

Fakultet informatike i digitalnih tehnologija

Sveučilišni prijediplomski studij Informatika

Seminarski rad iz kolegija

MULTIMEDIJSKE TEHNOLOGIJE

Ray Tracing

Autor: Tin Švagelj

Voditelj kolegija: doc. dr. sc., Miran Pobar

Rijeka, 29. siječnja 2025.

SADRŽAJ

1. Uvod	1
1.1. Motivacija	1
1.2. Povijest razvoja	1
2. Hardverska podrška	2
3. Temeljni način rada	3
3.1. Koraćanje unaprijed	3
3.2. Koraćanje unazad	3
4. Geometrija scene	5
5. Svijetlo, boje i kolorimetrija	6
6. Kamera	7
7. Materijali	8
7.1. Teksture	8
8. Izvori i interakcije svjetlosti	9
8.1. Refleksija	9
9. Otklanjanje buke/šuma	12
10. Usporedba s klasičnom rasterizacijom	13
10.1. Odrazi	13
10.2. Ambijentalna okluzija	13
10.3. Sjene	14
11. Zaključak	14
Literatura	15

1. Uvod

Metoda praćenja zraka svjetlosti (engl. *ray tracing*, RT) se zasniva na relaksaciji problema simulacije ponašanja svjetlosti u zatvorenom optičkom sustavu. Ray tracing metode nastoje provesti idealnu simulaciju ponašanja svijetla kako bi postigle rezultate bliske stvarnima.

1.1. Motivacija

Autor je odabrao ovu temu za seminar jer unatoč nekom osnovnom znanju u principu rada ray tracing algoritama nije nikada ručno implementirao ray tracing algoritam te je imao loš uvid u stvarnu složenost implementacije. Velik dio te složenosti zapravo uvodi arhitekturiranje pogona za iscrtavanje (engl. *rendering engine*) kada se koriste apstraktna sučelja za programiranje (engl. *abstract programming interface*, API) grafičkih kartica (engl. *graphics card/graphics processing unit*, GPU) poput Vulkan, OpenGL, DirectX ili Metala.

1.2. Povijest razvoja

Prvi temelj ovog algoritma je objavio P. W. Ford 1960. godine u radu “Nova shema praćenja zraka svjetlosti” (engl. “*New Ray Tracing Scheme*”), a bazirao ga je na jednadžbama koje je H. A. Buchdahl laboratorijski testirao i opisao u monografiji “Koeficijenti optičke aberacije” (engl. “*Optical Aberration Coefficients*”).

Fordov algoritam je bio osmišljen za aksijalno simetrične optičke sustave te je baratao s “idealnim zrakama svjetlosti” koje ne prate stvarne zakone refrakcije svijetla nego paraksijalnu optiku koja dopušta linearne aproksimacije refrakcije. Uporaba parakoničnih koordinata je pojednostavilo matematičku analizu, te pružilo iznimno brzu metodu izračuna za sustave koji sadrže isključivo sfere. [1]

Kroz narednih 20 godina je bilo nekoliko manjih pomaka, no zbog hardverskih ograničenja je ova tehnika vidjela značajan napredak tek sredinom 80ih godina (20. st.), kada je na Siggraphu i drugim manjim konferencijama, kao i u akademiji bilo objavljeno preko 150 različitih radova, članaka i prezentacija na temu.

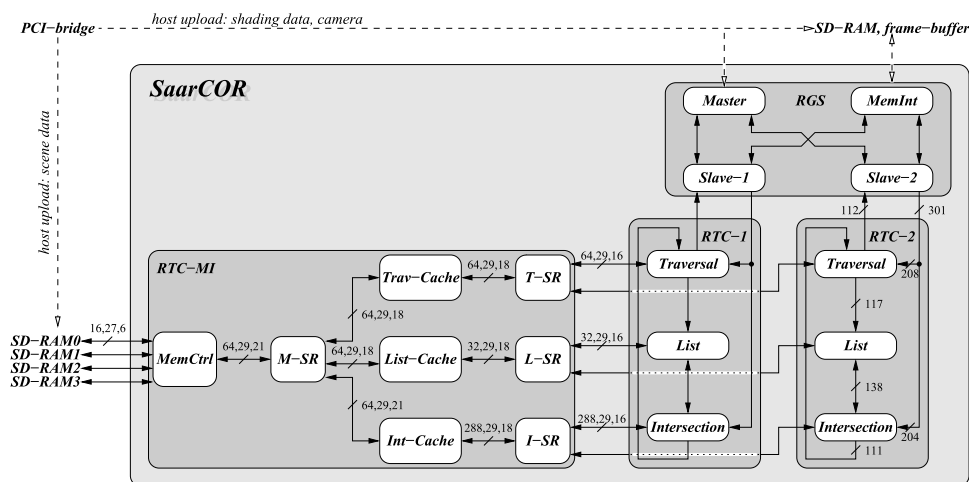
Mnoge od tih tehnika su i danas primjenjive, no zbog tadašnjih hardverskih ograničenja su bile primarno korištene za prijevremen (engl. *offline*) prikaz.

Nvidia je 2020. godine objavila Ampere seriju grafičkih kartica (engl. *graphics processing unit*, GPU) za radne stanice (profesionalna primjena) [2] koje imaju specijalizirane hardverske komponente za određene izračune koji su opisani u 2. dijelu, te nedugo zatim i komercijalnu seriju RTX grafičkih kartica. Te kartice su omogućile provođenje jednostavnijih ray tracing algoritama u realnom vremenu (engl. *online*).

