

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2909

VOLUMETRIJSKO PRAĆENJE PUTA

Marko Mijolović

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2909

VOLUMETRIJSKO PRAĆENJE PUTA

Marko Mijolović

Zagreb, lipanj 2022.

DIPLOMSKI ZADATAK br. 2909

Pristupnik: **Marko Mijolović (0036505756)**

Studij: Računarstvo

Profil: Računarska znanost

Mentorica: prof. dr. sc. Željka Mihajlović

Zadatak: **Volumetrijsko praćenje puta**

Opis zadatka:

Proučiti postupak praćenja puta. Proučiti model transportne jednadžbe svjetlosti u volumetrijskom okruženju. Razraditi postupak prikaza scene temeljen na metodi praćenja puta u volumetrijskom okruženju. Posebno obratiti pažnju na fizikalno osnove pri izračunu osvjettljenja. Programski ostvariti razrađeni postupak i na nizu scena prikazati rezultate. Načiniti testiranje na nizu primjera. Analizirati i ocijeniti ostvarene rezultate. Diskutirati upotrebljivost ostvarenih rezultata kao i moguća proširenja. Izraditi odgovarajući programski proizvod. Koristiti programski jezik C++ i okruženje OpenCL. 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.

Rok za predaju rada: 27. lipnja 2022.

Sadržaj

1.	Uvod	1
2.	Fizikalne osnove	2
2.1.	Opis problema	2
2.2.	Radiometrija	2
2.2.1.	Energija.....	3
2.2.2.	Tok zračenja	3
2.2.3.	Ozračenje	3
2.2.4.	Jakost zračenja.....	4
2.2.5.	Sjajnost	5
2.3.	Fotometrija	6
2.4.	Interakcija svjetlosti s površinama	7
2.4.1.	Dvosmjerna distribucija refleksivnosti.....	7
2.4.2.	Ispodpovršinsko raspršivanje (engl. <i>subsurface scattering</i>)	7
2.5.	Interakcija svjetlosti s česticama u prostoru	8
2.5.1.	Apsorpcija.....	8
2.5.2.	Emisija	9
2.5.3.	Raspršivanje	9
2.5.4.	Jednadžba transporta elektromagnetskog zračenja.....	11
3.	Algoritam praćenja puta	15
3.1.	Monte Carlo integracija	15
3.1.1.	Definicija	15
3.1.2.	Ruski rulet	16
3.1.3.	Uzorkovanje po važnosti	16
3.2.	Primjena Monte Carlo integracije na jednadžbu transporta svjetlosti.....	17
3.2.1.	Prvi slučaj: scena bez sudjelujućih medija	17

3.2.2.	Općeniti slučaj	20
4.	Implementacijski detalji	21
4.1.	Geometrijski objekti	21
4.2.	Akceleracijske strukture za izračun presjeka.....	21
4.2.1.	Hijerarhija omeđujućih volumena	22
4.3.	Materijali predmeta.....	24
4.3.1.	Terminologija	24
4.3.2.	Zrcalna refleksija i transmisija	26
4.3.3.	Lambertova refleksija	28
4.3.4.	Modeli refleksije temeljeni na teoriji malih djelića površine (engl. <i>microfacet theory</i>)	28
4.3.5.	Torrance-Sparrow model.....	33
4.4.	Sudjelujući mediji.....	34
4.4.1.	Fazne funkcije.....	34
4.4.2.	Tipovi sudjelujućih medija	36
4.5.	Paralelizacija izvođenja algoritma.....	36
5.	Rezultati.....	40
6.	Zaključak	43
	Literatura	44
	Sažetak.....	46
	Summary.....	47

1. Uvod

Jedan od glavnih zadataka računalne grafike je fotorealistično iscrtavanje scene simulacijom transporta svjetlosti kroz matematičke modele scene. U početku razvoja računalne grafike matematički modeli scene su bili vrlo jednostavni, a za simulaciju transporta svjetlosti su se većinom koristile fenomenološke metode koje nisu imale čvrstu podlogu u fizici. Razvojem sklopovlja matematički modeli scene su postajali sve kompleksniji te se za simulaciju transporta svjetlosti koriste sve složenije aproksimacije koje daju bolje rezultate. Istovremeno s razvojem sklopovlja, zbog široke uporabe računalne grafike, rasli su i zahtjevi koje su korisnici postavljali na grafičke aplikacije te je to dovelo do uspostave jedne cijele industrije koja ima za cilj fotorealistično iscrtavanje raznih vizualnih fenomena. Sve je to dovelo do razvoja i široke primjene algoritma praćenja puta koji danas predstavlja zlatni standard u filmskoj industriji, a u zadnje vrijeme se počeo koristiti i u aplikacijama koje se izvršavaju u stvarnom vremenu zahvaljujući Nvidia RTX platformi [1].

Cilj ovog rada je istražiti i implementirati metodu volumetrijskog praćenja puta, varijantu metode praćenja puta koja stavlja naglasak na volumetrijske učinke u sceni. Implementacija će biti napravljena u jeziku C++ i koristit će programski model SYCL za iskorištavanje enormnog paralelizma kojeg pružaju moderni grafički procesori.

Rad je podijeljen u šest poglavlja. U drugom poglavlju bit će predstavljene fizikalne osnove metode volumetrijskog praćenja puta. U trećem poglavlju bit će predstavljene Monte Carlo metode te njihova primjena u algoritmu praćenja puta. U četvrtom poglavlju bit će izneseni odabrani implementacijski detalji. U petom poglavlju prikazat će se dobiveni rezultati. U šestom poglavlju bit će iznesen zaključak.

2. Fizikalne osnove

Ovo poglavlje se uglavnom temelji na [2].

2.1. Opis problema

Zadatak iscrtavanja je zapravo stvaranje dvodimenzionalne slike koju bi stvorila hipotetska kamera koja se nalazi u sceni. Za potrebe toga, potreban je matematički opis scene koji se sastoji od pozicije objekata u sceni i njihovih fizikalnih svojstava. Konačna slika se tada dobiva simulacijom fizikalnih zakona, tj. simulacijom transporta svjetlosti kroz scenu.

Postoji više različitih fizikalnih modela svjetlosti s različitim svojstvima. Najtočniji je model dan kvantnom elektrodinamikom koja daje relativistički kvantnomehanički opis elektromagnetizma [3]. Pojednostavljenjem tog modela, dobiva se model svjetlosti opisan elektromagnetskim poljima i Maxwellovim jednažbama. Daljnjim pojednostavljenjem, ako se ignoriraju učinci difrakcije i interferencije, dobiva se model geometrijske optike. Unutar njega postoje i daljnja pojednostavljenja kao što je ignoriranje učinaka polarizacije.

U računalnoj grafici za potrebe simulacije transporta svjetlosti se najčešće koristi relativno jednostavan model geometrijske optike. Razlog je taj da učinci kvantizacije, valnih fenomena i polarizacije nemaju veliki utjecaj na konačnu sliku. Korišteni model pretpostavlja nekoherentnu, nepolariziranu svjetlost koja putuje pravocrtno dok se ne dogodi događaj raspršivanja (engl. *scattering*). Moguće su dvije vrste raspršivanja: raspršivanje na površini objekta i raspršivanje u volumenu. Više o ovome će biti govora u nadolazećim odjeljcima.

2.2. Radiometrija

Radiometrija je grana optike koja se bavi mjerenjem svojstava elektromagnetskih valova [4]. U ovom odjeljku bit će opisane četiri radiometrijske veličine koje opisuju elektromagnetsko zračenje: tok zračenja (engl. *flux*), jakost zračenja (engl. *intensity*), sjajnost (engl. *radiance*) i ozračenje (engl. *irradiance*). Sve navedene veličine ovise o valnoj duljini te su definirane spektralnom distribucijom snage (engl. *spectral power distribution* – SPD) koja opisuje iznos mjerenja za svaku valnu duljinu.

2.2.1. Energija

Polazna točka u izvodu svih navedenih veličina je energija, mjerena u džulima (J). Izvori svjetlosti emitiraju fotone koji su opisani svojom valnom duljinom i energijom koja je dana izrazom (2.1).

$$Q = \frac{hc}{\lambda} \quad (2.1)$$

$c = 299,472,458 \text{ m/s}$ predstavlja brzinu svjetlosti u vakuumu, $h \approx 6.626 \times 10^{-34} \frac{\text{m}^2 \text{kg}}{\text{s}}$ je Planckova konstanta, a λ je valna duljina.

2.2.2. Tok zračenja

Mjerenja energije provode se kroz neko određeno vrijeme, no nas najviše zanima opis zračenja u nekom određenom trenutku. Tok zračenja predstavlja količinu energije po jedinici vremena te po definiciji odgovara općenitoj definiciji snage u fizici (izraz (2.2)).

$$\Phi = \lim_{\Delta t \rightarrow 0} \frac{\Delta Q}{\Delta t} = \frac{dQ}{dt} \quad (2.2)$$

Mjerna jedinica toka zračenja je džul po sekundi ($\frac{J}{s}$), tj. vat (W).

Ako je poznat tok zračenja, onda je energija emitirana tijekom intervala zadanog vremenskim trenucima t_0 i t_1 dana izrazom (2.3).

$$Q = \int_{t_0}^{t_1} \Phi(t) dt \quad (2.3)$$

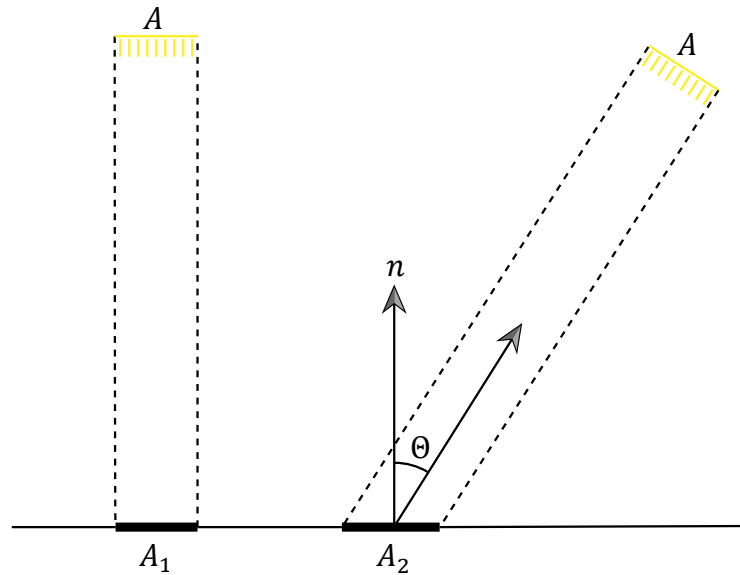
2.2.3. Ozračenje

Ozračenje opisuje koliko je neka površina ozračena elektromagnetskim zračenjem te je opisana izrazom (2.4) gdje je A površina, a p točka na površini.

$$E(p) = \frac{d\Phi(p)}{dA} \quad (2.4)$$

2.2.3.1 Primjer: Lambertov zakon

Iz jednadžbe ozračenja slijedi Lambertov zakon: količina energije koja dolazi na površinu je proporcionalna s kosinusom kuta između normale na površinu i smjerom izvora svjetlosti. Na slici Slika 2.1 je prikazan izvor svjetlosti s površinom iznosa A i tokom zračenja Φ koji obasjava površinu. Ako se izvor svjetlosti nalazi pod kutom od 90° (lijeva strana slike), onda je iznos površine A_1 jednak A . Tada ozračenje u bilo kojoj točki na A_1 iznosi $E_1 = \frac{\Phi}{A}$. Na desnoj strani slike je prikazan općeniti slučaj kada normala površine zatvara kut θ s vektorom čije se ishodište nalazi na površini, a pokazuje u smjeru izvora svjetlosti. Tada ozračenje u bilo kojoj točki na A_2 , uz pretpostavku da je A malog iznosa, iznosi $E_2 = \frac{\Phi(p) \cos \theta}{A}$.



Slika 2.1 Lambertov zakon

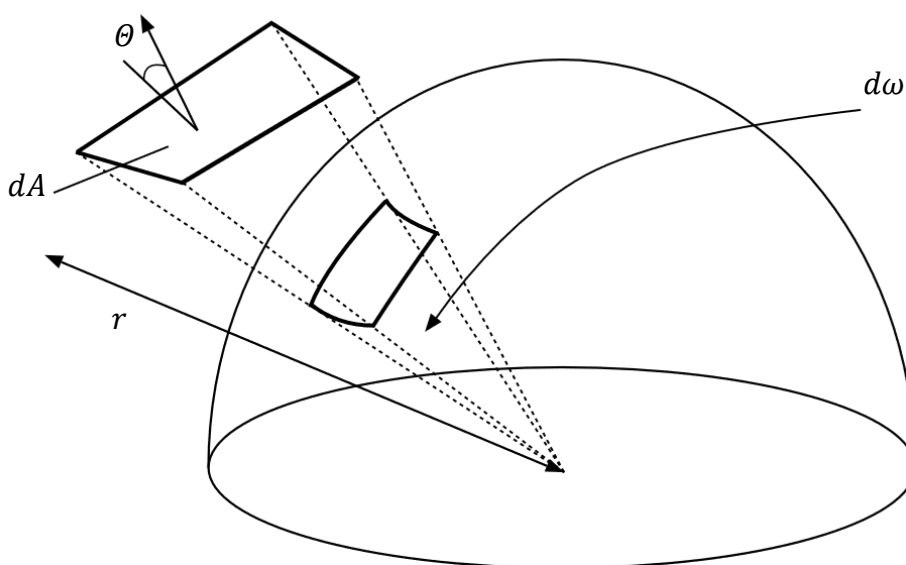
2.2.4. Jakost zračenja

Prije definicije jakosti zračenja, potrebno je definirati pojam prostornog kuta.

