no se convierte en otra línea recta en los modelos de interface. Veamos un ejemplo.

Consideremos la interpolación entre los colores rojo y verde. En el modelo RGB, el rojo se corresponde con el (1, 0, 0) y el verde con el (0, 1, 0). Suponiendo la misma proporción de ambos colores en la interpolación obtendremos el color (0.5, 0.5, 0). Si este color lo pasamos al modelo HSV, mediante el algoritmo apropiado, al que ya se ha hecho referencia, se obtiene el color (60º, 1, 0.5).

Por otro lado, el color rojo en el modelo HSV viene representado por (0º, 1, 1) y el verde por (120º, 1, 1). Su interpolación, suponiendo proporciones iguales, genera el color (60º, 1, 1). Como puede verse, interpolar dos colores y luego pasar el color resultante al otro modelo de color, no produce el mismo resultado que transformar primero los colores que se van a interpolar y después realizar su interpolación.

Si el objetivo es interpolar dos colores del mismo matiz y mantener dicho matiz para todos los colores resultantes de la interpolación entonces los más apropiados son los modelos HSV o HLS.

En la Síntesis de Imágenes hay al menos tres ocasiones en las que la interpolación del color es de gran importancia. Se interpolan colores cuando se tengan que pintar (rellenar) los trozos de superficies (parches), utilizando el método standard de visualización. También es importante la interpolación cuando no se disponga de información suficiente (aliasing) sobre ciertos puntos (o zona) de una superficie, por lo que se recurre a la interpolación de los colores circundantes para "maquillar" aquellos puntos que carezcan de información. Por último, otra ocasión en la que se utiliza profusamente la interpolación de colores es cuando se desea realizar la combinación o fusión gradual de imágenes, superficies, etc. (*blending*).

Para finalizar, un ejemplo de interpolación no lineal consiste en asociar un objeto con un color determinado, basándose en la distancia entre el objeto y el observador. Esta técnica se utiliza mucho en los sistemas de simulación de vuelo donde los objetos muy distantes se asocian al color del horizonte. Gracias a este tipo de interpolación, se transmite la sensación de profundidad.

B) Pseudocolor o color falso

La asignación de color a los datos (o pseudocolor) es una práctica muy común de la visualización gráfica de la información numérica. El proceso general del pseudocolor consiste en transformar un conjunto de datos (normalmente no visibles) en un conjunto de colores o escalas (tonos) de un color. Por lo tanto, el objetivo del pseudocolor es el de *incrementar la percepción de datos o resultados que normalmente no producen salida visual*.

En consecuencia, esta técnica se utiliza en numerosas aplicaciones en las que se recogen datos cuya visualización favorece su análisis. Ejemplos típicos de la utilización del pseudocolor son los mapas de temperaturas, los mapas geográficos y algunas exploraciones médicas.

En este último caso, la información es generada mediante scanner, resonancias magnéticas (RMN), etc. A continuación se representa en una pantalla mediante el pseudocolor de forma que se relacionan ciertos colores con unos datos concretos (una ecografía, por ejemplo, es capaz de dibujar la silueta del feto asociando datos resultantes de aplicar la técnica del ultrasonido con diferentes colores).

Cuando se trata de mapas geográficos, cada altura sobre el nivel del mar se representa mediante un color diferente, utilizando normalmente un número constante de los colores (de 6 a 10). De este modo, las fronteras entre desniveles se aprecian perfectamente debido al contraste de los colores adyacentes. Además, los colores suelen seleccionarse en relación al contexto o la realidad física; así, el blanco se utiliza para montañas nevadas, verde para zonas con hierba, azul para extensiones de agua, etc. Es deseable que los colores adyacentes reflejen también el desnivel que existe entre dos zonas contiguas.

Vemos, por tanto, que la selección de los colores es muy importante cuando se utiliza el pseudocolor. Normalmente se eligen los colores resultantes de la descomposición de la luz blanca (gama del arco iris), empleando el rojo para representar conceptos como alto o caliente y el azul para intensidades bajas o frías.

