Környezet-leképezés

Szécsi László

3D Grafikus Rendszerek

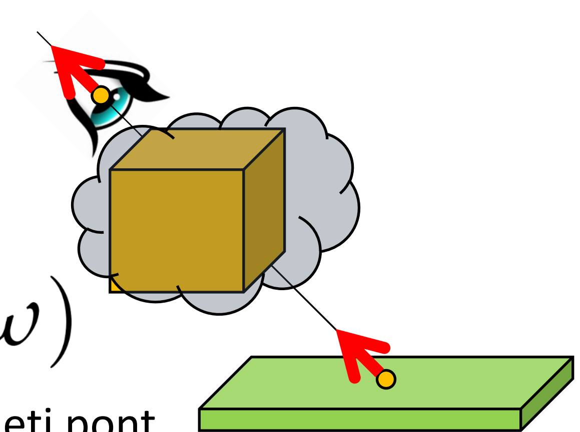
10. előadás

Radiancia $L(\boldsymbol{x}, \omega)$

- Sűrűség a pozícióra és az irányokra nézve egyaránt
- Kifejezi, mennyi fény halad
  - a tér egy bizonyos pontjában
  - egy bizonyos irányba

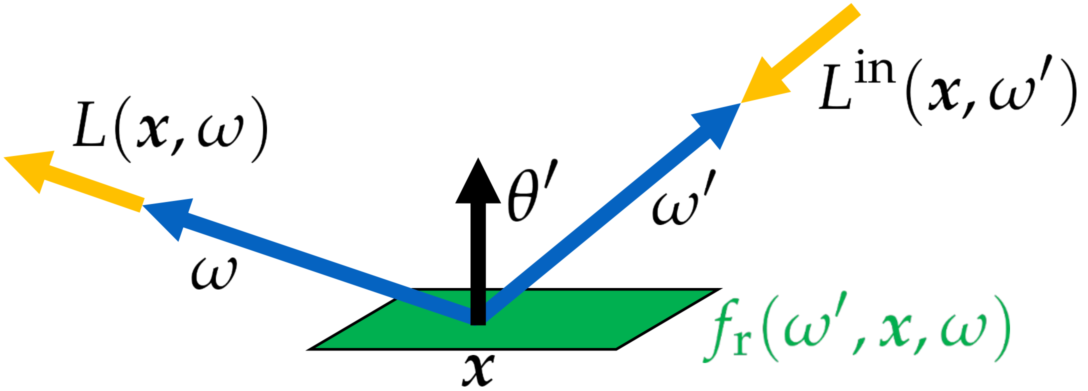
szem

$$L(\boldsymbol{e}, \omega) = L(\boldsymbol{x}, \omega)$$



felületi pont

Az árnyalási egyenlet



az ω irányba visszavert radiancia

minden bejövő ω' irányra összegezve

az onnan bejövő L^{in} radiancia, szorozva

az ω irányba történő visszaverődés valószínűségével

$$L(\boldsymbol{x},\omega) = \int_{\Omega} L^{\mathrm{in}}(\boldsymbol{x},\omega')\cos\theta' f_{\mathrm{r}}(\omega',\boldsymbol{x},\omega)\mathrm{d}\omega'$$

Különleges esetek

- ideális, sima felület
  - csak egy adott irányból bejövő fény verődhet vissza a kimenő irányba
  - csak egy adott irányból bejövő fény törhet a kimenő irányba

  nem kell integrál
- csak egy irányból jön be fény

Sima felületek

Dirac-delta az ω_{r} ideális visszaverődési iránynál

$$L(\boldsymbol{x},\omega) = \int_{\Omega} L^{\mathrm{in}}(\boldsymbol{x},\omega')\cos\theta' f_{\mathrm{r}}(\omega',\boldsymbol{x},\omega)\,\mathrm{d}\omega'$$

$$L(\boldsymbol{x},\omega) = L^{\mathrm{in}}(\boldsymbol{x},\omega_{\mathrm{r}})R + L^{\mathrm{in}}(\boldsymbol{x},\omega_{\mathrm{t}})T$$