Prostorni kut je fizikalna veličina koja opisuje količinu vidnog polja koju neki objekt pokriva s obzirom na neku poziciju. Mjerna jedinica prostornog kuta je steradian (sr). Definicija prostornog kuta je dana izrazom (2.5).

$$d\omega = \frac{dA \cos \theta}{r^2} \quad (2.5)$$

$d\omega$ je iznos diferencijalnog prostornog kuta, dA je promatrana diferencijalna površina, θ je kut između normale na promatranu površinu i vektora od središta referentne kugle do promatrane točke na površini, a r udaljenost do površine (Slika 2.2). Na primjer, prostorni kut koji zatvara cijela kugla iznosi 4π sr, a polukugla 2π sr.



Slika 2.2 Definicija prostornog kuta

Sada možemo definirati jakost zračenja. Jakost zračenja predstavlja kutnu gustoću snage koju emitira (infinitesimalno mali) izvor svjetlosti te je definirana izrazom (2.6).

$$I = \frac{d\Phi}{d\omega} \quad (2.6)$$

2.2.5. Sjajnost

Sada napokon možemo definirati i najvažniju radiometrijsku veličinu – sjajnost. Sjajnost predstavlja mjeru ozračenja s obzirom na prostorni kut i definirana je izrazom (2.7).

$$L(p, \omega) = \frac{dE_{\omega}(p)}{d\omega} \quad (2.7)$$

E_{ω} predstavlja ozračenje okomito na smjer ω . Razlog zašto je sjajnost veoma važna u računalnoj grafici i općenito radiometriji je taj da je ona konstantna s obzirom na zrake svjetlosti u vakuumu.

2.3. Fotometrija

Fotometrija je također grana optike koja je srodna radiometriji. Razlika između njih je ta da se fotometrija fokusira samo na zračenje koje ljudi vide, dok se radiometrija odnosi na cijeli spektar elektromagnetskog zračenja.

Sve radiometrijske veličine imaju svoje fotometrijske veličine koje se dobiju tako da se radiometrijske veličine integriraju s obzirom na spektralnu odzivnu krivulju (engl. *spectral response curve*) $V(\lambda)$ koja opisuje relativnu osjetljivost ljudskog oka na različite valne duljine elektromagnetskog zračenja.

Na primjer, osvjetljenje (engl. *luminance*) je fotometrijska veličina koja mjeri koliko svijetlo SPD izgleda ljudima. Ona uzima u obzir npr. da ljudima SPD s određenom količinom energije u području zelenih valnih duljina izgleda svjetlije od SPD-a s istom količinom energije u području plavih valnih duljina. Osvjetljenje je definirano izrazom (2.8).

$$Y = \int_{\lambda} L(\lambda)V(\lambda)d\lambda \quad (2.8)$$

Mjerna jedinica za osvjetljenje je kandela po metru kvadratnom ($\frac{cd}{m^2}$).

2.4. Interakcija svjetlosti s površinama

Kada zraka svjetlosti obasja površinu, dio te svjetlosti se reflektira, dio se apsorbira, a dio se transmitira.

2.4.1. Dvosmjerna distribucija refleksivnosti

Distribucija reflektirane svjetlosti je opisana funkcijom dvosmjerne distribucije refleksivnosti (engl. *bidirectional reflectance distribution function* – BRDF). BRDF je definiran kao omjer izlazne sjajnosti i ulaznog ozračenja (izraz (2.9)).

$$f_r(p, \omega_o, \omega_i) = \frac{dL_o(p, \omega_o)}{dE(p, \omega_i)} = \frac{dL_o(p, \omega_o)}{L_i(p, \omega_i) \cos(\theta_i) d\omega_i} \quad (2.9)$$

BRDF-ovi imaju dva važna svojstva:

- Recipročnost: $\forall \omega_i, \omega_o, \quad f_r(p, \omega_i, \omega_o) = f_r(p, \omega_o, \omega_i)$
- Očuvanje energije: ukupna reflektirana energija je manja ili jednaka ukupnoj upadnoj energiji, tj: $\forall \omega_o, \quad \int_{H^2(n)} f_r(p, \omega_o, \omega') \cos(\theta') d\omega' \leq 1$

Distribucija transmitirane svjetlosti je slično opisana funkcijom dvosmjerne distribucije transmisivnosti (engl. *bidirectional transmittance distribution function* – BTDF). Zbog jednostavnosti, često se BTDF i BRDF kombiniraju u jednu funkciju zvanu dvosmjerna distribucija raspršivanja (engl. *bidirectional scattering distribution function* – BSDF) koja se označava s $f(p, \omega_o, \omega_i)$.

Koristeći navedene definicije, izlazna sjajnost u smjeru ω_o zbog upadne svjetlosti u točki p iz svih mogućih smjerova je definirana izrazom (2.10).

$$L_o(p, \omega_o) = \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos \theta_i d\omega_i \quad (2.10)$$

Ovaj izraz je ključan za iscrtavanje te je jedan od glavnih zadataka sustava za iscrtavanje računanje ovog integrala na raznim točkama u sceni.

2.4.2. Ispodpovršinsko raspršivanje (engl. *subsurface scattering*)

Neke površine se ne mogu opisati samo BRDF-om i BTDF-om. To su površine kod kojih je veoma istaknuto ispodpovršinsko raspršivanje svjetlosti. To je fenomen koji se očituje u tome da zrake svjetlosti koje ulaze u površinu u jednoj točki izlaze iz površine u točki koja

je dosta udaljena od ulazne točke. Primjeri takvih površina su ljudska koža, vosak i neke tekućine.

Takve površine se opisuju BSSRDF-om (engl. *bidirectional scattering-surface reflectance distribution function*). BSSRDF je definiran kao omjer izlazne sjajnosti i ulaznog toka zračenja (izraz (2.11)).

$$S(p_o, \omega_o, p_i, \omega_i) = \frac{dL_o(p_o, \omega_o)}{d\Phi(p_i, \omega_i)} \quad (2.11)$$

U ovom slučaju pri računanju izlazne sjajnosti u smjeru ω_o zbog upadne svjetlosti u točki p iz svih mogućih smjerova sada je potrebno integrirati i po površini (izraz (2.12)).

$$L_o(p_o, \omega_o) = \int_A \int_{H^2(n)} S(p_o, \omega_o, p_i, \omega_i) L_i(p_i, \omega_i) \cos\theta_i d\omega_i dA \quad (2.12)$$

2.5. Interakcija svjetlosti s česticama u prostoru

Do sada smo pretpostavili da zrake svjetlosti putuju kroz vakuum te da do raspršivanja svjetlosti dolazi samo na površinama čvrstih objekata. U stvarnosti dolazi do interakcije svjetlosti s česticama u prostoru, npr. s česticama koje čine maglu, česticama koje čine dim ili česticama koje čine Zemljinu atmosferu. Upravo je raspršivanje svjetlosti na česticama koje čine atmosferu zaslužno za to da je nebo plave boje te da su zalasci sunca crveni. Ovaj odjeljak donosi matematički opis raspršivanja svjetlosti na putu kroz sudjelujući medij (engl. *participating medium*) – medij koji sadrži veliki broj jako malih čestica.

Na distribuciju sjajnosti u prostoru sa sudjelujućim medijima utječu tri procesa: apsorpcija, emisija i raspršivanje.

2.5.1. Apsorpcija

Apsorpcija predstavlja smanjenje sjajnosti zbog pretvorbe energije zračenja u neki drugi oblik energije, npr. toplinu. Apsorpcija je opisana koeficijentom apsorpcije σ_a , koji predstavlja funkciju gustoće vjerojatnosti da se svjetlost apsorbira po jedinici puta proputovanog u sudjelujućem mediju. Općenito, koeficijent apsorpcije može ovisiti o poziciji u sudjelujućem mediju i o smjeru, no u praksi se najčešće koristi samo ovisnost o poziciji. Mjerna jedinica koeficijenta apsorpcije je $\frac{1}{m}$. Promjena sjajnosti zbog apsorpcije je opisana izrazom (2.13).

$$dL_o(p, \omega) = -\sigma_a(p, \omega)L_i(p, -\omega)dt \quad (2.13)$$

2.5.2. Emisija

Emisija predstavlja pojačanje sjajnosti zbog čestica koje emitiraju fotone. Promjena sjajnosti zbog emisije je opisana izrazom (2.14) gdje $L_e(p, \omega)$ označava emitiranu sjajnost dodanu zraci po jedinici puta, u točki p i smjeru ω .

$$dL_o(p, \omega) = L_e(p, \omega)dt \quad (2.14)$$

2.5.3. Raspršivanje

Na putu kroz sudjelujući medij, zraka svjetlosti se sudara s česticama i može biti raspršena u različite smjerove. Ako uzmemo u obzir neki konkretan smjer, tada prolaskom kroz sudjelujući medij sjajnost u tom smjeru se generalno mijenja zbog dva učinka: raspršivanja prema van i raspršivanja prema unutra.

Raspršivanje prema van odnosi se na smanjenje sjajnosti u pojedinom smjeru na putu kroz sudjelujući medij zbog raspršivanja svjetlosti u druge smjerove, a raspršivanje prema unutra odnosi se na povećanje sjajnosti u pojedinom smjeru zbog raspršivanja svjetlosti iz drugih smjerova u promatrani smjer.

2.5.3.1 Raspršivanje prema van i izumiranje

Raspršivanje prema van je opisano koeficijentom raspršivanja σ_s koji predstavlja funkciju gustoće vjerojatnosti da se dogodi raspršivanje prema van po jedinici puta. Kao i u slučaju apsorpcije, smanjenje sjajnosti zbog raspršivanja prema van je opisano izrazom (2.15).

$$dL_o(p, \omega) = -\sigma_s(p, \omega)L_i(p, -\omega)dt \quad (2.15)$$

Ukupna redukcija sjajnosti zbog učinaka raspršivanja prema van i apsorpcije određena je sumom $\sigma_a(p, \omega) + \sigma_s(p, \omega)$. Ovaj kombinirani učinak se naziva izumiranje (engl. *extinction*), a definiran je koeficijentom izumiranja $\sigma_t(p, \omega) = \sigma_a(p, \omega) + \sigma_s(p, \omega)$.

Dvije vrijednosti vezane uz koeficijent izumiranja će nam biti korisne. Jedna od njih je albedo $\rho = \frac{\sigma_s}{\sigma_t}$. Albedo poprima vrijednosti između 0 i 1, a opisuje relativnu vjerojatnost raspršivanja u odnosu na apsorpciju. Druga od njih je prosječna duljina koju svjetlost propuše u mediju prije interakcije s česticom, a iznosi $\frac{1}{\sigma_t}$.

Ukupna redukcija sjajnosti je tada opisana izrazom (2.16).

$$dL_o(p, \omega) = -\sigma_t(p, \omega)L_i(p, -\omega)dt \quad (2.16)$$

Integracijom tog izraza dobivamo veličinu zvanu propusnost (engl. *transmittance*) koja daje postotak sjajnosti transmitiran između dvije točke. Propusnost je definirana izrazom (2.17).

$$T_r(p \rightarrow p') = e^{-\int_0^d \sigma_t(p+t\omega, \omega)dt} \quad (2.17)$$

$d = \|p - p'\|$ je udaljenost između točaka p i p' , a ω je jedinični vektor smjera između te dvije točke.

Uočimo kako je vrijednost propusnosti uvijek između 0 i 1. Također uočimo kako je $Tr(p \rightarrow p) = 1$ i u vakuumu (zbog $\sigma_t = 0$) je $Tr(p \rightarrow p') = 1, \forall p'$. Također, ako koeficijent izumiranja ne ovisi o smjeru (već samo o poziciji), vrijedi $Tr(p \rightarrow p') = Tr(p' \rightarrow p)$. Također uočimo da vrijedi veoma važno svojstvo multiplikativnosti $Tr(p \rightarrow p'') = Tr(p \rightarrow p')Tr(p' \rightarrow p''), \forall p'$ između p i p'' . Ovo svojstvo je veoma važno u praktičnoj implementaciji volumetrijskog raspršivanja.

2.5.3.2 Raspršivanje prema unutra

Pod pretpostavkom da je udaljenost između čestica sudjelujućeg medija dovoljno velika (barem nekoliko puta veća od iznosa radijusa čestica), možemo ignorirati međusobne interakcije među česticama. Tada je kutna distribucija raspršene svjetlosti definirana faznom funkcijom $p(\omega, \omega')$.

Za sve fazne funkcije vrijedi svojstvo dano izrazom (2.18). To svojstvo zapravo govori da fazne funkcije definiraju gustoću vjerojatnosti za raspršivanje u određenom smjeru.

$$\int_{S^2} p(\omega, \omega')d\omega' = 1 \quad (2.18)$$

Povećanje sjajnosti po jedinici puta zbog raspršivanja prema unutra i emisije dano je izrazima (2.19) i (2.20).

$$dL_o(p, \omega) = L_s(p, \omega)dt \quad (2.19)$$

$$L_s(p, \omega) = L_e(p, \omega) + \sigma_s(p, \omega) \int_{S^2} p(p, \omega_i, \omega)L_i(p, \omega_i)d\omega_i \quad (2.20)$$

2.5.4. Jednadžba transporta elektromagnetskog zračenja

Jednadžba transporta elektromagnetskog zračenja je temeljna jednadžba koja opisuje distribucija sjajnosti u sceni.

