

Sugárkövetési algoritmusok (2. rész)

Ismét jelentkezik a sugarak szerelmeseinek szóló cikkünk, melyben tovább folytatjuk a fények birodalmában megkezdett utazásunkat. A fénysugarak rekurzív követésével olyan jelenségek valóság-hű szintézise válik lehetővé, amelyek más eljárásokkal csak igen nehézkesen állíthatóak elő. Részben ennek is köszönhető a több mint 26 éves múltra visszatekintő algoritmus aktualitása napjainkban is.



■ Sugárkövetés a 16. században
(Albrecht Dürer – Underweysung der Messung mit dem Zirkel und Richtscheit)



■ 1. ábra A rekurzív sugárkövetés lehetővé teszi az árnyékok, a tükröződés és a fénytörés modellezését

■ Az előző cikkben beszéltünk a sugárkövetés alapvető elemeiről és fogalmairól, valamint a *sugárvetés* (*ray casting*) megvalósításáról. Ebben a cikkben szó szerint tovább lépünk a fények útján, azaz a tükrözött és tört sugarak követését is megkíséreljük. De lesznek még itt további finomságok is, megismerkedünk két árnyalási modellel, további másodfokú *implicit* felületekkel, valamint egy procedurális objektummal. Meginvitálom hát a tisztelt olvasót, tartson velem e csodálatos utazáson. Útravalóként nem árt, ha egy kis matematikai tudást is csomagolunk, szükség lehet rá...

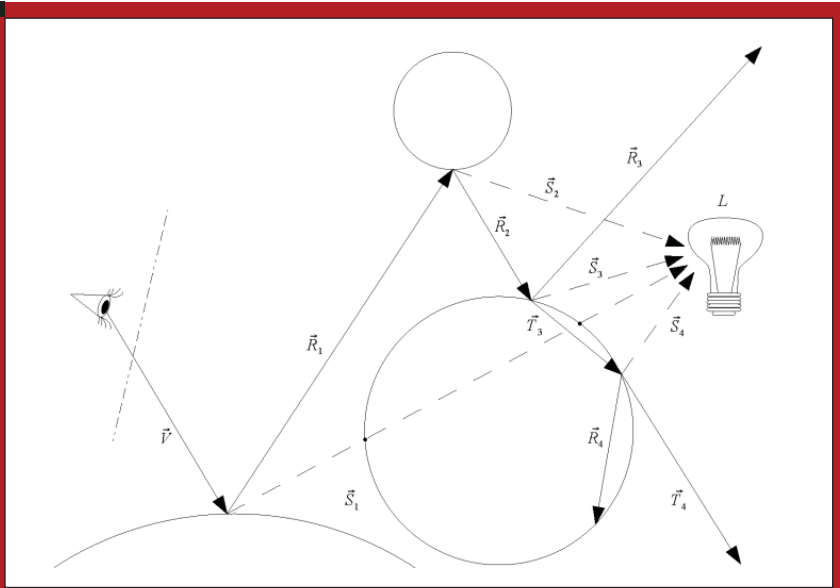
Rekurzív sugárkövetés

Az előző cikkben ismertetett *sugárvetés* (*ray casting*) a térben elhelyezett objektumok látható felületeinek meghatározására hivatott, azaz a szemből

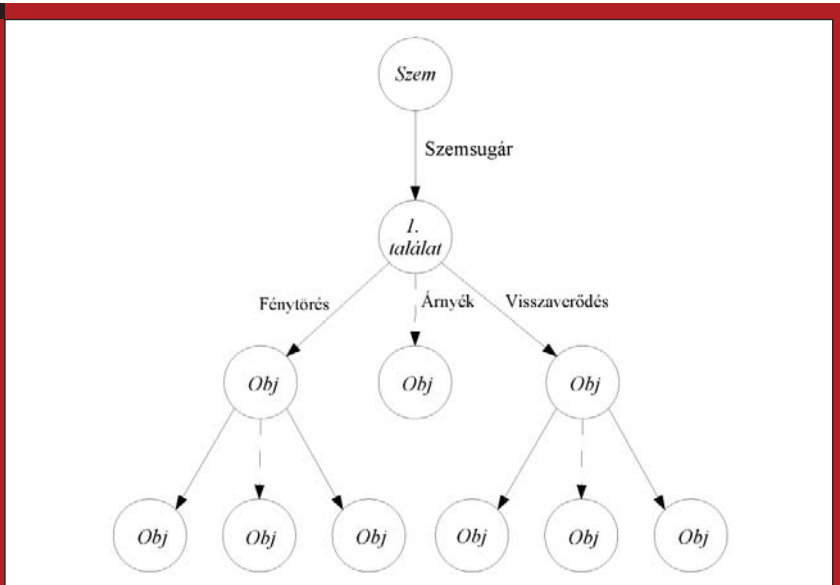
kilőtt sugárhoz megkeresi azt a felületi pontot, ahová az becsapódik. A *rekurzív sugárkövetés* ennél tovább merészkedik: a fény útját ennél tovább követi, így lehetővé válik olyan jelenségek szintézise, amelyeket más technikákkal bonyolult vagy talán nem is lehetséges előállítani. A szóban forgó jelenségek közül három olyan van, amely minden *sugárkövető* programban megtalálható: tükröződés (*reflexió*), fénytörés (*refrakció*), valamint a felületek által vetett árnyékok.

