

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1462

**PRESLIKAVANJE TONOVA BOJA KOD SLIKA
VISOKO DINAMIČNOG RASPONA**

Hrvoje Pejković

Zagreb, lipanj 2016.

Zagreb, 3. ožujka 2017.

DIPLOMSKI ZADATAK br. 1462

Pristupnik: **Hrvoje Pejčković (0036459052)**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Preslikavanje tonova boje kod slika velikog dinamičkog raspona**

Opis zadatka:

Proučiti tehniku podešavanja velikog dinamičkog raspona sjajnosti (engl. High-dynamic-range imaging HDRI) posebice kod računalno generiranih slika. Proučiti tehniku preslikavanja tonova boje na smanjeni opseg raspona tonova boje (engl. tone mapping) te probleme vezane uz gama korekciju. Razraditi programsko rješenje koje će implementirati proučene tehnike primjenom različitih algoritama. Načiniti ocjenu ostvarenih rezultata i implementiranih algoritama. Izraditi odgovarajući programski proizvod. Po potrebi koristiti grafičke programske alate. Rezultate rada načiniti dostupne putem Interneta. Radu priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Zadatak uručen pristupniku: 10. ožujka 2017.
Rok za predaju rada: 29. lipnja 2017.

Mentor:



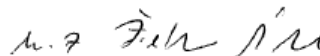
Prof. dr. sc. Željka Mihajlović

Djelovođa:



Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za
diplomski rad profila:



Prof. dr. sc. Siniša Srbljić

Zahvaljujem se svojoj obitelji što su me stalno gnjavili da završim faks, i naravno mentorici Željki Mihajlović koja je uvijek bila dobra prema meni 😊

Sadržaj

Uvod	6
1. Visoko dinamični raspon slika (engl. High dynamic range imaging HDRI).....	7
2. Preslikavanje tonova boja (engl. Tone mapping TM)	8
2.1. Svrha i metode preslikavanja tonova boje	8
3. Gama korekcija (engl. Gamma correction) [3]	10
3.1. Implementacija gama korekcija.....	12
3.2. sRGB teksture	13
3.3. Gama korekcija kod slabljenja svjetla (engl. Attenuation)	14
4. Metode preslikavanja tonova boja.....	15
4.1.Reinhard metoda za preslikavanje tonova boja u HDR slikama	15
4.1.1. Algoritam	15
4.1.2. Početno mapiranje osvjetljenja.....	15
4.2. Puma – visoko kvalitetna retinex bazirana metoda za preslikavanje tonova boja	17
4.2.1. Predobrada slike	17
4.2.2. Retinex-bazirano preslikavanje tonova boje	19
5. Implementacija.....	21
5.1. Izgled scene	21
5.2. Reinhardova metoda	22
5.1.1. Algoritam	22
5.1.2. Prostor boje	23
5.2. Utjecaj ekspozicije na scenu	24
5.3.Reinhardov samopodešavajući doprinos	27
5.4. Burgess metoda za preslikavanje tonova boje [9].....	29
5.4. Uncharted2 metoda preslikavanja tonova boje	29
5.5. Utjecaj gama korekcije	31
5.6. Slika s više boja	31
Rezultati.....	33
Zaključak.....	34
Literatura	35
Sažetak.....	36
Abstract	37

Uvod

U radu je obrađena tema preslikavanja tonova boje kod slika visoko dinamičnog raspona zajedno s gama korekcijom. Tema se većinu vremena smještala u domenu računalnog vida, ali u nedavno vrijeme, s razvitkom tehnologije, sve se više pronalazi i u računalnoj grafici. Opisan je postupak stvaranja slika visoko dinamičnog raspona, nekoliko tehnika preslikavanja tonova boje koje proizlaze iz domene fotografije, i opisana je gama korekcija koja se uvijek koristi uz preslikavanje tonova boje. Na kraju je prikazana implementacija ovih tehnika u računalnoj grafici i njihov učinak, odnosno usporedba rezultata.

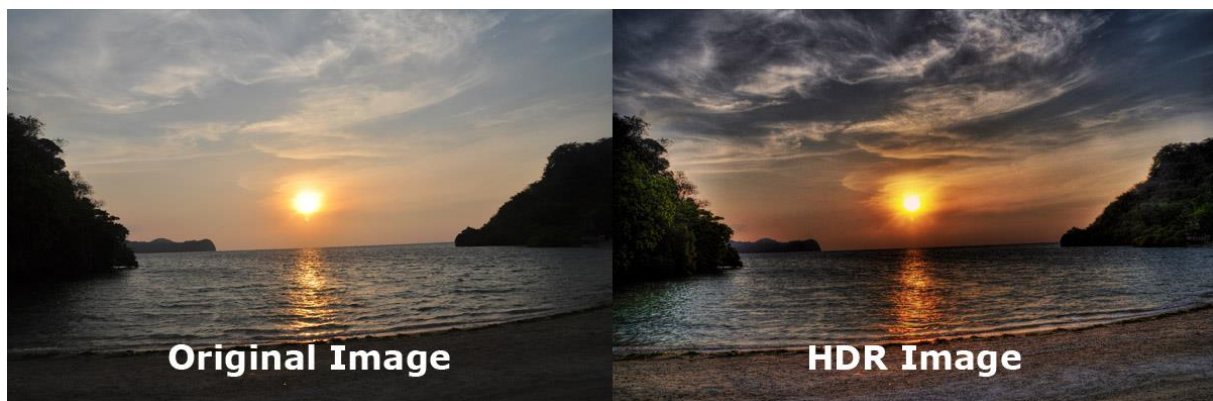
1. Visoko dinamični raspon slika (engl. High dynamic range imaging HDR)

HDR [1] je tehnika koja se koristi u stvaranju slika i fotografiranju kako bi se stvorio veći dinamični raspon osvjetljenja nego što je moguće sa standardnim digitalnim tehnikama. Cilj je ostvariti slični raspon osvjetljenja kakav vidi i čovjek. Ljudsko oko, kroz adaptaciju šarenice i drugih metoda, se konstantno prilagođava na visoki raspon osvjetljenja u okruženju. Mozak kontinuirano interpretira ovu informaciju kako bi promatrač vidio u širokom rasponu svjetla.

