

A számítógépes grafika alapjai

Színek és megvilágítás

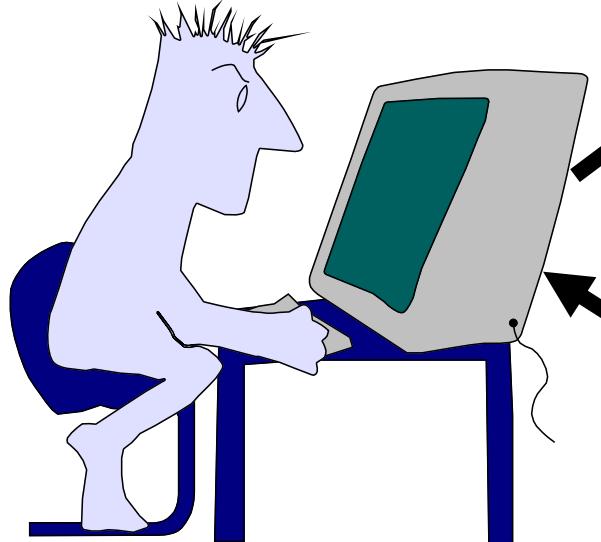
Előadó: Benedek Csaba

Tananyag : Szirmay-Kalos László, Benedek Csaba



Emlékeztető....

illúzió



Képpontok:

- vörös
- kék
- zöld

modellezés

képszintézis

Metafórák:

- optika
- 2D rajzolás
- festés
- analógiák

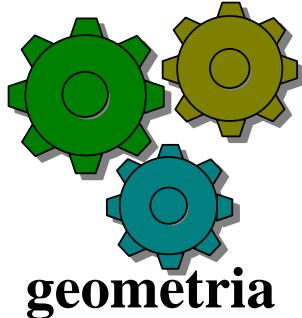
számok

Virtuális világ
modell

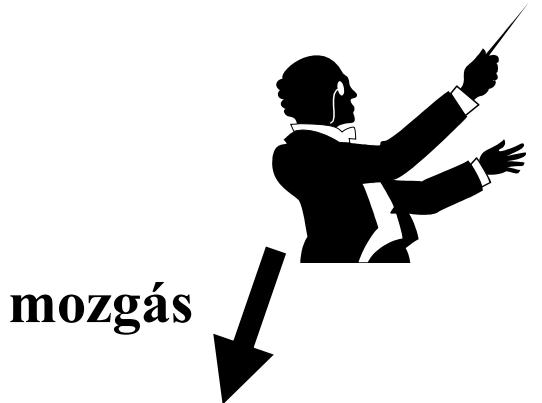
számítás

mérés

Modellezés



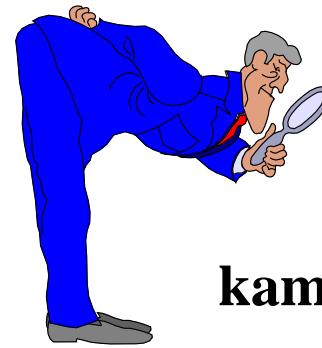
modell



stúdió
objektumok

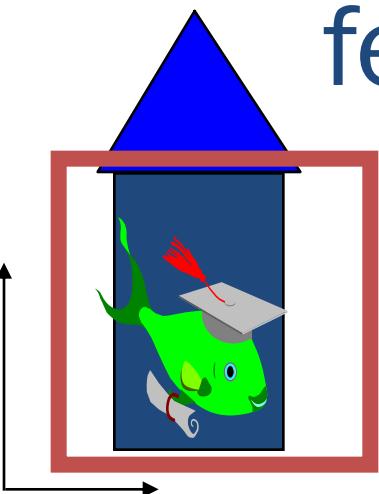


Szín, anyag - most
(textúrák : jövő órán)

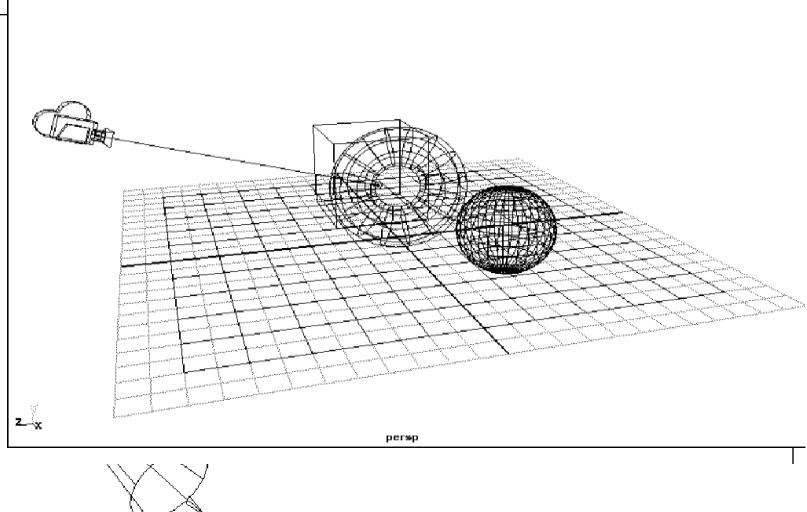


kamera

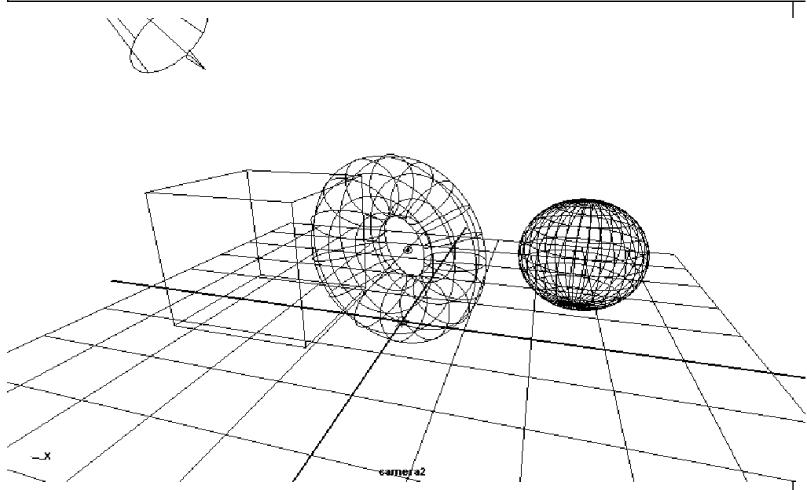
Képszintézis feladatai



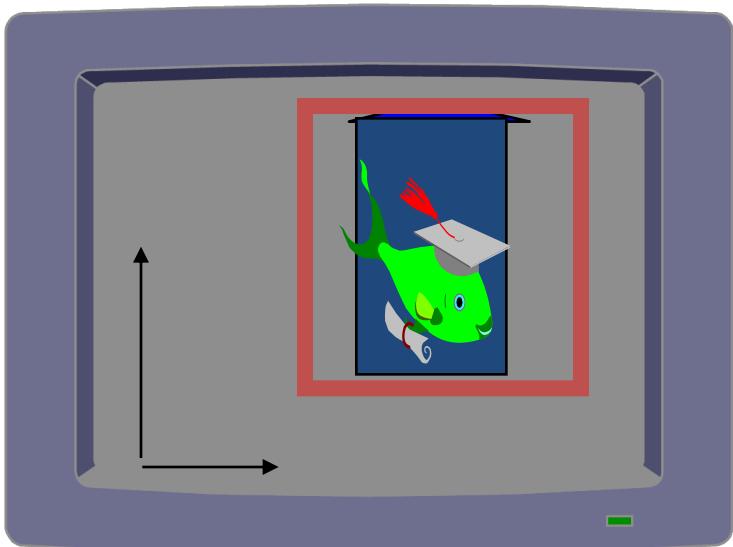
Vektorizáció



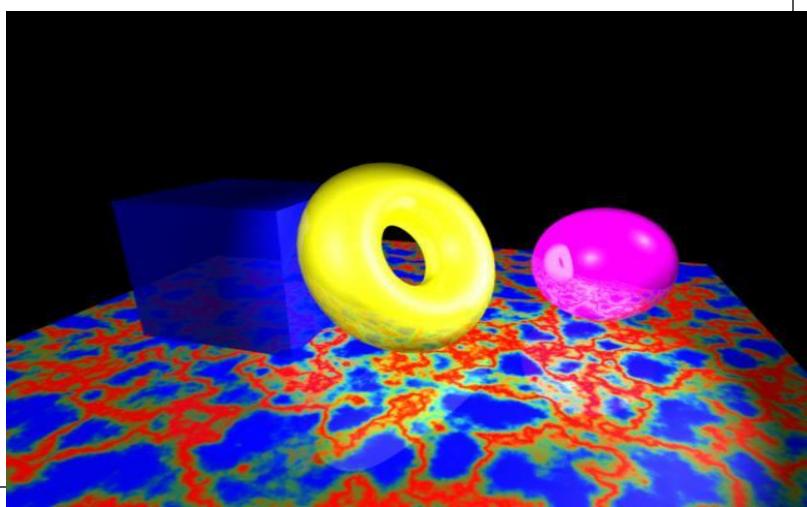
Transzformációk



Vágás



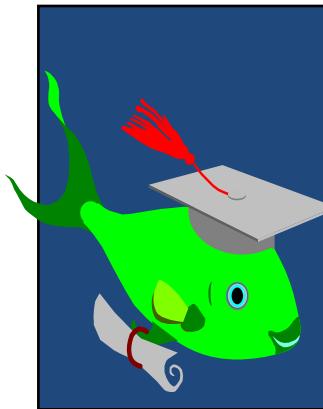
Takarás



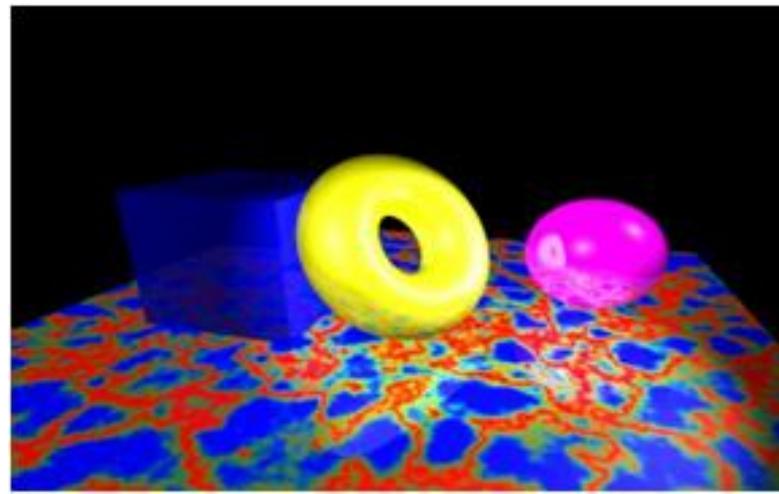
Árnyalás

Árnyalás

- A látszódó objektum adott képpontban érvényes színének meghatározása
 - 2-D grafika: saját szín alkalmazása
 - saját anyag + a térben fennálló fényviszonyok együttes függvénye



Zöld hal

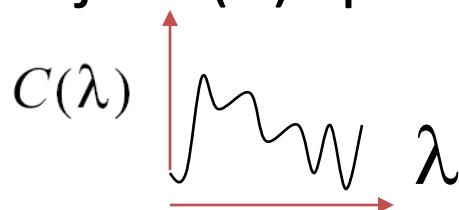


„Sárga” tórusz

Színkezelés alapjai a 3D grafikában

Fotometriai alapok

- Fény = elektromágneses hullámok szuperpozíciója
- Fényforrás jellemzője: $C(\lambda)$ spektrális energia eloszlás



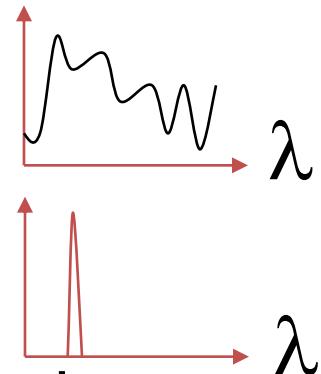
- időegységre eső kibocsátott energia (tehát: teljesítmény)
„sűrűsége” az adott hullámhosszon (analógia: folytonos
valószínűségi változó sűrűségfüggvénye)
- A fényforrás által kibocsátott összes teljesítmény:

$$P = \int_0^{\infty} C(\lambda) d\lambda$$

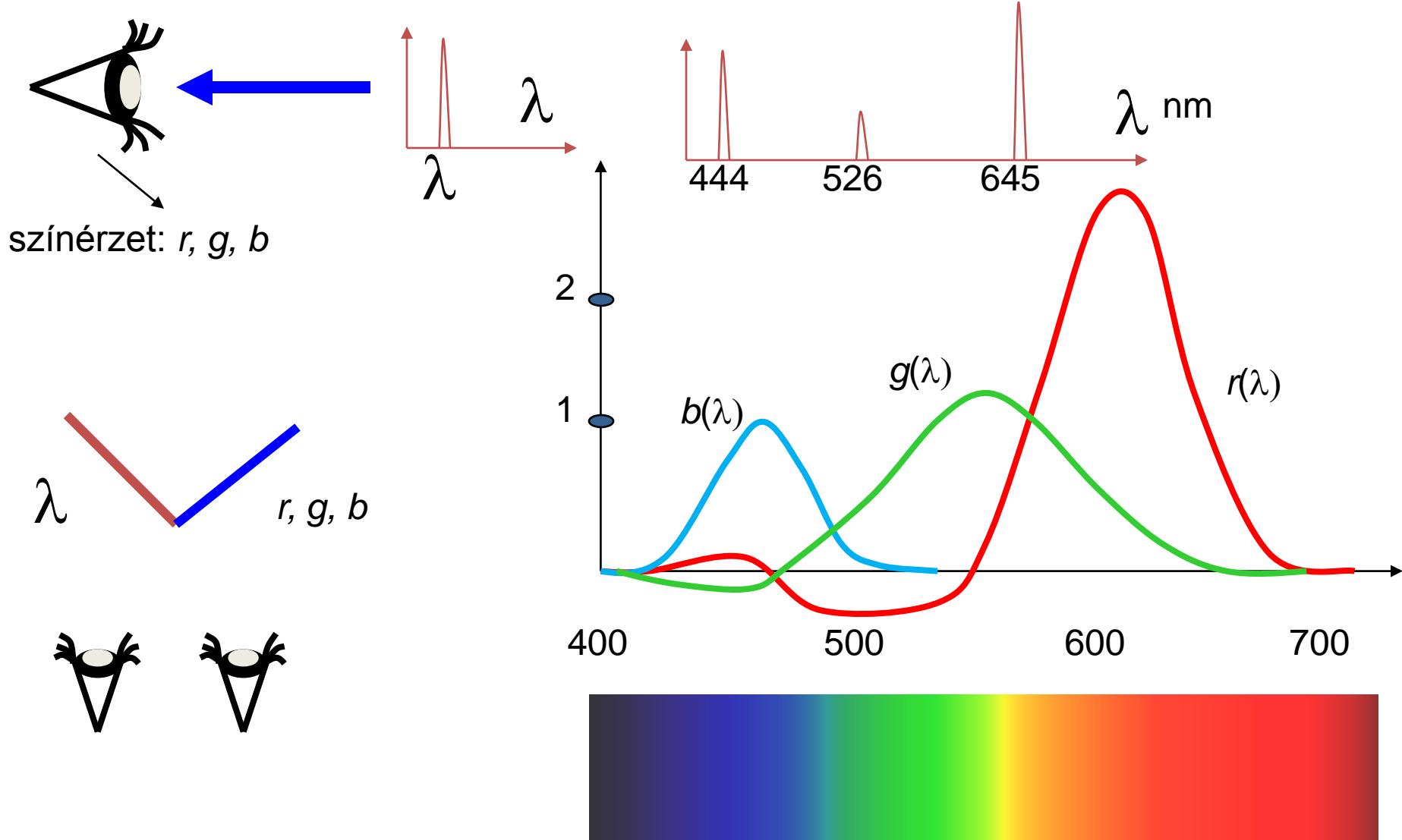
Színelméleti alapok

- Fény - Színérzet

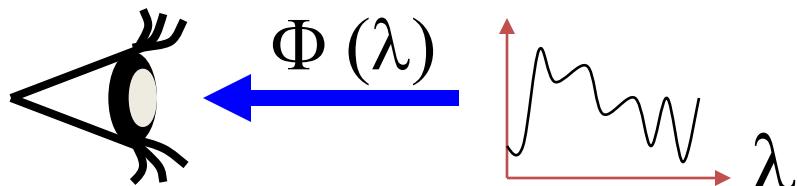
- Összetett fény: „sok hullámhossz” 
- Monokromatikus fény: csak egyfélé hullámhosszú összetevő 
- Érzékelés 3 érzékelővel: tristimulus értékek
 - ugyanazt a színérzettel többféle keverék-fénynyaláb is adhatja



Színérzékelés: monokromatikus fény

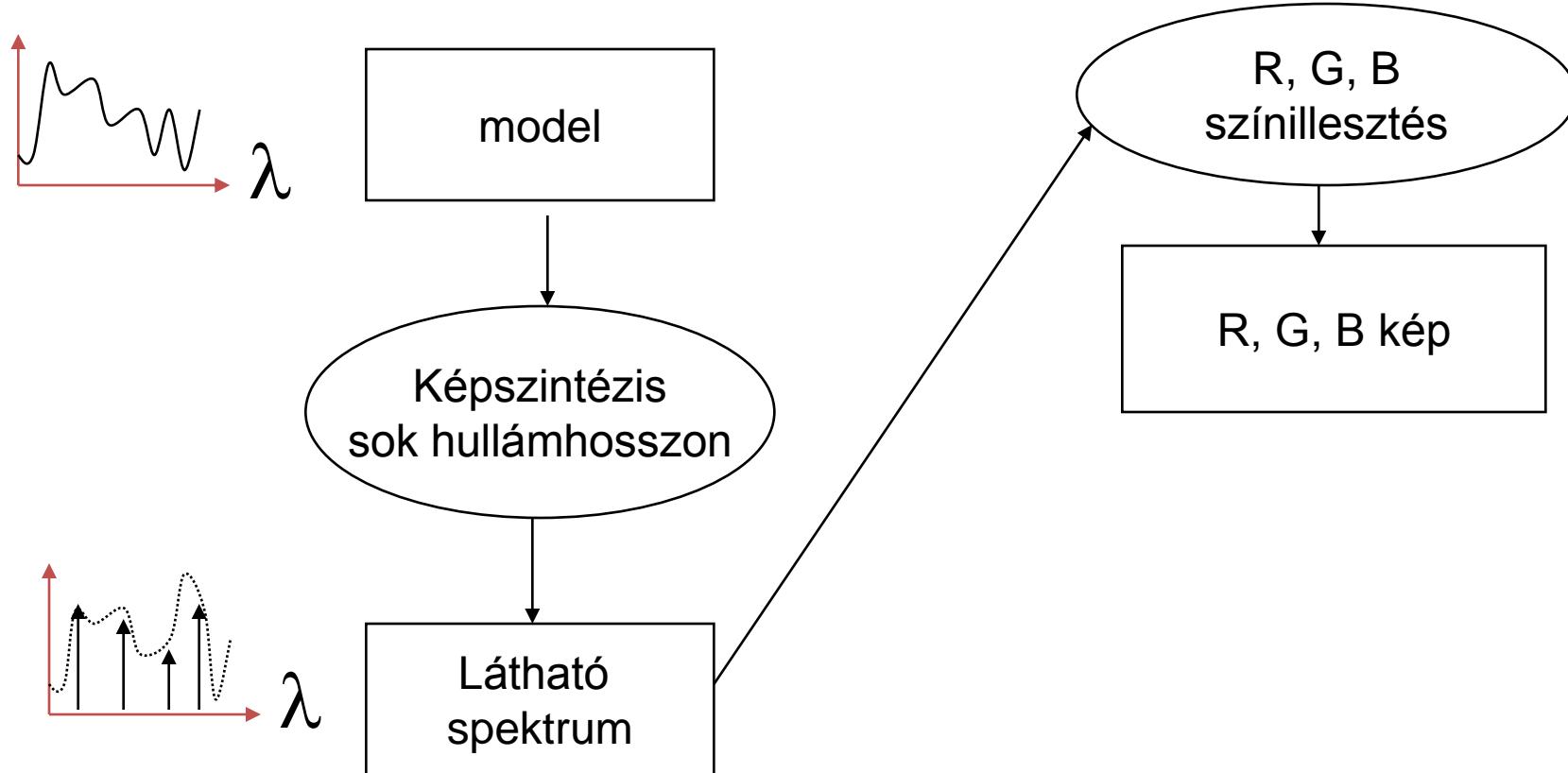


Színérzékelés: nem monokromatikus fény

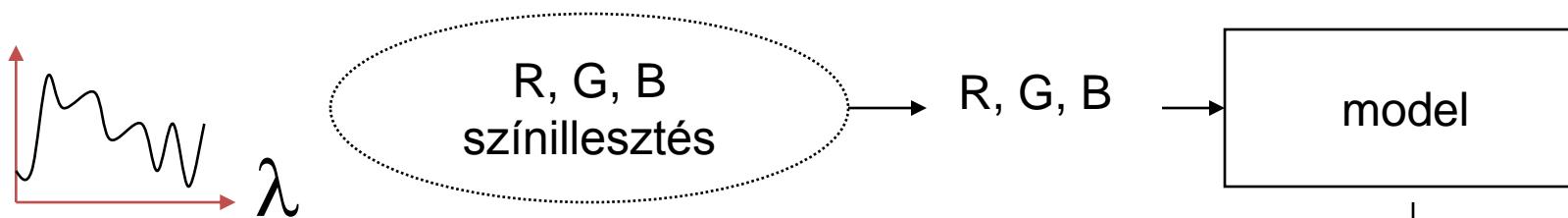


$$\begin{aligned}r &= \int \Phi(\lambda) r(\lambda) d\lambda \\g &= \int \Phi(\lambda) g(\lambda) d\lambda \\b &= \int \Phi(\lambda) b(\lambda) d\lambda\end{aligned}$$

