

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1595

**Fizikalno temeljen model osvjetljenja i
osvjetljenje objekta temeljeno na slici**

Mladen Gelemanović

Zagreb, lipanj 2018.

**SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA DIPLOMSKI RAD PROFILA**

Zagreb, 9. ožujka 2018.

DIPLOMSKI ZADATAK br. 1595

Pristupnik: **Mladen Gelemanović (0036479732)**
Studij: Računarstvo
Profil: Računarska znanost

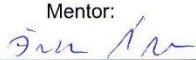
Zadatak: **Fizikalno temeljen model osvjetljenja i osvjetljenje objekta temeljeno na slici**

Opis zadatka:

Proučiti model osvjetljavanja Cook-Torrancea te razraditi doprinos utjecaja pojedinih članova modela na prikaz konačno iscrtanog objekta. Razraditi različite implementacije u ostvarivanju modela osvjetljavanja Cook-Torrancea. Proučiti i razraditi osvjetljenje objekta temeljeno na slici. Načiniti programsku implementaciju koja omogućuje analizu i usporedbu razrađenih modela. Na različitim primjerima prikazati ostvarene rezultate. Diskutirati utjecaj različitih parametara. Načiniti ocjenu rezultata i implementiranih algoritama.
Izraditi odgovarajući programski proizvod. Koristiti programski jezik C++ i grafičko programsko sučelje OpenGL. Rezultate rada načiniti dostupne putem Interneta. Radu priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Zadatak uručen pristupniku: 16. ožujka 2018.
Rok za predaju rada: 29. lipnja 2018.

Mentor:


Prof. dr. sc. Željka Mihajlović

Predsjednik odbora za
diplomski rad profila:


Prof. dr. sc. Siniša Srbljić

Djelovodja:


Doc. dr. sc. Tomislav Hrkać

Sadržaj

Uvod	2
1. Fizikalni model svjetla	3
1.1. Interakcija svjetla i tvari u različitim medijima	4
1.2. Raspršena i zrcalna svjetlost.....	5
1.3. Mikro površine	6
1.4. Fresnelov efekt.....	7
2. Matematički opis fizikalno temeljenog osvjetljenja	8
2.1. Pod površinska refleksija (raspršeni član)	9
2.2. Površinska refleksija (zrcalni član)	9
2.2.1. Funkcija distribucije normala.....	11
2.2.2. Funkcija geometrijskog pokrivanja	11
2.2.3. Funkcija Fresnelove refleksije.....	12
3. Osvjetljenje temeljeno na slikama.....	13
3.1. Izračun mape zračenja	14
3.2. Predfiltriranje okoline.....	17
3.3. Izračun BRDF za okolinu.....	19
4. Implementacija.....	22
4.1. Korišteni model materijala	22
4.2. Visoki raspon boja i gamma ispravak boja	24
4.3. Mogućnosti razvijenog programa i rezultati	25
Zaključak	31
Literatura	32

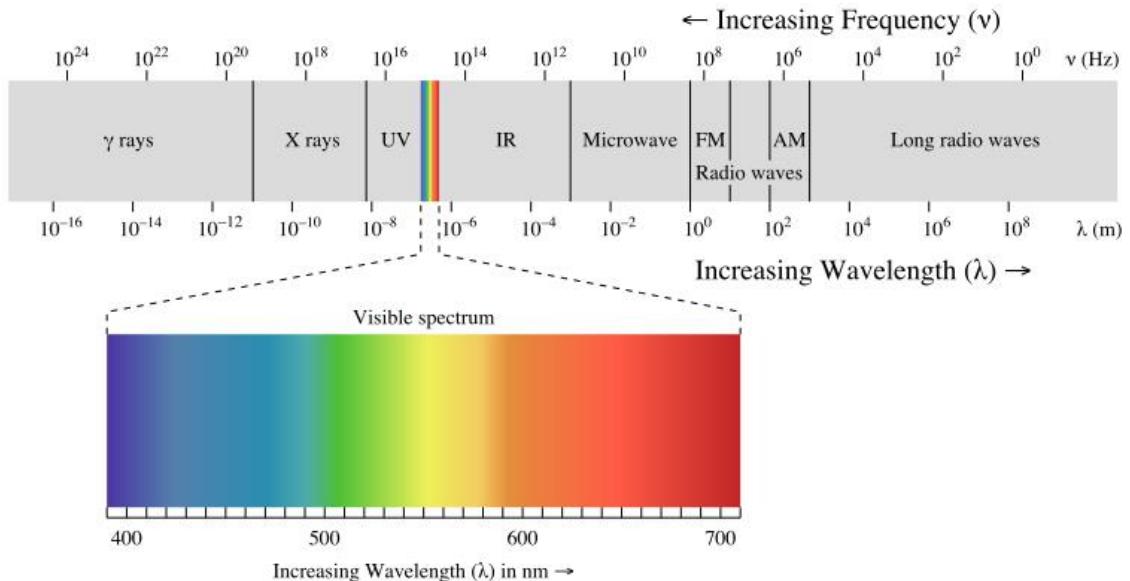
Uvod

Računalno generirane slike sve se više koriste danas u mnogim različitim područjima ljudskog djelovanja, kao što su medicina, industrija ili zabava. Početci računalne grafike bili su skromni, s obzirom da su računala bila puno slabija nego što su danas. Stoga su mnogi modeli i algoritmi računalne grafike koristili mnogobrojne trikove i aproksimacije kako bi se slika iscrtala s tadašnjom tehnologijom. Međutim, situacija je danas drugačija. S jačim računalima, moguće je implementirati bolje, fizikalno ispravnije načine iscrtavanja objekata i scena na ekran. Generalno se teži postizanju što realističnijih rezultata, kako bi se postigli detaljniji objekti, prirodnije scene, animacije i osvjetljenje ili ostvarili novi efekti i načini utjecanja na gotovu sliku.

Ovaj rad će objasniti teoriju i pokazati rezultate jednog modela fizikalno temeljenog osvjetljenja (eng. *physically based rendering*, PBR). Najprije će biti dan kratki opis nekih od najznačajnijih karakteristika stvarnog svjetla, nakon čega će biti opisan korišteni matematički model osvjetljenja. Zatim će se objasniti najbitniji dijelovi implementacije i pokazati rezultati izrađenog programa u izvođenju. Ovaj rad je temeljen na PBR pristupu koji je Disney istražio [1], i koji je Epic Games prilagodio za iscrtavanje u stvarnom vremenu [2]. Pristup je baziran na metalnom tijeku rada (eng. *metallic workflow*) i dobro je dokumentiran, u širokoj upotrebi i daje vizualno lijepo rezultate. Cilj ovoga rada je pokazati kako Disney/Epic PBR radi na različitim modelima u različitim okolinama. Dodatno, radi boljeg razumijevanja korištenog modela, omogućen je prikaz samo pojedinih komponenti osvjetljenja.

1. Fizikalni model svjetla

Za izradu fizikalno temeljenog modela osvjetljenja na računalu, potrebno je poznavati fizikalne karakteristike svjetla [3]. Svjetlo je elektromagnetski transverzalni val, odnosno val koji oscilira okomito u odnosu na smjer gibanja vala. Bitna karakteristika svjetla je njegova valna duljina. Valne duljine elektromagnetskih valova pokrivaju veliki raspon vrijednosti, takozvani elektromagnetski spektar (Slika 1). Neke od vrsta elektromagnetskih valova su: gamma zrake, ultraljubičaste zrake, vidljivo svjetlo, infracrvene zrake, radio valovi. Ono što nas zanima je vidljivo svjetlo, odnosno elektromagnetski valovi valne duljine od 400nm (ljubičasto svjetlo) do 700nm (crveno svjetlo).



Slika 1. Elektromagnetski spektar.

Još jedna važna veličina za razumijevanje fizike osvjetljenja je indeks refrakcije (raspršenja). Indeks refrakcije je kompleksni broj koji određuje kakav utjecaj će tvari imati na ponašanje svjetla. Realni dio indeksa određuje kako tvar utječe na brzinu kretanja svjetlosti, odnosno koliko je svjetlo sporije u odnosu na brzinu kretanja kroz vakuum. Kompleksni dio indeksa određuje koliko se svjetlo pretvara u druge oblike energije prolaskom kroz tvar, odnosno da li tvar apsorbira svjetlo.

1.1. Interakcija svjetla i tvari u različitim medijima

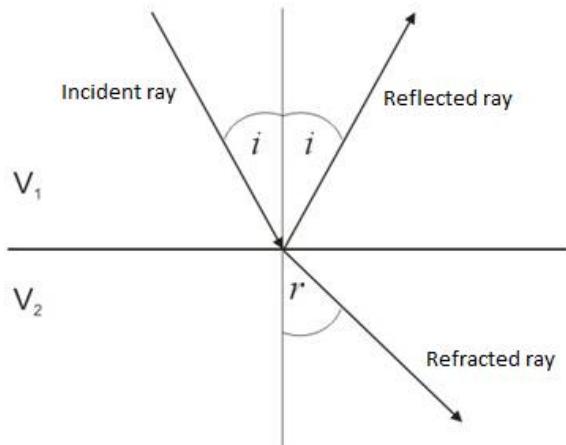
Najjednostavniji slučaj interakcije svjetla i tvari je kretanje svjetla kroz homogeni medij, odnosno medij s uniformnim indeksom refrakcije. Transparentni medij (npr. staklo, voda) ima indeks refrakcije s jako malim iznosom kompleksnog dijela indeksa, što znači da se svjetlo prolaskom kroz njega ne apsorbira i ne mijenja smjer kretanja. Ako medij ima indeks sa značajnom vrijednosti kompleksnog broja, dio svjetla će se apsorbirati u medij. Apsorbiranje će se pojačavati s povećanjem udaljenosti koje svjetlo prolazi kroz medij. Drugim riječima, prolaskom svjetla kroz takav medij, svjetlo će gubiti na intenzitetu, i ako medij apsorbira samo određene valne duljine, promijenit će se i boja svjetla.

Druga vrsta medija je heterogeni medij. To su mediji u kojima indeks refrakcije nije svugdje jednak. U slučaju da je promjena indeksa spora i kontinuirana, svjetlo će se zakriviti prolaskom kroz takav medij, dok će nagle promjene indeksa refrakcije uzrokovati lomljenje zrake svjetlosti u različitim smjerovima. Ako je medij takav da promijeni smjer kretanja svjetlosti u donekle nasumičnim smjerovima, pričamo o oblačnom mediju. S druge strane, neprozirni medij (eng. *translucent, opaque*) toliko mijenja smjer kretanja svjetlosti da on postaje potpuno nasumičan. Izgled svake tvari ovisi o relativnoj količini raspršivanja i apsorbiranja svjetlosti.