2. Hardverska podrška

Prethodno specijaliziranom hardveru, ray tracing metode su funkcionirale tako što se na zaslonu prikazao običan kvadrat koji pokriva cijelu površinu pogleda, vertex shader bi služio samo za prosljeđivanje podataka fragment shaderu, a fragment shader bi bio zadužen za simulaciju zraka, izračun i akumulaciju podataka koji su pruženi uzorkovanjem zraka koje su pridružene individualnim pikselima.

2002. godine su Jörg Schmittler, Ingo Wald i Philipp Slusallek demonstrirali prednost uporabe specijaliziranog hardvera za provjeru intersekcija zraka sa geometrijom sadržanoj u sceni (engl. *scene geometry*, nadalje geometrija scene ili geometrija). [3] Slika 1 prikazuje primjer takve jezgre za koračanje zrakom svjetlosti (engl. *Ray Tracing Core*, engl. *RT Core*, RT jezgra)



Slika 1: SaarCOR kao primjer RT jezgre. Izvor: [3]

Arhitektura modernih RT jezgri nije značajno različita u funkcionalnosti koju pruža, pa je Slika 1 dovoljna aproksimacija za ovaj seminarski rad. Ostatak seminarskog rada pretpostavlja uporabu RT jezgri grafičke kartice, ili u kontekstu objašnjenja praktičnog dijela prijevremen prikaz putem procesora (engl. *central processing unit*, CPU).

Uz specijaliziran hardver su u Vulkan, D3D12 i Metal dodana proširenja koja omogućuju njegovu uporabu.

Ta proširenja generalno dodaju:

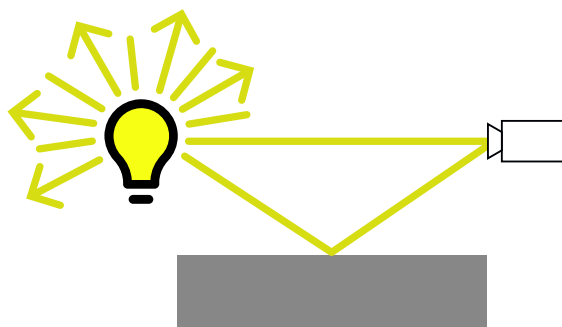
- strukture i funkcije za prijenos podataka o geometriji (trokutima ili AABB okvirima) i zrakama,
- programabilno sučelje za provođenje testiranja presjeka zraka s geometrijom u sklopu programa provedenom na grafičkoj kartici (engl. *shader*),
 - fazu obrade zaprimljenih podataka na grafičkim karticama.

3. Temeljni način rada

Temeljno razlikujemo algoritme koji koračaju zrakama svjetlosti unaprijed ili unazad. Pravilan odabir algoritma ovisi o području primjene. Za svrhu prikaza računalne grafike namijenjene zabavi su algoritmi koračanja zrakama unazad bolji izbor jer točnost nije bitna.

3.1. Koračanje unaprijed

Koračanje zrakama unaprijed je vjerodostojno stvarnom ponašanju svjetlosti u prirodi, no jako je potrošan za svrhu grafičkog prikaza jer većinu odaslanih zraka svjetlosti apsorbiraju predmeti u sceni ili se dovoljno udalje da više nemaju značajan utjecaj na rezultat prikaza, te nisu vidljivi kao što je primjetno na slici 2.

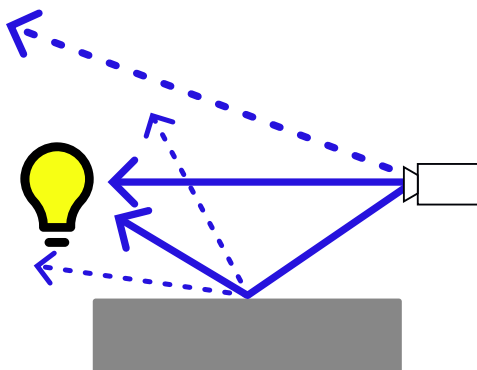


Slika 2: Zrake svjetlosti odaslane iz izvora

Apsorbirana svjetlost ima fizičke manifestacije na tijela poput uzbuđenja elektrona i zagrijavanja tvari, pa je za određene namjene (npr. fizičke simulacije [4]) ovaj pristup jedino smislen.

