

# **Diffúz fényvisszaverődések és kiterjedt fényforrások modellezése rekurzív sugárkövetéssel**

**Vass Gergely**

Konzulens: Dr. Szirmay-Kalos László

Budapesti Műszaki és Gazdaságtudományi Egyetem  
Irányítástechnika és Informatika Tanszék

## Tartalom

|                                                 |    |
|-------------------------------------------------|----|
| Tartalom.....                                   | 2  |
| 1 Bevezető .....                                | 3  |
| 2 Háttér .....                                  | 3  |
| 2.1 Képszintézis elterjedt algoritmusai.....    | 3  |
| 2.1.2. Rekurzív sugárkövetés .....              | 7  |
| 2.1.3. Globális illumináció .....               | 8  |
| 2.2 BRDF .....                                  | 9  |
| 3 Sugárkövetés Kiterjesztése .....              | 9  |
| 3.1 Diffúz felületek létrehozása .....          | 9  |
| 3.2 Mikro-, mezo- és makrostruktúra.....        | 10 |
| 3.3 Anyagi jellemzők beállítása .....           | 11 |
| 3.4 Teszt Környezet .....                       | 13 |
| 4 Képszintézis .....                            | 14 |
| 4.1 Gyűjtőséták .....                           | 14 |
| 4.2 Mintavételezés .....                        | 15 |
| 4.3 A zaj csökkentése a gyakorlatban.....       | 17 |
| 5 Összefoglalás és fejlesztési lehetőségek..... | 18 |
| Irodalom.....                                   | 19 |

## 1 Bevezető

Napjainkban mind az ingyenes, mind a megvásárolható 3D grafikai programok képesek az un. rekurzív sugárkövetés megvalósítására. Ez az algoritmus a felületek közötti fényvisszaverődések csupán egyetlen speciális esetét – a tökéletes tükröződést – képes kezelni, így nem alkalmas a szórt fénysugárzás számítására. Dolgozatomban bemutatom, hogy egy ilyen standard 3D grafikai szoftver segítségével hogyan lehet a részben, vagy tökéletesen szórt fényvisszaverődéseket kezelni. A módszer – a megjelenítő algoritmusok nagy részével ellentétben – a kiterjedt fényforrások kezelésére is alkalmassá teszi a 3D trükksoftvert. Különös figyelmet fordítottam arra, hogy a gyakorlatban előforduló lehetséges nehézségeket feltárjam, és – ha lehet – megoldási lehetőséget javasoljak. Ennek köszönhetően a létező összes gépgeneráló algoritmussal és egy egyszerű képfeldolgozó program segítségével kipróbálható az általam javasolt módszer.

## 2 Háttér

A számítástechnika rohamos fejlődésének hatására a számítógépes grafika mára már eljutott a szinte tökéletes fotórealizmusig [6]. Léteznek algoritmusok, melyek a fizika törvényeinek megfelelően szimulálják a fény viselkedését, így – meglehetősen hosszú idő alatt – olyan képeket generálnak, amelyek nehezen különböztethetők meg egy fényképtől. Az ilyen programoknak napjainkban még nem léteznek elterjedt ingyenes változatai, és a piaci forgalomban lévő szoftverek is csupán a modellezésre használt 3d programok csekély hányadát támogatják.

A gyakorlatban használt programok ezzel ellentétben súlyos kompromisszumokat kötnek az “épeszű” számítási idő elérésére. A következőkben áttekintjük, hogy ezek az elterjedt algoritmusok mire képesek és mire nem, valamint azt, hogy milyen trükkel tudjuk a programok képességeit növelni.

### 2.1 Képszintézis elterjedt algoritmusai

A rasztergrafikai rendszerekben a képszintézis elemi lépése az egyes pixelek színének meghatározása. Ehhez meg kell tudnunk mondani, hogy egy meghatározott irányból mennyi fény jut a kamerába. Ez egy borzasztóan nehéz feladat, ha minden lehetséges fizikai törvényt figyelembe akarunk venni. Ha eltekintünk a fény hullámoptikai tulajdonságaitól és csak a felületek megjelenítésére korlátozzuk vizsgálatainkat, azaz nem kezeljük a tömör testeket, gőzöket, gázokat stb., akkor ez a feladat a *képszintézis egyenletének* megoldását jelenti [1].

A képszintézis egyenlete:

$$L(\vec{x}, \omega) = L^e(\vec{x}, \omega) + \int_{\Omega} L(h(\vec{x}, -\omega'), \omega') \cdot f_r(\omega', \vec{x}, \omega) \cdot \cos \theta' d\omega'$$

ahol

$L(\vec{x}, \omega)$ : felületi sugársűrűség (radiancia)

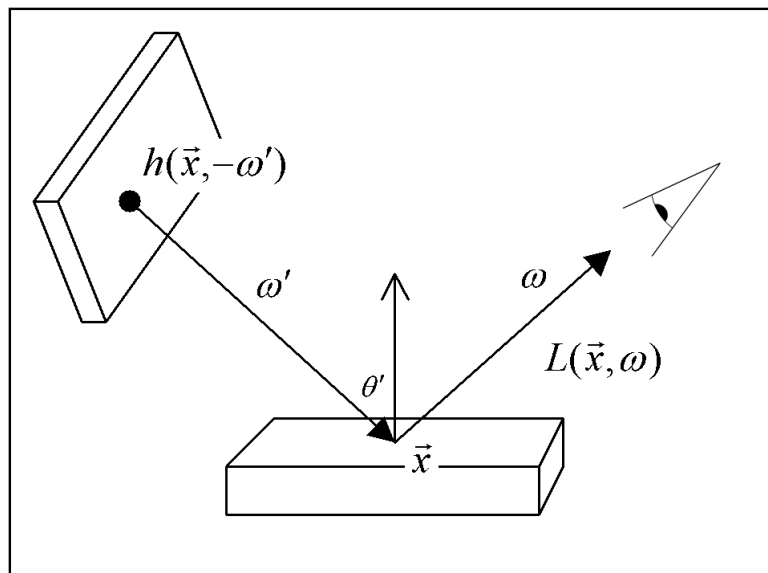
$\vec{x}$ : az éppen árnyalt pont helyvektora,

$\omega$ : az árnyalt pontból a kamera felé néző irányvektor,

$\omega'$ : az összes lehetséges beesési vektor egyike,

$h(\vec{x}, -\omega')$ :  $\vec{x}$ -ből  $-\omega'$  irányban látható pont,

$\theta'$ : a beesési vektor és a normál vektor által beárt szög.



*A képszintézis egyenletének geometriája*

Az egyenlet segítségével azt szeretnénk kiszámolni, hogy adott helyvektorral megadott pontból bizonyos szögbe – nyilván a kamera felé – mekkora fény mennyiség érkezik.  $L(\vec{x}, \omega)$  az 'x' helyvektorral jelzett pont  $\omega$  irányú radianciáját jelöli. Ezt úgy kapjuk meg, hogy a felület saját emissziójához – ezt  $L^e(\vec{x}, \omega)$  jelöli – hozzáadjuk a különböző irányokból érkező fényenergia kamera felé visszaverődő hányadát. Az utóbbi tag kiszámításához integrálni kell a térszögekre három tag szorzatát: az illető irányból érkező fény mennyiségét, az ún. BRDF függvényt ( $f_r(\omega', \vec{x}, \omega)$ ), mely az anyagi jellemzőknek megfelelően megmondja, hogy bizonyos irányból beérkező foton mennyi eséllyel lép ki a kilépési irányba, illetve a beesési szög koszinuszát. A beérkező fény kiszámításához a *láthatósági függvényt* használjuk, ( $h(\vec{x}, \omega)$ ) ami megadja, hogy adott pontból, adott irányba, mely másik felületi pont látszik.