Izvest ćemo ju iz principa očuvanja energije. Iz njega slijedi da razlika između izlazne i ulazne energije u sustav odgovara razlici emitirane i apsorbirane energije. Isto načelo se može primijeniti i na snagu pa dobivamo izraz (2.21).

$$\Phi_o - \Phi_i = \Phi_e - \Phi_a \quad (2.21)$$

2.5.4.1 Jednostavni slučaj

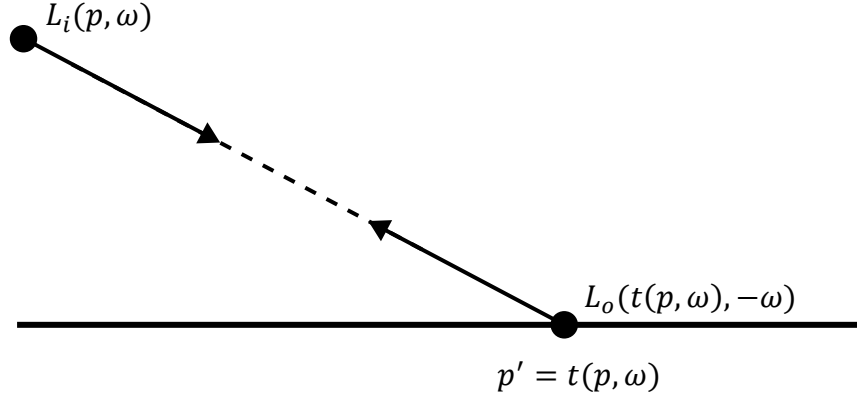
Za početak razmotrimo slučaj kada se scena sastoji samo od površina, bez sudjelujućih medija. Tada iz principa očuvanja energije slijedi izraz (2.22).

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) \cos(\theta_i) d\omega_i \quad (2.22)$$

U općenitom slučaju, jednadžbu transporta nije moguće riješiti analitički te će nam biti potrebne tehnike numeričke integracije koje će biti uvedene kasnije. Sada ćemo izvesti jednostavniju formulaciju jednadžbe transporta koja će nam omogućiti upotrebu numeričke integracije kasnije.

Iz razloga što smo pretpostavili da se scena ne sastoji od sudjelujućih medija možemo izvesti vezu između upadne sjajnosti u točki p i izlazne sjajnosti iz neke druge točke u sceni p' . Ako definiramo funkciju bacanja zrake (engl. *ray casting*) $t(p, \omega)$ (t je skraćenica za engl. *trace*) koja vraća točku p' na prvoj površini koju zraka (p, ω) presijeca (Slika 2.3), tada možemo upadnu sjajnost zapisati izrazom (2.23).

$$L_i(p, \omega) = L_o(t(p, \omega), -\omega) \quad (2.23)$$



Slika 2.3 Funkcija bacanja zrake

Koristeći navedenu funkciju, možemo jednadžbu transporta zapisati izrazom (2.24).

$$L(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i) L(t(p, \omega_i), -\omega_i) \cos(\theta_i) d\omega_i \quad (2.24)$$

Ova reprezentacija je još uvijek kompleksna iz razloga što je veza između geometrijskih objekata u sceni implicitno definirana funkcijom bacanja zrake. Pokušajmo eksplicitno zapisati tu vezu zamjenom integrala po smjerovima integralom po površinama.

Uvedimo oznaku za izlaznu sjajnost iz točke p' u točku p (izraz (2.25)).

$$L(p' \rightarrow p) = L(p', \omega), \quad \omega = \widehat{p - p'} \quad (2.25)$$

Tada BSDF u točki p' možemo zapisati kao u izrazu (2.26).

$$f(p'' \rightarrow p' \rightarrow p) = f(p', \omega_o, \omega_i), \quad \omega_i = \widehat{p'' - p'}, \quad \omega_o = \widehat{p - p'} \quad (2.26)$$

Sada možemo jednadžbu transporta zapisati pomoću integrala po površinama (izrazi (2.27), (2.28) i (2.29)).

$$L(p' \rightarrow p) = L_e(p' \rightarrow p) + \int_A f(p'' \rightarrow p' \rightarrow p) L(p'' \rightarrow p') G(p'' \leftrightarrow p') dA(p'') \quad (2.27)$$

$$G(p \leftrightarrow p') = V(p \leftrightarrow p') \frac{\cos\theta \cos\theta'}{\|p - p'\|^2} \quad (2.28)$$

$$V(p \leftrightarrow p') = \begin{cases} 1, & \text{ako je } p' \text{ vidljivo iz } p \text{ (i obrnuto)} \\ 0, & \text{inače} \end{cases} \quad (2.29)$$

Koristeći formulaciju s integralom po površinama, možemo izvesti formulaciju jednadžbe transfera kao zbroj po putevima tako da iterativno sjajnost pod integralom s desne strane

jednakosti zamjenjujemo s izrazom s lijeve strane. Tada integral možemo zapisati kao u izrazu (2.30).

$$L(p_1 \rightarrow p_0) = \sum_{n=1}^{\infty} P(\overline{p}_n) \quad (2.30)$$

Gdje $P(\overline{p}_n)$ označava iznos sjajnosti raspršene na putu s $n + 1$ točaka. $(\overline{p}_n = p_0, p_1, \dots, p_n)$, p_0 se nalazi u projekcijskoj ravnini, a p_n na izvoru svjetlosti i vrijedi izraz (2.31).

$$P(\overline{p}_n) = \int_A \int_A \dots \int_A L_e(p_n \rightarrow p_{n-1}) T(\overline{p}_n) dA(p_2) \dots dA(p_n) \quad (2.31)$$

Gdje je $T(\overline{p}_n)$ propusnost (engl. *throughput*) koja je definirana izrazom (2.32).

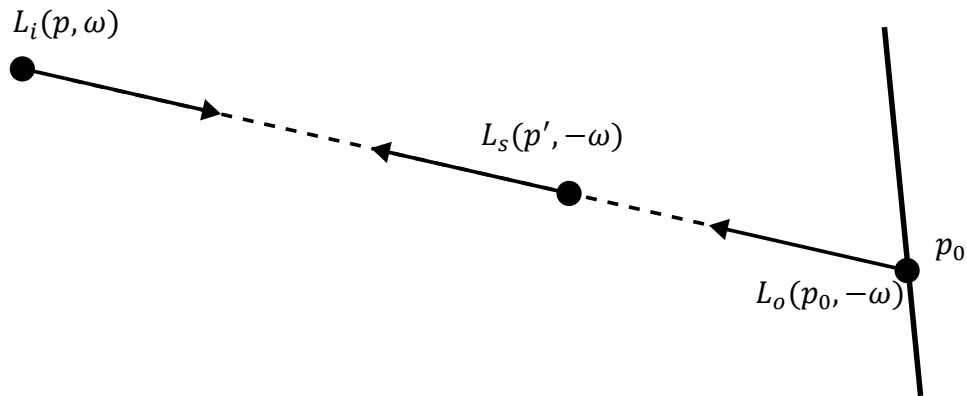
$$T(\overline{p}_n) = \prod_{i=1}^{n-1} f(p_{i+1} \rightarrow p_i \rightarrow p_{i-1}) G(p_{i+1} \leftrightarrow p_i) \quad (2.32)$$

2.5.4.2 Općeniti slučaj

U općenitom slučaju, u sceni postoje i površine i sudjelujući mediji. Tada iz principa očuvanja energije slijedi izraz (2.33).

$$L_i(p, \omega) = T_r(p_0 \rightarrow p_1) L_o(p_0, -\omega) + \int_0^t T_r(p' \rightarrow p) L_s(p', -\omega) dt' \quad (2.33)$$

$p_0 = p + t\omega$ je točka na površini, a $p' = p + t'\omega$ su točke na zraci (Slika 2.4).



Slika 2.4 Općeniti slučaj jednadžbe

Jednadžba opisuje dva faktora koji doprinose sjajnosti. Prvi je reflektirana i emitirana sjajnost s površine (faktor L_o) koja je modulirana s faktorom transmisivnosti zbog postojanja sudjelujućeg medija. Drugi faktor je sjajnost nadodana na točkama na zraci zbog emisije medija i raspršivanja sjajnosti prema unutra (faktor L_s).

Jednadžbu transporta elektromagnetskog zračenja je moguće i u općenitom slučaju raspisati kao zbroj po putevima koji sadrže različiti broj točaka. Formulacija i izvod zahtijevaju napredne matematičke alate te su zbog toga izostavljeni ovdje, no mogu se pronaći u [5].

3. Algoritam praćenja puta

Ovo poglavlje se uglavnom temelji na [2].

3.1. Monte Carlo integracija

3.1.1. Definicija

Standardne numeričke integracijske tehnike ne daju dobre rezultate na visoko dimenzionalnim integralima kao što su integrali koje moramo riješiti pri iscertavanju. Nasreću, Monte Carlo metode daju dosta dobre rezultate. One koriste nasumičnost kako bi izračunale vrijednost integrala bez obzira na njihovu dimenzionalnost.

Monte Carlo algoritmi spadaju u razred nasumičnih algoritama. Ono što je srodno svim tim algoritmima je da u sebi nekako koriste nasumičnost. Još jedan primjer nasumičnih algoritama su Las Vegas algoritmi. Razlika između Las Vegas i Monte Carlo algoritama je ta da Las Vegas uvijek daju isti rezultat na kraju (npr. uzimanje nasumičnog elementa kao *pivot* elementa u *quicksort* algoritmu), dok Monte Carlo algoritmi daju različite rezultate koji ovise o konkretnim nasumičnim brojevima korištenim tijekom izvođenja algoritma. Ipak, Monte Carlo algoritmi u prosjeku daju točno rješenje.

Monte Carlo integracija je algoritam koji računa integrale koristeći nasumične uzorke. Za izračunati integral Monte Carlo integracijom potrebno je samo moći evaluirati integrand u proizvoljnim točkama domene. Recimo da trebamo izračunati integral $\int_a^b f(x)dx$. Monte Carlo procjenitelj integrala onda je dan s $F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$, gdje su X_i točke nasumično odabrane s funkcijom gustoće vjerojatnosti $p(X_i)$. Iako se pokazuje da je očekivana vrijednost procjenitelja jednaka vrijednosti integrala, tj. $E[F_N] = \int_a^b f(x)dx$. Ova metoda se trivijalno proširuje i na višedimenzionalne integrale.

Pokazuje se da pogreška procjenitelja opada s kvadratnim korijenom broja uzetih uzoraka, $O(\sqrt{N})$. Iako standardne metode numeričke integracije konvergiraju brže kada je integral jednodimenzionalan, njihove performanse opadaju eksponencijalno povećanjem dimenzije integrala pa su upravo Monte Carlo metode idealne za integrale s kakvima se susrećemo u problemu iscertavanja.

3.1.2. Ruski rulet

Jedna metoda za ubrzavanje Monte Carlo integracije je metoda zvana ruski rulet. Ideja je veoma jednostavna. Promotrimo točke u domeni čija je vrijednost integranda 0. Za njih očito nije niti potrebno računati vrijednost integranda te ih možemo preskočiti. Ruski rulet nam omogućuje da preskočimo i one točke u domeni koje imaju relativno mali doprinos ukupnoj vrijednosti integrala. Za koristiti metodu ruskog ruleta potrebno je odabrati neku vrijednost vjerojatnosti q s kojom ćemo ignorirati uzorke. Znači integrand ćemo evaluirati s vjerojatnosti $1 - q$, ali ćemo ga skalirati s $\frac{1}{1-q}$ kako bi očekivana vrijednost Monte Carlo procjenitelja i dalje bila jednaka vrijednosti integrala (izrazi (3.1) i (3.2)).

$$F' = \begin{cases} \frac{F}{1-q}, & \xi > q \\ 0, & \text{inače} \end{cases} \quad (3.1)$$

$$E[F'] = (1-q) \left(\frac{E[F]}{1-q} \right) = E[F] \quad (3.2)$$

Ruski rulet gotovo uvijek povećava varijancu metode, no dosta povećava i efikasnost metode ako se uzme dobra vrijednost vjerojatnosti stopiranja.

3.1.3. Uzorkovanje po važnosti

Jedna metoda za smanjenje varijance je uzorkovanje po važnosti. Ona koristi činjenicu da Monte Carlo procjenitelj konvergira brže ako su uzorci uzeti iz distribucije $p(x)$ koja je slična $f(x)$. Intuitivno ima smisla da uzorke koji više doprinose ukupnoj vrijednosti integrala uzorkujemo s većom vjerojatnosti. Ipak, treba biti oprezan s ovom metodom jer ako se odabere loša funkcija gustoće vjerojatnosti, varijanca će se čak i povećati.

