

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6648

# **FIZIKALNO TEMELJEN PRIKAZ OBLAKA**

Marko Mijolović

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6648

# **FIZIKALNO TEMELJEN PRIKAZ OBLAKA**

Marko Mijolović

Zagreb, lipanj 2020.

## ZAVRŠNI ZADATAK br. 6648

Pristupnik: **Marko Mijolović (0036505756)**

Studij: Računarstvo

Modul: Računarska znanost

Mentor: prof. dr. sc. Željka Mihajlović

Zadatak: **Fizikalno temeljen prikaz oblaka**

### Opis zadatka:

Proučiti vrste oblaka i fizikalno temeljenu interakciju svjetlosti s oblacima. Razraditi model volumetričkog modela oblaka i prolaz svjetlosti kroz ovakav medij. Ostvariti prikaz razrađenih modela. Na nizu primjera prezentirati ostvarene rezultate. Diskutirati utjecaj različitih parametara. Načiniti ocjenu rezultata i implementiranih algoritama. Izraditi odgovarajući programski proizvod. Koristiti programski jezik C++ i grafičko programsko sučelje OpenGL. Rezultate rada načiniti dostupne putem Interneta. Radu priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 12. lipnja 2020.



# SADRŽAJ

<b>1. Uvod</b> .....	<b>1</b>
<b>2. Fizika svjetlosti</b> .....	<b>2</b>
2.1. Radiometrijske veličine .....	2
2.1.1. Prostorni kut.....	2
2.1.2. Energija isijavanja.....	3
2.1.3. Tok isijavanja.....	3
2.1.4. Sjajnost.....	3
2.2. Fotometrijske veličine .....	4
2.2.1. Osvjetljenje .....	5
2.3. Interakcija svjetlosti s česticama u prostoru .....	6
2.3.1. Jednadžba transporta elektromagnetskog zračenja.....	7
2.3.2. Jednadžba volumetrijskog iscrtavanja.....	8
<b>3. Opis oblaka</b> .....	<b>11</b>
3.1. Klasifikacija oblaka .....	11
3.2. Interakcija svjetlosti s oblacima .....	12
<b>4. Modeliranje oblaka</b> .....	<b>15</b>
4.1. Funkcije šuma.....	15
4.1.1. Perlinov šum.....	15
4.1.2. Worleyev šum.....	17
4.1.3. Fraktalni šum.....	18
4.2. Funkcije za modeliranje oblaka .....	18
4.3. Položaj oblaka u sceni.....	19
4.4. Struktura oblaka .....	20
<b>5. Iscrtavanje oblaka</b> .....	<b>27</b>
5.1. Postupak pomicanja duž zrake .....	27

5.2. Fazna funkcija .....	28
5.3. Aproksimacija višestrukog raspršivanja .....	30
5.4. Aproksimacija ambijentnog osvjetljenja .....	32
5.5. Mapiranje tona slike .....	33
5.6. Gama ispravljanje .....	34
<b>6. Optimizacije .....</b>	<b>35</b>
6.1. Rani izlazak .....	35
6.2. Jeftino uzorkovanje .....	35
6.3. Plavi šum .....	35
<b>7. Rezultati .....</b>	<b>37</b>
<b>8. Zaključak .....</b>	<b>40</b>
<b>9. Literatura .....</b>	<b>41</b>
<b>Sažetak .....</b>	<b>43</b>
<b>Summary .....</b>	<b>44</b>

## 1. UVOD

Računalna grafika od samih početaka mnoštvo svojih algoritama temelji na fizikalnim teorijama. Međutim, tadašnja računala su bila jako spora te primjena algoritama i metoda temeljenih na fizikalnim teorijama nije bila izvediva u stvarnom vremenu. Zbog toga su razvijeni brojni fenomenološki modeli koji su aproksimirali stvarne fizikalne pojave s većom ili manjom vjerodostojnošću. Razvojem grafičkog sklopovlja mnoge metode koje prije nisu bile izvedive u stvarnom vremenu to postaju te se počinju primjenjivati i u interaktivnim aplikacijama kao što su računalne igre.

Zadnjih nekoliko godina većina novih računalnih igara na tržištu implementira neki od fizikalno temeljenih modela sjenčanja (engl. *physically based shading*). No velika većina tih igara fizikalno temeljene modele koristi samo za prikaz površina, dok su igre koje koriste fizikalne modele i za prikaz sudjelujućih medija (engl. *participating media*) kao što su vatra, dim, magla, oblaci itd. nešto rjeđe na tržištu.

Cilj ovog rada je istražiti i implementirati jedan takav fizikalni model prikaza. Konkretno u radu će biti napravljen fizikalno temeljen prikaz oblaka, no prikazane metode mogu se iskoristiti i za prikaz drugih sudjelujućih medija, uz promjenu odgovarajućih parametara.

Rad je podijeljen u devet poglavlja. U drugom poglavlju opisane su osnove radiometrije i interakcije svjetlosti s česticama prostora. Treće poglavlje opisuje fizikalnu teoriju oblaka – konkretnu problematiku kojom se rad bavi. U četvrtom poglavlju dan je postupak modeliranja oblaka na računalu za potrebe prikaza. Peto poglavlje opisuje kako se spomenuti model oblaka iscrtava. U šestom poglavlju su opisani neki od isprobanih optimizacijskih postupaka. Sedmo poglavlje komentira dobivene rezultate, dok osmo poglavlje iznosi zaključak rada.

## 2. FIZIKA SVJETLOSTI

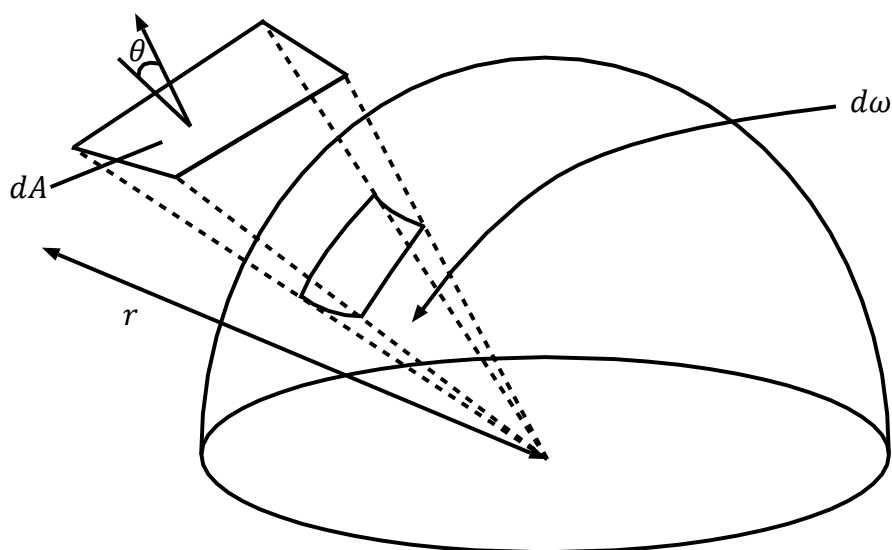
Predmet ovog rada je fizikalno temeljeno ostvarenje prikaza oblaka pa je zbog toga potrebno definirati neke osnovne pojmove kojima se kvantificira svjetlost. Dva sustava za opis i mjerenje svjetlosti su radiometrija i fotometrija.

### 2.1. Radiometrijske veličine

Ovo potpoglavlje definira potrebne radiometrijske veličine za razumijevanja rada. Valja napomenuti da je ovo samo mali podskup svih veličina koje radiometrija definira.

#### 2.1.1. Prostorni kut

Prostorni kut (engl. *solid angle*) je bezdimenzionalna fizikalna veličina koja opisuje količinu vidnog polja koju neki objekt pokriva, tj. to je fizikalna veličina koja opisuje koliko veliko neki objekt izgleda za promatrača. Mjerna jedinica prostornog kuta je steradian (*sr*). Uz oznake kao na slici Slika 2.1, matematička definicija prostornog kuta glasi:  $d\omega = \frac{dA \times \cos\theta}{r^2}$ , pri čemu je  $d\omega$  iznos diferencijalnog prostornog kuta,  $dA$  promatrana diferencijalna površina,  $\theta$  kut između normale na površinu  $dA$  i vektora od središta kugle do promatrane točke na površini, a  $r$  udaljenost do površine. Primjerice, prostorni kut kojeg zatvara cijela kugla je  $4\pi sr$ , dok je prostorni kut kojeg zatvara polukugla  $2\pi sr$ .



Slika 2.1 definicija prostornog kuta



### 2.1.2. Energija isijavanja

U radiometriji, energija isijavanja (engl. *radiant energy*) je pojam koji obuhvaća energiju elektromagnetskog zračenja. Oznaka energije isijavanja je  $Q$ , dok je mjerna jedinica energije isijavanja džul ( $J$ ). Primjerice, energija isijavanja Sunca je oko  $1.28 \times 10^{45} J$ .

### 2.1.3. Tok isijavanja

Tok isijavanja (engl. *radiant flux*) je količina energije isijavanja koja je emitirana, reflektirana, transmitirana ili primljena u jedinici vremena. Matematička definicija toka isijavanja dana je izrazom (2.1). Mjerna jedinica toka isijavanja je  $W = J \times s^{-1}$ .

$$\phi = \frac{\partial Q}{\partial t} \quad (2.1)$$

### 2.1.4. Sjajnost

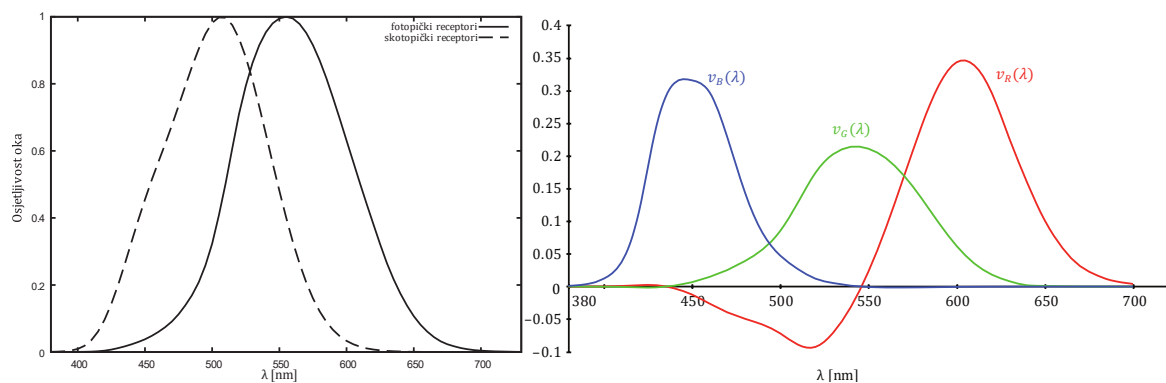
Sjajnost (engl. *radiance*) je količina toka isijavanja emitiranog, reflektiranog, transmitiranog ili primljenog od strane promatrane površine po jedinici prostornog kuta, po jedinici projicirane površine. Matematička definicija sjajnosti dana je izrazom (2.2), dok je mjerna jedinica sjajnosti  $W \times sr^{-1} \times m^{-2}$ . Općenito, sjajnost ovisi i o frekvenciji svjetlosti koja se promatra pa se definira i veličina nazvana spektralna sjajnost, no u računalnoj grafici se još uvijek najčešće ovisnost o frekvenciji svjetlosti izražava pomoću pripadnih RGB komponenata (pogotovo u interaktivnoj računalnoj grafici).

$$L = \frac{\partial^2 \phi}{\partial \omega \partial A \cos(\theta)} \quad (2.2)$$

## 2.2. Fotometrijske veličine

Razlika između radiometrije i fotometrije je u tome da radiometrija u svojim izrazima obuhvaća čitav spektar elektromagnetskog zračenja, dok se fotometrija fokusira samo na zračenje koje čovjek vidi.

Kada se mjeri zračenje na određenoj valnoj duljini, a koristi se fotometrijski sustav, u obzir se uzima težinska funkcija osjetljivosti oka na tu valnu duljinu. Ova funkcija zove se svjetlosna učinkovitost (engl. *luminous efficiency*) i označava se s  $v(\lambda)$ . Ona se može rastaviti na tri različite funkcije  $v_R(\lambda)$ ,  $v_G(\lambda)$  i  $v_B(\lambda)$  koje odgovaraju osjetljivosti ljudskog oka na crvenu, zelenu i plavu boju. Te funkcije su prikazane na slici Slika 2.2.



**(a)  $v(\lambda)$  za fotopičke receptore (čunjići) i za skotopičke receptore (štapići)**

**(b)  $v(\lambda)$  za crvenu, zelenu i plavu komponentu boje**

**Slika 2.2 Funkcije svjetlosne učinkovitosti ljudskog oka<sup>1</sup>**

Pretvorba iz radiometrijskih u fotometrijske veličine provodi se na način opisan izrazom (2.3). Pri tome  $F(\lambda)$  predstavlja fotometrijsku veličinu, a  $R(\lambda)$  radiometrijsku veličinu.

$$F(\lambda) = 683 \frac{\text{lm}}{\text{W}} \times R(\lambda) \times v(\lambda) \quad (2.3)$$

<sup>1</sup> Preuzeto iz [1] i [4]

RGB komponente zadane spektralne distribucije mogu se dobiti izrazima (2.4).

$$\begin{aligned}F_R &= 683 \frac{lm}{W} \int_{380 \text{ nm}}^{830 \text{ nm}} R(\lambda) \times v_R(\lambda) d\lambda \\F_G &= 683 \frac{lm}{W} \int_{380 \text{ nm}}^{830 \text{ nm}} R(\lambda) \times v_G(\lambda) d\lambda \\F_B &= 683 \frac{lm}{W} \int_{380 \text{ nm}}^{830 \text{ nm}} R(\lambda) \times v_B(\lambda) d\lambda\end{aligned}\tag{2.4}$$

### 2.2.1. Osvjetljenje

Osvjetljenje (engl. *luminance*) je fotometrijski analog sjajnosti te je to zapravo veličina koju ljudsko oko vidi [1]. To znači da je za prikaz scene na računalu potrebno za svaki slikovni element izračunati pripadajuće osvjetljenje. Oznaka za osvjetljenje je  $L_v$  (v je kratica od engl. *visual*, a navodi se da bi se razlikovale radiometrijske i fotometrijske veličine), a mjerna jedinica je  $cd \times m^{-2}$ .

