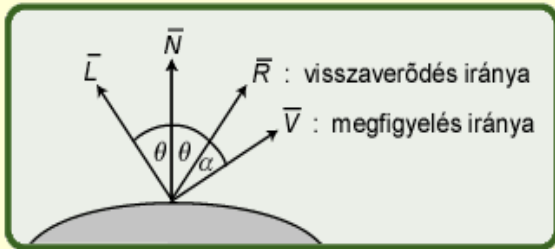


Tükröző (specular) visszaverődés

Fényes felületekről (tükörről)



1

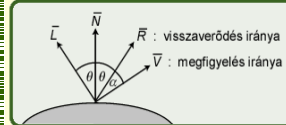
Tükröző (specular) visszaverődés

Phong-féle modell:

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

n : a tükrözési visszaverődés kitevője (csillogás) (tompá) $1 \leq n \leq 1000$ (éles fény)

$W(\theta)$: a tükrözötten visszaverődő fény hányada, lehet konstans, k_s ($0 \leq k_s \leq 1$),



2

Tükröző (specular) visszaverődés

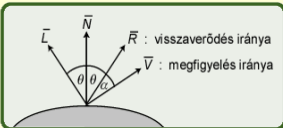
Phong-féle modell:

A tárgy anyagát is figyelembe véve:

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

Több fényforrásra:

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + \sum_i f_{atti} I_{p\lambda i} [k_d O_{d\lambda} (\underline{N} \underline{L}_i) + k_s O_{s\lambda} (\underline{R}_i \underline{V})^n]$$



3

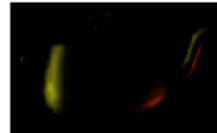
Megvilágítási modellek (példa)



szórt



diffúz



tükröző, csillogás = 20



szórt + diffúz + tükröző

4

Poligonokból álló felületek fényességének meghatározása

- Minden pontban kiszámítjuk a megvilágítási egyenlet szerinti intenzitást (nagyon drága módszer)

5

Poligonokból álló felületek fényességének meghatározása

1. Konstans fényesség

Az egész poligon ugyanolyan intenzitású. Jó, ha:

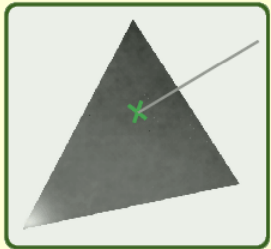
- végtelen távoli fényforrás (\underline{N} , \underline{L} konstans)
- végtelen távoli megfigyelő (\underline{N} , \underline{V} konstans)
- poligon oldalú felület



6

Poligonokból álló felületek fényességének meghatározása

2. Interpolált fényesség

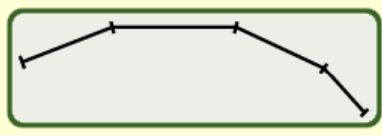


Az intenzitást a csúcsokban számított intenzitásból kapjuk interpolációval

7

Poligonokból álló felületek fényességének meghatározása

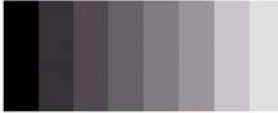
3. Poligon-hálózat fényessége



Az egyes poligonok konstans fényessége csak kiemelné a poligonok közötti éleket

8


Mach-féle hatás



Az intenzitás változását eltűnőve érezzük ott, ahol az intenzitás folytonossága megszűnik.

Érzékelt fényesség

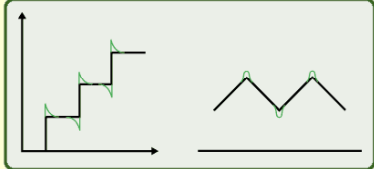
Tényleges megvilágítás



9

Poligonokból álló felületek fényességének meghatározása

3. Poligon-hálózat fényessége



Mach-hatás

Megoldás:
minden poligon fényességét változó intenzitásúnak generáljuk

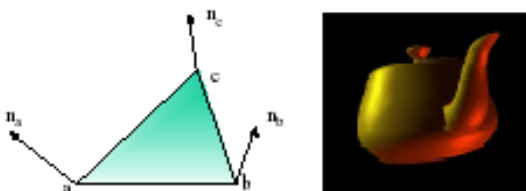
10

Gouraud-féle fényesség

1. A poligonok normálisait ismerve határozzuk meg a csúcspontok normálisait (pl. az ott érintkező poligonok normálisainak átlagaként)
2. Számoljuk ki az intenzitásokat a csúcspontokban
3. Az élek mentén lineáris interpolációval számoljuk az intenzitást
4. Az élek között (a pásztázó vonalak mentén) lineáris interpolációval számoljuk az intenzitást