HDR slike mogu predstavljati veći raspon razina osvjetljenja nego što se može prikazati tradicionalnim metodama, kao što su mnoge scene iz pravog života koje sadrže jaka osvjetljenja, direktno sunce, ekstremne sjene ili blage maglice. Ovo se često postiže stapanjem nekoliko istih slika različitih ekspozicija. Neke HDR kamere stvaraju fotografije s limitiranim rasponom ekspozicije, što je nisko dinamični raspon (engl. low dynamic range LDR) i rezultira u gubitku detalja u jako svijetlim dijelovima i sjenama.

Dva primarna tipa HDR slika su računalno generirane slike i slike kao rezultat stapanja više LDR slika.

Zbog ograničenja prikazivanja kontrasta na digitalnim medijima, široki spektar osvjetljenja HDR slike se mora smanjiti kako bi bio vidljiv. Metoda prikazivanja HDR slike na standardnom monitoru se zove preslikavanje tonova boje. Ova metoda reducira cjelokupni kontrast HDR slike kako bi se prilagodio prikazu na uređajima s nisko dinamičnim rasponom prikaza, i koristi se kako bi se očuvao lokalni kontrast (ili se pretjeruje zbog umjetničkih učinaka).



Slika 1.1. Lijevo je prikazana originalna LDR slika, a desno HDR verzija te slike. (Izvor: Wikipedia)

Na slici 1.1. se može vidjeti očita korist HDR slika. Detalja je puno više i slika automatski izgleda ljepše i privlačnije.

2. Preslikavanje tonova boja (engl. Tone mapping TM)

Preslikavanje tonova boja [2] je tehnika koja se koristi u procesiranju slika i računalnoj grafici kako bi se preslikao jedan set boja u drugi i time aproksimirao prikaz HDR slika u mediju koji ima limitirani slikovni prikaz. Printeri, CRT ili LCD monitori, i projektori, svi oni imaju limitirani dinamični raspon koji je nedovoljan da reproducira kompletni raspon intenziteta svjetla prisutnih u prirodnim scenama. Preslikavanje tonova boja adresira problem jakog reduciranja kontrasta iz osvjetljenja scene u raspon koji odgovara uređaju, pritom čuvajući detalje slike.

2.1. Svrha i metode preslikavanja tonova boje

Namjera preslikavanja tonova boje se može različito izraziti ovisno o određenoj aplikaciji. U nekim slučajevima stvaranje samo estetski zadovoljavajućih slika je glavni cilj, dok u nekim drugim aplikacijama cilj može biti stvaranje što više mogućih detalja, ili maksimiziranje kontrasta. Cilj u realističnim aplikacijama je dobiti što bolje podudaranje između prave scene i dobivene slike iako uređaj za prikaz ne može reproducirati cijeli raspon osvjetljenja.

Razni operatori algoritama preslikavanja tonova boje su se razvili u posljednjim godinama. Razvrstat ih se može u dva glavna tipa:

- Globalni operatori: to su ne-linearne funkcije bazirane na svjetlosnim i drugim globalnim varijablama slike. Jednom kad je optimalna funkcija pronađena na temelju određene slike, svaki piksel u slici se preslikava na isti način, neovisno o vrijednostima susjednih piksela slike. Takve tehnike su jednostavne i brze, ali mogu prouzročiti gubitak kontrasta. Primjeri takvih globalnih operatora su kontrastna redukcija i inverzija boje.
- Lokalni operatori: parametri ne-linearne funkcije se mijenjaju u svakom pikselu, ovisno o značajkama izvučenih iz okolnih piksela. U drugim riječima, učinak algoritma se mijenja u svakom pikselu ovisno o lokalnim značajkama slike. Ovakvi algoritmi su više komplicirani od globalnih. Mogu stvoriti neke

nepoželjne učinke i izlaz može izgledati nerealistično, ali također mogu, ako se koriste pravilno, stvoriti i najbolju sliku, pošto je ljudsko oko najviše osjetljivo na lokalni kontrast.

Jednostavni primjer globalnog filtra preslikavanja tonova boje je $V_{out} = \frac{V_{in}}{V_{in+1}}$, gdje je V_{in} osvjetljenje originalnog piksela i V_{out} je izlazno osvjetljenje filtriranog piksela. Ovakva funkcija će preslikati ulazno osvjetljenje iz raspona $[0, \infty)$ u mogući prikaz $[0, 1)$. Dok ovakav filter stvara dobar kontrast za dijelove slike sa slabim osvjetljenjem, dijelovi s jačim osvjetljenjem kontinuirano dobivaju sve manji kontrast što je osvjetljenje filtrirane slike veće.

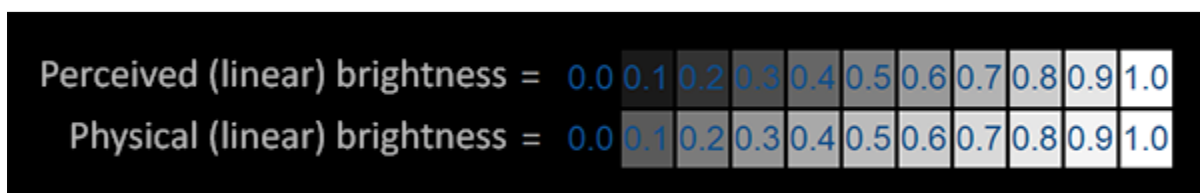
Još jedna sofisticiranija grupa TM algoritma se bazira na kontrastu ili gradijentnim metodama, koje su lokalne. Takvi operatori se koncentriraju na očuvanju kontrasta između susjednih regija nego na apsolutne vrijednosti cijele slike. Ovaj pristup je motiviran činjenicom da je ljudsko oko osjetljivije na kontrast u slikama, nego li na apsolutne vrijednosti intenziteta. Ovakve metode često proizvode jako oštre slike, koje očuvaju jako dobro male kontrastne detalje; no ipak, često se to dobiva žrtvujući cjelokupni kontrast slike pa slika može izgledati malo plosnato, i kao posljedica se može dogoditi određeni sjaj oko tamnih objekata.