### 2.3. Interakcija svjetlosti s česticama u prostoru

U stvarnom svijetu prostor se sastoji od velikog broja čestica koje utječu na fotone koji prolaze kroz prostor. Priroda te interakcije prostora i svjetlosti definira razdiobu sjajnosti u prostoru [2].

Ovisno o svojoj kompoziciji, prostor će različito utjecati na svjetlost koja kroz njega protječe. Gustoća čestica nekog dijela prostora može biti homogena (uniformna), npr. zrak ili voda, ili heterogena (ovisi o lokaciji u prostoru), npr. oblaci ili para.

Tradicionalno je u računalnoj grafici fokus bio na iscrtavanju čvrstih tijela koja imaju veliku gustoću čestica. Primjer takvih tijela su uobičajeni dielektrici ili metali koji se modeliraju pomoću funkcije distribucije dvosmjerne refleksije (engl. *bidirectional reflectance distribution function*). Takva tijela imaju neproziran izgled zbog svoje velike gustoće čestica te se zbog toga mogu vjerno iscrtavati pomoću navedene aproksimacije.

Primjeri prostora manje gustoće čestica su voda, magla, zrak, para itd. Takvi prostori ne mogu se modelirati pomoću funkcije distribucije dvosmjerne refleksije, već se oni modeliraju stohastički – skalarnim poljem vjerojatnosti sudara fotona s česticama prostora. Vjerojatnost sudara fotona je definirana pomoću koeficijenta  $\sigma(\vec{x})$  – funkcija gustoće vjerojatnosti sudara po jediničnoj udaljenosti. U općenitom slučaju, funkcija gustoće vjerojatnosti nije jednaka za sve valne duljine fotona, već definira spektralnu razdiobu koju u interaktivnoj računalnoj grafici uobičajeno izražavamo pripadnim RGB koeficijentima.

Svojstva prostora se definiraju pomoću koeficijenata apsorpcije i raspršenja, te pomoću fazne funkcije i emisijskog polja sjajnosti [2].

Apsorpcija (engl. *absorption*) je pojava sudara fotona s česticama medija pri kojima se energija elektromagnetskog zračenja u potpunosti pretvara u toplinu te je opisana koeficijentom apsorpcije  $\sigma_a$ .

Raspršenje (engl. *scattering*) je pojava promjene smjera kretanja fotona pri sudaru s česticama medija te je opisana koeficijentom raspršenja  $\sigma_s$  i faznom funkcijom  $f_p$ . Pojava raspršenja ne mijenja ukupan iznos sjajnosti, već samo raspoređuje sjajnost na više različitih smjerova.

Fazna funkcija (engl. *phase function*)  $f_p(\vec{x}, \vec{\omega}, \vec{\omega}')$ , definira kutnu distribuciju kojoj se podvrgava raspršenje te se za medije koji se sastoje od kuglastih čestica može parametrizirati kao funkcija jedne varijable – kuta između dva vektora smjera. Fazna funkcija mora biti normalizirana kako bi bila u skladu sa zakonom očuvanja energije, tj. mora vrijediti izraz (2.5).

$$\int_0^{2\pi} \int_0^\pi f_p(\theta) \sin\theta d\theta d\varphi = 1 \quad (2.5)$$

Također, važno svojstvo fazne funkcije je recipročnost – zamjenom ulaznog i izlaznog smjera vrijednost fazne funkcije se ne mijenja. Fazna funkcija općenito može biti izotropna (jednako raspršivanje u svim smjerovima) ili anizotropna (nejednoliko raspršivanje).

Emisija (engl. *emission*) fotona se događa kada čestice prostora dosegnu visoku razinu unutrašnje energije te se kao posljedica toga emitira foton. U principu, prostori koji posjeduju emisijska svojstva ponašaju se kao i svaki drugi izvori svjetlosti, no zbog svojih volumetrijskih svojstava modeliraju se odvojenim poljem sjajnosti  $L_e(\vec{x}, \vec{\omega})$ .

### 2.3.1. Jednadžba transporta elektromagnetskog zračenja

Distribucija sjajnosti u prostoru je definirana jednadžbom transporta elektromagnetskog zračenja [2]. Ona opisuje polje sjajnosti u ravnoteži  $L(\vec{x}, \vec{\omega})$  parametrizirano pozicijom  $\vec{x}$  i smjerom  $\vec{\omega}$ . Za njen cjelovit opis potrebno je prvo opisati dijelove od kojih se ona sastoji: apsorpcija, raspršenje prema van, raspršenje prema unutar i emisija.

Smanjenje sjajnosti u točki  $\vec{x}$  zbog apsorpcije je opisano jednadžbom (2.6) koja kaže da je derivacija sjajnosti u smjeru  $\vec{\omega}$  proporcionalna samoj sjajnosti, a da je faktor proporcionalnosti koeficijent apsorpcije  $\sigma_a$ .

$$(\vec{\omega} \cdot \vec{\nabla})L(\vec{x}, \vec{\omega}) = -\sigma_a(\vec{x})L(\vec{x}, \vec{\omega}) \quad (2.6)$$

Smanjenje sjajnosti u točki  $\vec{x}$  i smjeru  $\vec{\omega}$  zbog raspršenja prema smjerovima koji su različiti od  $\vec{\omega}$  (raspršenje prema van) je opisano jednadžbom (2.7). Smanjenje sjajnosti je opet proporcionalno samoj sjajnosti te je faktor proporcionalnosti koeficijent raspršenja  $\sigma_s$ .

$$(\vec{\omega} \cdot \vec{V})L(\vec{x}, \vec{\omega}) = -\sigma_s(\vec{x})L(\vec{x}, \vec{\omega}) \quad (2.7)$$

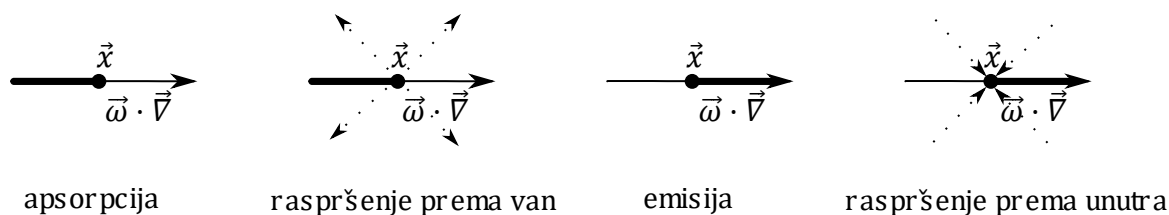
Povećanje sjajnosti u točki  $\vec{x}$  i smjeru  $\vec{\omega}$  zbog raspršenja iz svih ostalih smjerova  $\vec{\omega}'$  koji pridonose smjeru  $\vec{\omega}$  (raspršenje prema unutra) opisano je jednačbom (2.8). Pri tome  $S^2$  označava sfernu domenu integracije oko pozicije  $\vec{x}$ .

$$(\vec{\omega} \cdot \vec{V})L(\vec{x}, \vec{\omega}) = \sigma_s(\vec{x}) \int_{S^2} f_p(\vec{x}, \vec{\omega}, \vec{\omega}')L(\vec{x}, \vec{\omega}')d\vec{\omega}' \quad (2.8)$$

Povećanje sjajnosti zbog emisije je opisano jednačbom (2.9) koja kaže da je povećanje sjajnosti u točki  $\vec{x}$  i smjeru  $\vec{\omega}$  proporcionalno vrijednosti emisijskog polja sjajnosti u istoj točki i smjeru, a da je koeficijent proporcionalnosti koeficijent apsorpcije  $\sigma_a$ .

$$(\vec{\omega} \cdot \vec{V})L(\vec{x}, \vec{\omega}) = \sigma_a(\vec{x})L_e(\vec{x}, \vec{\omega}) \quad (2.9)$$

Ova četiri sastavna dijela jednačbe transporta elektromagnetskog zračenja prikazana su na slici Slika 2.3.



**Slika 2.3 Sastavni dijelovi jednačbe transporta elektromagnetskog zračenja**

Konačna jednačba transporta elektromagnetskog zračenja dobiva se zbrajanjem svih njenih dijelova. Uz uvođenje koeficijenta izumiranja (engl. *extinction*)  $\sigma_t = \sigma_a + \sigma_s$  dobiva se jednačba (2.10).

$$\begin{aligned} (\vec{\omega} \cdot \vec{V})L(\vec{x}, \vec{\omega}) &= -\sigma_t(\vec{x})L(\vec{x}, \vec{\omega}) + \sigma_a(\vec{x})L_e(\vec{x}, \vec{\omega}) \\ &+ \sigma_s(\vec{x}) \int_{S^2} f_p(\vec{x}, \vec{\omega}, \vec{\omega}')L(\vec{x}, \vec{\omega}')d\vec{\omega}' \end{aligned} \quad (2.10)$$

### 2.3.2. Jednačba volumetrijskog iscrtavanja

Jednačba transporta elektromagnetskog zračenja formulira distribuciju sjajnosti iz perspektive zrake svjetlosti te je zbog toga prikladna za metode konačnih elemenata kao što je isijavanje (engl. *radiosity*). Formulacija te jednačbe koja je prikladna za

metode praćenja zrake dobiva se formalnom integracijom jednadžbe transporta elektromagnetskog zračenja te se naziva jednadžba volumetrijskog iscrtavanja [2]. Radi jednostavnije notacije uvodi se oznaka za raspršenje prema unutra:

$$L_s(\vec{x}, \vec{\omega}) = \int_{S^2} f_p(\vec{x}, \vec{\omega}, \vec{\omega}') L(\vec{x}, \vec{\omega}') d\vec{\omega}' \quad (2.11)$$

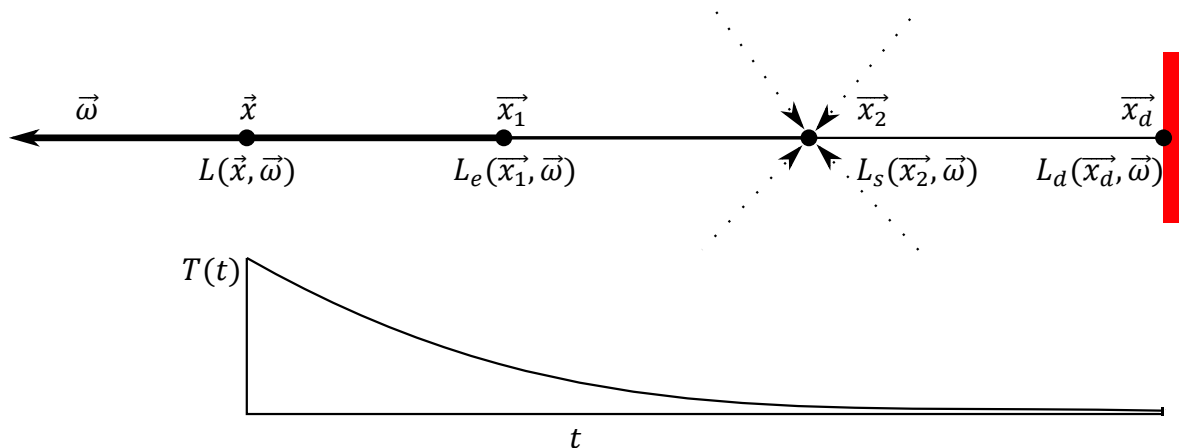
Također se uvodi oznaka za prozračnost (engl. *transmittance*) koja predstavlja mjeru smanjivanja sjajnosti zbog apsorpcije i raspršenja prema van između točaka  $\vec{x}$  i  $\vec{x}_t$ :

$$T(t) = \exp\left(-\int_{s=0}^t \sigma_t(\vec{x}_s) ds\right) \quad (2.12)$$

Jednadžba volumetrijskog iscrtavanja tada glasi:

$$L(\vec{x}, \vec{\omega}) = \int_0^d T(t) [\sigma_a(\vec{x}_t) L_e(\vec{x}_t, \vec{\omega}) + \sigma_s(\vec{x}_t) L_s(\vec{x}_t, \vec{\omega})] dt + T(d) L_d(\vec{x}_d, \vec{\omega}) \quad (2.13)$$

Prema [2], jednadžba volumetrijskog iscrtavanja može se interpretirati kao generalizacija jednadžbe iscrtavanja [3] koja uključuje i volumetrijske strukture, a čvrsta tijela predstavlja kao medije sa skoro beskonačnom gustoćom čestica i kompliciranim faznim funkcijama. Slika 2.4 prikazuje vizualizaciju jednadžbe volumetrijskog iscrtavanja koristeći notaciju iz jednadžbe (2.13).