Uno de los problemas que presenta el pseudocolor es la posibilidad de generar contornos falsos o, lo que es lo mismo, fronteras falsas. Reciben este nombre porque representan discontinuidades que no existen en los datos originales. En el caso de los mapas terrestres, estas fronteras falsas son muy útiles para precisar las diferencias de nivel. En otros contextos, las fronteras falsas no deben ser representadas gráficamente; por tanto las fronteras falsas ayudan a la interpretación de los datos o no, dependiendo del contexto. Cuantos menos colores se utilicen, más claras se mostrarán las fronteras falsas.

1.4 Tercera fase: ¿De qué intensidad y color es la luz que llega al observador?

1.4.1 Modelo de Cook y Torrance

El modelo de intensidad de Phong es relativamente sencillo. Como sabemos, trabaja con aproximaciones empíricas, especialmente en la componente especular, pero carece de una base física. Esto hace que presente algunos defectos, como pronto veremos.

Con el fin de evitar los fallos que se producen en los modelos empíricos se desarrollaron modelos con una base física (Blinn, Cook-Torrance, Hall, etc.). En este apartado veremos el modelo de Cook y Torrance por ser uno de

los más extendidos.

A diferencia de Phong, Cook y Torrance basan el término especular en un modelo físico de superficie, desarrollado por los físicos Torrance y Sparrow y adaptado por Blinn para la síntesis de imágenes. El modelo tiene en cuenta la energía incidente (en vez de la intensidad) y el cambio de color en los reflejos especulares, es decir, el color de la luz especular no siempre será igual al de la fuente que la emite.

El resultado es una mejora en la representación de las superficies, especialmente las metálicas, debido principalmente a que se refina el cálculo del término especular cuando los ángulos de incidencia son grandes. Estas mejoras conllevan un aumento considerable del tiempo de computación, por lo que este modelo de iluminación generalmente se utiliza sólo para la representación de la imagen final, usando modelos más simples en las etapas preliminares.

El modelo de Cook y Torrance, al igual que el de Phong, considera tres componentes: la reflexión difusa, la especular, ambas debidas a la iluminación directa, y la componente ambiental.

Las diferencias con el modelo de Phong pueden resumirse en los siguientes puntos:

1. Tiene en cuenta la energía incidente, en vez de la intensidad.
2. El factor especular se basa en un modelo físico que considera que la superficie de los objetos está formada por microcaras o microsuperficies.
3. Introduce cambios de color en los reflejos especulares.

A) Expresión general del modelo de Cook y Torrance

La energía incidente es proporcional a la intensidad y al tamaño de la fuente de iluminación. El "factor tamaño" se incluye en las ecuaciones utilizando el **ángulo sólido** del haz de luz incidente. El ángulo sólido se mide en función del área proyectada sobre la superficie de un hemisferio por un cono, cuyo vértice se encuentra en el centro del hemisferio (figura 21).

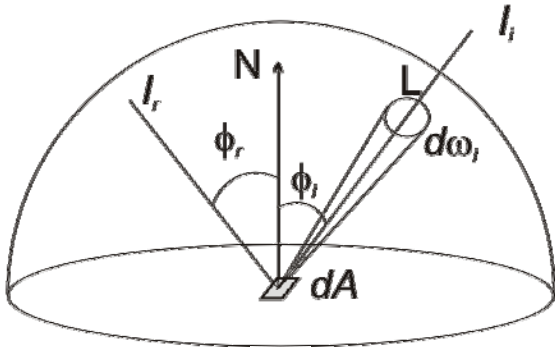

figura 21: concepto de ángulo sólido

Cuando dω es pequeño, se puede aproximar su cálculo suponiendo que el ángulo sólido es igual al área proyectada por la fuente sobre un hemisferio— con el punto de convergencia situado en su centro —dividido por el

cuadrado de la distancia entre la fuente y el punto de convergencia.