3.2. Koračanje unazad

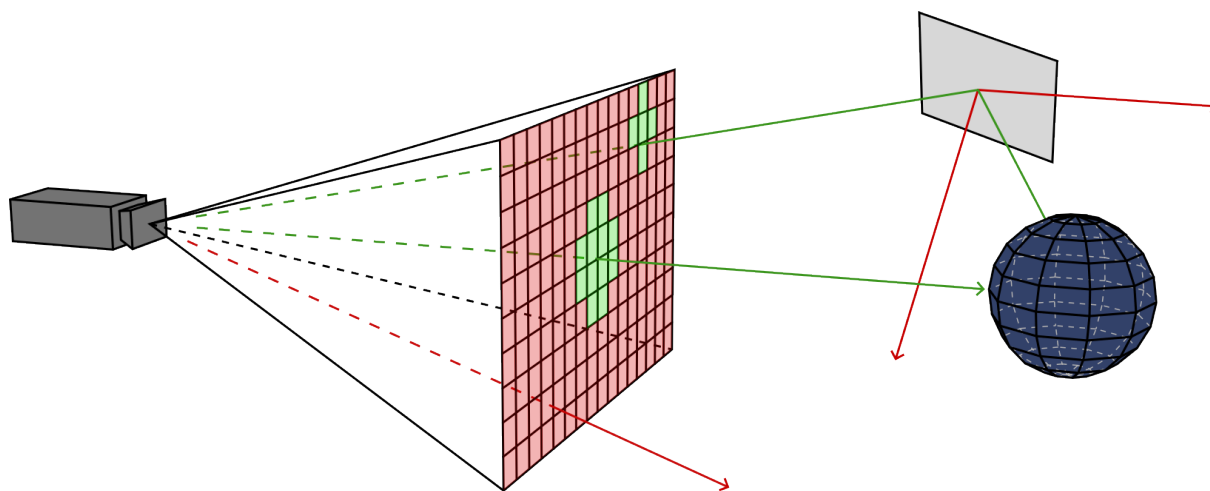
Koračanje zrakama unazad se razlikuje od prethodno opisanog pristupa po tome što ne simulira zrake svjetlosti nego unazadno uzorkuje osvjetljenje sa scene odašiljajući zrake uzorkovanja (nadalje zrake) iz kamere kao što je prikazano na slici 3.



Slika 3: Zrake uzorkovanja odaslane iz kamere

Slika 4 prikazuje shematski prikaz principa rada unazadnog koračanja zrakama. Unazadni RT algoritam treba:

1. stvoriti jednu ili više zraka kojima će boja svakog piksela prikaza biti uzorkovana:
 - pozicija svake zrake je jednaka poziciji kamere, a smjer zrake ovisi o vrsti projekcije, leći kamere, te poziciji piksela kojem je zraka pridružena,
2. prolaskom kroz predmete u sceni, pronaći najbliži kameri kojeg zraka dodiruje,
3. provesti prikladnu interakciju zavisno o materijalu,
 - ako se radi o reflektivnom materijalu ponoviti uzorkovanje od 1. koraka za uzorkovanu točku kako bi se odredila reflektirana svjetlost (L_i),
 - ako se radi o nesavršenom zrcalu, potrebno je uzorkovati nekoliko zraka, pa zatim
4. pohraniti uzorkovanu vrijednost (ili težinski prosjek više njih) u međuspremniku (engl. *buffer*) za prikaz.



Slika 4: Pronalaženje presjeka zraka s tijelima

4. Geometrija scene

Određivanje presjeka zrake s geometrijom ovisi o načinu na koji je geometrija scene definirana. Geometrija scene može biti definirana pravilima ili diskretnim podacima.

I kontekstu računalne grafike su diskretni podaci češći i njihova je primjena proširenija jer pravilima zadan oblik često zahtijeva veću razinu truda za postizanje rezultata koji su neprimjetno bolji od diskretnih za mnoge vrste primjena.

Pravilima zadana geometrija zauzima manje prostora za pohranu kod jednostavnih geometrijskih tijela poput sfera, cilindara, diskova, stožaca i dr. pa je njena primjena bolja za takve slučajeve. Slaganjem različitih jednostavnijih tijela se mogu postići puno složeniji oblici korištenjem Booleovih operatora (engl. *boolean operators*), no neke iznimno nepravilne površine je i dalje teško ispravno prikazati na ovaj način pa je diskretna geometrija često jednostavniji odabir. Iako se svako tijelo može izraziti pravilima, diskretna način pohrane može biti znatno jednostavniji za izračun od rješavanja iznimno složenih jednadžbi za svaki presjek zrake s geometrijom (ali i provjeru sudara).

Praktični dio ovog seminarskog rada koristi pravilima zadanu sferu koja je prikazana u priloženom kôdu 1, jer je provjera kolizije, te udaljenosti kolizije zrake sa sferom jednostavna.

```
pub struct Sphere {
    pub pos: Vec3,
    pub radius: f32,
}

impl Intersect<Ray> for Sphere {
    type Result = f32;

    fn intersect(&self, ray: &Ray) -> Option<f32> {
        let direction = ray.origin - self.pos;
        let ray_magnitude_squared = ray.direction.length_squared();
        let alignment = 2.0 * direction.dot(ray.direction);
        let max_travel = direction.length_squared() - self.radius * self.radius;
        let discriminant_squared = alignment * alignment - 4.0 *
            ray_magnitude_squared * max_travel;

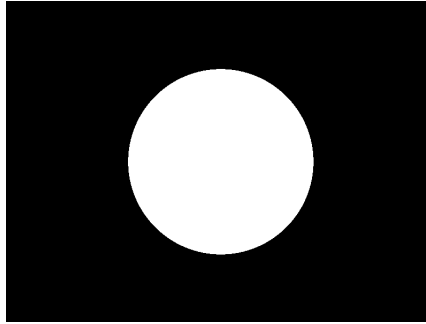
        if discriminant_squared > 0.0 {
            let discriminant = discriminant_squared.sqrt();
            let t1 = (-alignment - discriminant) / (2.0 * ray_magnitude_squared);
            if t1 > 0.0 {
                return Some(t1)
            }
            let t2 = (-alignment + discriminant) / (2.0 * ray_magnitude_squared);
            if t2 > 0.0 {
                return Some(t2)
            }
        }

        None
    }
}
```

Kôd 1: Provjera kolizije zrake sa sferom i udaljenosti kolizije

Kod provjere presjeka je za složenije scene s mnogo geometrije ili složenom geometrijom praktično koristiti akceleracijske strukture za obilazak sadržajem scene (engl. *scene traversal acceleration structures*). Česte su primjene hijerarhije omeđujućih volumena (engl. *bounding volume heirarchy*, BVH) i Kd-stabla (engl. *Kd-tree*) jer dopuštaju potpuno izbjegavanje zahtjevnih izračuna za kolizije. [5, dio 4.]