Slika 2.4 Vizualizacija jednadžbe volumetrijskog iscrtavanja

Umjesto koeficijenata apsorpcije i raspršenja, moguće je odabrati i drugu parametrizaciju prostora koristeći prethodno definiran koeficijent izumiranja te novi parametar – koeficijent refleksije (engl. albedo)  $\rho$ :

$$\rho = \frac{\sigma_s}{\sigma_s + \sigma_a} = \frac{\sigma_s}{\sigma_t} \quad (2.14)$$

Vrijednost koeficijenta refleksije medija kreće se u rasponu  $[0, 1]$ . Kada je koeficijent refleksije blizu 0, većina svjetlosti se apsorbira te medij izgleda tamno, npr. ispušni plinovi. Vrijednost blizu 1 znači da se većina svjetlosti raspršuje, a ne apsorbira te rezultira medijem koji izgleda svijetlo, npr. zrak, oblaci itd.

Ta parametrizacija je ponekad prikladna jer je koeficijent refleksije prostora često konstantan te je samo skalarno polje koeficijenta izumiranja potrebno modelirati volumnim elementima (engl. *voxel*) ili proceduralnim funkcijama [2].

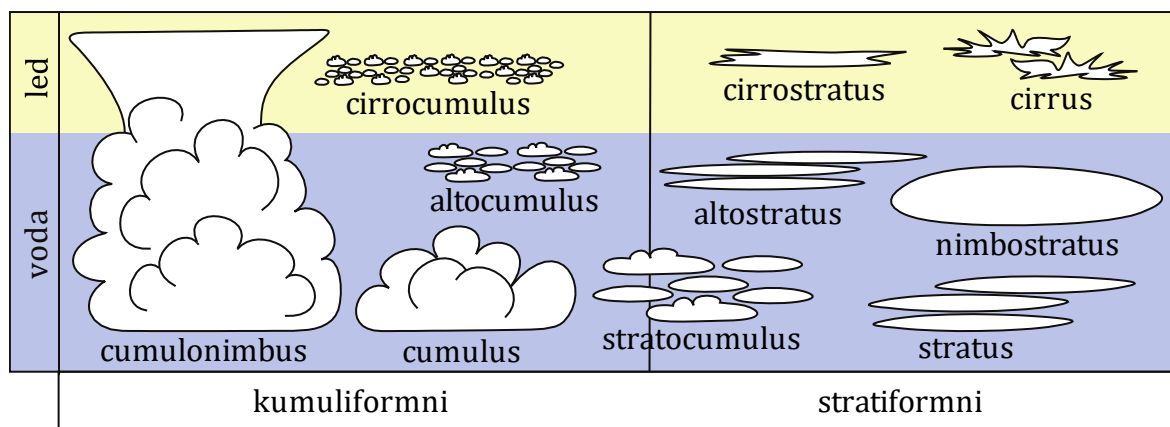


### 3. OPIS OBLAKA

Oblaci su vidljive nakupine vodenih kapljica, ledenih kristala ili ostalih čestica koje lebde u planetarnoj atmosferi. Na Zemlji nastaju kada se zrak zasiti vodenom parom, bilo zbog povećanja količine vodene pare u zraku ili zbog smanjenja temperature zraka ispod temperature rosišta. Pojavljuju se u zemljinoj homosferi koja se sastoji od troposfere, stratosfere i mezosfere. Oblaci u troposferi su najčešći, dok se oblaci u stratosferi i mezosferi formiraju jako rijetko, većinom u polarnim područjima.

#### 3.1. Klasifikacija oblaka

Oblaci u troposferi imaju latinska imena zahvaljujući opće prihvaćenoj nomenklaturi koju je napravio Luke Howard 1802. godine. On je podijelio oblake prema njihovom karakterističnom izgledu te prema visini na kojoj se pojavljuju u troposferi. Kombinirajući te dvije podjele, napravio je podjelu oblaka u 10 osnovnih tipova. Primjeri oblaka za svaki od tih tipova prikazani su na slici Slika 3.1.



Slika 3.1 Luke Howardova klasifikacija oblaka<sup>2</sup>

Imena rodova su formirana iz sljedećih korijena [4]:

- cumulo- znači da oblaci imaju karakterističan „napuhani“ (engl. *puffy*) izgled te su njihovi rubovi dosta oštri
- strato- znači da oblaci imaju karakterističan slojeviti (engl. *layered*) oblik, a rubovi su im često mekani, a znaju biti i pramenasti (engl. *wispy*)

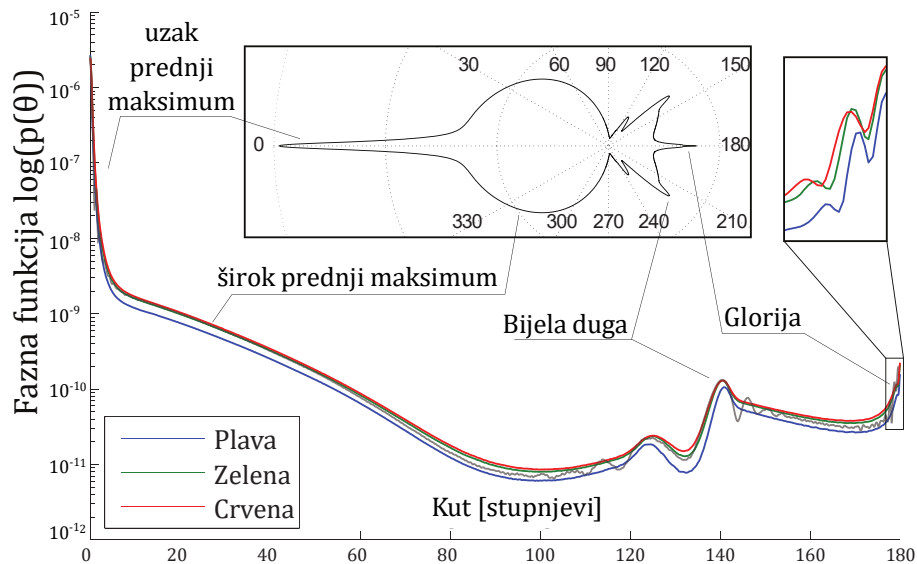
<sup>2</sup> Preuzeto iz [4]

- cirro- označava oblake od ledenih kristala koji se pojavljuju u gornjem dijelu troposfere
- nimbo- označava oblake koji donose kišu (dolazi u kombinaciji s cumulo- ili strato-)
- alto- označava visoke oblake koji se nalaze ispod cirro- oblaka

Ovi tipovi sadrže još i mnogo podtipova, a fokus u ovom radu će biti na tipove *stratus*, *stratocumulus* i *cumulus*.

### 3.2. Interakcija svjetlosti s oblacima

Osim strukturom, izgled oblaka je u velikoj mjeri određen i načinom na koji svjetlost interaktira s česticama koje tvore oblak. Taj način je određen faznom funkcijom medija, a u slučaju oblaka, za njeno računanje nije moguće koristiti metode geometrijske optike jer je radijus vodenih kapljica koje tvore oblake sumjerljiv s valnim duljinama vidljive svjetlosti. Zbog toga, za proračun fazne funkcije oblaka potrebno je koristiti Mie teoriju, koja daje numeričko rješenje Maxwellovih jednadžbi za raspršivanje elektromagnetskog zračenja od strane kuglastih čestica. Ova fazna funkcija ovisi o mnogo parametara kao što su temperatura zraka, valna duljina svjetlosti, radijus čestica itd. Ako se pretpostavi da su varirajući parametri samo valna duljina svjetlosti i radijus čestica, a da su ostali parametri konstantni, moguće je izračunati efektivnu faznu funkciju oblaka koja se može koristiti za svaku česticu u oblaku [4]. Primjer takve fazne funkcije generirane programom *MiePlot* [5] dan je na slici Slika 3.2.



**Slika 3.2 Logaritamski graf efektivnih Mie faznih funkcija za RGB komponente dobivenih uz korištenje gama distribucije vodenih kapljica<sup>3</sup>**

Kao što se vidi na slici, Mie funkcija primarno raspršuje fotone prema naprijed, ima veoma uzak maksimum za male kutove te veoma kompleksno ponašanje za kutove veće od 90 stupnjeva. Karakteristično primarno raspršivanje prema naprijed vidljivo je na slici Slika 3.3.



**Slika 3.3 Raspršivanje svjetlosti prema naprijed<sup>4</sup>**

<sup>3</sup> Preuzeto iz [4]

<sup>4</sup> Preuzeto s <https://www.flickr.com/photos/102617753@N04/14283793150>

Također je vidljivo da je Mie funkcija veoma ovisna o valnoj duljini svjetlosti te upravo zbog toga dolazi do karakterističnih pojava za oblake kao što su bijela duga (engl. *fogbow*) i glorija (engl. *glory*) čiji se učinci vide na slici Slika 3.4.



**Slika 3.4 Bijela duga i glorija<sup>5</sup>**

Glorija je optička pojava koja se očituje kao jedan ili više obojenih prstena na površini oblaka na suprotnoj strani od Sunca. Središte tih prstenova nalazi se na pravcu koji spaja Sunce i promatrača. Glorija nastaje kao posljedica raspršenja svjetlosti u interakciji s vodenim kapljicama.

Bijela duga je pojava slična klasičnoj dugi, samo što ona ima manje izražen spektar boja. Za razliku od klasične duge, bijela duga se formira zbog interakcije svjetlosti s vodenim kapljicama koje tvore maglu ili oblake, a zbog izrazito malih veličina tih kapljica, nije izražena separacija boja kao kod klasične duge.

---

<sup>5</sup> Preuzeto s <https://en.wikipedia.org/>

## 4. MODELIRANJE OBLAKA

Za potrebe modeliranja oblaka potrebno je definirati dvije stvari:

- Položaj oblaka u sceni
- Strukturu oblaka

Položaj u sceni modelira se pomoću geometrijskog primitiva koji definira volumen unutar kojeg se oblaci mogu pojavljivati i pomoću vremenske teksture koja definira položaj oblaka unutar tog volumena. U ovom radu, vremenska tekstura se generira proceduralno, pomoću funkcija šuma (engl. *noise functions*). Time je definiran položaj oblaka, no još je potrebno definirati i njihovu strukturu. To će također biti napravljeno pomoću funkcija šuma, no za razliku od vremenske teksture koja je dvodimenzionalna, teksture koje definiraju strukturu oblaka moraju biti trodimenzionalne. Ovaj pristup modeliranju oblaka inspiriran je postupkom opisanim u [6].

### 4.1. Funkcije šuma

U ovom radu koristi se kombinacija trodimenzionalnih Perlin i Worley funkcija šuma za generiranje modela oblaka. Kako su algoritmi generiranja šuma veoma zahtjevni za izračunavanje, funkcije šuma su izračunate unaprijed te spremljene u teksture koje se uzorkuju u sjenčaru zaduženom za iscrtavanje oblaka. Pri tome se vrijednost dobivena uzorkovanjem tekstura šuma koristi kao gustoća medija.

#### 4.1.1. Perlinov šum

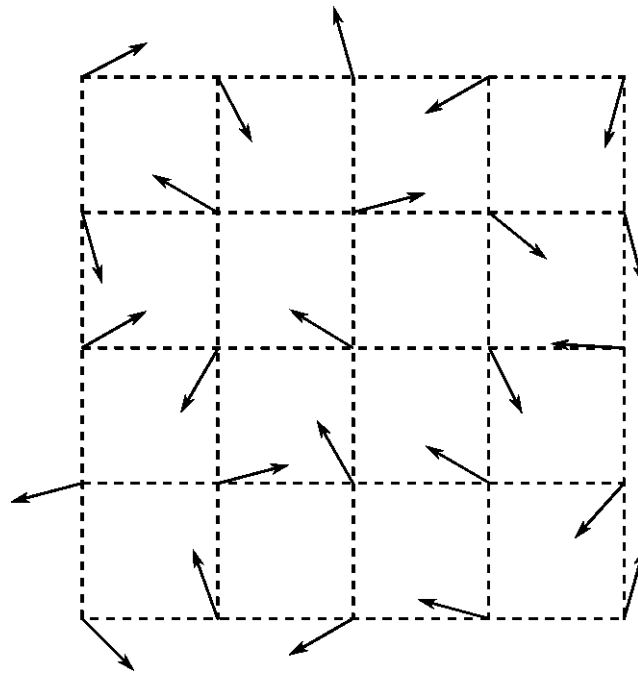
Perlinov šum je primjer gradijentnog šuma kojeg je osmislio Ken Perlin 1983. godine za prikaz raznih kompleksnih prirodnih pojava [7].

Perlinov šum se najčešće implementira kao dvodimenzionalna ili trodimenzionalna funkcija, ali može se definirati za bilo koji broj dimenzija. Algoritam se temelji na  $n$ -dimenzionalnoj rešetci. Svakoj točki rešetke pridruži se slučajani normalizirani  $n$ -dimenzionalni vektor koji predstavlja gradijent u toj točki. Primjer dvodimenzionalne rešetke je prikazan na slici Slika 4.1.

