



DOPRINOS PBR PRISTUPA KVALITETU *gITF* MODELA

THE CONTRIBUTION OF PBR APPROACH TO A BETTER *gITF* MODEL RENDERING

Miloš Ribar, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – Opisi modernih koncepata iz računarske grafike, njihov doprinos u industriji i u samom izgledu scene koju grafički softver daje kao rezultat rada, koristeći moderne tehnike renderinga.

Ključne reči: Računarska grafika, svetlosne jednačine, šejderi, rendering baziran na fizici, BRDF, *gITF*

Abstract – Descriptions of modern computer graphics concepts, their contributions to the industry and the look of the scene that the graphics software renders.

Keywords: Computer graphics, light equations, shaders, physically based rendering, BRDF, *gITF*

1. UVOD

Pored filmske industrije, industrija video igara je jedna od najrazvijenijih grana industrije zabave u 21. veku. Budžeti za izradu proizvoda i jedne i druge grane industrije su multimilionski, i na njihovoj izradi je zaposleno više timova sa velikim brojem ljudi.

Glavni zahtev konzumenata multimedijalnog sadržaja je da on bude što realističniji i pruži konzumentima osećaj kao da su stvarno u filmskoj sceni koju gledaju, ili u video igri koju igraju. PBR je skup tehnika senčenja i renderovanja koja se bavi problemom realizma.

Ljudi su počeli ozbiljno da shvataju PBR još 1980. godine, kada se prvi put pojavio koncept rendering metode poznate kao *ray tracing*. Ovaj koncept je bio prvi put predstavljen od strane američkog računarskog naučnika Turner Viteda (eng. *J. Turner Whitted*) u njegovom naučnom radu: „Poboljšan model osvetljenja za osenčeni displej”.

Godine 1981. i 1982. pojavilo se unapređenje u PBR pristupu renderinga u vidu usavršenog refleksionog modela. Kuk-Torens (eng. *Cook-Torrance*) refleksioni model je uveo pojam mikrofasete i mikrofasetne refleksije u grafiku. Kompanija NVidia je početkom 2019-e godine izbacila novu seriju grafičkih kartica sa arhitekturom koja služi isključivo za *ray tracing* u realnom vremenu.

2. FOTOREALIZAM I RAY TRACING

Fotorealizam je proces stvaranja slike koristeći rendering mehanizme, tako da dobijena slika bude vrlo realistična.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivetić, red. prof.

Postoji više vrsta ray tracing algoritama i njihovih varijacija. *Light-based* algoritmi generišu zrake iz izvora svetlosti. Najčešći, *eye-based* funkcionišu po obrnutom principu, gde se zraci generišu iz kamere. Skoro svi fotorealistični rendering sistemi su zasnovani na korišćenju ray tracing algoritma i njegovih varijacija. Svaka varijacija algoritma mora da poštuje simuliranje skupa fenomena i prisutnost određenih objekata u sceni.

2.1. KAMERA

Krucijalni problem koji kamera treba da reši je da odredi koje će boje biti svaki piksel na slici. Zadatak kamere je da uzme jedan piksel na slici i da izgeneriše zrak koji prolazi kroz taj piksel. Zrak može, a ne mora da se sudari sa nekim objektom na svom putu, može se prelomiti kroz taj objekat i nastaviti svoju putanju, a može se i reflektovati od tog objekta. Ako se zrak sudari sa objektom, računa se incidentno svetlo (ukupno svetlo koje pada na površinu objekta) koje doprinosi boji posmatranog piksela. Zraci se generišu iz kamere, i oni se propagiraju po sceni.

2.2. PRESEK ZRAKA I OBJEKTA

Presek zraka sa objektom na sceni se mora precizno pronaći. Na tački preseka zraka sa objektom treba odrediti njegova svojstva u toj tački kao što su normala površine ili materijal od kog je taj objekat sačinjen. Svaki put kada kamera izgeneriše zrak, zadatak renderera je da odredi da li postoji presek sa objektom i gde se on tačno nalazi. Zbog ovoga, *ray tracing* algoritam najčešće traje $O(I \log N)$, gde je I broj piksela, a N broj objekata na sceni. Presek zraka i objekta se može naći pomoću kombinacije jednačine zraka $r(t) = o + td$, implicitne funkcije $F(x, y, z) = 0$ i ubacivanjem jednačine objekta u implicitnu funkciju koja sadrži nepoznatu t . Za sferu se dobija jednačina koja ima oblik kao jednačina ispod.

$$(o_x + td_x)^2 + (o_y + td_y)^2 + (o_z + td_z)^2 - r^2 = 0.$$

Ako ne postoje realni koreni jednačine, zrak se ne sudara sa sferom. Ako pak postoje, najmanji realni koren će biti tačka sudaranja zraka i objekta.