Slika 2.1. Primjer preslikavanja tonova boje u HDR slici. Ovakav algoritam je više umjetničke prirode, nego realne (Izvor: Wikipedia)

3. Gama korekcija (engl. Gamma correction) [3]

Kada se izračunaju sve finalne boje piksela scene, treba ih prikazati na zaslonu. U starim danima digitalnih slika, većina monitora su bile katodni CRT monitori. Ovakvi monitori su imali fizikalno svojstvo da dva puta veći napon ne bi rezultiralo u dva puta više svjetline. Dvostruki napon bi rezultirao u osvjetljenju koje je jednako potencijskom odnosu od otprilike 2.2 što je znano kao gama monitora. Ovo je (slučajno) slično kako i čovjek mjeri svjetlost samo s inverznim potencijskim odnosom. Da bi se bolje razumjelo što sve ovo znači, pogledajte sljedeću sliku:

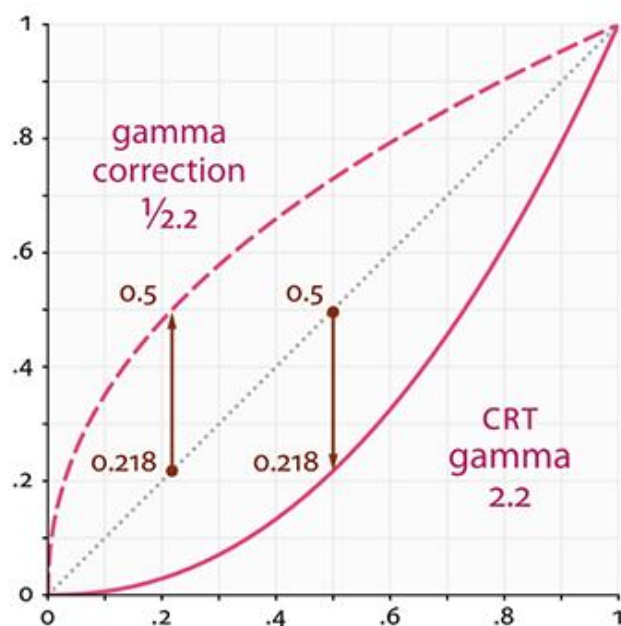


Slika 3.1. Ljestvice stvarnog fizikalnog osvjetljenja (dolje) i percipiranog osvjetljenja (gore)

Gornji redak izgleda kao točna ljestvica osvjetljenja ljudskom oku, povećanje osvjetljenja (od 0.1 do 0.2 npr.) stvarno izgleda kao dvostruko više svjetlije s finim konzistentnim razlikama. No ipak, kad govorimo o fizikalnim svojstvima svjetla, onda drugi redak prikazuje točno osvjetljenje. Na donjem retku, povećanje osvjetljenja za dvostruko vraća točnu mjeru nove svjetlosti, ali pošto naše oko prihvaća osvjetljenje drukčije (osjetljivo na promjene kod mračnih boja), onda to izgleda čudno.

Pošto ljudsko oko preferira vidjeti svjetlost prema gornjoj ljestvici, monitori (čak i danas) koriste potencijski odnos za prikaz boja kako bi se originalno fizičko osvjetljenje mapiralo u nelinearno osvjetljenje prikazano na gornjoj ljestvici; u osnovi zato što izgleda bolje.

Ovo nelinearno mapiranje na monitoru stvarno čini svjetlost da izgleda bolje u našim očima, ali kada dođe do prikazivanja grafike postoji jedan problem: sve boje i osvjetljenja koje konfiguriramo u aplikacijama su bazirane na temelju onog što percipiramo na monitoru te zbog toga sva se konfiguracija događa u nelinearnom prostoru. Pogledajte graf ispod:



Slika 3.2. Odnosi linearnog prostora, CRT game i gama korekcije

Isctkana linija predstavlja vrijednosti boja/svjetlina u linearnom prostoru i puna linija predstavlja prostor boja na prikazu monitora. Ako dvostruko povećamo boju u linearnom prostoru, rezultat će stvarno biti dvostruko veća vrijednost. Na primjer, ako uzmemo vektor boje $L = (0.5, 0.0, 0.0)$ koji reprezentira polu-tamno crveno svjetlo, ako bi poduplali ovu boju u linearnom prostoru postalo bi $(1.0, 0.0, 0.0)$ kao što se može vidjeti na grafu. No ipak, pošto se boje prikazuju na monitoru, polu-tamna crvena boja se prikazuje kao $(0.218, 0.0, 0.0)$ što se vidi na grafu. Ovdje se nailazi na problem, jer kada poduplamo tamno-crvenu boju u linearnom prostoru, ona zapravo postaje 4.5 puta svjetlija na monitoru! Primijetite da na krajnje vrijednosti ovaj problem ne utječe, samo na vrijednosti između.

Pošto su boje konfigurirane bazirano na prikazu monitora, svi izračuni u linearnom prostoru su fizikalno netočni. Ovo postaje sve više uočljivije kada se koriste napredniji algoritmi osvjetljenja, kao što se vidi na slici 3.3.



Slika 3.3. Lijeva slika prikazuje igru bez gama korekcije, a desna slika prikazuje igru sa gama korekcijom (Izvor: <https://learnopengl.com>)

Na slici se može vidjeti da se s gama korekcijom, boje ponašaju puno prirodnije, odnosno puno bolje se stapaju boje i tamna područja su manje tamna i prikazuju više detalja. Cjelokupna slika izgleda puno bolje sa samo malom modifikacijom.

Bez pravilne gama korekcije na monitora, osvjetljenje izgleda loše i dizajneri će imati problema sa stvaranjem realističnih i lijepih rezultata jer neispravno renderirane slike se ne mogu jednostavno ispraviti u post-processing koraku. Rješenje je iskoristiti gama korekciju.

3.1. Implementacija gama korekcija

Ideja gama korekcije je da se aplicira inverz iznosa game zaslona na finalnu boju prije prikaza na monitoru. Ako pogledamo maloprije prikazanu sliku 3.2., vidjet ćemo iscrtkanu liniju koja prikazuje inverz gama krivulje monitora. Pomnožimo li svaki od linearnih izlaza s ovom inverznom gama krivuljom (učineći ih svjetlijima) čim su boje prikazane na monitoru, gama krivulja monitora je aplicirana, i rezultirajuća boja postaje linearna. Zapravo što radimo je da srednje boje posvijetlimo kako bi kad ih monitor potamni, sve se korigiralo.