Promotrimo integrale oblika $\int f(x)g(x)dx$. Često znamo dobru funkciju gustoće vjerojatnosti za računanje $\int f(x)dx$ i $\int g(x)dx$ pa je pitanje možemo li nekako iskoristiti to znanje. Odgovor na to pitanje nam daje metoda višestrukog uzorkovanja po važnosti. Ona kombinira dvije funkcije gustoće vjerojatnosti p_f i p_g na način prikazan izrazom (3.3).

$$\frac{1}{n_f} \sum_{i=1}^{n_f} \frac{f(X_i)g(X_i)w_f(X_i)}{p_f(X_i)} + \frac{1}{n_g} \sum_{j=1}^{n_g} \frac{f(Y_j)g(Y_j)w_g(Y_j)}{p_g(Y_j)} \quad (3.3)$$

n_f je broj uzetih uzoraka iz distribucije p_f , n_g je broj uzetih uzoraka iz distribucije p_g , a w_f i w_g su težinske funkcije izabrane tako da je očekivana vrijednost ovog procjenitelja jednaka $\int f(x)g(x)dx$. Dobar izbor za težinske funkcije je balansirajuća heuristika dana izrazom (3.4).

$$w_s(x) = \frac{n_s p_s(x)}{\sum_i n_i p_i(x)} \quad (3.4)$$

3.2. Primjena Monte Carlo integracije na jednadžbu transporta svjetlosti

3.2.1. Prvi slučaj: scena bez sudjelujućih medija

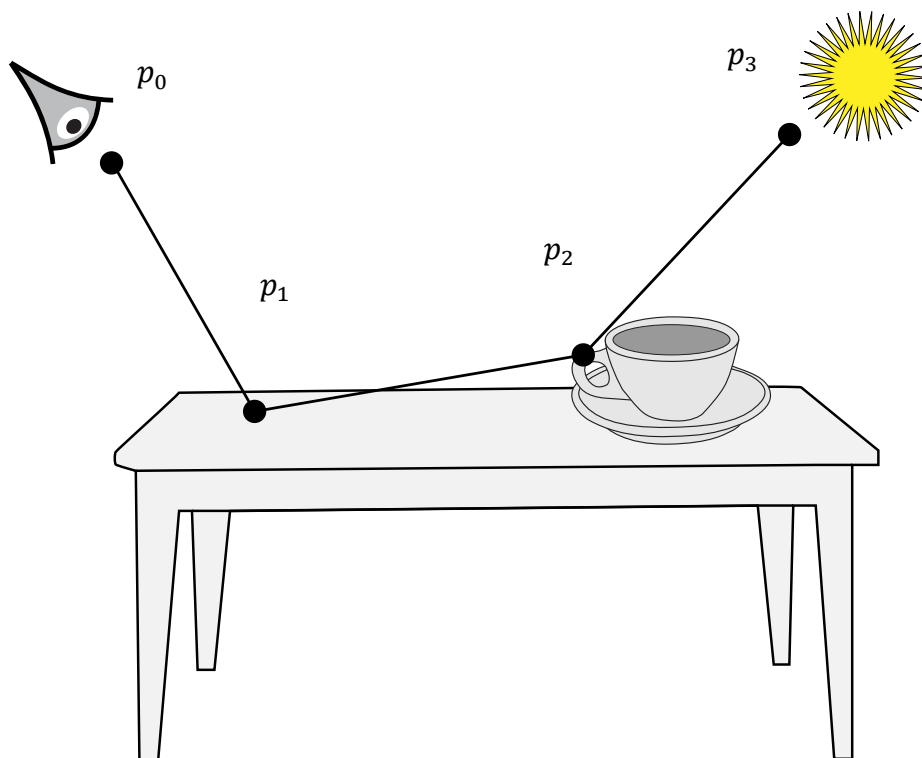
Sada kad smo uveli metodu Monte Carlo integracije možemo ju primijeniti na jednadžbu transporta svjetlosti koja je dana izrazom (2.30). Odmah primjećujemo dva problema:

- Kako procijeniti vrijednost sume beskonačnog broja pribrojnika s konačnim brojem izračuna?
- Ako uzmemo u obzir samo jedan pribrojnik $P(\bar{p}_l)$, kako generirati jedan ili više putova $\bar{p}_l$ kako bi izračunali Monte Carlo procjenitelj integrala?

Za fizikalno ispravne scene možemo iskoristiti činjenicu da putovi s više točaka na njima raspršuju više svjetlosti od putova s manje točaka (u prosjeku). Ovo slijedi iz činjenice da BSDF-ovi poštuju princip očuvanja energije. Zbog te činjenice možemo iskoristiti ruski rulet na način da prestanemo uzorkovati nakon izračuna prvih nekoliko $P(\bar{p}_l)$ pribrojnika. Primjerice, ako pri izračunu svakog pribrojnika sume s nekom vjerojatnosti odustanemo od daljnje evaluacije sume, nakon nekog vremena ćemo sigurno prestati s njenom evaluacijom. Na ovaj način dobit ćemo rezultat koji je nepristran (izraz (3.5)).

$$\frac{1}{1 - q_1} (P(\bar{p}_1)) + \frac{1}{1 - q_2} (P(\bar{p}_2)) + \frac{1}{1 - q_3} (P(\bar{p}_3)) + \dots \quad (3.5)$$

Time smo dali odgovor na prvo pitanje. Ostalo nam je još dati odgovor na to kako procijeniti doprinos pojedinog pribrojnika $P(\bar{p}_l)$. Za definirati put $\bar{p}_l$ potrebno je $l + 1$ točaka, gdje se zadnja točka p_l nalazi na izvoru svjetla, a prva točka p_0 je točka na projekcijskoj ravnini. Vidi sliku Slika 3.1 za primjer puta s četiri točke.



Slika 3.1 Primjer puta s četiri točke ($\overline{p_3}$)

Ako pogledamo izraz za $P(\overline{p}_i)$ vidimo da je to zapravo višestruki integral po površini objekata u sceni. Najjednostavniji način za uzorkovati točke p_i bi bio da svaka točka u sceni koja se nalazi na nekom objektu ima istu vjerojatnost uzorkovanja.

Problem s ovim načinom uzorkovanja je taj da mnogi generirani putove neće doprinosti rezultatu jer sadrže dvije susjedne točke koje nisu međusobno vidljive jer se između njih nalazi neki objekt. Postoji još jedan problem, a to je da ako u sceni postoje objekti koji se ponašaju kao savršeno zrcalo, vjerojatnost da ovaj način uzorkovanja točaka odabere točan smjer refleksije je 0 jer savršeno zrcalo reflektira svjetlost u samo jednom smjeru. Čak i ako objekt nije savršeno zrcalo, ali reflektira svjetlost primarno u jednom smjeru, većina putova generiranih na ovakav način će imati mali doprinos te će zbog toga varijanca Monte Carlo metode biti jako velika.

Oba ova problema se mogu riješiti tako da se put generira inkrementalno, točku po točku, počevši od točke na projekcijskoj ravnini. Na svakoj točki se uzorkuje BSDF kako bi se generirao novi smjer. Zatim se prati zraka u tom novom smjeru i pronalazi se najbliži presjek s nekim objektom u sceni, a točka presjeka uzima kao sljedeća točka na putu. Na ovaj način

efektivno generiramo uzorke koji će imati veliki doprinos konačnom rezultatu tako da pri svakoj točki generiramo smjerove koji imaju veliki lokalni doprinos.

Ovaj pristup uzorkuje BSDF-ove, no jednažba transporta svjetlosti koristi integrale po površini objekata u sceni, što znači da treba provesti transformaciju funkcije gustoće vjerojatnosti (izraz (3.6)).

$$p_A(p_i) = p_\omega \frac{|\cos\theta_i|}{\|p_{i+1} - p_i\|^2} \quad (3.6)$$

Također, faktor vidljivosti između p_i i p_{i+1} je trivijalno jednak 1 zbog načina na koji smo birali te točke. Efektivno, može se reći da algoritam praćenja puta radi uzorkovanje po važnosti s obzirom na komponentu vidljivosti. Ako iskoristimo opisanu metodu te za zadnju točku (koja se mora nalaziti na izvoru svjetlosti) koristimo neku funkciju gustoće vjerojatnosti koja definira razdiobu točaka na izvorima svjetlosti $p_A(p_i)$, iznos Monte Carlo procjenitelja dan je izrazom (3.7).

$$\frac{L_e(p_i \rightarrow p_{i-1})f(p_i \rightarrow p_{i-1} \rightarrow p_{i-2})G(p_i \leftrightarrow p_{i-1})}{p_A(p_i)} \prod_{j=1}^{i-2} \frac{f(p_{j+1} \rightarrow p_j \rightarrow p_{j-1})\cos\theta_j}{p_\omega(p_{j+1} - p_j)} \quad (3.7)$$

Pseudokod algoritma praćenja puta je prikazan u isječku Isječak 3.1. Primijetimo da se tijekom procjene doprinosa puta $P(\bar{p}_i)$ iskoriste točke prethodnog puta duljine $i - 1$ (sve osim zadnje točke koja se nalazi na izvoru svjetlosti). Zbog ovoga je potrebno pratiti samo jednu novu zraku, umjesto praćenja i zraka koliko bi trebalo da krećemo iz nule. Ponovno korištenje točaka na ovaj način uvodi korelaciju između svih pribrojnika $P(\bar{p}_i)$. Korelacija smanjuje kvalitetu rezultata, ali u praksi se često koristi jer pruža puno veću efikasnost od alternativa.

Isječak 3.1 Algoritam praćenja puta

```
Petlja {
    Izračunaj presjek zrake s objektima u sceni;
    Izađi iz petlje ako nije pronađen presjek;
    Uzorkuj osvjetljenje od izvora svjetlosti za izračun doprinosa
        trenutnog puta;
    Uzorkuj BSDF za pronalazak novog smjera puta;
    Primijeni ruski rulet (možda izađi iz petlje);
}
```

3.2.2. Općeniti slučaj

U prethodnom pododjeljku smo izveli algoritam praćenja puta za slučaj scene bez prisustva sudjelujućih medija. Pseudokod algoritma praćenja puta koji uzima u obzir sudjelujuće medije nije veoma različit te je prikazan u isječku Isječak 3.2.

Isječak 3.2 Algoritam volumetrijskog praćenja puta

```
Petlja {  
    Izračunaj presjek zrake s objektima u sceni;  
    Ako zraka prolazi kroz sudjelujući medij, uzorkuj ga;  
    Ako je pronađena interakcija s medijem {  
        Uzorkuj osvjetljenje od izvora svjetlosti za izračun doprinosa  
        trenutnog puta;  
        Uzorkuj faznu funkciju za pronalazak novog smjera puta;  
    } inače {  
        // Nastavi kao u slučaju bez sudjelujućih medija.  
        Izađi iz petlje ako nije pronađen presjek;  
        Uzorkuj osvjetljenje od izvora svjetlosti za izračun doprinosa  
        trenutnog puta;  
        Uzorkuj BSDF za pronalazak novog smjera puta;  
    }  
    Primijeni ruski rulet (možda izađi iz petlje);  
}
```

Primijetimo da se u pseudokodu prvo računa presjek zrake s objektima u sceni, a tek se onda uzorkuje sudjelujući medij. Ako se tada ustanovi da treba obratiti događaj raspršivanja u sudjelujućem mediju, onda nam je cijeli trud oko pronalaska presjeka s objektima u sceni postao beskoristan. U scenama kod kojih se ovo često događa, bilo bi dobro zamijeniti redoslijed te dvije operacije – prvo uzorkovati sudjelujući medij, a tek onda računati presjek zrake s objektima u sceni. Ovaj pristup bi doveo do dosta velikog ubrzavanja kada bi koristili prikladne akceleracijske strukture (one su objašnjene u kasnijem poglavlju).

4. Implementacijski detalji

Kompletna programska implementacija dostupna je na [6], a u ovom poglavlju su navedeni neki od najvažnijih implementacijskih detalja kao što su objašnjenje reprezentacije geometrijskih objekata u sceni, reprezentacije materijala predmeta, reprezentacije sudjelujućih medija itd.

4.1. Geometrijski objekti

U sustavu su dostupni razni geometrijski oblici kao što su kugle, trokuti i diskovi. Za sve navedene geometrijske oblike je potrebno definirati funkciju koja provjerava presijeca li dana zraka zadani geometrijski objekt i ako da, odrediti točku presjeka.

Uzmimo kuglu kao primjer. U prostoru objekta, kugla se nalazi u ishodištu te njena jednažba glasi $x^2 + y^2 + z^2 = r^2$, gdje je r radijus kugle. Neka je zadana zraka definirana ishodištem o i smjerom d , tj. točke na zraci su dane izrazom $o + td$. Uvrštavanjem tog izraza u jednažbu kugle dobivamo $(o_x + td_x)^2 + (o_y + td_y)^2 + (o_z + td_z)^2 = r^2$. Sve vrijednosti su nam poznate osim t što znači da možemo riješiti kvadratnu jednažbu. Jednažba može imati nula, jedno ili dva (realna) rješenja, ovisno o odnosu zrake i kugle.

Funkcije računanja presjeka sa zrakom za ostale geometrijske objekte su dane u raznim referencama, npr. [7].

4.2. Akceleracijske strukture za izračun presjeka

Ovaj odjeljak se većinom temelji na [2].

U praktičnoj implementaciji algoritma praćenja puta velika većina vremena se provede u izračunu najbliže točke presjeka s objektima u sceni. Naivan način za implementaciju ove operacije bi bio da se ispita presjek zrake sa svakim objektom u sceni te se odredi presjek koji je najbliži. Ta implementacija ima vremensku složenost $O(N)$ gdje je N broj primitiva u sceni. U većini scena ovaj način je daleko od optimalnog jer skoro uvijek zraka presijeca samo mali broj objekata u sceni.

Kako je ovo jako važan problem, mnogo istraživača je radilo na smišljanju akceleracijskih struktura koje bi ubrzale ovu operaciju. Cilj akceleracijskih struktura je omogućiti brzo

odbacivanje grupe geometrijskih primitiva (koje onda ne testiramo za presjek sa zrakom) i usmjeravanje pretraživanja tako da se bliži presjeci obrade prije tako da se objekti koje zraka također presijeca, ali se nalaze na većoj udaljenosti mogu odbaciti. Generalno postoje dva različita pristupa pri izradi akceleracijskih struktura: podjela po prostoru i podjela po objektima.

Akceleracijske strukture koje spadaju pod pristup podjele po prostoru dijele trodimenzionalan prostor na regije te vode evidenciju o tome koji objekti se nalaze u pojedinim regijama. U nekim algoritmima rezolucija podjele može biti adaptivna na način da se regije koje sadrže više objekata dijele na više novih regija. Kada je potrebno pronaći presjek zrake s objektima u sceni, pronađe se niz regija kroz koje zraka prolazi te se samo objekti u tim regijama testiraju za presjek.

S druge strane, akceleracijske strukture koje spadaju pod pristup podjele po objektima generalno kreću s početnim objektom kojeg onda dijele na sve manje i manje objekte. Primjerice, model sobe se može podijeliti na modele četiri zida, stropa, poda i stolice. Kada je potrebno pronaći presjek zrake s modelom sobe, provjeri se presijeca li zraka omeđujući volumen modela sobe (najčešće omeđujuća kutija poravnata s koordinatnim osima (engl. *axis aligned bounding box* - AABB)). Ako je odgovor negativan, ne treba se provjeravati presjek s pojedinim objektima od kojih se soba sastoji. U suprotnom slučaju, provjerava se presjek s omeđujućim volumenima svakog pojedinog objekta. Recimo da zraka presijeca omeđujući volumen modela stolice koji se dalje sastoji od manjih modela. Onda bi se testirao presjek s njihovim omeđujućim volumenima i to bi se rekurzivno ponavljalo sve dok ne bi došli do objekta koji se ne sastoji od još manjih objekata i tada odredili točnu točku presjeka ili utvrdili da presjek ne postoji.