Vezessük be a fény-felület kölcsönhatást leíró  $\tau$  integráloperátort:

$$(\tau L)(\vec{x}, \omega) = \int_{\Omega} L(h(\vec{x}, -\omega'), \omega') \cdot f_r(\omega', \vec{x}, \omega) \cdot \cos \theta' d\omega'$$

Így felírhatjuk a képszintézis egyenletét jóval tömörebb formában:

$$L = L^e + \tau L$$

Ez egy Fredholm féle másodfajú integrálegyenlet, amelyben az ismeretlen a sugársűrűség függvény. Láthatóan az ismeretlen mind az integráloperátoron kívül, mind pedig azon belül megjelenik, így a két oldal között csatolás van. Az egyenlet megoldása ezen csatolás feloldását jelenti, amely azért okoz nehézséget, mert az integráloperátor invertálható. A megoldást szerencsére egy egyszerű trükkel megkaphatjuk. A jobb oldali ismeretlen függvény helyére helyettesítsük be a teljes jobb oldalt rekurzívan, amely az integrálegyenletet egy végtelen sorra alakítja át:

$$L = L^e + \tau L^e + \tau^2 L^e + \tau^3 L^e + \dots$$

A sor tagjai a közvetlen kisugárzást, majd az egyszeres, kétszeres, háromszoros stb. visszaverődéseket jelölik. A tagok egyre növekvő dimenziójú integrálok. Még akkor is nagyon nehezen kiértékelhető a fenti egyenlet, ha csupán az első két tagot vesszük számításba és elhanyagoljuk a sor további részét. A következőkben áttekintjük, hogy hogyan egyszerűsítik – még ennél is jobban – a különböző elterjedt képgeneráló algoritmusok a fenti egyenletet a számítás megkönnyítése érdekében.

### 2.1.1. Lokális illumináció

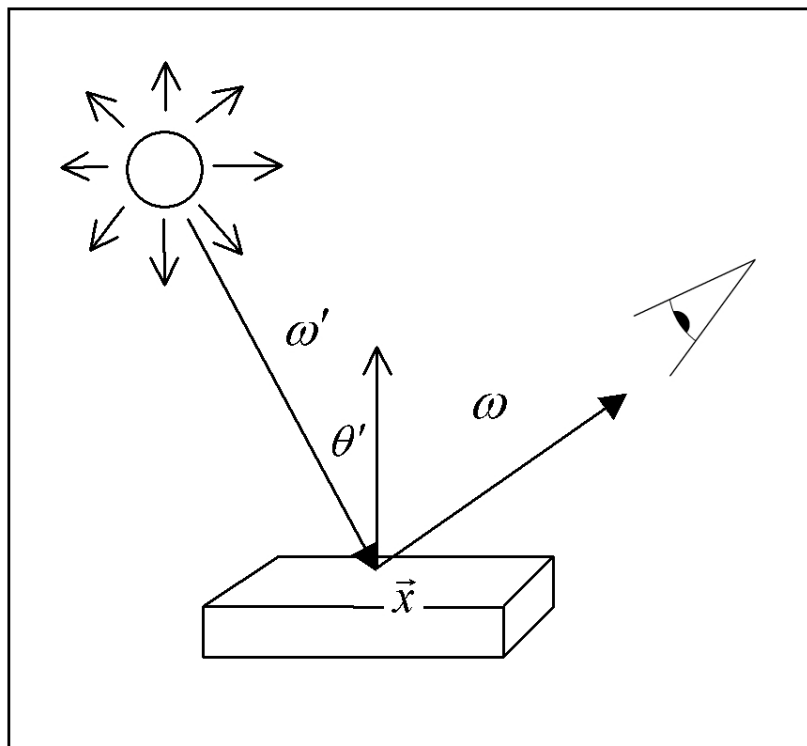
A napjainkban legelterjedtebb, illetve ingyen beszerezhető 3D programok az ún. *lokális illuminációs modellel* dolgoznak, ami a fenti egyenlet súlyos egyszerűsítését jelenti. A végtelen tagú sornak csupán az első két tagjával foglalkozik, így a feladat egy integrál megoldására egyszerűsödik. Természetesen ez is nehéz feladat, így további egyszerűsítésekkel élünk. Az összes térszögre úgy tudunk könnyen integrálni, ha az integrál mögött álló függvények legalább egyikét dirac-impulzusokból állítjuk össze.

Amennyiben a bejövő fényenergiát írjuk le dirac impulzusokkal, akkor csak véges számú térszögből számítható bejövő fényre. Ez a gyakorlatban azt jelenti, hogy csak *véges számú, pontszerű fényforrást* vesszük figyelembe a felületek árnyalásánál. Ezt az elhanyagolást nyilván nem tehetnénk meg, ha az így generált kép túl irreális lenne. A valóságban azonban – az esetek többségében – a felületek színét alapvetően az elsődleges fények determinálják. Amennyiben léteznek hangsúlyos másodlagos fényforrások, akkor azokat modellezhetjük absztrakt, „láthatatlan” elsődleges fényekkel. A lokális illumináció tartalmaz egy irányfüggetlen ambiens komponenst is, mely az elhanyagolt felületek között verődő fényt hivatott pótolni. Így sikerült – meglehetősen durván – elhanyagolni a visszacsatolást – azaz a felületek közötti fényvisszaverődést – valamint a kiterjedt fényforrásokat.

A fenti probléma így az alábbi egyenlet megoldására vezethető vissza:

$$L(\vec{x}, \omega) = L^e(\vec{x}, \omega) + \sum_{i=1}^{\text{fények száma}} LIGHT_i(\vec{x}, \omega') \cdot f_r(\omega', \vec{x}, \omega) \cdot \cos\theta' + AMBIENT$$

ahol a  $LIGHT_i(\vec{x}, \omega')$  függvény megadja, hogy az  $i$ -edik absztrakt fényforrás a megadott irányban lévő  $\vec{x}$  pontba mennyi fényt sugároz. Az  $\omega$  irányt a kamera és az árnyalt pixel,  $\omega'$  irányt az árnyalt pont és az aktuális fényforrás relatív helyzete határozza meg. Míg az eredeti egyenlet – a visszacsatolás miatt – magas dimenziójú integrálokat tartalmazó végtelen sor kiértékelését jelenti, addig a lokális illumináció véges, fix lépésben megadja a pont fényességét.



*A lokális illuminációs egyenlet geometriája*

## 2.1.2. Rekurzív sugárkövetés

A szintén elterjedt *sugárkövetéses módszer* a lokális illuminációs render-programok gyakori kiegészítése. Ezzel a technikával a tökéletes tükröződések és az átlátszó felületek fénytörését tudjuk megjeleníteni. A tükrözés megjelenítése rettentően [9] fontos a csillogó vagy tükröző anyagok esetében, hiszen az ilyen felületek – ezüst, üveg vagy krómozott tárgyak – kinézetét szinte kizárólag a tükröződés határozza meg.

A módszer a rendelkezésre álló lokális illuminációs algoritmust használja: a nézeti irányvektor és a felület normálvektor alapján kiszámított tükör- ill. törési irányba meghívjuk az árnyaló algoritmust. Ha az így útjára bocsátott sugár ismét egy törő vagy tükröző felülettel találkozik, akkor egy új irányban indítunk újabb sugarat. Így a módszer rekurzív alkalmazásával képesek vagyunk a többszörös tükröződések és törések megjelenítésére:

$$L(\vec{x}, \omega) = L^e(\vec{x}, \omega) + \sum_{i=1}^{\text{fények száma}} \text{LIGHT}_i(\vec{x}, \omega') \cdot f_r(\omega', \vec{x}, \omega) \cdot \cos \theta' + \\ + L(h(\vec{x}, \omega^{\text{tükör}}), -\omega^{\text{tükör}}) \cdot RL + L(h(\vec{x}, \omega^{\text{törés}}), -\omega^{\text{törés}}) \cdot RR + \text{AMBIENT}$$

A két új tag a tükrőrirányból ill. a törésirányból érkező fényt veszi figyelembe RL ill. RR súlyozó tényezővel. Ezek nem feltétlenül konstans értékek, hiszen a valóságban irányfüggő az anyagok visszaverési és törési tulajdonsága. Látható, hogy ez a probléma is tartalmaz visszacsatolást, azaz a tökéletes megoldáshoz akár végtelen sok tagot ki kell értékelni. Mivel RL és RR összege az energia megmaradás miatt mindig kisebb mint 1, ezért a tükrő- és törésirányból érkező tag kiszámításánál elég csupán néhány lépésig elvégezni a számítást, hiszen egyre kisebb tagokat kapunk.

