

İŞIN İZLEME (RAY TRACING)

1.Giriş

Işın İzleme (Ray Tracing) yöntemi, bakış noktasından çıkan ve görüntü düzlemindeki piksellerin herbirinden geçen ışınlar ile 3D ortamdaki cisimler arasında yapılan kesişim testleri sonucu belirlenen en yakın cismin (eğer bu cisim ışını yansıtma özelliğine sahipse veya saydam olup ışın içinden doğrudan veya kırılarak geçiyorsa ondan yansıyıp/kırılıp çarptığı ilk cismin rengi de hesaba katılarak son) renginin bir boyama modeli (örneğin Phong boyama modeli) ile belirlenmesi olarak tanımlanabilir. Işın vektörel bir büyüklük olduğu için öncelikle vektörel işlemler hakkında bilgiler verilecek ve ardından ilerleyen bölümlerde ışın izleme yöntemi detaylı olarak incelenecektir.

2. Vektörel İşlemlerle İlgili Temeller

Herhangi bir vektörün boyunu bulmak için (x,y,z) koordinatlarının karelerinin toplamının karekökü alınır. Örneğin $R=(0,6,8)$ vektörünün boyu $|R| = \sqrt{0^2 + 6^2 + 8^2} = 10$.

Herhangi bir vektörün boyunu 1 birim yapma işlemine “normalizasyon” denir ve işlem için (x,y,z) koordinat bileşenlerinin her biri vektörün boyuna bölünür. Yukarıdaki (0,6,8) vektörü normalize edildiğinde $(0/10, 6/10, 8/10) = (0, 0.6, 0.8)$ bulunur. Normalize edilmiş vektörün boyu hesaplandığında $\sqrt{(0)^2 + (0.6)^2 + (0.8)^2} = 1$ olduğu görülür. Normalize edilmiş vektöre “birim vektör” denir. (Vektörel büyüklükler koyu, skaler büyüklükler normal font ile yazılacaktır).

Vektörler arasında skaler ve vektörel olmak üzere iki temel çarpım işlemi vardır. R_1 ve R_2 gibi iki vektörün sırasıyla skaler ve vektörel çarpımları aşağıdaki gibi yapılır:

$$R_1 * R_2 = R_{1x}R_{2x} + R_{1y}R_{2y} + R_{1z}R_{2z} = |R_1| * |R_2| * \cos(\beta)$$

$$R_1 \times R_2 = (R_{1y}R_{2z} - R_{1z}R_{2y}, R_{1z}R_{2x} - R_{1x}R_{2z}, R_{1x}R_{2y} - R_{1y}R_{2x})$$

İki vektörün skaler çarpımında (x,y,z) koordinatları ayrı ayrı çarpılıp toplanır veya vektörlerin boylarının aralarındaki açının kosinüsüyle çarpımı olarak da hesaplanabilir. Dolayısıyla birim vektörlerin skaler çarpımı aralarındaki açının kosinüsünü verir. Skaler çarpım Phong boyama modelinin diffuse ve specular renk bileşenlerinin hesaplanmasında kullanılır. R_1 ve R_2 'nin vektörel çarpımıyla

onlara dik olan bir vektör elde edilir. Dolayısıyla normal hesabı vektörel çarpımla yapılır. Normalin doğrultusunu köşe noktalarının sırası belirler. Örneğin bir üçgenin normali hesaplanıyorsa normal üçgenin köşe noktalarının saat yönündeki yüzüne dik olur.

Köşe noktalarının koordinatları V_0, V_1, V_2 şeklinde verilen üçgenin yüzey normali N vektörü, $(V_1 - V_0)$ ve $(V_2 - V_0)$ vektörlerinin vektörel çarpımı ile şöyle hesaplanır: $V_0 = (0, 40, 120)$ $V_1 = (30, -40, 60)$ $V_2 = (-30, -40, 60)$ $(V_1 - V_0) = (30, -80, -60)$ $(V_2 - V_0) = (-30, -80, -60)$ $N = (V_1 - V_0) \times (V_2 - V_0) = (-80 \cdot -60 - -60 \cdot -80, -60 \cdot -30 - 30 \cdot -60, 30 \cdot -80 - -80 \cdot -30)$ $N = (0, 3600, -4800)$ N vektörü normalize edilirse $(0, 0.6, -0.8)$ elde edilir.