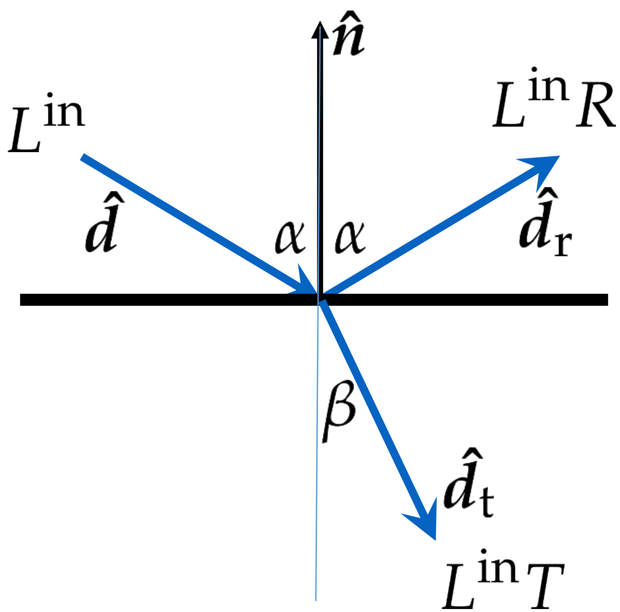
ideális visszaverődési irány

transzmittancia

reflektancia

ideális törési irány

Irányok meghatározása



L_{in}	bejövő radiancia	
$\hat{\boldsymbol{n}}$	felületi normálvektor	ismert
$\hat{\boldsymbol{d}}$	bejövő fényirány	
R	reflektancia	
T	transzmittancia	
α	beesési/visszavert szög	
β	tört szög	
$\hat{\boldsymbol{d}}_{\text{r}}$	ideális visszaverődés iránya	
$\hat{\boldsymbol{d}}_{\text{t}}$	ideális törés iránya	keresett

Törésmutató és reflektancia

- a reflektancia, transzmittancia és a törési szög az anyagi jellemzőktől függenek
  - törésmutató μ
  - kioltási tényező κ
- reflektancia, transzmittancia meghatározása $T = 1 - R$
  - legegyszerűbb modell: konstans reflektancia (tükör színe)
  - Fresnel Szirmay-féle közelítéssel:

$$R \approx R_0 + (1 - R_0)(1 - \cos\alpha)^5, \quad R_0 = \frac{(\mu - 1)^2 + \kappa^2}{(\mu + 1)^2 + \kappa^2}$$

- törési szög meghatározása
  - Snelius-Descartes

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

Elnyelő és átlátszó anyagok

Elnyelő anyag: fém

- tört rész elveszik
- reflektancia hullámhosszfüggő
- arany:

$$\dddot{\mu} = \begin{pmatrix} 0.21 & 0.485 & 1.29 \end{pmatrix}$$

$$\dddot{\kappa} = \begin{pmatrix} 3.13 & 2.23 & 1.76 \end{pmatrix}$$

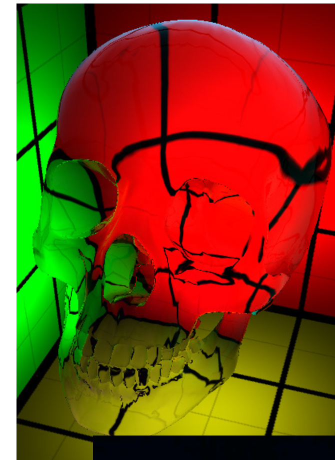
- ezüst:

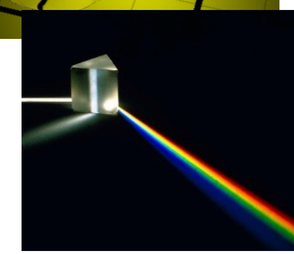
$$\dddot{\mu} = \begin{pmatrix} 0.15 & 0.14 & 0.13 \end{pmatrix}$$

$$\dddot{\kappa} = \begin{pmatrix} 3.7 & 3.11 & 2.47 \end{pmatrix}$$

Átlátszó anyag:

dielektrikum, víz





- egy törésirányt akarunk, törésmutató ne függjön a hullámhossztól
- üveg: $\kappa = 0$

$$\mu = 1.46$$

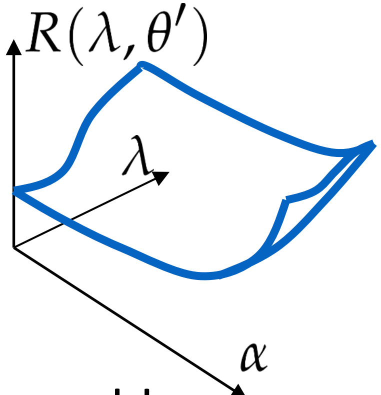
- víz:

$$\mu = 1.333$$

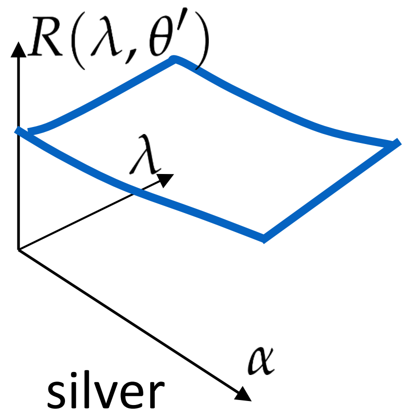
Elnyelő anyagok

fémek

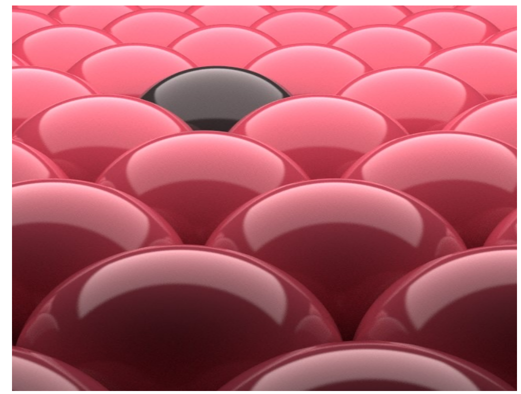
A reflektancia hullámhosszfüggő

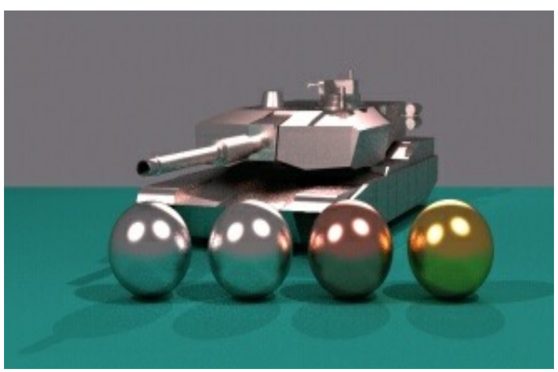


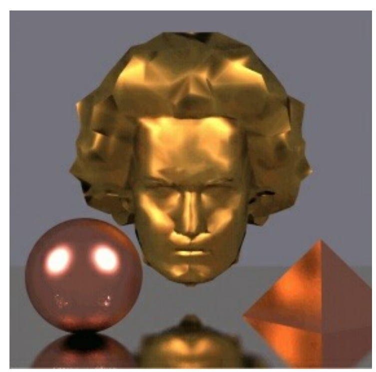
gold



silver

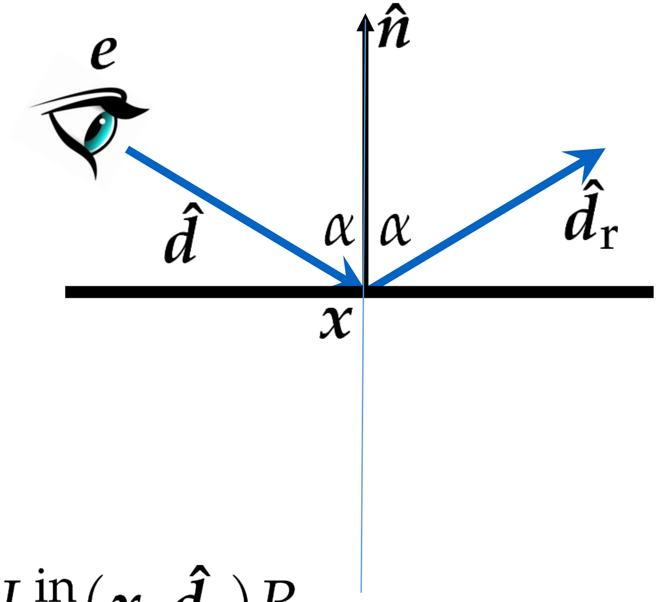






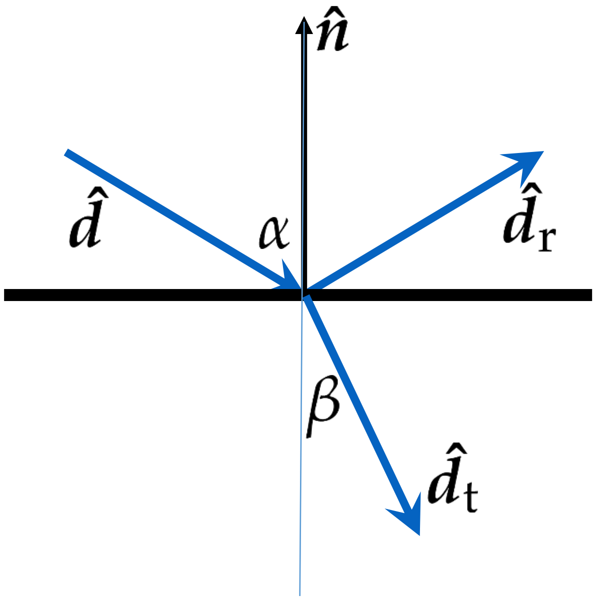
Elsődleges sugár esete

$$\hat{\boldsymbol{d}} = -\hat{v}$$



$$L(\boldsymbol{e}, -\hat{\boldsymbol{d}}) = L^{\mathrm{in}}(\boldsymbol{x}, \hat{\boldsymbol{d}}_{\mathrm{r}})R$$

Normálvektor és törésmutató



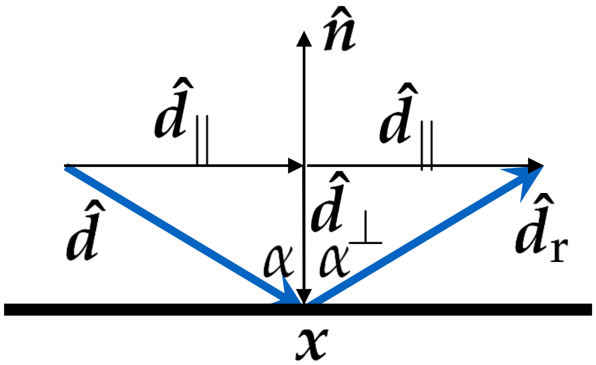
$$\cos\alpha = -\hat{\boldsymbol{d}} \cdot \hat{\boldsymbol{n}}$$