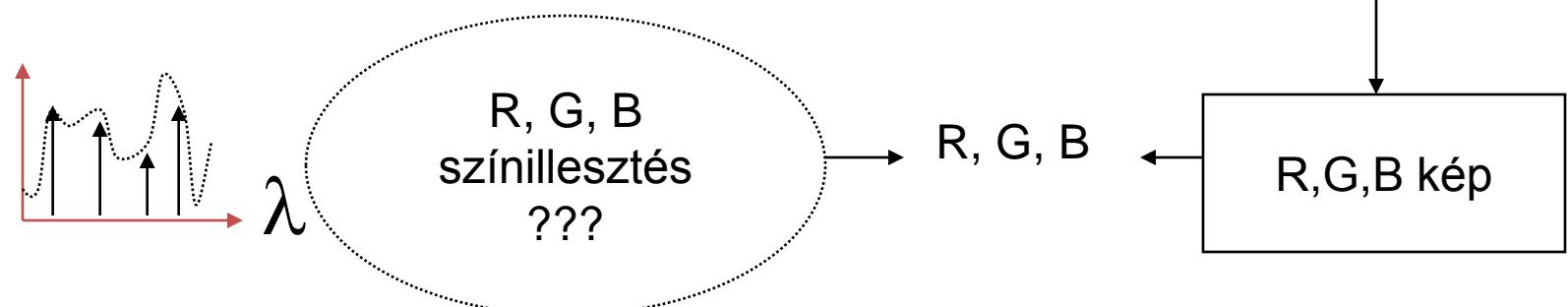
Spektrális színkezelés



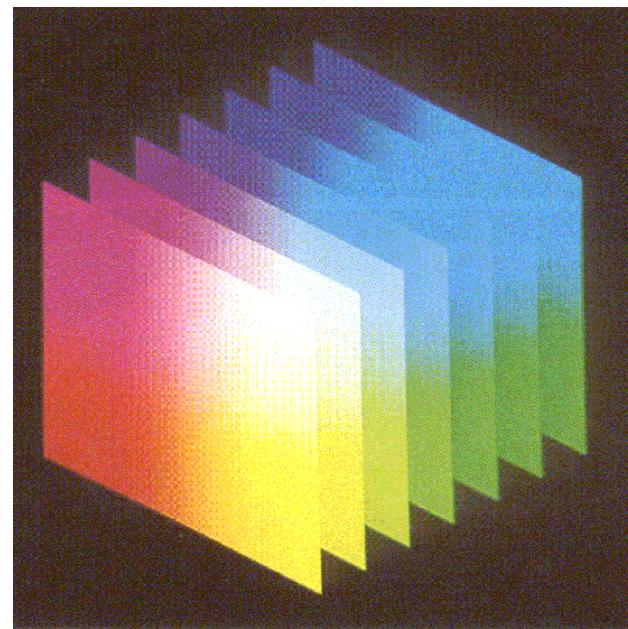
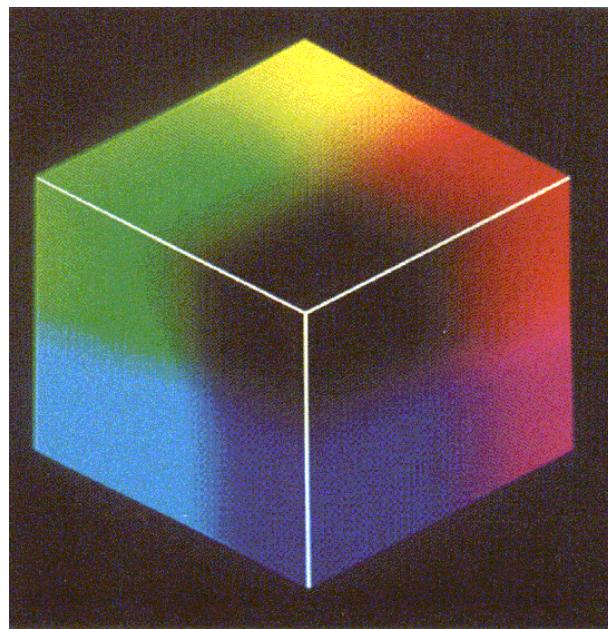
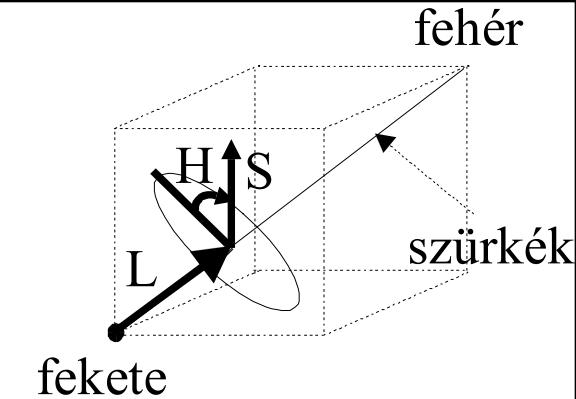
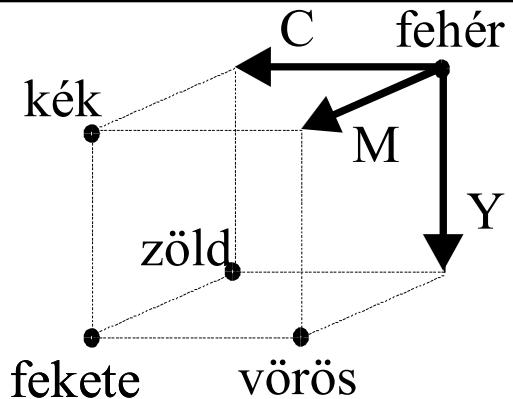
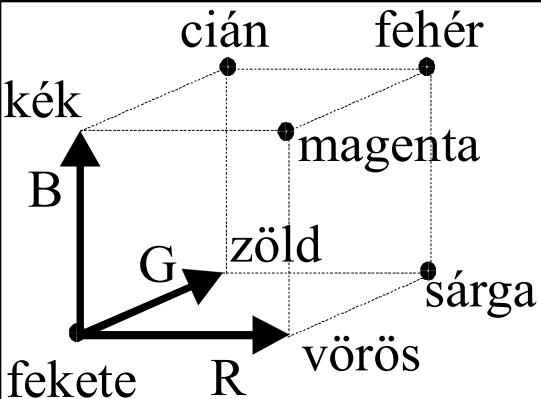
Nem spektrális színkezelés



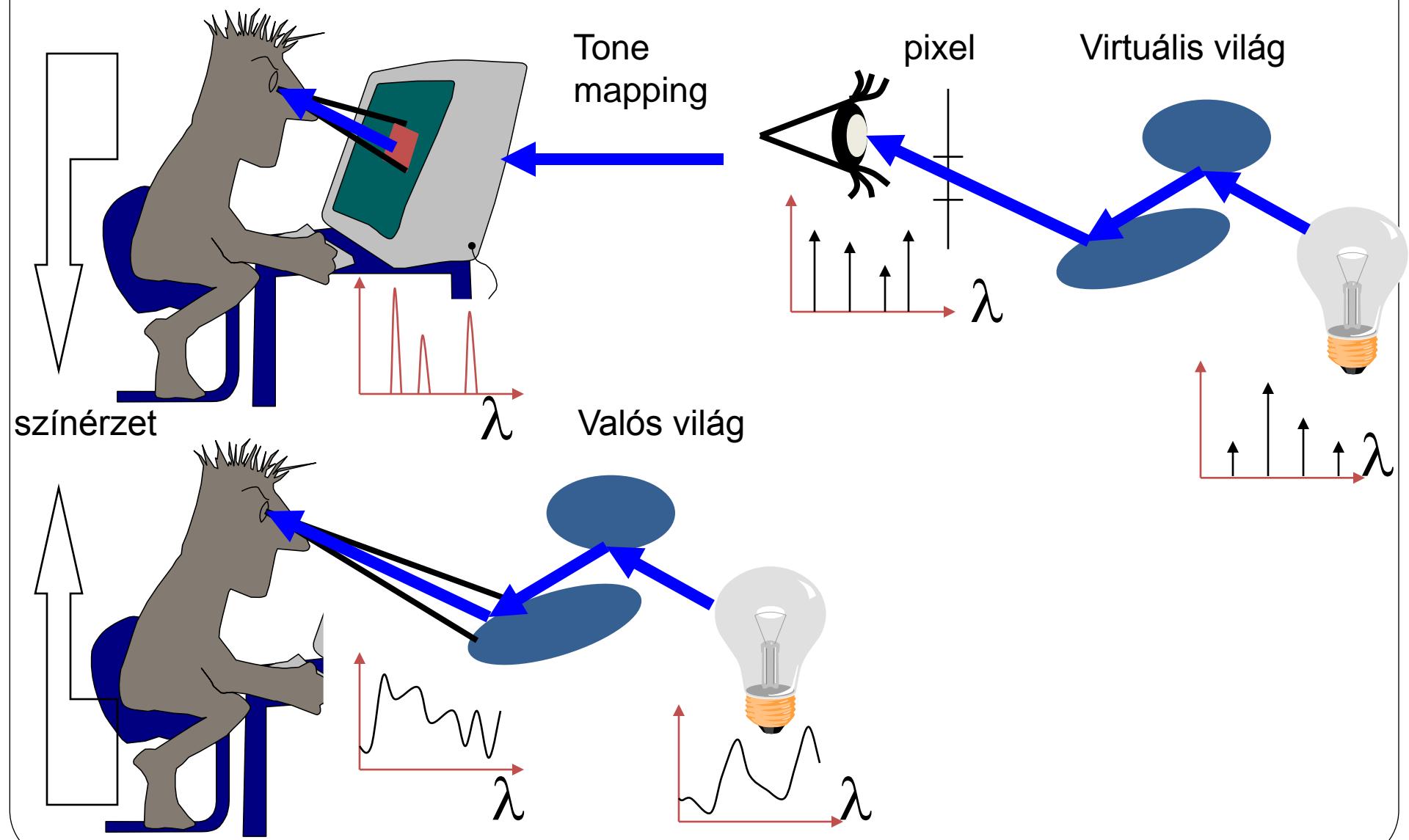
Akkor pontos, ha képszintézis „lineáris”



Színek definiálása



Képszintézis = valós világ illúziója



Anyagok a 3D grafikában

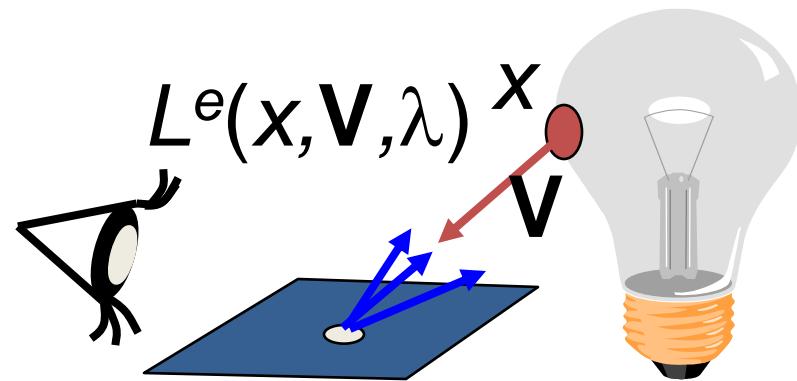
- Emittáló anyagok - fényt kibocsátanak ki
 - „fényforrás”
- Diffúz felületek ~ „matt” anyagok
 - minden irányban ugyanúgy verik vissza a fényt
- Spekuláris vagy tükröző felületek
 - a beérkező fénysugár energiájának jelentős részét az ideális visszaverődési irány környezetébe verik vissza
 - „csillogó” vagy „polírozott” anyagok
- Átlátszó felületek
 - A beérkező fény energiáját minimális veszteséggel eresztik át
- Áttetsző felület
 - „homályosan” látunk keresztül rajtuk
- Összetett anyagok
 - A fentiek „keveréke”- legtöbb valós anyag ilyen

Fényforrások

- Pontszerű fényforrás
 - Intenzitás a távolság négyzetének arányában csökken
- Irány fényforrás
 - Irány és intenzitás a tér minden pontjában azonos
- Ambiens fényforrás
 - minden pontban és irányban azonos intenzitású
- Szpotlámpa
 - Jellemzői: helye, iránya, kúpos hatóterület
- Égbolt fény

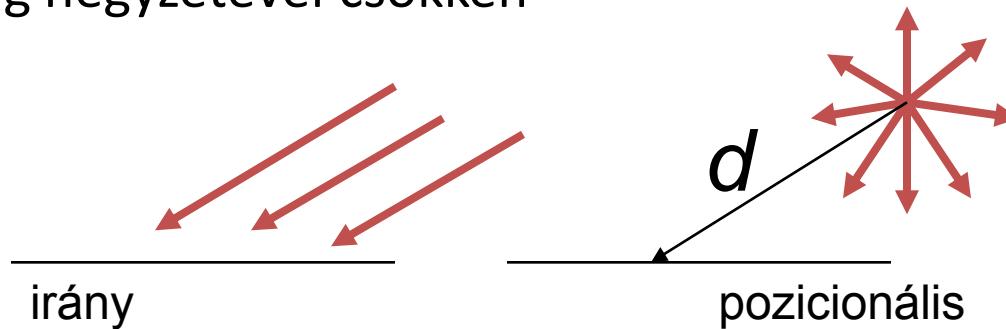
Fényforrás modellek

- **Geometria+sugárúság:**



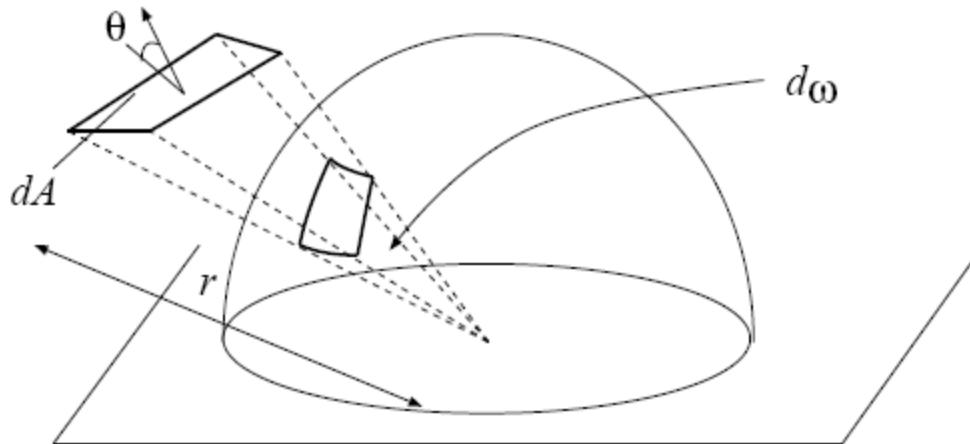
- **Absztrakt fényforrások:**

- **Irány fényforrások:** egyetlen irányba sugároz, a fénysugarak párhuzamosak, az intenzitás független a pozíciótól
- **Pozicionális fényforrás:** egyetlen pontból sugároz, az intenzitás a távolság négyzetével csökken



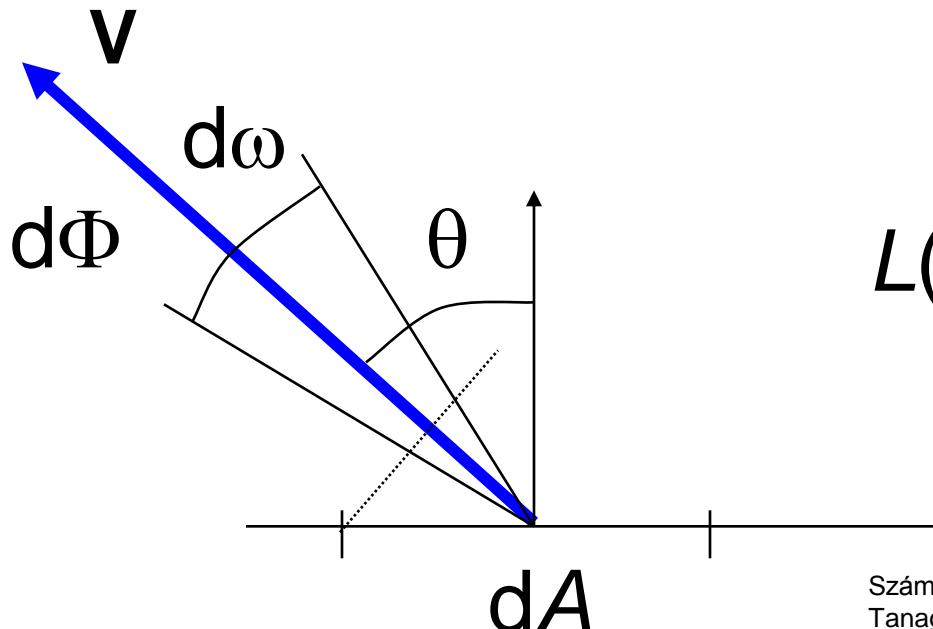
Térbeli irányok definíciója

- 2D szög = az egységkör ívének hossza
 - tartalmazott irányok a körcikk ívének csúcsába mutatnak
- 3D szög (térszög): az egységgömb felületének a része
- Felületelem láthatósági szöge $d\omega = dA \cos\theta / r^2$



Fény erősség jellemzése

- **Fluxus F:** egységnyi idő alatt, adott hullámhossz tartományban átadott energia [mértékegység: W Watt]
- **Radiancia (intenzitás), L:** dA felületelemet $d\omega$ szögben elhagyó dF fluxus osztva a kilépési irányból látható terüettel és a térszöggel
 - Egy egységnyi látható felület által egységnyi térszög alatt kibocsátott teljesítmény [Watt/ sr/ m²]



$$L(x, V) = \frac{d\Phi}{dA \cos\theta d\omega}$$

Fény-felület kölcsönhatás

BRDF: Bi-directional Reflection Distribution Function

Sugársűrűség = Bejövő

BRDF

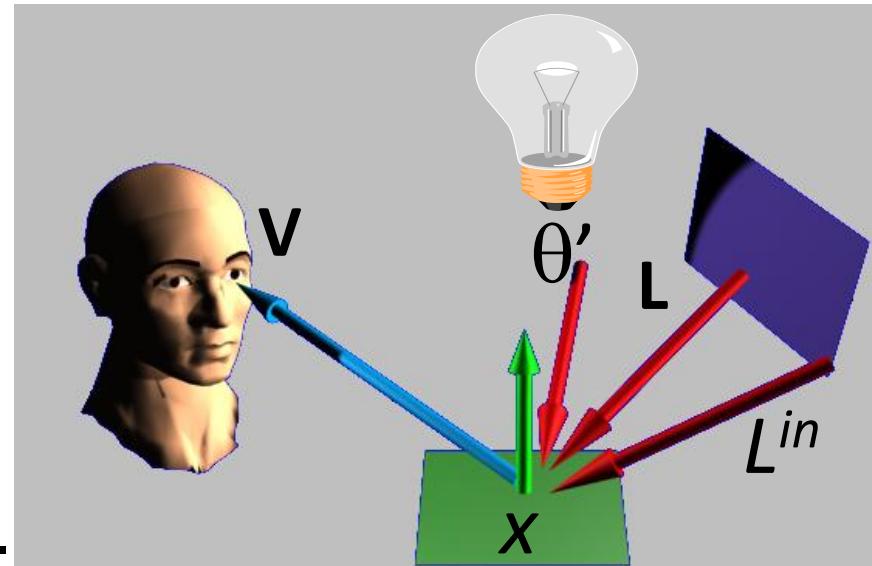
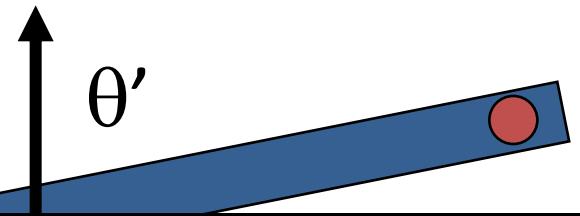
Geometria

$$L(x, \mathbf{V}) = L^{in}(x, \mathbf{L}) \cdot f_r(\mathbf{L}, x, \mathbf{V}) \cdot \cos\theta'$$