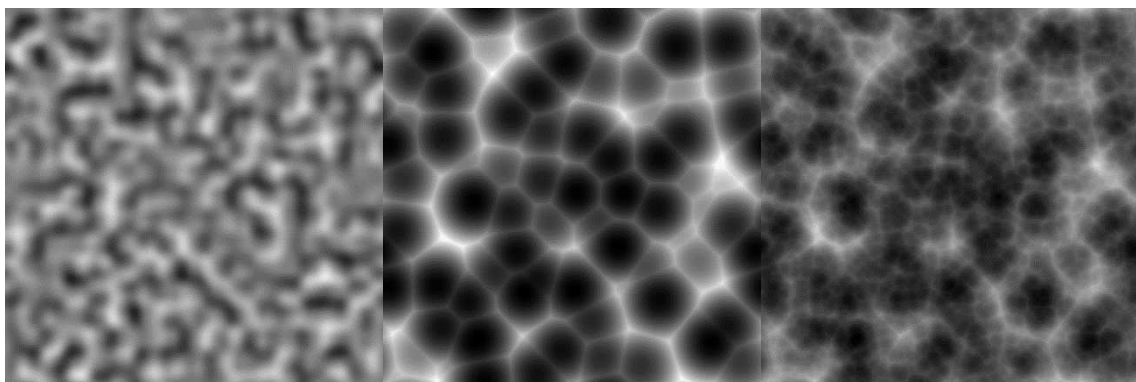
Kako bi se evaluirao Perlinov šum za određenu točku, pronade se polje u rešetci kojoj pripada točka. Uzmu se krajnje točke tog polja te se za svaku od njih izračuna skalarni

produkt gradijentnog vektora u toj točki i vektora razlike između točke za koju se računa šum i krajnje točke polja. Za dobivanje krajnjeg rezultata napravi se interpolacija svih dobivenih vrijednosti. Ako se točkama pridodaju gradijentni vektori u ponavljajućem uzorku, može se dobiti ponavljajuća tekstura (engl. *tileable*).

Primjer Perlinovog šuma u dvije dimenzije je prikazan na slici Slika 4.2 pod (a). Dobivena tekstura se čini nasumična, no ipak ima neku strukturu pa je zbog toga prigodna za prikaz raznih prirodnih pojava poput drva, mramora ili oblaka.



Slika 4.1 Dvodimenzionalna rešetka s gradijentnim vektorima



(a) Perlinov šum

(b) Worleyev šum

(c) Fraktalni šum

Slika 4.2 Primjeri tekstura šuma

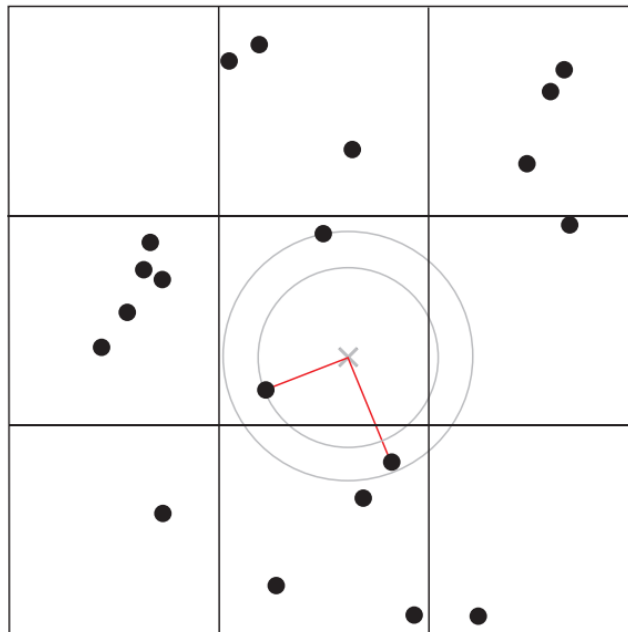
#### 4.1.2. Worleyev šum

Worleyev šum, ili stanični šum (engl. *cellular noise*), je primjer šuma zasnovanog na točkama za razliku od Perlinovog šuma koji je zasnovan na gradijentima. U računalnu grafiku uveo ga je Steven Worley 1996. godine [8].

Kao i Perlinov šum, može se implementirati kao dvodimenzionalna ili trodimenzionalna funkcija. Osnovna ideja je da se nasumično rasporede točke u prostoru te kada se želi evaluirati šum za neku točku, kao vrijednost šuma uzima se udaljenost do n-te najbliže točke.

Primjer prikazan na slici Slika 4.3 demonstrira proračun funkcije šuma za točku označenu s x. Na slici su prikazane dvije najbliže točke točki x te njihove pripadne udaljenosti. U ovisnosti koju udaljenost uzmemo kao vrijednost šuma, mogu se dobiti različite funkcije šuma. Najčešće se pak za n uzima da je jednak 1, odnosno kao vrijednost šuma uzima se udaljenost do najbliže točke.

Primjer Worleyevog šuma u dvije dimenzije prikazan je na slici Slika 4.2 pod (b).



Slika 4.3 Proračun vrijednosti staničnog šuma za odabranu točku<sup>6</sup>

<sup>6</sup> Preuzeto s <http://www.carljohanrosen.com/>

### 4.1.3. Fraktalni šum

Prethodno opisane funkcije Perlinovog i Worleyevog šuma čine temelj za daljnje modeliranje oblaka. Pomoću njih se gradi fraktalni šum koji uvjerljivo aproksimira pojave poput oblaka ili dima.

To se radi tako da se napravi težinska kombinacija više različitih tekstura šuma različitih frekvencija uzorkovanja. Ako su omjeri frekvencija različitih tekstura potencije broja 2, ponavljajuće svojstvo pojedinih tekstura će se zadržati i u konačnoj teksturi fraktalnog šuma. Broj različitih tekstura često se još zove i broj oktava.

Primjer teksture fraktalnog šuma dan je na slici Slika 4.2 pod (c). Za baznu teksturu je korištena tekstura Worleyevog šuma sa slike Slika 4.2 pod (b) te su na nju nadodane još dvije oktave Worleyevog šuma. Možemo primijetiti kako je osnovna struktura još uvijek prepoznatljiva, no teksturi su pridodani detalji viših frekvencija od osnovne teksture. Od tuda i dolazi naziv fraktalni šum.

## 4.2. Funkcije za modeliranje oblaka

Ovo potpoglavlje daje definicije funkcija koje će biti korištene u postupku modeliranja oblaka.

Funkcija *Remap* koja je prikazana izrazom (4.1.) preslikava vrijednosti iz originalnog intervala u novi interval. Pri tome  $v \in R$  je vrijednost koju treba preslikati,  $l_o \in R$  je donja, a  $h_o \in R$  gornja granica originalnog intervala,  $l_n \in R$  je donja, a  $h_n \in R$  gornja granica novog intervala.

$$Remap(v, l_o, h_o, l_n, h_n) = l_n + \frac{(v - l_o) \times (h_n - l_n)}{h_o - l_o} \quad (4.1.)$$

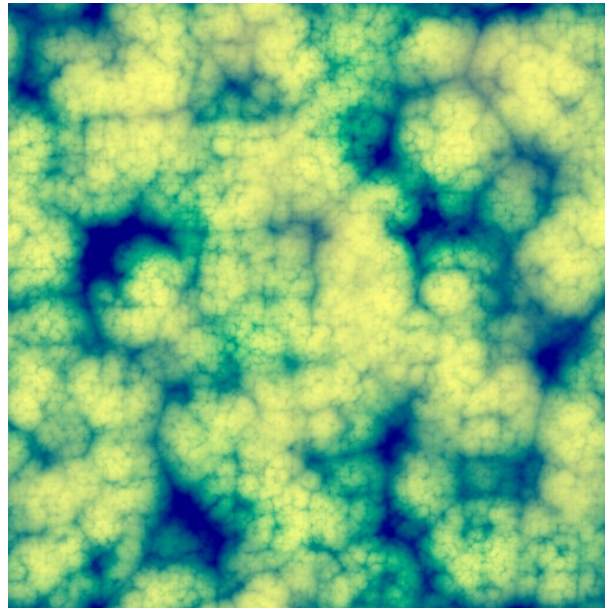
Funkcija *SAT* koja je prikazana izrazom (4.2.) služi za održavanje vrijednosti u intervalu  $[0, 1]$ .

$$SAT(v) = \begin{cases} 0, & \text{ako } v < 0 \\ 1, & \text{ako } v > 1 \\ v, & \text{inače} \end{cases} \quad (4.2.)$$

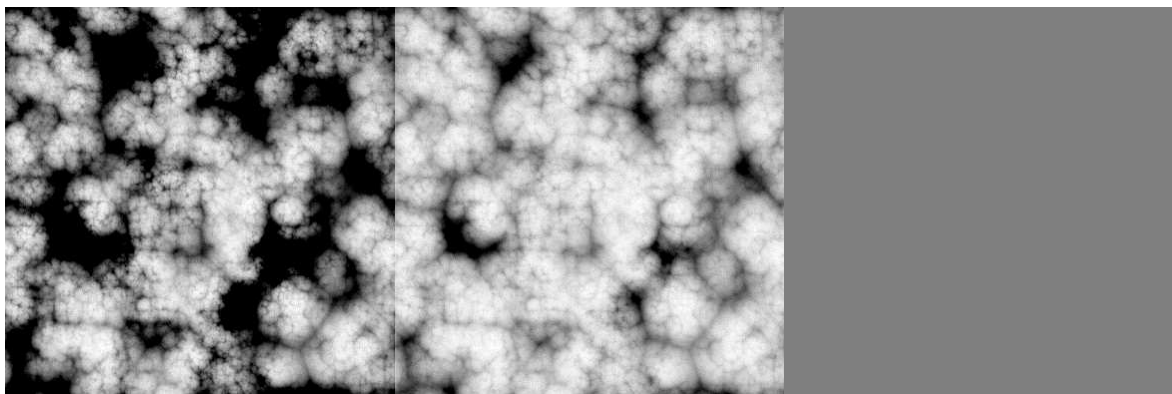


### 4.3. Položaj oblaka u sceni

Položaj oblaka u sceni definiran je pomoću nekoliko parametara. Svakoj točki u sceni pridružen je tekstel (engl. *texel*) u vremenskoj teksturi jednostavnom paralelnom projekcijom tako da se y koordinata odbaci. Primjer jedne korištene vremenske teksture je prikazan na slici Slika 4.4. Ista tekstura, ali prikazana po komponentama (crveni, zeleni i plavi kanal boje) prikazana je na slici Slika 4.5



Slika 4.4 Vremenska tekstura



(a) Crveni kanal

(b) Zeleni kanal

(c) Plavi kanal

Slika 4.5 Vremenska tekstura prikazana po komponentama (kanalima boje)

Kanal crvene boje sadrži parametar pokrivenosti oblacima (engl. *cloud coverage*) u pojedinoj točki. Ovaj parametar određuje prostornu razdiobu oblaka jer će on množiti vrijednost gustoće oblaka dobivene uzorkovanjem trodimenzionalnih tekstura što će biti objašnjeno u potpoglavlju 4.4. U konkretnom primjeru na slici Slika 4.5 u crveni kanal pohranjena je tekstura nastala postupkom kombiniranja dva fraktalna šuma (jedan temeljen na Perlinovom šumu, drugi na Worleyevom šumu).

Kanal zelene boje sadrži alternativnu vrijednost parametra pokrivenosti oblacima u pojedinoj točki. U konkretnom primjeru na slici Slika 4.5 u ovom kanalu je također sadržana tekstura nastala postupkom kombiniranja dva fraktalna šuma, no uz nešto drugačije parametre. Funkcija kojom se kombiniraju te dvije teksture prikazana je izrazom (4.3.).

$$LERP(Weather.r, Weather.g, GlobalCoverage) \quad (4.3.)$$

Pri tome, *LERP* označava linearnu interpolaciju, *Weather* vrijednost dobivenu uzorkovanjem vremenske teksture u trenutnoj točki, a *GlobalCoverage* parametar koji se može ručno podesiti ili koristiti u animaciji.

Kanal plave boje sadrži tip oblaka koji se nalazi u određenoj točki prostora. Značenja vrijednosti su sljedeće:

- 0.0 – tip oblaka je *stratus*
- 0.5 – tip oblaka je *stratocumulus*
- 1.0 – tip oblaka je *cumulus*

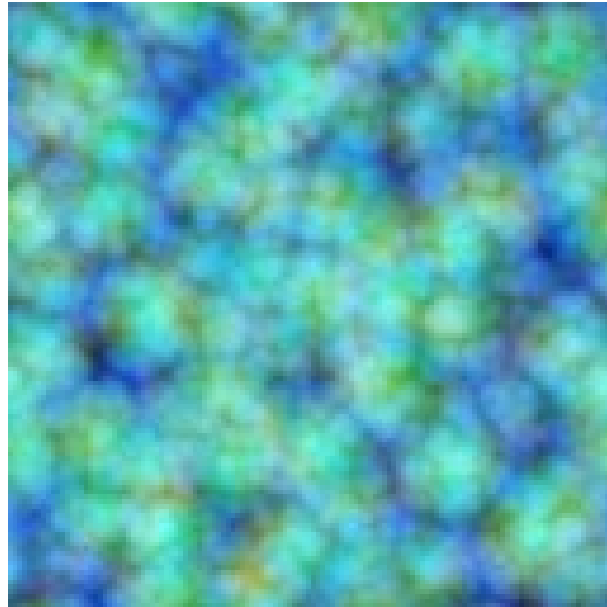
U konkretnom primjeru na slici Slika 4.4 svi oblaci u sceni su tipa *stratocumulus*.