Provjerom kolizija se može grubo iscrtati sadržaj scene kao što je prikazano za sferu u slici 5.



Slika 5: Prikaz zraka koje presijecaju sferu (bijelo) i koje ne (crno)

5. Svjetlo, boje i kolorimetrija

U fizičkom smislu, svjetlost se može razmatrati istovremeno kao val elektromagnetskog zračenja i skup kvantnih čestica koje nazivamo fotonima. Drugi oblik je jednostavniji za simulaciju pa se pretpostavlja u kontekstu RT algoritama i ovom seminarskom radu. Bitna karakteristika fotona je njegova **valna duljina** (engl. *wavelength*).

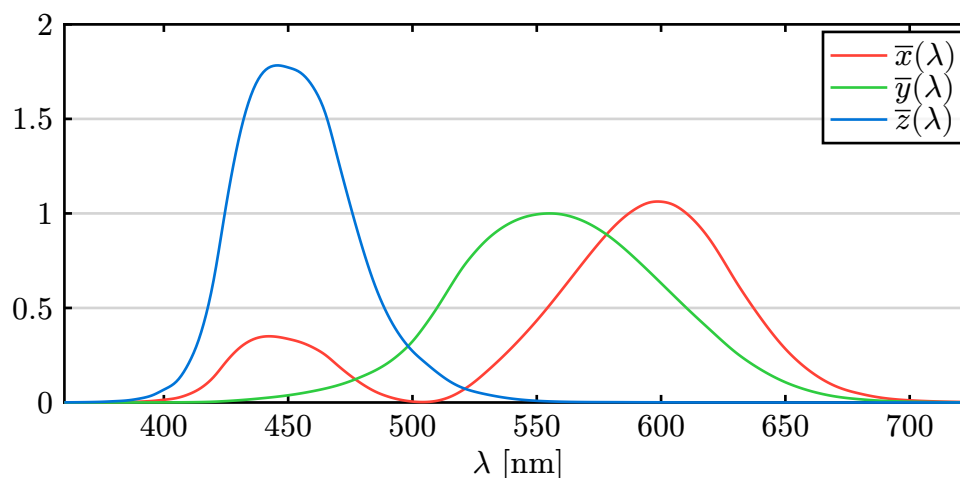
Kada razmatramo skup fotona u jednom snopu svjetlosti, grupiramo sve valne duljine u njemu sadržanih fotona u **spektar valnih duljina** (engl. *wavelength spectrum*), a brojnost fotona pojedinih valnih duljina predstavlja intenzitet radijacije (engl. *radiation intensity*) za tu valjnu duljinu koji je izražen kao $\frac{\mu\text{mol}}{\text{m}^2\text{s}}$, tj. broj 10^6 fotona koja prođu kroz površinu od 1m^2 u jednoj sekundi.

Spektar valnih duljina utječe na percipiranu boju svjetlosti ako se nalazi u rasponu od 400nm do 700nm.

U kontekstu RT aplikacija, boje mogu biti pohranjene kao zrake svjetlosti kada je željena točnija simulacija, no za svrhe prikaza modela, u računalnim igricama i animaciji su češće pohranjenje **diskretno** u prostoru boja spremnom za prikaz poput sRGB.

1931. godine je internacionalna komisija za osvjetljenje (fr. *Commission internationale de l'éclairage*, CIE) formirala prostor boja (engl. *color space*) koji opisuje kako "standardni promatrač" (engl. "*standard observer*") vidi različite nijanse boja u vezi s podražajima fotoosjetljivih receptora ljudskog oka. Taj prostor boja je utemeljen na mjerenjima koja su proveli William David Wright i John Guild 1920-ih godina, i zove se XYZ.

Slika 6 prikazuje preslikavanje valnih duljina svjetlosti u XYZ prostor boja.



Slika 6: CIE funkcije za preslikavanje valne duljine u XYZ. Izvor: [6]

Iako je udaljen od načina na koji se boje često predstavljaju u digitalnom obliku (sRGB), ima prednost jer može u potpunosti opisati cijeli spektar boja koje ljudsko oko vidi. Zbog toga se

koristi za predstavljanje boja u aplikacijama kod kojih je bitno da mogu ispravno upravljati bojama koje će kasnije biti prikazane korisniku.

Za jednostavnije RT algoritme je diskretan način predstavljanja boja praktičniji za uporabu jer zauzima manje prostora (4B po zraci) i zahtjeva manje izračuna prilikom interakcija, naspram pohrane realističnih podataka o zrakama koji imaju veće zahtjeve (ovisno o razlučivosti pohranjenog spektra).