A most ismertetésre kerülő *rekurzív sugárkövetés* algoritmus *Turner Whitted* 1980-ban publikált munkáján alapszik. Hogy milyen módon követhetjük tovább a sugár útját? A megoldás a *rekurzióban* keresendő. A *rekurzió* egy olyan módszer, melynek során úgy számítunk ki egy végeredményt, hogy ugyanazt a műveletet ismételjük egymás után egy adott *operanduson*,

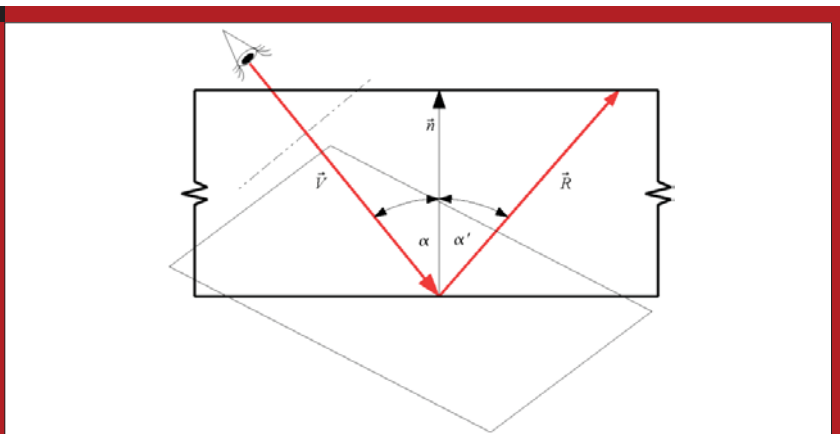
minden művelet egy részeredménnyel zárul, amelyet a sorban következő műveletkor (következő *rekurzió*) felhasznál. *Végtelen rekurzió* az olyan *rekurzió*, amely soha nem ér véget, ez ugye a legtöbb esetben használhatatlan. Nekünk inkább olyan *rekurzió* kellene, amely egy bizonyos számú végrehajtás után kilép (egy jó példa a *végtelen rekurzióra* a Linux világában járatos olvasóknak minden bizonnyal ismerős „*GNU*”, amely a „*GNU is Not Unix*” rövidítése). A *rekurzív sugárkövetés* – a *sugárvetés*-hez hasonlóan – a *szemsugár* kilövésével kezdődik, majd az eltalált felület megállapítása után a folyamat megismétlődik: a kapott beesési pontból indulva újabb sugarakat indítunk útjukra, ám ezúttal a szem helyett a beesési pontból kiindulva a tükröződés, fénytörés, illetve a fényforrások irányába (*reflexiós, refrakciós és árnyéksugarak*)



2. ábra Fényutak a térben (S – árnyéksugarak, R – reflexiós sugarak, T – refrakciós sugarak)



3. ábra A sugárfa



4. ábra Ideális visszaverődés (reflexió)

azért, hogy megtudjuk mely felületek tükröződnek, mely felületek takarják a fényforrásokat (mennyi közvetlen, azaz *direkt megvilágítás* érkezik a fényforrásokból), illetve mely felületet látjuk egy átlátszó objektumon keresztül. Ezeket a fénysugarakat *másodlagos sugaraknak* (*secondary rays*), míg a szemből indított sugarakat *szemsugaraknak* (*eye rays*) vagy *elsődleges sugaraknak* (*primary rays*) nevezzük.

Ha a *másodlagos sugarak* által eltalált felület is tükröző vagy áttetsző, újabb *másodlagos sugarakat* bocsátunk útjukra, azaz minden tört és visszavert sugár elindíthat újabb tört, visszavert és *árnyéksugarakat*. Az ismétlődő *rekurziókkal* (*másodlagos sugarak* kilövésével) egy ún. *sugárfa* (*ray tree*) épül fel, ahol a *szemsugár* a fa gyökere.

A *rekurziót* addig folytatjuk, amíg az aktuális rekurzió a kép aktuális pixeléhez megfelelő mértékben járul hozzá, egyébként a *fényút* követését nincs értelme tovább folytatni.

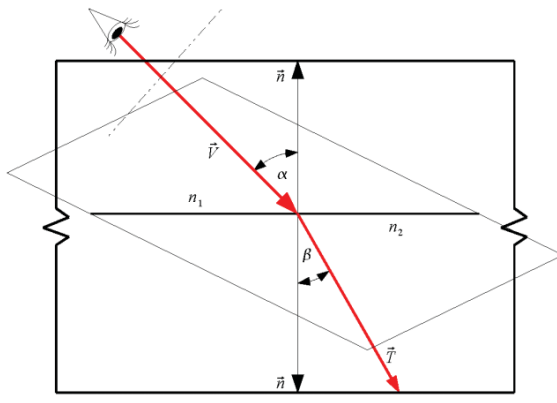
Ezt legegyszerűbb módon úgy oldhatjuk meg, hogy egy konstanssal definiáljuk a *sugárfa* maximális mélységét, azaz ha a rekurziós szám elérte ezt az értéket, az aktuális eredményt tekintjük végleges eredménynek.

Az előző részben ismertetett *sugárvetés* algoritmus alapjait felhasználva játszani könnyedséggel készíthetünk *rekurzív sugárkövetőt*, hiszen a *rekurzív ray tracing* működését tekintve nem más, mint *rekurzív sugárvetés*. A következő három fejezetben a már említett jelenségekkel fogunk kicsit részletesebben foglalkozni.

A tükröződés (reflexió)

Ha a fény valamely tükröző felület határára érkezik, akkor a beérkező fény egy része onnan visszaverődik, másik része pedig behatol a közegbe és ott elnyelődik azaz hővé alakul. Sima felületeknél a visszaverődés tükrös, durva felületeknél szórt, azaz diffúz jellegű. Mi a tökéletesen sima felületek ideális tükrözésével foglalkozunk, a lehetséges felületi durvaságot elhanyagoljuk, illetve a *beesési szögtől* függő tükrözött fényintenzitással nem foglalkozunk.