La energía incidente por unidad de superficie y unidad de tiempo (E_i), se calcula como

[20]

$$E_i = I_i(\mathbf{N} \bullet \mathbf{L})d\omega_i$$

donde I_i es la intensidad de la fuente y $d\omega_i$ el ángulo sólido del haz de luz incidente.

La utilización de la energía incidente para calcular la intensidad de la luz reflejada tiene importancia respecto a los resultados, ya que una fuente de iluminación con la misma intensidad y ángulo de iluminación que otra, pero con el ángulo sólido mayor, hace que la superficie iluminada sea proporcionalmente más brillante.

Por otro lado, se conoce como *reflectividad bidireccional* (R_r) a la relación entre la energía incidente E_i y la intensidad reflejada I_r, siendo

[21]

$$R_r = \frac{I_r}{E_i},$$

es decir, la proporción de energía que se refleja. Sustituyendo en [20] se obtiene

[22]

$$I_r = R_r I_i(\mathbf{N} \bullet \mathbf{L})d\omega_i$$

Dado que en la Síntesis de Imágenes es normal considerar la reflectividad bidireccional como la suma de la reflexión difusa y la especular, queda entonces que:

[23]

$$R_r = K_d R_d + K_e R_e$$

donde R_d, R_e representan la reflectividad bidireccional difusa y especular, respectivamente, y K_d, K_e son los respectivos coeficientes de reflexión, siendo $K_d + K_e = 1$.

Sustituyendo la ecuación [23] en la [22] se obtiene

[24]

$$I_r = I_i(\mathbf{N} \bullet \mathbf{L})d\omega_i(K_d R_d + K_e R_e)$$

Si a la expresión anterior se le añade la componente ambiental se obtiene la ecuación del modelo de C&T, para una sola fuente de luz:

[25]

$$I_r = I_a R_a + I_i(\mathbf{N} \bullet \mathbf{L})d\omega_i(K_d R_d + K_e R_e)$$

siendo R_a e I_a la reflectividad e intensidad ambientales, respectivamente. La componente ambiental puede mejorarse introduciendo un *coeficiente de blo-*

queo que determina la proporción del hemisferio que no es obstruido por los objetos cercanos.

La ecuación anterior para *n* fuentes quedaría como sigue:

[26]

$$I_r = I_a R_a + \sum_n I_{i,n}(\mathbf{N} \bullet \mathbf{L}_n) d\omega_{i,n}(K_d R_d + K_e R_e)$$

i.- Cálculo de la reflexión especular (R_e)

Recordemos que en el modelo de Phong la dispersión de la contribución especular alrededor del vector de reflexión **R** se modela mediante el $cos^n \phi$. En cambio, en el modelo de Cook y Torrance se basa en un modelo físico desarrollado por Torrance y Sparrow, que supone que las superficies están formadas por muchas caras pequeñas (microcaras) orientadas aleatoriamente; se considera que cada microcara es un reflector perfecto.

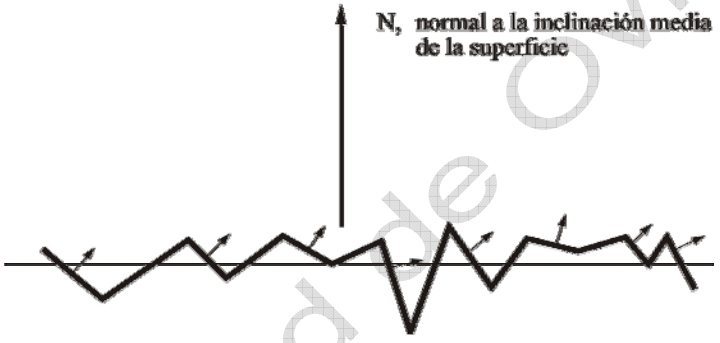

figura 22. superficie modelada por microcaras reflectantes.

Suponiendo entonces que el punto de la superficie donde se pretende calcular la intensidad especular reflejada está formado por unas cuantas de estas microcaras, la reflexión especular en dicho punto viene dada por

[27]

$$R_e = \frac{FDG}{\pi \ (\mathbf{N} \cdot \mathbf{V})(\mathbf{N} \cdot \mathbf{L})}$$

Veamos qué significa cada uno de los términos de esta ecuación.