No postoje primjene gdje je neophodno koristiti realističnu reprezentaciju jer daje puno točnije rezultate. Ona je neizbježna za točnu simulaciju:

- interakcije svjetla s prizmama koje drugačije usmjeravaju svjetlost ovisno o njenoj valnoj duljini (disperzija),
- interakcije svjetla s materijalima koji drugačije apsorbiraju i/ili fluoresciraju svjetlost različitih valnih duljina.

Pretvorba iz XYZ prostora boja u sRGB prostor se provodi jednostavnom linearnom transformacijom koja je prikazana u kodu 2. Memoizacijom je moguće izbjeći potrebu za učestalim izračunom u svrhu pretvorbe boja.

```
static CIE_TO_RGB: Mat3 = Mat3::from_cols_array(&[
    3.2406255, -0.9689307, 0.0557101,
    -1.537208, 1.8757561, -0.2040211,
    -0.4986286, 0.0415175, 1.0569959,
]);

impl From<CieXyz> for SrgbU8 {
    fn from(value: CieXyz) -> Self {
        let result: Vec3 = CIE_TO_RGB * Into::::into(value);
        SrgbU8 {
            r: (result.x * 255.) as u8, g: (result.y * 255.) as u8,
            b: (result.z * 255.) as u8, a: (value.a * 255.) as u8,
        }
    }
}
```

Kôd 2: Pretvorba XYZ boja u sRGB

Od inicijalne valne duljine je dobivena XYZ boja linearnom interpolacijom susjednih vrijednosti iz tablice preslikavanja prikazanoj u slici 6.

Konačno, množenjem intenziteta zrake s XYZ bojama se dobivaju vrijednosti koje je potrebno normalizirati za prikaz. Dinamički raspon (engl. *dynamic range*) uzorkovanih boja boja će biti veći nego što simulirani senzor kamere može uhvatiti, i također veći nego što je moguće prikazati na zaslonu.

6. Kamera

Senzor, leća i otvor koje kamera koristi igraju značajnu ulogu u konačnom prikazu slike.

Senzor utječe na kvalitete slike u koje ubrajamo:

- razlučivost (engl. *resolution*),
- dinamički raspon,
- dubinu boja (engl. *color depth*),
- razinu šuma (engl. *noise*) u okruženjima niskog osvjetljenja,
- vidno polje (engl. *field of view*, FoV), i dr.

Leća, kao i njena udaljenost od sensora utječu na:

- vidno polje,
- dubinsku oštrinu (engl. *depth of field*, DoF),
- aberacije boje (engl. *chromatic aberrations*),
- distorzije, i

- druge karakteristike.

Aparatura (engl. *aperture*) kontrolira i utječe na:

- duljinu ekspozicije (engl. *exposure*),
- vinjetu (engl. *vignette*),
- boke (jap. ぼけ, *boke*),
- uvećanje, i
- druge karakteristike.

Može se također koristiti i kamera s otvorom malog radiusa (engl. *pinhole camera*), koja daje zamućenu sliku s izraženom vinjetom.

RT algoritmi mogu uzeti sve te karakteristike kamere u obzir kako bi izmijenili način na koji je scena prikazana, no pretežno se modelira samo nekolicina njih. Zbog jednostavnosti je u praktičnom dijelu ovog seminara korištena statična scena i idealna kamera u perspektivi s vidnim poljem horizontalnog raspona od 90°. Korištena kamera također ne prati duljinu ekspozicije, nego se pretpostavlja da su svi fotoreceptori senzora osvijetljeni istovremeno konstantnom jačinom svjetlost.

7. Materijali

Izgled materijala geometrije u sceni ovisi o brojnim svojstvima samog materijala kao i mediju u kojem se on i kamera nalaze.

Potpuno ispravna simulacija interakcije zraka svjetlosti s materijalima je nepraktična jer su mjerenja nekih svojstva materijala iznimno spor proces koji zahtjeva skupu opremu. Također, određena svojstva nije moguće dobro izmjeriti za neke materijale pa je potrebno koristiti aproksimacije. Simulacije koje se oslanjaju na veliku količinu svojstva materijala također zahtijevaju vrlo snažnu opremu i/ili puno vremena.

Iz tih razloga se za RT u realnom vremenu nastoji pojednostaviti ključna svojstva koja imaju utjecaj na konačan izgled materijala na osnovne koje značajno pridonose konačnom izgledu. Također, materijale se grupira ovisno o njihovom generalnom izgledu (mat, plastični, ...) kako bi se daljnje pojednostavio izračun.

Osnovna svojstva materijala mogu biti:

- osnovna/albedo boja,
- metaličnost površine,
- gruboća/hrapavost površine,
- indeks refrakcije za spekularnu refleksiju i refrakciju,
- prozirnost,
- svojstva hoda prilikom raštrkavanja svjetlosti u materijalu,
- boja emisije,
- boja i debljina obloga (engl. *film*), no i
- neka druga.

U praktičnom dijelu rada nisu korišteni materijali te je pretpostavljeno da je površina 2-sfere savršeno ogledalo.

7.1. Teksture

Materijali često nemaju uniformna svojstva po svojoj cijeloj površini (npr. boja, hrapavost, ...). Iz tog razloga je poželjno složiti RT algoritam koji dopušta upravljanje individualnim svojstvima materijala pomoću tekstura.

8. Izvori i interakcije svjetlosti

Svijetlo je emitirano (engl. *emission*) iz izvora. U grafici to aproksimiramo idealnim reprezentacijama svjetlosti, no u stvarnosti svijetlo emitiraju različiti materijali pod utjecajem nekih kemijskih ili fizičkih procesa.