Oba pristupa su veoma uspješna u rješavanju problema te nema nekog razloga uvijek preferirati jedan od njih, već njihove performanse ovise o konkretnim scenama nad kojima su primijenjeni.

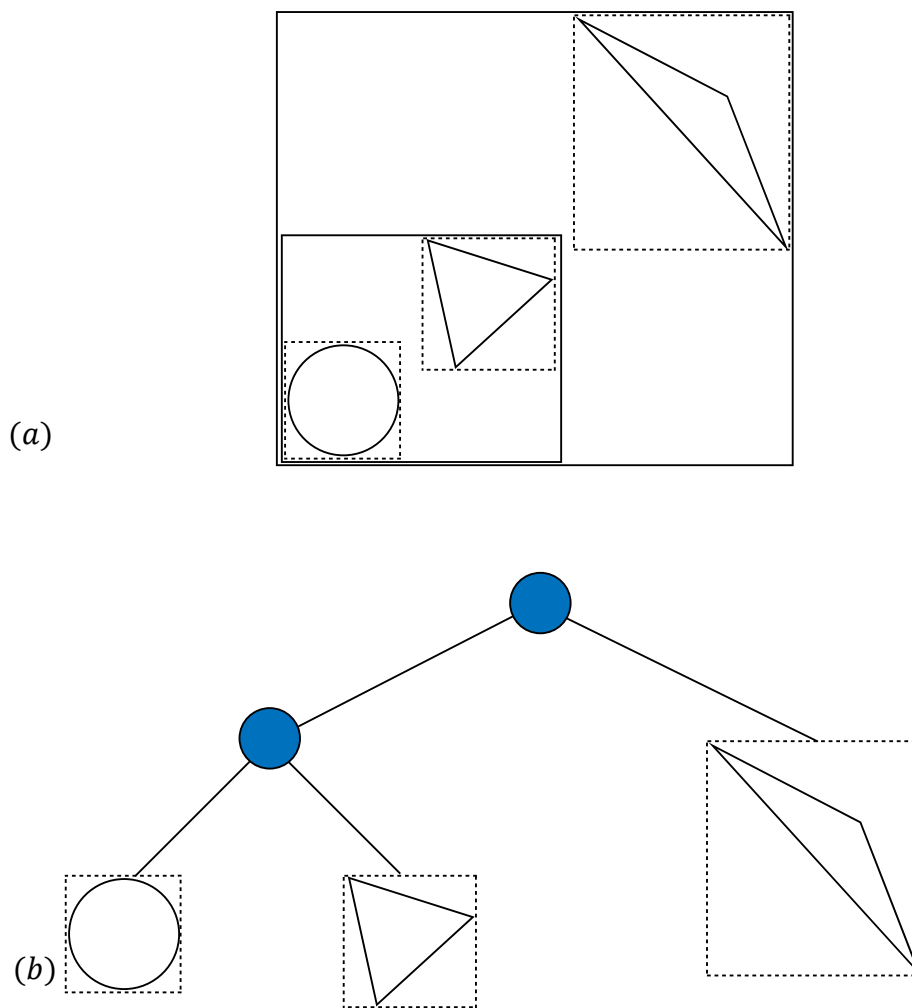
4.2.1. Hijerarhija omeđujućih volumena

Opišimo ovdje jednu akceleracijsku strukturu u širokoj upotrebi, a to je hijerarhija omeđujućih volumena (engl. *bounding volume hierarchy* – BVH).

Hijerarhija omeđujućih volumena se temelji na pristupu podjele po objektima gdje se scena particionira u hijerarhiju međusobno razdvojenih skupova. Geometrijski primitivi se

spremaju u listovima stabla, a svaki čvor koji nije list prema AABB koji omeđuje sve čvorove djece. Koristeći tu činjenicu, kada testiramo presjek zrake s čvorom u stablu i ustanovimo da zraka ne presijeca AABB čvora, možemo preskočiti ispitivanje cijelog podstabla definiranog tim čvorom.

Na slici Slika 4.1 prikazan je primjer hijerarhije omeđujućih volumena za jednostavnu scenu. Na dijelu slike (a) su prikazani geometrijski primitivi u sceni i njihovi AABB-ovi isprekidanom crtom. Primitivi se u ovom primjeru grupiraju po blizini pa se tako trokut i kružnica grupiraju u novi čvor koji sadrži AABB čvorova djece te se onda taj čvor grupira s čvorom trokuta u korijen hijerarhije. Na dijelu slike (b) je prikazana pripadajuća hijerarhija.



Slika 4.1 Primjer hijerarhije omeđujućih volumena

Jedno važno svojstvo ovog pristupa je da se svaki primitiv nalazi u samo jednom čvoru hijerarhije, što nije slučaj kod metoda temeljenim na podjeli po prostoru. Posljedica ovoga je da je memorija potrebna za BVH pristup ograničena jer je ukupan broj čvorova u hijerarhiji $2n - 1$ gdje je n broj primitiva.

Performanse BVH metode dosta ovise o odabranoj metodi particioniranja primitiva. Jedna metoda koja se često koristi je heuristika po površini koja je detaljno opisana u [2].

4.3. Materijali predmeta

U ovom odjeljku koji se većinom temelji na [2] bit će opisana reprezentacija materijala predmeta, tj. bit će definirana reprezentacija BRDF i BTDF modela.

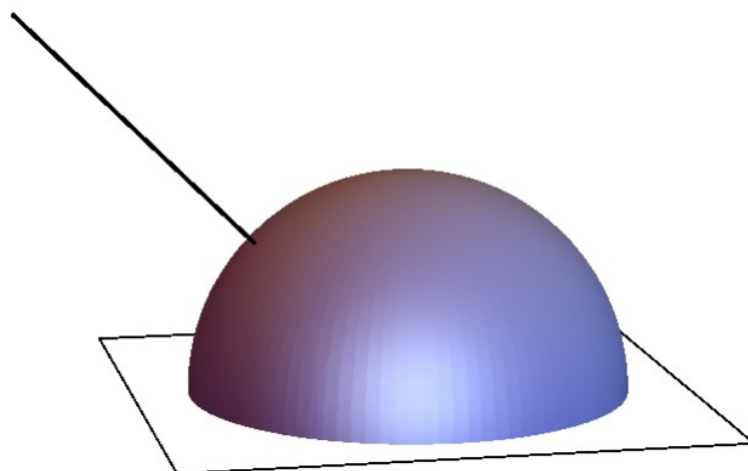
Modeli dolaze u nekoliko varijanta:

- Mjereni podaci – moguće je izmjeriti BRDF-ove i BTDF-ove u laboratoriju i koristiti te podatke izravno
- Fenomenološki modeli – matematički modeli koji nemaju nekakve fizikalne podloge, ali svejedno dosta dobro oponašaju fizikalno točne modele
- Simulacija – ako imamo detaljne informacije o površini, moguće je simulirati interakciju svjetlosti s površinom te onda koristiti te podatke na isti način kao mjerene podatke
- Modeli temeljeni na valnoj optici – ovi modeli proizlaze iz rješenja Maxwellovih jednadžbi, ali često zahtijevaju dosta računanja pa kao što je već bilo spomenuto ne koriste se tako često u računalnoj grafici
- Modeli temeljeni na geometrijskoj optici – ako su poznate detaljne informacije o površini ponekad je moguće izvesti BRDF i BTDF modele u zatvorenoj formi zbog pretpostavki koje geometrijska optika uvodi

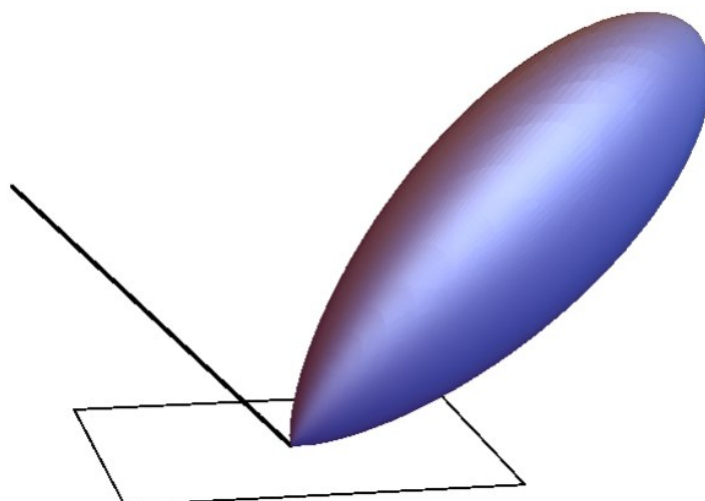
4.3.1. Terminologija

Refleksija svjetlosti na površinama se može podijeliti u četiri kategorije: difuzna, sjajno zrcalna (engl. *glossy specular*), savršeno zrcalna, i retroreflektivna.

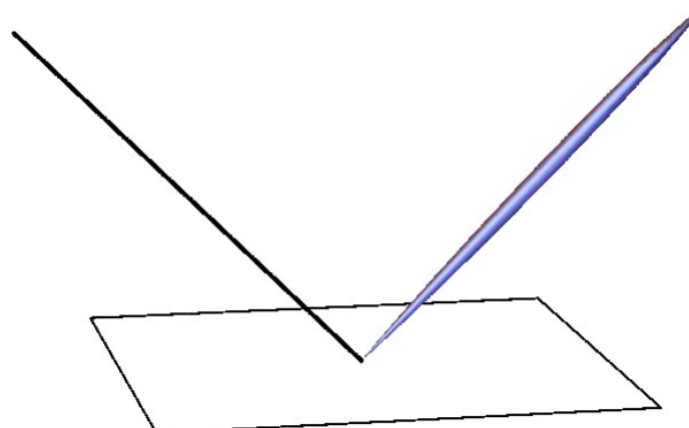
Difuzne površine jednako reflektiraju svjetlost u svim smjerovima (Slika 4.2). Primjer stvarne površine koja se ponaša jako slično difuznim površinama je primjerice mat boja. Sjajno zrcalna površina (npr. plastika) reflektira svjetlost primarno u uskom snopu smjerova (Slika 4.3). Savršeno zrcalna površina (npr. zrcalo) reflektira svjetlost samo u jednom smjeru (Slika 4.4). Retroreflektivna površina (npr. Mjesec) reflektira svjetlost u uskom snopu primarno prema nazad (Slika 4.5).



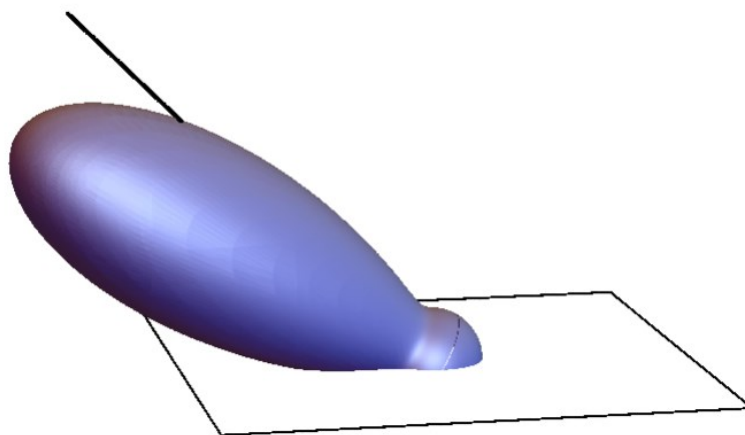
Slika 4.2 Difuzni BRDF



Slika 4.3 Sjajno zrcalni BRDF



Slika 4.4 Savršeno zrcalni BRDF



Slika 4.5 Retroreflektivni BRDF

4.3.2. Zrcalna refleksija i transmisija

Ponašanje svjetla pri interakciji sa savršenim zrcalom je veoma jednostavno opisati. Za dani ulazni smjer ω_i , sva svjetlost se raspršuje u samo jednom izlaznom smjeru ω_o . U slučaju refleksije taj smjer je smjer koji s normalom na površinu zatvara isti kut kao i ulazni smjer, tj. $\theta_i = \theta_o$ i $\phi_o = \phi_i + \pi$ gdje je θ polarni kut, a ϕ azimutalni kut.

U slučaju transmisije, opet vrijedi $\phi_o = \phi_i + \pi$, a θ_t je dan Snellovim zakonom. Snellov zakon povezuje kut θ_t koji čini transmitirani smjer s normalom na površinu i kut θ_i koji čini upadna zraka s normalom na površinu. Snellov zakon dan je izrazom (4.1).

$$\eta_i \sin \theta_i = \eta_t \sin \theta_t \quad (4.1)$$

η_i je indeks refrakcije medija kroz koji putuje upadna zraka, dok je η_t indeks refrakcije medija kroz koji putuje transmitirana zraka. Indeks refrakcije opisuje koliko sporije svjetlost putuje u određenom mediju u odnosu na brzinu svjetlosti u vakuumu. Indeks refrakcije u općenitom slučaju varira s obzirom na valnu duljinu svjetlosti što se može primijetiti kada bijela svjetlost prolazi kroz prizmu. Tada se upadna svjetlost razdijeli na komponente, fenomen zvan disperzija. U računalnoj grafici se uobičajeno ova ovisnost o valnoj duljini ignorira jer pojednostavljuje izračune, a nije toliko ključna za krajnji rezultat.