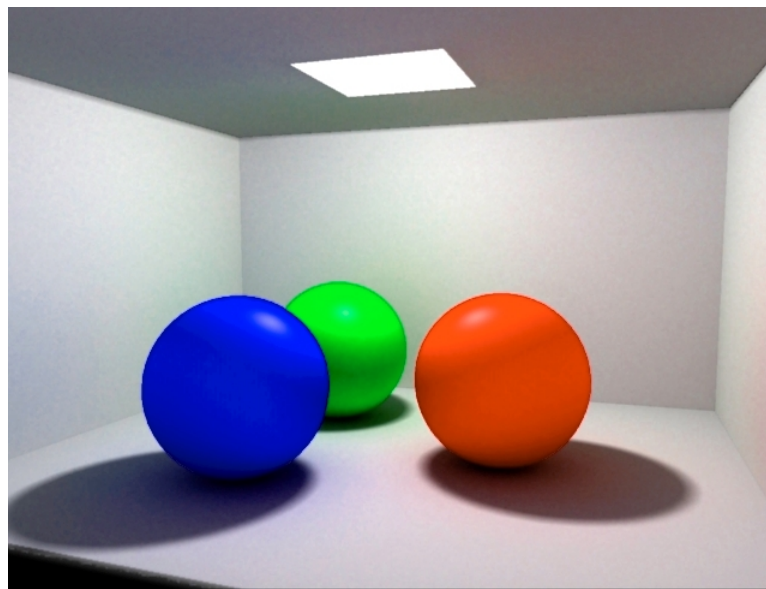
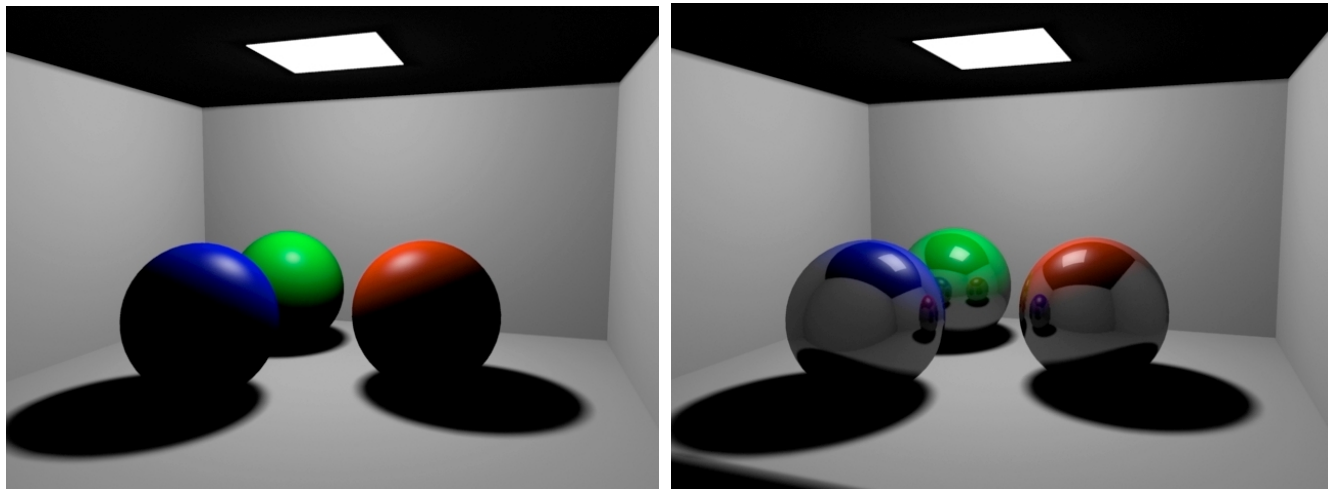
Tekintsük most a rekurzív sugárkövetés egyenletét a lokális illumináció nélkül:

$$L(\vec{x}, \omega) = L^e(\vec{x}, \omega) + L(h(\vec{x}, \omega^{\text{tükör}}), -\omega^{\text{tükör}}) \cdot RL + L(h(\vec{x}, \omega^{\text{törés}}), -\omega^{\text{törés}}) \cdot RR$$

Ez gyakorlatilag az árnyalási egyenlet, azzal az apró egyszerűsítéssel, hogy a BRDF-et – nem a beérkező fényt (!)– Dirac impulzussal írjuk le. Vegyük észre, hogy a fenti egyenlet – a lokális illuminációval ellentétben – az integrálegyenlet végtelen sorának minden tagját figyelembe veszi. Ezt fogjuk kihasználni módszerünkben.

### 2.1.3. Globális illumináció

Léteznek módszerek – pl. Monte Carlo integrálás, photon mapping – melyek az árnyalási egyenletet az eredeti formájában próbálják megoldani, így nem csupán a felületek közötti tökéletes, hanem a szórt visszaverődést is kezelik. Ezeket az algoritmusokat – melyek a másodlagos fényforrások hatását is realiztikusan reprodukálják – nevezik globális illuminációs módszereknek. A bevezetőben említett okok miatt ezek csak nehezen elérhetőek, így célunk a lokális illumináció és a sugárkövetés felhasználásával az árnyalási egyenlet lehető legjobb megoldása.



*Lokális illumináció, sugárkövetés és globális illuminációval előállított képek.*



## 2.2 BRDF

A képszintézis egyenletének megoldásához meg kell adnunk minden felülethez annak BRDF függvényét. A BRDF meghatározására több módszer létezik, tekintsük a legismertebbeket [6]:

*Direkt méréssel* fizikailag korrekt eredményt kaphatunk, de viszonylag nagy háttértárra van szükség a BRDF függvény „táblázatos” tárolásához. Ezeket a méréseket erre a célra kialakított berendezésekkel, ún. gonio-reflektométerrel végzik, melyek képesek különböző beesési szögek mellett a visszavert fény spektrumát érzékelni. Bár a végeredmény szempontjából a valós BRDF eltárolása és felhasználása lenne a legjobb megoldás, de a nagy tárigényen kívül problémát okoz a mérés gyors és egyszerű elvégzése is, hiszen a BRDF függvény egy 5 dimenziós függvény.

A *microfacet modell* az anyagok mikroszkopikus felületi egyenetlenségeit ragadja meg azáltal, hogy a felületet apró lapocskák tömkelegének tekinti [7] [8]. Ezen lapocskák – microfacet-ek – apró tükrökként szolgálnak, és csak tükrőirányban verik vissza a fényt. A BRDF meghatározásához ki kell számolni, hogy a lapocskák állásának eloszlása alapján mennyi fény verődik a kamerába. Ezek a modellek hatékonyan képesek modellezni a különböző felületek visszaverési tulajdonságait és nem igényelnek nagy tárkapacitást. Mivel a különböző anyagok microfacet modelljét nem lehet könnyen – gombnyomásra – előállítani, ezért ez produkciós környezetben nem terjedt el.

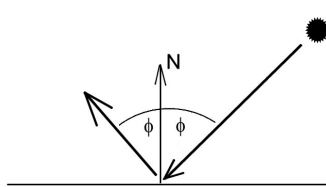
A gyakorlatban használt programok az ún. *empirikus modelleket* használják [1]. Ezek a reprezentációk a BRDF függvényt különböző matematikai képletekkel próbálják közelíteni, melyeknek gyakran nem igazodnak a természet törvényeihez, fizikailag nem plauzibilis. Ezek a modellek intuitíve könnyen parametrizálhatóak és a hétköznapi szemlélőt nem zavarja ha az eredmény fizikailag nem korrekt. Az elterjedt 3d programok általában ezt a modellt használják.