U svrhu pojednostavljivanja simulacije, koriste se 3 osnovna izvora svjetlosti:

- točkasti izvor (engl. *point light*),
- usmjereni izvor (engl. *directional light*), i
- ambijentno osvjetljenje (engl. *ambient light*).

U nekim slučajevima se može modelirati i reflektor, no on je specijalizirana verzija točkastog izvora koji ima ograničen smjer emisije na neki zadani kut, te ponekad prigušenje prema rubovima.

Kod unazadnih RT algoritama se dio geometrije smatra neosvijetljenim ako putanja uzoraka ne završava u izvoru svjetlosti. Konačno osvjetljenje za neosvijetljene dijelove scene je ambijentalno (ako se koristi).

U stvarnosti ambijentalno osvjetljenje ne postoji nego je ono proizvod indirektne svjetlosti obližnjih izvora ili sunca. No postizanje ambijentalnog osvjetljenja na taj način nije praktično jer bi zahtijevalo glomazan broj rekurzija algoritma što bi učinilo RT nepraktično sporim.

Usmjereni izvori se skoro nikada ne pojavljuju u prirodi, no u računalnoj primjeni se koriste za izvore svjetlosti koji su dovoljno udaljeni od sjenčane geometrije scene da su emitirane zrake svjetlosti gotovo paralelne. Zbog ograničenja hardvera, tj. pohrane decimalnih bojeva i pogrešaka pri računu s iznimno malim ili velikim vrijednostima istih, modeliranje vrlo dalekih izvora svjetlosti ne bi davalo točne rezultate.

Usmjereni snop svjetlosti se može postići jedino uz pomoć polarizacijskih filtera ili stimuliranom emisijom (npr. laseri).

Točkasti izvor svjetlosti je najbliži stvarnim (spontanim) izvorima, iako stvarni izvori ne emitiraju svjetlost istog intenziteta u svim smjerovima s iste pozicije.

U praktičnom dijelu je modeliran usmjereni izvor koji je opisan jednostavno vektorom smjera.

8.1. Refleksija

Zbog toga što na svaku uzorkovanu točku može djelovati svjetlost iz različitih smjerova, za svaku uzorkovanu točku je (u idealnoj implementaciji) potrebno rekurzivno uzorkovati dolazeću svjetlost (L_i) iz svih točaka koje čine površinu jedinične sfere \mathcal{S}_2 centrirane oko točke presjeka prethode zrake s geometrijom. Kada je materijal neproziran ili je od interesa samo refleksija, dovoljno je uzorkovati polusferu (\mathcal{H}_2) čija ravna stranica je tangenta na površinu, a zakrivljena je udaljenija od ravne.

Formula za izračun konačnog osvjetljenja koje je vidljivo u nekoj točki je:

$$L_0(p, \omega_0) = \underbrace{L_e(p, \omega_0)}_{\text{emitirani sjaj}} + \int_{\mathcal{H}^2} \underbrace{f(p, \omega_0, \omega_i)}_{\text{BRDF}} \underbrace{L_i(p, \omega_i)}_{\text{dolazeći sjaj}} |\cos(\theta_i)| d\omega_i \quad (1)$$

gdje je:

- p neka obasjana točka koju promatramo,
- ω_0 fazor koji označava smjer iz kojeg je točka p promatrana,
- $L_0(p, \omega_0)$ ukupan sjaj koji napušta točku p u smjeru ω_0 (engl. *outgoing radiance*),
- $L_e(p, \omega_0)$ sjaj kojeg sam materijal emitira (engl. *emitted radiance*) u smjeru ω_0 ,

- integral $\int_{\mathcal{H}^2} \dots d\omega_i$ djeluje kao **težinski zbroj vrijednosti podintegralnog umnoška** za sve smjerove $d\omega_i$ iz kojih može doprijeti svjetlost. Integrira po površini jedinične 2-polukugle \mathcal{H}^2 , te se sastoji od:
 - $f(p, \omega_0, \omega_i)$ dvosmjerne funkcije distribucije refleksije i transmisije (engl. *Bidirectional Reflectance Distribution Function*, BRDF),
 - $L_i(p, \omega_i)$ je sjaj koji dolazi u točku p od drugih izvora svjetlosti (engl. *incoming radiance*), te odbijanjem od reflektivnih površina, te konačno
 - $|\cos(\theta_i)|$ je geometrijsko prigušenje (engl. *geometric attenuation*) koje osigurava da je svjetlost koja se reflektira u smjeru ω_0 najizraženija za savršeni kut refleksije a smanjuje se za pliće kuteve.

Jer je uzorkovanje svih mogućih zraka koje pridonose osvjetljenju nemoguće (jer ih je beskonačno mnogo), koriste se Las Vegas ili Monte Carlo aproksimacije za određivanje podintegralnog izraza. Ove aproksimacije pojednostavljaju problem određivanja stvarne vrijednosti integrala na određivanje nasumično odabranih uzoraka. [5, dio 13.]