11

Gouraud-féle fényesség



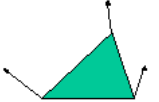
12

Phong-féle fényesség

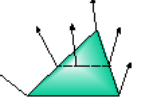
Phong-féle fényesség:

1. Számoljuk ki a normálvektorokat a csúcspontokban,
2. Interpolációval a csúcspontok között az élek mentén a normálvektorokat,
3. Interpolációval az élek között,
4. Intenzitás számítása

Sokszor jobb, mint a Gouraud-féle módszer




Gouraud



Phong

13

Gouraud- és Phong-féle fényesség



Konstans

Gouraud

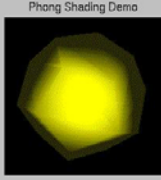
Phong

14

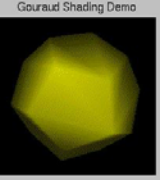
Példák

Gömb közelítése szabályos dodekaéderrel

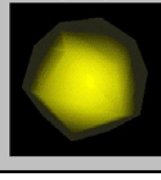
$n=100$



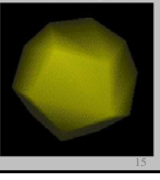
Phong Shading Demo

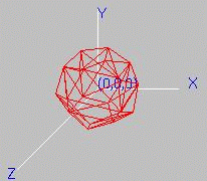


$n=800$



Phong Shading Demo






15

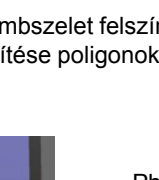
Példák

Gömbszelet felszín közelítése poligonokkal




Gouraud

Phong




16


Példák



szórt



Phong (szórt + diffúz)



Phong (szórt + diffúz + tükröző)

17

Fényforrás (OpenGL)

Világítási komponensek

RGBA értékeivel definiálható:

- **szórt** (`GL_AMBIENT`)
- **diffúz** (`GL_DIFFUSE`):
a megvilágított felületről minden irányban azonos a visszaverődés
- **tükröző** (`GL_SPECULAR`):
fényes felületről - csillogás

18

Fényforrás (OpenGL)

A specifikálható fényforrások száma: max. 8

- **pozicionált:** az objektum közelében van
- **irányított:** végtelen távoli a pozícióvektor negyedik koordinátája 0.0

A fényforrás fénysugara:

- szűk
- fókuszált
- széles

19

Fényforrás (OpenGL)

```
void glLight{if}(enum light, enum pname,
                T param);
void glLight{if}v(enum light, enum pname,
                 T *param);
```

light: kijelöli a fényforrást
(GL_LIGHT0, ..., GL_LIGHT7),
pname: a beállítandó tulajdonság,
param: a beállítandó tulajdonság értéke.

20

Fényforrás (OpenGL)

pname	alapértelmezés	Jelentés
GL_AMBIENT	(0.0, 0.0, 0.0, 1.0)	Szórt fény RGBA intenzitása
GL_DIFFUSE	(1.0, 1.0, 1.0, 1.0)	Diffúz fény RGBA intenzitása
GL_SPECULAR	(1.0, 1.0, 1.0, 1.0)	Tükröző fény RGBA intenzitása
GL_POSITION	(0.0, 0.0, 1.0, 0.0)	A fényforrás (x, y, z, w) pozíciója
GL_SPOT_DIRECTION	(0.0, 0.0, -1.0)	A fény (x, y, z) iránya
GL_SPOT_EXPONENT	0.0	Reflektorfény exponens
GL_SPOT_CUTOFF	180.0	Reflektorfény kúpszöge
GL_CONSTANT_ATTENUATION	1.0	Konstans elnyelő faktor
GL_LINEAR_ATTENUATION	0.0	Lineáris elnyelő faktor
GL_QUADRATIC_ATTENUATION	0.0	Négyzetes elnyelő faktor

21

Fényforrás (OpenGL)