## 3 Sugárkövetés Kiterjesztése

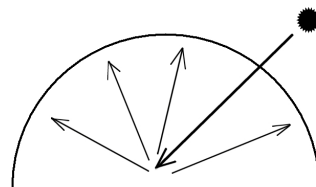
A lokális illuminációs modell csak a pontszerű fényforrásokat veszi figyelembe és a többszörös fényvisszaverődéseket sem kezeli, így nem tudjuk használni a fényszóródás, a derítés szimulálásához. A rekurzív sugárkövetés ellenben képes a felületek között történő fényátadást kezelni, bár annak nyilván csupán egy speciális esetét. Célunk, hogy a sugárkövetés felhasználásával szórt fényvisszaverést is képesek legyünk kezelni, felhasználva a fent leírt microfacet modell ötletét

### 3.1 Diffúz felületek létrehozása

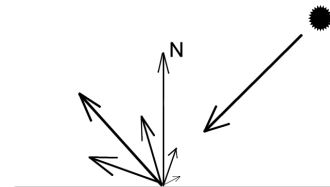
Némi cselt kell alkalmaznunk, hogy a diffúz anyagokat is tudjuk modellezni a tökéletes tükröződés felhasználásával. A diffúz modell a „durva” felületeket írja le, melyek a beérkező fényt a féltér minden irányába többé kevésbé egyenletesen szórják szét.



*Tökéletes tükröződés*



*Diffúz fényvisszaverés*



*Fényes anyagok tükröződése*

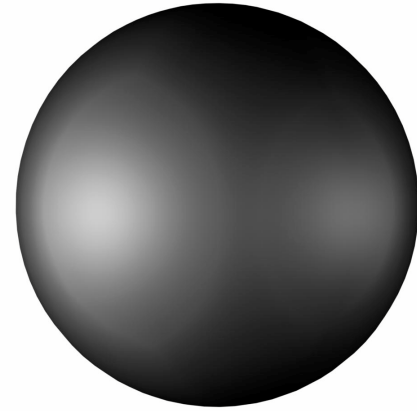
A szórt fényvisszaverés elérésére a microfacet modell gondolatát használhatjuk fel, hiszen ott apró, tökéletesen tükröző lapocskákkal szimuláljuk a felület fényvisszaverő tulajdonságait. A felületet tehát megpróbáljuk úgy kialakítani, hogy az apró domborulataival a diffúz anyag hatását keltse. Tekintsük át, hogy a 3d programokban milyen lehetőségeink vannak a felület domborzatának kialakítására.

### 3.2 Mikro-, mezo- és makrostruktúra

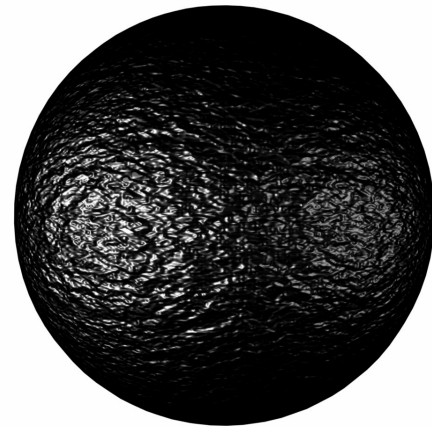
*Mikrostruktúra:* a lokális illuminációs modell használata esetén minden felülethez megadunk egy empirikus BRDF függvényt (Phong, Lambert stb.). Ez a függvény hivatott modellezni az atomi, mikroszkopikus, illetve a már nem látható apró felületi egyenetlenségeket. Mivel a sugárkövetés ezt nem veszi figyelembe, számunkra nem érdekes.

*Makrostruktúra:* a felületi struktúrát, anyag-szerkezetet – valamilyen procedurális algoritmus segítségével – akár létre is hozhatjuk virtuális modell formájában. Használhatunk polygon, subdivision vagy NURBS felületeket erre a célra. Ha többé kevésbé realiztikus eredményt szeretnénk, akkor nagyon nagy részletességű felületeket kell létrehozni. Ez pedig kezelhetetlenül nagy tárhelyet és memóriát igényelne.

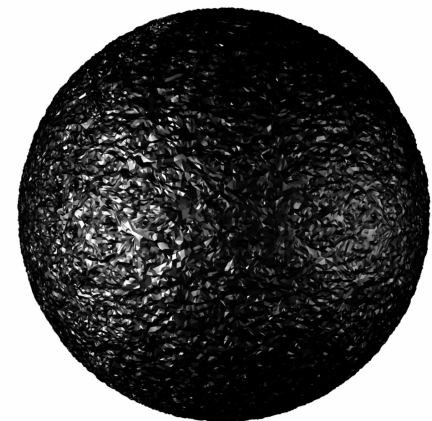
*Mezostruktúra:* A 3D programok az anyagok szerkesztésénél lehetőséget kínálnak ún. textúrák alkalmazására. Ezek segítségével a klasszikus illuminációs modellek paramétereit módosíthatjuk térben és időben. Számunkra az ún. bump map – bucka leképezés – érdekes, ez a textúra ugyanis a felület egy pontjának árnyalása előtt a normál vektort tetszőleges irányba forgatja. Mivel ezt a sugárkövetésnél figyelembe vesszük, alkalmas arra, hogy a felület apró egyenetlenségeit, domborulatait modellezzük vele [5]. Mivel a domborzat egy – akár procedurális – textúra alapján jön létre, ez a módszer nem igényel különösebb tárhelykapacitást.



*mikrostruktúra*



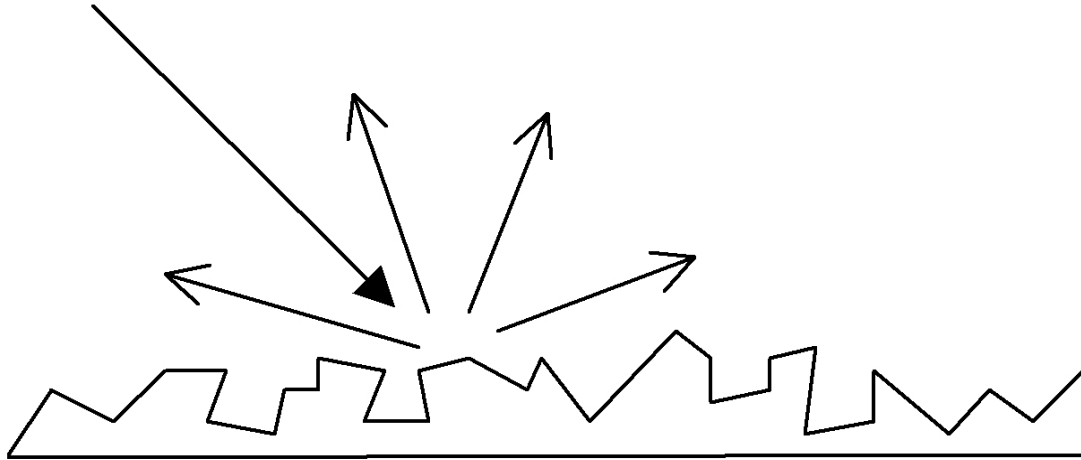
*mezostruktúra*



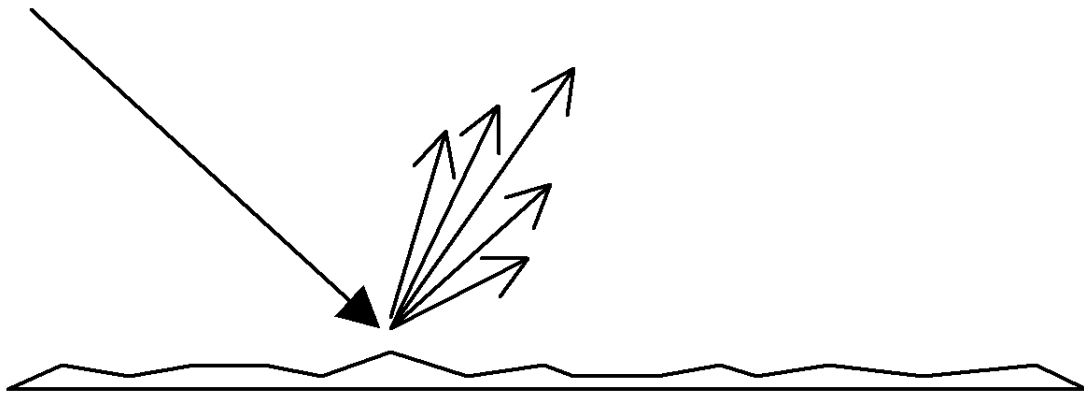
*makrostruktúra*

### 3.3 Anyagi jellemzők beállítása

A microfacet modellhez hasonlóan itt a bump map domborzata fogja determinálni az anyag BRDF függvényét, vagyis azt, hogy a beérkező fény hogyan verődik vissza. Célunk a diffúz viselkedés elérése, így olyan sűrű domborzatot kell létrehozunk a felületen, hogy az a tér minden irányába szétszórja a fényt.