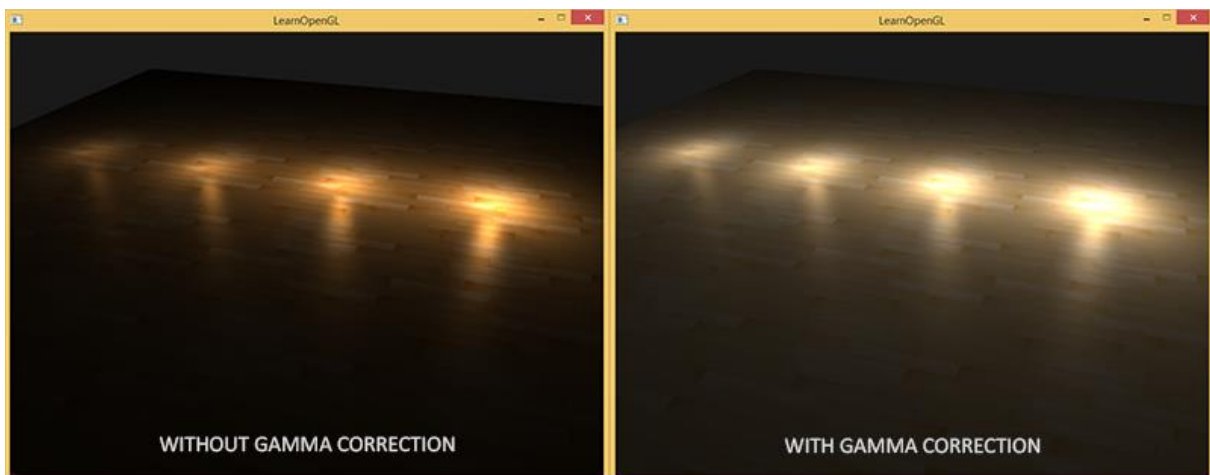
Recimo da imamo tamno-crvenu boju (0.5, 0.0, 0.0). Prije prikaza na monitoru prvo apliciramo gama korekciju na vrijednost. Boje prikazane na monitoru su otprilike skalirane s potencijom 2.2 tako da inverz zahtjeva da se te boje skaliraju s potencijom $1/2.2$. Gama ispravljena boja tada postaje $(0.5, 0.0, 0.0)^{1/2.2} = (0.5, 0.0, 0.0)^{0.45} = (0.73, 0.0, 0.0)$. Korektirana boja se tada predaje monitoru koji kao

rezultat prikazuje $(0.73, 0.0, 0.0)^{2.2} = (0.5, 0.0, 0.0)$. Vidi se da koristeći gama korekciju, monitor napokon točno prikazuje boje kako ih linearno postavljamo u aplikaciji.

3.2. sRGB teksture

Pošto monitori uvijek prikazuju boje s gamom apliciranom u sRGB prostoru, kad god crtamo, mijenjamo ili slikamo sliku na računalu, zapravo bismo bazirano na tome što vidimo na monitoru. To efektivno znači da sve slike koje kreiramo ili mijenjamo nisu u linearnom prostoru, nego u sRGB prostoru, na primjer, ako poduplamo tamno-crvenu boju na ekranu na temelju onog što vidimo, to neće biti stvarno dvostruko više crvene.

Kao rezultat, ljudi koji se bave teksturama kreiraju sve teksture u sRGB prostoru pa ako ih želimo koristiti u našim grafičkim aplikacijama, trebamo to uzeti u obzir. U slučaju kada se koristi gama korekcija, teksture će izgledati puno svjetlije nego što bi trebale.



Slika 3.2.1. Lijevo je prikazana sRGB tekstura bez gama korekcije, a desno je prikazana ta ista sRGB tekstura s gama korekcijom

Teksture na desnoj slici su presvijetle i to se događa jer su zapravo gama ispravljene dva puta! Pošto je tekstura već gama ispravljena, s nanošenjem još jedne gama korekcije, zapravo smo je napravili dva puta i slika će biti svjetlija nego što želimo.

Kako bi se ovo popravilo, sve što je potrebno jest prebaciti boje teksture u linearni prostor prije nego se koriste algoritmi osvjetljenja. To se postiže inverzom gama korekcije, odnosno boju treba kvadrirati s gamom monitora, jer se time zapravo poništi stvorena gama korekcija i opet se može raditi u linearnom prostoru.

3.3. Gama korekcija kod slabljenja svjetla (engl. Attenuation)

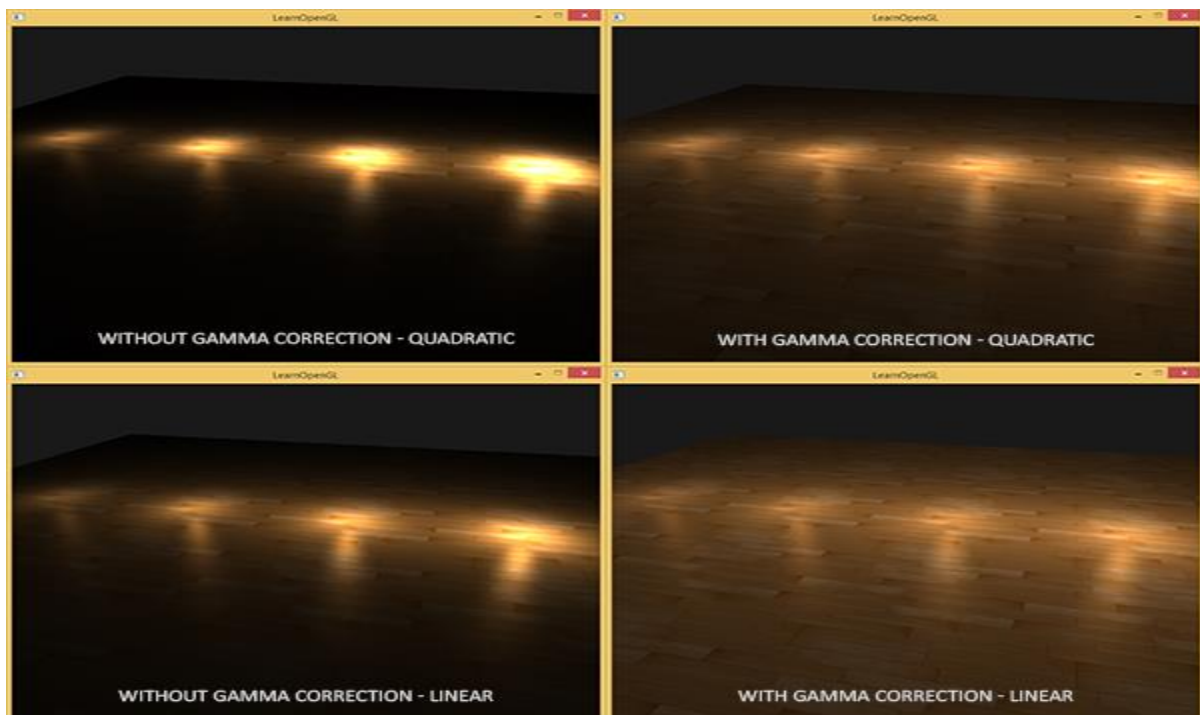
Još jedna stvar koju treba uzeti u obzir kod gama korekcije je slabljenje svjetla. U pravom svijetu, svjetlo slabi otprilike recipročno proporcionalno kvadratu udaljenosti od izvora svjetlosti.