3. Işının Tanımı ve Birincil Işınlara Üretilmesi

Başlangıç noktası ve doğrultuya sahip vektörel bir büyüklük olan R ışını:

$$R = R_0 + tR_d$$

olarak ifade edilir. Burada R_0 ışının başlangıç noktası (origin), R_d de doğrultusudur (direction). t ise ışının 3D uzayda R_d doğrultusu boyunca kaç birim gideceğini belirleyen skaler bir değerdir. Doğrultu vektörü R_d 'nin hesaplanabilmesi için 2 noktaya ihtiyaç vardır. Bunlar R_1 ve R_2 olarak alınırsa R_1 'den R_2 'ye doğru olan doğrultu vektörü $R_d = R_2 - R_1$ ile hesaplanır. Işın izlemede doğrultu vektörleri birim vektör olmalıdır. Dolayısıyla R_d normalize edilerek boyu 1 birim yapılır. Işın izlemede ilk adım başlangıç noktasından çıkıp 3D görüntü düzlemindeki piksellerin her birinden geçecek olan birincil ışınların doğrultularının belirlenmesi işlemidir. Bunun için piksel koordinatlarından başlangıç noktasının koordinatı çıkarılır. Ardından normalize edilerek doğrultunun boyu 1 birim yapılır. Işınlar ile 3D ortamdaki cisimler arasında kesişim testleri yapılarak görüntü düzlemine hangi cismin şeklinin çizileceği belirlenir. Şekil 1'den görüldüğü gibi ışınların 3D görüntü düzleminde geçtikleri pikseller ile en son ekranda üretilen görüntüdeki pikseller farklı koordinat sistemlerini kullanmaktadır. Örneğin bilgisayar ekranındaki 800x450 çözünürlükteki bir görüntünün (x, y) koordinatları sol üst köşede $(0, 0)$ sağ alt köşede de $(799, 449)$ 'dur. Aynı çözünürlükteki bir görüntü düzleminin sol üst köşesinin koordinatları $(-400, 224, 500)$ olmalıdır (görüntü düzleminin bakış noktasına uzaklığı 500 birim alınmıştır). Dolayısıyla ekrandaki herhangi bir pikselin görüntü düzlemindeki karşılığını bulmak için bir dönüşüm yapmak

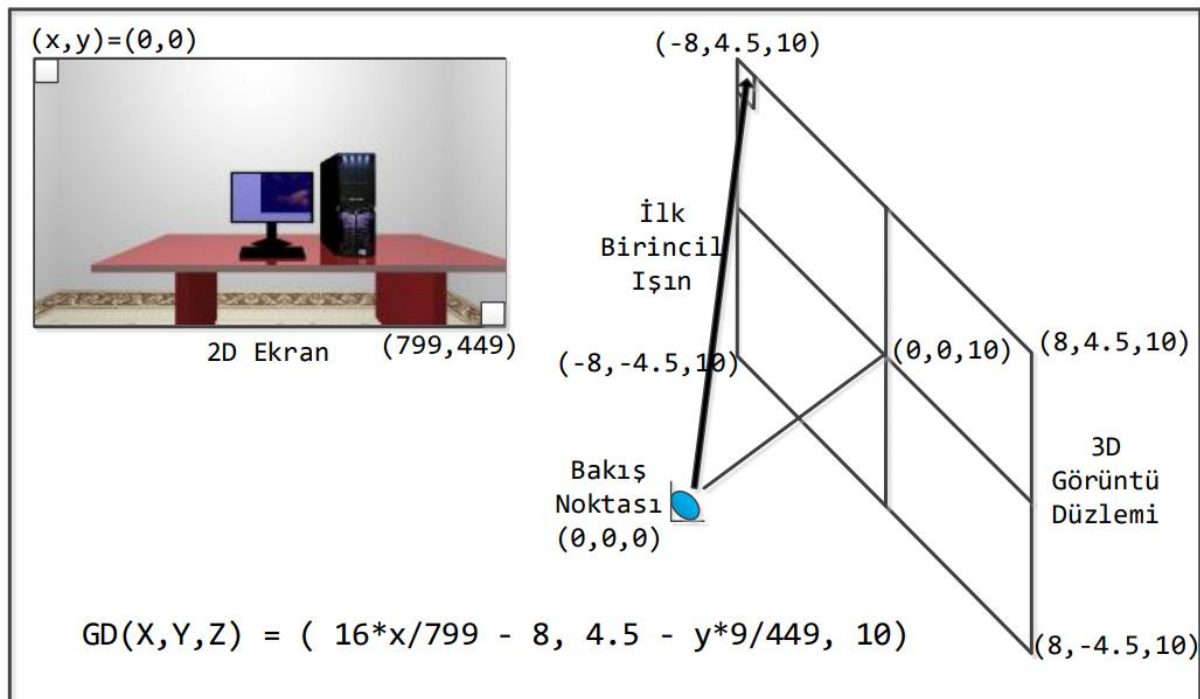
gerekir. Ekrandaki 800x450 çözünürlüğünde bir görüntünün (x,y) koodinatlarının 3D Görüntü Düzlemi GD(X,Y,Z)'deki karşılığı aşağıdaki ifade ile bulunur:

$$GD(X,Y,Z) = (x - 399, 224 - y, 500)$$

Yukarıdaki dönüşümün pratikte kullanılması pek tercih edilmez. Çünkü çizilecek görüntünün çözünürlüğü değiştirilmek istenildiğinde sadece 399, 224 gibi çözünürlüğe bağlı değerleri değiştirmek yeterli olmaz. Görüntü Düzlemine uzaklığın da değiştirilmesi gerekir. Dolayısıyla yalnızca çözünürlüğü değiştirerek dönüşüm imkânı sağlayan aşağıdaki ifade kullanılmalıdır:

$$GD(X,Y,Z) = (16*x/799 - 8, 4.5 - y*9/449, 10)$$

Burada görüntü düzlemi (16x9) boyunda seçilmiştir. Üretilcek görüntünün çözünürlüğü 800x450'den farklı mesela 1600x900 olduğunda sadece 799'u 1599 ve 449'u 899 yapmak yeterlidir.



Şekil 1: 2D Ekran ve 3D Görüntü Düzlemi Arasındaki İlişki

4. Işın-Üçgen Kesişim Testi

3D cisimler çoğunlukla üçgenler ile temsil edilirler. Burada anlatılacak olan ışın-üçgen kesişim testi iki aşamadan oluşmaktadır:

1. Işın ile üçgenin üzerinde olduğu düzlemsel yüzey arasında kesişim testi.

2. Işın yüzey ile kesişiyorsa kesişim noktasının üçgenin içinde olup olmadığını belirleme.

Birinci aşama için üçgenin tanımladığı yüzeyin denklemini çıkarmak gerekmez. Bilindiği gibi yüzey denklemi $Ax+By+Cz+D=0$ 'dır. Burada (A,B,C) yüzey normalidir. Yukarıda V_0, V_1, V_2 şeklinde verilen üçgenin yüzey denklemini çıkaralım:

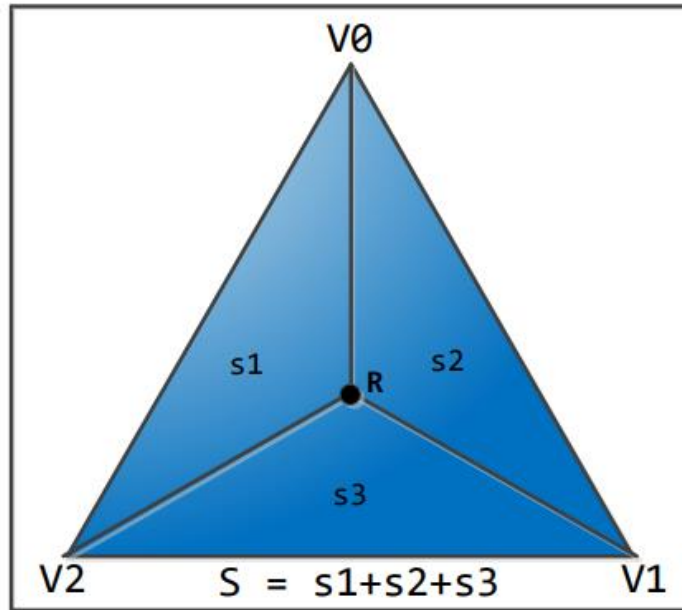
Daha önce üçgenin normali $N=(0,0.6,-0.8)$ olarak hesaplanmıştı. Dolayısıyla $N=(A,B,C)$ biliniyor. Yüzeyin üzerinde olduğu için yüzey denklemini sağlayacağından üçgenin köşe noktalarından herhangi biri D 'nin hesabı için kullanılabilir. Dolayısıyla $Ax+By+Cz+D=0$ 'daki (x,y,z) yerine köşe noktalarından herhangi birinin mesela V_0 'ın (x,y,z) 'sini yazıp sıfıra eşitlersek D 'yi :