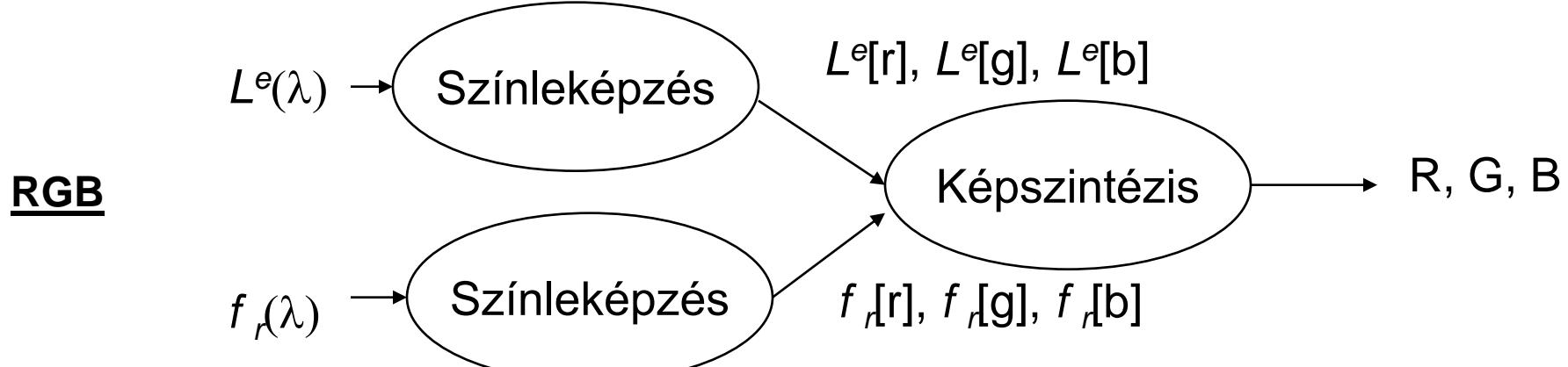
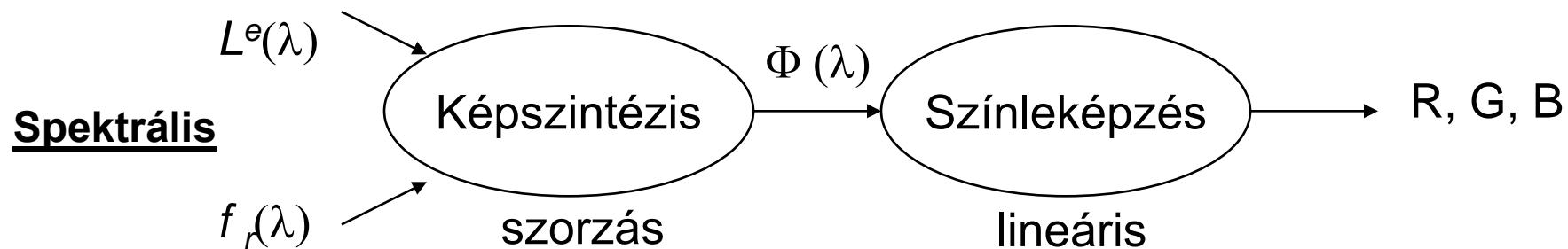
$$f_r(\mathbf{L}, x, \mathbf{V}) = \frac{L(x, \mathbf{V})}{L^{in}(x, \mathbf{L}) \cos\theta'}$$

Helmholtz törvény:

$$f_r(\mathbf{L}, x, \mathbf{V}) = f_r(\mathbf{V}, x, \mathbf{L})$$



Spektrális versus RGB képszintézis



Anyagmodellek- diffúz anyagok

- Radiancia = Bejövő · BRDF · cosθ'

$$L_I(x, \mathbf{V}, \omega) = L_I^{in}(x, \mathbf{L}) \cdot f_{r, I}(\mathbf{L}, x, \mathbf{V}) \cdot \cos\theta'$$

- Diffúz anyag: a BRDF a nézeti iránytól független

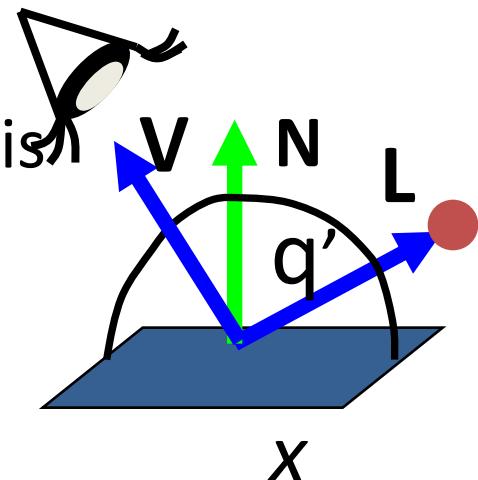
- Helmholtz: a BRDF megvilágítási iránytól is független

- A BRDF

$$f_{r, I}(\mathbf{L}, x, \mathbf{V}) = k_{d, I}(x)$$

- Diffúz visszaverődés = nagyon rücskös

- sokszoros fény-anyag kölcsönhatás
- Színes - I: pl R, G és B komponensekre külön számítás

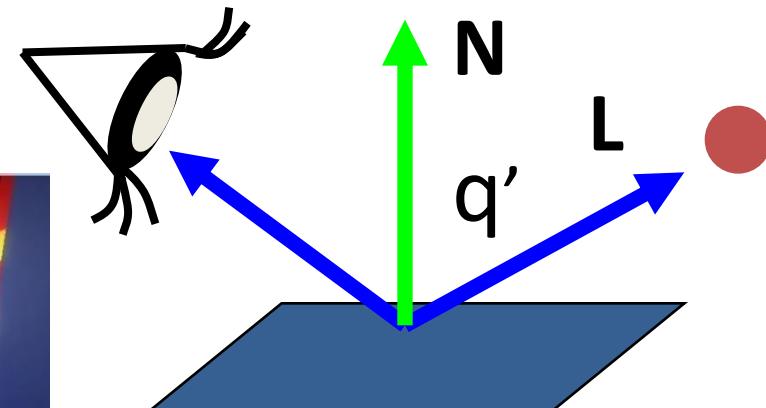
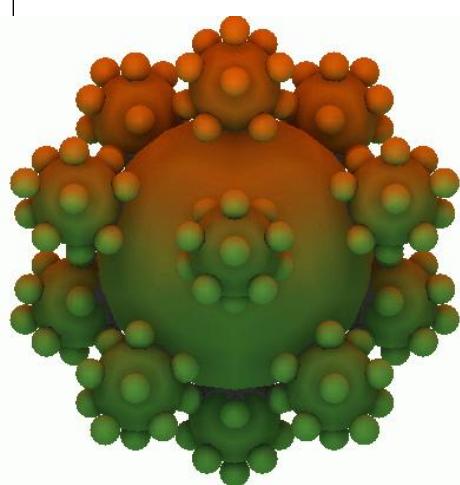


Lambert törvény

- Pont/irány fényforrásra válasz
 - BRDF irányfüggetlen,
DE a sugársűrűség függ a
megvilágítási iránytól

$$L^{ref} = L^{in} k_d \cos^+ \theta'$$

$$\cos \theta' = \mathbf{N} \cdot \mathbf{L}$$



Sima felületek

Fresnel egyenletek

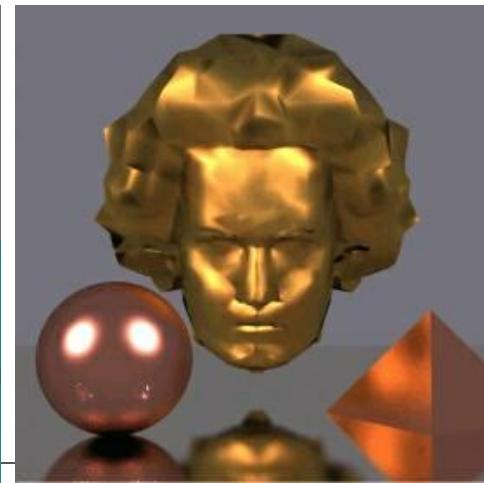
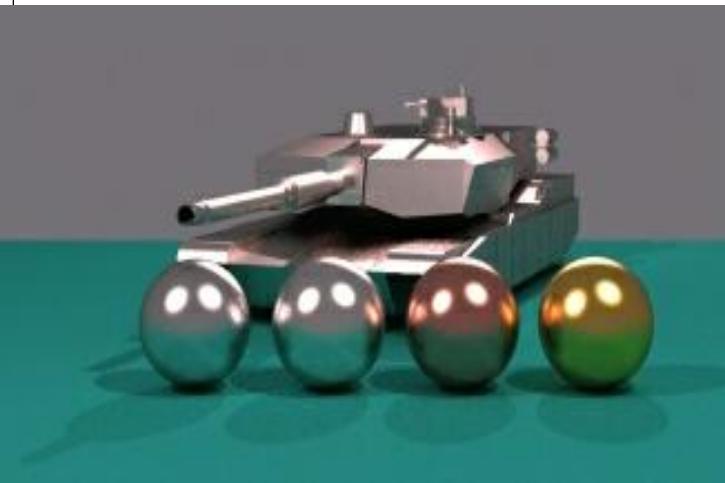
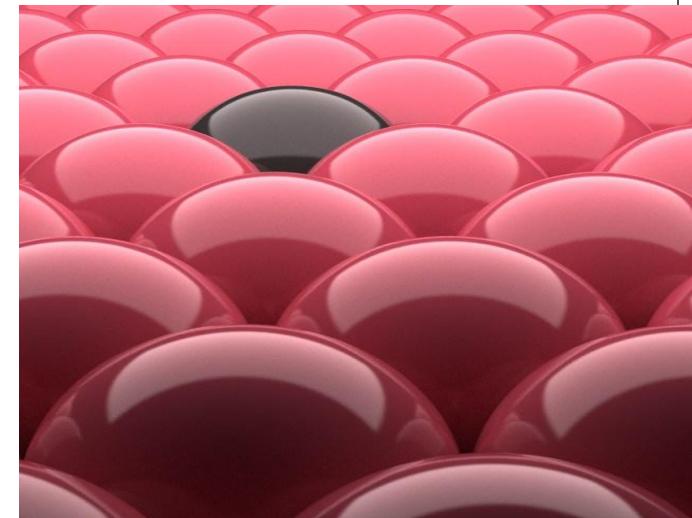
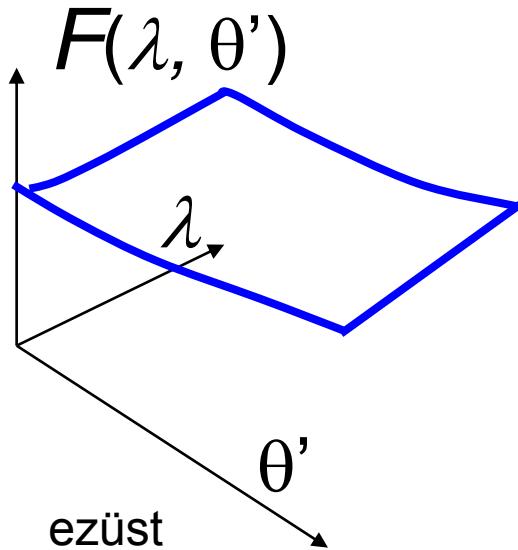
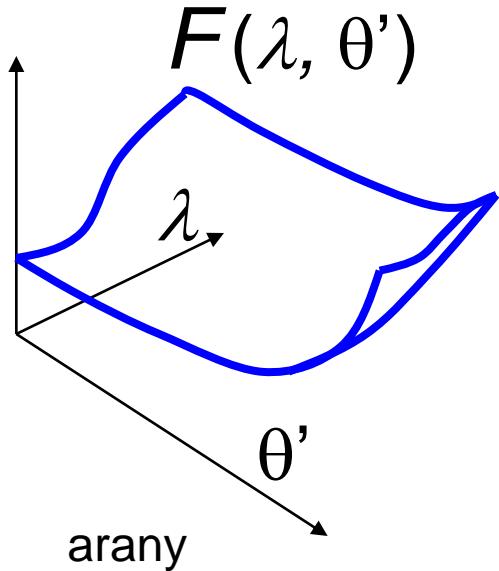
$$L^{in} \quad \theta' \quad | \quad \theta' \quad L^{in} \cdot F$$
$$n = \frac{\sin \theta'}{\sin \theta} \quad | \quad \theta \quad L^{in} \cdot (1-F)$$



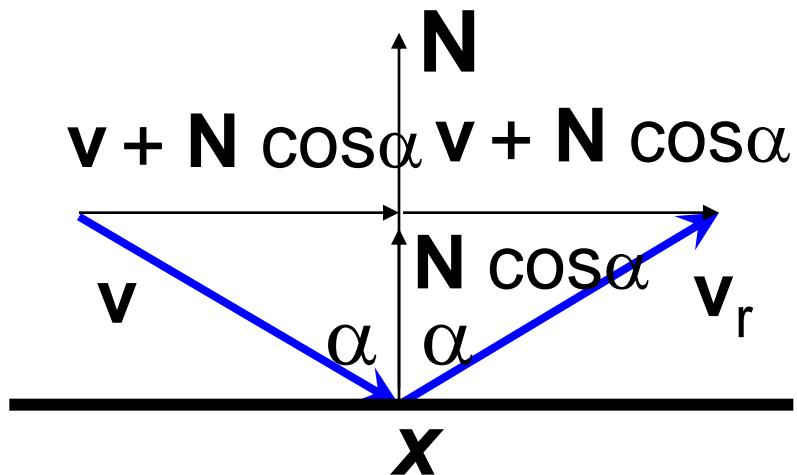
„Sima” = 1 pixelben látható felület síknak tekinthető

- Ideális tükör/törő felület: csak a tükör/törés irányban veri vissza a fényt.
- Elnyelés nagysága az anyag $F(\lambda, \theta')$ Fresnel együtthatójától függ – jelentős függés a megvilágítás irányától és a hullámhossztól!

Fresnel függvény



Tükörirány számítása



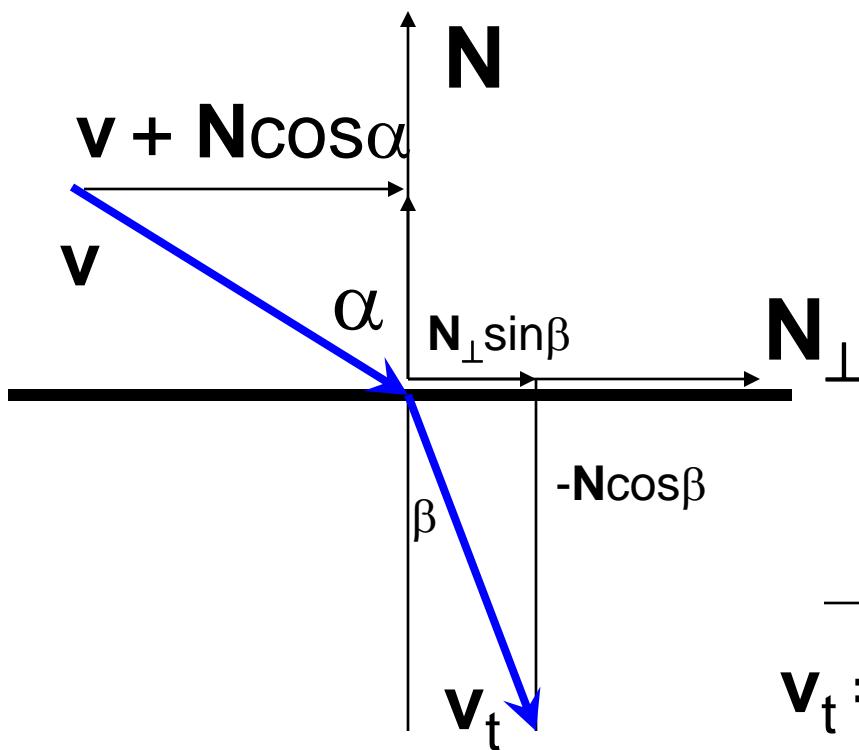
$$\cos \alpha = -(\mathbf{v} \cdot \mathbf{N})$$

$$\mathbf{v}_r = \mathbf{v} + 2 \mathbf{N} \cos \alpha$$

$$\mathbf{L} = \mathbf{v}_r, \mathbf{V} = \mathbf{v}$$

```
ReflectDir( L, N, V ) {
    L = V - N * (N * V) * 2;
}
```

Törési irány



$$n = \frac{\sin \alpha}{\sin \beta}$$

Snellius-
Descartes

$$N_{\perp} = \frac{v + N \cos \alpha}{\sin \alpha}$$

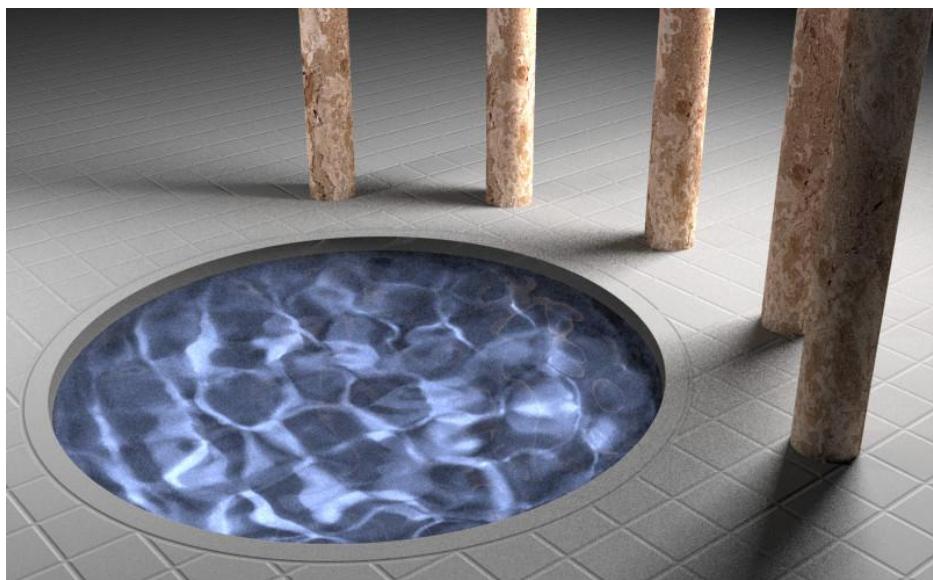
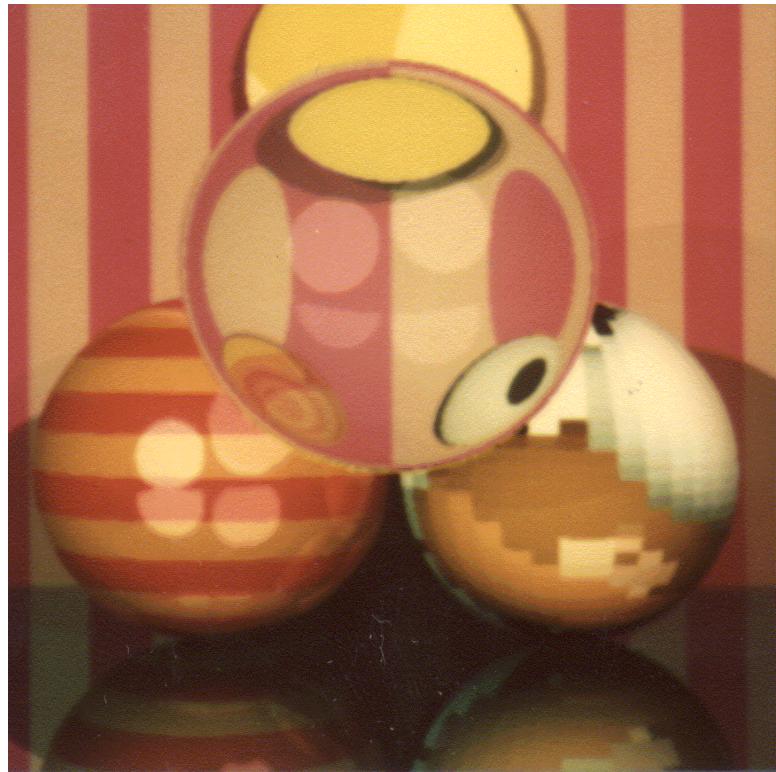
$$v_t = N_{\perp} \sin \beta - N \cos \beta$$

$$v_t = v/n + N(\cos \alpha/n - \cos \beta)$$

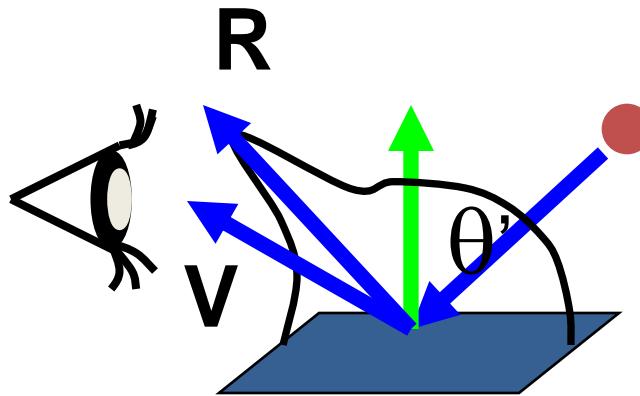
$$\cos \beta = \sqrt{1 - \sin^2 \beta} = \sqrt{1 - \sin^2 \alpha/n^2}$$

$$v_t = v/n + N \left(\cos \alpha/n - \sqrt{1 - (1 - \cos^2 \alpha)/n^2} \right)$$