U slučaju kada se indeks refrakcije promijeni, ponašanje svjetla može se izračunati pomoću Maxwellovih jednadžbi. Međutim, te jednadžbe u većini slučaja nemaju analitičko rješenje. Jedan od slučaja koji ima točno rješenje je od posebnog značaja kada pričamo o osvjetljenju površina. Taj slučaj je kada su dva medija različitih indeksa refrakcije odijeljena beskonačno velikom i ravnom površinom. Kod predstavljanja površine nekog objekta, s jedne strane te granice se nalazi indeks refrakcije zraka, dok s druge strane je indeks refrakcije osvijetljenog objekta. Treba napomenuti kako ni jedna površina nije beskonačno velika ili savršeno ravna, ali u odnosu na veličinu valnih duljina vidljive svjetlosti, možemo ih takvim smatrati.

U ovom slučaju, kada zraka svjetlosti dođe do neke površine, ona se neće raspršiti kontinuirano u svim smjerovima, nego će se podijeliti na dvije zrake (**Error! Reference source not found.**). To su reflektirana i refraktirana zraka. Kut reflektirane zrake jednak je po iznosu kutu dolazne zrake svjetlosti. Kut refraktirane

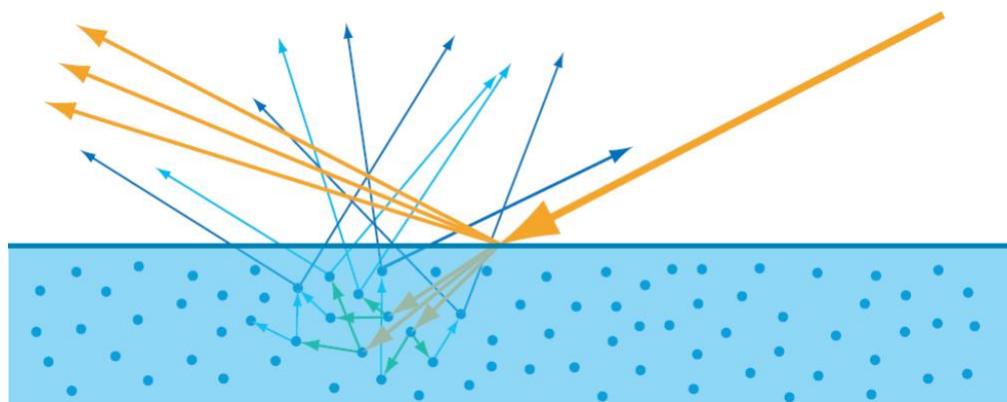
zrake ovisi o indeksu refrakcije medija i računa se pomoću Snellovog zakona. Omjer reflektirane i refraktirane svjetlosti određen je Fresnelovim jednadžbama.



Slika 2. Nagla promjena indeksa refrakcije na ravnoj površini između dva medija uzrokuje da se svjetlo rasprši u dva smjera.

1.2. Raspršena i zrcalna svjetlost

Kod interakcije površine i svjetla razlikujemo dvije vrste svjetla koje se reflektira (Slika 3). Kad zraka svjetlosti pogodi površinu, dio svjetlosti se odbija od površine (refleksija) dok drugi dio svjetlosti prodire dublje u obasjani objekt (refrakcija). To svjetlo može izaći iz objekta ili se raspršiti dublje u objekt. Svjetlo koje ne izađe iz objekta pretvara se u energiju, dok dio koji izađe zovemo raspršeno svjetlo (eng. *diffuse light*). Boju objekta vidimo kao raspršeno svjetlo (objekt koji upija većinu svjetla a raspršuje crvenu svjetlost izgleda kao crveni objekt).



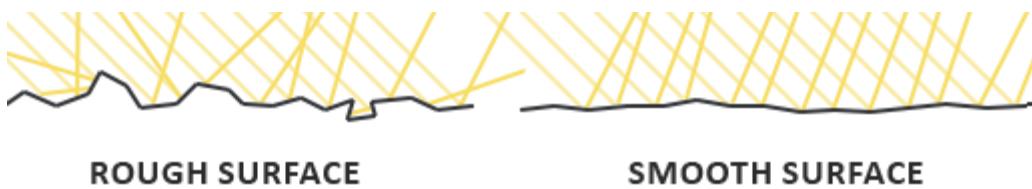
Slika 3. Prikaz raspršenog i zrcalnog svjetla.

Dio svjetlosti koji se pri sudaru svjetla s površinom odmah odbije je zrcalno svjetlo (eng. *specular light*). Zrcalno svjetlo se odbija u suprotnom smjeru od smjera dolaska svjetla u odnosu na normalu površine. To svjetlo daje zrcalni efekt objektu, jer se svjetlo odbija u jednom smjeru, za razliku od raspršenog svjetla koje izlazi iz objekta u svim smjerovima.

Za fizikalno ispravan model svjetla treba navesti i zakon očuvanja energije. Naime, svjetlo koje dođe do površine objekta se razdvaja na raspršeno i zrcalno svjetlo. Zbroj energija izlaznog svjetla mora biti jednak količini energije koja je osvijetlila objekt. Drugim riječima, što više svjetla objekt raspršuje, to izgleda manje zrcalno.

1.3. Mikro površine

Do sada opisano ponašanje svjetla je dovoljno za izradu fizikalno ispravnog programa za iscrtavanje. Međutim, većina stvarnih površina ima mnoge sitne nesavršenosti po sebi, što znači da je površina gruba, iako ne izgleda tako. U teoriji, svaka površina se na mikroskopskoj razini može predstaviti kao skup izuzetno malenih, savršeno reflektivnih zrcala koje zovemo mikro površine (eng. *microfacets*). Te sitne nesavršenosti utječu na refleksiju svjetla (Slika 4), jer zrake svjetla sada više ne upadaju sve pod istim kutom. Rezultat je da grublja površina stvara mutniju refleksiju.

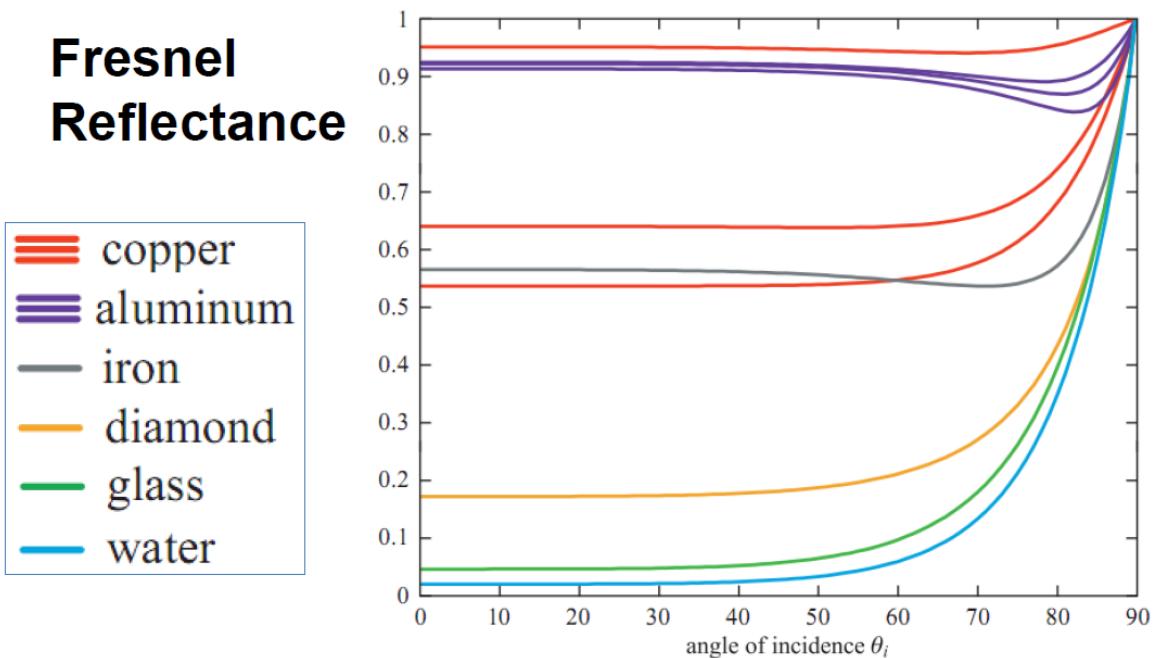


Slika 4. Prikaz raspršenja svjetla na više i manje gruboj površini.

Postavlja se problem prikazivanja tih sitnih nesavršenosti na računalu. Naime, ispravno prikazivanje svih detalja bilo bi memorijski i računalno previše zahtjevno. Stoga se u izradi programa za fizikalno ispravno sjenčanje koristi generalna mjera gruboće objekta.

1.4. Fresnelov efekt

Fresnelov efekt [4] je svojstvo svih materijala da mijenjaju reflektivnost ovisno o kutu upada svjetla. Svjetlo koje dođe do površine pod većim kutom će biti više reflektirano nego svjetlo koje dolazi ravno do površine. Kao rezultat, objekti će imati svjetliju refleksiju pri rubovima. Za sve materijale vrijedi da refleksija postaje potpuna na samim rubovima, odnosno da se svi glatki materijali na rubu ponašaju kao savršeno neobojano zrcalo. Dodatno, većina materijala ima sličnu krivulju koja definira količinu refleksije ovisno o upadnom kutu (Slika 5), gdje su metalni objekti najveće iznimke.



Slika 5. Fresnelova refleksija za različite materijale, ovisno o upadnom kutu.

S implementacijske strane, preporučljivo je da se kod izrade fizikalno ispravnih slika definira samo početna refleksivnost objekta, te da sustav za iscrtavanje izračuna dodatnu refleksivnost ovisno o kutu. Time se smanjuje količina potrebnih varijabli koje umjetnik mora odrediti i osigurava se fizikalna ispravnost.