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

$$0 \cdot 0 + 0.6 \cdot 40 + -0.8 \cdot 120 + D = 0$$

$$D = 72$$

olarak buluruz. Dolayısıyla yüzey denklemi $0.6y - 0.8z + 72 = 0$ 'dır.



Şekil 2: Alan Testi

Işın yüzey ile kesişiyorsa üçgenin köşe noktaları gibi ışının yüzey üzerindeki koordinatları da yüzey denklemini sağlamalıdır.

Dolayısıyla şöyle yazabiliriz: $A(R_{ox}+tR_{dx}) + B(R_{oy}+tR_{dy}) + C(R_{oz}+tR_{dz}) + D = 0$
Yukarıdaki denklem t 'ye göre düzenlenirse:

$$t = - \frac{AR_{ox} + BR_{oy} + CR_{oz} + D}{AR_{dx} + BR_{dy} + CR_{dz}} = - \frac{N * R_o + D}{N * R_d}$$

$t > 0$ ise ışın yüzey ile kesişiyor demektir.

$t < 0$ ise ışın yüzey ile kesişmiyor demektir.

$t = \infty$ ise yani $N * R_d = 0$ ise ışın yüzeye paralel demektir.

Örnek olarak $R_o=(0,0,0)$ başlangıç noktasından $R_d=(0,0,1)$ doğrultusu boyunca giden R ışınının bu yüzey ile kesişip kesişmediğini t hesabı ile belirleyelim:

$$t = - \frac{N * R_o + D}{N * R_d} = - \frac{72}{-0.8} = 90$$

Işının yüzey üzerindeki koordinatları $R=R_o+tR_d=(0,0,0)+90(0,0,1)=(0,0,90)$ olarak bulunur. Böylece kesişim testinin I. Aşaması tamamlanmış oldu.

II. aşamada $(0,0,90)$ noktasının üçgenin içinde olup olmadığına karar verilmelidir. Bunun için değişik yöntemler denenebilir. Burada “alan testi” yöntemi kullanılacaktır. Buna göre Şekil 2’den de görüldüğü gibi yukarıda hesaplanan kesişim noktasından üçgenin köşelerine doğrular çizerek 3 alt üçgen oluşturulur. Bu alt üçgenlerin alanları toplamı büyük üçgenin alanına eşitse kesişim noktası üçgenin içinde demektir.

V_0, V_1, V_2 üçgeninin alanı $|(V_1-V_0) \times (V_2-V_0)|$ ile hesaplanabilir. $R=(0,0,90)$ noktası kullanılarak oluşturulan s_1, s_2 ve s_3 alt üçgenlerin alanları ve V_0, V_1, V_2 üçgeninin S alanını hesaplandığında $S=6000$, $s_1=1500$, $s_2=1500$ ve $s_3=3000$ çıkar. $S=s_1+s_2+s_3$ olduğundan kesişim noktası üçgenin içindedir.

```

float Intersect(Vertex Ro, Vertex Rd)
{
    Vertex normal;
    Vertex R;
    float S, s1, s2, s3;

    normal = (V1-V0).CrossProduct(V2-V0);

    float D = -(V0 * normal);
    float t = -(Ro * normal + D) / (normal * Rd);

    if (t > 0)
    {
        R = Ro + t * Rd;

        S = (V1 - V0).CrossProduct(V2 - V0).Length();
        s1 = ( R - V0).CrossProduct(V2 - V0).Length();
        s2 = (V1 - V0).CrossProduct( R - V0).Length();
        s3 = (V1 - R).CrossProduct(V2 - R).Length();

        float fark = (float)Math::Abs(S - (s1 + s2 + s3));
        float epsilon = 0.005F;

        if (fark <= epsilon) return t; else return 0;
    }
    else return 0;
}