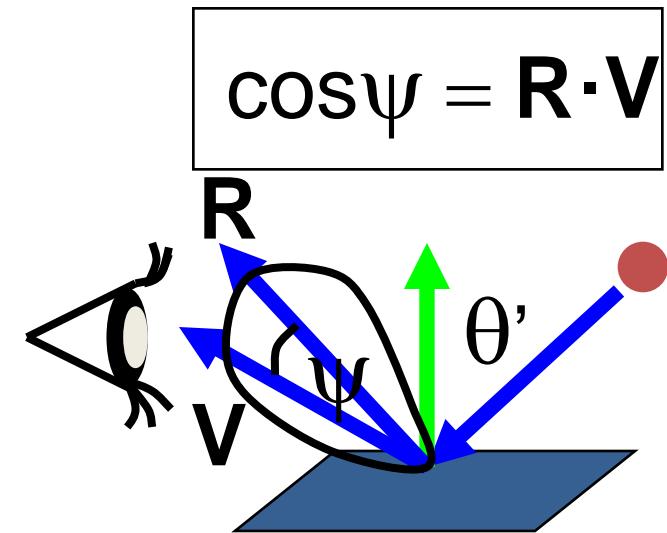
Fénytörő anyagok



Spekuláris visszaverődés: Phong modell



= diffúz +

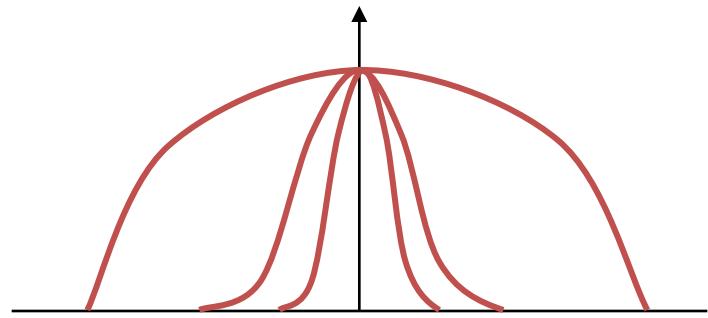


Kell egy függvény, ami nagy $\psi=0$ -ra és gyorsan csökken

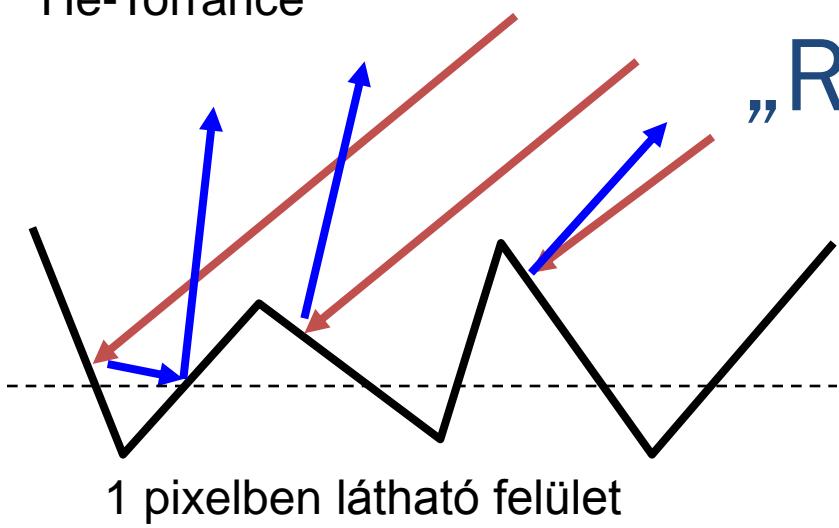
$$L^{ref} = L^{in} k_s (\cos^+ \psi)^n$$

$$f_r(\mathbf{L}, \mathbf{x}, \mathbf{V}) = k_s \cos^n \psi / \cos \theta'$$

Nem szimmetrikus!

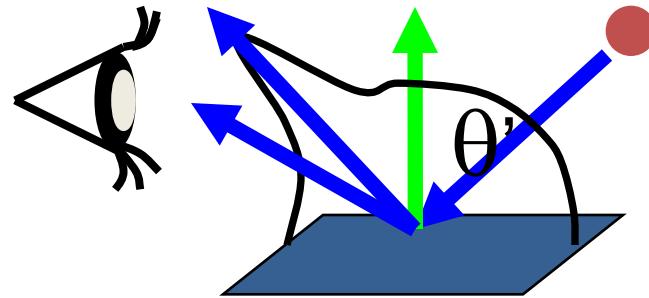


Cook-Torrance
He-Torrance

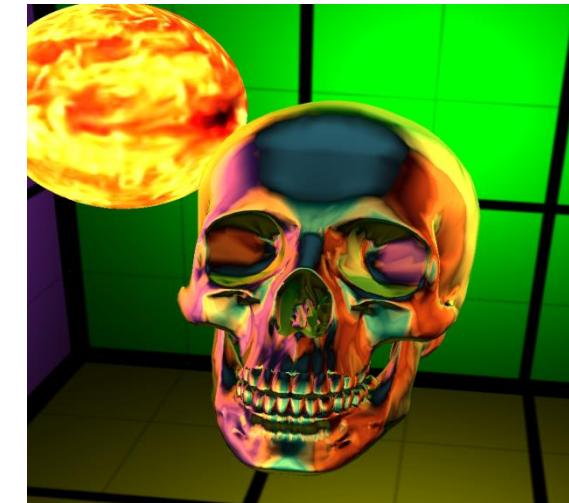
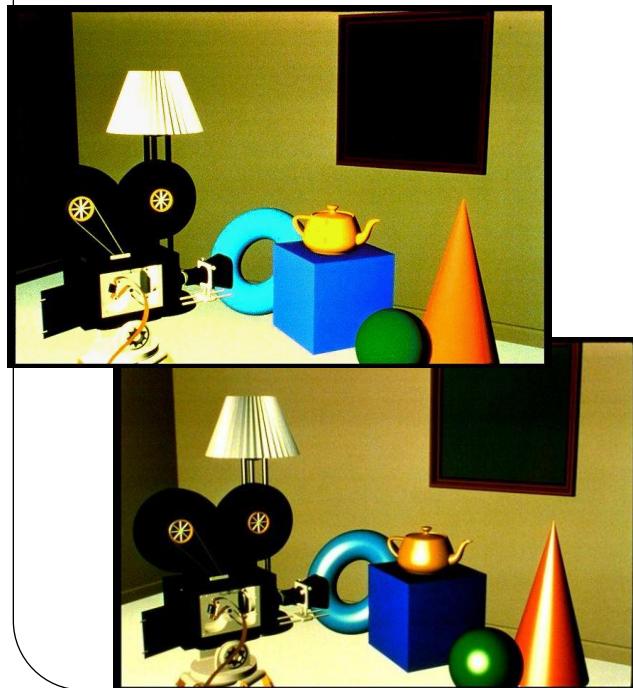


1 pixelben látható felület

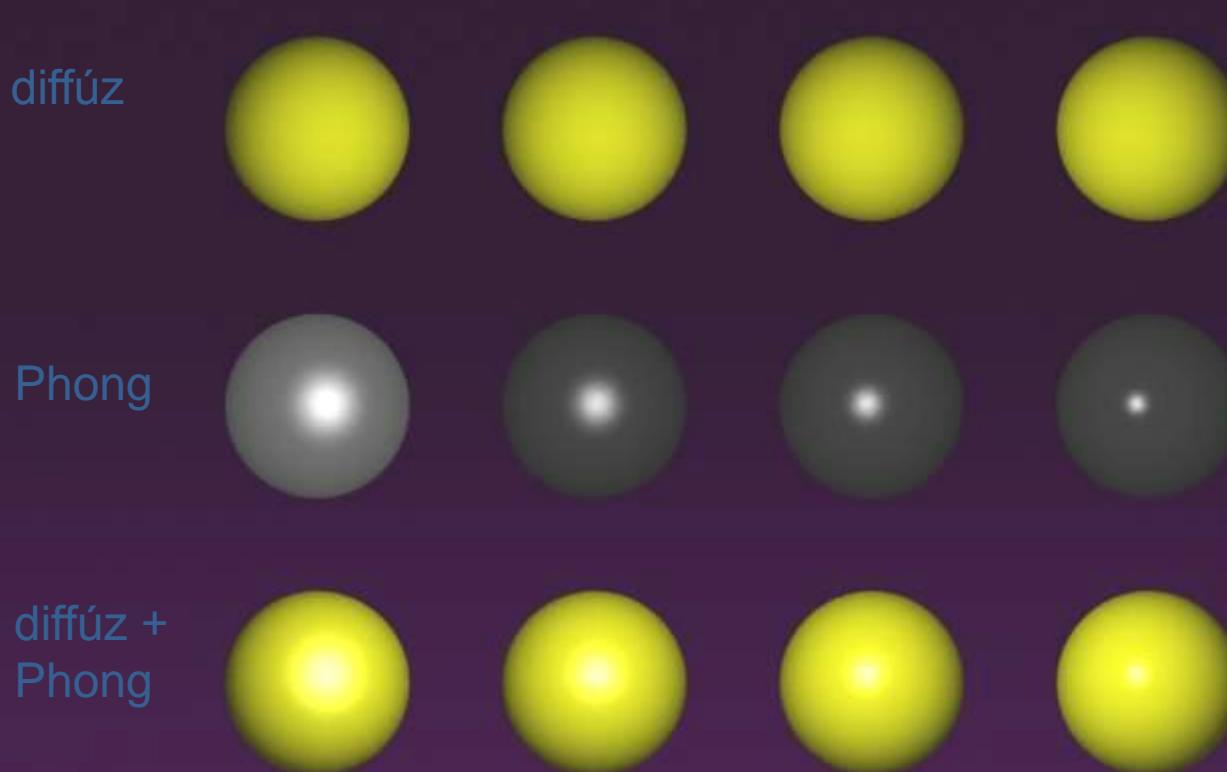
„Rücskös” felületek



Mi: viselkedésileg érvényes modell



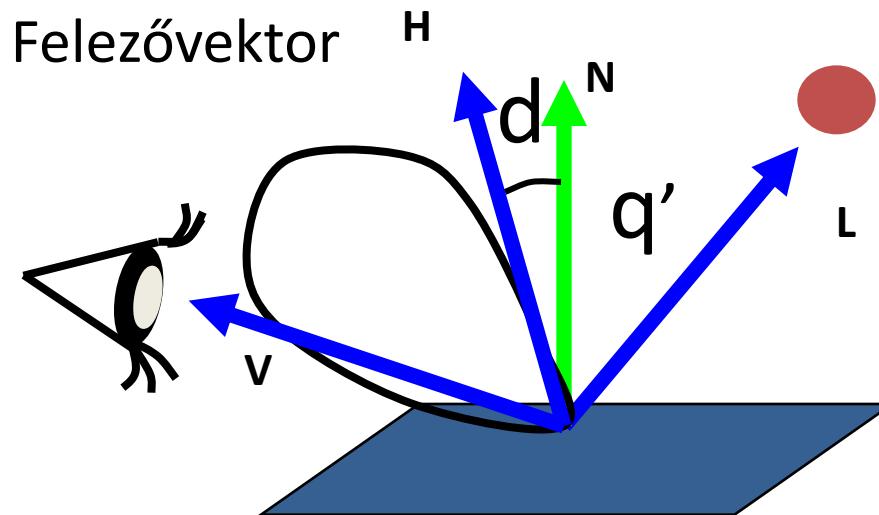
Diffúz+Phong anyagok



Sokszoros fény-anya
kölcsönhatás
„Saját szín”

Egyszeres fény-
anyag
kölcsönhatás,
nemfémeknél
hullámhossz
független

Phong-Blinn modell (OpenGL)



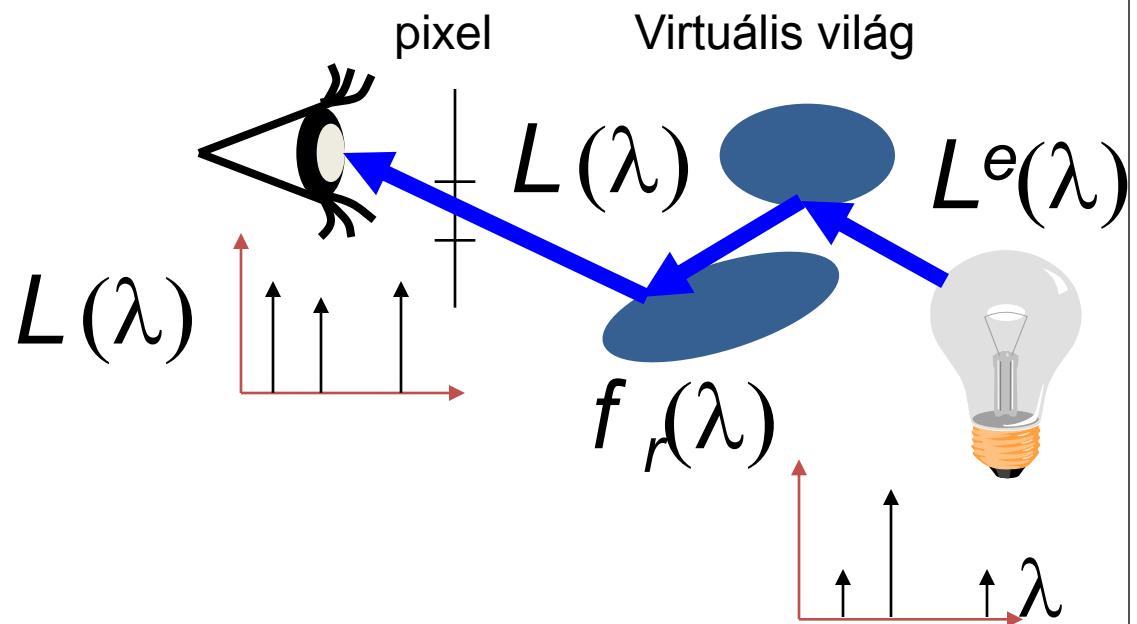
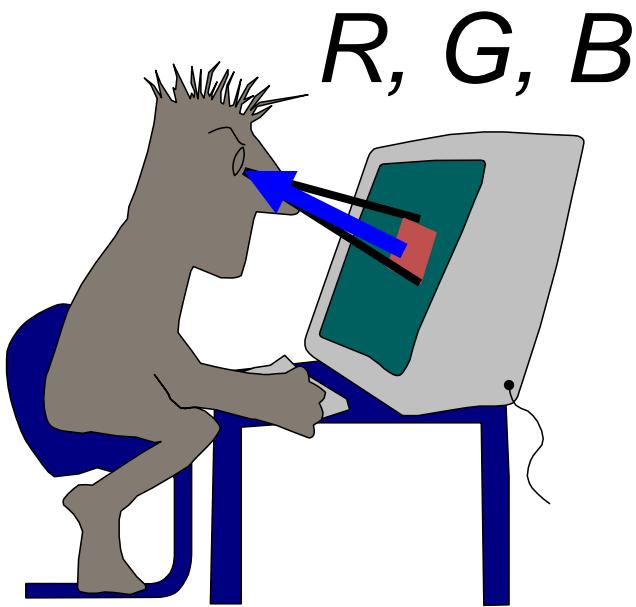
$$H = (L+V) / |L+V|$$

$$\cos d = N \cdot H$$

$$L^{ref} = L^{in} k_s (\cos^+ d)^n$$

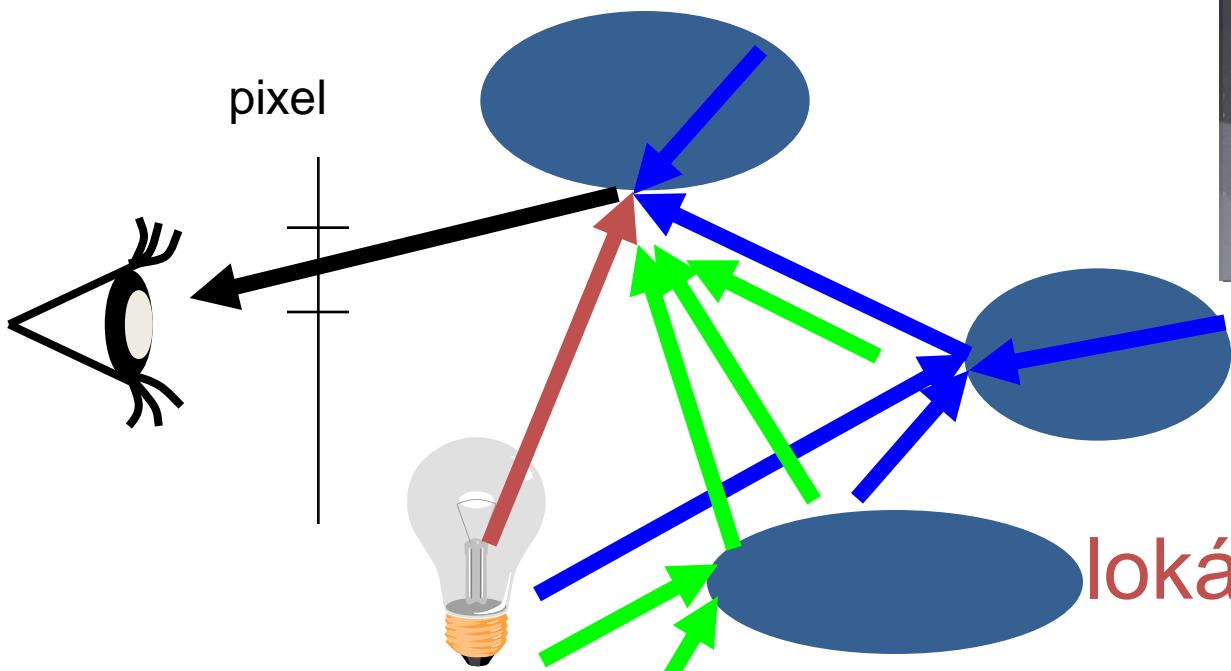
$$f_r(L, x, V) = k_s \cos^n d / \cos \theta'$$