4.3.2.1 Fresnelove jednadžbe

Omjer reflektirane i transmitirane svjetlosti dan je Fresnelovim jednadžbama koje predstavljaju rješenje Maxwellovih jednadžba na glatkim površinama. Fresnelove jednadžbe daju omjer reflektirane i transmitirane svjetlosti za dva stanja polarizacije (paralelna i

okomita polarizacija). U ovom radu ćemo pretpostaviti da je svjetlost nepolarizirana, tj. da je nasumično orijentirana u odnosu na val svjetlosti. Tada je omjer dan srednjom vrijednosti kvadriranih izraza za okomitu i paralelnu polarizaciju.

U ovom trenutku valja napraviti podjelu materijala na dielektrike i na vodiče jer dielektrici imaju realan indeks refrakcije, dok vodiči imaju kompleksan indeks refrakcije. Obje vrste materijala su opisane istim Fresnelovim jednažbama, ali jednažbe poprimaju jednostavniju formu u slučaju dielektrika zbog realnih indeksa refrakcije.

Fresnelove jednažbe za dielektrike su opisane izrazima (4.2) i (4.3).

$$r_{\parallel} = \frac{\eta_t \cos \theta_i - \eta_i \cos \theta_t}{\eta_t \cos \theta_i + \eta_i \cos \theta_t} \quad (4.2)$$

$$r_{\perp} = \frac{\eta_i \cos \theta_i - \eta_t \cos \theta_t}{\eta_i \cos \theta_i + \eta_t \cos \theta_t} \quad (4.3)$$

$r_{\parallel}$ je Fresnelov koeficijent refleksija za paralelno polariziranu svjetlost, dok je $r_{\perp}$ Fresnelov koeficijent refleksije za okomito polariziranu svjetlost.

Za nepolariziranu svjetlost, Fresnelov koeficijent refleksije je dan izrazom (4.4).

$$F_r = \frac{1}{2} (r_{\parallel}^2 + r_{\perp}^2) \quad (4.4)$$

Zbog principa očuvanja energije, Fresnelov koeficijent transmisije iznosi $1 - F_r$. Fresnelove jednažbe za vodiče su ovdje izostavljene, a mogu se pronaći u [2].

4.3.2.2 Zrcalna refleksija

Sada možemo izvesti BRDF za zrcalnu refleksiju. S obzirom na definiciju Fresnelovih jednažbi, tražimo BRDF tako da vrijedi izraz (4.5).

$$L_o(\omega_o) = \int f_r(\omega_o, \omega_i) L_i(\omega_i) \cos \theta_i d\omega_i = F_r(\omega_r) L_i(\omega_r) \quad (4.5)$$

ω_r je zrcalni vektor refleksije za ω_o reflektiran oko normale na površinu.

Intuitivno, želimo da BRDF ima vrijednost 0 svugdje osim za reflektirani smjer što sugerira korištenje Dirac delta distribucije. Lako se pokazuje da je traženi BRDF definiran izrazom (4.6).

$$f_r(p, \omega_o, \omega_i) = \frac{F_r(\omega_r) \delta(\omega_i - \omega_r)}{\cos \theta_r} \quad (4.6)$$

4.3.2.3 Zrcalna transmisija

Izvod BTDF-a za zrcalnu transmisiju je dan u [2], a ovdje ćemo samo predstaviti krajnji rezultat. BTDF za zrcalnu transmisiju je definiran izrazom (4.7).

$$f_r(\omega_o, \omega_i) = \frac{\eta_o^2 (1 - F_r(\omega_i)) \delta(\omega_i - T(\omega_o, n))}{\eta_i^2 \cos \theta_i} \quad (4.7)$$

$T(\omega_o, n)$ je vektor zrcalne transmisije dobiven transmisijom ω_o kroz površinu s normalnom na površinu n . Faktor $1 - F_r(\omega_i)$ objašnjava lako uočljivu činjenicu – transmisija je jača kod skoro okomitih kutova. Na primjer, možemo vidjeti veoma duboko kada gledamo okomito u more, no kada gledamo u more pod malim kutom, većina svjetla se reflektira.

4.3.3. Lambertova refleksija

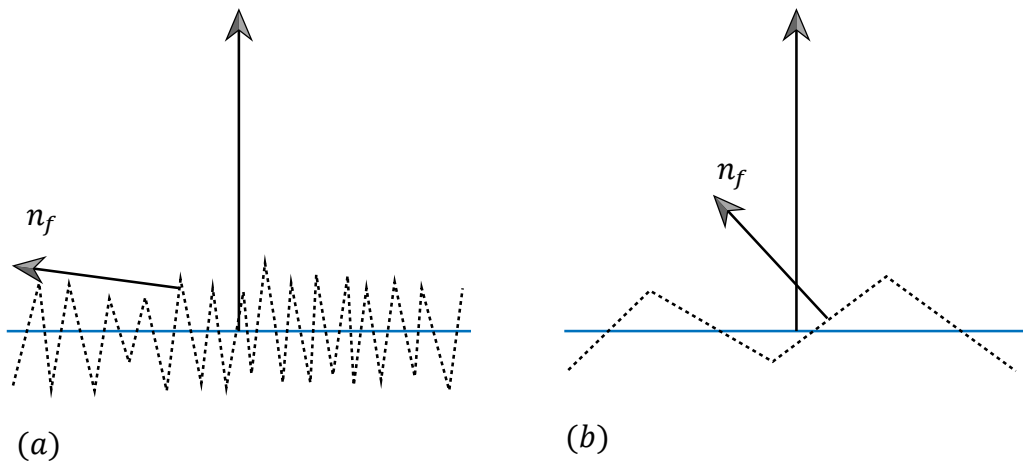
Lambertov BRDF model modelira savršeno difuznu površinu koja raspršuje svjetlost jednako u svim izlaznim smjerovima te predstavlja jedan od najjednostavnijih BRDF modela. Lambertov BRDF model je definiran izrazom (4.8).

$$f_r(\omega_o, \omega_i) = \frac{R}{\pi} \quad (4.8)$$

R je SPD koji opisuje omjer ulaznog svjetla koji se reflektira.

4.3.4. Modeli refleksije temeljeni na teoriji malih djelića površine (engl. *microfacet theory*)

Mnogi modeli refleksije se temelje na ideji da se grube površine modeliraju nakupinom malih djelića površine. Često se takve površine modeliraju kao visinska polja, a distribucija normala na male djeliće površine se modelira statistički. Slika 4.6 prikazuje poprečni presjek relativno grube površine ((a) dio slike) i dosta glađe površine ((b) dio slike). Na slici je s n_f označena normala na mali djelić površine. Primijetimo da gruba površina ima puno veću varijaciju u distribuciji normala na djeliće površine n_f u odnosu na glatku površinu.



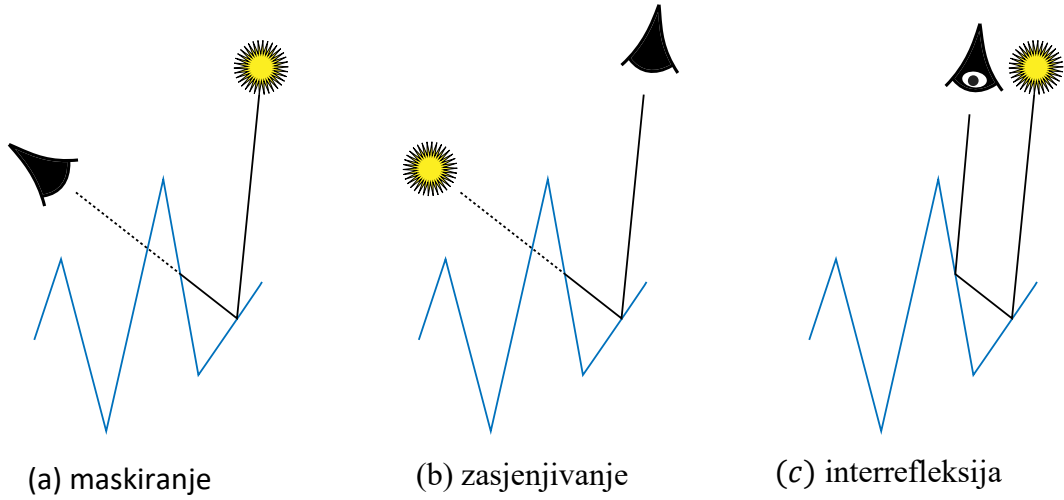
Slika 4.6 Primjer dvije različite površine

BRDF modeli temeljeni na teoriji malih djelića površine pretpostavljaju da se diferencijalna površina koja je osvijetljena sastoji od mnoštva malih djelića površine pa se raspršivanje svjetlosti na toj površini modelira statistički.

Dvije glavne komponente ovakvog BRDF modela su distribucija normala na male djeliće površine i BRDF koji opisuje refleksiju na pojedinim djelićima površine. Zadatak nam je koristeći te dvije informacije odrediti zatvorenu formu BRDF-a koji opisuje refleksiju svjetla s takve površine.

Pri izračunu refleksije takvog modela treba uzeti u obzir lokalne učinke na nivou malih djelića površine. Tri takva učinka su prikazana na slici Slika 4.7. Pod (a) je prikazano maskiranje – promatrani djelić površine nije vidljiv zbog toga što je prekriven nekim drugim djelićem površine. Pod (b) je prikazano zasjenjivanje (engl. *shadowing*) – slično kao i maskiranje, ali u ovom slučaju svjetlost ne dolazi do promatranog djelića površine. Pod (c) je prikazana interrefleksija – svjetlo se reflektira na više djelića površine prije nego što dođe do promatrača.

Konkretni BRDF modeli temeljeni na teoriji malih djelića površine ove efekte uzimaju u obzir s različitim nivoima točnosti, što ćemo imati priliku vidjeti u nastavku.



Slika 4.7 Tri važna geometrijska efekta

4.3.4.1 Oren-Nayar difuzna refleksija

Pokazalo se da stvarni difuzni objekti nisu baš najbolje modelirani Lambertovim BRDF-om te je zbog toga razvijen novi model koji može vjernije modelirati stvarne grube površine uz pretpostavku da su mali djelići površine oblika slova V i da se normale pokoravaju sfernoj Gaussovoj distribuciji.

Konačni model koji uzima u obzir maskiranje, zasjenjivanje i interrefleksije nema rješenje u zatvorenoj formi, no postoji aproksimacija koja dosta dobro aproksimira model, a dana je izrazom (4.9) gdje je $A = 1 - \frac{\sigma^2}{2(\sigma^2 + 0.33)}$, $B = \frac{0.45\sigma^2}{\sigma^2 + 0.09}$, $\alpha = \max(\theta_i, \theta_o)$, $\beta = \min(\theta_i, \theta_o)$.

$$f_r(\omega_i, \omega_o) = \frac{R}{\pi} (A + B \max(0, \cos(\phi_i - \phi_o)) \sin \alpha \tan \beta) \quad (4.9)$$

Na slici Slika 4.8 je prikazana usporedba Lambertovog i Oren-Nayar BRDF modela. Može se primijetiti kako se kod Oren-Nayar modela javlja pojačana refleksija na siluetama objekta.



Slika 4.8 Usporedba Lambert BRDF (lijevo) i Oren-Nayar BRDF (desno) modela

4.3.4.2 Funkcije distribucije normala

Važna karakteristika površine opisane malim djelićima površine je distribucijska funkcija $D(\omega_h)$ koja opisuje koliki udio djelića površine ima normalu ω_h . Te funkcije moraju biti normalizirane što slijedi iz principa očuvanja energije što znači da mora vrijediti izraz (4.10).

$$\int_{H^2(n)} D(\omega_h) \cos \theta_h d\omega_h = 1 \quad (4.10)$$

Jedna često korištena distribucija je Beckmann-Spizzichino distribucija definirana izrazom (4.11) gdje je σ kvadratna sredina (engl. *root mean square* – RMS) nagiba djelića površine, a $\alpha = \sqrt{2} \sigma$.