ha negative:
másik oldalról érkezik

$$\hat{\boldsymbol{n}} \leftarrow -\hat{\boldsymbol{n}}$$

$$\mu \leftarrow \frac{1}{\mu}$$

Ideális visszaverődés iránya



$$\cos\alpha = -\hat{n}\cdot\hat{d}$$

$$\hat{d}_{\perp} = (\hat{n}\cdot\hat{d})\hat{n}$$

$$\hat{d}_{\|} = \hat{d} - \hat{d}_{\perp}$$

$$\hat{d}_{\mathrm{r}} = \hat{d}_{\|} - \hat{d}_{\perp}$$

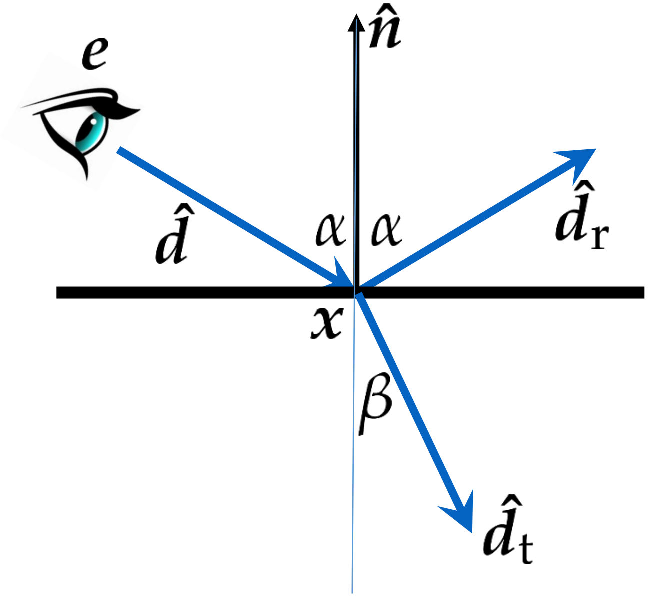
```
vec3 dr = reflect(d, normal);
```

Átlátszó anyagok

üveg, víz, dielektrikumok

Elsődleges sugár esete

$$\hat{\boldsymbol{d}} = -\hat{\boldsymbol{v}}$$



$$L(\boldsymbol{e}, -\hat{\boldsymbol{d}}) = L^\mathrm{in}(\boldsymbol{x}, \hat{\boldsymbol{d}}_\mathrm{r})R + L^\mathrm{in}(\boldsymbol{x}, \hat{\boldsymbol{d}}_\mathrm{t})T$$

Ideális visszaverődés iránya

- ugyanaz, mint az elnyelőnél

Törési szög koszinuszának meghatározása

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

Snelius-Descartes

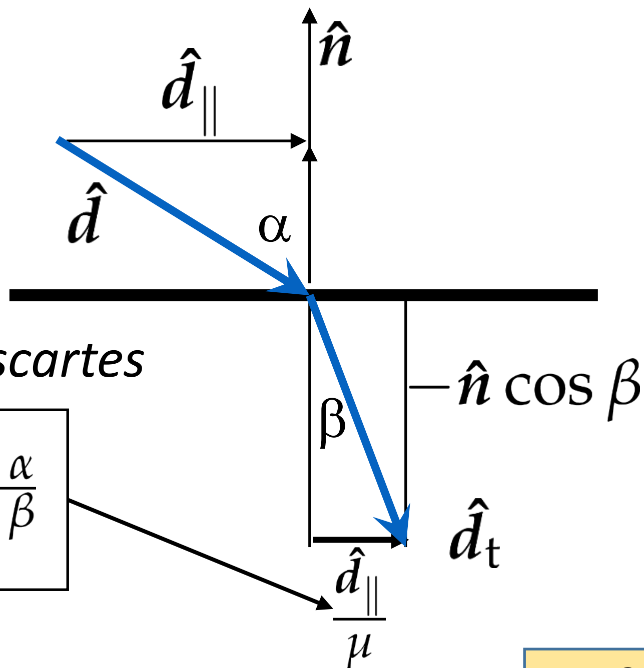
$$\cos\beta = \sqrt{1-\sin^2\beta} = \sqrt{1-\frac{\sin^2\alpha}{\mu^2}} = \sqrt{\underbrace{1-\frac{1-\cos^2\alpha}{\mu^2}}}$$