Képszintézis



- Pixelben látható felület meghatározása
- A látható pont szem irányú sugársűrűsége

Megoldási kompromisszumok



lokális illumináció
rekurzív sugárkövetés
globális illumináció



Sugárkövetés: ray-casting, ray-tracing

Sugárkövetés

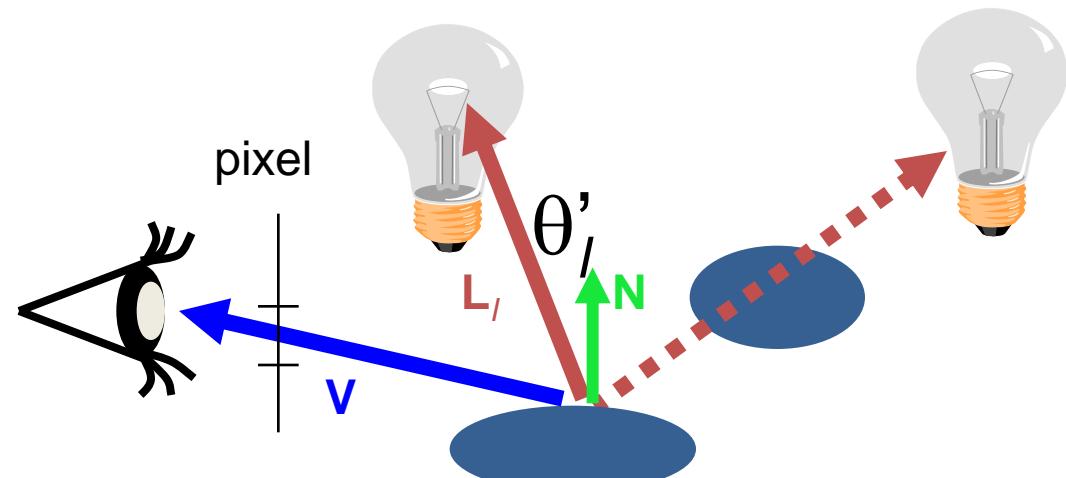
- A képernyő pixeleire egymástól függetlenül oldja meg a takarási és árnyalási feladatot
 - szimuláljuk a fény terjedését, a fénysugarak és felületek ütközését
 - fotonkövetés: szemből indul, a fény útját visszafele rekurzívan követjük
 - Ingyenes sugárkövető program: Pov-Ray
(<http://www.povray.org>)

Sugárkövetés

- Egyszerűsített illuminációs modell
 - csak az ideális (koherens) visszaverődési/törési irányokra számol többszörös visszaverődést illetve törést
 - inkoherens komponensek: elhanyagolja az indirekt megvilágítást, csak az absztrakt fényforrások direkt hatását veszi figyelembe

Lokális illumináció absztrakt fényforrásokkal

Csak absztrakt
fényforrások
direkt megvilágítása

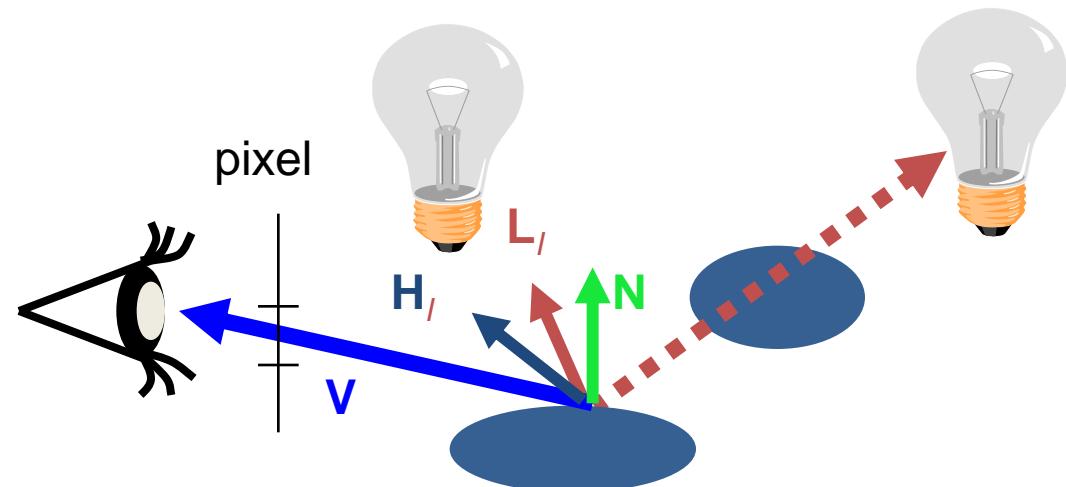


$$L(\mathbf{V}) \approx \sum_I L_I(\mathbf{L}_I) * f_r(\mathbf{L}_I, \mathbf{N}, \mathbf{V}) \cdot \cos \theta',$$

Absztrakt fényforrásokból származó megvilágítás.
(Irányforrás = konstans; Pontforrás = távolság négyzetével csökken
Ha takart, akkor zérus)

Lokális illumináció absztrakt fényforrásokkal

Csak absztrakt
fényforrások
direkt megvilágítása

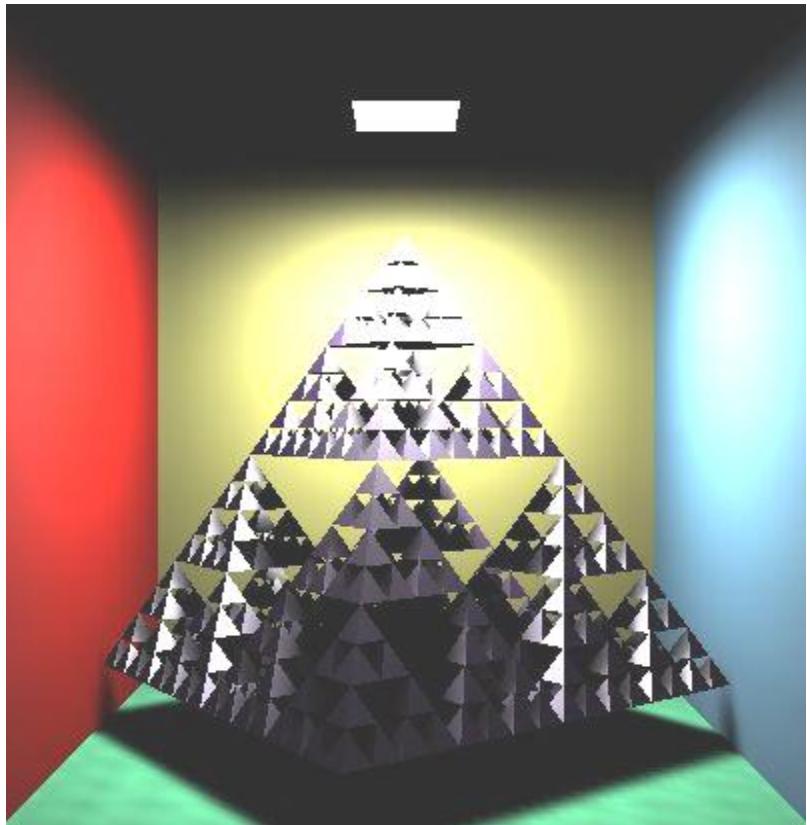


$$L(V) \approx \sum_I L_I(L_I) * \{k_d \cdot (L_I \cdot N)^2 + k_s \cdot ((H_I \cdot N)^2)^{shin}\}$$

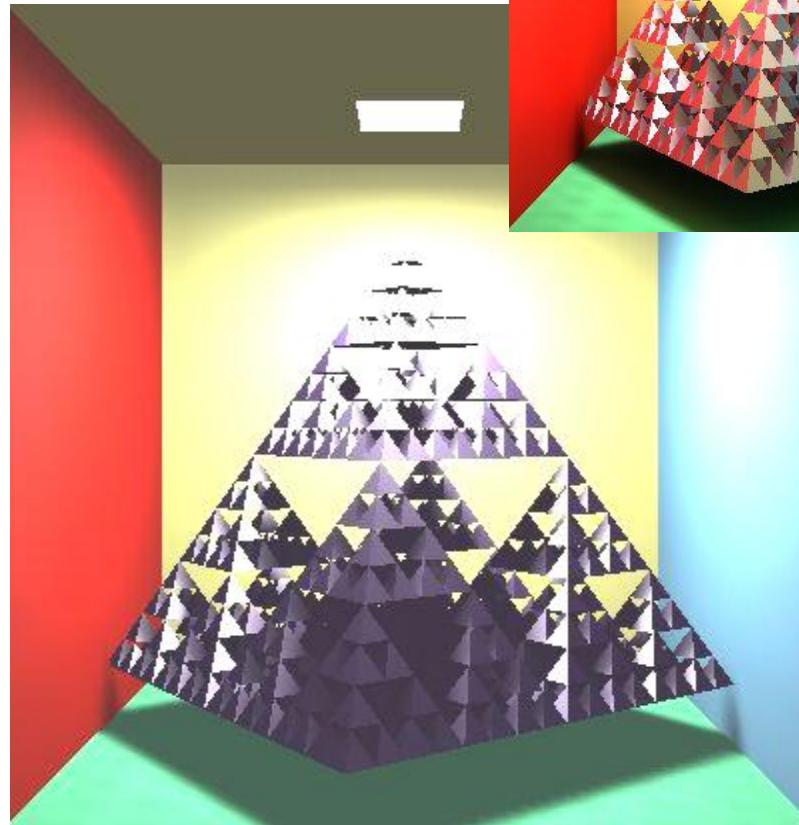
Absztrakt fényforrásokból származó megvilágítás.
(Irányforrás = konstans; Pontforrás = távolság négyzetével csökken
Ha takart, akkor zérus)

Ambiens tag

Lokális illumináció

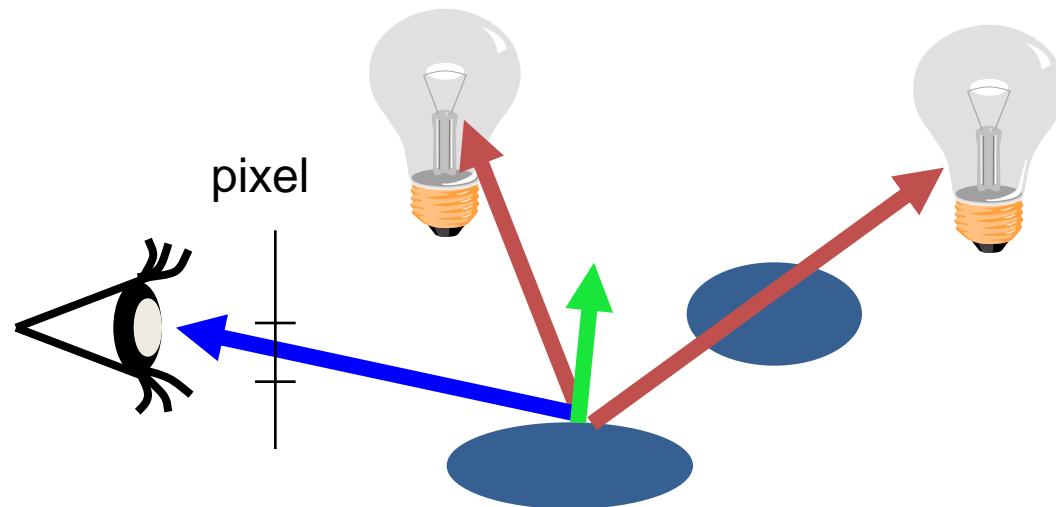


+ ambiens tag



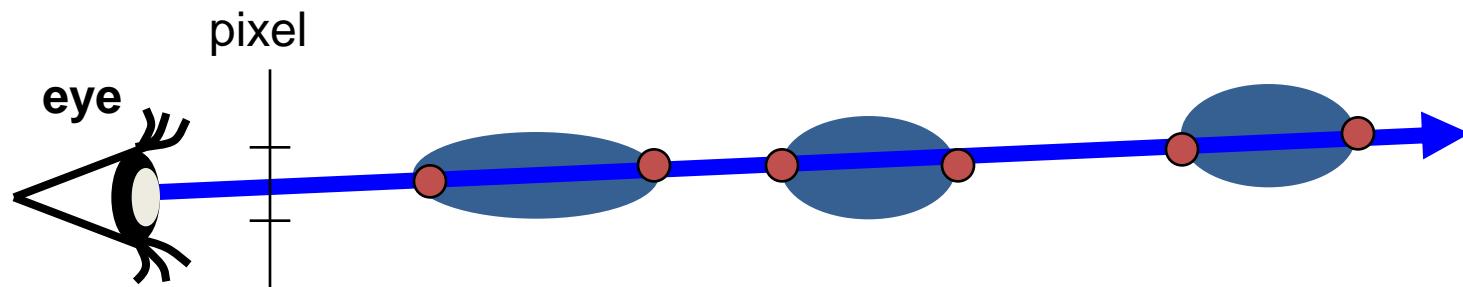
$$L(\mathbf{V}) \approx \sum_i L_i(\mathbf{L}_i)^* f_r(\mathbf{L}_i, \mathbf{N}, \mathbf{V}) \cdot \cos \theta_i + k_a^* L_a$$

Lokális illumináció árnyék nélkül



- Láthatóság számítás a szemből
- A fényforrás fényének visszaverése a nézeti irányba: felületi normális

Láthatóság a szemből

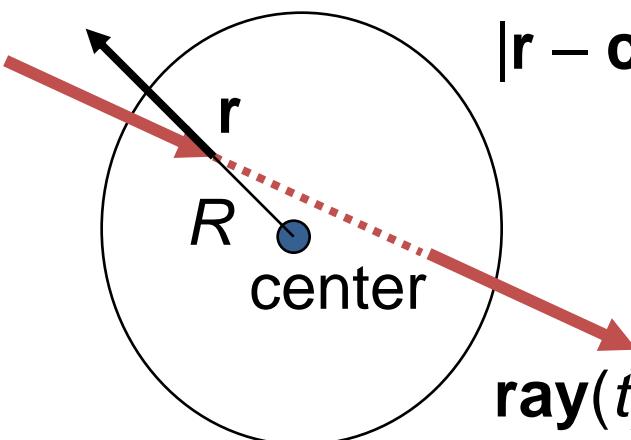


$$\mathbf{ray}(t) = \mathbf{eye} + \mathbf{v} \cdot t, \quad t > 0$$

FirstIntersect(ray \Rightarrow t, iobject, x)

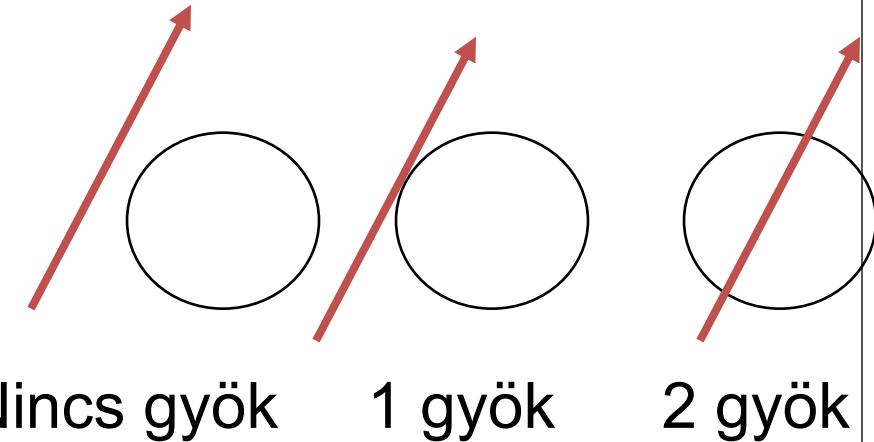
```
t = FLT_MAX;  
FOR each object  
    tnew = Intersect( ray, object );      // < 0 ha nincs metszés  
    IF (tnew > 0 && tnew < t) { t = tnew; iobject = object; }  
ENDFOR  
IF (t < FLT_MAX) { x = eye + v·t; RETURN (t, iobject, x); }  
RETURN „no intersection”  
END
```

Metszéspont számítás gömbbel



$$|\mathbf{r} - \mathbf{center}|^2 = R^2$$

$$\mathbf{ray}(t) = \mathbf{eye} + \mathbf{v} \cdot t$$



$$|\mathbf{ray}(t) - \mathbf{center}|^2 = (\mathbf{ray}(t) - \mathbf{center}) \bullet (\mathbf{ray}(t) - \mathbf{center}) = R^2$$

$$(\mathbf{v} \bullet \mathbf{v})t^2 + 2 ((\mathbf{eye} - \mathbf{center}) \bullet \mathbf{v})t + ((\mathbf{eye} - \mathbf{center}) \bullet (\mathbf{eye} - \mathbf{center})) - R^2 = 0$$

Wanted: a pozitív megoldások közül a kisebb

Felületi normális: $(\mathbf{ray}(t) - \mathbf{center})/R$

Implicit felületek

- A felület pontjai: $f(x,y,z) = 0$ vagy $f(\mathbf{r}) = 0$
- A sugár pontjai: $\mathbf{ray}(t) = \mathbf{eye} + \mathbf{v} \cdot t$
- A metszéspont: $f(\mathbf{ray}(t)) = 0$,
 - 1 ismeretlenes, ált. nemlineáris egyenlet: t^*
 - $(x^*, y^*, z^*) = \mathbf{eye} + \mathbf{v} \cdot t^*$
- Normálvektor = $\text{grad } f \Big|_{x^*, y^*, z^*}$
 - $0 = f(x, y, z) = f(x^* + (x - x^*), y^* + (y - y^*), z^* + (z - z^*)) \approx f(x^*, y^*, z^*) + \frac{\partial f}{\partial x}(x - x^*) + \frac{\partial f}{\partial y}(y - y^*) + \frac{\partial f}{\partial z}(z - z^*)$

Az érintősík
egyenlete:

$$\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \bullet (x - x^*, y - y^*, z - z^*) = 0$$

Kvadratikus felületek

Kvadratikus felület:

$$[x, y, z, 1] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \quad \text{Másodfokú egyenlet}$$



Ellipszoid

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0$$

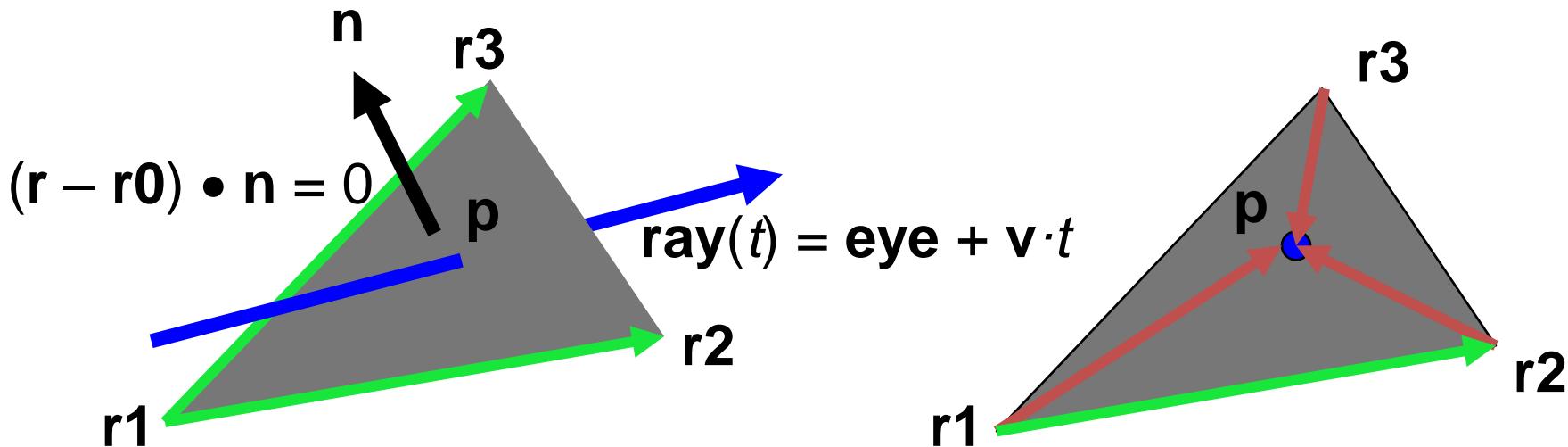
Végtelen kúp

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - z^2 = 0$$

Végtelen henger

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$$

Háromszög



1. Síkmetszés:
normál:

$$(\text{ray}(t) - \mathbf{r}_1) \bullet \mathbf{n} = 0, \quad t > 0$$
$$\mathbf{n} = (\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_1)$$

$$t = \frac{(\mathbf{r}_1 - \text{eye}) \bullet \mathbf{n}}{\mathbf{v} \bullet \mathbf{n}}$$