U kontekstu određivanja dolazeće svjetlosti se koristi tehnika koja se zove “ruski rulet”. Kod ruskog ruleta, uzorkuje se nekoliko zraka svjetlosti umjesto svih zraka (za cijelu jediničnu 2-sferu) te pridonos “pobjedničkih” zraka dijeli s $1 - P$, gdje je P vjerojatnost da će promatrana zraka biti otklonjena iz izračuna. [5, dio 13., str. 787]

Dobro je za “nasumično odabrane” zrake za daljnji hod odabrati zrake s manjim geometrijskim prigušenjem (sličnog smjera kao zraka savršene refleksije), jer će one generalno dati bolju aproksimaciju dolazeće svjetlosti.

Kôd 3 prikazuje izračun korišten za određivanje vrijednosti uzoraka na 2-sferi koja je savršeno zrcalo.

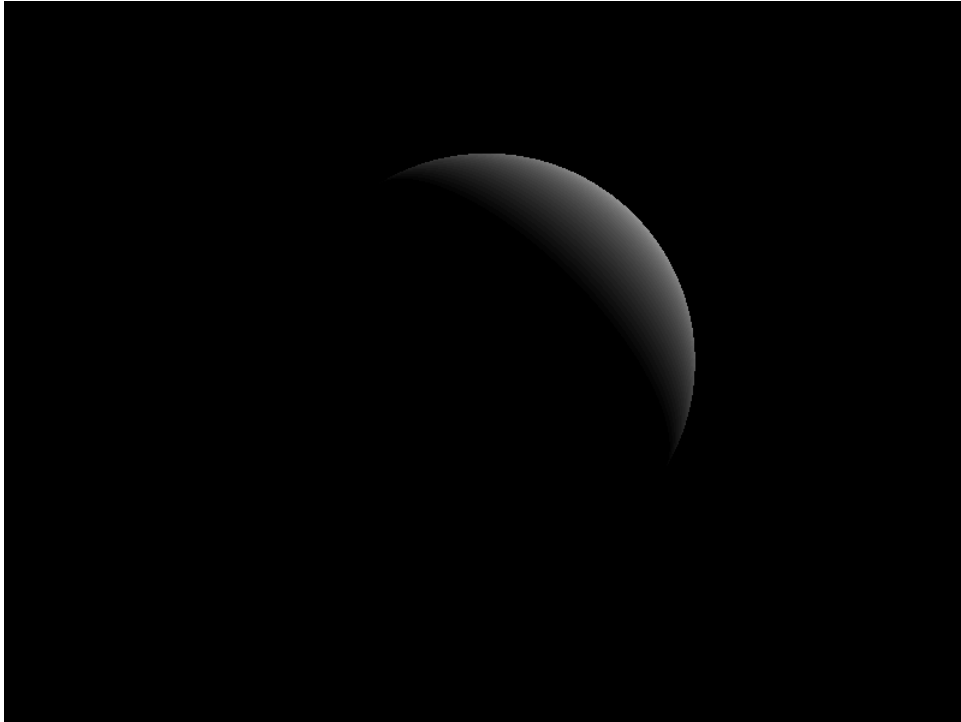
```
let target = ray.source.expect("camera ray must have target");

if let Some(distance) = sphere.intersect(&ray) {
  let hit_point = ray.origin + (ray.direction * distance);
  let normal = (hit_point - sphere.pos).normalize();
  let light_intensity = normal.dot(directional_light).max(0.0);

  let color = (255.0 * light_intensity) as u8;
  img.put_pixel(target.x, target.y, Rgb([color, color, color]));
} else {
  img.put_pixel(target.x as u32, target.y as u32, Rgb([0, 0, 0]));
}
```

Kôd 3: Izračun refleksije

Konačan prikaz praktičnog rada je prikazan u slici 7.



Slika 7: Konačan prikaz

9. Otklanjanje buke/šuma

Kod RT algoritama je osim u najjednostavnijim slučajevima potrebno provesti otklanjanje buke/šuma (engl. *noise*). Buka je rezultat korištenja Monte Carlo/Las Vegas aproksimacija koje uvode nasumičnost uzorkovanja u svrhu ubrzanja izračuna. Radi se o velikom nedostatku ovog načina prikaza grafike kojeg je u principu moguće samo mitigirati uporabom brojnih algoritmi za poboljšavanje rezultata ili provođenjem dovoljno/vrlo velikog broja uzoraka gdje ta buka postaje manje primjetna.

U zadnje vrijeme počinju se primjenjivati tehnike iz strojnog učenja u svrhu otklanjanja buke kako bi se ona otklonila blizu realnog vremena.



Slika 8: Prikaz rezultata prije i nakon uklanjanja buke. Izvor: [7]

10. Usporedba s klasičnom rasterizacijom

Zbog duge primjene, za klasične metode rasterizacije (nadalje rasterizacija) je razvijen velik broj tehnika za postizanje različitih efekata koji su primjetni u stvarnosti. Mnogi od tih efekata su nešto jednostavniji za postignuti uporabom RT algoritama no zbog toga su zahtjevniji za provođenje.

10.1. Odrizi

Odrizi (engl. *reflections*) se u klasičnim metodama rasterizacije izvode tako da se scena prije rasterizacije glavnog prikaza prvobitno prikaže s pozicije koja je zrcaljena pozicija kamere s obzirom na reflektivnu površinu. Ovisno o sadržaju scene, ova tehnika može biti iznimno zahtjevna, pogotovo kada je u istom kadru vidljivo više reflektivnih površina. U nekim slučajevima je moguće provesti prikaz pomoću trikova poput rekreacije zrcaljenog sadržaja scene unutar/iza ogledala što značajno umanjuje ili potpuno otklanja zahtjevnost dodatnog prikaza jer je zaseban prikaz (engl. *render pass*) potreban samo ako ogledalo treba sadržavati dinamične dijelove scene koji nisu unaprijed poznati.