Először tisztázzunk néhány alapfogalmat. A 4. ábrán látható *beeső fénysugár* becsapódási pontjába állított *felületi normálist* *beesési merőlegesnek*, a *beeső*



5. ábra Ideális fénytörés (refrakció)

és a visszavert fénysugárnak a beesési merőlegessel bezárt szögét *beesési* illetve *visszaverődési szögnek* hívjuk. Ahhoz, hogy a tükrözött fény útját tovább tudjuk követni, szükségünk van a beesési pontból indított, tükrözött fénysugár irányvektorára, melynek kiszámításához az ideális fényvisszaverődés törvényét hívjuk segítségül, mely szerint a beeső fénysugár, a beesési merőleges és a visszavert fénysugár egy síkban vannak, valamint a beesési szög megegyezik a visszaverődési szöggel. Az utóbbi szabály fordítva is érvényes, tehát a fénysugarak visszirányú követésénél is teljesül.

Ennek ismeretében a tükrözött sugár R irányvektora a következőképpen számítható:

$$\vec{R} = \vec{V} - 2 \cdot (\vec{n} \cdot (\vec{n} \cdot \vec{V}))$$

Fénytörés (refrakció)

Ha a fény két optikailag átlátszó közeg határára ér, akkor vagy visszaverődik, vagy belép az új közegbe. A fény terjedési útja mindkét esetben (általában) megváltozik, de a fénytörés jelenségéről akkor beszélünk, ha a fény be is lép az új közegbe és a megtört irányban folytatja útját. Ennek az irányváltozásnak az az oka, hogy a fény különböző sebességgel terjed a két különböző anyagban. A tört fény irányának meghatározásához segítségül hívjuk a *Snellius-Descartes* törvényt, azaz az *ideális fénytörés törvényét*, mely kimondja, hogy a *beeső fénysugár*, a *megtört fénysugár* és a *beesési merőleges* egy síkban vannak, illetve a *beesési szög szinusz*a egyenesen arányos a *törési szög*

szinuszával, az arányossági tényező pedig a második közeg elsőre vonatkoztatott relatív törésmutatója.

$$n_{2,1} = \frac{c_1}{c_2} = \frac{\sin \alpha}{\sin \beta} = \frac{n_2}{n_1}$$

A különböző anyagokat a sugárkövetés során az ún. *abszolút törésmutatóval* (*index of refraction*) jellemezzük, amely a fény vákuumbeli és az adott anyagban mért terjedési sebességének a hányadosa (másképp megfogalmazva az anyag vákuumhoz viszonyított *relatív törésmutatója*), tehát 1-nél mindig nagyobb szám.

Egy átlátszó anyag abszolút törésmutatóját felhasználva a fény vákuumból anyagba lépésekor fennáll a következő összefüggés

$$\sin \alpha = n \cdot \sin \beta$$

Könnyen belátható, hogy a fenti egyenletnek nem lesz megoldása, ha a beesési szög meghalad egy határértéket, nevezetesen a *teljes visszaverődés határszögét*.

$$\Theta = \arcsin \frac{n_1}{n_2}$$

Ha a beesési szög meghaladja a teljes visszaverődés szögét, a fény nem tud behatolni az anyagba, így a beesési pontban a fent ismertetett reflexió jelensége jön létre, a felület tükrözőként viselkedik.

A fénytörés esetében ki kell számítanunk a tört fény irányát, hisz a sugarat ebben az irányban kell tovább követnünk, ennek kiszámítása a következő formulák alapján történik:

$$\cos \alpha = \vec{V} \cdot \vec{n} \quad \cos \beta = \sqrt{1 - \left(\frac{n_1}{n_2}\right)^2 \cdot (1 - (\cos \alpha)^2)} \quad \vec{T} = \left(\frac{n_1}{n_2}\right) \cdot \vec{V} + \left(\cos \beta - \frac{n_1}{n_2} \cdot \cos \alpha\right) \cdot \vec{n}$$

A fénytörési mutató a valóságban a fény hullámhosszától is függ (gondoljunk csak a prizmajelenségre), ettől mi eltekintünk, azt minden hullámhosszon konstansként értelmezzük.

A sugárkövetés során figyelniük kell arra hogy a fény a becsapódási pontban belép az objektumba vagy éppen kilépni készül onnan. Az objektumból kilépő fény számításakor a felületből kifelé mutató normál vektor inverzét kell használnunk (azaz a felületi *normális* befelé mutat).

Néhány anyag törésmutatója a teljesség igénye nélkül: vákuum=1, levegő=1.000292, műanyag=1.11, üveg=1.22, víz=1.33, alkohol=1.362, gyémánt=2.417.