2.3. SVETLOSNI IZVORI I DISTRIBUCIJA SVETLOSTI

Presek zraka i objekta daje fragment koji treba osenčiti kao i svojstva objekta u toj tački. Potrebno je saznati kolika je količina svetlosti koja napušta fragment objekta. Ovo se može saznati tako što se uzmu u obzir svi svetlosni zraci koji padaju na fragment. Ovi proračuni

zahtevaju uvođenje radiometrijskih i geometrijskih formula distribucije svetlosti. Kod izračunavanja uticaja svetlosti na izgled objekta treba uzeti u obzir opadanje energije svetlosti sa rastojanjem svetlosnog izvora i objekta, i nagib objekta u odnosu na svetlosni izvor.

Snaga svetlosti koja pristiže na površinu objekta trpi prigušenje pri putovanju svetlosnog zraka kroz prostor. Ovo je opisano funkcijama (1) i (2).

$$f_{att} = 1/r^2 \quad (1)$$

$$f_{att} = 1/(k_c + k_l d + k_q d^2) \quad (2)$$

Ako je diferencijalna površina dA nagnuta za neki ugao θ u odnosu na normalu i zrak svetlosti, snaga kojom svetlost utiče na taj deo površine je skaliran koeficijentom $\cos\theta$.

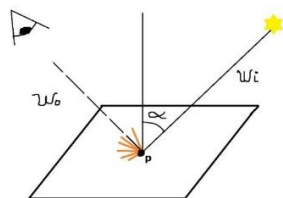
2.3. VIDLJIVOST

Između posmatranog objekta i svetlosnog zraka može da se nađe prepreka i potrebno je testirati da li je zrak pogodno prepreku i da li ta prepreka stvara senku na objektu.

Svaki svetlosni zrak iz izvora svetlosti doprinosi osvetljenju objekta samo ako je put od svetlosnog izvora do objekta neometan. Ovo se može proveriti tako što se napravi jedan zrak sa izvorom u tački na posmatranoj površini, sa pravcem i smerom ka svim izvorima svetlosti. Ovi zraci se zovu **zraci senke**. Parametar t koji se odnosi na neki presek se upoređi sa parametrom t zraka koji putuje ka svetlosnom izvoru. Ako svetlost nije blokirana na ovom putu, njen uticaj se uključuje u dalje proračune.

2.3. ODBIJANJE SVETLOSTI OD POVRŠINE OBJEKTA

Svaki objekat mora da sadrži opis kako se svetlost ponaša posle sudaranja sa njim. Najbitnije svojstvo je rasipanje incidentne svetlosti. Potreban podatak je količina svetlosne energije koja se odbija duž snopa svetlosti koji se sudara sa posmatranom tačkom na objektu. Ova situacija je prikazana slikom 1.



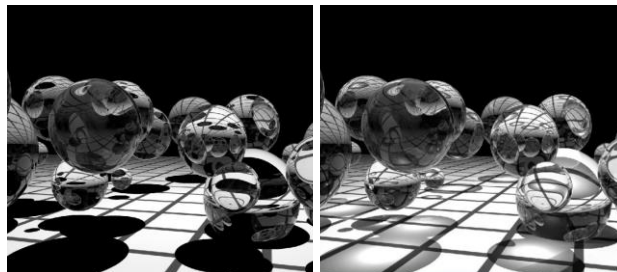
Slika 1: Rasipanje svetlosti posle udarca o tačku na površini objekta.

Potrebno je odrediti energiju zraka koji prati pravac kamere ω_0 . Funkcija koja opisuje raspršivanje svetlosti se naziva funkcija distribucije bidirekcionne refleksije (BRDF) koja se može zapisati u obliku $f_r(p, \omega_i, \omega_0)$.

2.3. INDIRECTAN TRANSPORT SVETLOSTI

Pošto svetlost može da stigne na površinu nakon odbijanja od neke druge objekte u sceni, važno je pratiti svaki zrak sve dok on ne izgubi svoju energiju. Ako zrak

iz kamere udara u ogledalo, dolazi do refleksije svetlosti u odnosu na normalu površine ogledala i upadnog ugla svetlosnog zraka. U ovoj situaciji se algoritam može pozvati rekurzivno. Ova tehnika se može iskoristiti i za transparentne objekte, gde se umesto reflektovanog zraka koristi prelomljeni. Svaka varijanta algoritma daje različite rezultate, gde se to najbolje vidi u senkama transparentnih objekata. Slika 2 predstavlja odnos između **photon mapping** i Vajtedovog **ray tracing** algoritma, gde **photon mapping** koristi **ray tracing** u drugom prolazu.