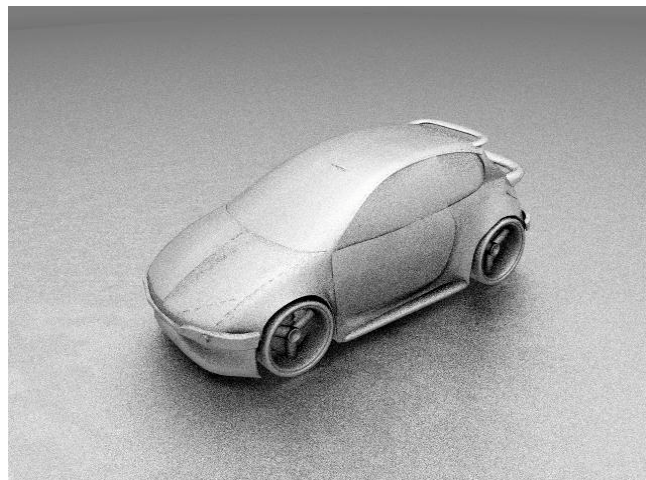
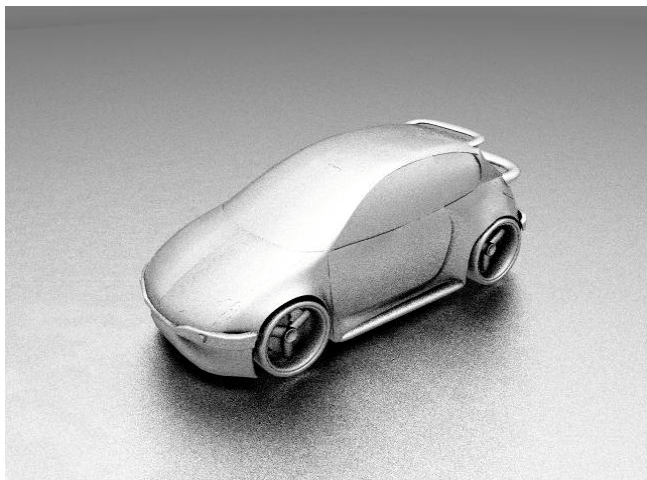
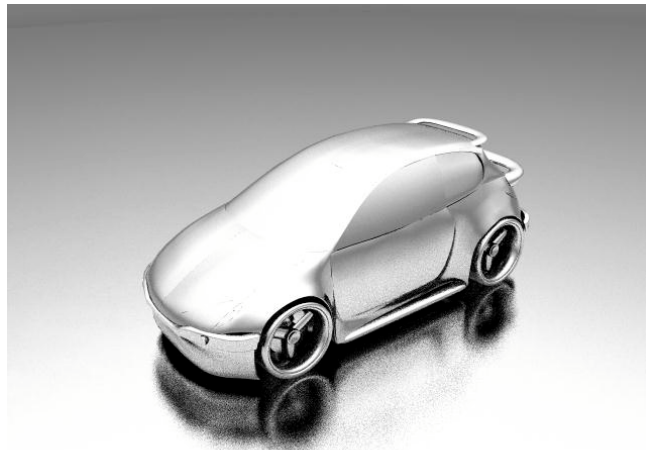
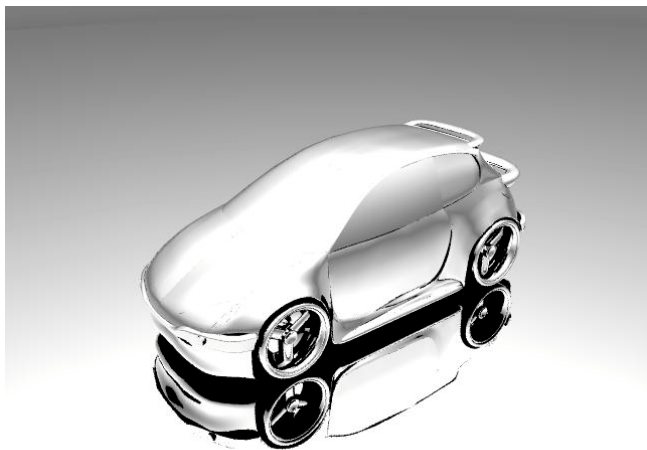
Ha a domborzat eléggé véletlenszerű, akkor a felületi normálvektorok iránya is közel azonos valószínűséggel néz a féltér minden irányába. Így a felületet különböző irányokból tekintve ugyanakkora valószínűséggel verődik a kamera felületre érkező fény. Ezt a tökéletesen diffúz viselkedést „enyhíthetjük” ha a domborzat nem annyira durva.



Ha az „érdes” felület olyan, hogy a normál vektorok többsége az eredeti felület normál vektorának irányába néz, un. glossy tükröződést, elmosott csillogást fogunk kapni. Hogy mennyire elmosódott a visszaverés – mennyire diffúz az anyag – azt a bump map durvasága vagy finomsága determinálja.

Létezik sok anyag – selyem, szőr, szatén, – amely sajátos szerkezete miatt jellegzetes fényvisszaverő tulajdonsággal rendelkezik. Ezek a jellemző szerkezetek a szövet, a szőrszálak vagy a rostok szálirányából adódnak. Ezek az anyagok a különböző irányokba másképpen verik vissza a fényt, hiszen a szabályos szerkezet miatt egyes szögek árnyékolódnak, míg más irányokba sokkal több fény juthat. Ha az általunk létrehozott domborzatot ennek megfelelően alakítjuk ki, vagyis az

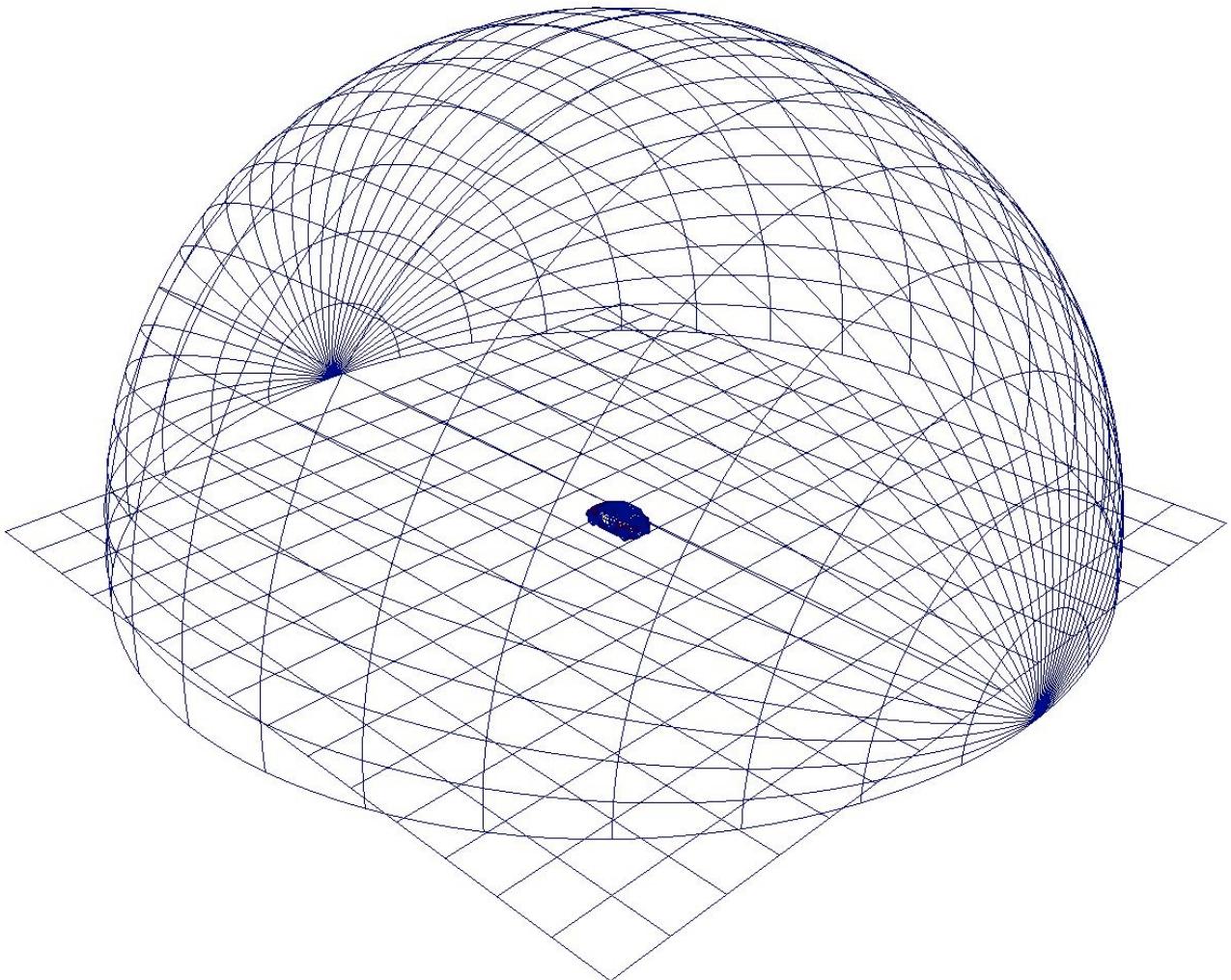
anyag szerkezetét megpróbáljuk az apró buckákkal reprodukálni, akár ilyen hatást is elérhetünk [7]. Természetesen az esetlegesen látható hullámoptikai jelenségek így sem fognak megjelenni. A tesztek során procedurális zaj – „brownian” – textúrát használtam, mely a Brown mozgásnak megfelelő véletlenszerű zajt generálja. Fix frekvencia mellett a zaj erősségével, vagyis a buckák magasságával befolyásoltam a tükröződés elmosottságát.