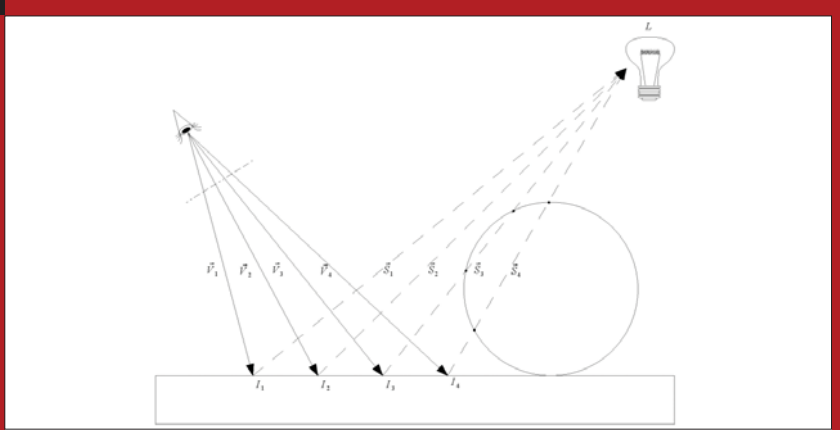
Árnyékok

Egy naív sugárkövető egy adott felületi pont fényességét a pontban vett felületi normális, a fényforrás illetve a kamera elhelyezkedését egy illuminációs modellben felhasználva számítja ki. Arról azonban elfeledkeznek, hogy egy objektum el is takarhatja a fényforrást előle, azaz a pont árnyékba kerülhet.

Képzeljünk magunkat egy felület valamely pontjába, s tekintsünk a fényforrás felé. Ha nem látjuk a fényforrást, biztosak lehetünk benne, hogy árnyékban vagyunk, mégpedig egy olyan felület által, mely a fényforrás és a szemünk között helyezkedik el. Ahhoz, hogy meg tudjuk mondani, hogy a felületi pontból látjuk-e a fényforrást, egy másodlagos sugarat lövünk ki, amely a *beesési pontból* indulva a fényforrás felé mutat, majd ezen a sugáron megkeressük a legközelebbi pozitív t paraméterű metszéspontot. Ezután meg kell néznünk, hogy ez a metszéspont az *árnyéksugár* kiindulópontja és a fényforrás között van-e. Ha igen, akkor a vizsgált felületi pont árnyékban van, az árnyéksugár által eltalált objektum eltakarja a fényforrást a felületi pont elől.

Egy árnyéksugár kilövésének célja nem más, mint megállapítani, hogy egy adott pontba mennyi fény érkezik egy fényforrásból.

Áttetsző, fénytörő felületek esetében az eljárás kicsit bonyolódik.



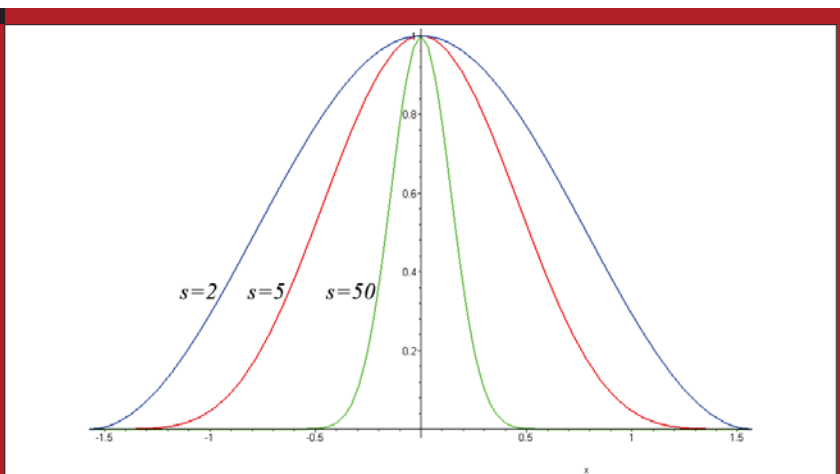
6. ábra Árnyéksugarak kilövése különböző beesési pontokból

Képzeljünk el egy lencsét, amely a fókuszpontjába gyűjti a rajta áthaladó fénysugarakat. Ebben az esetben több sugár egy pontba gyűjtéséről beszélünk, s mivel a *fényutat* visszafelé követjük (*backward ray tracing*), sok-sok árnyéksugárra lenne szükségünk ahhoz, hogy a fókuszpontba gyűjtött fény nagyságát meg tudjuk becsülni (A kilőtt *árnyéksugarak* oroszlánrésze nem a fényforrásban érne véget). Könnyen belátható, hogy az *árnyéksugarakon* értelmezett fénytörés és tükröződés erre a problémára nem nyújt hatékony megoldást. Az ilyen és ehhez hasonló *indirekt illuminációs* jelenségek modellezésére a *globális illuminációs* algoritmusok nyújtanak hatékony megoldást. Az árnyéksugár kilövése folyamán az áttetsző anyagok fénytörését elhanyagoljuk és úgy számolunk velük, mint ha a irányváltoztatás nélkül haladna át rajtuk. Meg kell keresnünk az

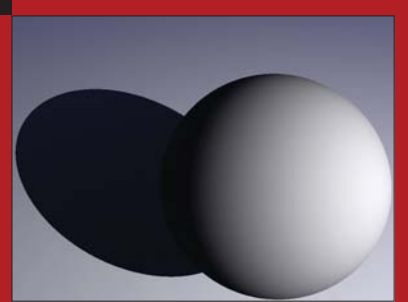
összes olyan felületet, amely az árnyéksugáron a fényforrás és a vizsgált felületi pont között van, s ezen testek átlátszósági együtthatóival meg kell szűrünk a fényforrás által kibocsátott fényt. Az átlátszósági együttható megmutatja, hogy a beérkező fényenergia mely részét nyeli el és mely része hatol be az anyagba.