A fényforrás távolságával a fény intenzitása gyengül

OpenGL-ben a gyengítő faktor:

$$f_{att} = 1/(e_k + e_l ||VP|| + e_n ||VP||^2),$$

e_k : konstans gyengítő faktor

GL_CONSTANT_ATTENUATION,

e_l : lineáris gyengítő faktor

GL_LINEAR_ATTENUATION,

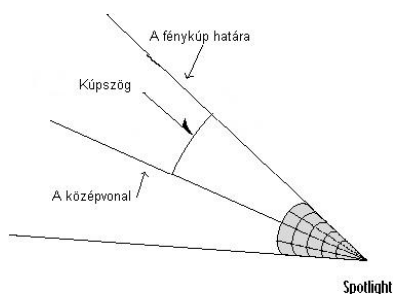
e_n : négyzetes gyengítő faktor

GL_QUADRATIC_ATTENUATION,

$||VP||$: a tárgy és a fényforrás távolsága

22

Reflektorszerű fényforrás (OpenGL)



Középvonal: GL_SPOT_DIRECTION
Intenzitás eloszlás: GL_SPOT_EXPONENT
Kúpszög: GL_SPOT_CUTOFF

23

Világítási modell (OpenGL)

```
void glLightModel{if}(enum pname,
                    T param);
void glLightModel{if}v(enum pname,
                      T *param);
```

pname: a világítási modell tulajdonság kijelölése:

GL_LIGHT_MODEL_AMBIENT a szórt megvilágítás értékeinek megadása

Alapértelmezés: RGBA = (0.2, 0.2, 0.2, 1.0)

24

Világítási modell (OpenGL)

GL_LIGHT_MODEL_TWO_SIDE: egy- vagy kétoldalas világítási számításokat kell alkalmazni a poligonoknál.
Ha **param=0.0**, akkor csak az első oldal világít, különben mindkettő

GL_LIGHT_MODEL_LOCAL_VIEWER: hogyan kell kiszámítani a spekuláris (tükröző) fényvisszaverődés szögét
Alapértelmezés: 0.0: a z tengely irányából, más érték esetén a nézőpontból

25

Objektumok fényvisszaverő tulajdonságai (OpenGL)

Az objektumok tulajdonságai:

- szín komponensek (meghatározzák a fénykomponensek visszavert hányadát),
- fényvisszaverő képesség: szórt, diffúz és tükröző fény számára,
- az objektumok saját fénye, emissziós érték.

26

Objektumok fényvisszaverő tulajdonságai (OpenGL)

Szín:

```
void glColorMaterial(Glenum face,
                    Glenum mode);
```

face: GL_FRONT, GL_BACK, GL_FRONT_AND_BACK
Alapértelmezés: GL_FRONT_AND_BACK

mode: GL_EMISSION, GL_AMBIENT, GL_SPECULAR, GL_DIFFUSE, GL_AMBIENT_AND_DIFFUSE

Alapértelmezés: GL_AMBIENT_AND_DIFFUSE

27

Objektumok fényvisszaverő tulajdonságai (OpenGL)

```
void glMaterial{if}(enum face, enum pname,
                  T param);
void glMaterial{if}v(enum face,
                   enum pname, T params);
```

face: GL_FRONT, GL_BACK, GL_FRONT_AND_BACK
pname: a specifikálandó paraméter neve,
param(s): az érték, vagy értékek, amelyekre a **pname** által jelzett paramétert be kell állítani.

28

Objektumok fényvisszaverő tulajdonságai (OpenGL)

pname	Alapértelmezés	Jelentés
GL_AMBIENT	(0.2, 0.2, 0.2, 1.0)	szórt RGBA fényvisszaverés
GL_DIFFUSE	(0.8, 0.8, 0.8, 1.0)	diffúz RGBA fényvisszaverés
GL_SPECULAR	(0.0, 0.0, 0.0, 1.0)	tükröző RGBA fényvisszaverés
GL_EMISSION	(0.0, 0.0, 0.0, 1.0)	emissziós fény intenzitás
GL_SHININESS	0	tükröző exponens (csillogás)
GL_AMBIENT_AND_DIFFUSE		szórt és diffúz szín együtt
GL_COLOR_INDEXES	(0, 1, 1)	szórt, diffúz és tükröző szín indexek

29

Feladat (OpenGL)

Rajzoljuk meg egy kocka perspektivikus képét megvilágítással és a kocka saját színeivel!

```
main-be:
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
SetupRC-be:
glEnable(GL_DEPTH_TEST);
RenderScene-be:
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glEnable(GL_DEPTH_TEST);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(fovy, aspect, near, far);
gluLookAt(ex,ey,ez, cx,cy,cz, ux,uy,uz);
```