#### 4.4. Struktura oblaka

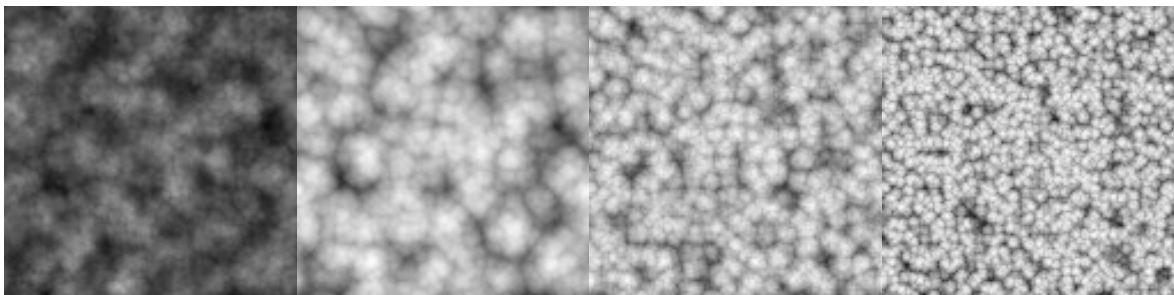
Jednom kada znamo položaj oblaka u sceni potrebno je definirati i strukturu oblaka. Struktura oblaka se definira pomoću dvije trodimenzionalne teksture šuma – jedna tekstura niže frekvencije definira osnovni oblik, a druga tekstura više frekvencije definira detalje oblaka – pomoću nje se vrši erozija osnovnog oblika oblaka.

Za definiranje osnovnog oblika koristi se trodimenzionalna tekstura dimenzija  $128 \times 128 \times 128$  piksela koja je prikazana na slici Slika 4.6. Tekstura po komponentama (kanalima boje) prikazana je na slici Slika 4.7. Crveni kanal teksture

sadrži kombinaciju Perlin i Worley šuma kojeg je Andrew Schneider nazvao Perlin-Worley šum [9], dok preostali kanali sadrže teksture fraktalnog šuma temeljenog na Worleyevom šumu (svaki sljedeći kanal sadrži frekvencije dva puta veće od prethodne).



**Slika 4.6** Volumna tekstura koja definira osnovni oblik oblaka



**(a)** Crveni kanal

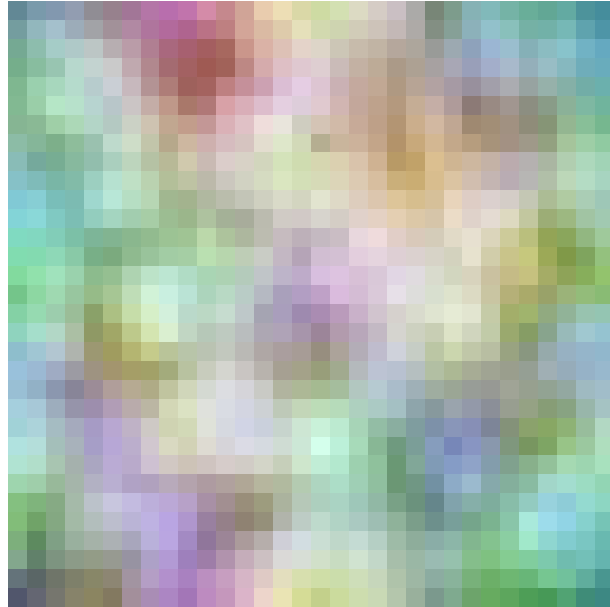
**(b)** Zeleni kanal

**(c)** Plavi kanal

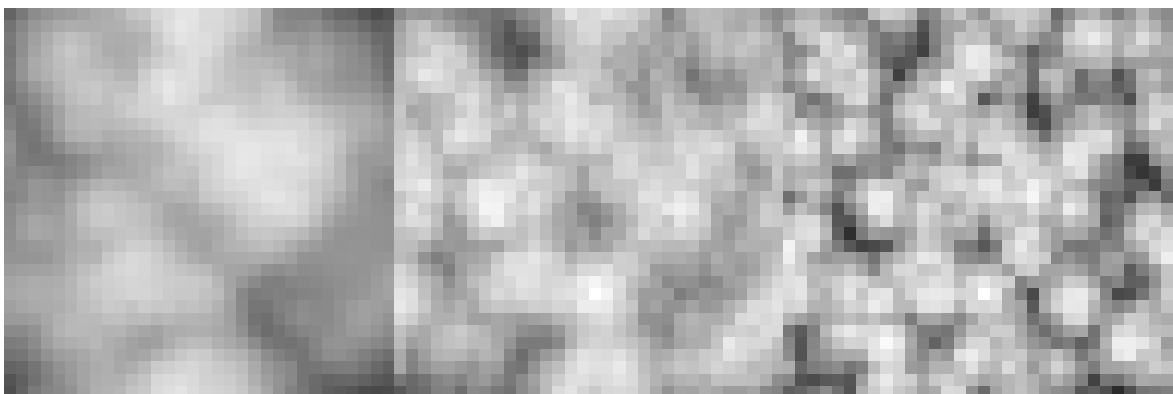
**(d)** Alfa kanal

**Slika 4.7** Volumna tekstura koja definira osnovni oblik oblaka prikazana po komponentama (kanalima boje)

Za definiranje detalja oblaka koristi se trodimenzionalna tekstura dimenzija  $64 \times 64 \times 64$  piksela koja je prikazana na slikama Slika 4.8 i Slika 4.9. Ona u svojim kanalima sadrži teksture fraktalnog šuma temeljene na Worleyevom šumu uz različite frekvencije.



**Slika 4.8** Volumna tekstura koja definira detalje oblaka



**(a)** Crveni kanal

**(b)** Zeleni kanal

**(c)** Plavi kanal

**Slika 4.9** Volumna tekstura koja definira detalje oblaka prikazana po komponentama (kanalima boje)

Izgradnja strukture oblaka počinje izrazom (4.4.).

$$LowFreqFBM = 0.625 \times LowFreq.g + 0.25 \times LowFreq.b + 0.125 \times LowFreq.a \quad (4.4.)$$

U izrazu (4.4.) *LowFreq* predstavlja četverokomponentni vektor dobiven uzorkovanjem volumne teksture koja definira osnovni oblik oblaka. Postupak se nastavlja izrazom (4.5.).

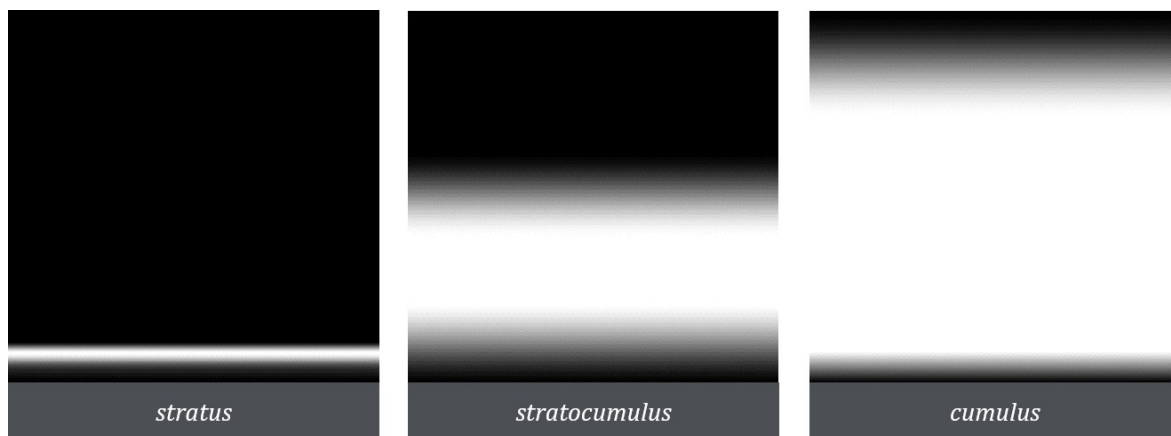
$$BaseCloud = SAT(Remap(LowFreqFBM, LowFreq.r, 1, 0, 1)) \quad (4.5.)$$

Nakon primijenjenog izraza (4.5.), dobivena je osnovna gustoća oblaka u točki. Ona se dalje modificira u skladu s pokrivenosti oblacima u toj točki. Pokrivenost oblaka u točki ovisi o vrijednosti vremenske teksture u toj točki i o visini točke u sloju oblaka te se računa na sljedeći način:

$$Coverage = LERP(Weather.r, Weather.g, GlobalCoverage) \times GetHeightCoverage(CurrentPoint.y, Weather.b) \quad (4.6.)$$

Funkcija *GetHeightCoverage* vraća vrijednost u skladu s gradijentima koji su prikazani na slici Slika 4.10. Ona računa relativnu visinu trenutne točke u sloju oblaka te u ovisnosti o vrijednosti tipa oblaka u trenutnoj točki uzorkuje gradijent tog tipa oblaka pomoću relativne visine. Gradijenti predstavljaju visinsku razdiobu pojedinih tipova oblaka u pojasu oblaka. Dobiveni parametar pokrivenosti oblacima se koristi na sljedeći način:

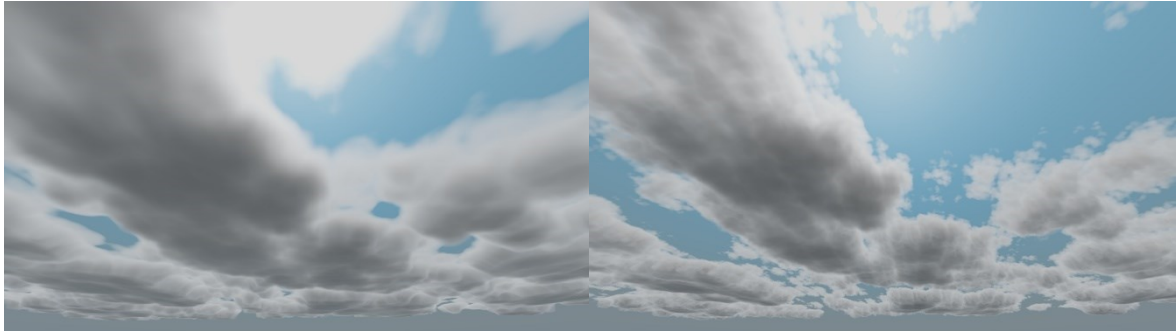
$$BaseCloudWithCoverage = SAT(Remap(BaseCloud, 1 - Coverage, 1, 0, 1)) \times Coverage \quad (4.7.)$$



Slika 4.10 Visinski gradijenti pojedinih tipova oblaka<sup>7</sup>

Nakon primijenjenog izraza (4.7.), osnovni oblik oblaka je postignut i to izgleda kao na slici Slika 4.11 pod (a).

<sup>7</sup> Preuzeto iz [9]



(a) Osnovni oblik

(b) Nadodani detalji na osnovni oblik

Slika 4.11 Različite faze postupka modeliranja oblaka

Na taj osnovni oblik dodaju se detalji na sljedeći način:

$$\begin{aligned}
 HighFreqFBM & \\
 &= 0.625 \times HighFreq.r + 0.25 \times HighFreq.g \quad (4.8.) \\
 &+ 0.125 \times HighFreq.b
 \end{aligned}$$

$$\begin{aligned}
 HighFreqModifier & \\
 &= LERP(HighFreqFBM, 1 \quad (4.9.) \\
 &- HighFreqFBM, SAT(GetRelativeHeight() \times 10))
 \end{aligned}$$

Pri tome funkcija *GetRelativeHeight* vraća relativnu visinu trenutne točke u odnosu na početnu visinu trenutnog tipa oblaka i koristi se kako bi vrhovi oblaka bili više pramenasti.

Konačna struktura oblaka dobiva se pomoću izraza (4.10.) te je prikazana na slici Slika 4.11 pod (b).

$$\begin{aligned}
 FinalCloud = & \\
 &SAT(Remap(BaseCloudWithCoverage, \quad (4.10.) \\
 &HighFreqModifier \times HighFreqFactor, 1, 0, 1))
 \end{aligned}$$

Dobiveni rezultati postupka modeliranja za različite tipove oblaka prikazani su na slikama: Slika 4.12 za oblake tipa *stratus*, Slika 4.13 za oblake tipa *stratocumulus* i Slika 4.14 za oblake tipa *cumulus*.

Valja napomenuti da je proces modeliranja isti za sve tipove oblaka, jedina razlika je u vrijednosti plavog kanala pohranjenog u vremenskoj teksturi.



**Slika 4.12** Oblaci tipa *stratus*



**Slika 4.13** Oblaci tipa *stratocumulus*



**Slika 4.14** Oblaci tipa *cumulus*

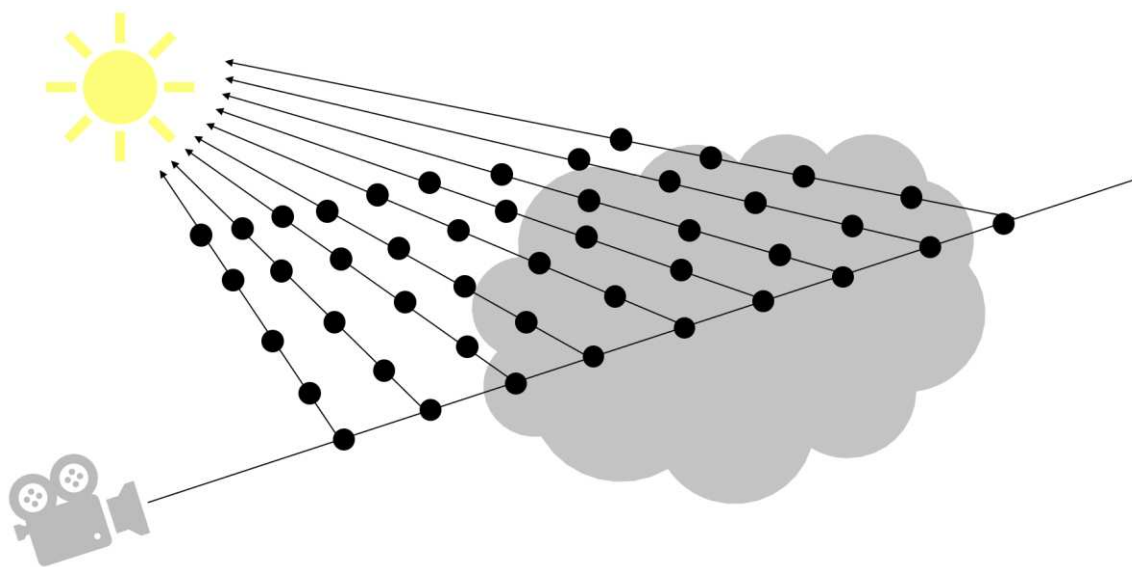


## 5. ISCRTAVANJE OBLAKA

Kao što je navedeno u potpoglavlju o osvjetljenju, za iscrtavanje oblaka potrebno je za svaki slikovni element odrediti pripadno osvjetljenje. Standardni način iscrtavanja grafičkih objekata modeliranih pomoću funkcije gustoće jest metodom pomicanja duž zrake [10].