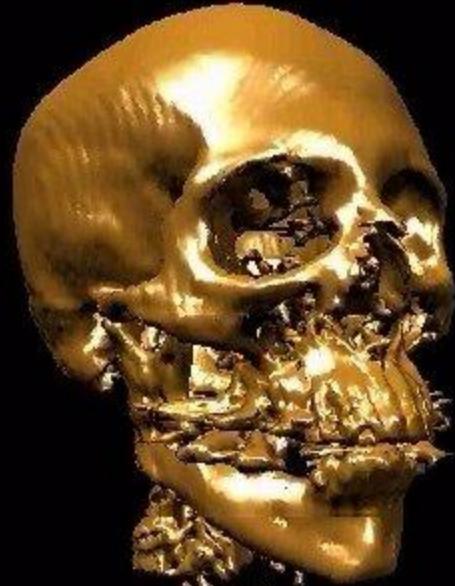
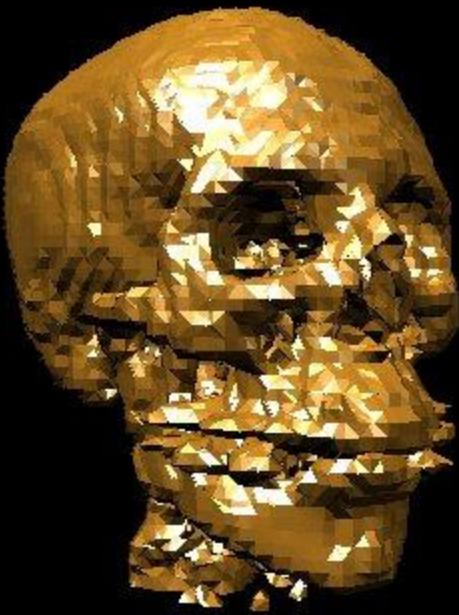
2. A metszéspont a háromszögön belül van-e?

$$((\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{p} - \mathbf{r}_1)) \bullet \mathbf{n} > 0$$

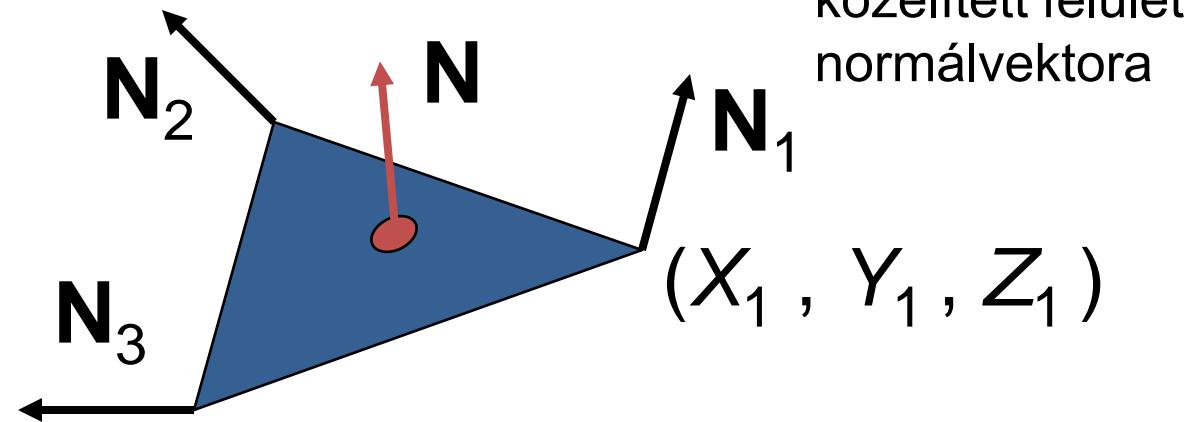
$$((\mathbf{r}_3 - \mathbf{r}_2) \times (\mathbf{p} - \mathbf{r}_2)) \bullet \mathbf{n} > 0$$

$$((\mathbf{r}_1 - \mathbf{r}_3) \times (\mathbf{p} - \mathbf{r}_3)) \bullet \mathbf{n} > 0$$

Felületi normális: \mathbf{n}
vagy árnyaló normálok
(shading normals)



Árnyaló normálok



$$\mathbf{N} = A X + B Y + C$$

← A, B, C

$$\begin{aligned}\mathbf{N}_1 &= A X_1 + B Y_1 + C \\ \mathbf{N}_2 &= A X_2 + B Y_2 + C \\ \mathbf{N}_3 &= A X_3 + B Y_3 + C\end{aligned}$$

3 változós
lineáris
egyenletsz.

Parametrikus felületek

$\mathbf{r}(u, v), \quad u, v \text{ in } [0, 1]$

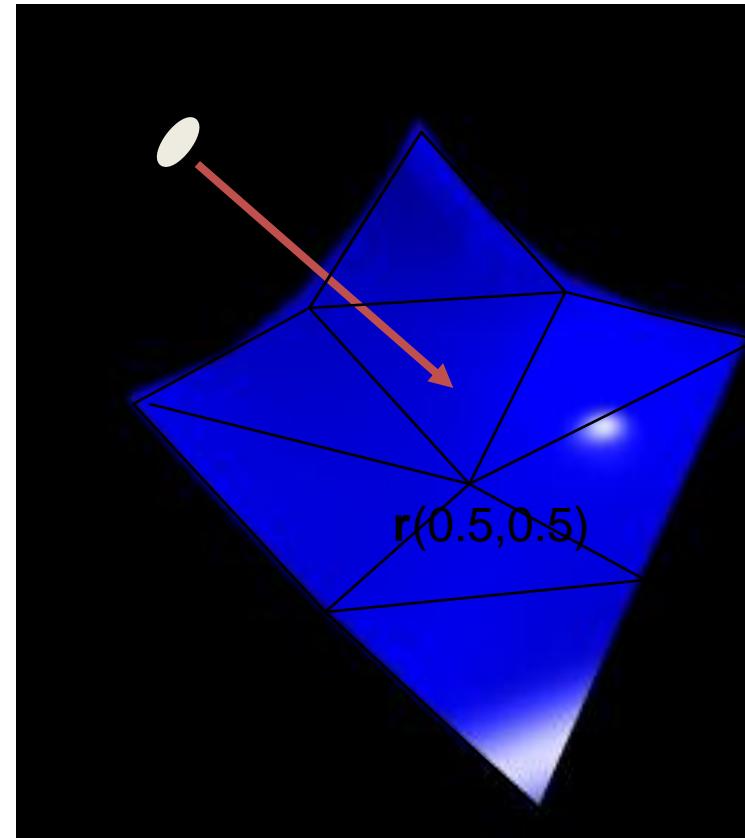
$\mathbf{ray}(t) = \mathbf{eye} + \mathbf{v} \cdot t, \quad t > 0$

$$\mathbf{r}(u, v) = \mathbf{ray}(t)$$

Háromismeretlenes ált.
nem lineáris egyenletrendszer
megoldás: u^*, v^*, t^*

Teszt:

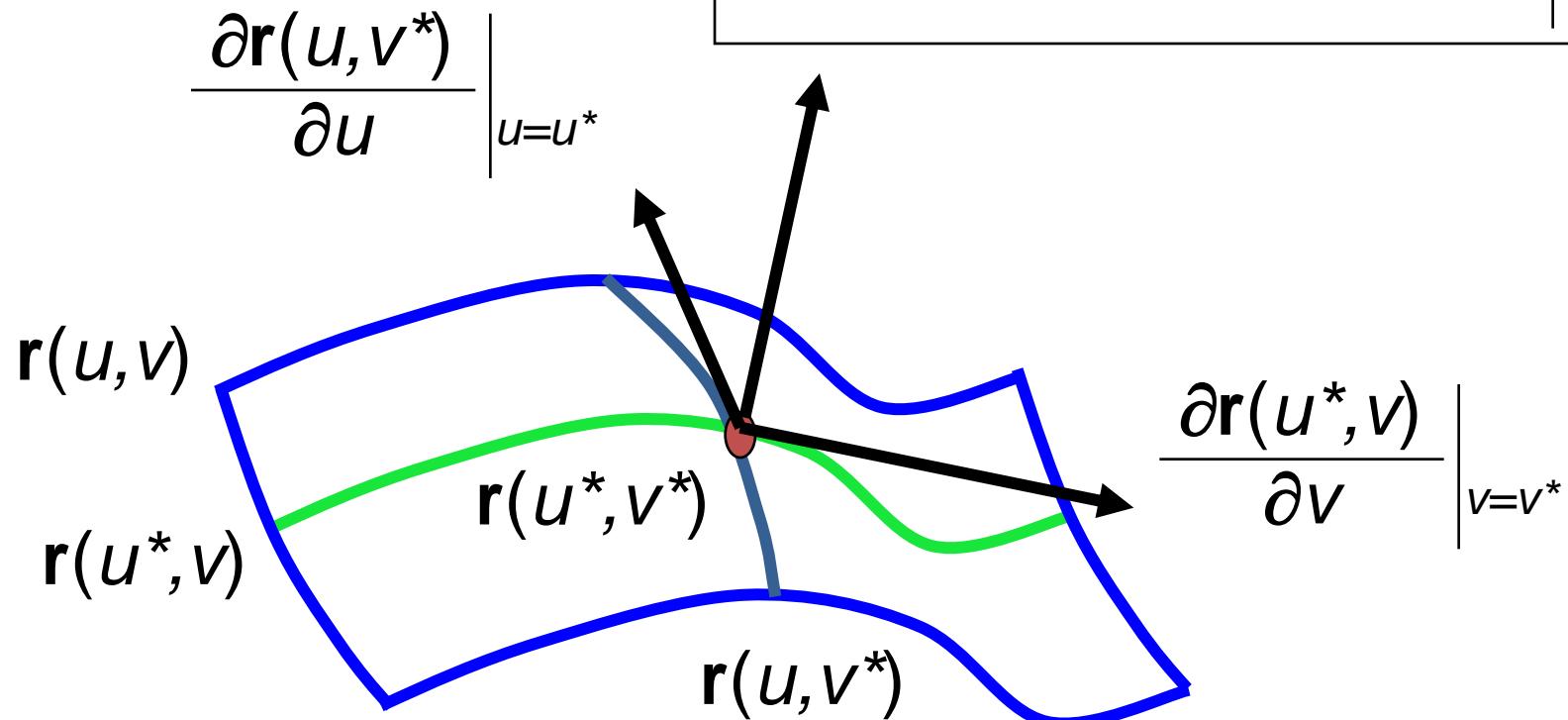
$$0 < u^*, v^* < 1, \\ t^* > 0$$



Rekurzív tesszelláció:
nem robustus

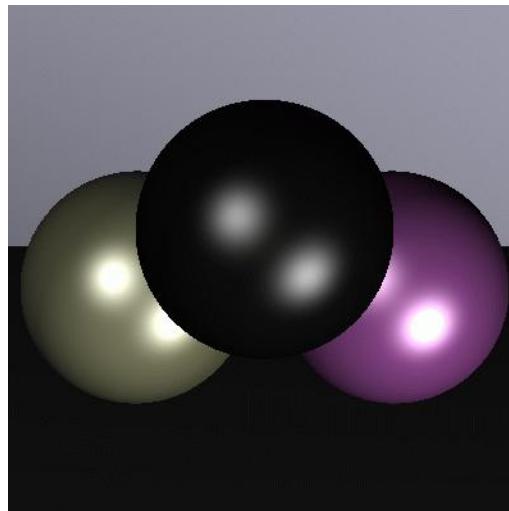
Parametrikus felületek normálvektora

$$\mathbf{n}(u, v) = \frac{\partial \mathbf{r}(u, v)}{\partial u} \times \frac{\partial \mathbf{r}(u, v)}{\partial v} \Big|_{\begin{array}{l} u=u^* \\ v=v^* \end{array}}$$

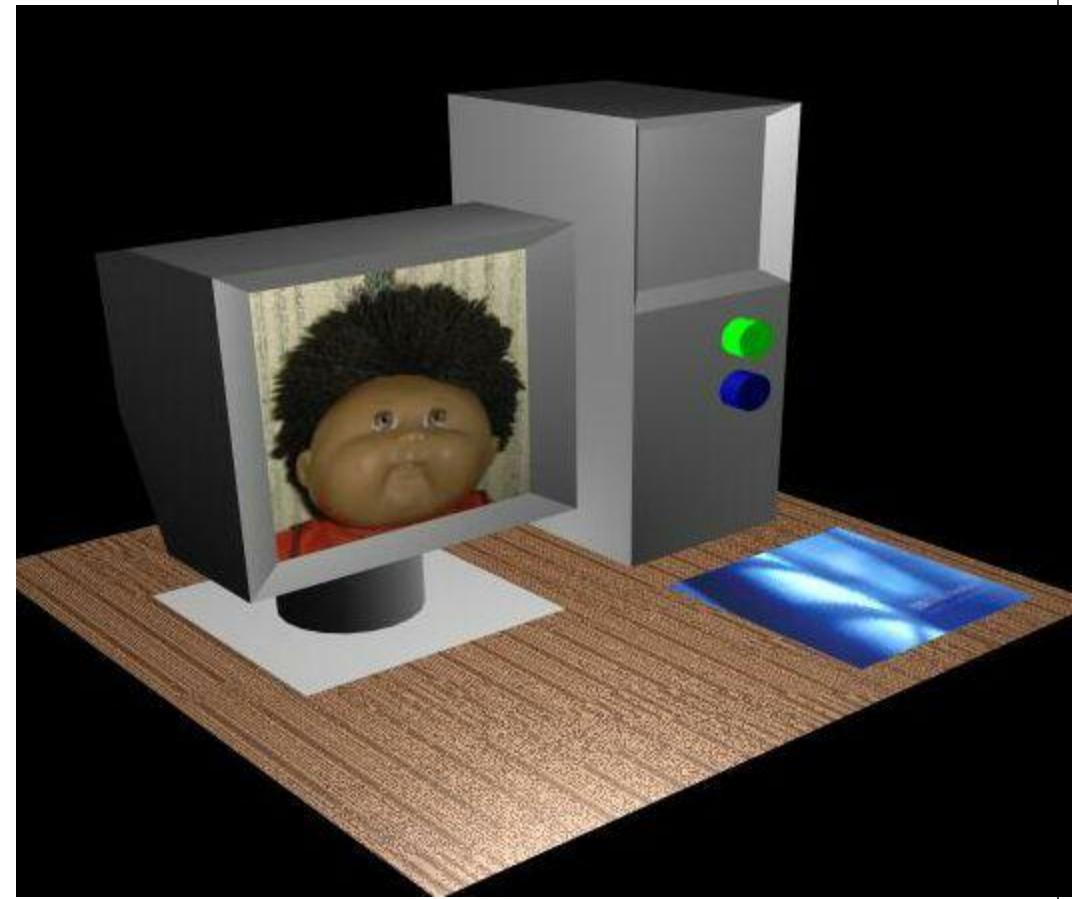
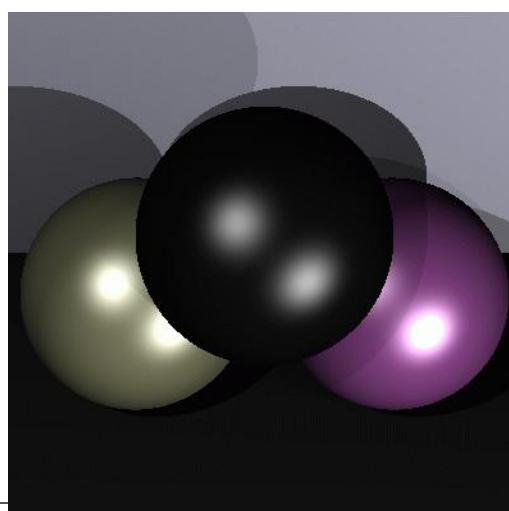


Ray-casting képek

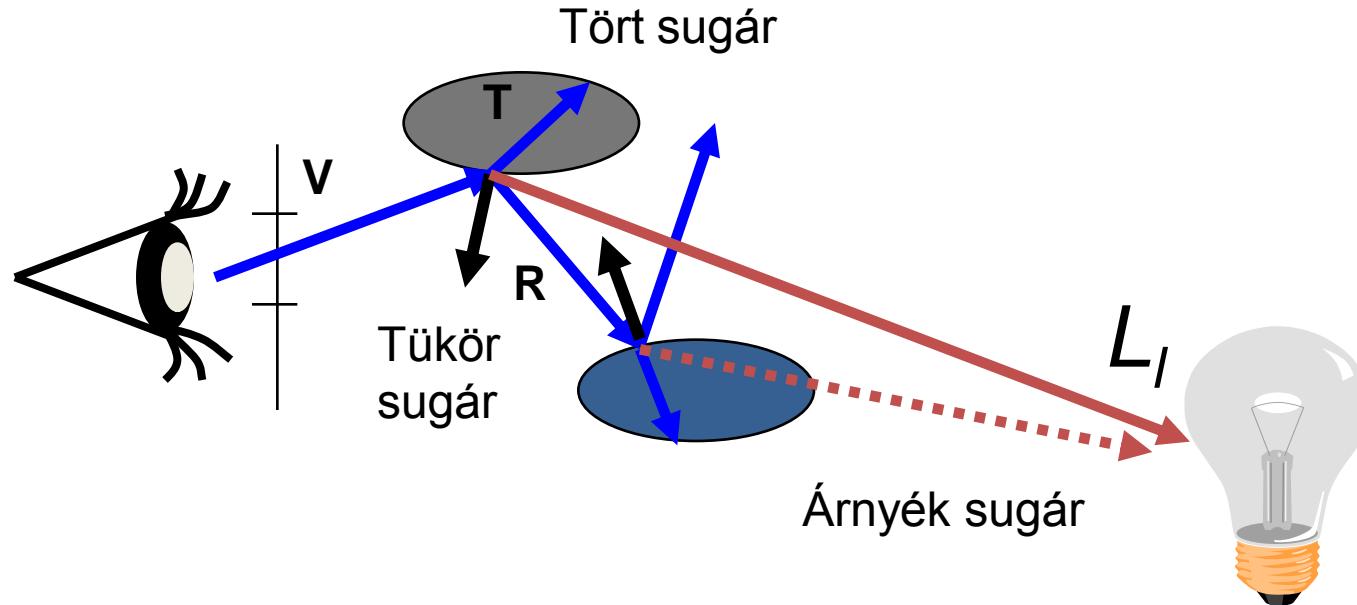
Lokális
illumináció



Lokális
illumináció
+
árnyékok



Rekurzív sugárkövetés



$$L(V) \approx \sum_i L_i(L_i) * (k_d \cdot (L_i \bullet N)^+ + k_s \cdot ((H_i \bullet N)^{shine}) + k_a \cdot L_a$$

$$+ k_r \cdot L^{\text{in}}(R)$$

Fresnel

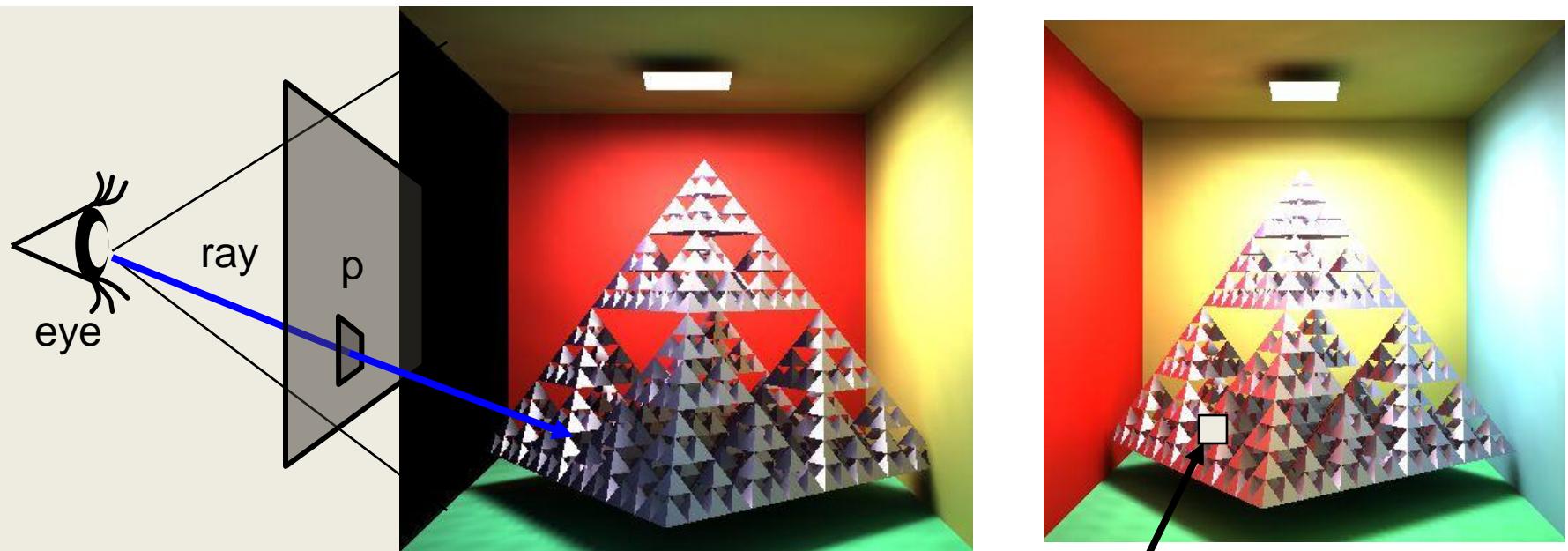
$$+ k_t \cdot L^{\text{in}}(T)$$

Tükör irányból
érkező fény

1-Fresnel

Törési irányból
érkező fény

Sugárkövetés (Ray-tracing)