*Különböző erősségű bump map használatának hatása.*

### 3.4 Teszt Környezet

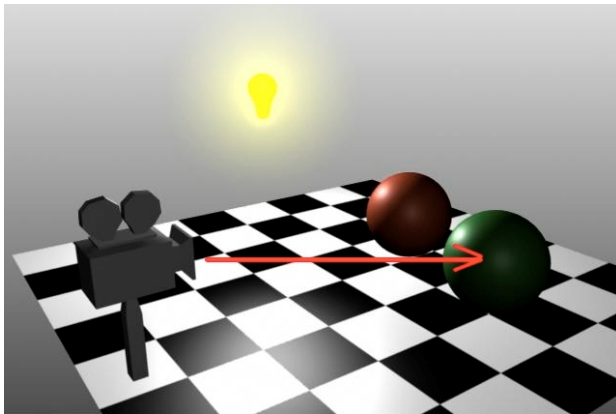
Vizsgálódásaimat az Alias|Wavefront 3d programcsomagjával, a Maya 3.0-as verziójával végeztem. A tesztekhez használt virtuális világ egy autó modelljéből, egy alapsíkból és egy félgömb alakú kupolából állt. Az autót NURBS felületekből állítottam össze. Ezeket a felületeket a Maya – sajnos – nem micropolygonokká tesszellálja a képszintézis során, hanem a felhasználó állítja be manuálisan a végső polygon-sűrűséget. A tesztek során kiderült, hogy az általam használt módszer különösen kihangsúlyozza a tesszelláció durvaságait, így kénytelen voltam meglehetősen finomra állítani azt. Az alapsík és az autó a fent leírt sűrű bump map-pel ellátott anyagot kapta, hiszen a földön és az autó apró mélyedéseiben akartam a derítéseket és az elmosott árnyékokat tettenérni. A kupola egy önfénnyel rendelkező és nem tükröző felület volt. Ez a kupola szolgált fényforrásként. Természetesen sugárkövetés használata nélkül a generált kép fekete maradt, hiszen nem definiáltam a lokális illuminációhoz szükséges absztrakt, pontszerű fényeket. A kupola önfénye a legmagasabb pontján fehér, lejjebb egyre szürkébb színt kapott a szép, valószerűbb megvilágítás elérése érdekében.



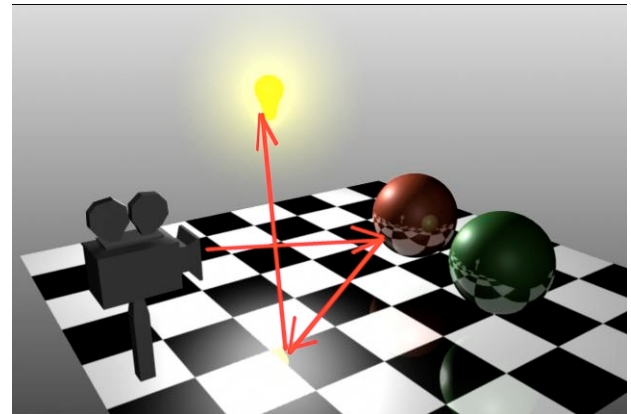
*A tesztkörnyezet drótvázás megjelenítése, középen az autó látható.*

## 4 Képszintézis

A képszintézis az általunk tárgyalt programok esetében mindig a kamerából – a pixeleken keresztül – történő sugarak bocsátásával kezdődik [3]. Ezekkel a sugarakkal a virtuális tér objektumait, felületeit elmetsszük. Ha a lokális illuminációs megjelenítésre hagyatkoznánk, akkor a metszéspontokban kiszámított színekkel meg is állnánk. Mivel algoritmusunk a rekurzív sugárkövetést használja, ezért az árnyaláshoz további sugarakat bocsátunk ki, hogy kiszámítsuk a tér különböző irányából beérkező fény mennyiségét.



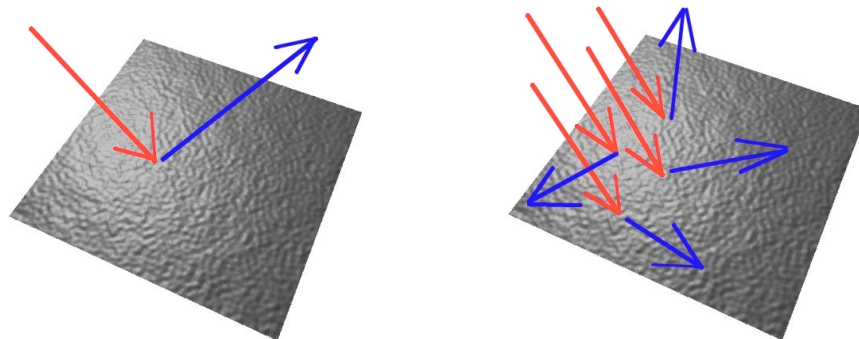
*Lokális illumináció*



*Sugárkövetés*

### 4.1 Gyűjtőséták

A rekurzív sugárkövetés során a kamerából indított sugarak elérik a felületeket, és a sűrű domborzat alapján kiszámított, többé kevésbé véletlen irányba újabb sugarat bocsátanak. A sugarak egészen addig verődnek felületről felületre, míg egy olyan testhez érnek, amely nem tükröző tulajdonságú. Minden visszaverődésnél a testek saját fényét ill. elnyelő tulajdonságát is figyelembe veszik az algoritmusok, így nem csak a tükröződés elmosottságát hanem erősségét is tetszőlegesen állíthatjuk. A tükröződés erősségének beállítására nem csak egy konstans áll rendelkezésünkre, hanem a három kitüntetett szín (R,G,B) mindegyikéhez egy-egy. Ha ezeket nem ugyan arra az értékre állítjuk, a testeket színezhajjuk, hiszen akár a valóságban bizonyos frekvenciájú fényt vissza fognak verni, bizonyosakat meg elnyelnek.



*Ha több gyűjtősétát indítunk egy pixel színének meghatározásához, azok különböző irányokba verődnek vissza*

Az erős önfénnyel rendelkező felületek mint fényforrások funkcionálnak, hiszen ezek jelennek meg a tükröződésekben. Mivel általában eltekintünk a fényforráson tükröződő fénytől – ki fésülködne egy lámpaizzó előtt – ezért ha egy sugár elér egy fényforrást akkor nem indítunk újabb sugarakat. Ennek alapján a fényforrások tükrözési együtthatóját 0-ra állítottam.