### 5.1. Postupak pomicanja duž zrake

Postupak opisuje Slika 5.1. Ideja je da se iz očišta šalju zrake u scenu kroz svaki slikovni element. Pronalaze se sjecišta s geometrijskim primitivom (npr. kvadrom poravnatim s koordinatnim osima scene) u kojem se nalaze oblaci te se pomiče od početne točke do krajnje točke uz zadani broj koraka. U svakom koraku uzorkuje se gustoća oblaka te se akumulira optička gustoća (engl. *optical depth*), odnosno smanjuje prozračnost. Također se u svakom koraku šalje još jedna zraka prema Suncu da bi se utvrdilo osvjetljenje. Konačna prozirnost slikovnog elementa kroz koji se šalje zraka određena je isključivo gustoćom oblaka na zraci koja potječe iz očišta, a boja (RGB koeficijenti) isključivo gustoćom oblaka između pojedinih uzoraka i Sunca.



Slika 5.1 Pomicanje duž zrake<sup>8</sup>

<sup>8</sup> Preuzeto iz [11]

Pseudokod postupka dan je isječkom Isječak 5.1 koji zapravo predstavlja numeričku integraciju izraza (2.13). Pri tome vrijedi  $L_e(\vec{x}, \vec{\omega}) = 0, \forall \vec{x}, \vec{\omega}$  jer oblaci ne emitiraju svjetlost. Također je pretpostavljeno da su koeficijenti izumiranja ( $\sigma_t$ ) i raspršivanja ( $\sigma_s$ ) konstantni, a jedini parametar koji se mijenja je gustoća oblaka.

### Isječak 5.1 Pseudokod pomicanja duž zrake

```

for each pixel
    Transmittance = 1.0
    Luminance = Vec3(0.0)
    Ray = GetRayFor(Pixel)
    Start, End = GetStartAndEndPointFor(Ray)
    for each step along the ray
        Position = PositionFor(Step)
        Density = CloudDensityFor(Position)
        CurrentExtinctionCoeff = Density*ExtinctionCoeff
        CurrentScatteringCoeff = Density*ScatteringCoeff
        CurrentTransmittance = Exp(-CurrentExtinctionCoeff*StepSize)
        InScattering = GetInScatteringFor(Position, Ray)
        CurrentLuminance = CurrentScatteringCoeff*(InScattering
            - InScattering*CurrentTransmittance)/CurrentExtinctionCoeff
        Luminance += Transmittance*CurrentLuminance
        Transmittance *= CurrentTransmittance;
    end for
    FinalColour = BackgroundColour*Transmittance + Luminance
end for

```

## 5.2. Fazna funkcija

Jedna veoma popularna fazna funkcija u računalnoj grafici je Henyey-Greenstein fazna funkcija koja je preuzeta iz astronomije [12]. Njena matematička definicija glasi:

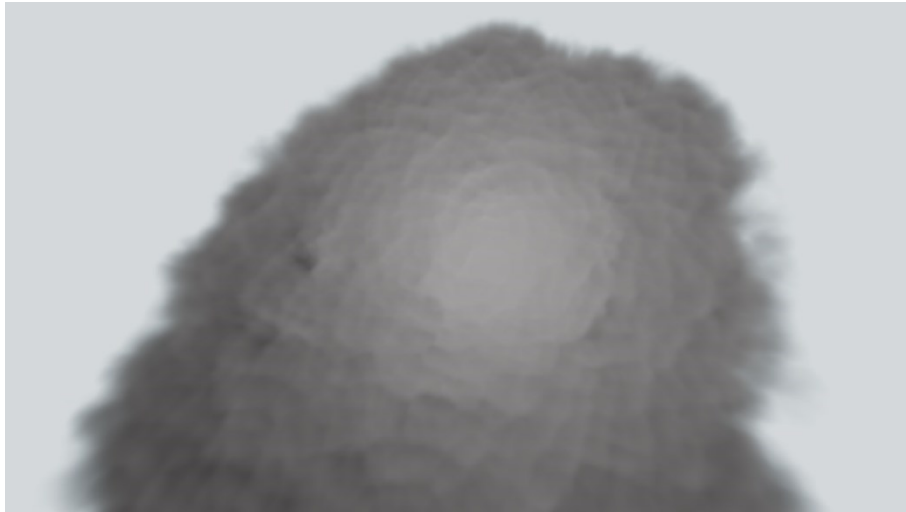
$$f_p(\theta, g) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g\cos\theta)^{3/2}} \quad (5.1.)$$

Podešavanjem parametra  $g \in [-1, 1]$ , ova funkcija može se koristiti za raspršivanje pretežito unatrag ( $g \in [-1, 0)$ ), unaprijed ( $g \in (0, 1]$ ) ili za izotropno raspršivanje ( $g = 0$ ). Također je moguće koristiti težinsku kombinaciju faznih funkcija s različitim

parametrom  $g$  pa time dobiti raspršivanje unaprijed i unatrag. Na slici Slika 5.2 prikazan je rezultat ako se koristi težinska kombinacija prikazana sljedećim izrazom:

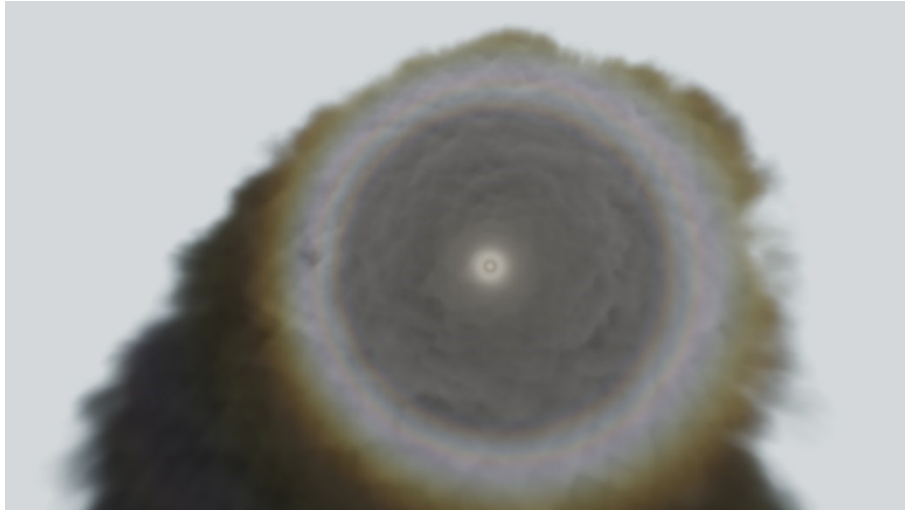
$$f_{p,total}(\theta) = 0.8 \times f_p(\theta, 0.9) + 0.2 \times f_p(\theta, -0.5) \quad (5.2.)$$

Vidimo da je moguće ugrubo modelirati raspršivanje unaprijed i unatrag, no ovom funkcijom ipak nije moguće modelirati kompleksne učinke bijele duge i glorije.



**Slika 5.2 Henyey-Greenstein fazna funkcija**

Isprobana je i fizikalno utemeljena Mie fazna funkcija koja je generirana pomoću programskog alata MiePlot [5] i spremljena u jednodimenzionalnu teksturu koja se potom uzorkuje u sjenčaru zaduženom za iscrtavanje oblaka. Rezultati korištenja te fazne funkcije vidljivi su na slici Slika 5.3 koja prikazuje uspješno reproduciranje učinaka bijele duge i glorije.



Slika 5.3 Mie fazna funkcija

### 5.3. Aproximacija višestrukog raspršivanja

Postupak sa slike Slika 5.1 uzima u obzir samo jednostruko raspršivanje (engl. *single scattering*), što može biti prihvatljiva aproksimacija za medije s malim koeficijentom refleksije. To nije slučaj za oblake, koeficijent refleksije oblaka je  $\approx 1$  što znači da oblaci gotovo svu upadnu svjetlost raspršuju, a jako malo apsorbiraju te je njihov koeficijent izumiranja dosta velik – za oblake tipa *stratus* izmjeren je  $\sigma_t \in [0.04, 0.06]$ , dok je za oblake tipa *cumulus* izmjeren  $\sigma_t \in [0.05, 0.12]$  [13].

Da bi simulirali višestruko raspršivanje, trebalo bi izmijeniti algoritam tako da na svakom koraku uzima u obzir zrake iz svih smjerova. To je veoma skup postupak i kao takav ne koristi se ni u filmskoj produkciji, a kamoli u računalnoj grafici u stvarnom vremenu. Višestruko raspršivanje moguće je aproksimirati tako da se umanjuje koeficijent izumiranja za zrake prema izvoru svjetla kako bi više svjetlosti došlo do točke koju se sjenča [14]. Umjesto jednog faktora skaliranja, moguće je napraviti sumaciju preko različitih faktora skaliranja. Također se može napraviti umanjenje ekscentriteta fazne funkcije  $g$  i koeficijenta raspršenja. Tada se sjajnost pojedine točke na zraci računa na sljedeći način:

$$L = \sum_{i=0}^{N-1} L_i \quad (5.3.)$$

$$L_i = \sigma_s b^i L_{light}(\vec{\omega}_i) f_p(\theta, c^i g_0) e^{-\alpha^i \int_0^t \sigma_t(s) ds} \quad (5.4.)$$

Pri tome su  $N$  („broj oktava“),  $a$  („slabljenje“),  $b$  („doprinos“) i  $c$  („slabljenje ekscentriteta“) parametri koje je moguće mijenjati. Slika 5.4 prikazuje izgled oblaka ako se u obzir uzima samo jednostruko raspršivanje. U tom slučaju samo će površina oblaka koja je okrenuta prema Suncu biti osvjetljena dok će ostatak oblaka biti skroz crn.



**Slika 5.4 Samo jednostruko raspršivanje**

Primjenom izraza (5.3.) i (5.4.) dobiva se rezultat prikazan na slici Slika 5.5. Pri tome je za jednostruko raspršivanje korištena Mie fazna funkcija, a za višestruko raspršivanje težinska kombinacija dvije Henyey-Greenstein fazne funkcije prikazane izrazom (5.2.).



**Slika 5.5 Višestruko raspršivanje**

## 5.4. Aproximacija ambijentnog osvjetljenja

Postupak sa slike Slika 5.1 također ne uzima u obzir ni ambijentno osvjetljenje. Za ispravnu simulaciju ambijentnog osvjetljenja opet bi bilo potrebno pratiti zrake u svim smjerovima, što je danas računski neizvedivo u stvarnom vremenu. Međutim, ambijentno osvjetljenje moguće je aproksimirati tako da se pretpostavi da oblak ima homogenu gustoću – vrijednost uzorkovane gustoće u trenutnoj poziciji. Također se pretpostavi da postoje dva izvora ambijentnog osvjetljenja – nebo i Zemlja i da je fazna funkcija izotropna (jednako raspršuje svjetlost u sve smjerove). Uz te pretpostavke jednačba volumetrijskog iscrtavanja ima analitičko rješenje [15].

U postojeći pseudokod iz isječka Isječak 5.1 ambijentno osvjetljenje se dodaje tako da se modificira funkcija *GetInScatteringFor* za koju je bilo pretpostavljeno da vraća samo raspršenu svjetlost od Sunca tako da vraća raspršenu svjetlost od Sunca, neba i Zemlje. Pseudokod računanja ambijentne komponente osvjetljenja dan je u isječku Isječak 5.2. Pri tome su *LuminanceBottom* i *LuminanceTop* vrijednosti osvjetljenja koje se dobiju uzimanjem prosječne vrijednosti od nekoliko uzoraka osvjetljenja neba i Zemlje.