2. Matematički opis fizikalno temeljenog osvjetljenja

Za izradu modela osvjetljenja koji se smatra fizikalno ispravnim, potrebno je zadovoljiti sljedeća tri svojstva: mora biti zasnovan na modelu mikro površina, mora zadovoljavati zakon očuvanja energije i mora koristiti fizikalno ispravnu funkciju dvosmjerne distribucije refleksije (eng. *bidirectional reflectance distribution function*, BRDF nadalje). Model mikro površina i zakon očuvanja energije su već opisani u prethodnom poglavlju. Za objašnjenje fizikalno ispravnog BRDF, prvo je potrebno objasniti jednadžbu iscrtavanja (eng. *render equation*) i jednadžbu refleksije (eng. *reflectance equation*). Jednadžba iscrtavanja je složena jednadžba koja je trenutno najbolji model za simuliranje vizualnih efekata svjetla. Fizikalno temeljeno osvjetljenje koristi jednadžbu refleksije (Jednadžba 1), specijaliziraju verziju jednadžbe iscrtavanja.

$$L_o(\mathbf{v}) = \int_{\Omega} f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\mathbf{l} \quad (1)$$

Ova jednadžba nam govori da je odlazni sjaj L_o (eng. *radiance*) jednak integralu (preko svih smjerova iznad površine) dolaznog sjaja L_i pomnožen s BRDF (oznaka f) i kosinusom kuta između normale na površinu \mathbf{n} i vektora upada svjetla \mathbf{l} (smjer od površine prema izvoru svjetlosti, suprotan dolaznom smjeru). Vektor pogleda označen je s \mathbf{v} (smjer od površine prema kameri, odlazni smjer).

BRDF je funkcija koja na temelju ulaznog smjera i izlaznog smjera svjetla računa količinu osvjetljenja u smjeru izlazne zrake. Za fizikalno ispravni model svjetla moraju vrijediti sljedeća svojstva za BRDF: vrijednost uvijek mora biti pozitivna, vrijedi Helmholtzov zakon reciprociteta (BRDF daje istu vrijednost ako ulazna i izlazna zraka svjetla zamijene uloge) i vrijedi zakon očuvanja energije (izlazna energija mora biti manja ili jednaka ulaznoj energiji zato jer površina ne može reflektirati više od 100% dolaznog svjetla).

BRDF vrijednosti se mogu izmjeriti iz stvarnih objekata ili izračunati analitički. Jedan od analitičkih metoda izračunavanja BRDF je Lambertov model refleksije koji se često koristi u računalnoj grafici. Korištenjem određenih analitičkih modela možemo koristiti BRDF za anizotropne refleksije, podržavati Fresnelov efekt,

omogućiti korištenje u Monte-Carlo algoritmima ili omogućiti uređivanje objekta korištenjem malog broja intuitivnih parametara. Postoji više fizikalno ispravnih BRDF, ali većina sustava za iscrtavanje u stvarnom vremenu (uključujući i ovaj rad) koristi Cook-Torrance BRDF (Jednadžba 2) [5]. Jednadžba se sastoji od omjera refraktirane svjetlosti (k_d), raspršenog člana ($f_{lambert}$), omjera reflektirane svjetlosti (k_s) i zrcalnog člana ($f_{cook-torrance}$).

$$f_r = k_d \cdot f_{lambert} + k_s \cdot f_{cook-torrance} \quad (2)$$

2.1. Pod površinska refleksija (raspršeni član)

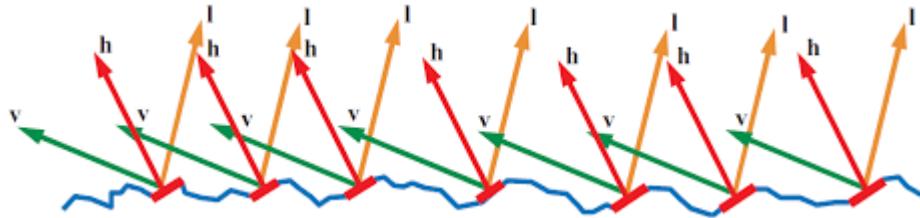
Postoji više modela za izračun pod površinske refleksije. Najjednostavniji i najčešće korišteni model među njima je Lambertov model. Epic Games je u svom radu razmotrio model iz Disneyevog rada, ali su došli do zaključka da je razlika u kvaliteti rezultata malena i nije vrijedna skupljeg modela. Lambertov BRDF je konstantna vrijednost koja ovisi o albedo boji materijala (c_{diff}). Raspršeni član (eng. *diffuse term*) Cook-Torranceovog modela BRDF se računa prema danom izrazu (Jednadžba 3).

$$f_{lambert}(\mathbf{l}, \mathbf{v}) = \frac{c_{diff}}{\pi} \quad (3)$$

2.2. Površinska refleksija (zrcalni član)

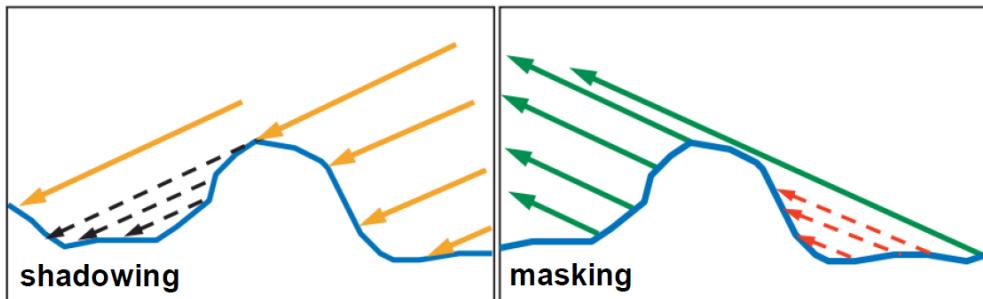
Zrcalni član (eng. *specular term*) Cook-Torranceovog modela BRDF je komplikiraniji za izračunati od raspršenog člana. Izračun je temeljen na modelu mikro površina, što znači da svaka točka površine reflektira dolaznu zraku svjetlosti u samo jednu odlaznu zraku. Taj kut refleksije je određen pomoću normale \mathbf{m} na pogodjenu mikro površinu. Kada se računa BRDF, dolazni smjer svjetla \mathbf{l} i odlazni smjer svjetla \mathbf{v} su zadani. To znači da od svih točaka površine, samo one točke koje imaju takvu orientaciju da reflektiraju \mathbf{l} u \mathbf{v} mogu pridonijeti vrijednosti BRDF (Slika 6). Možemo vidjeti da takve mikro površine imaju normalu koja se nalazi na

pola puta između vektora \mathbf{l} i \mathbf{v} . Taj vektor na sredini između \mathbf{l} i \mathbf{v} ćemo nazvati poluvektor \mathbf{h} .



Slika 6. Točke površine koje pridonose vrijednosti zrcalnog člana BRDF.

Treba napomenuti kako neće sve točke površine kod kojih se normala na mikro površinu \mathbf{m} i poluvektor \mathbf{h} poklapaju, stvarno i pridonositi refleksiji. Razlog tome je što mikro geometrija može blokirati dolaznu zraku svjetlosti \mathbf{l} , odlaznu zraku svjetlosti \mathbf{v} ili obje zrake (Slika 7). Kad je dolazna zraka blokirana govorimo o efektu zasjenjivanja (eng. *shadowing*), dok ako je odlazna zraka blokirana govorimo o efektu maskiranja (eng. *masking*). Po teoriji mikro površina, zaklonjeno svjetlo nikako ne utječe na zrcalni član.



Slika 7. Na lijevoj slici vidimo neke točke površine koje su blokirane od strane dolaznog svjetla, što znači da do njih ne dolazi svjetlo (u sjeni su) i stoga ne mogu ni reflektirati nikakvo svjetlo. Na desnoj slici vidimo da je odlazna zraka svjetlosti (reflektirana od površine) blokirana i neće biti vidljiva.

Na temelju prijašnjih razmatranja možemo predstaviti zrcalni član Cook-Torranceovog modela BRDF (Jednadžba 4). U danoj formuli možemo vidjeti tri funkcije koje aproksimiraju određeni dio reflektivnih svojstva površine: funkcija distribucije normala (D), funkcija geometrijskog pokrivanja (G) i funkcija Fresnelove refleksije (F). Svaka od tih funkcija ima više verzija. Neke od tih verzija daju realističnije rezultate, dok su druge računski efikasnije. U sljedećim poglavljima, svaka od tih funkcija će biti opisana i bit će dana matematička formulacija odabrane verzije.

$$f_{\text{cook-torrance}}(\mathbf{l}, \mathbf{v}) = \frac{F(\mathbf{v}, \mathbf{h}) \cdot G(\mathbf{l}, \mathbf{v}, \mathbf{h}) \cdot D(\mathbf{h})}{4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})} \quad (4)$$

2.2.1. Funkcija distribucije normala

Funkcija distribucije normala (eng. *normal distribution function*) statistički procjenjuje relativnu količinu površine mikro površina koje su točno orijentirane u danom smjeru. Ova funkcija određuje veličinu, sjajnost i oblik zrcalnog svjetlosnog naglaska (eng. *specular highlight*). Ulazni parametri ove funkcije su: vektor normale na površinu \mathbf{n} , poluvektor \mathbf{h} i mjera gruboće površine (reparametrizirano s $\alpha = \text{Roughness}^2$). Rezultat funkcije je skalarni, nenegativni broj (rezultat može biti veći od 1, što označava jako visoku koncentraciju točaka površine koji imaju normalu koja pokazuje u danome smjeru). U ovome radu je korištena GGX/Trowbridge-Reitz verzija funkcije distribucije normala, zadana sljedećom jednadžbom.

$$D(\mathbf{h}) = \frac{\alpha^2}{\pi((\mathbf{n} \cdot \mathbf{h})^2(\alpha^2 - 1) + 1)^2} \quad (5)$$

2.2.2. Funkcija geometrijskog pokrivanja

Geometrijska funkcija predstavlja vjerojatnost da je točka površine s danom normalom \mathbf{m} na mikro geometriju vidljiva iz smjera svjetla \mathbf{l} i iz smjera pogleda \mathbf{v} . Kod BRDF koji koristi mikro površine, vektor \mathbf{m} možemo zamijeniti s poluvektorom \mathbf{h} . Pošto ova funkcija računa vjerojatnost, rezultat funkcije je skalarni broj između 0 i 1. Geometrijska funkcija pokrivanja je ključna za zakon očuvanja energije kod BRDF. Mjera gruboće materijala je ulazni parametar i za ovu funkciju, gdje grublje površine znače da je veća vjerojatnost zasjenjivanja mikro površina. Za korištenje u funkciji, mjeru gruboće je potrebno reparametrizirati (Jednadžba 6). Korištena geometrijska funkcija je kombinacija GGX i Schlick-Beckmann aproksimacije koja se naziva Schlick-GGX (Jednadžba 7). Za ispravnu procjenu geometrije, potrebno je uzeti u obzir i smjer gledanja (geometrijska opstrukcija, eng. *geometry obstruction*) i smjer svjetla (geometrijsko zasjenjivanje, eng. *geometry shadowing*). Oba smjera mogu se uzeti u obzir sa Smithovom metodom (Jednadžba 8).

$$k = \frac{(Roughness + 1)^2}{8} \quad (6)$$

$$G_1(\mathbf{v}) = \frac{\mathbf{n} \cdot \mathbf{v}}{(\mathbf{n} \cdot \mathbf{v})(1 - k) + k} \quad (7)$$

$$G(\mathbf{l}, \mathbf{v}, \mathbf{h}) = G_1(\mathbf{l}) \cdot G_1(\mathbf{v}) \quad (8)$$

2.2.3. Funkcija Fresnelove refleksije

Funkcija Fresnelove refleksije računa udio svjetla koji se reflektira od optički ravne površine. Rezultat funkcije ovisi o kutu pod kojim se gleda na površinu i o indeksu refrakcije materijala. Iz izračunatog udjela i zakona očuvanja energije moguće je direktno izračunati udio refraktiranog svjetla.