[1] Slika 2: Leva slika predstavlja Vajtedovu metodu **ray tracing** algoritma. Desna slika predstavlja **photon mapping** algoritam.

3. RENDERING BAZIRAN NA FIZICI

Rendering zasnovan na fizici je skup rendering tehnika koje su zasnovane na istoj teoriji, koje za cilj imaju simulaciju izgleda stvarnog sveta. Ovakvi rendering sistemi aproksimiraju ponašanje svetlosti u realnom svetu oslanjajući se na svetlosne fizičke formule iz oblasti radiologije, gde je glavni fokus stavljen na interakciju svetlosti sa raznim materijalima. Materijali nose sa sobom razne podatke u jednom fragmentu, o tome koliko su hrapavi, da li se u tom fragmentu nalaze udubljenja ili oštećenja, kakva je boja materijala i kakav je odsjaj u tom fragmentu. Postoji više vrsta tekstura koje doprinose realnosti izgleda objekata. Ove teksture prave umetnici.

Da bi jedan PBR svetlosni model bio baziran na fizici, mora da poštuje sledeće uslove:

1. Baziran je na modelu mikrofasetnih površina.
2. Mora poštovati zakone očuvanja energije.
3. Mora koristiti aproksimaciju BRDF jednačine kao rešenje svetlosne formule.

3.1. MIKROFASETNI MODEL

Svaki grafički objekat je presvučen nekim materijalom gde taj materijal može biti grub u većoj ili manjoj meri. Grubost materijala ogleda se u sitnim izbočinama na njegovoj površini. Te izbočine se ponašaju kao sićušna ogledala i nisu vidljive golim okom. Pogledati sliku 3. Haotičan raspored mikrofaseti označava jako grub materijal, a grubost utiče na haotičnu refleksiju svetlosnih zraka, što dovodi do većeg područja spekularne refleksije.



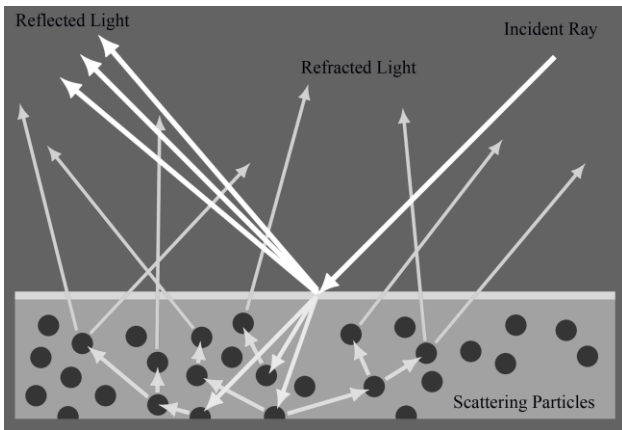
[2] Slika 3: Mikrofaseti na grubim i glatkim površinama.

Vrednost grubosti površine se opisuje statističkim parameterom grubosti (roughness). Na osnovu ovog parametra se može odrediti koliko je mikrofaseta orijentisano ka vektoru poluputanje \vec{h} .

3.2. OČUVANJE ENERGIJE

Tehnike senčenja zasnovane na fizici uključuju i zakon održanja energije. Jačina izlazeće svetlosti nikada ne sme da pređe jačinu ulazne svetlosti, i da bi neki sistem bio fizički baziran, ovo je važan uslov. Ovo se ne odnosi na objekte koji emituju svetlost.

Pretpostavka je da je svaka površina napravljena od dva sloja. Primer se može videti na slici 4. U momentu kada svetlosni zrak padne na površinu objekta, emitujuća svetlost objekta deli se na reflektovanu i refraktovanu svetlost. Refraktovana svetlost ulazi u međuprostor između dva sloja materijala i sudara se sa česticama koje čine materijal, gde potencijalno ostaje apsorbovana.



[2] Slika 4: Dvoslojna arhitektura materijala.

Očuvanje energije je obezbeđeno tako što se prvo izračuna uticaj spekularne refleksije, a potom se difuzna izvede iz spekularne.

3.3. FUNKCIJA DISTRIBUCIJE BIDIREKCIONE REFLEKSIJE

Aproksimira izgled neprozirne površine na osnovu svojstava materijala kojom je presvučena. Kao ulazne parametre uzima vektore ω_i, ω_0 i koeficijent grubosti mikrofaseta materijala a . Svaka BRDF funkcija mora da ima izlaz ne veći od 1.0.