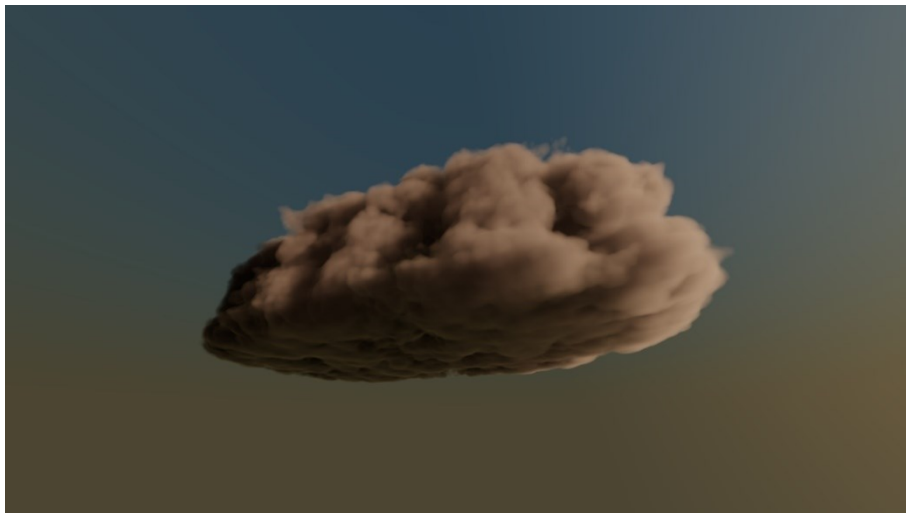
### Isječak 5.2 Računanje ambijentnog osvjetljenja

```
PhaseFunction = 1/(4*PI) // izotropna fazna funkcija
Height = GetHeightInCloudLayerFor(Position)
ScatteringBottom = 2*PI*LuminanceBottom*Exp(-CurrentExtinctionCoeff*Height)
DistToTop = GetDistanceToTopOfCloudLayerFor(Position)
ScatteringTop = 2*PI*LuminanceTop*Exp(-CurrentExtinctionCoeff*DistToTop)
TotalInScattering = PhaseFunction*(ScatteringBottom + ScatteringTop)
```

Slika 5.6 prikazuje učinak dodavanja ambijentnog osvjetljenja na scenu prikazanu slikom Slika 5.5. Slika 5.7 prikazuje kako se osvjetljenje može dinamički mijenjati. Vidljivo je kako oblak prilikom zalaska sunca poprima nijanse žutocrvene boje.



**Slika 5.6 S ambijentnim osvjetljenjem**



**Slika 5.7 Oblak prilikom zalaska sunca**

## **5.5. Mapiranje tona slike**

Pri korištenju fizikalno temeljenih modela osvjetljenja, izračunato osvjetljenje za pojedini slikovni element može poprimiti veliki raspon vrijednosti (od nekoliko  $cd \times m^{-2}$  za tamne scene, do nekoliko milijardi  $cd \times m^{-2}$  pri gledanju u sunce). Ove vrijednosti je potrebno mapirati u normalizirani raspon  $[0, 1]$  kako bi se mogle prikazati na standardnim ekranima. Ovaj postupak se zove mapiranje tona (engl. *tone mapping*). Postoji mnogo različitih metoda mapiranja tona, a izrazom (5.5.) prikazana je jedna jednostavna metoda temeljena na razini izlaganja zračenju [16]. Pri tome je

*Exposure* parametar koji se može odrediti iz fizikalnih parametara kamere i prosječnog osvjetljenja scene ili se može ručno podesiti.

$$FinalColour = vec3(1.0) - exp(-Colour * Exposure) \quad (5.5.)$$

## 5.6. Gama ispravljanje

Prilikom izračuna osvjetljenja, preslikavanja tekstura i drugih operacija pretpostavljamo da su vrijednosti s kojima radimo linearne, no treba imati na umu da spremnici ekrana koriste nelinearna mapiranja boja [17]. Veza između vrijednosti u spremniku ekrana i vrijednosti sjajnosti emitiranih iz ekrana dana je prijenosnom funkcijom ekrana. Prijenosna funkcija je dio sklopovlja ekrana te zbog toga postoje različiti standardi za monitore, televizije i projektore. Prilikom kodiranja linearnih vrijednosti boje za prikaz treba primijeniti inverz prijenosne funkcije ekrana kako bi se poništio njen nelinearan učinak. Ovaj postupak naziva se gama ispravljanje (engl. *gamma correction*).

Standardna prijenosna funkcija za monitore propisana je standardom prostora boje sRGB. Za preslikavanje linearnih vrijednosti u sRGB nelinearno kodirane vrijednosti, koristi se inverz sRGB prijenosne funkcije [17]:

$$y = f_{sRGB}^{-1}(x) = \begin{cases} 1.055x^{1/2.4} - 0.055, & x > 0.0031308 \\ 12.92x, & x \leq 0.0031308 \end{cases} \quad (5.6.)$$

Često se koristi i pojednostavljeni izraz (od kojega i dolazi naziv gama ispravljanje) s vrijednošću  $\gamma = 2.2$ :

$$y = f_{display}^{-1}(x) = x^{1/\gamma} \quad (5.7.)$$



## 6. OPTIMIZACIJE

Isprobano je nekoliko optimizacijskih postupaka kako bi se pokušalo smanjiti vrijeme potrebno za iscrtavanje oblaka.

### 6.1. Rani izlazak

Jedna jednostavna optimizacija je da se prestane pomicanje duž zrake jednom kada prozračnost padne ispod određenog praga. To se može napraviti jer svi dodatni koraci koji bi bili provedeni sve manje i manje doprinose konačnoj boji slikovnog elementa.

### 6.2. Jeftino uzorkovanje

Još jedna jednostavna optimizacija je da ako vrijednost dobivena uzorkovanjem osnovne volumne teksture oblaka iznosi nula, nema potrebe uzorkovati volumnu teksturu detalja oblaka, niti raditi proračune osvjetljenja.

### 6.3. Plavi šum

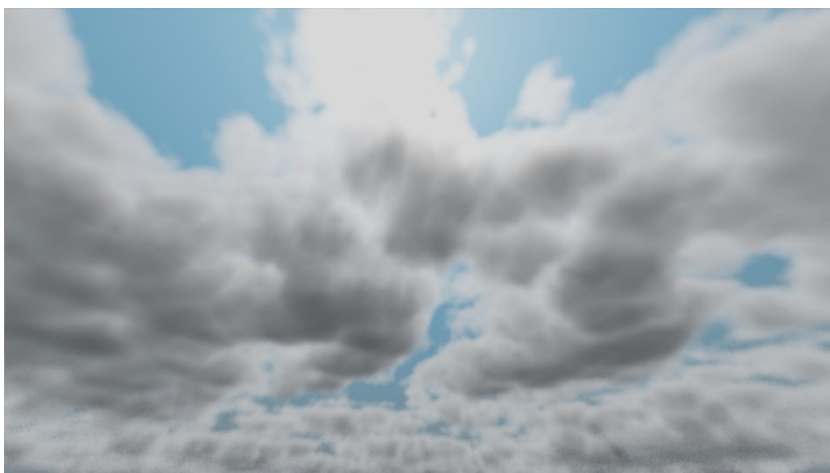
Za ispravno iscrtavanje potreban je veliki broj koraka. Jedna metoda kojom se može smanjiti broj koraka, a dobiti donekle prihvatljive rezultate je da se koristi plavi šum da se odmakne početna točka na zraci [18]. To je napravljeno tako da se za svaki slikovni element uzorkuje tekstura plavog šuma te se tako dobije slučajan vektor kojeg se pridoda početnoj poziciji na zraci. Time efektivno zamjenjujemo vezivanje boja (engl. *colour banding*) za šum, što je prihvatljivije ljudskom oku. Slika 6.1 pokazuje rezultat ako se koristi 64 koraka na zraci koja potječe iz očišta. Može se primijetiti izraženo vezivanje boja. Slika 6.2 prikazuje istu scenu iscrtanu korištenjem istog broja koraka, ali uz dodatno korištenje plavog šuma. Može se primijetiti izražen šum koji se može donekle ublažiti ako se na iscrtanu scenu primijeni Gaussovo zamućenje. Time se dobiva slika Slika 6.3.



**Slika 6.1 Bez plavog šuma**



**Slika 6.2 S plavim šumom**



**Slika 6.3 S plavim šumom i Gaussovim zamućenjem**

## 7. REZULTATI

U sklopu ovog rada napravljena je programska implementacija koja se izvršava na grafičkoj kartici u stvarnom vremenu.

Program je napisan u programskom jeziku C++ te za komunikaciju s grafičkom karticom koristi grafičko programsko sučelje OpenGL kroz biblioteku GLbinding. Za potrebe stvaranja OpenGL konteksta, stvaranja prozora te obradu korisničkog unosa koristi se biblioteka GLFW. Kao matematička biblioteka koristi se biblioteka GLM, a za potrebe učitavanja te spremanja tekstura s i na disk koristi se biblioteka stb\_image. Za izradu grafičkog korisničkog sučelja koristi se biblioteka Dear ImGui. Programi za sjenčanje su pisani u jeziku GLSL.

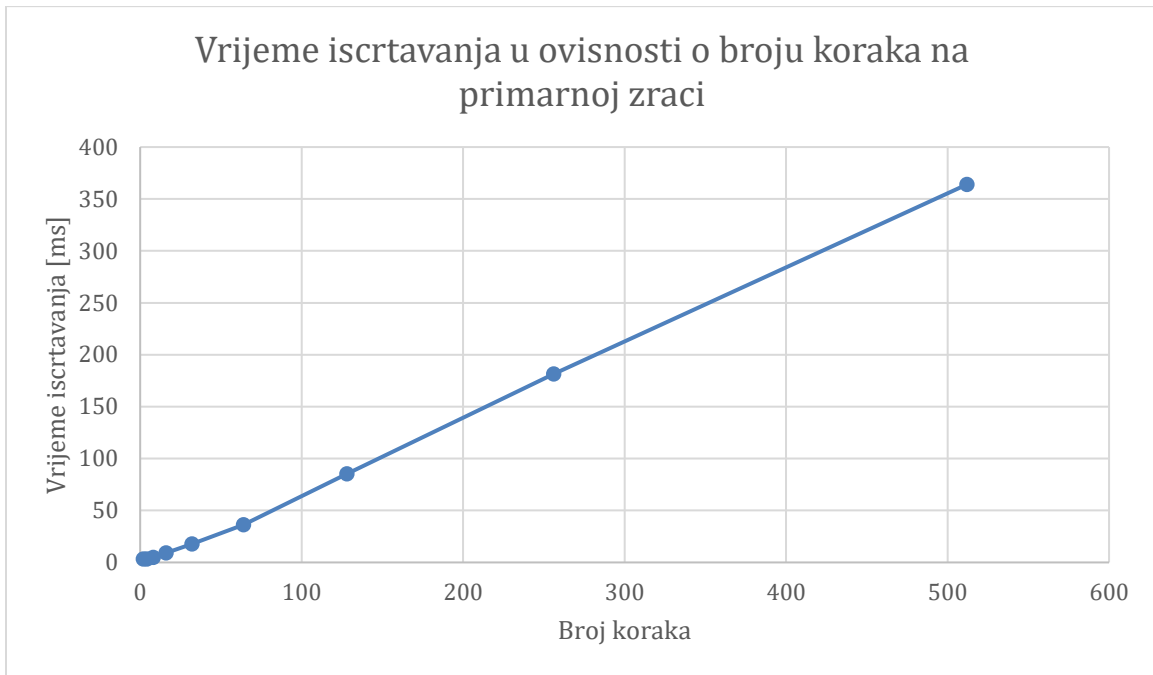
Cjelokupan projekt s pripadnim programskim kodovima te korištenim teksturama dostupan je na Gitlabu<sup>9</sup>.

Programska implementacija testirana je na računalu s AMD Radeon 5700 XT grafičkom karticom, na rezoluciji  $1280 \times 720$ . Slika 7.1 prikazuje ovisnost vremena iscrtavanja o broju koraka na primarnoj zruci (pri tome je broj koraka na sekundarnim zrakama postavljen na 16), a Slika 7.2 prikazuje vrijeme iscrtavanja u ovisnosti o broju koraka na sekundarnim zrakama (pri tome je broj koraka na primarnoj zruci postavljen na 64). Iz navedenih slika vidljivo je da je ovisnost u oba slučaja više-manje linearna, što je i očekivano. Također je vidljivo da je postupak dosta zahtjevan čak i za moderne grafičke kartice.

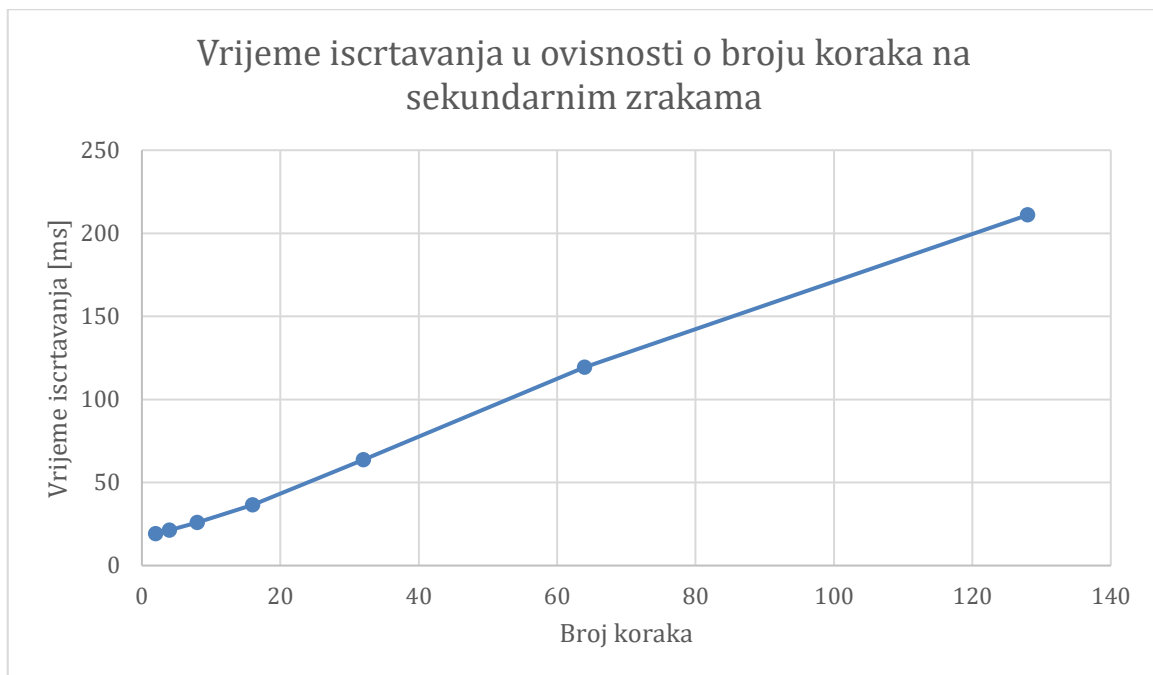
Proveden je još jedan test da se vidi učinkovitost ranog izlaska koji je opisan u potpoglavlju 6.1. Slika 7.3 prikazuje ovisnost vremena iscrtavanja o koeficijentu izumiranja za scenu koja je cijela prekrivena oblacima tipa *cumulus*. Zbog toga što je cijela iscrtana slika ispunjena oblacima vidljiv je veliki dobitak na performansama povećanjem koeficijenta izumiranja zbog ranog izlaska. Međutim, kada se iscrtava scena koja ima prosječnu prekrivenost oblacima (Slika 7.4), dobitak na performansama drastično opada, no još uvijek je donekle primjetljiv.