```

5. Görünmeyen Yüzeylerin Kaldırılması (Hidden Surface Removal)

Görüntü düzleminde geçen ışıklardan herhangi biri gittiği doğrultu boyunca 1'den fazla cisimle kesişiyorsa ekrana bunlardan görüntü düzlemine en yakın olanı çizilmelidir. Bunun için kesişim testleri ile hesaplanan t uzaklıkları sıralanır ve en küçük t uzaklığına sahip cismin görüntüsü çizilir. Diğer cisimler görünmeyen yüzey olarak adlandırılırlar.

$U0(0, 30, 40)$	$U1(40, -30, 120)$	$U2(-40, -30, 120)$
$Y0(-50, 30, 124)$	$Y1(50, 30, 124)$	$Y2(0, -30, 44)$
$Z0(-30, 0, 37)$	$Z1(30, 40, 117)$	$Z2(30, -40, 117)$

Yukarıda sırasıyla kırmızı, yeşil ve mavi renklere sahip $U0, U1, U2$ üçgeni $Y0, Y1, Y2$ üçgeni ve $Z0, Z1, Z2$ üçgeninin köşe noktalarının koordinatları verilmiştir. Başlangıç noktası $R_0(0,0,0)$ 'dan çıkan ve görüntü düzleminde $(0,0,10)$ noktasındaki pikselden geçen ışın ile bu üçgenler arasında ışın-üçgen kesişim testi yöntemine göre $t_u=80$, $t_v=84$, $t_z=77$ uzaklık değerleri hesaplanır. Uzaklıklar sıralandığında mavi renkli Z üçgeninin bakış noktasına daha yakın olduğu görülür. Dolayısıyla ışının geçtiği piksel mavi renge boyanır. Kırmızı renkli U ve yeşil renkli Y üçgenleri görüntü düzleminde $(0,0,10)$ noktasındaki pikselden görünmeyen üçgenlerdir.

7. Phong Boyama Modeli

Işıklar ile üçgenler/kürelerle kesişim testleri yapıp görüntü düzlemine en yakın olan üçgen/küre belirlendiğinde bu üçgenin/kürenin rengi doğrudan ilgili piksele setlenmez. Çünkü ışık kaynağının o noktayı ne oranda aydınlatmış olduğu dikkate alınmalıdır. Phong boyama modeline göre ışık kaynağına bağlı olarak ambient, diffuse ve specular olmak üzere üç renk bileşeni, toplamı 1 olan katsayılarla çarpılıp toplanarak pikselin son rengi belirlenir.

7.1. Diffuse Renk Bileşeni

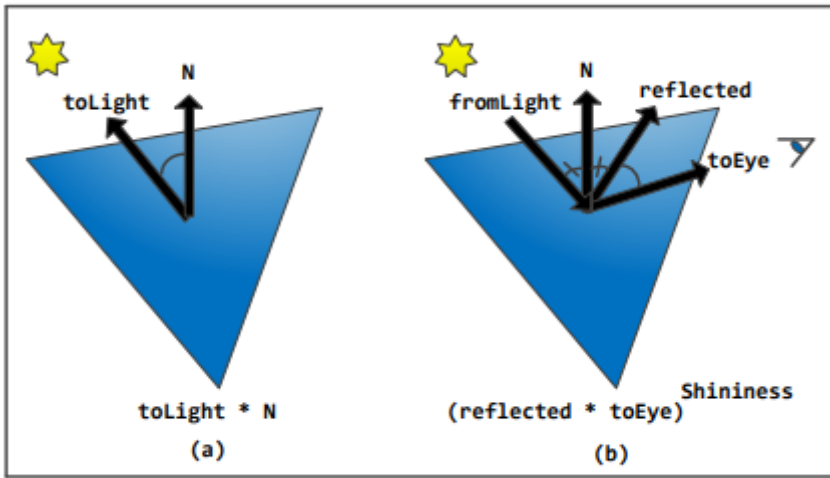
Diffuse katsayı Şekil 4-(a)'daki gibi yüzey normali ile ışık kaynağına doğru olan vektör skaler çarpılarak belirlenir. Her ikisinin de birim vektör olduğu varsayıldığında skaler çarpımları aralarındaki açının kosinüsünü verir. Negatif değerler için yüzey ışık kaynağı tarafından aydınlatılmıyor demektir. Işık kaynağı yüzeye tam dik vuruyorsa yani yüzey normali ile aralarındaki açı 0 derece ise $\cos(0)=1$ olduğundan maksimum aydınlatma; açı 90 derece ise de $\cos(90)=0$ minimum (sıfır) aydınlatma söz konusudur. Hesaplanan diffuse katsayı ile yüzeyin rengi çarpılarak Phong boyama modelinin diffuse renk bileşeni belirlenir.