De todas las microcaras que forman la superficie *sólo las que poseen la orientación en el espectro (familia) de direcciones del vector H* (vector medio estudiado en el modelo de Phong) han de ser consideradas activas en la reflexión especular. La proporción de microcaras que poseen dicha orientación queda representada en la ecuación [27] por la *función de distribución de las microcaras* (**D**).

Una de las aportaciones originales de Cook y Torrance con respecto a sus precursores es la utilización de la *función de distribución de Beckmann* para calcular la proporción de microcaras que poseen la orientación apropiada. Dicha función, para superficies no pulidas, viene dada por:

[28]

$$D = \frac{1}{4m^2 \cos^4 \beta} e^{-[\tan^2 \beta / m^2]}$$

donde β es el ángulo entre **N** y **H**, y *m* es la desviación standard de las pendientes de las microcaras. Cuando *m* es pequeña, la orientación de la microcaras varía poco con respecto a la normal, por lo que la reflexión estará muy concentrada sobre la trayectoria teórica (similar a valores altos de *n* en el modelo de Phong). Por el contrario, si *m* grande, las múltiples reflexiones en las microcaras dará como resultado una reflexión difusa (superficie difusora), al presentar las microcaras pendientes pronunciadas, es decir, con una orientación casi perpendicular a la normal de la superficie.

Además de la orientación, otro factor importante a tener en cuenta es *la cantidad de microcaras que ve el observador*. De esta cuestión se encarga el factor (**N**·**V**).

Cuanto menor sea el valor de **N**·**V**, mayor será el número de microcaras que verá el observador, es decir, que su aportación es inversa, razón por la cual se encuentra en el denominador de la ecuación [27]. En otras, palabras, a medida que aumenta el ángulo entre **N** y **V** (y disminuye el valor del coseno de dicho ángulo), es mayor la superficie que se ve a lo largo de la dirección de visión, y por tanto, también será mayor el total de microcaras visibles. En términos de reflexión especular esto significa que, en principio, cuanto mayor sea el ángulo entre **N** y **V**, mayor será la intensidad especular que llega al observador. Sin embargo, el término **G** de la ecuación [27] tiene algo que decir a este respecto, como pronto veremos.

Argumentos similares a los anteriores se utilizan para justificar la presencia en [27] del producto escalar (**N**·**L**), aunque en este caso se incluye en la ecuación para determinar la proporción de microcaras iluminadas por la fuente, en función del ángulo de la luz incidente.

Como acabamos de comentar, otro término que se ha de tener en cuenta en el cálculo de la reflexión especular es el *factor de atenuación geométrica* (**G**), basado en el hecho de que algunas microcaras pueden tapar la luz a otras, lo que atenuaría la intensidad del reflejo en ese punto. Esta atenuación puede deberse a dos causas, ambas relacionadas con el ángulo de incidencia de la luz y la inclinación de las microcaras: el sombreado y el enmascaramiento, según muestra la figura 23.

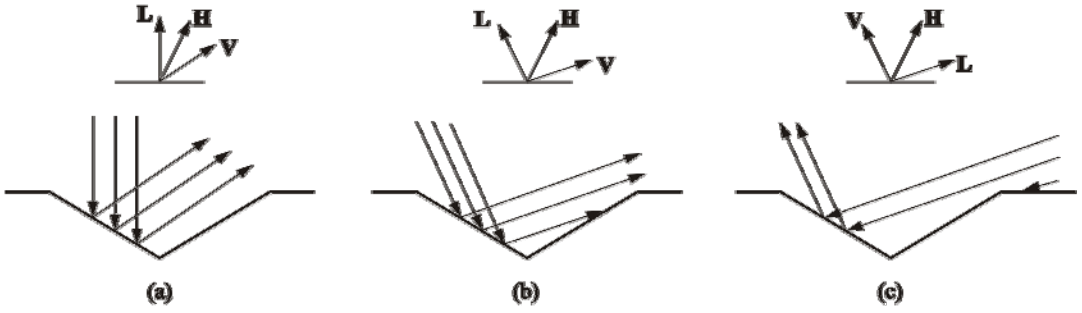

figura 23: efectos de enmascaramiento (b) y sombreado (c)

G puede tomar valores en el rango[0, 1]. Si G es 0, el sombreado (o el enmascaramiento) es total.