Rendereri koji rade u realnom vremenu najčešće koriste **Cook-Torrance BRDF** koja je oblika u nastavku teskta. Varijabla c predstavlja albedo materijala, D predstavlja funkciju raspodele mikrofaseta, G je funkcija raspodele geometrijske okluzije, a F je Frenelova jednačina.

$$f_r = k_d f_{\text{ambert}} + k_s f_{\text{cook-torrance}}$$

$$f_{\text{ambert}} = \frac{c}{\pi DFG}$$

$$f_{\text{cook-torrance}} = \frac{c}{4(\omega_0 * n)(\omega_i * n)}$$

4. REALISTIČNO OSVETLJENJE U PBR ENDŽINIMA

Najznačajniji deo u prezentaciji jedne scene u video igrima i filmovima, predstavlja osvetljenje. Ono mora biti postavljeno tačno onako kako treba da ne bi došlo do lošeg mešanja boja i dobijanja nepregledne scene.

Primena PBR tehnika nije toliko komplikovana ako postoji nekoliko izvora svetlosti. Međutim, u realnosti, okruženje i objekti u okruženju predstavljaju dodatni izvor svetlosti. U tom slučaju, ne može se znati unapred koliko izvora svetlosti postoji na sceni. Tehnike osvetljenja koje rešavaju ovakav slučaj zovu se tehnike osvetljenja zasnovanog na slici ili *Image Based Lighting*.

IBL tehnike osvetljenja oslanjaju se na prostor u kom se objekti nalaze, gde je taj prostor ustvari mapa prostora (cubemap). Svaki tekstel mape okruženja može biti posmatran kao jedan izvor svetlosti. Da bi integral izračunao efikasnije, većinu stvari treba izračunati unapred. Rendering jednačina se jednostavno rastavi na spekularni i difuzni integral.

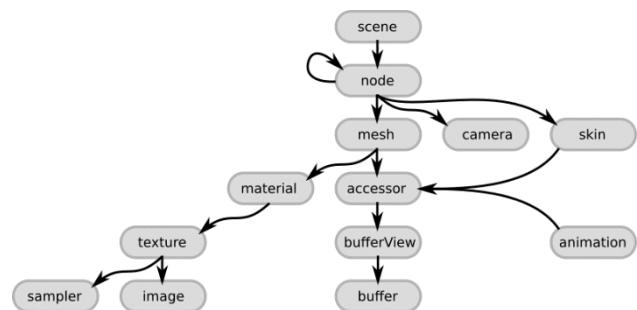
$$L_0(p, \omega_0) = k_d \frac{c}{\pi} \int_{\Omega} L_i(p, \omega_i) \cos\theta d\omega_i$$

$$L_0(p, \omega_0) = \int_{\Omega} \frac{DFG}{4(\omega_0 * n)(\omega_i * n)} L_i(p, \omega_i) \cos\theta d\omega_i$$

5. glTF FORMAT PODATAKA

Neobrađeni 3D podaci mogu doći u raznim formatima, kao što su .obj, .ply, .stl datoteke. Ovi formati datoteka ne sadrže informacije o tome kako bi jedan 3D model, ili čitava scena, trebala biti prikazana na ekranu. Struktura scene se mora analizirati, a geometrijski podaci se moraju pretvoriti u format koji grafička aplikacija podržava. svaka aplikacija mora da implementira svoj importer, loader i konvertere za sve formate datoteka koje će podržavati. Ovo je dosta veliki problem za programere.

Format podataka Khronos grupe, glTF pokušava da reši gore pomenuti problem tako što će standardizovati prikaz 3D sadržaja u klijentskim aplikacijama. On nije još jedan format podataka, već je definicija transmissionog podatka za 3D scene.



[3] Slika 5: Povezanost delova glTF objekta

6. VULKAN GRAFIČKA BIBLIOTEKA

Vulkan je retained biblioteka, koja sav posao oko kreiranja grafičkih komponenti ostavlja programeru. Najmanji detalj koji se tiče biblioteke i grafičkog pajplajna programer mora sam da kreira. Većina posla u korišćenju ove biblioteke svodi se na inicijalizaciju struktura i pozivanje različitih funkcija koje će komunicirati direktno sa drajverom, preko različitih slojeva biblioteke.

7. LITERATURA

- [1] Matt Pharr, Wenzel Jakob, Greg Humphreys, Physically Based Rendering: From Theory to Implementation, 2018
- [2] Jeremy Romanovski, Let's Get Physical: <https://www.jeremyromanowski.com/blog/2015/11/20/lets-get-physical-part-1-of-3>
- [3] glTF github repozitorijum <https://github.com/KhronosGroup/glTF>

Kratka biografija:



Miloš Ribar rođen je u Beogradu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Doprinos *PBR* pristupa kvalitetu *glTF* modela odbranio je 2019.god.
kontakt:
milosribar@yahoo.com