30

ÁRNYÉKOLÁS

Az árnyékolás általánosan
Egyszerű árnyék
Árnyék generálása pásztázó-vonal
algoritmussal
Kétmenetes árnyékolási algoritmus
Árnyék tér
Kétmenetes z-pufferes árnyékolási algoritmus

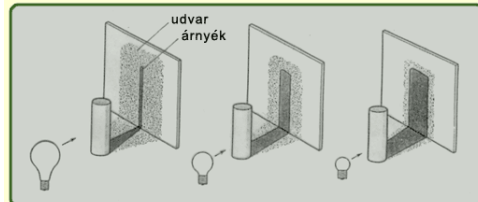
31

Az árnyékolás általánosan

Az árnyékolás általánosan Árnyék (Shadow)

Algoritmus, amely meghatározza, hogy melyik felszín látható a fényforrásból nézve

Általában bonyolult (sok mindentől függ, pl. a fényforrás méretétől)



32

Az árnyékolás általánosan

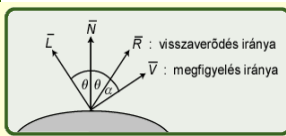
Egyszerűsítés:
Pontszerű fényforrásokra:

$$I_\lambda = I_{ba} k_a O_{d\lambda} + \sum_i S_i f_{att} l_{p\lambda} [k_d O_{d\lambda} (N L_i) + k_s O_{s\lambda} (R_i V)^n]$$

Ahol

$$S_i = \begin{cases} 0 & \text{ha az } i \text{ forrásból ez} \\ & \text{a pont nem látszik} \\ 1 & \text{különben.} \end{cases}$$

Csak poligon határu testekkel foglalkozunk egyetlen fényforrás esetén: egyszerű árnyék

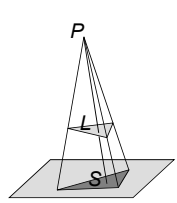


33

Egyszerű árnyék

1. (Blinn 1988) Egyetlen tárgy árnyéka sík felszínen

Pl. pontszerű fényforrás esetén az árnyék a vízszintes síklapon ($z_s = 0$) perspektív vetítéssel

$$M_{per}^i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{pmatrix}$$


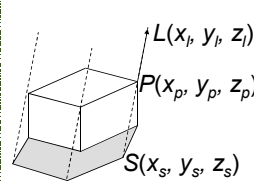
34

Egyszerű árnyék

Tetszőleges helyzetű síklap: transzformáció, vetítés

Pl.: párhuzamos megvilágításnál: síklap $\rightarrow z = 0$

$$x_s = x_p - z_p x_l / z_l$$

$$y_s = y_p - z_p y_l / z_l$$


$$M_{par} = \begin{pmatrix} 1 & 0 & -x_l/z_l & 0 \\ 0 & 1 & -y_l/z_l & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

35

Egyszerű árnyék

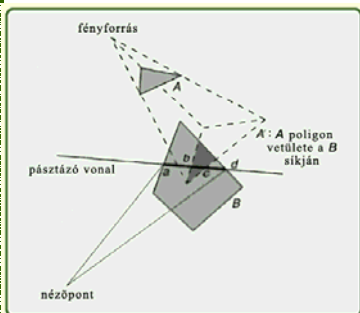
Árnyékolás:

poligon (tárgy) $\xrightarrow{\text{projekció}}$ vetített poligon $\xrightarrow{\text{pásztázás}}$ z-buffer

36

Egyszerű árnyék

2. Árnyék generálás pásztázó-vonal algoritmussal (APPEL 1968, BOUKNIGHT, KELLEY 1970)



A látható felszín meghatározására szolgáló pásztázó-vonal algoritmus kibővítése árnyék generálással

37

Árnyék generálása pásztázó-vonal algoritmussal

A **B** poligonon az **A** poligon vetülete (**A'** poligon) adja meg az árnyékot

Tehát a pásztázó vonal és **A'** metszéspontjai határozzák meg az árnyék határait

Az árnyék „ki-be kapcsolása” ugyanúgy történik, mint a látható felszín meghatározásánál

Sok számolás: n poligon esetén $n(n-1)$ vetület számítása

38

Árnyék generálása pásztázó-vonal algoritmussal

Adatstruktúra: látható felszín, árnyék

Algoritmus:

Pásztázás - a látható szakaszokra megvizsgáljuk, hogy árnyékban vannak-e

39

Árnyék generálása pásztázó-vonal algoritmussal

Esetek:

Ha az aktuális pásztázandó szakaszhoz nincs árnyék poligon, akkor normális pásztázás, különben

ha az árnyék poligon nem takarja a pásztázó szakaszt, akkor normális pásztázás,

ha teljesen takarja, akkor árnyékolás,

ha részben takarja, akkor a pásztázó szakaszt részekre osztjuk, és a részekre megismételjük az eljárást.