Az Epsilon távolság

Az *epsilon* távolság használatának oka a *számítási hiba*, mely a számbábrázolás véges pontosságából adódik. Amikor megpróbáljuk megkeresni, hogy egy fénysugár melyik felületbe csapódik be, megkapjuk a fénysugárnak a legközelebbi felülettel vett metszéspontjának pozitív *t* paraméterét. A számítási pontatlanság miatt azonban ezt a *t* paramétert visszahelyettesítve a sugár paraméteres egyenletrendszerébe, jó eséllyel egy olyan pontot kapunk, amely nem pontosan



8. ábra A $\cos(x)$ függvényt különböző hatványokra emelve a Phong által megfigyelt jelenséget figyelhetjük meg

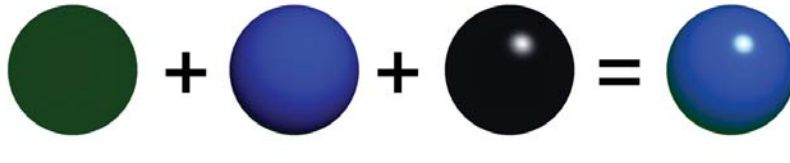


7. ábra Csendélet: diffúz gömb és árnyéka

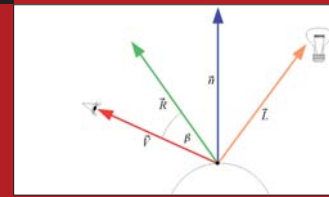
az eltalált felületen nyugszik, hanem a sugár mentén pozitív vagy negatív irányban el van tolva egy minimális távolsággal, ez a távolság a már említett *számítási hiba*. Ha a kapott hibás pontot visszahelyettesítjük az eredeti felület egyenletébe, az az egyenletet nem fogja kielégíteni (mivel nem pontosan a felületen nyugszik). Így tehát nem csoda, hogy ha a kapott hibás pontból indítjuk a további sugarakat és a hibás pont a felületen belülré került, azt tapasztaljuk, hogy a metszéspontkeresés ugyanazt a felületet hozza ki a legközelebbi felületnek, mint amely felszínéről indítottuk a *másodlagos sugarat*. Ennek a problémának a megoldására jól bevált megoldás egy olyan nagyon kicsi *epsilon* távolságot definiálni, amelynél kisebb távolságban levő metszéspontokat figyelmen kívül hagyunk.

Phong modellje

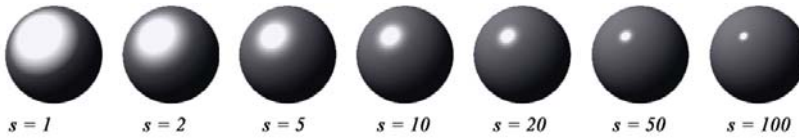
Az előző cikkben tárgyalt *diffúz modellel* nem lehet a fényes, „polírozott” felületű anyagok érzetét visszaadni, az csak durva felületek leegyszerűsített modellezésére alkalmas. A fényes felületekre az jellemző, hogy a beérkező fényenergia oroszlánrészét az ideális tükrörány környezetébe verik vissza, ennek köszönhető, hogy az ilyen felületeken egy ún. *spekuláris fényfolt* jön létre. A róla elnevezett eljárást *Bui Tuong Phong* amerikai kutató fejlesztette ki. *Phong* modellje *empirikus*, azaz pusztán a megfigyelésen alapszik, nem a fény fizikai értelemben vett természetéből származtatott. Megfigyelte, hogy fényes felületeken a *spekuláris fényfolt* nagysága a felületi anyag „polírozottságától”, azaz fényességétől függ. Fényes felületű anyagoknál a fényfolt kicsi és ahogy távolodunk az ideális visszaverődési



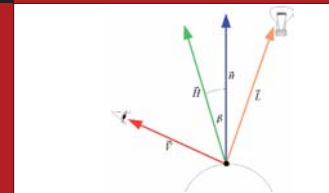
■ 9. ábra A Phong modell tagjai balról jobbra: ambiens tag, diffúz tag és spekuláris tag. Ezek összegéből képződik a jobb oldalon látható kép.



■ 10. ábra A Phong modell vektorkomponensei



■ 11. ábra Phong modellel számított gömbök különböző s fényességi paraméterekkel. Jól megfigyelhető a spekuláris fényfolt átmérőjének változása



■ 12. ábra A Blinn-Phong modell vektorkomponensei

iránytól, egy bizonyos pont után meredeken zuhan az intenzitása. Matt felületeken a *fényfolt* nagy, az intenzitáscsökkenés pedig kisebb mértékű. A megfigyelt viselkedés megvalósításához *Phong* a koszinusz függvényt vette alapul, az eredeti függvény hatványozásával modellezte a *fényfolt*-jelenséget.

Phong modelleje a következő *árnyalási egyenlettel (rendering equation)* írható fel:

$L_p = \text{ambiens tag} + \text{diffúz tag} + \text{spekuláris tag}$ azaz

$$L_p = k_a i_a + \sum (k_d (L \cdot n)_d + k_s (R \cdot V)^s i_s)$$

ahol a $k_a i_a$ az *ambiens tag*, a $k_d (L \cdot n)_d$ a *diffúz tag*, a $k_s (R \cdot V)^s i_s$ pedig a *spekuláris tag*. A diffúz és *spekuláris* tagok számítását és összegzését minden fényforrásra el kell végezni, így kiszámolva a felületi pontban az *eredő illuminációt*.

Az ambiens tag az indirekt megvilágítás hatását hanyag módon próbálja modellezni, intenzitása a felület minden pontjában azonos. A diffúz tagot már jól ismerhetjük az előző cikkből, a *spekuláris* tagról pedig már ejtettünk pár szót, ő a felelős a *fényfoltok* megjelenéséért.