Render()

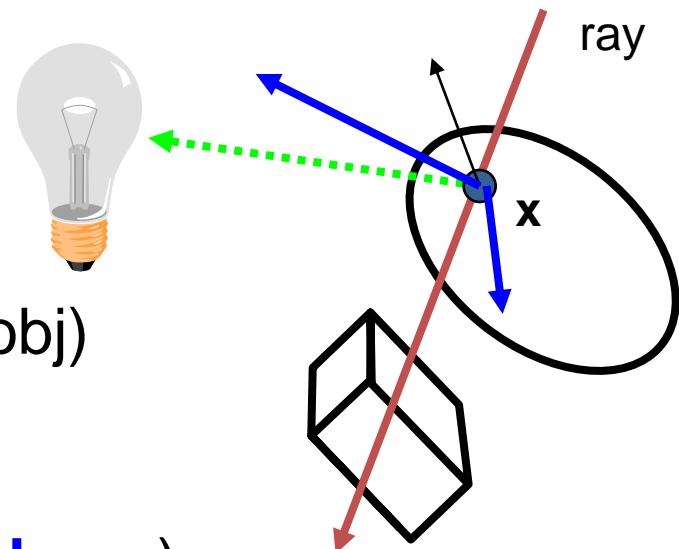
```
for each pixel p  
    Ray r = ray( eye  $\Rightarrow$  pixel p )  
    color = Trace(ray)  
    WritePixel(p, color)  
endfor  
end
```

p color

Sugárkövetés: Trace függvény

Color Trace(ray)

```
IF (FirstIntersect(ray ⇒ obj, x) < 0)
    RETURN La
ENDIF
color = DirectLightSource(x, ray.v, obj)
IF ( obj.mirror )
    ReflectDir( ray, reflected ray)
    color += obj.kr * Trace( reflected ray )
ENDIF
IF ( obj.refractive && RefractDir( ray, refracted ray ) )
    color += obj.kt * Trace( refracted ray )
ENDIF
RETURN color
```



Sugárkövetés: Trace függvény

Color Trace(ray, d)

IF (d > dmax) RETURN L_a

IF (FirstIntersect(ray \Rightarrow obj, x) < 0)
RETURN L_a

ENDIF

color = DirectLightSource(x, ray.v, obj)

IF (obj.mirror)
ReflectDir(ray, reflected ray)

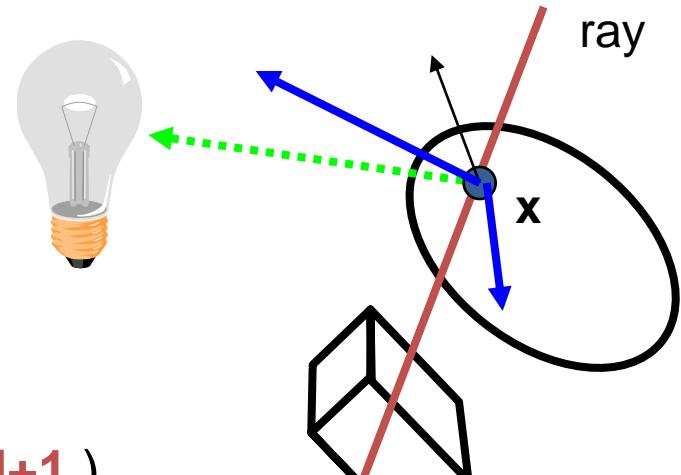
color += obj.k_r * Trace(reflected ray, d+1)

ENDIF

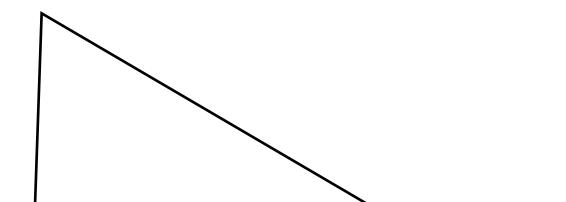
IF (obj.refractive && RefractDir(ray, refracted ray))
color += obj.k_t * Trace(refracted ray, d+1)

ENDIF

RETURN color



$$ray.v - 2 \cdot \mathbf{N} \cdot (\mathbf{N} \cdot ray.v)$$



$$ray.v / n + \mathbf{N} \cdot ((\mathbf{N} \cdot ray.v) / n - \sqrt{1 - (1 - (\mathbf{N} \cdot ray.v)^2) / n^2})$$

DirectLightSource

DirectLightSource(x, v, obj)

color = obj.k_a * L_a

FOR each lightsource l DO

 shadowray.eye = x; shadowray.v = light[l].pos – x;

 (t,y) = FirstIntersect(shadowray);

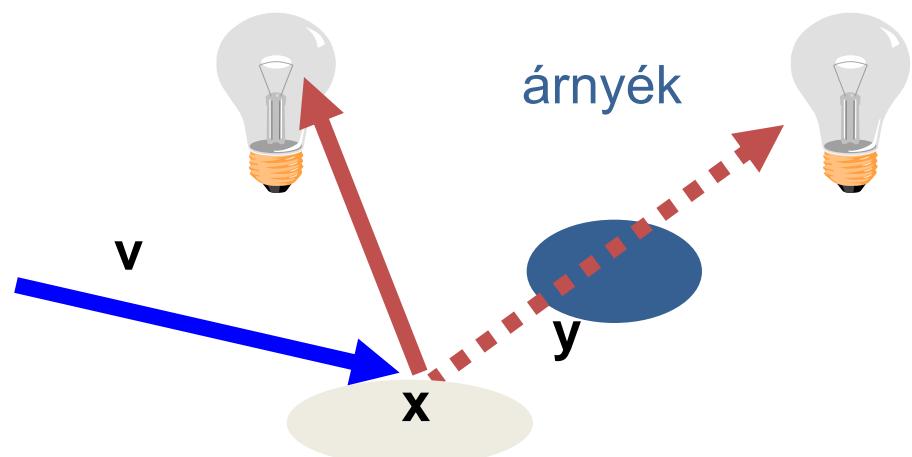
 IF (t < 0 || |x-y| > |x-light[l].pos|)

 color += light[l].Intensity *
 (obj.k_d · (L_l · N)⁺ + obj.k_s · ((H_l · N)⁺)^{obj.shine})

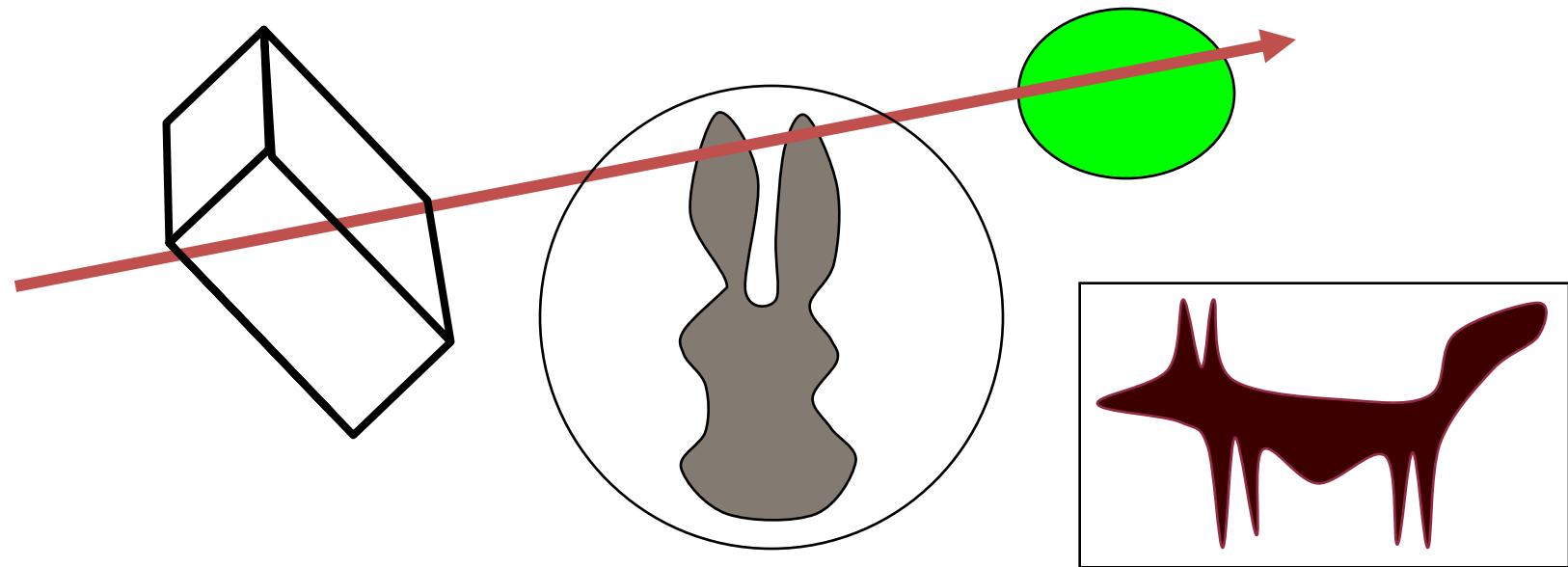
 ENDIF

ENDFOR

RETURN color

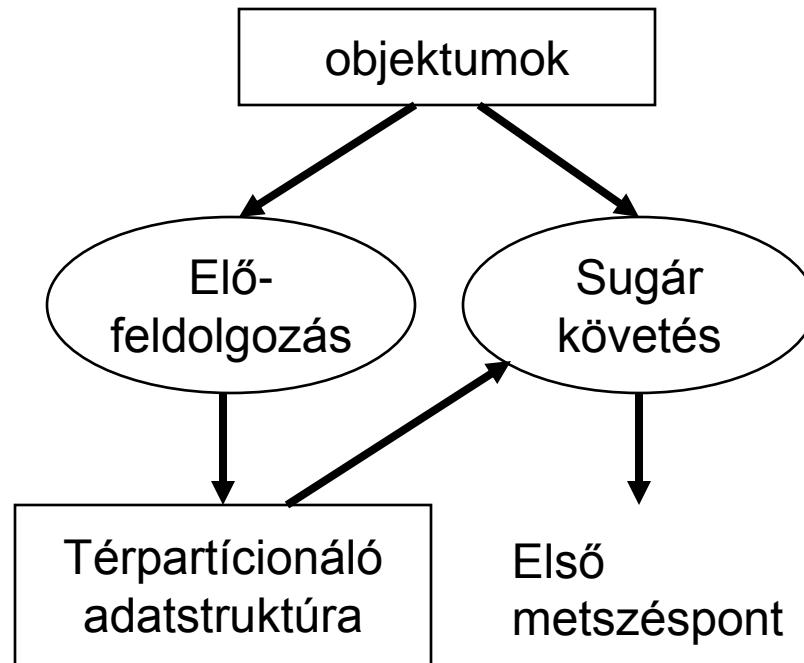
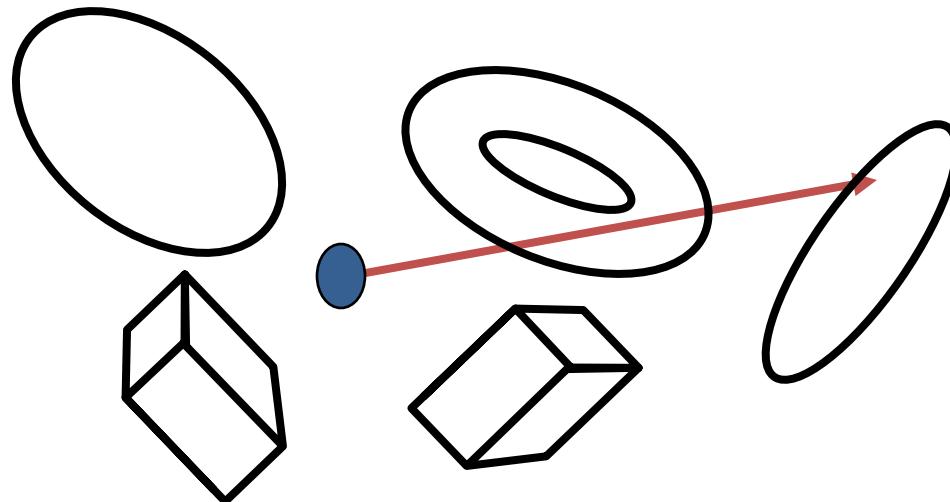


Befoglaló térfogat (Bounding Volume)



```
double IntersectBV( ray, object ) // < 0 ha nincs
  IF ( Intersect( ray, bounding volume of object) < 0)
RETURN -1;
  RETURN Intersect( ray, object );
END
```

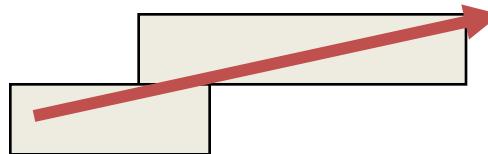
Térpartícionáló módszerek



Adatstruktúra:

- Ha ismert a sugár, akkor a potenciális metszett objektumok számát csökkenti
- Ha a potenciálisak közül találunk egyet, akkor a többi nem lehet közelebb

Reguláris térháló



Sugárkövetés:

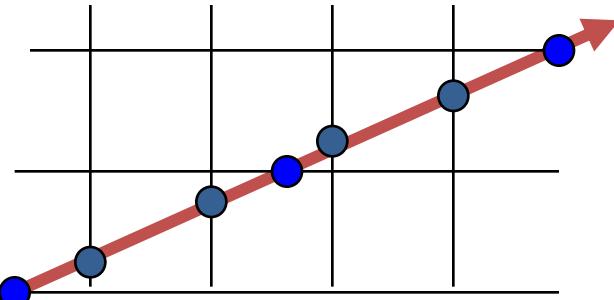
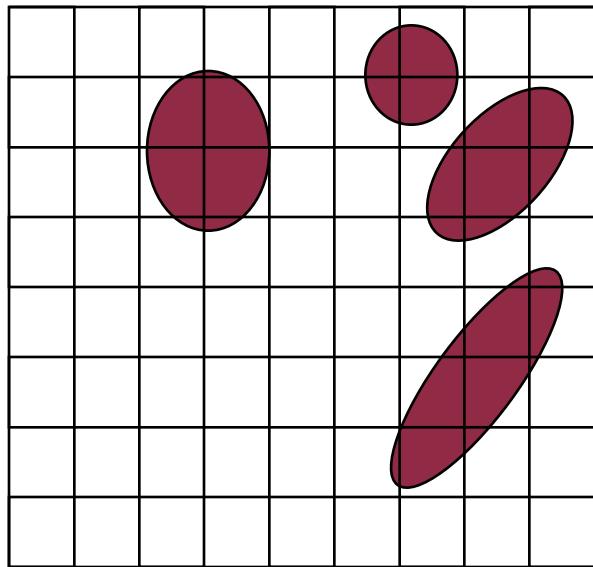
FOR each cell of the line // line drawing

Metszés a cellában lévőkkel

IF van metszés RETURN

ENDFOR

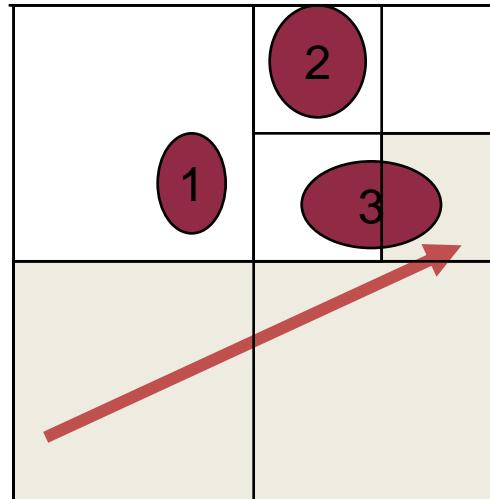
átlagos eset komplexitás: $O(1)$



Előfeldolgozás:

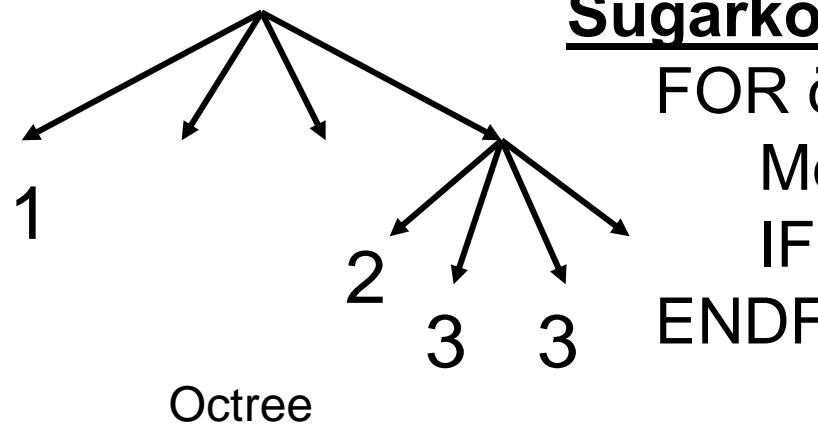
Minden cellára a metszett objektumok
komplexitás: $O(n \cdot c) = O(n^2)$

Nyolcas (oktális) fa



Faépítés(cella):

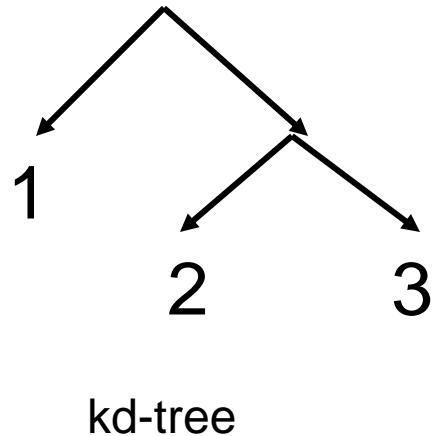
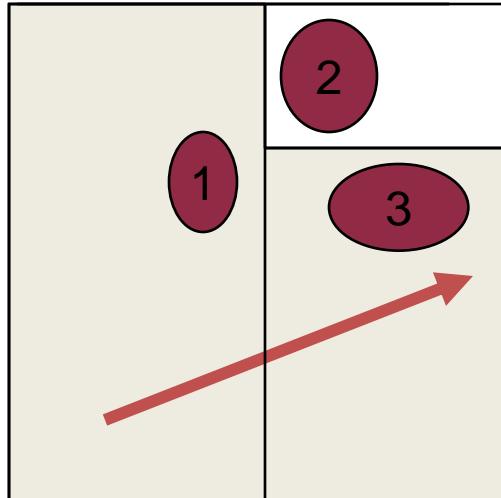
```
IF a cellában kevés objektum van  
cellában regisztráld az objektumokat  
ELSE  
    cellafelezés: c1, c2, ..., c8  
    Faépítés(c1); ... Faépítés(c8);  
ENDIF
```



Sugárkövetés:

```
FOR összes sugár által metszett cellára  
    Metszés a cellában lévőkkel  
    IF van metszés RETURN  
ENDFOR
```

Binary Space Partitioning fa (kd-fa)