40

Egyszerű árnyék

3. Kétmenetes árnyékolási algoritmus

(ATHERON, WEILER, GREENBERG 1978)

Észrevétel: csak azokat a poligonokat kell árnyékolni, amelyek

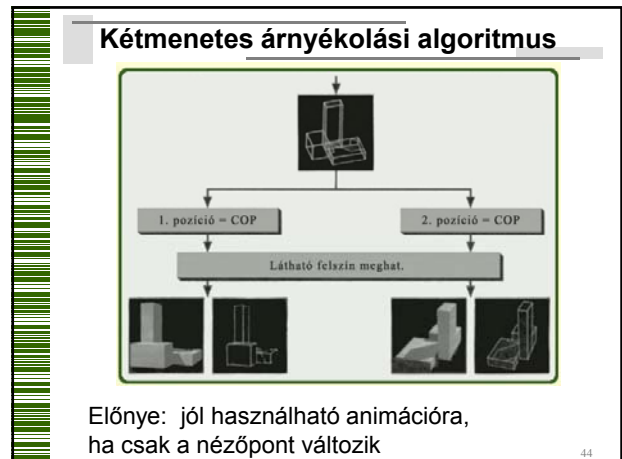
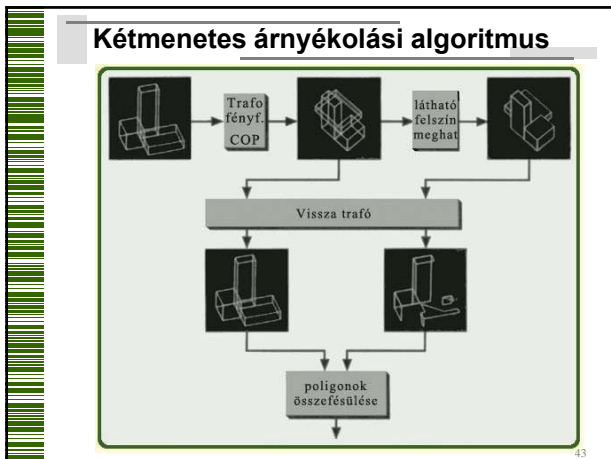
- láthatók a nézőpontból és
- nem láthatók a fényforrásból

41

Kétmenetes árnyékolási algoritmus

1. lépés: A fényforrásból látható felszín meghatározása (a megvilágított poligonok listája)
 - a) transzformáció: vetítés a fényforrásból
 - b) látható felszín meghatározása
 - c) vissza transzformáció
2. lépés: a poligonok adatbázisának összefésülése (mi van árnyékban és mi nincs)
3. lépés: a nem látható felszínek eltávolítása (a látható poligonok listája)
4. lépés: megjelenítés
pl. pásztázó vonal algoritmussal

42



Egyszerű árnyék

4. Kétmenetes z-pufferes árnyékolási algoritmus

1. lépés: számítsuk ki a z-puffert a fényforrásból nézve
2. lépés: ha egy pont látható (a megfigyelő helyéről), akkor transzformáljuk a fényforrás-középpontú vetítési rendszerbe, ahol a z-puffer tartalmából eldönthető, hogy árnyékban van-e

≡

a transzformált pont távolabb van-e a fényforrástól, mint a z-pufferhez tartozó pont?

45

Egyszerű árnyék

5. Árnyék tér (Shadow Volumes) (CROW 1977)

A tárgy árnyék tere: a térnek az a része, amit a tárgy eltakar a fényforrás elől. Poligonok és „árnyék poligonok” határolják. A fényforrás helyéből és a tárgy kontúrvonalaiából meghatározható az árnyék tér.

Az árnyék poligonokat nem kell megjeleníteni, de felhasználhatók az árnyékoláshoz.