Işık kaynağının koordinatları $(0,40,0)$ olduğu durumda yüzey normali $(0,1,0)$ olan kırmızı renkli bir üçgen üzerindeki $(0,0,30)$ noktasının diffuse renk bileşenini

hesaplayalım: Işık kaynağına doğru olan vektör $(0,40,0)-(0,0,30)=(0,40,-30)$ 'dur. Normalize edildiğinde $(0,0.8,-0.6)$ olur. Diffuse katsayı $(0,0.8,-0.6)*(0,1,0)=0.8$ çıkar. Kırmızı renk $(255,0,0)$, hesaplanan diffuse katsayı ile çarpılırsa diffuse renk bileşeni $(204,0,0)$ olarak bulunur.

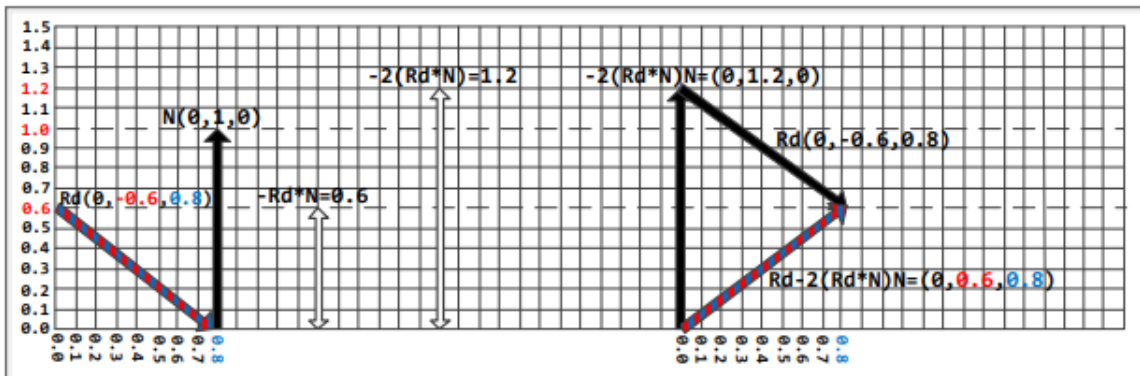
7.2. Specular Renk Bileşeni

Specular katsayı Şekil 4-(b)'deki gibi ışık kaynağından yüzeye doğru olan vektörün yüzey normalinden yansıma vektörü ile bakış noktasına doğru olan vektör arasındaki açının belli bir skaler değer kadar kuvveti alınarak hesaplanır. Hesaplanan specular katsayı ile ışık kaynağının rengi çarpılarak Phong boyama modelinin specular renk bileşeni belirlenir.



Şekil 4: Diffuse ve Specular Katsayı Hesabı

Şekil 4-(b)'deki specular katsayı hesabındaki reflected yansıma vektörünün hesaplanması Şekil 5'te gösterilmiştir. Burada gelen ışının doğrultusu R_d , yüzey normali N , olduğu durumda yansıma vektörü $R_d - 2(R_d \cdot N)N$ ile hesaplanır.



Şekil 5: Yansıma Vektörü Hesabı

7.3. Ambient Renk Bileşeni

Bir odada masa sehpa gibi eşyaların altları gibi ışık kaynağı tarafından doğrudan aydınlatılmayan yüzeyler vardır. Buralara duvarlardan veya diğer eşyalardan yansıyan ışıkların vurması nedeniyle az da olsa aydınlatılmaktadırlar. Phong modelinin ambient renk bileşeni bu dolaylı aydınlatmayı modeller ve cismin kendi rengi 0..1 arası bir katsayı ile çarpılarak ambient renk bileşeni hesaplanır.