Pitagorasz

ha negatív: teljes belső visszaverődés

$$R \leftarrow 1$$
$$\cos\beta \leftarrow 0$$

Ideális törési irány meghatározása



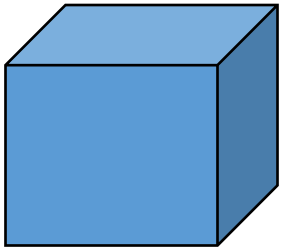
Snelius-Descartes

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

$$\hat{\boldsymbol{d}}_{\mathrm{t}} = \frac{\hat{\boldsymbol{d}}_{\|}}{\mu} - \hat{\boldsymbol{n}}\cos\beta$$

```
vec3 dt = refract(d, normal, mu);
```

Cube map



- 6 szeletes textúratömb
- kocka felületén
- minden irányt lefed
- irány függvényében is tudunk tárolni valamit
- HW: a textúra irányvektorral címezhető
  - kiszámítja melyik lap milyen u,v

Environment map

- nem a felületi jellemzőket, hanem a megvilágítást tároljuk textúrában
- $L_{\mathrm{env}}(\omega)$ – bejövő radiancia az irány függvényében
  - nem függ a felületi pont pozíciójától
  - végtelen távoli környezet
- minden irányból jön be fény
  - összes texelre kellene összegezni
  - DE: ideális tükör: csak 1 irányból érdekes mi jön be

Env map FS konstans 1 reflektanciával

```
uniform struct {
  vec3 position;
} camera;
uniform struct {
  samplerCube background;
} material;
in vec4 worldPos;
in vec4 worldNormal;
void main(void) {
  vec3 normal = normalize(worldNormal.xyz);
  vec3 x = worldPos.xyz / worldPos.w;
  vec3 viewDir = normalize(camera.position - x);
  fragmentColor = texture(background,
               reflect(-viewDir, normal));
}
```

Hogyan rajzoljuk magát a környezetet?

- nagy valamire feszítve
  - sky box
  - sky dome
- full viewport quad
  - teljes nézeti ablakot lefedő négyszög
  - a pixel shaderben megkeressük az irányt
  - Z bufferelés: mindennél hátrább van
    - ezt rajzoljuk először VAGY
    - a full-screen quad Z-je legyen 0.9999 (norm. képenyőkoordinátában)

Sugárirány kiszámítása NDC-ből

$$\boldsymbol{x}_{\mathrm{w}}\boldsymbol{VP} = \boldsymbol{x}_{\mathrm{ndc}}$$

$$\boldsymbol{d} = \boldsymbol{x}_{\mathrm{w}} - \boldsymbol{e}$$

$$\hat{\boldsymbol{d}} = \frac{\boldsymbol{d}}{|\boldsymbol{d}|}$$

$$\boldsymbol{d} = \boldsymbol{x}_{\mathrm{ndc}}(\boldsymbol{VP})^{-1} - \boldsymbol{e}$$

$$\boldsymbol{d} = \boldsymbol{x}_{\mathrm{ndc}}(\boldsymbol{VP})^{-1}\boldsymbol{E}^{-1}$$

$$\boldsymbol{d} = \boldsymbol{x}_{\mathrm{ndc}}(\boldsymbol{EVP})^{-1}$$

(**EVP**)$^{-1}$ -et nevezzük rayDirMatrix-nak

Konstruktorban, ill. mozgatás után

```
this.rayDirMatrix = new Mat4();
this.updateRayDirMatrix();
```

Full viewport quad VS

```
in vec4 vertexPosition;
in vec2 vertexTexCoord;
out vec4 rayDir;

uniform mat4 rayDirMatrix;

void main(void) {
  rayDir = vertexPosition
                            * rayDirMatrix;
  gl_Position = vec4(vertexPosition.xy, 0.999, 1);
}
```

homogén
osztással mi lesz?

minden mögé

Háttér FS

```
uniform vec3 eye;
in vec4 rayDir;
uniform samplerCube background;

void main(void) {
  vec3 d = normalize(rayDir.xyz);

  fragmentColor = texture (background, d);
}
```