Las ecuaciones que permiten calcular el valor de G en cada caso vienen dadas por:

[29]

$$G_a = 1,\ G_b = \frac{2(\mathbf{N}\cdot\mathbf{H})(\mathbf{N}\cdot\mathbf{V})}{\mathbf{V}\cdot\mathbf{H}},\ G_c = \frac{2(\mathbf{N}\cdot\mathbf{H})(\mathbf{N}\cdot\mathbf{L})}{\mathbf{V}\cdot\mathbf{H}}$$

Ver que para el cálculo de G_b y G_c no es preciso cambiar el denominador, ya que (**V**·**H**) = (**L**·**H**), al ser **H** el vector bisectriz del ángulo que forman los vectores **V** y **L**. Para más detalles sobre la obtención de estas ecuaciones, ver [Fole90] o [Watt89].

G se define como el mínimo de los tres valores. Así,

[30]

$$G = \text{mín}[G_a, G_b, G_c]$$

Finalmente, el último factor de la ecuación [27] que nos queda por ver es **F**, conocido como el *término de Fresnel*, el cual se define como:

[31]

$$F = \frac{1}{2}\left(\frac{\tan^2(\phi-\theta)}{\tan^2(\phi+\theta)} + \frac{\text{sen}^2(\phi-\theta)}{\text{sen}^2(\phi+\theta)}\right) = \frac{1}{2}\frac{\text{sen}^2(\phi-\theta)}{\text{sen}^2(\phi+\theta)}\left(1+\frac{\cos^2(\phi+\theta)}{\cos^2(\phi-\theta)}\right)$$

donde ϕ es el ángulo de incidencia de la luz, *dado con respecto a* **H,** y θ es el ángulo de refracción (ver la figura 24), cumpliéndose que $\text{sen}\,\theta = \frac{\eta_i}{\eta_t}\cos\phi$, siendo η_i y η_t los índices de refracción de los dos medios.

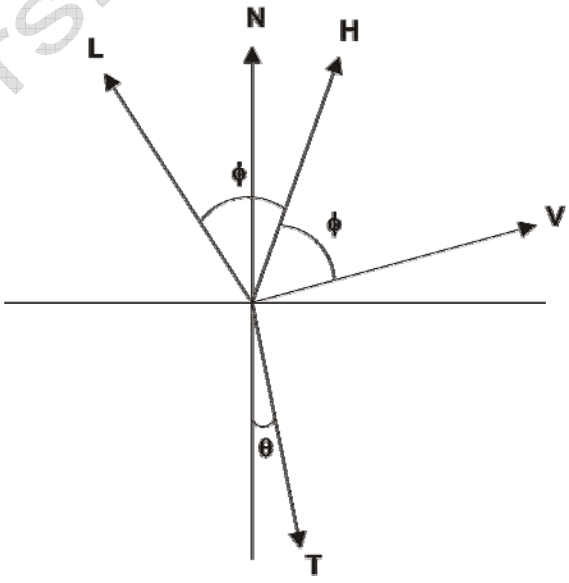

figura 24: reflexión y refracción de la luz incidente

La ecuación de Fresnel se puede reescribir como sigue:

[32]

$$F = \frac{1}{2}\frac{(g-c)^2}{(g+c)^2}\left[1+\frac{(c(g+c)-1)^2}{(c(g-c)+1)^2}\right]$$

donde $c = \cos\phi = \mathbf{L}\cdot\mathbf{H} = \mathbf{V}\cdot\mathbf{H}$ y $g^2 = c^2 + \eta^2 - 1$, siendo $\eta_\lambda = \frac{\eta_{t\lambda}}{\eta_{i\lambda}}$

Como sabemos, las distintas longitudes de onda de la luz se refractan de forma diferente en un mismo material, que es lo mismo que decir que *el índice de refracción del material depende de la longitud de onda incidente*, de ahí que a η se le asocie la longitud de onda λ (η_λ).

ii.- Variación del color en la reflexión especular

Otra de las novedades del modelo de C&T es que *el color de la reflexión especular no es constante* e igual al de la fuente de luz, como ocurre en el modelo de Phong.