8. Yansıma (Reflection), Saydamlık (Transparency) ve Kırılma (Refraction)

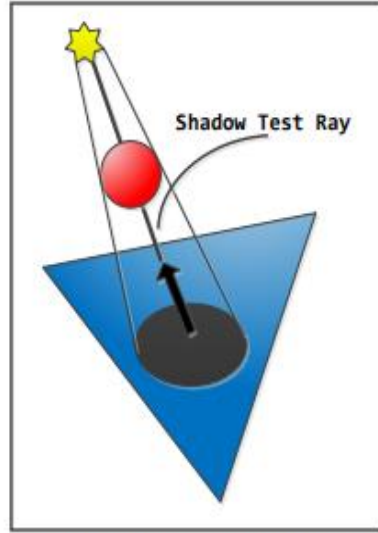
Işını yansıtan yüzeyler için yansıyan ışının doğrultusunun (reflected direction) $R_d = 2R_d \cdot N \cdot N$ ile hesaplanacağından specular bileşene ait reflected vektörü hesaplanırken bahsedilmişti. Işını yansıtma özelliğine sahip cisim için Şekil-5'teki gibi yansıma doğrultusu hesaplandıktan sonra yeni doğrultu boyunca yollanan ışın ile cisimler üzerinde kesişim testleri yapılarak en yakın cisim belirlenir. Onun rengi 0..1 arası bir katsayı ile çarpılarak yansıma ile görülen cismin rengi hesaplanır. Saydam cisimlerde ışın cismin içinden geçerken doğrultusu değişmez. Yani cismin içinden geçen ışının doğrultusu (transmitted direction) gelen ışının doğrultusu R_d ile aynı alınıp yukarıdaki işlemler tekrarlanır. Işın saydam bir küreden geçerken (ışının başlangıç noktasının kürenin dışında olduğu varsayıldığından) küreyle önce dıştan sonra da içten kesişerek yoluna devam eder. Işın küreyle dıştan kesiştikten sonra aynı doğrultu boyunca içten kesişecek şekilde tekrar yollandığında ışınküre kesişim testi t uzaklığı hesabında

```
if (l2 > r2) return s - q;  
else return s + q;
```

ışının başlangıç noktası küre üzerinde yani $l2 = r2$ olduğundan normalde `else` kısmının koşması beklenir ama ışın-üçgen alan testindeki gibi $l2$ ve $r2$ `float` sayılar olduğundan çoğu zaman birebir aynı olmazlar. Bazen $l2 > r2$ bile olabilir ve $t = s - q$ olarak hatalı hesaplanacağından küre üzerinde gürültülere sebep olur. Yukarıdaki kodun `else` kısmının koşmasını garanti etmek için `if (l2 > r2 + 0.001)` olarak güncellenir. `if ((int)l2 > (int)r2)` de aynı işlevi görür.

9. Gölge Testi

Herhangi bir yüzeyin başka bir yüzeyin gölgesinde kalıp kalmadığını belirlemek için Şekil 6'da gösterildiği gibi yüzey üzerindeki kesişim noktasından ışık kaynağına doğru gölge test etme ışını yollanır. Bu ışın ile üçgenler/küreler arasında kesişim testleri yapılarak en yakın üçgen/küre belirlenir. Hesaplanan uzaklık değeri ışık kaynağına olan uzaklıktan küçük ise yüzey bu üçgenin/kürenin gölgesinde kalıyor demektir.



Şekil 6 : Gölge Test Etme Işını

10. Düzlemsel Yüzey Üzerine Doku Kaplama

Işın izleme ile doku kaplama için farklı yöntemler kullanılabilir. Burada Tomas Möller'in ışın-üçgen kesişim testinden faydalanılarak doku kaplama anlatılacaktır. Möller'in yöntemine göre kesişim testi yapıldığında yalnızca kesişim noktasına uzaklık olan t değeri değil aynı zamanda kesişim noktasının barisentrik koordinatları da hesaplanabilir. Barisentrik koordinatlar ve üçgenin köşe noktaları kullanılarak üçgen üzerinde herhangi bir noktaya gidebilmek için aşağıdaki ifade kullanılır:

$$t(u,v) = V_0 + u(V_1-V_0) + v(V_2-V_0)$$