10.2. Ambijentalna okluzija

Ambijentalna okluzija (engl. *ambient occlusion*, AO) se pojavljuje kada geometrija sadrži kuteve manje od 180° , a postaje zamjetljivija kod oštrijih kutova. Slika 9 prikazuje istu scenu s (a) i bez (b) AO - vidi se zamjetna razlika u kvaliteti prikaza zbog dobivenih osjećaja dubine.

Kod rasterizacije je ambijentalna okluzija u prostoru zaslona (engl. *screen-space ambient occlusion*, SSAO) tehnika koja daje iznimno uvjerljive rezultate a usporedno je jeftinija od njoj prethodećih tehnika koje su koristile stvarnu geometriju scene. [8, SSAO]



(a) bez SSAO

(b) sa SSAO

Slika 9: Usporedba scene sa i bez ambijentalne okluzije. Izvor: [8, SSAO]

SSAO tehnika ima nedostatak što je ne razumije svojstva materijala geometrije na koju je primjenjena. Zbog toga i reflektivne površine dobivaju jednake sjene okluzije iako su one u stvarnosti manje izražene jer reflektivni kutovi “zarobe” manje svjetlosti.

Kod RT tehnika se ambijentalna okluzija pojavljuje prirodno primjenom formule za refleksiju (1) ako površina nije savršeno zrcalo. No za tupe kutove može zahtijevati značano povećanje broja uzoraka.

10.3. Sjene

Sjene se u rasterizaciji se mogu postignuti na mnogo različitih načina ovisno o željenoj kvaliteti, i oni variraju u algoritamskoj složenosti, količini utrošene radne memorije, i drugim karakteristikama. Jedan od klasičnih pristupa je prije glavnog prikaza, prikazati scenu iz perspektive svakog od izvora svjetlosti te pohraniti dubinu svih tijela koja mogu bacati sjenu u zaseban spremnik dubine (engl. *depth buffer*) te ih potom koristiti pri konačnom prikazu za prikaz tamnije sjenčanje fragmenata koji su u sjeni. [9]

Nedostatak ovog pristupa je što zahtjeva iznimno veliku rezoluciju spremnika dubine kako bi sjene za predmete koji su udaljeniji od izvora svjetlosti izgledale dobro. Također, potreban je dodatan trud kako bi se osiguralo da sjene ispravno postaju zaglađenije što su udaljenije od geometrije koja ih baca (ili oštrije-bliže).

U slučaju RT algoritama se radi o jednostavnijoj značajki za implementaciju - potrebno je povećati broj uzoraka kako bi se postigao bolji izgled sjena na rubovima. Tu je korisno koristiti RT algoritam koji dopušta promjenjiv broj zraka za određene dijelove scene.

11. Zaključak

Iako ray tracing metode pružaju mnoga unaprijeđenja u realizmu prikazanih scena, nisu "čaroban metak"¹. Njihova implementacija zahtjeva iznimno puno truda kako bi se pokrile sve namjene, te postoji puno mjesta gdje je potrebno .

¹rješenje koje ne treba alternative jer zadovoljava sve moguće slučajeve primjene

Literatura

- [1] P. W. Ford, „New ray tracing scheme“, *JOSA*, sv. 50, izd. 6, str. 528–533, 1960.
- [2] R. Smith, „Quadro No More? NVIDIA Announces Ampere-based RTX A6000 & A40 Video Cards For Pro Visualization“. Pristupljeno: 21. rujan 2024. [Na internetu]. Dostupno na: <https://www.anandtech.com/show/16137/nvidia-announces-ampere-rtx-a6000-a40-cards-for-pro-viz>
- [3] J. Schmittler, I. Wald, i P. Slusallek, „SaarCOR: a hardware architecture for ray tracing“, u *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, 2002, str. 27–36.
- [4] Y. Qin, A. Michalowski, R. Weber, S. Yang, T. Graf, i X. Ni, „Comparison between ray-tracing and physical optics for the computation of light absorption in capillaries – the influence of diffraction and interference“, *Opt. Express*, sv. 20, izd. 24, str. 26606–26617, stu. 2012, doi: 10.1364/OE.20.026606.
- [5] M. Pharr i W. Jakob, *Physically Based Rendering, treće izdanje*. Morgan Kaufmann, 2016.
- [6] T. Smith i J. Guild, „The C.I.E. colorimetric standards and their use“, *Trans. Opt. Soc.*, sv. 33, izd. 3, str. 73–134, sij. 1931.
- [7] M. Mara, M. McGuire, B. Bitterli, i W. Jarosz, „An Efficient Denoising Algorithm for Global Illumination“, u *Proceedings: High Performance Graphics*, Los Angeles, California, USA: ACM, srp. 2017. doi: 10/gfzndq.
- [8] „LearnOpenGL“. Pristupljeno: 23. siječanj 2025. [Na internetu]. Dostupno na: <https://learnopengl.com/>
- [9] M. Holenko Dlab, *Modeli i postupci za sjenčanje i osvjetljavanje*. Sveučilište u Rijeci, Fakultet informatike i digitalnih tehnologija, 2022.