---

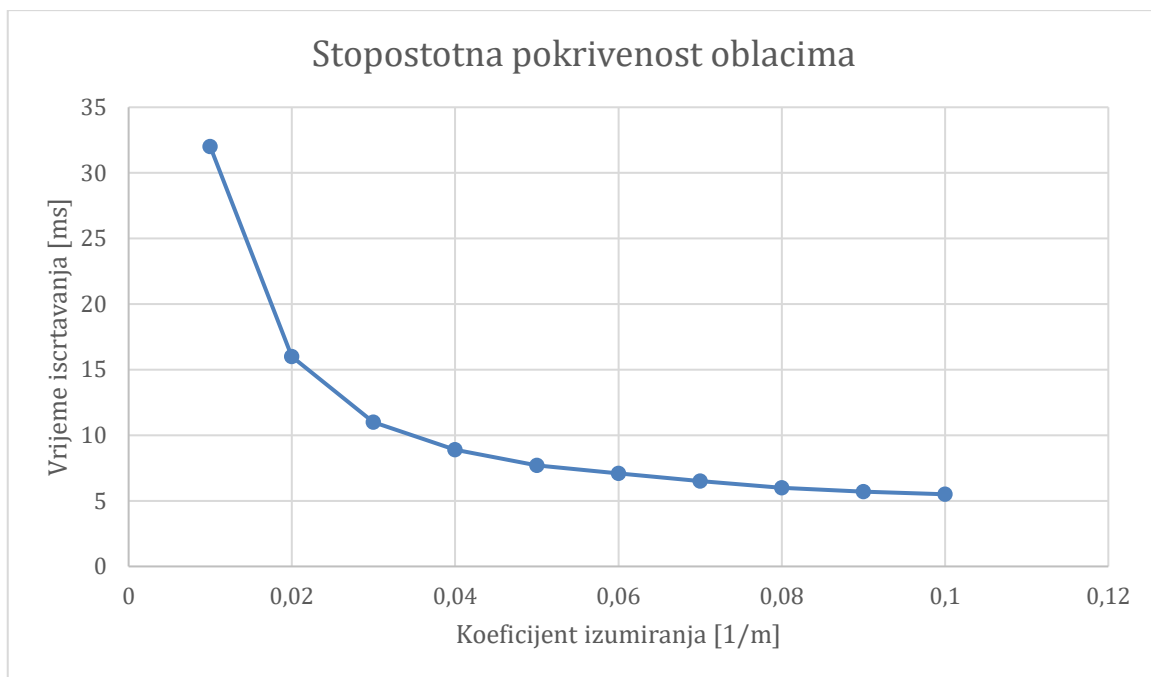
<sup>9</sup> <https://gitlab.com/markomijolovic/physically-based-cloud-rendering>



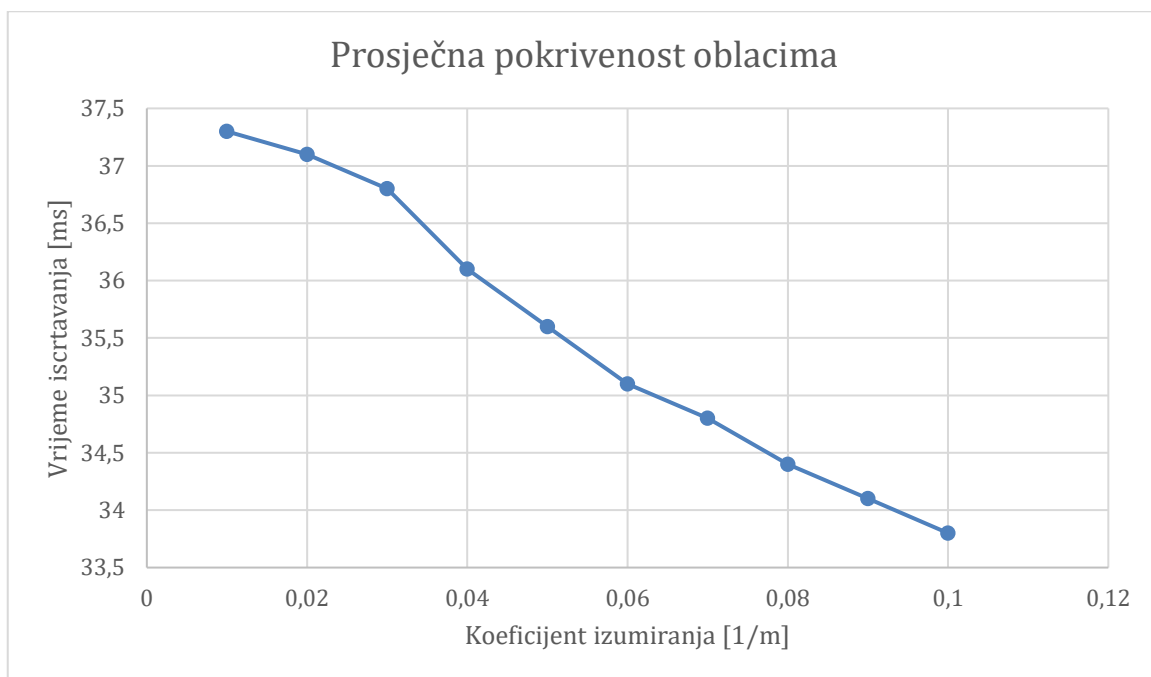
**Slika 7.1 Vrijeme iscrtavanja u ovisnosti o broju koraka na primarnoj zruci**



**Slika 7.2 Vrijeme iscrtavanja u ovisnosti o broju koraka na sekundarnim zrakama**



**Slika 7.3 Vrijeme iscertavanja u ovisnosti o koeficijentu izumiranja**



**Slika 7.4 Vrijeme iscertavanja u ovisnosti o koeficijentu izumiranja**

## 8. ZAKLJUČAK

Postupak modeliranja i iscrtavanja oblaka predstavljen u ovom završnom radu može reproducirati oblake pod dinamičnim svjetlosnim okolnostima. Iako je fokus na oblacima, prikazane metode mogu se koristiti i za iscrtavanje drugih sudjelujućih medija uz odgovarajuću promjenu parametara modela.

Proceduralno modeliranje oblaka temeljeno na funkcijama šuma pokazalo se vizualno uvjerljivo i jednostavno za implementirati. Trodimenzionalne teksture oblaka također bi se mogle generirati i na druge načine, npr. rješavanjem jednažbi toka fluida.

Henye-Greenstein fazna funkcija koja je korištena u mnogim radovima koji se bave prikazom oblaka ne može ispravno reproducirati vizualne učinke koji se uočavaju pri interakciji svjetlosti s oblacima. Fizikalno temeljena Mie fazna funkcija daje puno bolje rezultate, a cijena toga nije velika – jedno dodatno čitanje teksture u sjenčaru.

Prikazan postupak iscrtavanja se pokazao veoma računski intenzivan čak i za moderne grafičke kartice. Da bi se navedeni postupak mogao koristiti u komercijalnim programskim proizvodima bilo bi potrebno napraviti brojne optimizacije. Jedna od optimizacija koja je omogućila primjenu sličnog postupka u računalnoj igri *Red Dead Redemption 2* je vremenska ponovna projekcija (engl. *temporal reprojection*) [19].

## 9. LITERATURA

- [1] Fairchild, Mark. 2002. Color Appearance Models. Proceedings of SPIE - The International Society for Optical Engineering.
- [2] Julian Fong, Magnus Wrenninge, Christopher Kulla, and Ralf Habel. 2017. Production volume rendering: SIGGRAPH 2017 course. In ACM SIGGRAPH 2017 Courses (SIGGRAPH '17). Association for Computing Machinery, New York, NY, USA, Article 2, 1–79.
- [3] James T. Kajiya. 1986. The rendering equation. In Proceedings of the 13th annual conference on Computer graphics and interactive techniques (SIGGRAPH '86). Association for Computing Machinery, New York, NY, USA, 143–150.
- [4] Antoine Bouthors. Real-Time Realistic Rendering of Clouds. PhD thesis. Université Joseph Fourier. 2008.
- [5] Philip Laven. MiePlot.  
<http://www.philiplaven.com/mieplot.htm>, pristupano 07.05.2020.
- [6] Andrew Schneider. Real-Time Volumetric Cloudscapes. In GPU Pro 7, CRC Press, 2016, pp. 97–127.
- [7] Ken Perlin. 1985. An image synthesizer. SIGGRAPH Comput. Graph. 19, 3 (Jul. 1985), 287–296.
- [8] Steven Worley. 1996. A cellular texture basis function. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96). Association for Computing Machinery, New York, NY, USA, 291–294.
- [9] Andrew Schneider and Nathan Vos. Nubis: Authoring Realtime Volumetric Cloudscapes with the Decima Engine. SIGGRAPH Advances in Real-Time Rendering in Games course. Aug. 2017.
- [10] Eva Cerezo, Frederic Perez-Cazorla, Xavier Pueyo, Francisco Seron, François X. Sillion. A Survey on Participating Media Rendering Techniques. Visual Computer, Springer Verlag, 2005.

- [11] Andrew Schneider. Nubis: Realtime Volumetric Cloudscapes in a Nutshell. 2018 Eurographics Conference, Delft, The Netherlands 19 April 2018
- [12] The Henyey-Greenstein phase function.  
[https://www.astro.umd.edu/~jph/HG\\_note.pdf](https://www.astro.umd.edu/~jph/HG_note.pdf), pristupano 07.05.2020.
- [13] M. Hess, P. Koepke, and I. Schult. "Optical Properties of Aerosols and Clouds: The Software Package OPAC". In: Bulletin of the American Meteorological Society 79.5 (1998), pp. 831–844.
- [14] Wrenninge, Magnus & Kulla, Chris & Lundqvist, Viktor. (2013). Oz: the great and volumetric.
- [15] Benoît Mayaux. 2013. Real-Time Volumetric Rendering course notes.  
<http://patapom.com/topics/Revision2013/Revision%202013%20-%20Real-time%20Volumetric%20Rendering%20Course%20Notes.pdf>, pristupano 10.05.2020.
- [16] Joey DeVries. Learn OpenGL. HDR.  
<https://learnopengl.com/Advanced-Lighting/HDR>, pristupano 06.04.2020.
- [17] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. 2018. Real-Time Rendering, Fourth Edition (4th. ed.). A. K. Peters, Ltd., USA.
- [18] Alan Wolfe. Ray Marching Fog With Blue Noise.  
<https://blog.demofox.org/2020/05/10/ray-marching-fog-with-blue-noise/>, pristupano 21.04.2020.
- [19] Fabian Bauer. Creating the Atmospheric World of Red Dead Redemption 2: A Complete and Integrated Solution. SIGGRAPH Advances in Real-Time Rendering in Games course. Jul. 2019.



## **SAŽETAK**

### **Fizikalno temeljen prikaz oblaka**

U ovom završnom radu prikazane su osnove fizikalno temeljene interakcije svjetlosti s česticama u prostoru. Opisana su dva sustava za kvantificiranje svjetlosti – radiometrija i fotometrija. Objasnjena su fizikalna svojstva oblaka te njihov utjecaj na izgled oblaka. Dan je postupak modeliranja oblaka temeljen na funkcijama šuma. Opisan je postupak iscrtavanja oblaka metodom pomicanja duž zrake. Objasnjeno je način aproksimacije ambijentnog svjetla te višestrukog raspršenja svjetlosti unutar oblaka. Opisano je nekoliko jednostavnih optimizacija koje smanjuju vrijeme potrebno za iscrtavanje. Prikazani su rezultati te su izneseni komentari istih.

**Ključne riječi:** fizikalno temeljeno sjenčanje, iscrtavanje volumena, fizika svjetlosti, funkcije šuma, pomicanje duž zrake

## **SUMMARY**

### **Physically based cloud rendering**

This bachelor's thesis presents physically based interaction of light with particles inside a volume. Two systems for quantifying light are introduced – radiometry and photometry. Physical features of clouds and their effect on the appearance of clouds is presented. A cloud model based on noise functions is introduced. Ray tracing is presented as a method of rendering volumetric objects. Methods for approximating ambient illumination and multiple scattering inside a cloud are given. Some optimization methods are introduced with the goal of reducing rendering times. Results of the thesis are presented and evaluated.

**Keywords:** physically based shading, volume rendering, physics of light, noise functions, ray tracing