En la ecuación de la reflexión especular [27], la responsabilidad de la variación del color está a cargo del término de Fresnel. Según podemos ver en la ecuación [32], F depende de ϕ (ángulo de incidencia) y de η (índice de refracción). Sin embargo, η varía según sea la longitud de onda (color) del rayo incidente, es decir, que en definitiva $F = f(\phi, \lambda)$.

Esta dependencia que tiene el factor especular con el ángulo de incidencia y con la longitud de onda implica que el color va cambiando en los reflejos especulares a medida que el ángulo de incidencia ϕ se acerca a los 90° respecto a la normal.

Así, cuando la luz incidente tiene la misma dirección que **H**, entonces $\phi = 0^o$ (recordemos que ϕ está dado con respecto a **H**, y no a **N**). En este caso, $c = 1$ y $g = \eta_\lambda$. Sustituyendo estos valores en la ecuación [32], queda

[33]

$$F_{\lambda 0} = \left(\frac{\eta_\lambda - 1}{\eta_\lambda + 1}\right)^2$$

En el otro extremo, cuando $\phi = 90^o$, entonces $c = 0$. Sustituyendo de nuevo en [32] queda

[34]

$$F_{\lambda 90} = 1$$

Vemos por la ecuación [33] que, cuando la luz incidente es casi perpendicular a la superficie ($\phi = 0°$), entonces toma el valor $F_{\lambda 0}$, y por lo tanto la reflexión especular dependen del índice de reflexión, y en último término de λ. Por el contrario, cuando $\phi = 90°$, el término de Fresnel se hace constante. En definitiva, el valor de la componente especular depende de η_λ para todos los valores de ϕ, excepto cuando vale 90^o.

Que el modelo de intensidad tenga presente la variación de R_e en función de (ϕ, λ) está más acorde con la realidad, ya que la mayor parte de los materiales responden con variaciones en R_{eR}, R_{eG}, y R_{eB}, a medida que varían los parámetros ϕ y λ. La excepción más significativa a esta regla está representada por los plásticos, de ahí que el modelo de Phong sea tan apropiado para visualizar objetos fabricados con estos materiales, y no tanto cuando los materiales son de otro tipo, como por ejemplo metálicos.

Si son conocidos los índices de refracción para las diferentes longitudes de onda, entonces pueden aplicarse directamente el la ecuación de Fresnel. Sin embargo, lo más normal es que se desconozcan dichos índices, por lo que deberán ser calculados. Para ello, se recurre a las tablas (o bases de datos) de reflectividad, las cuales se han construido midiendo la reflectividad de los principales materiales, con $\phi = 0^o$ y para diferentes valores de λ.

A partir de estas tablas, η_λ se determina mediante la ecuación:

[35]

$$\eta_\lambda = \frac{1+\sqrt{F_{\lambda 0}}}{1-\sqrt{F_{\lambda 0}}}$$

que se deduce directamente de la ecuación [33].

Para finalizar, dado que es computacionalmente prohibitivo calcular F para cada λ del espectro, Cook y Torrance proponen que se calcule el valor medio de λ (λ_m), que será utilizado para calcular $F_{m\phi}$, para un valor dado del ángulo de incidencia. A partir de $F_{m\phi}$ se calculan los valores R,G,B, aunque no de modo tan simple como se hacía en el modelo de Phong. Para más detalles sobre el cálculo del color ver [Fole90] (pg. 770), o [Watt89] (pg. 74).