$$\text{Attenuation} = 1.0 / (\text{distance} * \text{distance})$$

No ipak, kada se koristi formula slabljenja svjetla, efekt slabljenja je prejak i svjetlo ima jako mali radijus koji ne izgleda fizikalno točno. Zbog tog razloga, koristi se druga jednadžba, u kojoj je slabljenje linearno proporcionalno s udaljenosti.

$$\text{Attenuation} = 1.0 / \text{distance}$$

Linearna jednadžba daje bolje rezultate kada se ne koristi gama korekcija, ali kada uključimo gama korekciju onda slabljenje svjetla izgleda jako slabo i onda zapravo fizikalno točna kvadratna jednadžba odjednom daje bolje rezultate. Slika dolje prikazuje razlike.



Slika 3.3.1. Gore lijevo – scena nad kojom se koristi kvadratna jednadžba bez gama korekcije; Gore desno – scena nad kojom se koristi kvadratna jednadžba ali s gama korekcijom; Dolje lijevo – scena nad kojom se koristi linearna jednadžba slabljenja svjetla bez gama korekcije; Dolje desno – scena nad kojom se koristi linearna jednadžba slabljenja svjetla s gama korekcijom

Razlog ovoj razlici je zbog toga što slabljenje svjetla mijenja osvjetljenje, a kako nismo promatrali scenu u linearnom prostoru, izabrali smo funkciju slabljenja koja najbolje izgleda na monitoru, ali nismo bili fizikalno ispravni. Ako koristimo kvadratnu funkciju bez gama korekcije, dogodi se ovo: $(1.0/distance^2)^{2.2}$ kada se prikaže na monitor. Ovo stvara puno jače slabljenje nego što bi htjeli. To objašnjava zašto je linearna funkcija puno bolja kad se ne koristi gama korekcija jer je jednadžba tada $(1.0/distance)^{2.2}$ što efektivno predstavlja fizikalni ekvivalent puno bolje.

Kako bi sumirali sve ovo, gama korekcija dopušta da se radi u linearnom prostoru. Pošto linearni prostor ima puno više smisla u fizičkom svijetu, većina fizikalnih jednadžbi ovako daje bolje rezultate kao pravo slabljenje svjetla. Što su naprednije tehnike osvjetljenja koje se koriste, to je lakše dobiti dobro izgledajuće rezultate s gama korekcijom.

4. Metode preslikavanja tonova boja

4.1. Reinhard metoda za preslikavanje tonova boja u HDR slikama

U ovom poglavlju bit će opisana jedna od osnovnih i najčešćih metoda za preslikavanje tonova boja, a to je Reinhard-ova metoda.

4.1.1. Algoritam

Sistem zone koji je opisan u [4] se koristi kako bi se razvio novi algoritam za preslikavanje tonova boje za digitalne slike, koje su kreirane od strane algoritma za izradu prikaza ili HDR slike. Ukratko, sistem zone je metodologija koja previđa kako će se scensko osvjetljenje preslikati u skup zona za prikazivanje.

4.1.2. Početno mapiranje osvjetljenja

Prvo se mijenja tonski raspon slike baziranog na scenskom ključu. Kao što su mnoge tonsko reprodukcijske metode, koristimo se s log-prosječnim osvjetljenjem kao korisna aproksimacija na ključ scene. Vrijednost \bar{L}_w se izračunava ovako:

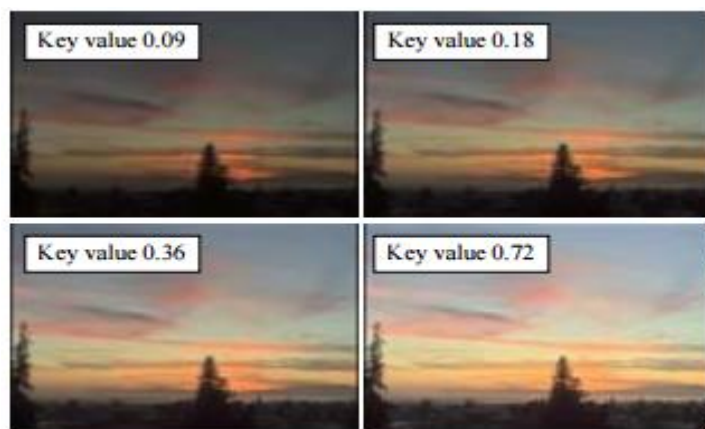
$$\bar{L}_w = \frac{1}{N} \exp(\sum_{x,y} \log(\alpha + L_w(x, y))) \quad (1)$$

Gdje je $L_w(x, y)$ „svjetsko“ osvjetljenje za pixel(x,y), N je totalni broj piksela u slici a α je mali broj koji se koristi da bi se izbjegao logaritam od 0. Ako scena ima normalni

ključ htjeli bismo mapirati ovo u srednje-sivo slike, ili 0.18 na skali od 0 do 1. Iz toga slijedi sljedeća jednačba:

$$L(x, y) = \frac{a}{Lw} Lw(x, y) \quad (2)$$

Gdje je $L(x, y)$ skalirano osvjetljenje i $a = 0.18$. Za niski-ključ ili visoki-ključ slike dopuštamo različite vrijednosti varijable a . Tipično ta vrijednost varira između 0.18 do 0.36 i od 0.72 do 0.09, i 0.045. Primjer variranja je dan na slici 4.1.1.