46

Árnyék tér

Az árnyék poligonok síkjának normálisa mutasson az árnyék térből kifelé

A nézőpont felé néző árnyék poligon: $s(P)=s(P)+1$

Hátra néző árnyék poligon: $s(P)=s(P)-1$

Árnyékszám a P pontban: $s(P)$
 P árnyékban van $\leftrightarrow s(P) > 0$

47

Árnyék tér

Példák: Árnyékban van-e **A**, **B** és **C**?
V a megfigyelő

$VA : +1 \rightarrow s(A) = 1$	$VA : 1 - 1 + 1 \rightarrow s(A) = 1$
$VB : +1 - 1 \rightarrow s(B) = 0$	$VB : 1 + 1 - 1 \rightarrow s(B) = 1$
$VC : +1 + 1 \rightarrow s(C) = 2$	$VC : 1 - 1 + 1 + 1 \rightarrow s(C) = 2$

48

SUGÁRKÖVETÉS – RAY TRACING

Általános sugárkövetés

Sugárkövetés látható felszín meghatározására

Metszéspontok kiszámítása

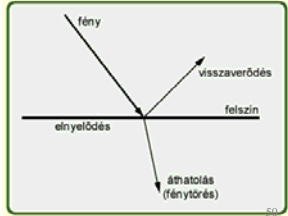
49

SUGÁRKÖVETÉS – RAY TRACING

Általános sugárkövetés
Módszer realiztikus képek előállítására.

Alapelv: A képen látható felszíni pontok színe (fényessége) más felszíni pontokból kiinduló fénysugarak hatásának az eredménye

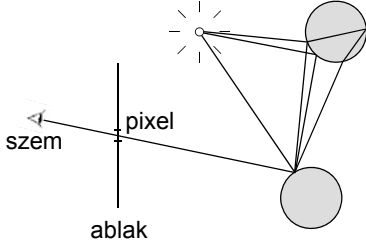
Lehetséges hatások:



50

SUGÁRKÖVETÉS – RAY TRACING

Kövessük a fénysugár útját a nézőponttól kiindulva visszafelé:



51

SUGÁRKÖVETÉS – RAY TRACING

Ha egy „fénysugár” találkozik egy tárggyal, akkor a tárgy felszínének az a pontja olyan színű lesz, amit

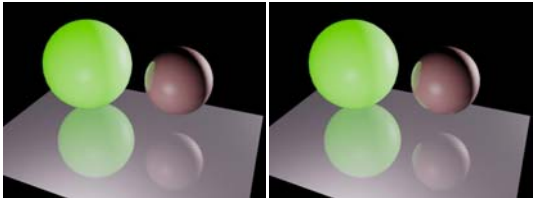
- ♦ a pont által kisugárzott,
- ♦ a pontból az adott irányban visszavert (diffúz és tükröző), valamint
- ♦ a ponton áthatolt fénysugarak együttesen határoznak meg.

Ezeknek a fénysugaraknak a színét úgy határozhatjuk meg, hogy az útjukat ugyanígy követjük visszafelé (rekurzió)