Materijali imaju baznu razinu reflektivnosti F_0 (eng. *base reflectivity*) kad se gleda ravno na površinu (pod kutem od 0°). Kako se kut gledanja povećava, tako se povećava i udio svjetla koji se reflektira. Pod kutem od 90° svo svjetlo se reflektira od površine (prije objašnjeni Fresnelov efekt). Potpuna Fresnelova jednadžba je poprilično kompleksna, stoga se najčešće u računalnoj grafici koristi Fresnel-Schlickova aproksimacija (Jednadžba 9).

$$F(\mathbf{v}, \mathbf{h}) = F_0 + (1 - F_0)(1 - (\mathbf{h} \cdot \mathbf{v}))^5 \quad (9)$$

Treba napomenuti kako je Fresnel-Schlickova aproksimacija definirana samo za nemetalne površine. Za metalne površine, računanje bazne reflektivnosti materijala se ne može napraviti koristeći indeks refrakcije, pa je potrebna potpuno drugačija Fresnelova jednadžba za metale. Kako želimo koristiti samo jednu jednadžbu za oba tipa materijala, uvodimo još jednu aproksimaciju. To postižemo računanjem bazne reflektivnosti i korištenjem te vrijednosti u Fresnel-Schlickovoj aproksimaciji. Dodatno, baznu reflektivnost metala potrebno je obojati (eng. tint) nijansom boje površine (interpolacija s mjerom metalnosti između početne bazne reflektivnosti i boje površine).

3. Osvjetljenje temeljeno na slikama

Osvjetljenje temeljeno na slikama (eng. *image based lighting*, IBL) [6] je skup tehnika osvjetljenja objekata koje ne koriste analitičke izvore svjetlosti, nego smatraju čitavu okolinu kao jedan veliki izvor svjetlosti. To se postiže korištenjem okoline direktno u jednadžbama osvjetljenja, odnosno, svaki slikovni element okoline tretiramo kao novi izvor svjetlosti. Ovim postupcima, moguće je efikasno prikazati globalno svjetlo i ugođaj okoline, dajući objektima bolji osjećaj pripadanja toj okolini. Osvjetljenje temeljeno na slikama se može smatrati preciznijim prikazom ambijentalnog svjetla, ili čak grubom aproksimacijom globalnog osvjetljenja (eng. *global illumination*). Iz ovih razloga je zanimljivo koristiti osvjetljenje temeljeno na slikama uz fizikalno temeljeno osvjetljenje jer objekti tada izgledaju još više fizikalno ispravno.

Za korištenje osvjetljenja temeljenog na slikama, morat ćemo opet proučiti jednadžbu refleksije (Jednadžba 1). Kao što je prije navedeno, potrebno je riješiti integral za sve dolazne zrake svjetla preko čitave hemisfere. U prijašnjem poglavlju, to je bilo jednostavno za napraviti, jer smo znali koje zrake svjetla doprinose integralu (po jedna zraka za svaki izvor svjetlosti). Sada, svaki dolazni smjer svjetla koji dolazi od okoline može pridonositi integralu, što komplicira račun. Za rješavanje ovog integrala moramo zadovoljiti dva uvjeta: potrebno je dobiti sjaj (eng. *radiance*) scene za proizvoljni vektor smjera i računanje integrala mora biti dovoljno brzo da se izvodi u stvarnom vremenu.

Jedan od načina prikaza okolinskog zračenja (eng. *irradiance*) je u obliku obrađene okolinske kubične mape (eng. *environment cubemap*). S takvom mapom, svaki slikovni element mape možemo vizualizirati kao jedan emitirajući izvor svjetlosti. Uzorkovanjem te mape s proizvoljnim vektorom smjera možemo dobiti sjaj scene za taj smjer, time zadovoljavajući prvi uvjet. Za rješavanje integrala, potrebno je uzorkovati okolinsku mapu za sve moguće smjerove preko hemisfere, što je preskupo da se obavlja za svaki poziv programa za sjenčanje fragmenata (eng. fragment shader). Za efikasnije rješavanje integrala i zadovoljavanje drugog uvjeta, morat ćemo većinu potrebnog izračuna obaviti na početku izvođenja programa (jednom) i onda koristiti te međurezultate tijekom izvođenja programa za računanje

osvjetljenja temeljenog na slikama. Za to postići, bit će potrebno jednadžbu refleksije detaljnije razmotriti. U nastavku se nalazi raspisani oblik te jednadžbe za referencu (Jednadžba 10).

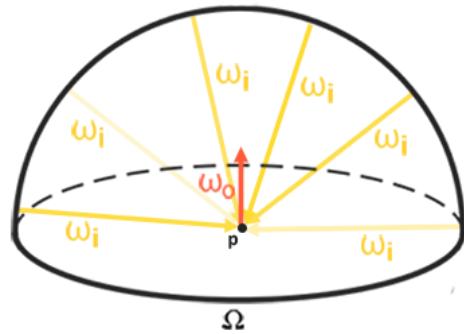
$$L_o(\mathbf{v}) = \int_{\Omega} \left(k_d \frac{c}{\pi} + k_s \frac{FGD}{4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})} \right) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\mathbf{l} \quad (10)$$

3.1. Izračun mape zračenja

Kada pobliže promotrimo jednadžbu refleksije, uočavamo da su raspršeni i zrcalni član neovisni jedan od drugoga i da integral možemo podijeliti. U ovome poglavlju ćemo promotriti kako riješiti raspršeni dio integrala. Još jednu stvar koju primjećujemo iz jednadžbe je činjenica da je Lambertov raspršeni član konstantna vrijednost i može se izvući iz integrala, Konačno, jednadžbu ćemo raspisati na drugačiji način. Vektore smjera \mathbf{l} i \mathbf{v} ćemo maknuti iz jednadžbe i zamijeniti ih s točkom površine p i vektorima smjera ω_o i ω_i . Time dobivamo novi oblik jednadžbe za raspršeni član integrala (Jednadžba 11).

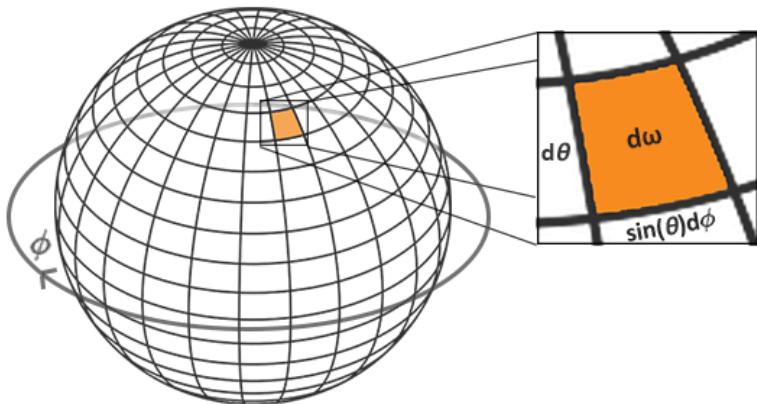
$$L_o(p, \omega_o) = k_d \frac{c}{\pi} \int_{\Omega} L_i(p, \omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i \quad (11)$$

Ovaj integral sada ovisi samo o ω_i (pod pretpostavkom da se p nalazi u središtu okolinske mape). Sada možemo izračunati raspršeni integral za svaki smjer ω_o pomoću konvolucije i spremiti rezultat u novu kubičnu mapu. Za konvoluciju okolinske mape potrebno je izračunati integral za svaki ω_o tako da diskretno uzorkujemo veliki broj vektora smjera ω_i preko hemisfere Ω i usrednjimo njihov sjaj. Hemisfera koju koristimo za dobivanje uzoraka ω_i je orijentirana u smjeru vektora ω_o kojeg konvoluiramo (Slika 8). Ova nova kubična mapa sadrži rezultat integrala za svaki vektor smjera ω_o i može se smatrati kao suma svog indirektnog raspršenog svjetla scene koje pogađa neku površinu orijentiranu u smjeru ω_o . Tu mapu nazivamo mapa zračenja (eng. *irradiance map*) i ona nam dopušta da direktno uzorkujemo (predizračunato) zračenje scene za proizvoljni smjer.



Slika 8. Hemisfera iz koje se uzorkuju vektori smjera, orientirana u smjeru vektora kojeg trenutno konvoluiramo.

Vektore smjera koje uzorkujemo iz hemisfere će biti uniformno raspoređeni po hemisferi. Integral je kontinuirana funkcija i njegovo rješavanje s diskretnim uzorkovanjem će biti samo aproksimacija integrala. Bolja aproksimacija integrala se postiže s uzorkovanjem više vektora smjera iz hemisfere. Korak integracije $d\omega$, nam predstavlja probleme u implementaciji. Stoga, kao korak integracije ćemo koristiti ekvivalentne sferne koordinate ϕ i θ (Slika 9).



Slika 9. Odnos između kuta ω i sfernih koordinata ϕ i θ .