Slika 4.1.1. Reinhard metoda za različite vrijednosti ključa

Glavni problem s jednačbom (2) je da mnoge scene imaju dominantno normalni dinamični raspon, ali imaju nekoliko visoko osvijetljenih dijelova kao npr. nebo. U tradicionalnoj fotografiji ovaj problem se rješava kompresijom i visokih i niskih osvijetljenja. No ipak, moderna fotografija je odbacila ove načine i koristi uglavnom kompresiju visokih osvijetljenja. Jednostavni operator preslikavanja tonova boja s ovim karakteristikama je:

$$L_d(x, y) = \frac{L(x, y)}{1 + L(x, y)} \quad (3)$$

Visoka osvijetljena su skalirana otprilike s $1/L$, dok su niska osvijetljenja skalirana s 1. Nazivnik uzrokuje lagano miješanje između ova dva skaliranja. Ova formula garantira da će staviti sva osvijetljenja u raspon koji se može prikazati. No ipak, ovo

nije uvijek poželjno. Formula se može proširiti kako bi dozvolila visokim osvjetljenjima da izgore na kontrolirani način.

$$L_d(x, y) = \frac{L(x, y)(1 + \frac{L(x, y)}{L_{white}^2})}{1 + L(x, y)} \quad (4)$$

gdje je L_{white} najmanje osvjetljenje koje će se mapirati u kompletno bijelu. Ova funkcija je mješavina jednadžbe 3 i linearnog mapiranja. Ako se L_{white} vrijednost postavi na maksimalno osvjetljenje u sceni, neće se pojaviti prezasićenost. Ako se podesi na beskonačno, onda se funkcija prebacuje u jednadžbu 3. Po standardu L_{white} se postavlja u maksimalno osvjetljenje u sceni. Za mnoge HDR slike, ovakva metoda je dovoljna da sačuva detalje u tamnijim područjima, dok sažima visoka osvjetljenja u raspon koji se može prikazati.

4.2. Puma – visoko kvalitetna *retinex* bazirana metoda za preslikavanje tonova boja

Retinex algoritmi su bazirani na tome kako ljudska mrežnica (engl. Retina) vidi boje te od tuda naziv.

Umjesto da se direktno mijenja vrijednost osvjetljenja, alternativna metoda je predložena da se pripremi HDR slika kroz odgovarajuću predobradu da bi bila prikladna za nepromijenjenu metodu svjetlosnog podešavanja. Jedna takva predobrada se može sastojati od apliciranja operatora preslikavanja tonova boje kako bi se dobila LDR slika. Ovo bi spriječilo bilo kakve potencijalne razmake visokog intenziteta između susjednih područja slike i eliminirati glavnu prepreku u uspješnoj aplikaciji metoda svjetlosnog podešavanja. Na ovaj način, poslije inicijalnog preslikavanja tonova boje, svaka teorijski dobro definirana metoda svjetlosnog podešavanja bi se trebala moći koristiti za točno što i treba. Slični pristupi su bili već predloženi, ali glavni problem je da su takva i slična rješenja jako kompleksne metode predobrade, a prate ih još kompleksnije metode podešavanja svjetla.

4.2.1. Predobrada slike

Jedan način da se pronađe jednostavan operator preslikavanja tonova boje za predobradu bi mogao biti oponašanje čovjekovog vizualnog sustava (human visual

system HVS). Odzivna krivulja intenziteta svjetla za stupiće i čunjiće u mrežnici se može prikazati funkcijom:

$$r(I) = \frac{I^n}{I^n + I_s^n} \quad (5)$$

Gdje je I intenzitet svjetla, I_s je razina na kojoj je odziv pola maksimuma, n je osjetljivost eksponenta kontrole, a $r(I)$ je odziv za kojeg se pretpostavlja da je koreliran s percipiranom svjetlosti. Formula (5) ima povijest uspjeha s uspješnom aplikacijom u ovakvim sustavima iako postoje točniji modeli HVS percepcije. Za $n = 1$ jednadžba se pojednostavljuje u Naka-Rushton jednadžbu. U svojoj proširenoj i praktičnijoj formi, jednadžba izgleda ovako

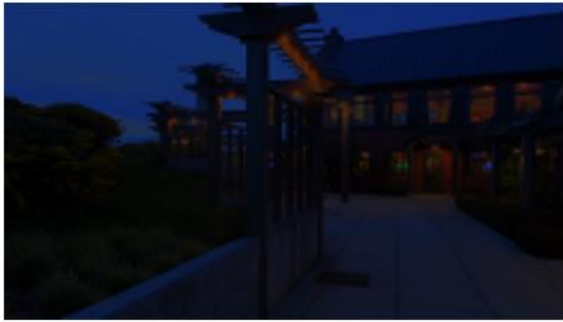
$$L' = \frac{\frac{L}{L_w}}{\frac{L}{L_w} + a} = \frac{L}{L + aL_w} \quad (6)$$

Gdje je L početno osvjetljenje, L_w je ključ slike, a a je parametar skaliranja. Dijeleći inicijalno osvjetljenje s ključem slike je često korišten korak. Ključ slike je aproksimiran s geometrijskim prosjekom osvjetljenja, kao i kod Reinhard metode

$$L_w = \exp\left(\frac{1}{N} \sum_i \ln(L(i) + \epsilon)\right) \quad (7)$$

Gdje je N broj piksela, $L(i)$ je osvjetljenje i -tog piksela, a ϵ je mala vrijednost da se izbjegne logaritam od 0. Rezultat funkcije S oblika slične onoj opisanoj formulom 4 su lošiji od trenutno najboljih (engl. State-of-the-art) algoritama za preslikavanje tonova boja u smislu konačne kvalitete LDR slike i često se koriste sofisticiranije metode. No ipak, zbog svoje jednostavnosti, dobar je kandidat za spomenuto inicijalno mapiranje tonova prije predaje slike iz predobrade metodi za podešavanje svjetla.

Pokazano je kako kvaliteta rezultata globalnih TMO-a se može značajno poboljšati tako da ih se koristi nad V kanalom HSV prostora boje umjesto nad Y kanalom YUV prostora boja.



(a)



(b)

Slika 4.2.1. Ista scena mapirana jednadžbom (7) koristeći (a) sivi (b) V kanal slike.