A kamerából indított sugarak így gyűjtősétákat végeznek felületről felületre, hiszen minden visszaverődésnél a tükörirányból érkező fény mennyiséget próbáljuk megállapítani. Ha minden felület tükröző lenne, a gyűjtőséták a végtelenségig tartanának. Ennek a programokban megadható maximális visszaverődési érték szabhat határt, de sajnos az így „elvarrt” sugarak sok esetben nem érik el a fényeket. Okosabb megoldás, ha a visszaverődések számát csak egy nagyon nagy értékben korlátozzuk, a fent említett okok miatt az önfénnyel rendelkező felületekről nem verjük vissza a sugarakat és ezeket a fényforrásokat a lehető legnagyobb kiterjedésűre vesszük. Így a gyűjtőséták nagy része néhány visszaverődés után a fényforrásokban „elakad”, és marad idő azon kevés sugár lekövetésére melyek csak sok pattanás után érik el a fényeket.

## 4.2 Mintavételezés

A képpont árnyalásához a képszintézis egyenletében szereplő integrált kell minél jobban közelíteni. Egy általános integrál közelítésére integrálformulákat használhatunk:

$$\int_V f(z) dz \approx \frac{1}{n} \cdot \sum_{i=1}^n f(z_i) \cdot s(z_i)$$

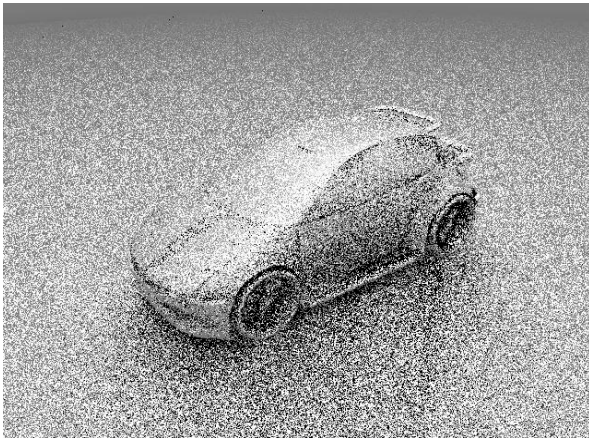
$n$  a minták száma,  $s()$  a súlyozó függvény és  $z_i$  pontok a mintavételi helyek. Esetünkben térszögek szerint kell integrálni, így a mintavételi helyek irányvektorok lesznek. Ezeket az irányokat a kamerából induló, majd visszaverődő gyűjtőséták iránya határozza meg.

Ha a pixelből csak egy ilyen sétát indítanánk – azaz az integrált egyetlen minta alapján próbálnánk kiértékelni – nagyon rossz közelítést kapnánk az eredménynek. Célunk az, hogy minden pixeleken keresztül lehetőleg minél több sétát indítsunk. Vegyük észre, hogy módszerünk – akár a Monte-Carlo integrálás – nem egyenletesen veszi a mintákat a féltérből, hanem fontosság szerint. A felület domborzata alapján nyilván azokba az irányokba fog több sugár verődni, merről a legtöbb fény jut a kamerába. Mivel fontosság szerint vettük a mintákat konstans egy súlyfüggvényt használunk az integrálformulában:

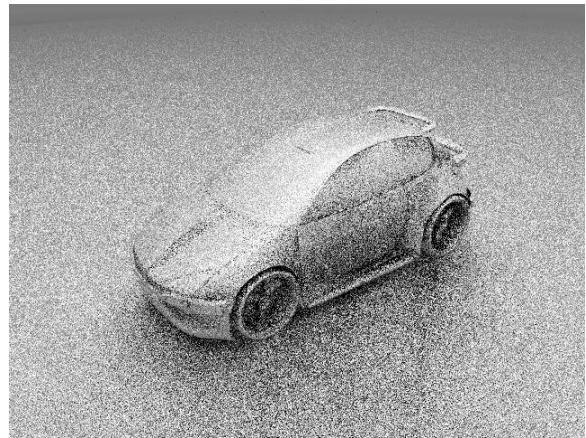
$$L(\vec{x}, \omega) \approx L^e(\vec{x}, \omega) + \frac{\sum_{i=1}^n L(h(\vec{x}, -\omega^i), \omega^i) \cdot R}{n}$$

ahol  $n$  a pixelenkénti minták száma, és  $-\omega^i$  az  $i$ -edik sugár visszaverődési iránya. Ez a képlet nem tartalmazza a fénytörést, ugyanis vizsgálódásaimban nem használtam átlátszó felületeket, de a tört sugarak is ugyan úgy viselkednének mint a visszavert sugarak. Mennyinek kell lennie  $n$ -nek, hogy kielégítő eredményt kapjunk?

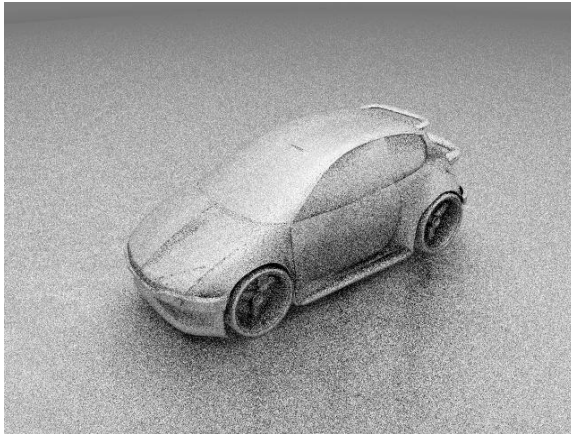
A minták még kielégítő számát az határozza meg, hogy mennyire erős zajt vagyunk hajlandóak elfogadni. Produkciós környezetben ez nagyon kicsi kell, hogy maradjon, de ha a képet mint referenciát használjuk a virtuális világ bevilágításához, akkor még elviselhető a jól látható zaj is. A különböző gyűjtőséta / pixel értékek lineárisan befolyásolják a renderidőt, azaz ha megduplazzuk a minták számát kétszer olyan hosszú számítási időre számíthatunk. A következő ábrán jól látható a zaj erőssége a gyűjtőséták számának függvényében.