Kut ϕ koristimo za uzimanje uzoraka iz prstena oko hemisfere za kuteve između 0 i 2π . Kut θ koristimo za uzimanje uzoraka iz rastućih prstena hemisfere za kuteve između 0 i $\pi/2$. Time dobivamo novi oblik jednadžbe integrala (Jednadžba 12). Dobiveni dupli integral ćemo riješiti korištenjem Monte Carlo procjenitelja (eng. estimator) na svakom integralu. Sada imamo diskretni oblik jednadžbe (Jednadžba 13) koji možemo iskoristiti u kodu programa za sjenčanje.

$$L_o(p, \phi_o, \theta_o) = k_d \frac{c}{\pi} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} L_i(p, \phi_i, \theta_i) \cos(\theta) \sin(\theta) d\phi d\theta \quad (12)$$

$$L_o(p, \phi_o, \theta_o) = k_d \frac{c}{\pi} \frac{2\pi}{N_1} \frac{\pi}{2N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} L_i(p, \phi_i, \theta_i) \cos(\theta) \sin(\theta) \quad (13)$$

Dvostruku sumu predstavljamo kao dvostruku petlju u kodu. Unutar petlje uzimamo obje sferne koordinate i pretvaramo ih u 3D vektor u Kartezijevom prostoru. Zatim taj vektor transformiramo iz prostora tangente (eng. *tangent space*) u prostor svijeta (eng. *world space*) i onda taj vektor koristimo da uzorkujemo okolinsku mapu. Svaki rezultat uzorka zbrajamo i na kraju taj zbroj dijelimo s brojem uzorka da usrednjimo rezultat. Vrijednost koju dobijemo iz okolinske mape množimo s faktorom kosinusa (zato jer je svjetlo slabije pri većim kutevima) i faktorom sinusa (zato jer iz viših područja hemisfere uzorkujemo manje površine). Time smo dobili program za sjenčanje (eng. *shader*) koji računa zračenje okoline.

Sve što je sada potrebno, je na početku programa poslati okolinsku mapu na grafičku karticu i iscrtati kocku šest puta, tako da svaki put kamera gleda u novu stranu kocke. Rezultat crtanja se ne prikazuje na ekranu, nego se zapisuje u novu kubičnu mapu (Slika 10), koja će služiti kao mapa zračenja tijekom izvođenja programa.



Slika 10. Kubična mapa okoline (lijevo) i izračunata mapa zračenja okoline (desno).

3.2. Predfiltriranje okoline

Sve što nam je sada ostalo da implementiramo osvjetljenje temeljeno na slikama je da riješimo dio integrala koji nam je ostao, zrcalni dio. Za razliku od raspršenog dijela integrala, zrcalni dio nije konstantna vrijednost preko integrala, nego ovisi o dolaznom i odlaznom smjeru svjetla. Rješavanje integrala za sve dolazne smjerove, u kombinaciji sa svim odlaznim smjerovima, je kombinatorna eksplozija i preskupo je da to računamo u stvarnom vremenu. Epic Games je predložio rješenje u kojem obave konvoluciju zrcalnog dijela za potrebe crtanja u stvarnom vremenu, uz neke kompromise, koje su nazvali aproksimacija podijeljene sume (eng. *split sum approximation*). Ova aproksimacija podijeli zrcalni dio integrala na dva dijela koje možemo zasebno konvoluirati i kasnije iskoristiti u PBR programu za sjenčanje, time dobivajući zrcalni dio osvjetljenja temeljenog na slikama. Kao i u prijašnjem poglavlju, promijenit ćemo oblik jednadžbe zrcalnog integrala, zamjenjujući \mathbf{I} i \mathbf{v} za p , ω_o i ω_i (Jednadžba 14). Cook-Torranceov BRDF (Jednadžba 4) ćemo kraće zapisati kao $f_r(p, \omega_i, \omega_o)$. Na kraju, podjelom integrala na dva dijela dobivamo aproksimaciju podijeljenom sumom (Jednadžba 15).

$$L_o(p, \omega_o) = \int_{\Omega} f_r(p, \omega_i, \omega_o) L_i(p, \omega_i)(\mathbf{n} \cdot \omega_i) d\omega_i \quad (14)$$

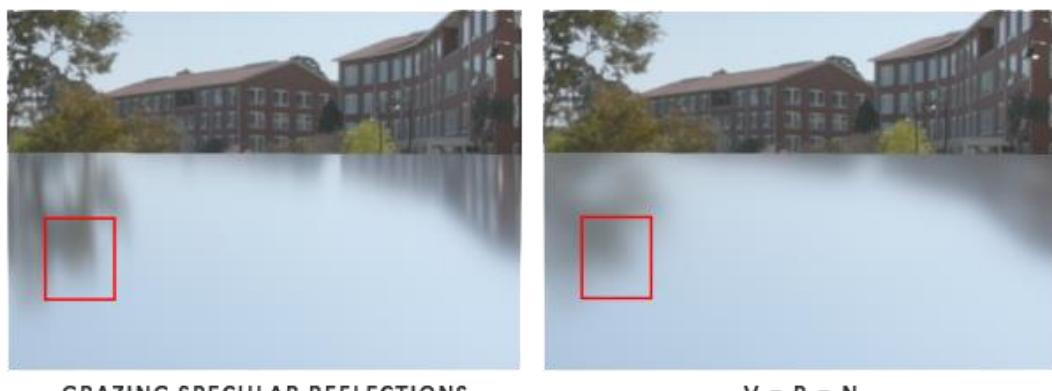
$$L_o(p, \omega_o) = \int_{\Omega} L_i(p, \omega_i) d\omega_i * \int_{\Omega} f_r(p, \omega_i, \omega_o)(\mathbf{n} \cdot \omega_i) d\omega_i \quad (15)$$

Lijevi integral (nakon konvolucije) nazivamo predfiltriranom okolinom (eng. *pre-filtered environment map*). Predfiltrirana okolina je izračunata konvolucijska okolinska mapa (eng. *pre-computed environment convolution map*) i slična je mapi zračenja, ali sada u izračunu također uzimamo gruboću u obzir. Za veće razine gruboće, okolinska mapa se konvoluira sa sve više raspršenim vektorima uzorka (eng. *sample vectors*), što stvara zamućenje refleksije. Za svaku razinu gruboće za koju obavimo konvoluciju, rezultat spremamo u novu mipmap razinu predfiltrirane okoline (Slika 11).



Slika 11. Mipmap razine predfiltrirane okoline (sa rastućom razinom gruboće).

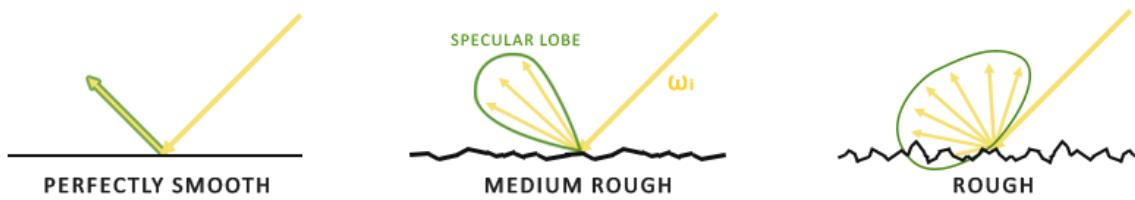
Vektore uzorka i njihovu snagu raspršenja generiramo pomoću funkcije distribucije normala koju koristimo u Cook-Torrance BRDF-u. Ta funkcija kao ulazne parametre uzima normalu i vektor smjera gledanja. Problem je što mi ne znamo vektor smjera gledanja kada računamo konvoluciju. Epic Games je ovdje uveo još jednu aproksimaciju, tako što su pretpostavili da se vektor smjera gledanja (V) poklapa s vektorom refleksije (R) i odlaznim vektorom (ω_o , koji je u ovom slučaju jednak normali na površinu N). Zbog ove aproksimacije gubimo izdužene refleksije kada gledamo površinu pod većim kutem (Slika 12), ali to smatramo dobriim kompromisom jer sada konvolucija predfiltrirane okoline ne ovisi o kutu gledanja.



Slika 12. Prikaz refleksija bez aproksimacije koju je Epic Games uveo (lijevo) i s aproksimacijom (desno).

Kod mape zračenja, konvoluciju okolinske mape smo napravili korištenjem vektora koji su bili uniformno raspoređeni preko hemisfere. Za zrcalnu refleksiju isplati se koristiti drugačiji pristup uzimanja vektora. Kod zrcalne refleksije, svjetlo

se ili reflektira u smjeru vektora refleksije ili približno u tom smjeru, ovisno o gruboći površine (Slika 13). Generalni oblik svih mogućih smjerova refleksije zovemo zrcalna školjka (eng. *specular lobe*). Pošto većina reflektiranih zraka završi u zrcalnoj školjci koja je orijentirana oko poluvektora mikro površina, ima smisla da vektore uzorka koje ćemo koristiti za konvoluciju integrala generiramo na slični način. Taj postupak nazivamo značajno uzorkovanje (eng. *importance sampling*).



Slika 13. Reflektirano svjetlo u potpunosti prati smjer refleksije na ravnoj površini (lijevo), povećanje gruboće površine uzrokuje raspršenje refleksije, ali generalni smjer i dalje prati smjer refleksije (sredina, desno).

Postupak predfiltriranja je sada jednostavan za objasniti. Prvi korak je izjednačavanje vektora V , R i N , nakon čega započinjemo petlju koja nam služi da uzmemo fiksni broj uzoraka. Unutar petlje, generiramo poluvektor mikro površine pomoću značajnog uzorkovanja. Iz vektora pogleda i poluvektora izračunamo vektor dolaznog smjera svjetla. Sada pomoću vektora svjetla uzorkujemo okolinsku mapu. Dobivenu vrijednost prvo pomnožimo s kosinusom kuta između odlaznog vektora i vektora dolaznog svjetla, nakon čega ju dodamo konačnoj vrijednosti (pod uvjetom da kosinus ima vrijednost veću od nule). Nakon što se uzmu svi uzorci, konačna vrijednost se dijeli sa zbrojem svih nenegativnih težina (kosinusa). Rezultat zapisujemo u određenu mipmap razinu (određenu razinom gruboće površine) predfiltrirane kubične mape. Čitav postupak ponavljamo nekoliko puta za različite razine gruboće površine.