Rekurzió 3-4 mélységig

52

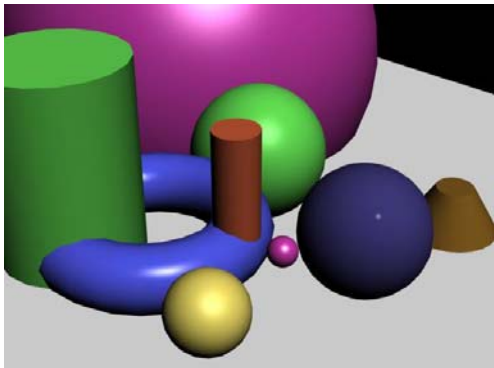
SUGÁRKÖVETÉS – RAY TRACING



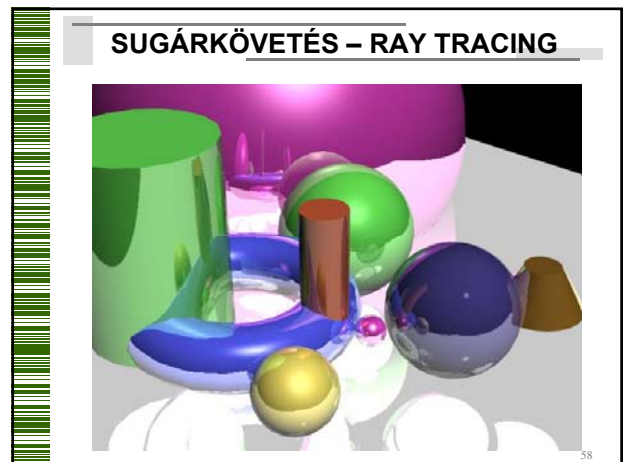
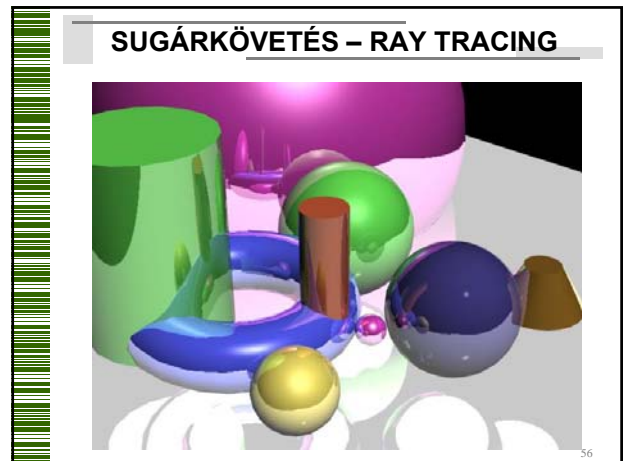
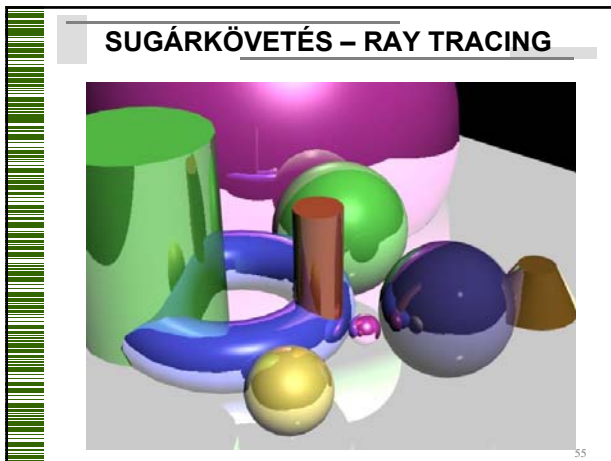
Rekurzió 2 *illetve nagyobb* mélységig

53

SUGÁRKÖVETÉS – RAY TRACING



54



SUGÁRKÖVETÉS – RAY TRACING

Sok geometriai számítás
(metszéspontok, tartalmazás, ...)

Részei:

- Látható felszín meghatározása
- Direkt megvilágítás számolása
- Globális megvilágítás számolása
- Árnyék meghatározása
- ...

59

SUGÁRKÖVETÉS – RAY TRACING

A sugarak követése független egymástól

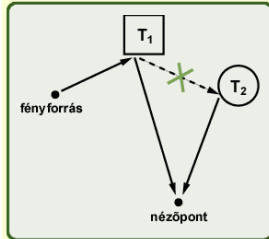
↓

Parallel feldolgozás (transzputer)

60

SUGÁRKÖVETÉS – RAY TRACING

Egyszerűsítési lehetőség: pl. Csak az egyszeres fényvisszaverődést vesszük figyelembe. Akkor csak a direkt megvilágítással kell számolnunk (a más tárgyakról visszavert fényt nem vesszük figyelembe)



61

SUGÁRKÖVETÉS – RAY TRACING

Program:

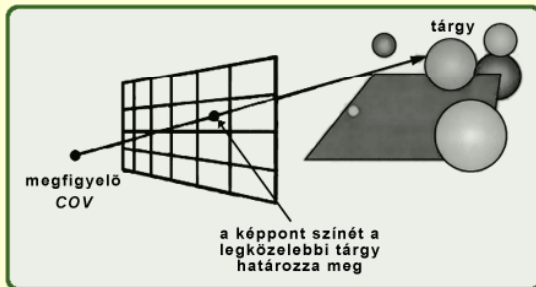
```

COV és az ablak kiválasztása;
for minden pásztázó vonalra do
  for minden képpontra do
    begin
      fény sugár meghatározása;
      for minden tárgyra do
        if a tárgyat metszi a fény sugár és
          eddig ez az első metszés pont
        then jegyezzük meg a metszés pontot
          és a tárgyat;
      Az első metszés ponthoz tartozó tárgy
      színének megfelelő lesz a képpont
    end;
  
```

62

Sugárkövetés látható felszín meghatározására

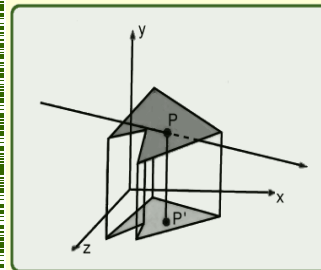
1. Visszafelé követjük a képzeletbeli fény sugár útját a megfigyelőtől a tárgyig



63

Sugárkövetés látható felszín meghatározására

2. Megvizsgáljuk, hogy a poligon síkjával alkotott metszés pont a poligon belsejében van-e



Párhuzamosan vetítünk pl. az xz síkra – az y koordinátát elhagyjuk – majd megnézzük, hogy P' a vetületben van-e

64

Sugárkövetés látható felszín meghatározására

Tapasztalat:

Sugárkövetésnél az idő 75-95%-a a metszés pontok kiszámításával telik el.