A k tényezők az anyag az adott tagra vonatkozó színét (*ambiens, diffúz és spekuláris színét*) jelzik, az i_a a globális ambiens fényt, az i_d és i_s tényezők pedig a fényforrás megfelelő tulajdonságait, azaz külön definiálható

a fényforrásból érkező *diffúz* és *spekuláris* intenzitás az anyagra nézve. Nézzük meg, hogyan számoljuk ki a *spekuláris* tagot. A formulában szereplő V vektor a becsapódási pontból mutat a szembe, azaz az eredeti beeső sugár inverze. L vektor a becsapódási pontból a fényforrásba mutat, R vektor pedig a fényforrásból érkező fotonok *ideális visszaverődési irányja*. Ahogy a V vektor távolodik az ideális visszaverődési irány környezetétől, úgy nő a béta szög is, azaz az R és V vektorok skaláris szorzata a koszinusz függvény szerint egyre rohamosabban zuhanni kezd, ez pedig pontosan az a jelenség, amelyet *Phong* megfigyelései alapján leírt.

A *spekuláris tagban* található s hatványkitevővel az anyag felületi fényességét (*shininess*) szabályozhatjuk. Minél magasabb ez az érték, annál fényesebb anyag fényfoltját adja a *spekuláris tag*.

A 11. ábrán különböző s értékekkel *renderelt* gömböket láthatunk, ez kitűnően reprezentálja a *spekuláris tag s* paraméterének hatását.

A Blinn-Phong modell

A *Phong* modellhez szorosan kapcsolódik az ún. *Blinn-Phong* modell, mely mindössze annyiban tér el az eredeti *Phong*-modelltől, hogy a *spekuláris tag* esetében a visszavert fény költséges kalkulációja helyett egy olyan H vektorral számolunk, amely a szemvektor és a beérkező fény között

félúton helyezkedik el (*halfway vector*) és a következőképpen számítható:

$$H = V + L$$

Ezt a vektort visszahelyettesítve a *Phong modell R* vektorába az

$$L_b = k_a i_a + \sum (k_d (L \cdot n)_d + k_s (H \cdot n)^s i_s)$$

összefüggéshez jutunk, amely nem más, mint a *Blinn-Phong megvilágítási modell* egyenlete.

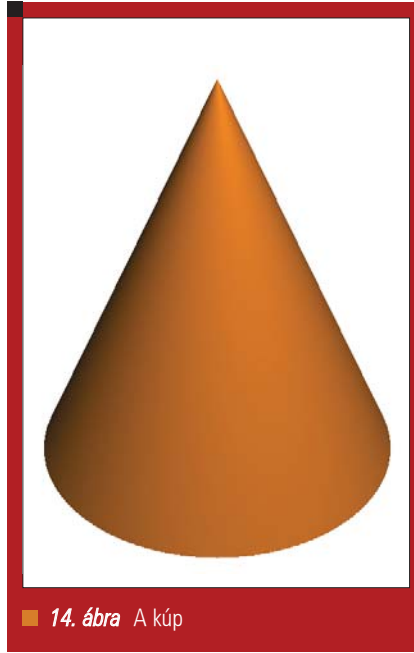
A *Blinn-Phong* modell figyelembe veszi a *spekularitás* változását annak függvényében, hogy milyen szögből nézzük a felületet, így az a *Phong modellel* szemben fizikailag helytállóbbnak (*plauzibilisebbnek*) nevezhető. Az érdekesség kedvéért említem meg, hogy az *OpenGL* is a *Blinn-Phong* árnyalással számolja ki a háromszögek csúcsainak színét, majd ezeket interpolálja a felületen (*Gouraud shading*).

Az egyszerűsített (lokális) árnyalási egyenlet

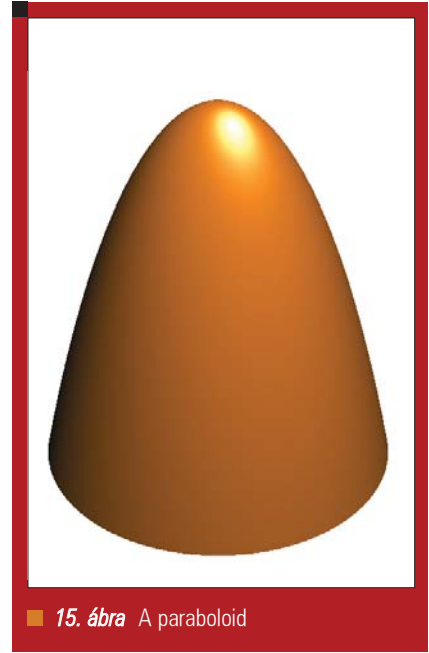
Eljutottunk oda, hogy van sok-sok komponensünk, ezekből össze kellene állítanunk egy általános árnyalási egyenletet, amely tartalmazza az eddig megismert komponenseket és bár elnagyoltan, de leírható vele a felületet adott irányba elhagyó fényenergia. Azért elnagyoltan, mert ha visszaemlékszünk az eddig tanultakra, az indirekt megvilágítás költséges számolásától az egyszerű *rekurzív sugárkövetés* eltelt, azokat teljesen elhanyagolva.