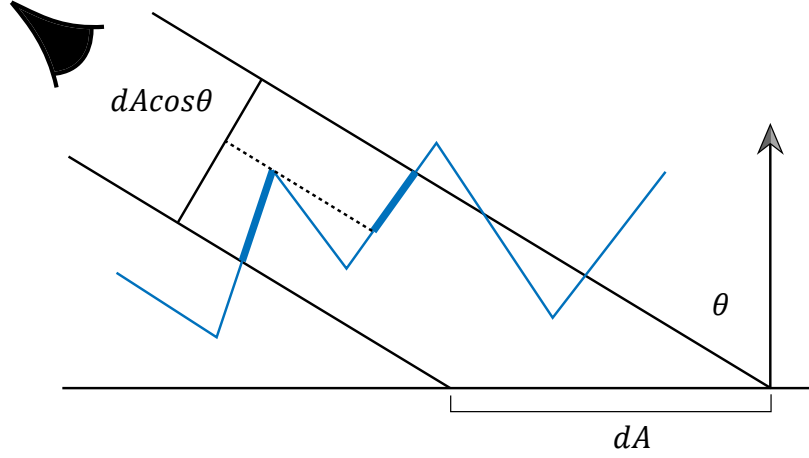
$$D(\omega_h) = \frac{e^{-\tan^2 \frac{\theta_h}{\alpha^2}}}{\pi \alpha^2 \cos^4 \theta_h} \quad (4.11)$$

4.3.4.3 Maskiranje i zasjenjivanje

Maskiranje i zasjenjivanje su definirani Smithovom funkcijom $G_1(\omega, \omega_h)$ koja daje udio djelića površine s normalom ω_h koji su vidljivi iz smjera ω . Primijetimo da vrijedi $0 \leq G_1(\omega, \omega_h) \leq 1$. U slučaju kada je vjerojatnost da je djelić površine vidljiv nezavisna o njegovoj normali, možemo pisati funkciju kao $G_1(\omega)$. Kao što je vidljivo na slici Slika 4.9, diferencijalna površina dA kada se promatra iz smjera ω koji zatvara kut θ s normalom na

površinu ima iznos površine $dA \cos \theta$. Tada i površina vidljivih djelića površine mora biti jednaka $dA \cos \theta$ što dovodi do normalizacijskog uvjeta za G_1 (izraz (4.12)).

$$\cos \theta = \int_{H^2(n)} G_1(\omega, \omega_h) \max(0, \omega \cdot \omega_h) D(\omega_h) d\omega_h \quad (4.12)$$



Slika 4.9 Prikaz površine i djelića površine

Već smo rekli da djelići površine tvore visinsko polje te zbog toga vrijedi da svaki djelić površine koji je okrenut natrag (engl. *backfacing*) zasjenjuje djelić površine koji je okrenut prema naprijed (engl. *frontfacing*) i ima istu projiciranu površinu u smjeru ω . Ako je $A^+(\omega)$ projicirana površina djelića površine koji su okrenuti prema naprijed s obzirom na smjer ω i ako je $A^-(\omega)$ projicirana površina djelića površine koji su okrenuti prema natrag s obzirom na smjer ω , tada je $\cos \theta = A^+(\omega) - A^-(\omega)$. Zato možemo zapisati Smithovu funkciju kao u izrazu (4.13).

$$G_1(\omega) = \frac{A^+(\omega) - A^-(\omega)}{A^+(\omega)} \quad (4.13)$$

Smithove funkcije se uobičajeno iskazuju pomoću pomoćne funkcije $\Lambda(\omega)$ koja daje omjer nevidljive maskirane površine i vidljive površine (izraz (4.14)).

$$\Lambda(\omega) = \frac{A^-(\omega)}{A^+(\omega) - A^-(\omega)} = \frac{A^-(\omega)}{\cos(\theta)} \quad (4.14)$$

Uvrstivši izraz (4.14) u izraz (4.13) dobivamo izraz (4.15).

$$G_1(\omega) = \frac{1}{1 + \Lambda(\omega)} \quad (4.15)$$

Trenutno funkcija $\Lambda(\omega)$ nije jednoznačno određena, no ako uvedemo pretpostavku da susjedne točke na površini nisu korelirane, onda je moguće odrediti jednoznačan $\Lambda(\omega)$ za dani $D(\omega_h)$. Navedena pretpostavka zapravo nije istinita u stvarnosti, ali su dobiveni modeli ipak dosta precizni.

Uz navedenu pretpostavku $\Lambda(\omega)$ za Beckmann-Spizzichino distribuciju je dana izrazom (4.16) gdje $a = \frac{1}{\alpha \tan(\theta)}$ i $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-x'^2} dx'$.

$$\Lambda(\omega) = \frac{1}{2} \left(\text{erf}(a) - 1 + \frac{e^{-a^2}}{a\sqrt{\pi}} \right) \quad (4.16)$$

Definirajmo još jednu korisnu funkciju $G(\omega_o, \omega_i)$ koja daje udio djelića površine u diferencijalnoj površini koji su vidljivi iz oba smjera, ω_o i ω_i . Jedan način za definirati tu funkciju je da pretpostavimo da su ta dva događaja uvjetno nezavisna iz čega slijedi izraz (4.17).

$$G(\omega_o, \omega_i) = G_1(\omega_o)G_1(\omega_i) \quad (4.17)$$

Ako malo bolje razmislimo shvatit ćemo da ova pretpostavka nije točna. Ako uzmemo $\omega_o = \omega_i$, onda iz izraza (4.17) dobivamo $G(\omega_o, \omega_i) = G_1(\omega_o)G_1(\omega_o) = G_1(\omega_i)G_1(\omega_i)$, no to u općem slučaju nije točno jer je u ovom slučaju svaki djelić površine vidljiv iz smjera ω_o također vidljiv i iz ω_i pa je $G(\omega_o, \omega_i) = G_1(\omega_o) = G_1(\omega_i)$ te smo dobili kontradikciju. Znači da naša pretpostavka nije bila dobra.

Malo točniji model se može dobiti ako pretpostavimo da se vidljivost djelića površina povećava s visinom točke na površini. Ova pretpostavka dovodi do izraza (4.18).

$$G(\omega_o, \omega_i) = \frac{1}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)} \quad (4.18)$$

4.3.5. Torrance-Sparrow model

Torrance-Sparrow model modelira metalne površine kolekcijom savršeno zrcalnih djelića površine. Iz toga slijedi da samo oni djelići površine čija je normala jednaka polukutnom vektoru (engl. *half-angle vector*) $\omega_h = \widehat{\omega_i + \omega_o}$ dovode do reflektiranja ω_i u ω_o .

Izvod Torrance-Sparrow BRDF-a dan je u [2], a ovdje će samo biti prikazan krajnji rezultat (izraz (4.19)).

$$f_r(\omega_o, \omega_i) = \frac{D(\omega_h)G(\omega_o, \omega_i)F_r(\omega_o)}{4\cos\theta_o\cos\theta_i} \quad (4.19)$$

Izvod Torrance-Sparrow BRDF-a ne ovisi o konkretnoj distribuciji normala djelića površine, niti o konkretnoj Fresnelovoj funkciji što znači da se može koristiti i za dielektrike i za vodiče.

4.4. Sudjelujući mediji

4.4.1. Fazne funkcije

Kao što postoji mnoštvo BSDF modela koji opisuju raspršivanje svjetlosti na površinama, postoji i mnoštvo reprezentacija faznih funkcija kao što su parametarski modeli (dobiveni npr. uzimanjem neke parametarske funkcije i namještanjem parametara da što bolje odgovaraju mjerenim podacima), analitički modeli dobiveni izvodom distribucije raspršivanja pod pretpostavkom da medij čine čestice poznatog oblika i materijala. Također se mogu izravno koristiti i mjereni podaci iz laboratorija ili podaci dobiveni simulacijom.

U većini sudjelujućih medija koje susrećemo u stvarnosti, fazna funkcija je jednodimenzionalna funkcija kuta θ između dva smjera ω_o i ω_i te se označava s $p(\cos\theta)$. Mediji koji imaju ovakvu faznu funkciju se nazivaju izotropni jer je njihov odziv na upadnu svjetlost lokalno invarijantan pod rotacijama. Fazne funkcije moraju biti normalizirane što slijedi iz principa očuvanja energije. Za izotropne medije trivijalno vrijedi svojstvo recipročnosti $p(\omega_i, \omega_o) = p(\omega_o, \omega_i)$ jer $p(\cos(-\theta)) = p(\cos(\theta))$. U anizotropnim medijima fazna funkcija je četverodimenzionalna funkcija dva smjera ω_i i ω_o te zadovoljava kompleksnije svojstvo recipročnosti.

Kao što mediji mogu biti izotropni ili anizotropni, i same fazne funkcije mogu biti izotropne ili anizotropne. Izotropna fazna funkcija opisuje raspršivanje svjetlosti u svim smjerovima jednako. Zbog svojstva normaliziranosti faznih funkcija postoji samo jedna takva fazna funkcija dana izrazom (4.20).

$$p(\omega_i, \omega_o) = \frac{1}{4\pi} \quad (4.20)$$

Jedna popularna fazna funkcija u računalnoj grafici je Henyey-Greenstein fazna funkcija koja je preuzeta iz astronomije [8]. Njena definicija je dana izrazom (4.21).

$$p_{HG}(\cos\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 + 2g(\cos\theta))^{\frac{3}{2}}} \quad (4.21)$$

Funkcija ima jedan parametar $g \in [-1,1]$ čijim se podešavanjem ova funkcija može koristiti za raspršivanje pretežito unatrag ($g \in [-1,0)$), unaprijed ($g \in (0,1]$) ili za izotropno raspršivanje ($g = 0$).

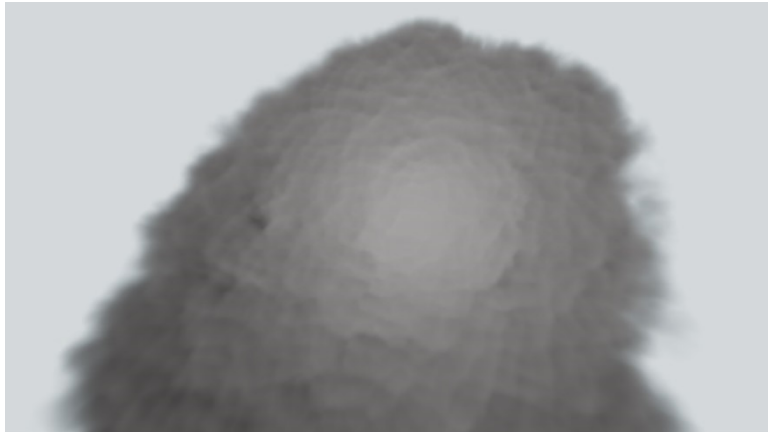
Za danu faznu funkciju, njezina Henyey-Greenstein aproksimacija se dobije tako da se odredi parametar g pomoću izraza (4.22).

$$g = \int_{S^2} p(-\omega, \omega') (\omega \cdot \omega') d\omega' = 2\pi \int_0^\pi p(-\cos\theta) \cos\theta \sin\theta d\theta \quad (4.22)$$

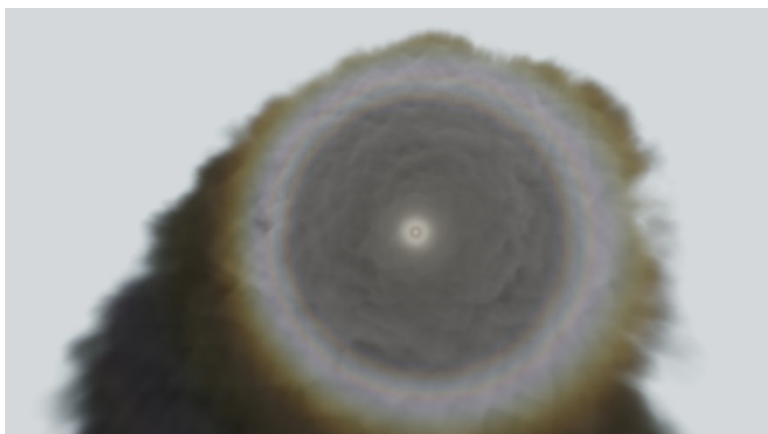
Kompleksnije fazne funkcije se mogu dobiti tako da se napravi težinska suma više Henyey-Greenstein faznih funkcija s različitim parametrima g (izraz (4.23)). Za težine w_i vrijedi da njihova suma mora biti jednaka 1 zbog očuvanja svojstva normaliziranosti.

$$p(\omega, \omega') = \sum_{i=1}^n w_i p_i(\omega, \omega') \quad (4.23)$$

U [9] je u kontekstu iscrtavanja oblaka dana usporedba korištenja težinske sume Henyey-Greenstein faznih funkcija i fizikalno utemeljene Mie fazne funkcije koja je dobivena pomoću simulacije i onda spremljena u jednodimenzionalnu teksturu i korištena u sjenčaru. Na slici Slika 4.10 je prikazan rezultat kada se koristi težinska suma Henyey-Greenstein faznih funkcija, a na slici Slika 4.11 rezultat kada se koristi Mie fazna funkcija. Primijetimo kako Mie fazna funkcija sadrži detalje karakteristične za oblake koje Henyey-Greenstein aproksimacija ne može dobro aproksimirati.



Slika 4.10 Težinska suma Henyey-Greenstein faznih funkcija



Slika 4.11 Mie fazna funkcija

4.4.2. Tipovi sudjelujućih medija

Osim fazne funkcije (koja je u sustavu uvijek Henyey-Greenstein fazna funkcija), medij je definiran i koeficijentima apsorpcije σ_a i raspršivanja σ_s (a time je definiran i koeficijent izumiranja σ_t). U ovisnosti o tome ovise li koeficijenti o poziciji u sudjelujućem mediju, razlikujemo dva tipa medija: homogene i heterogene medije.

Homogeni mediji su najjednostavniji mediji koji imaju u svakoj točki prostora jednake vrijednosti koeficijenata σ_a i σ_s . Heterogeni mediji, s druge strane, u svakoj točki prostora imaju (potencijalno) različite vrijednosti koeficijenata σ_a i σ_s .

4.5. Paralelizacija izvođenja algoritma

Algoritam praćenja puta ima veliku mogućnost paralelizacije jer su izračuni za različite slikovne elemente nezavisni te se mogu izvoditi istodobno na različitim procesorima. Jedna opcija iskorištavanja ovog paralelizma bi bila izvođenje istovremeno na više dretvi CPU-a. Ipak, današnji CPU-ovi imaju relativno malo jezgri (≈ 16) u odnosu na broj slikovnih elemenata ($1920 \times 1080 = 2073600$). Današnji GPU-ovi, nasuprot, imaju jako veliki broj jezgri (nekoliko tisuća) i pružaju radne okvire za programiranje opće namjene kao što su CUDA, OpenCL, DirectCompute itd.

Jedan veoma interesantan programski model je SYCL, standardiziran od strane Khronos grupe. SYCL je besplatni, višeplatformni, apstrakcijski sloj koji pruža programerima mogućnost pisanja programskog koda u standardnom ISO C++ jeziku, a koji se može izvršavati na raznim heterogenim procesorima kao što su GPU-ovi, CPU-ovi, FPGA pločice

itd [10]. SYCL je bio zamišljen kao programski model više razine unutar OpenCL specifikacije, no sada je on potpuno odvojen od OpenCL standarda te postoje razne implementacije SYCL specifikacije koje nisu nužno implementirane pomoću OpenCL-a, npr. Intelov DPC++ prevoditelj može koristiti CUDA implementaciju SYCL specifikacije [11].