3.3. Izračun BRDF za okolinu

Jedina stvar koja nam je ostala za implementaciju osvjetljenja temeljenog na slikama je rješavanje desnog integrala aproksimacije podijeljenom sumom (Jednadžba 15). Za rješavanje desnog integrala moramo konvoluirati BRDF preko kuta, gruboće površine i Fresnelove bazne reflektivnosti. Postupak je sličan integriranju zrcalnog BRDF s potpuno bijelom okolinskom mapom. Umjesto da

konvoluciju obavljamo preko tri varijable, mi ćemo baznu reflektivnost izvući van iz jednadžbe zrcalnog BRDF (Jednadžba 16.a). Prvo, integral pomnožimo i podijelimo s Fresnelovom refleksijom (Jednadžba 16.b), nakon čega Fresnelovu refleksiju s kojom množimo integral raspišemo (Jednadžba 16.c). Da nam bude lakše baratati s jednadžbom, član $(1 - \omega_o \cdot \mathbf{h})^5$ ćemo zapisati kao α i zatim ćemo raspisano Fresnelovu refleksiju napisati na drugačiji način (Jednadžba 16.d). Sada vidimo da integral možemo podijeliti na dva neovisna integrala (Jednadžba 16.e). Vrijednost F_0 je sada konstantna preko integrala i možemo ju izvući van, iz Cook-Torrance BRDF izbacimo Fresnelovu funkciju (jer smo BRDF podijelili s njom) i dobivamo $f_r' (p, \omega_i, \omega_o)$, te na kraju α ponovno raspišemo kao $(1 - \omega_o \cdot \mathbf{h})^5$. Time smo dobili dva integrala (Jednadžba 16) koji služe kao težina (eng. *weight, scale*) i pristranost (eng. *bias*) uz vrijednost F_0 .

$$L_o(p, \omega_o) = \int_{\Omega} f_r(p, \omega_i, \omega_o) (\mathbf{n} \cdot \omega_i) d\omega_i \quad (16.a)$$

$$= \int_{\Omega} \frac{f_r(p, \omega_i, \omega_o)}{F(\omega_o, \mathbf{h})} F(\omega_o, \mathbf{h}) (\mathbf{n} \cdot \omega_i) d\omega_i \quad (16.b)$$

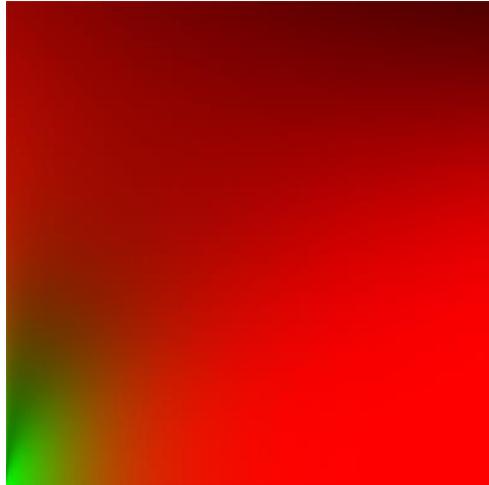
$$= \int_{\Omega} \frac{f_r(p, \omega_i, \omega_o)}{F(\omega_o, \mathbf{h})} (F_0 + (1 - F_0)(1 - \omega_o \cdot \mathbf{h})^5) (\mathbf{n} \cdot \omega_i) d\omega_i \quad (16.c)$$

$$= \int_{\Omega} \frac{f_r(p, \omega_i, \omega_o)}{F(\omega_o, \mathbf{h})} (F_0 \cdot (1 - \alpha) + \alpha) (\mathbf{n} \cdot \omega_i) d\omega_i \quad (16.d)$$

$$\begin{aligned} &= \int_{\Omega} \frac{f_r(p, \omega_i, \omega_o)}{F(\omega_o, \mathbf{h})} (F_0 \cdot (1 - \alpha)) (\mathbf{n} \cdot \omega_i) d\omega_i + \\ &\quad \int_{\Omega} \frac{f_r(p, \omega_i, \omega_o)}{F(\omega_o, \mathbf{h})} (\alpha) (\mathbf{n} \cdot \omega_i) d\omega_i \end{aligned} \quad (16.e)$$

$$\begin{aligned} &= F_0 \int_{\Omega} f_r'(p, \omega_i, \omega_o) (1 - (1 - \omega_o \cdot \mathbf{h})^5) (\mathbf{n} \cdot \omega_i) d\omega_i + \\ &\quad \int_{\Omega} f_r'(p, \omega_i, \omega_o) (1 - \omega_o \cdot \mathbf{h})^5 (\mathbf{n} \cdot \omega_i) d\omega_i \end{aligned} \quad (16.)$$

BRDF jednadžba se konvoluira preko svojih ulaznih parametara: kut između vektora \mathbf{n} i ω_o te gruboća. Rezultat konvolucije spremamo u 2D poglednu teksturu (eng. *lookup texture*) koju nazivamo BRDF integracijska mapa (eng. *BRDF integration map*). Tu tekstuру (Slika 14) koristimo u našem PBR programu za sjenčanje da dobijemo konačni konvoluirani zrcalni rezultat. 2D koordinate teksture se koriste kao ulazni parametri (x je $\mathbf{n} \cdot \omega_o$, y je gruboća), dok su vrijednosti teksture težina (crvena boja) i pristranost (zelena boja) uz vrijednost bazne reflektivnosti. Ovakva reprezentacija BRDF je moguća zato jer su svi ulazi i izlazni parametri cijeli brojevi koji imaju vrijednosti u rasponu od 0.0 do 1.0. Za konvoluciju BRDF koristimo vektore uzorka koje smo dobili korištenjem značajnog uzorkovanja (eng. *importance sampling*).



Slika 14. Izračunata tekstura BRDF okoline.

4. Implementacija

Ovaj rad je na računalu implementiran u programskom jeziku C++ i koristeći grafičko programsko sučelje OpenGL, verzija 3.3. Programi za sjenčanje pisani su u programskom jeziku GLSL. Kako bi se implementacija ubrzala, korištene su već gotove programske knjižnice, kako bi se izbjeglo nepotrebno trošenje vremena na implementaciju nekih osnovnih funkcionalnosti.

Knjižnica GLFW je korištena za stvaranje OpenGL grafičkog konteksta, baratanje prozorima te hvatanje i obradu korisničkog unosa (tipkovnica, miš). Za kontrolu verzije OpenGL-a, korištena je knjižnica GLAD. Za potrebe vektorske i matrične matematike korištena je knjižnica GLM. Slike su učitavane s knjižnicom stb_image, dok su 3D modeli učitavani s knjižnicom Assimp. Dear ImGui je knjižnica korištena za implementaciju grafičkog korisničkog sučelja.

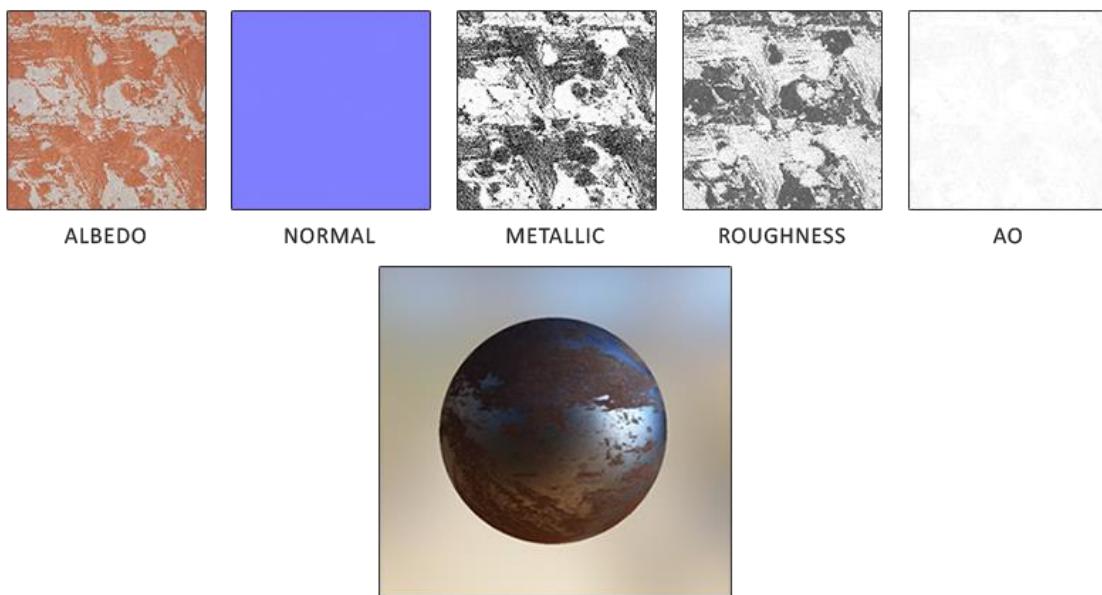
4.1. Korišteni model materijala

U prijašnjim poglavljima opisali smo matematički model iza fizikalno temeljenog osvjetljenja i osvjetljenja temeljenog na slikama. Vidjeli smo da izračun osvjetljenja uvelike ovisi o materijalu površine koju iscrtavamo. Svaki korišteni parametar materijala možemo prikazati kao 2D teksturu, što nam omogućava da kontroliramo svojstva površine na razini fragmenta. Postoje različiti modeli materijala koji se koriste u sustavima za crtanje fizikalno ispravnih slika. Ovaj rad koristi isti model materijala kao što je i Epic Games koristio u svojem radu (Slika 15). Umjetnici mogu stvarati teksture s vrijednostima koje su bazirane na fizikalnim svojstvima stvarnih materijala. Prednost ovakvog modela je što fizikalna svojstva površine ostaju nepromijenjena, neovisno o različitim okolinama i izvorima svjetlosti.

Prvi od korištenih parametara je albedo boja. Albedo specificira boju površine, ili baznu reflektivnost ako je površina metalna. Mapa raspršene boje je sličan parametar kao i mapa albedo boje, ali s razlikom da je iz albedo mape izbačena sva informacija o svjetlu (albedo je samo boja, bez ikakvih zatamnjenja ili sjena).

Mapa normala nam služi da na ravnoj površini prikažemo manje nepravilnosti. Geometrija površine se ne mijenja, ali zbog korištenja normale iz mape normala, dobivamo privid da je površina neravna.

Mapa metalnosti određuje da li se dio površine ponaša kao metal ili ne. Ovisno o pristupu, mapa metalnosti može imati ili samo crno-bijele vrijednosti (nešto ili je ili nije metal) ili vrijednosti u sivim tonovima (eng. grayscale). U drugom slučaju, govorimo o stupnju metalnosti površine.



Slika 15. Korišteni parametri materijala (gore), prikaz korištenog materijala (dolje).

Mapa gruboće određuje razinu gruboće površine. Iz prijašnjih poglavlja znamo da grublje površine imaju zamućenije refleksije, dok glađe površine imaju jasnije refleksije. Umjesto mape gruboće, mogli smo koristiti mapu glatkoće (neki umjetnici ju smatraju više intuitivnom za upotrebu od mape gruboće). Transformacija između tih mapa je izuzetno jednostavna ($\text{gruboća} = 1 - \text{glatkoća}$).

Zadnji parametar u našem modelu je mapa okluzije okoline (eng. *ambient occlusion*). Ova mapa određuje dijelove površine koji su zasjenjeni zbog nečega što se nalazi u okolini i u potpunosti prekriva neki dio površine.