Faépítés(cella):

```
IF a cellában kevés objektum van  
cellában regisztráld az objektumokat  
ELSE  
    sík keresés  
    cella felezés a síkkal: c1, c2  
    Faépítés(c1); Faépítés(c2);  
ENDIF
```

Sugárkövetés:

```
FOR each cell intersecting the line  
    Metszés a cellában lévőkkel  
    IF van metszés RETURN  
ENDFOR
```

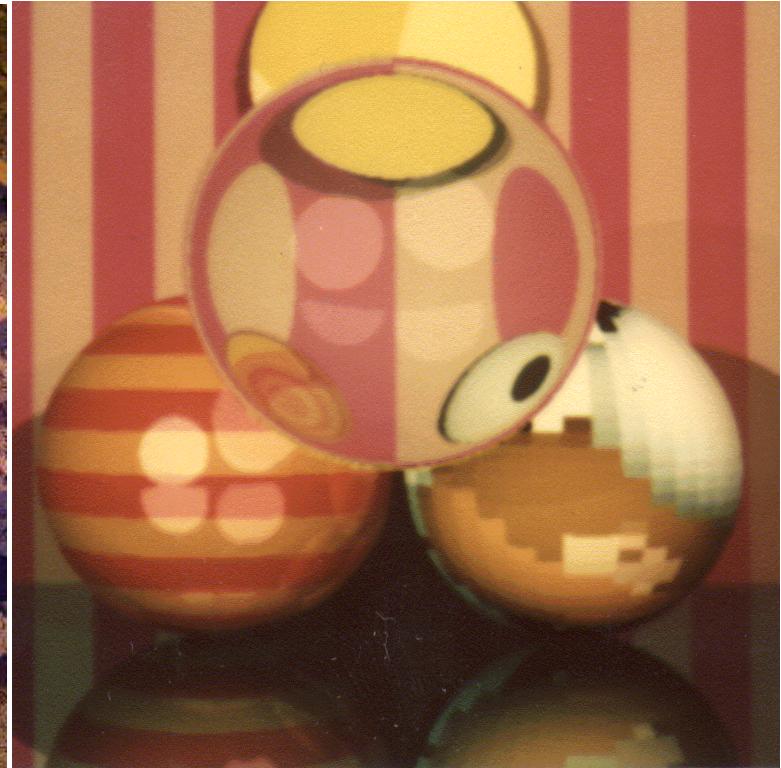
Sugárkövető program



```
typedef struct{double x,y,z}vec;vec U,black,amb={.02,.02,.02}; struct sphere{ vec cen,color;double rad,kd,ks,kt,kl,ir}*best,sph[]={0.,6.,.5,1.,1.,.9, .05,.2,.85,0.,1.7,-1.,8.,-.5,1.,.5,.2,1.,.7,.3,0.,.05,1.2,1.,8.,-.5,.1,8.,8, 1.,.3,.7,0.,.0.,1.2,3,.8,1.,.7,0.,0.,.6,1.5,-3.,-3.,12.,.8,1., 1.,5,0.,0.,0.,.5,1.5,};yx; double u,b,tmin,sqrt(),tan();double vdot(A,B)vec A ,B; {return A.x*B.x+A.y*B.y+A.z*B.z;}vec vcomb(a,A,B)double a;vec A,B; {B.x+=a*A.x;B.y+=a*A.y;B.z+=a*A.z;return B;} vec vunit(A)vec A;{return vcomb(1./sqrt( vdot(A,A)),A,black);}struct sphere *intersect(P,D)vec P,D;{best=0;tmin=1e30;s= sph+5;while(s-->sph)b=vdot(D,U=vcomb(-1.,P,s->cen)),u=b*b-vdot(U,U)+s->rad*s ->rad,u=u>0?sqrt(u):1e31,u=u<1e-7?b-u:b+u,tmin=u>=1e-7&&u<tmin?best=s,u: tmin;return best;}vec trace(level,P,D)vec P,D;{double d,eta,e;vec N; struct sphere*s,*l;if(!level--)return black;if(s=intersect(P,D));else return amb;color=amb;eta=s->ir;d= -vdot(D,N=vunit((-1.,P=vcomb(tmin,D,P),s->cen )));if(d<0)N=vcomb(-1.,N,black),eta=1/eta,d= -d;l=sph+5;while(l-->sph)if((e=l ->kl*vdot(U=vunit(vcomb(-1.,P,l->cen ))))>0&&intersect(P,U)==l)color=vcomb(e ,l->color,color);U=s->color;color.x*=U.x;color.y*=U.y;color.z*=U.z;e=1-eta* eta*(1-d*d);return vcomb(s->kt,e>0?trace(level,P,vcomb(eta,D,vcomb(eta*d-sqrt (e),N,black,black,vcomb(s->ks,trace(level,P,vcomb(2*d,N,D)),vcomb(s->kd, color,vcomb(s->kl,U,black))))));}

main(){printf("%d %d\n",32,32);while(yx<32*32) U.x=yx%32-32/2,U.z=32/2-yx++/32,U.y=32/2/tan(25/114.591559026);U=vcomb(255., trace(3,black,vunit(U)),black),printf("%.0f %.0f %.0f\n",U); }/*minray!*/
```

Sugárkövetés: eredmény



Számítási idő \propto Pixelszám x Objektumszám x (Fényforrás szám+1)

Árnyalás az OpenGL-ben

- Egyszerűsített árnyalási egyenlet
 - ahol a fizikai leírás túl bonyolult, tapasztalati úton bevált képleteket használ
 - használ spekuláris és diffúz visszaverődést, nincs azonban törés és tükrözés
 - bevezeti a diffúz és spekuláris fény fogalmát – ennek a különválasztásnak nincs fizikai jelentése, de a megjelenítéshez nagyobb szabadságot ad
- Csak a csúcspontokban számít színt
 - többi pontban interpolál a környező csúcspontok színéből

Megvilágítás modell

```
void glLightModel{if}{,v} (GLenum  
pname, TYPE param);
```

pname lehet:

- **GL_LIGHT_MODEL_AMBIENT**
 - ambiens RGBA fény az egész színtérben
- **GL_LIGHT_LOCAL_VIEWER**
 - lokális vagy végtelenben lévő (default) „nézeti pont”
- **GL_LIGHT_MODEL_TWO_SIDE**
 - poligonjainkhoz egy- vagy minden oldalon kell megvilágítást számolnunk

Fényforrás megadása

- Fény ki-be kapcsolás:
 - glEnable(GL_LIGHTING);
 - glDisable(GL_LIGHTING)
- `void glLight{if}{,v} (GLenum light,
GLenum pname, TYPE param);`
 - *light*: GL_LIGHT0, GL_LIGHT1, ..., or GL_LIGHT7
 - *param*: különböző paraméterek (lásd köv. slide-ok)
 - Az egyes fényforrásokat is külön engedélyezni kell (azaz bekapcsolni):
 - Pl. glEnable(GL_LIGHT0);

Fényösszetevők az OpenGL-ben

- Absztrakt fényösszetevők:
 - objektumok által kibocsátott fény
 - környezeti fény – ambient light - GL_AMBIENT
 - kompenzálja a többszörös visszaverődések elhanyagolását
 - szórt fény - diffuse light - GL_DIFFUSE:
 - irányjal rendelkezik, mindig valamely fényforrásból jön. Az objektumokkal ütközve minden irányba azonos módon és mértékben verődik vissza
 - csilllogás - specular light - GL_SPECULAR
 - van iránya és forrása, hatása nem csak az anyagtulajdonságoktól és a normálisuktól függ, hanem a nézőponttól is

Minta

- „Diffúz fényforrás” megadása:

```
GLfloat light_diffuse[] = { 1.0, 1.0,  
    1.0, 1.0 } ;  
  
glLightfv(GL_LIGHT0, GL_DIFFUSE,  
    light_diffuse) ;
```

OpenGL fényforrások

- 3 típus:
 - irányforrás
 - konstans fényerősség, adott irány
 - pontszerű
 - forrástól távolodva csökkenő fényerősség
$$a_l = \frac{1}{k_{0,l} + k_{1,l} d_l + k_{2,l} d_l^2}$$
 - $k_{0,l}$: GL_CONSTANT_ATTENUATION
 - $k_{1,l}$: GL_LINEAR_ATTENUATION
 - $k_{2,l}$: GL_QUADRATIC_ATTENUATION
 - szpotlámpa
 - kúpos hatóterület

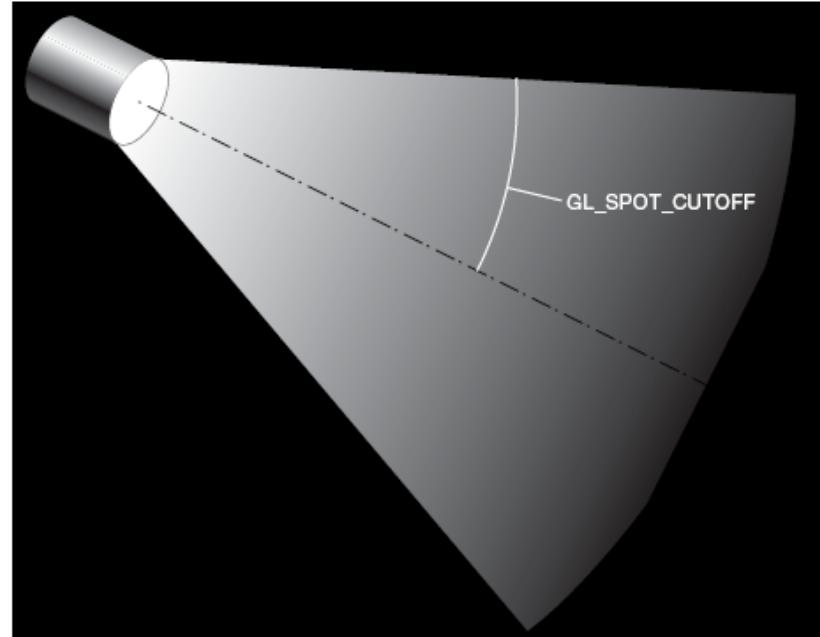
Fényforrás pozíciója

- **GL_POSITION** paraméter
 - a pozíció (x, y, z, w) koordinátai
 - $w \neq 1$: pontszerű fényforrás vagy szpotlámpa
 - $w=0$: végtelenben lévő fényforrás = irányforrás
 - default: (0.0, 0.0, 1.0, 0.0)
- Példa: pontforrás inicializálás és beállítása:

```
GLfloat light_position[] = { 1.0, 1.0, 1.0, 1.0 };  
glLightfv(GL_LIGHT0, GL_POSITION, light_position);  
glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 2.0);  
glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 1.0);  
glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, 0.5);  
 glEnable(GL_LIGHT0);
```

Spotlight (reflektor)

- Irány és vágási szög:
 - **GL_SPOT_DIRECTION**
 - **GL_SPOT_CUTOFF**



- Minta kód:

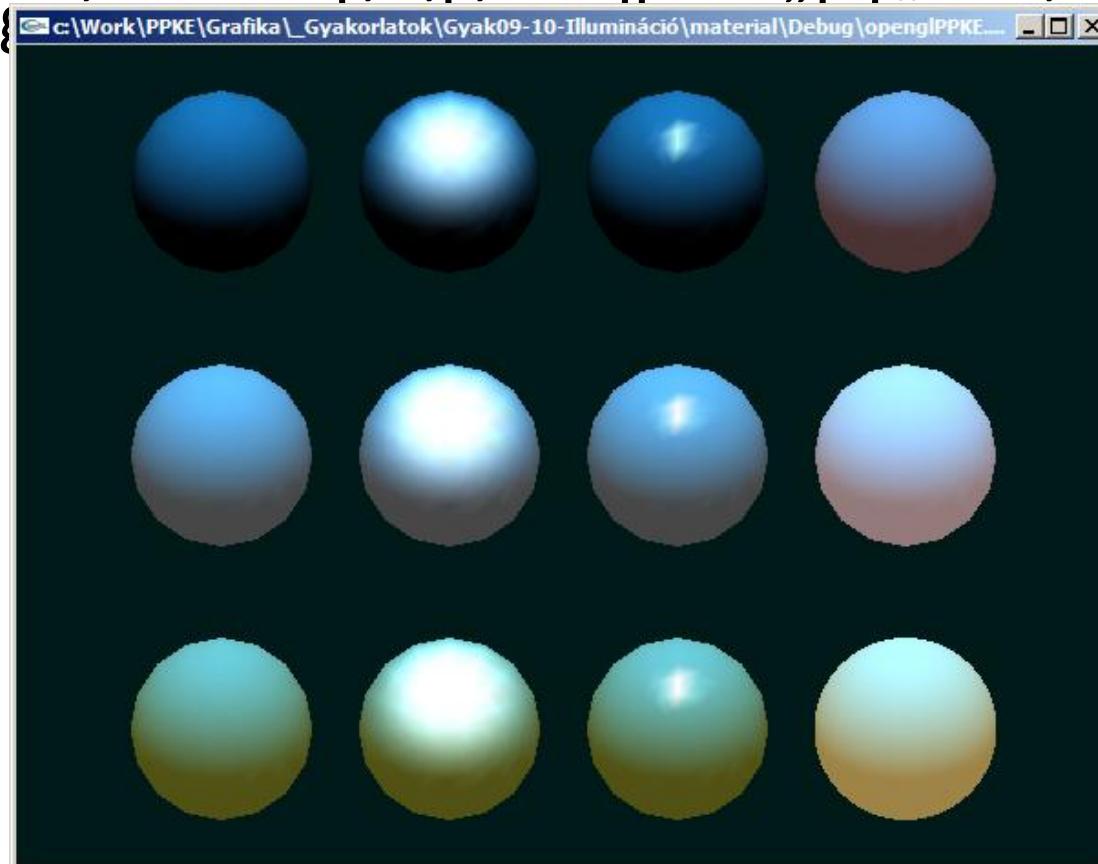
```
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 45.0);  
GLfloat spot_direction[]={-1.0,-1.0, 0.0};  
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION,  
spot_direction);
```

Anyagtulajdonságok

- void **glMaterial{if}**(GLenum *face*, GLenum *pname*,
TYPE param);
 - *face*: GL_FRONT, GL_BACK, vagy GL_FRONT_AND_BACK
 - *pname*: GL_AMBIENT, GL_DIFFUSE,
GL_AMBIENT_AND_DIFFUSE, GL_SPECULAR
GL_SHININESS (*n* kitevő a Phong-Blinn illuminációs
képletben); GL_EMISSION (az anyag által kibocsátott
fény színe – ha van)

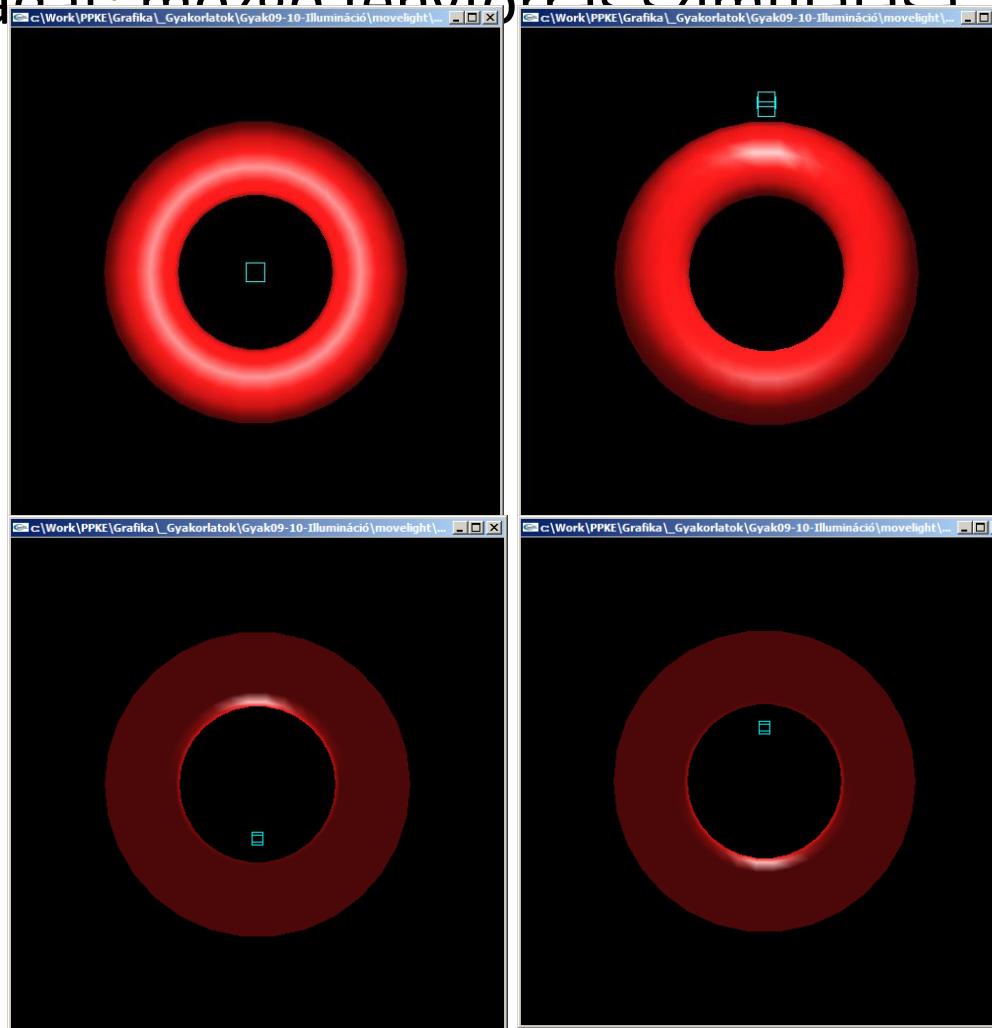
Gyakorlati feladatok

- 1. gömb megjelenítése és árnyalása különböző anyagokban a következő képen



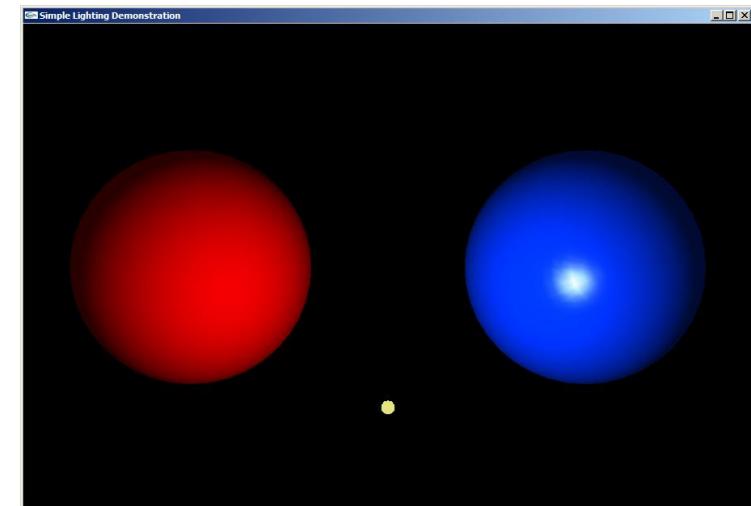
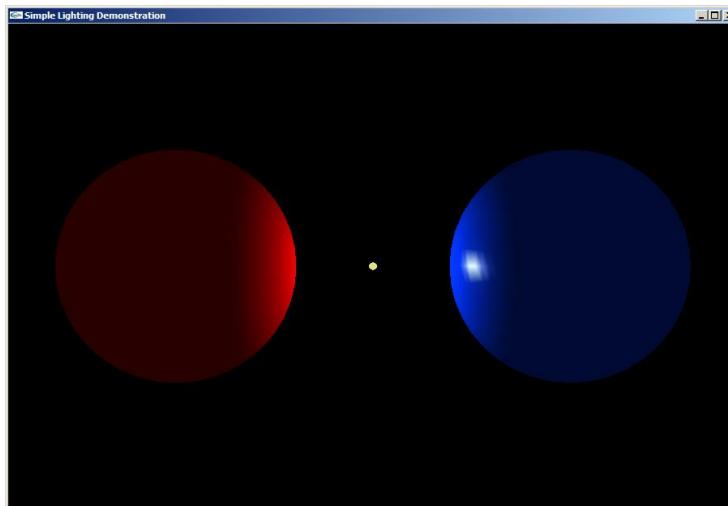
Gyakorlati feladatok

- 2. feladat: mozgó fényforrás szimulálása



Gyakorlati feladatok

- 3. Különböző tulajdonságú gömbök között mozgó fényforrás



Gyakorlati feladatok

- 4. szoba megjelenítő teáskannával