13. ábra A hengerpalást



14. ábra A kúp



15. ábra A paraboloid

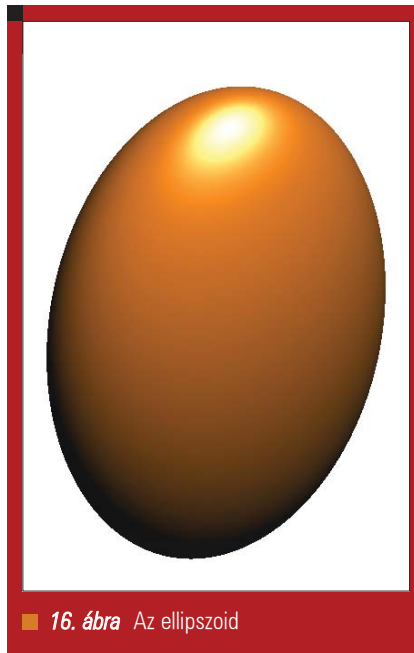
Ha összeírjuk egyetlen formulába az eddig tanultakat, megkapjuk az *árnyalási egyenlet egyszerűsített alakját*, amely általánosan jellemző a lokális illuminációs modellekre:

$$L = L_e + L_a + \sum(L_d + L_s) + k_r \cdot L_r + k_t \cdot L_t$$

Ahol k_r , k_t az anyag fényáteresztő ill. tükröző színe (melyek definiálják, hogy az anyag milyen intenzitással engedi át magán illetve tükrözi a *beeső fényt*); L_d , L_s a fenti modellekből ismert diffúz és *spekuláris* komponensek; L_a az anyag eredő *ambiens* színe ($L_a = i_a k_a$); L_r , L_t a felület adott pontjába a környezetből érkező ideálisan tört és tükrözött fényenergia. Végére hagyjuk az L_e komponens tárgyalását, amely a felület által a környezetbe sugárzott fény intenzitása, amennyiben az objektum fényt *emittál* a környezetbe. Mivel a hagyományos *ray tracing* algoritmus csak pontszerű fényforrásokat tud kezelni, ennek igazából nem lesz látványos eredménye.

Másodfokú felületek

Az előző részben tárgyaltuk a gömböt, mint a legegyszerűbb másodfokú felületet. Most további három másodfokú (*kvadratikus*) felülettel ismerkedünk meg. Itt csak a felületek egyenleteit közöljük, a sugárral vett metszéspontokat a gömbhöz hasonló módon, visszahelyettesítéssel és a t paraméterre való megoldás keresésével lehet elvégezni.



16. ábra Az ellipszoid

Hengerpalást

Az Y tengellyel egybeeső hossztengetyű, r sugarú végtelen hengerpalást felületét azok a pontok alkotják, melyek XZ síkon vett hossztengetytől mért távolsága r .

Így a hengerpalást minden felületi pontjára teljesül a következő másodfokú egyenlet:

$$\sqrt{P_x^2 + P_z^2} - r = 0$$

A végtelenbe emelkedő hengerpalást helyett a megszokott változatokat kaphatjuk, ha a koordinátákat a $Y_{\max} > Y > 0$ egyenlőtlenségekkel

korlátozzuk, azaz ha a metszéspont Y koordinátája Y_{\max} -nál nagyobb, vagy 0-nál kisebb, a kapott metszéspontot figyelmen kívül hagyjuk.

Kúp

Az Y koordinátatengellyel egybeeső hossztengetyű, h magasságú és a $z=0$ síkon vett r sugarú végtelen kúp egyenlete

$$\frac{P_x^2 + P_z^2}{C^2} - (P_y - h)^2 = 0 \text{ ahol } C = \frac{r}{h}$$

A hengerpalásthoz hasonlóan itt is megkaphatjuk a szokásos alakzatot, ha az Y koordinátákat a fenti egyenlőtlenséghez hasonlóan korlátozzuk.

Paraboloid

A parabola felületét azon pontok alkotják, melyek kiegyenlítik a következő *kvadratikus* egyenletet:

$$P_x^2 + P_z^2 - P_y = 0$$

Ellipszoid

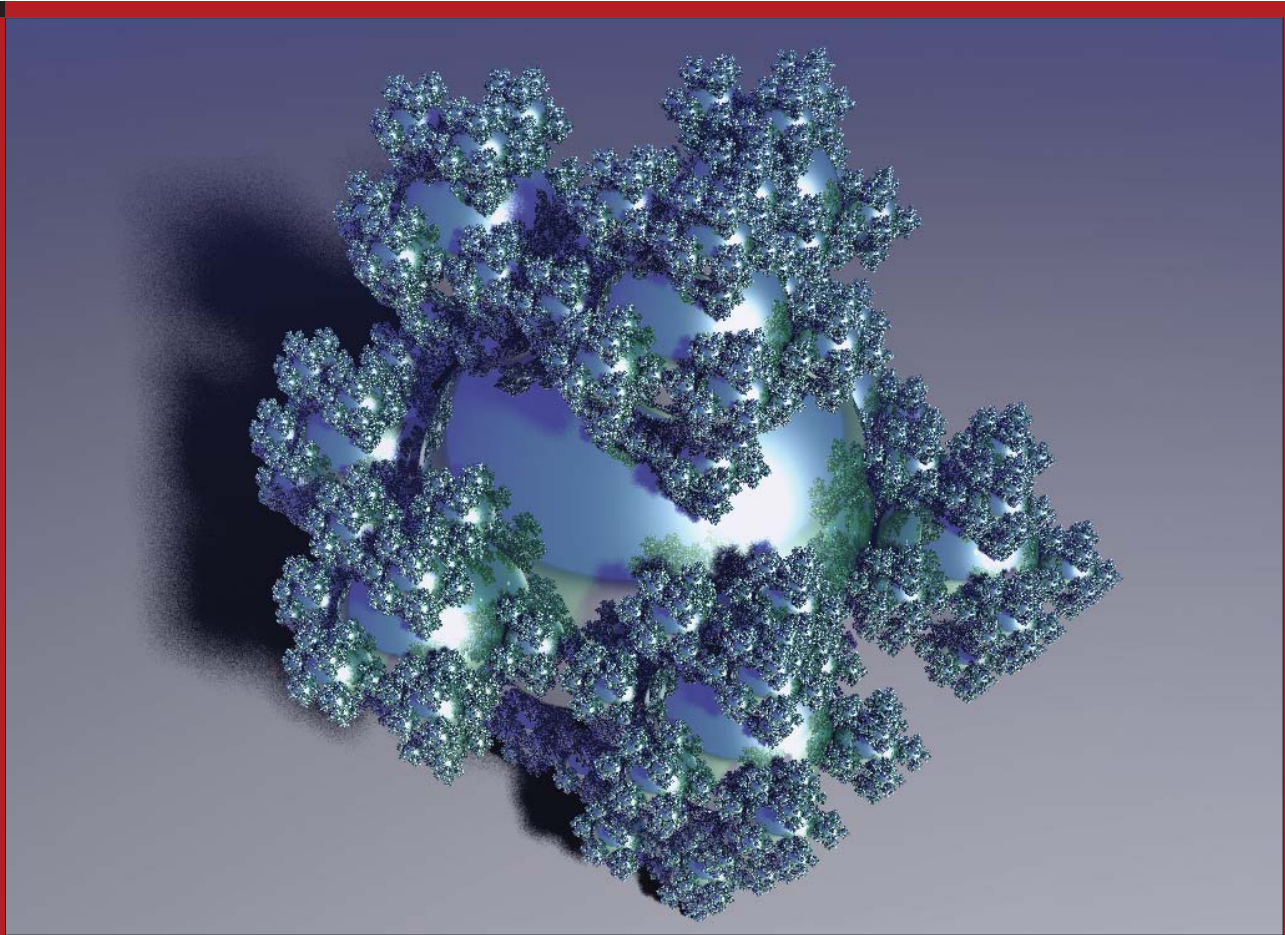
Az ellipszoid egyenlete

$$\frac{P_x^2}{a^2} + \frac{P_y^2}{b^2} + \frac{P_z^2}{c^2} - 1 = 0$$

ahol a , b , c az ellipszoid sugarait jelenti.

Felületi normálvektorok

A felületek P pontban vett normálisának előállításához az implicit felületünk egyenletéből kiszámítjuk



17. ábra A „gömbpely” 6 rekurziós lépés után. Az ábrán látható objektum megközelítőleg 3 millió gömbből áll

a felület P pontbeli *gradiensvektorát*. Egy felület P pontbeli *gradiensvektora* ugyanis mindig merőleges a felületre az adott pontban, tehát a felület P pontbeli *normálvektora*ként értelmezhető. A *gradiensvektor* komponenseit a felület egyenletének X, Y és Z szerinti *parciális deriváltjai* alkotják:

$$\nabla f = \frac{\partial f}{\partial X} i + \frac{\partial f}{\partial Y} j + \frac{\partial f}{\partial Z} k$$

Vegyük például az r sugarú, origó középpontú gömböt, ennek egyenletét $r^2 = X^2 + Y^2 + Z^2$ alakra hozva a következő parciális deriváltakat képezhetjük:

$$\frac{\partial f}{\partial X} = 2 \cdot P_X; \frac{\partial f}{\partial Y} = 2 \cdot P_Y; \frac{\partial f}{\partial Z} = 2 \cdot P_Z$$

Az így kapott *gradiensvektor* a gömb X,Y,Z pontbeli *normálvektora*. Hasonló eljárást alkalmazhatunk a fentebb tárgyalt *implicit felületeknél* is.

Procedurális objektumok

Egy kis fantáziával és kreativitással különböző elemi felületekből építkezve algoritmikusan létrehozhatunk összetettebb objektumokat, ezeket *procedurális objektumoknak* nevezik a szakirodalomban. A 17. ábrán látható „gömbpely” is egy ilyen *komplex, procedurálisan* előállított objektum, mely *rekurzív* módon generált gömbökből van felépítve. Algoritmikus generálása meglehetősen egyszerű, megfelelően mély *rekurzióval* rendkívül részletgazdag és *impresszív* látványt nyújt. A *rekurzió* minden lépésében az előző *rekurzió* által előállított minden egyes gömb körül 12 újabb gömb keletkezik oly módon, hogy azok felülete érinti az előző rekurzióból származó gömb felületét. Az újabb gömbök sugara a szülő (előző *rekurzióból* származó) gömb sugaránál kisebb, én a szülő gömb sugarának harmadolásával számoltam. Mivel a gömbök tárolása a számítógép memóriájában történik, valamint a metraszéptkeresés költsége a gömbök

számával nő, a rekurziós lépések számát viszonylag rövidre kell fognunk. Az előző cikkel kapcsolatban kaptam pár olvasói visszajelzést, köszönöm a pozitív, építő jellegű kritikákat. Az ebbe a cikkbe tervezett témák közül hely hiányában kimaradtakat a következő alkalommal igyekszem majd pótolni, ahol lesz szó *textúrázásról, sugártranszformációkról, a tóruszról* mint negyedfokú felületről, háromszög alapú modellek *rendereléséről*, valamint az elosztott sugárkövetésről (*distributed ray tracing*) is ejtünk majd pár szót a tervek szerint.



Szendi Ákos

(akos.szendi@gomortel.hu)

27 éves, szabadúszó programozóként tevékenykedik. A Miskolci

Egyetem villamosmérnök szakos hallgatója. Kevéske szabadidejében gitározni tanul vagy épp egy jó könyvet tart a kezében.