4.2. Visoki raspon boja i gamma ispravak boja

Pri iscrtavanju rezultata fizikalno temeljenog osvjetljenja, koristimo visoki raspon boja (eng. *high dynamic range*, HDR) i gamma ispravak boja. HDR nam omogućava boje čije su vrijednosti veće od 1.0. Zamislimo scenu sa nekoliko jako svjetlih izvora svjetlosti. Nakon izračuna osvjetljenja, postoje fragmenti sa izračunatom vrijednosti boje koja je veća od 1.0. Ekrani mogu prikazivati samo boje čije su vrijednosti u rasponu između 0.0 i 1.0. Stoga, fragmenti s većom vrijednosti od 1.0 će biti ograničeni odozgora na 1.0, i mi ćemo izgubiti detalje u konačnoj slici koja se prikazuje na ekranu. Kako bismo mogli prikazivati HDR boje na ekranu, potrebno je obaviti mapiranje tonova (eng. *tone mapping*). Mapiranje tonova transformira HDR vrijednosti u raspon vrijednosti koje ekrani mogu iscrtati. Ovaj rad je koristio Reinhardovo mapiranje tonova (Jednadžba 17), najjednostavniji algoritam za mapiranja tonova.

$$mappedColor = hdrColor / (hdrColor + 1) \quad (17)$$

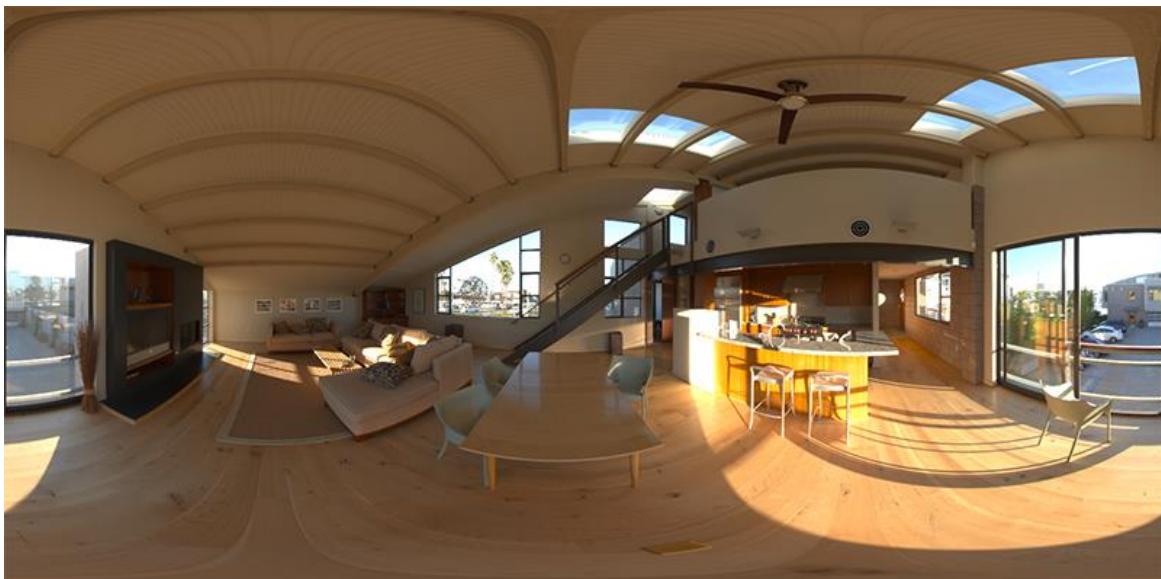
Kod prikaza boja na računalu, ekrani nelinearno mapiraju boje. Naime, ono što je fizikalno dvaput svjetlijе, ne izgleda dvaput svjetlijе ljudskom oku. Stoga, ekrani koriste relaciju potenciranja da prikažu boje (Jednadžba 18). Faktor potenciranja za većinu ekrana iznosi 2.2 i taj faktor zovemo gamma vrijednost ekrana. Ako želimo iscrtati fragment s bojom (0.5, 0.0, 0.0), ono što će ekran prikazati je fragment boje (0.218, 0.0, 0.0). Kako bismo na ekranu dobili željenu boju, potrebno je obaviti gamma ispravak boje (Jednadžba 19). Ideja gamma ispravka boje je primjena inverza gamma vrijednosti na boju koju želimo iscrtati, prije nego što se stvarno iscrtava na ekranu. Dakle, ako želimo prikazati (0.5, 0.0, 0.0) na ekranu, prvo obavimo gamma ispravak i dobijemo (0.73, 0.0, 0.0), nakon čega će ekran primijeniti gammu i iscrtati boju (0.5, 0.0, 0.0).

$$monitorColor = color^{\gamma} \quad (18)$$

$$correctedColor = color^{1/\gamma} \quad (19)$$

4.3. Mogućnosti razvijenog programa i rezultati

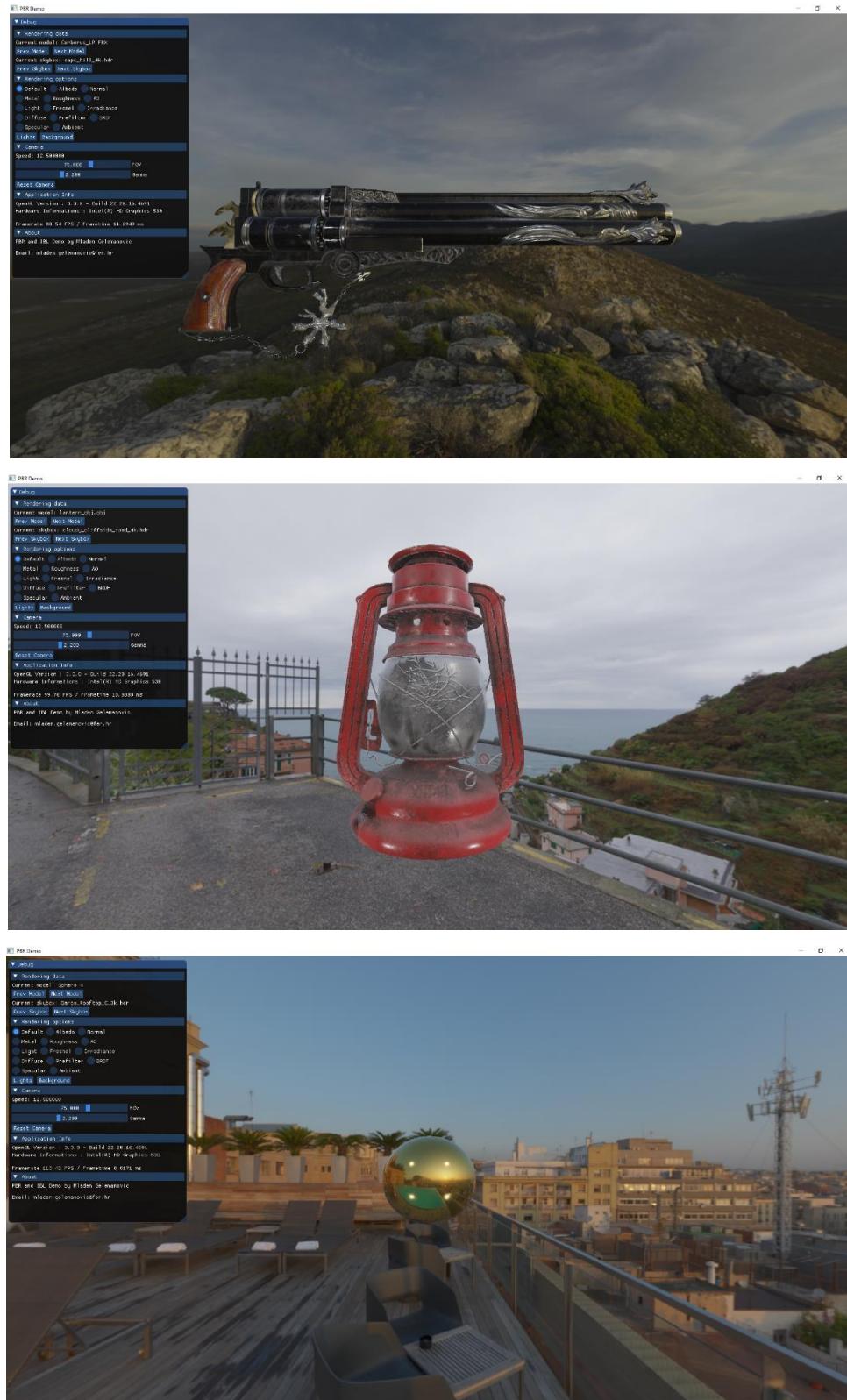
Razvijeni program podržava učitavanje proizvoljnih modela, materijala, izvora svjetlosti i okolina. Učitavanje se obavlja iz tekstualne datoteke, i jednom kad je program pokrenut, nije moguće učitati nove stvari u program. Treba napomenuti kako nije podržano korištenje više materijala za jedan model. Prilikom opisa modela koji želimo učitati, moguće je definirati translaciju, rotaciju i skaliranje modela. Kod modela koji sadrže više mreža geometrije (eng. *mesh*) omogućeno je skrivanje proizvoljnih mreža. Validni materijal mora imati sve potrebne parametre spremljene kao teksture i te teksture moraju biti točno imenovane (potrebna imena: *albedo*, *ao*, *metallic*, *normal*, *roughness*). Dodatno, sve teksture istog materijala moraju imati istu ekstenziju. Kod definiranja izvora svjetlosti za učitati, moguće je navesti poziciju i boju izvora svjetlosti. Okoline se učitavaju isključivo kao ravноправне kutne HDR mape (eng. *equirectangular map*, Slika 16). Za svaku učitanu okolinu se računa mapa zračenja i predfiltrirana mapa okoline.



Slika 16. Prikaz jedne od korištenih okolina.

Tijekom izvođenja, na ekranu se prikazuje jedan od učitanih modela (Slika 17) i jedna od učitanih okolina (prikaz okoline je moguće ugasiti po želji). Prikazani objekt će biti osvjetljen sa svim učitanim izvorima svjetlosti (učitani izvori svjetlosti se mogu ugasiti po želji). Trenutno prikazani objekt se može promijeniti pritiskom na tipku N ili preko korisničkog sučelja. Trenutno korištena okolina se može promijeniti

pritiskom na tipku B ili preko korisničkog sučelja. Program prekida s radom pritiskom na tipku Esc.