Uvedimo neke važne SYCL pojmove:

- domaćin (engl. *host*) je entitet koji započinje s izvršavanjem koda (najčešće CPU) i zadužen je za upravljanjem jezgrenim (engl. *kernel*) kodom koji se izvršava na akceleracijskom uređaju (najčešće GPU)
- uređaj (engl. *device*) je entitet koji izvršava posao (najčešće GPU) koji mu domaćin zadaje
- jezgreni kod je kod koji se izvršava na uređaju. On se može pisati u ISO C++17 standardu, ali neke značajke nisu dostupne zbog toga što bi imale previše negativan utjecaj na performanse (dinamički polimorfizam, dinamičke alokacije memorije, statičke varijable, pokazivači na funkcije, baratanje iznimkama, rekurzija...).

SYCL kod je jednoizvorni (engl. *single-source*) što znači da jedna jedinica prevođenja (engl. *translation unit*) sadrži i jezgreni kod i kod koji se izvršava na domaćinu, a koji upravlja jezgrenim kodom.

Jednostavan primjer SYCL koda je prikazan u isječku Isječak 4.1.

Isječak 4.1 Primjer jednostavnog SYCL programa

```
#include <array>
#include <iostream>
#include <CL/sycl.hpp>

auto main() -> int
{
    constexpr std::size_t size = 16;
    std::array<int, size> data {};

    // Kreiraj red zadataka na pretpostavljenom (default) uređaju.
    sycl::queue q {};

    // Kreiraj spremnik koristeći podatke u "data" polju.
    sycl::buffer b { data };

    q.submit([&](sycl::handler& h) {
        // Dohvaćanje pristupa spremniku na uređaju.
        sycl::accessor a { b, h };
        h.parallel_for(size, [=](auto& idx) {
            // Ovo je jedina linija koja se izvršava na uređaju.
            // Sve linije izvan ove lambda funkcije se
            // izvršavaju na domaćinu.
            a[idx] = idx;
        });
    });

    // Dohvaćanje pristupa spremniku na domaćinu.
    // Čekat će dok uređaj ne završi s izvođenjem jezgrene funkcije.
    sycl::host_accessor a { b };
    for (std::size_t i = 0; i < size; i++)
        std::cout << "data[" << i << "] = " << a[i] << '\n';
}
```

Važno je napomenuti da tipovi podataka koji se dijele između domaćina i uređaja moraju zadovoljavati C++ zahtjeve standardnog rasporeda (engl. *standard layout requirements*). Svi skalarni tipovi zadovoljavaju te zahtjeve, no valja biti oprezan kod korisnički definiranih razreda (engl. *class*). U suštini, razred koji zadovoljava zahtjeve standardnog rasporeda

nema virtualnih funkcija, niti virtualnih nadrazreda. Također, nema nadrazreda niti podatkovnih članova koji imaju tip koji ne zadovoljava zahtjeve standardnog rasporeda. Svi podatkovni članovi takvog razreda moraju biti definirani u istom razredu (mogu biti definirani i u nadrazredu) i moraju imati istu vidljivost. [12]

U implementaciji je korišten statički polimorfizam pomoću tipa označene unije (engl. *tagged union* ili engl. *variant*) [13]. Interesantno, C++ implementacija označene unije, `std::variant` po ISO C++ standardu ne mora zadovoljavati zahtjeve standardnog rasporeda. Jedna moguća implementacija označene unije koja zadovoljava zahtjeve standardnog rasporeda je dana u [14]. Također, potrebno je koristiti implementaciju `visit` funkcije koja se može izvršavati na uređaju. `std::visit` (na svim popularnim prevoditeljima) i implementacija `visit` dana u [14] se ne mogu izvršavati na uređaju jer koriste pokazivače na funkcije u svojoj implementaciji. Jedna moguća implementacija koja se može koristiti u kodu koji se izvršava na uređaju je dana u [15].

5. Rezultati

U ovom poglavlju su predstavljeni rezultati testiranja sustava na nekoliko scena preuzetih s [16].

Na slici Slika 5.1 je prikazan oblak modeliran kao heterogeni sudjelujući medij koji ima jako izražen učinak raspršivanja svjetlosti zbog visoke vrijednosti koeficijenta raspršivanja σ_s . Rezolucija slike je 1024×1024 , a broj uzetih uzoraka po slikovnom elementu je 1024.

Za usporedbu, na slici Slika 5.2 je prikazan sudjelujući medij s istom strukturom, ali s povećanim koeficijentom apsorpcije σ_a što je tipično za npr. dim. U ovom slučaju dominira apsorpcija nad raspršivanjem te je medij zbog toga dosta taman. I u ovom slučaju je rezolucija slike 1024×1024 , a broj uzetih uzoraka po slikovnom elementu je 1024.

Na slici Slika 5.3 je prikazana jedna sličica (engl. *frame*) simulacije fluida. Fluid je modeliran kao homogeni medij čija je granica definirana kompleksnom mrežom poligona, Staklena posuda u kojoj se fluid nalazi je modelirana pomoću zrcalnog BSDF-a. Rezolucija slike je 1200×700 , a broj uzetih uzoraka po slikovnom elementu je 1024.

Na slici Slika 5.4 je prikazana simulacija dima oko prepreke. Dim je i u ovom slučaju modeliran kao heterogeni sudjelujući s dosta izraženim koeficijentom apsorpcije σ_a , a prepreka je modelirana kao kugla s zrcalnim BSDF-om.

U svim primjerima izvor svjetlosti je modeliran mapom okoline (engl. *environment map*) koja definira distribuciju sjajnosti.



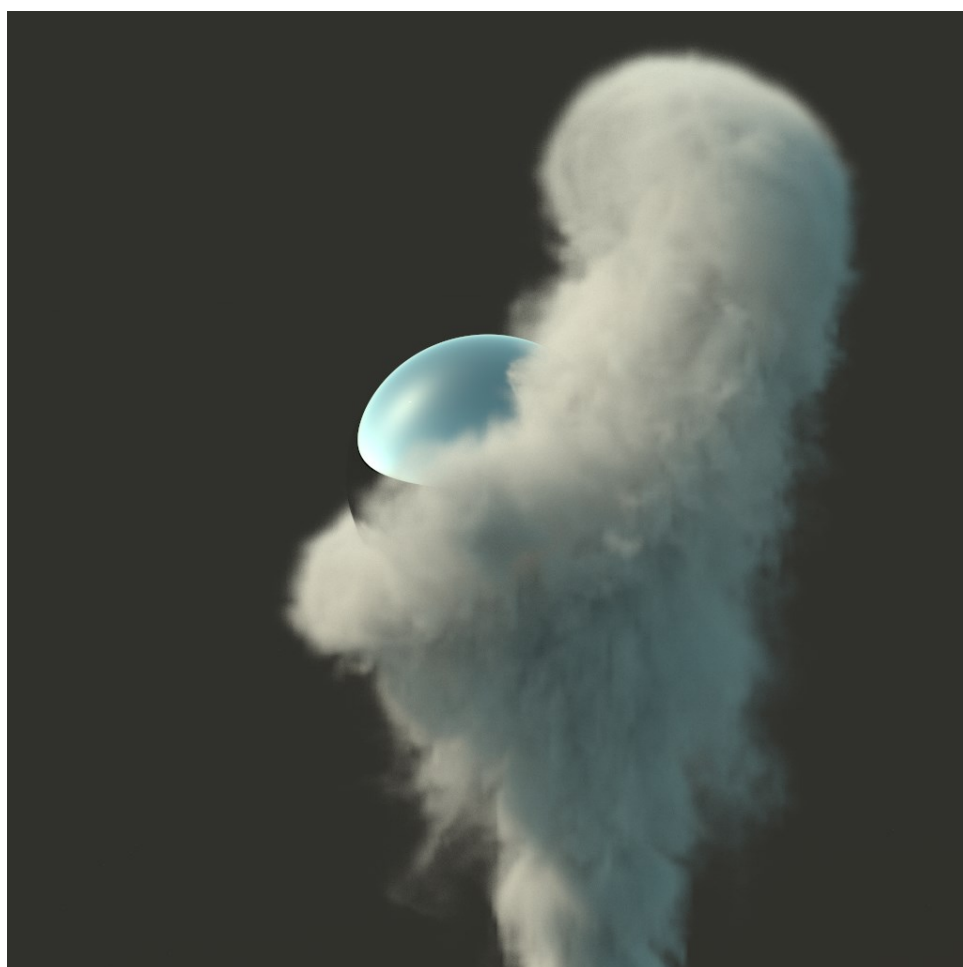
Slika 5.1 Oblak



Slika 5.2 Dim



Slika 5.3 Simulacija fluida



Slika 5.4 Simulacija dima

6. Zaključak

Algoritam volumetrijskog praćenja puta generira slike visoke razine fotorealističnosti jer ima čvrste fizikalne temelje. Jedan veliki nedostatak algoritma je visoka računska složenost što je rezultat prvenstveno složenosti izračuna presjeka zrake s objektima u sceni i složenosti radi činjenice da je potrebno uzeti jako puno uzoraka kako u konačnoj slici ne bi bilo šuma.

U radu su predstavljene akceleracijske strukture koje ublažavaju prvi faktor. Drugi faktor u ovom radu nije predstavljao problem jer algoritam volumetrijskog praćenja puta nije korišten u interaktivnom kontekstu. U tom kontekstu nije moguće priuštiti uzimanje nekoliko tisuća uzoraka kako bi se dobila slika bez šuma pa su za taj problem razvijene tehnike uklanjanja šuma (engl. *denoising*).

U sklopu ovog rada napravljena je implementacija sustava u programskom modelu SYCL što omogućava izvršavanje sustava i na CPU-ovima i na GPU-ovima. Moderne grafičke kartice, napravljene u zadnjih nekoliko godina, sadrže komponente koje služe za sklopovsku akceleraciju algoritma praćenja zrake. U sklopu ovog rada te mogućnosti nisu korištene pa to predstavlja jedno moguće proširenje rada koje bi dodatno poboljšalo performanse sustava kada bi se izvršavao na modernim grafičkim karticama.

Literatura

- [1] Radivoj, L. Algoritam praćenja zrake i praćenja puta za prikazivanje scena u stvarnom vremenu. Diplomski rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2021.
- [2] Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. Physically Based Rendering: From Theory to Implementation (3rd. ed.).
- [3] Wikipedija. Kvantna elektrodinamika.
https://hr.wikipedia.org/wiki/Kvantna_elektrodinamika, pristupano 24.6. 2020.
- [4] Wikipedija. Radiometrija.
<https://hr.wikipedia.org/wiki/Radiometrija>, pristupano: 24.6.2022.
- [5] Wenzel, J. Light Transport on Path-Space Manifolds. PhD thesis. Cornell University, 2013.
- [6] Marko Mijolović. Volumetric Path Tracing.
<https://gitlab.com/markomijolovic/volumetric-path-tracing>, pristupano 24.6.2022.
- [7] Tomas Akenine-Muller, Eric Haines, and Naty Hoffman. 2018. Real-Time Rendering, Fourth Edition (4th. ed.). A. K. Peters, Ltd., USA.
- [8] J. Patrick Harrington. The Henyey-Greenstein phase function.
https://www.astro.umd.edu/~jph/HG_note.pdf, pristupano 24.6.2022.
- [9] Mijolović, M. Fizikalno temeljen prikaz oblaka. Završni rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2020.
- [10] Khronos Group. *SYCL 2020 Specification*.
<https://www.khronos.org/registry/SYCL/specs/sycl-2020/pdf/sycl-2020.pdf>, pristupano 24.6.2022.
- [11] Intel. *Getting Started with oneAPI DPC++*.
<https://intel.github.io/llvm-docs/GetStartedGuide.html>, pristupano 24.6.2022.
- [12] Cppreference. C++ named requirements: Standard Layout.
https://en.cppreference.com/w/cpp/named_req/StandardLayoutType, pristupano 24.6.2022.

[13] Wikipedija. Tagged union.

https://en.wikipedia.org/wiki/Tagged_union, pristupano 24.6.2022.

[14] Michael Park. Variant.

<https://github.com/mpark/variant>, pristupano 24.6.2022.

[15] triSYCL. Path tracer.

https://github.com/triSYCL/path_tracer, pristupano 24.6.2022.

[16] PBRT. Scenes.

<https://pbrt.org/scenes-v3>, pristupano 24.6.2020.

Sažetak

Volumetrijsko praćenje puta

U ovom diplomskom radu opisan je problem fotorealističnog iscrtavanja u računalnoj grafici. Opisana su dva sustava za kvantificiranje svjetlosti, radiometrija i fotometrija, te pretpostavke na kojima se temelje. Opisana je temeljna jednažba iscrtavanja koju je potrebno riješiti za ostvarivanje prikaza. Objašnjen je način rješavanja navedene jednažbe Monte Carlo metodama. Opisane su tehnike za poboljšavanje rezultata Monte Carlo metode. Objašnjeni su neki načini ubrzavanja algoritma: izračun na grafičkom procesoru pomoću SYCL standarda te korištenje BVH akceleracijske strukture za ubrzavanje izračuna presjeka zrake s geometrijskim objektima u sceni. Prikazani su rezultati te su izneseni komentari istih.

Ključne riječi: fizikalno temeljeno iscrtavanje, praćenje zrake, praćenje puta

Summary

Volumetric path tracing

This master's thesis introduces the problem of photorealistic rendering in computer graphics. Two systems for quantifying light are introduced, radiometry and photometry, and the assumptions on which they are based. The fundamental rendering equation is introduced. Monte Carlo methods that are used to solve the aforementioned equation are introduced. Some techniques that improve the results of the Monte Carlo method are introduced. Some ways of accelerating the algorithm are introduced: computation on the GPU using the SYCL standard and the usage of the BVH acceleration structure for the acceleration of ray-object intersection tests. The results of the thesis are presented and evaluated.

Keywords: physically based rendering, ray tracing, path tracing