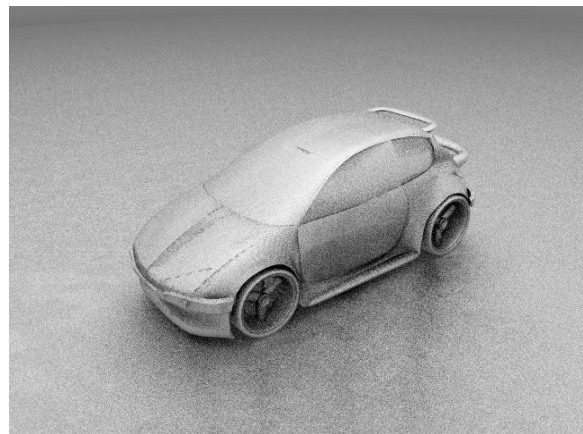
*1 minta pixelenként*



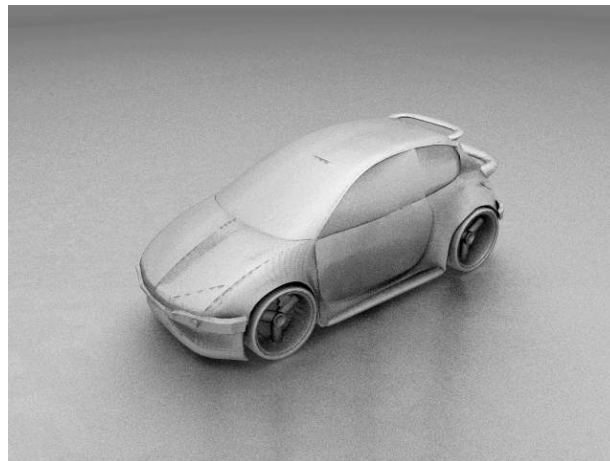
*3 minta pixelenként*



*10 minta pixelenként*



*32 minta pixelenként*



*320 minta pixelenként*

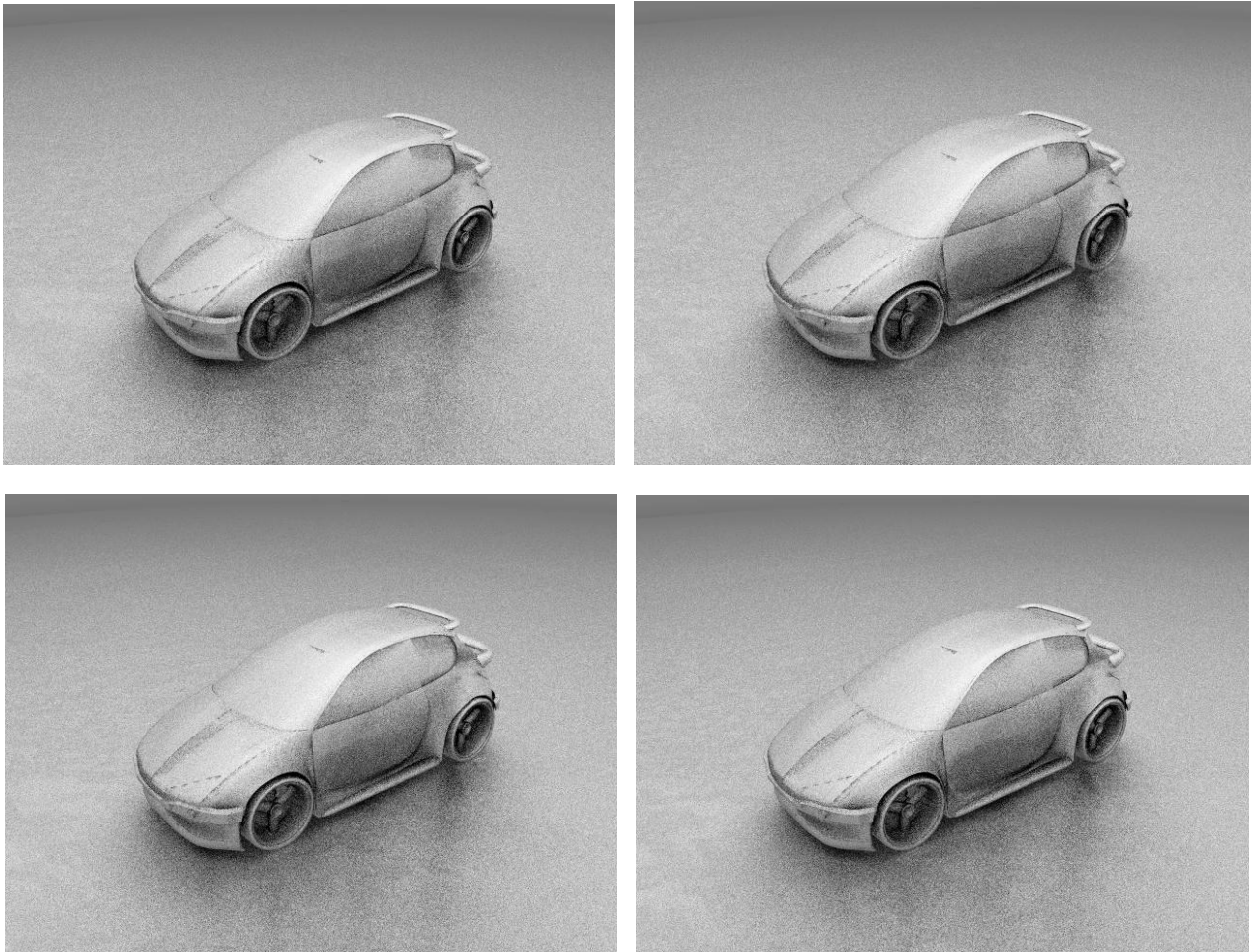


### 4.3 A zaj csökkentése a gyakorlatban

A vizsgálataimhoz használt program a pixelenkénti minták számát 32-ben maximálta [4], ami nem bizonyult elégnek a tesztképek alapján. Ezt a korlátozást úgy tudtam kiküszöbölni, hogy ugyanabból a kamera állásból több képet generáltam. Mivel a render program sztochasztikus mintavételezést használ, minden képen máshova estek a minták. Ezeket a képeket átlagolva ugyanazt a hatást érhettem el, mintha egy képet készítettem volna sokkal nagyobb mintavételezéssel.

Sajnos ez a módszer nem mindig kielégítő a minták számának növeléséhez. Ha a sztochasztikus mintavételezés belső megvalósítását nem ismerjük, lehet hogy tudunk nélkül nem teljesen véletlen lesz a minták eloszlása. Sőt az is könnyen előfordulhat, hogy képről képre ugyan oda esnek a minták. Vannak „butább” megjelenítő programok is, melyek nem is használnak sztochasztikus mintavételezést.

Az ilyen esetekben a bump map animálása lehet a megoldás. Mivel a domborzat olyan sűrű, hogy mintázatát nem szabad érzékelnünk a képen, apró elmozdítása nem jelent lényegi változtatást. Így megtehetjük, hogy képenként elmozdítjuk a textúrát és minden képen a pixelek ugyan azon pontjában veszünk mintát. Ezzel a módszerrel több képet átlagolva – melyeken a bump map elmozdulása miatt a visszaverődések is más irányba esnek – ugyanazt a hatást érhetjük el mintha mozdulatlan bump map mellett sztochasztikusan mintavételeztünk volna.



Minden képen eltoltam a bump map-et, szemmel látható különbség nincs.

## 5 Összefoglalás és fejlesztési lehetőségek

A bemutatott algoritmus jelentős kompromisszumok nélkül képes hagyományos 3d programok segítségével a felületek közötti fényszóródás és kiterjedt fényforrások modellezésére. Bár a képszintézis lassúsága és a képen látható zaj miatt a módszer csak ritkán használható produkciós környezetben [2], használata mégis hasznos lehet. Nagy segítség lehet például ahhoz, hogy megállapítsuk, hol jön létre derítés vagy elmosott árnyék a képen. Ezek alapján a pontszerű fénysugarakat úgy helyezhetjük el a virtuális világban, hogy a lokális illumináció segítségével is hasonló hatást érjünk el.

További kutatás témája lehet a bump map pontos vezérlésének megoldása, így lehetőség nyílna a különböző selymek, szatén anyagok realizisztikus modellezésére is. Mivel programról programra változik a textúrázás eszköztára, sajnos ez a probléma valószínűleg nem oldható meg általánosságban.

Szintén fontos probléma a zaj csökkentése, hiszen a fenti módszerrel nem készíthetünk fotorealisztikus képeket. Egy lehetséges, produkciókban is használható módszer lehet, ha valamilyen módon sikerülne úgy szűrni a képet, hogy azokat a lokális illuminációval generált képpel kombinálva, kompozitálva valószerű hatást eredményezzen.

## Irodalom

- [1] Szirmay-Kalos László: *Számítógépes grafika, ComputerBooks, Budapest 2000*
- [2] Apodaca, Gritz: *Advanced Renderman – Creating CG for Motion Picture (chapter 1.), Morgan Kaufmann, 2000*
- [3] Andrew Pearce, Kelvin Sung: *Maya software rendering – a technical overview, Alias|Wavefront, 1998*
- [4] Andrew Woo: *Aliasing artifacts in Maya – a technical overview, Alias|Wavefront, 1998*
- [5] Cabral, Max, Springmeyer: *Bidirectional reflection functions from surface bump maps. Proceedings of SIGGRAPH 87, p. 273-281*
- [6] Greenberg, Torrance, Shirley, Arvo, Ferwerda, Pattanaik, Lafortune, Walter, Foo, Trumbore: *A framework for realistic image synthesis. Proceedings of SIGGRAP 97, p. 477-494*
- [7] Ashikhmin, Premoze, Shirley: *A microfacet-based BRDF generator. Proceedings of SIGGRAPH 2000, p. 65-74*
- [8] Szirmay-Kalos L., Kelemen Cs.: *A microfacet based coupled specular-matte BRDF model with importance sampling. Eurographics Conference, 2001, Short presentations, pp 25-34.*
- [9] Vass Gergely: *Tökéletes tükröződések a 3D grafikában, Computer Arts Magazin, 2000 augusztus, p. 31-33*