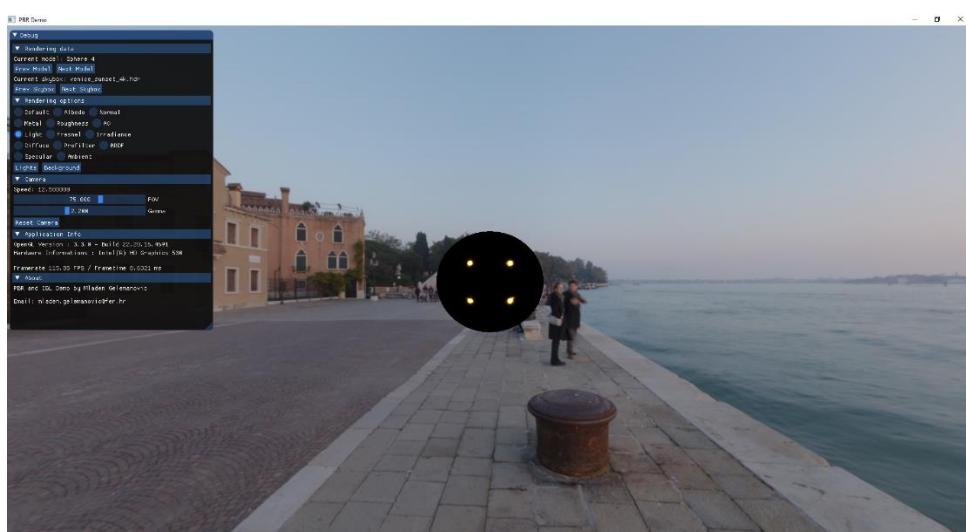
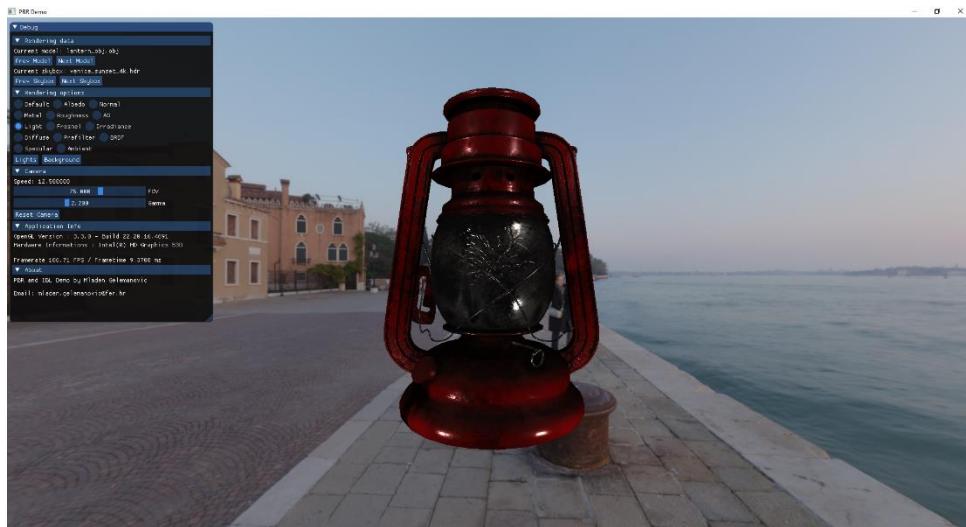


Slika 17. Prikazi različitih objekata u različitim okolinama.

Preko korisničkog sučelja omogućeni su različiti načini prikaza objekta: normalni prikaz (opcija *Default*), različiti parametri korištenog materijala (opcije *Albedo*, *Normal*, *Metal*, *Roughness*, *AO*) (Slika 18), prikaz utjecaja učitanih izvora svjetlosti (opcija *Light*) (Slika 19), prikaz Fresnelove refleksije korištene za IBL (opcija *Fresnel*) (Slika 20), prikaz mape zračenja (opcija *Irradiance*) (Slika 21) i raspršenog dijela IBL (opcija *Diffuse*), prikaz predfiltrirane mape okoline (opcija *Prefilter*) (Slika 22), prikaz okolinskog BRDF (opcija *BRDF*) (Slika 23), prikaz zrcalnog dijela IBL (opcija *Specular*) i prikaz čitavog IBL (opcija *Ambient*).



Slika 18. Prikaz parametara materijala (albedo i gruboća).

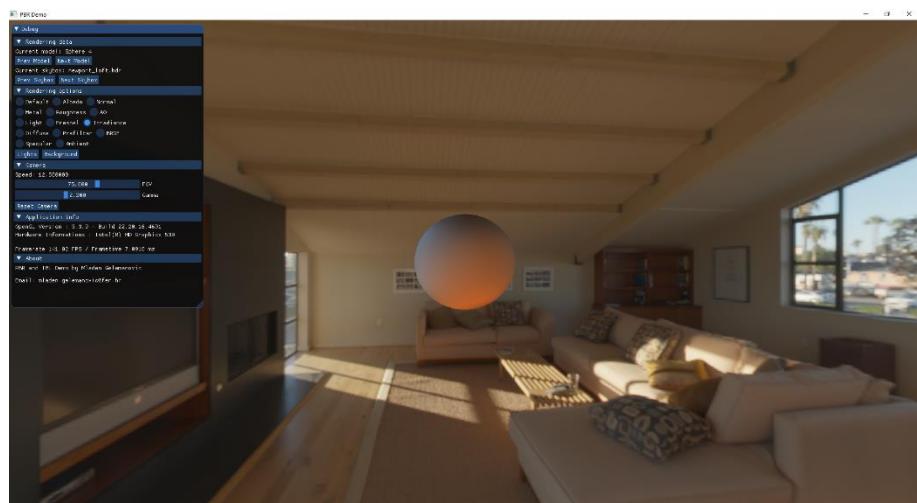


Slika 19. Prikaz utjecaja svjetla na razlicitim modelima.



Slika 20. Prikaz Fresnelove refleksije.

Implementirana kamera omogućuje kretanje oko objekta (tipke W/A/S/D), gledanje u proizvoljnom smjeru (pokreti miša), promjenu brzine kretanja kamere (kotačić miša) i promjenu vidnog polja (eng. *field of view*, FOV). Kameru je moguće vratiti u početnu poziciju i orientaciju. Pritiskom na tipku P, prestaje kontrola nad kamerom i pojavljuje se cursor. Ponovnim pritiskom na tipku P vraća se kontrola nad kamerom i ponovno se sakriva cursor. Preko korisničkog sučelja moguće je promijeniti gamma vrijednost.



Slika 21. Prikaz mape zračenja.



Slika 22. Prikaz predfiltrirane mape okoline.



Slika 23. Prikaz okolinskog BRDF iz različitih kuteva.

Korisničko sučelje također prikazuje korištenu verziju OpenGL-a, informacije o grafičkoj kartici te broj slika iscrtanih u jednoj sekundi i vrijeme (u milisekundama) potrebno za iscrtavanje jedne slike.

Razvijeni program nije računalno zahtjevan i trebao bi se moći pokretati na većini modernih računalnih sustava. Jedini uvjet za pokretanje programa je korištenje grafičke kartice koja podržava OpenGL verziju 3.3. Program ne koristi previše memorije (oko 100MB) i broj iscrtanih slika u sekundi ne opada drastično s povećanjem broja poligona koji se iscrtavaju. Jedino je pokretanje programa malo sporije, zbog učitavanja modela i tekstura te obavljanja potrebnih izračuna za osvjetljenje temeljeno na slikama.

Zaključak

Fizikalno temeljeno osvjetljenje (eng. *physically based rendering*, PBR) je moćan alat u izradi programa za crtanje. Korištenjem fizikalno temeljenog osvjetljenja u kombinaciji sa osvjetljenjem temeljenim na slikama (eng. *image based lighting*, IBL), moguće je postići izuzetno realistične rezultate. Nekoć skupi postupci se danas mogu koristiti u programima za crtanje u stvarnom vremenu, zahvaljujući većim grafičkim sposobnostima računala i brojnim metodama optimizacije. Fizikalno temeljeno osvjetljenje je u širokoj upotrebi danas, pogotovo u područjima izrade video igara i filmova.

U ovom radu dan je samo pregled jednog načina implementacije PBR i IBL sustava za crtanje. Rezultat rada je prikaz objekata koji izgledaju kao da se stvarno nalaze u toj okolini i osvjetljeni su sa pravim svjetлом. Osim realističnog prikaza, omogućen je i uvid u parametre materijala i međukorake postupka osvjetljenja.

Literatura

- [1] Burley, Brent, and Walt Disney Animation Studios. "Physically-based shading at disney." ACM SIGGRAPH. Vol. 2012. 2012.
- [2] Karis, Brian, and Epic Games. "Real shading in unreal engine 4." Proc. Physically Based Shading Theory Practice (2013).
- [3] Hoffman, Naty. "Background: physics and math of shading." Physically Based Shading in Theory and Practice 24.3 (2013): 211-223.
- [4] Russel, Jeff, "Basic theory of physically-based rendering", www.marmoset.co/posts/basic-theory-of-physically-basedrendering, pristupano 14.06.2018.
- [5] Schlick, Christophe. "An Inexpensive BRDF Model for Physically-based Rendering." Computer graphics forum. Vol. 13. No. 3. Blackwell Science Ltd, 1994.
- [6] Debevec, Paul. "Image-based lighting." ACM SIGGRAPH 2005

Fizikalno temeljen model osvjetljenja i osvjetljenje objekta temeljeno na slici

Sažetak

Navedene su fizikalne karakteristike stvarnog svjetla. Opisan je model mikro površina. Navedeni su matematički opisi Lambertovog modela za raspršeno osvjetljenje i Cook-Torranceov model za zrcalno osvjetljenje mikro površina. Opisane su funkcija distribucije normala, funkcija geometrijskog pokrivanja i funkcija Fresnelove refleksije. Objasnjen je postupak osvjetljenja temeljen na slikama. Ukratko su opisani najbitniji koraci implementacije. Prikazani su rezultati razvijenog modela osvjetljenja na različitim objektima u različitim okolinama.

Ključne riječi

OpenGL, C++, grafika, 3D, fizika svjetla, fizikalno temeljeno osvjetljenje, osvjetljenje temeljeno na slikama, funkcija dvosmjerne distribucije refleksije, BRDF

Physically Based Rendering and Image Based Lighting

Summary

Physical characteristics of real light are given. Model of microfacets is explained. Mathematical models for Lambertian diffuse shading model and Cook-Torrance microfacet specular shading model are given. Normal distribution function, geometric attenuation term and Fresnel reflectance term are described. Image based lighting is explained. Short overview of implementation. Results of rendering different objects in different environments are shown.

Key words

OpenGL, C++, graphics, 3D, physics of light, physically based rendering, image based lighting, bidirectional reflectance distribution function, BRDF