Gyorsítási lehetőségek:

- konstans kifejezések kiszámítása előre,
- poligonok vetületének kiszámítása előre,
- határoló testek használata,
- tárgyak hierarchikus struktúrába való rendezése, hogy minél kevesebb metszés pontot számoljunk.

65

Metszés pontok kiszámítása

A nézőpont (COV): (x_0, y_0, z_0)

A képpont közepe: (x_1, y_1, z_1)

A sugár parametrikus egyenlete:

$$x = x_0 + t(x_1 - x_0) = x_0 + t \Delta x$$

$$y = y_0 + t(y_1 - y_0) = y_0 + t \Delta y$$

$$z = z_0 + t(z_1 - z_0) = z_0 + t \Delta z$$

66

Metszéspontok kiszámítása

Metszéspont poligonnal:

Először meghatározzuk az egyenes és a poligon síkjának a metszéspontját. A sík egyenlete:

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

A sík és az egyenes metszéspontjára:

$$A(x_0 + t \Delta x) + B(y_0 + t \Delta y) + C(z_0 + t \Delta z) + D = 0$$

$$t = -\frac{Ax_0 + By_0 + Cz_0 + D}{A \Delta x + B \Delta y + C \Delta z}$$

*Ha $A \Delta x + B \Delta y + C \Delta z = 0$,
akkor az egyenes és a sík párhuzamos*

67

Metszéspontok kiszámítása

Metszéspont gömbbel: középpont: (a, b, c) ,
sugár: r

$$(x - a)^2 + (y - b)^2 + (z - c)^2 = r^2$$

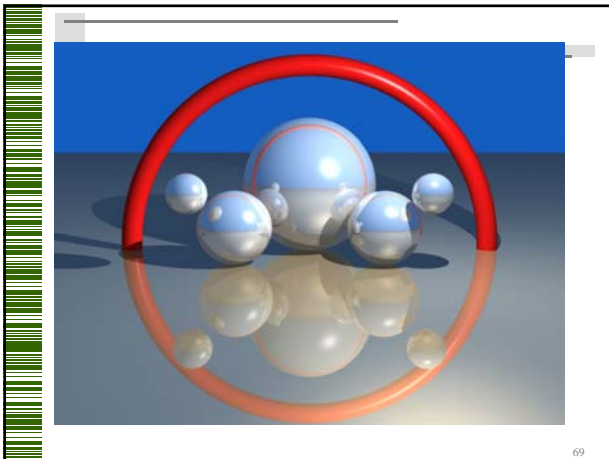
$$(x_0 + t \Delta x - a)^2 + (y_0 + t \Delta y - b)^2 + (z_0 + t \Delta z - c)^2 = r^2$$

t -re nézve másodfokú egyenlet, $2, 1, 0$ db megoldás

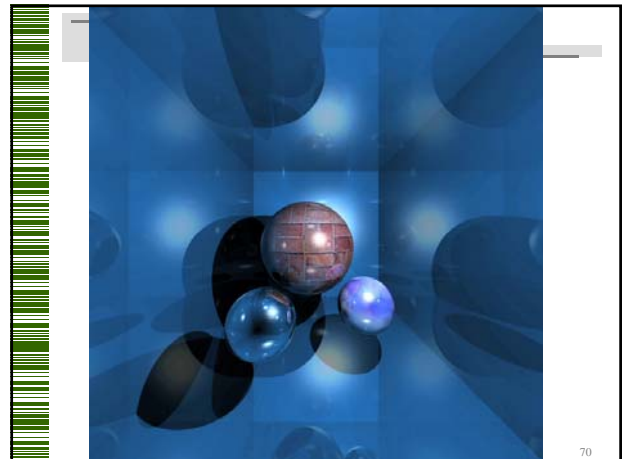
Normális vektor az (x, y, z) metszéspontban:

$$\left(\frac{x-a}{r}, \frac{y-b}{r}, \frac{z-c}{r} \right)$$

68



69



70



71