Parametar a u jednadžbi (6) kontrolira kako se koriste vrijednosti osvjetljenja. Manje vrijednosti parametra a stvaraju svjetlije slike i obrnuto kako je prikazano na slici 4.2.1, što ih naizgled čini više korisnima. No ipak, više vrijednosti unose viši kontrast, što se može uspješno iskoristiti da bi se dobila veća kvaliteta tokom podešavanja svjetla. Ovakva metoda se naziva Flash.

4.2.2. Retinex-bazirano preslikavanje tonova boje

Za predobradu slike koristit će se Flash metoda. Algoritam koji će se koristiti za modifikaciju svjetla je SLRMSR (engl. Smart Light Random Memory Sprays Retinex), a opisan je kako funkcionira u nastavku:

A. Algoritam SLRMSR

SLRMSR se oslanja na Retinex teoriju i procesira svaki piksel individualno. Glavni procesirajući dio je

$$L^*(i) = \frac{L(i)}{L(x_{Hi})} \quad (8)$$

Gdje je x_{Hi} indeks piksela s najvećim osvjetljenjem u raspršenom dijelu n piksela nasumično odabranih oko i -tog piksela. Rezultat dobiven poslije procesa odstranjivanja buke o kojem se može više pročitati u [4] može se opisati kao

$$L'(i) = \text{Denoise}(L^*(i)) \quad (9)$$

Jedan od posljednjih koraka SLRMSR je da preventira moguće prejako posvjetljivanje već jako svjetlih piksela, tako da se uvede prilagodljivi mehanizam mapiranja intenziteta, koji se dobiva kao

$$L''(i) = \lambda(i)L(i) + (1 - \lambda(i))Y'(i) \quad (10)$$

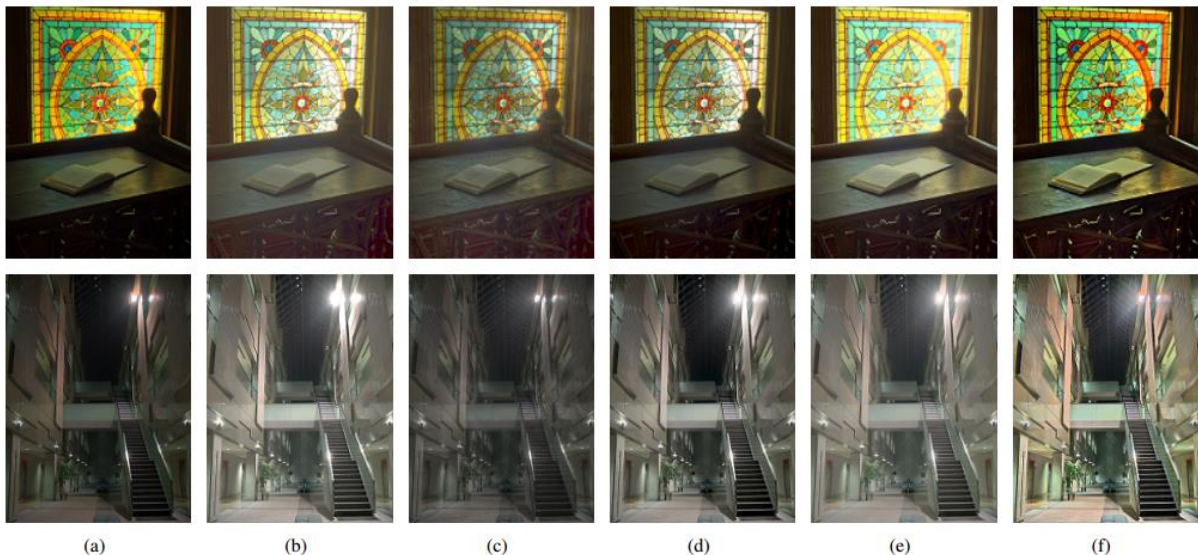
$$\lambda(i) = \left(\frac{Y_i(i)}{D}\right)^r$$

Gdje je r intenzitet remapirajućeg i podešavajućeg parametra, $L(i)''$ je finalni rezultat za i -ti piksel, koji se kasnije može opcionalno obraditi da poboljša oštrinu slike.

B. Kombinacija

Kada se Flash i SLRMSR kombiniraju, postoje tri parametra koja se trebaju postaviti: a , n , i r . Najčešće je najbolje eksperimentalno utvrditi koji su najbolji. Što se tiče parametra a , kao što je spomenuto, za više vrijednosti a Flash stvara tamnije slike, ali nakon što je osvjetljenje adaptirano od strane SLRMSR, kontrast je puno bolji nego u slučaju manjih vrijednosti a . Sa višim vrijednostima varijable a , dobitak u kontrastu je jako mali i zbog ovog razloga standardna vrijednost je $a = 10$.

Rezultati:



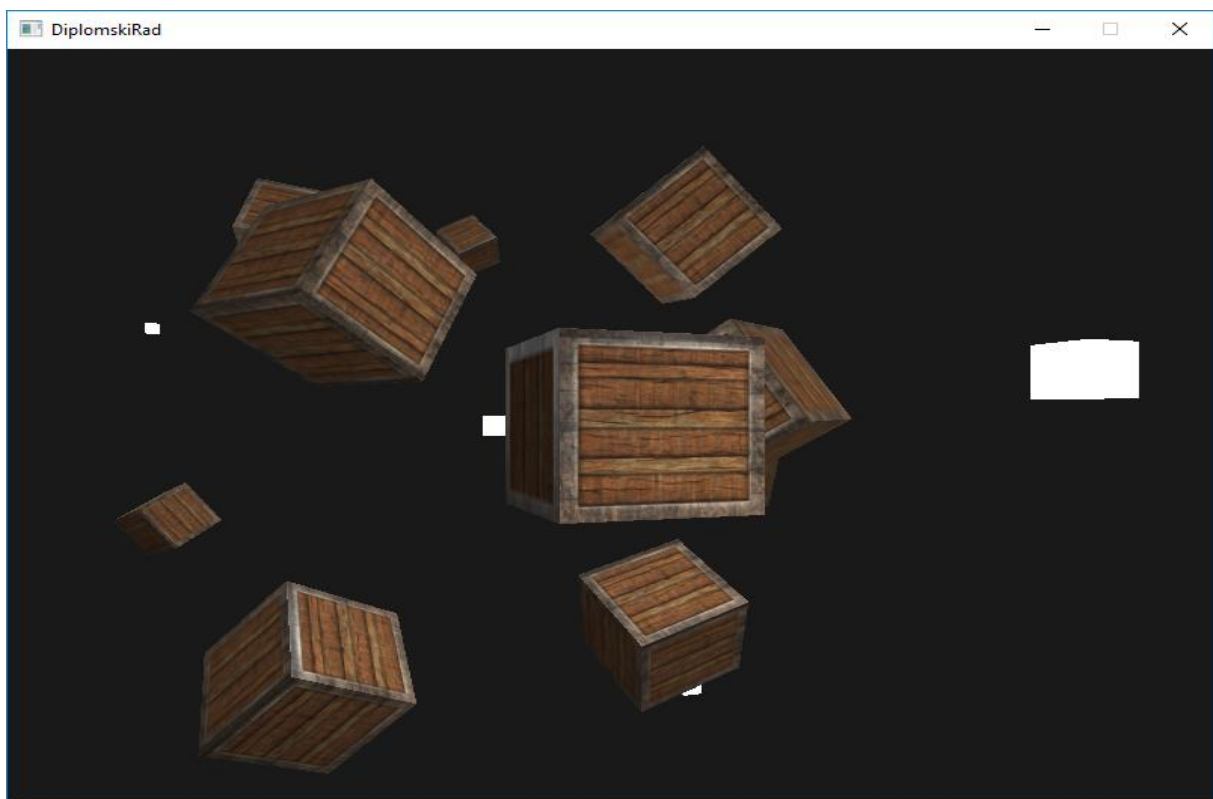
Slika 4.2.1. primjeri *tone mapping* operatora (TMO) : (a) Dragov TMO, (b) Durandov TMO, (c) Mantiukov TMO, (d) Mantiukov TMO, druga verzija, (e) Reinhardov TMO, (f) Puma.

Sa slike se može vidjeti da Puma metoda daje jako dobre rezultate u odnosu na druge popularne operatore preslikavanja tonova boje.

5. Implementacija

5.1. Izgled scene

Scena nad kojoj sam isprobavao tehnike HDR-a i preslikavanja tonova boja te gama korekcije, implementirana je koristeći programski jezik C++ i grafičku biblioteku OpenGL [6]. Scena je prikazana na slici 5.1.



Slika 5.1. Izgled scene

Prvo što sada treba, pošto je scena nisko dinamičnog raspona, jest proširiti taj raspon. To sam postigao skaliranjem slike i stvaranjem dvije nove, s većom i manjom ekspozicijom te linearnom interpolacijom te dvije slike na temelju originalne. Na taj način svjetliji dijelovi postaju još svjetliji, a tamniji dijelovi još tamniji te se automatski povećava kontrast slike. Naravno postoje kompleksne metode za dobivanje HDR slike, koristeći mapu zračenja slike, ali fokus mog rada je bio više na preslikavanju tonova boja, a ovakvo postizanje HDR slike mi je bilo dovoljno. Sada kada je generirana HDR scena potrebno je aplicirati metode preslikavanja tonova boja kako bi se raspon ponovno vratio u LDR područje koje se može prikazati na zaslonu, uz naravno očuvanje detalja.

5.2. Reinhardova metoda

Implementirao sam već opisanu Reinhard [7] metodu za preslikavanje tonova boja.

5.1.1. Algoritam

Algoritam ima 4 dijela.

1. Izračunavanje osvjetljenja

$$L_w(x, y) = (0.2126 * red) + (0.7152 * blue) + (0.0722 * green)$$

Gdje je $L_w(x, y)$ izračunata vrijednost osvjetljenja piksela. Formula se bazira na onome što vidi ljudsko oko. Iz razloga što ljudsko oko ne vidi crvenu, plavu i zelenu boju jednako, tipična $(R + G + B) / 3$ formula ne bi bila dobar izbor za računanje pikselovog osvjetljenja.

2. Prosječno osvjetljenje

$$\bar{L}_w = \exp\left(\frac{1}{N} \sum_i \ln(L_w(x, y) + \epsilon)\right)$$

Ova formula iterira po svim pikselima slike i računa prosječno osvjetljenje za sliku, gdje je L_w izračunato prosječno osvjetljenje slike. Da bi se ovo napravilo koristeći grafičku karticu, koristi se tehnika mipmapiranja (eng. Mipmapping) [8]. Kada se dođe do 1x1 piksela, dobilo se prosječno osvjetljenje. ϵ je malo podešenje kako bi se izbjegao logaritam od 0.

3. Skaliranje osvjetljenja

$$L(x, y) = \frac{\alpha}{L_w} * L_w(x, y)$$

Gdje je

$L(x, y)$ skalirajući faktor za osvjetljenje piksela.

α je ključ slike, tj. Parametar koji skalira ekspoziciju slike. Male vrijednosti (0.0 do 0.2) će podeksponirati sliku, a visoke vrijednosti (0.5 do 1.0) će preekspozionirati sliku.

L_w je izračunato u dijelu 2.

4. Konačna vrijednost osvjetljenja

$$L_d(x, y) = \frac{L(x, y) \left(1 + \frac{L(x, y)}{L_{white}}\right)}{1 + L(x, y)}$$

Gdje je

$L_d(x, y)$ vrijednost osvjetljenja kojom se skalira finalni piksel.

$L(x, y)$ je izračunat u dijelu 3.

L_{white} je najmanja vrijednost osvjetljenja koja će se mapirati u čisto bijelu boju. Standardno, po Reinhardu, ova vrijednost se postavlja na maksimalno osvjetljenje u sceni, čime se izbjegava izgaranje.

5.1.2. Prostor boje

Finalni korak je primijeniti $L_d(x, y)$ na piksel. Ali pošto ljudsko oko ne vidi crvenu, plavu i zelenu boju jednako, ne može se aplicirati skalarni faktor na RGB boju direktno. Prvo je potrebno konvertirati RGB piksel u CIE xyY prostor boja, gdje je Y osvjetljenje, a xy je kromatičnost piksela. Konverzija je prikazana u nastavku.

Konverzija iz RGB u XYZ pa u xyY.

$$XYZ = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{matrix} R \\ G \\ B \end{matrix}$$

$$xyY = \frac{X}{X+Y+Z}, \frac{Y}{X+Y+Z}, Y$$

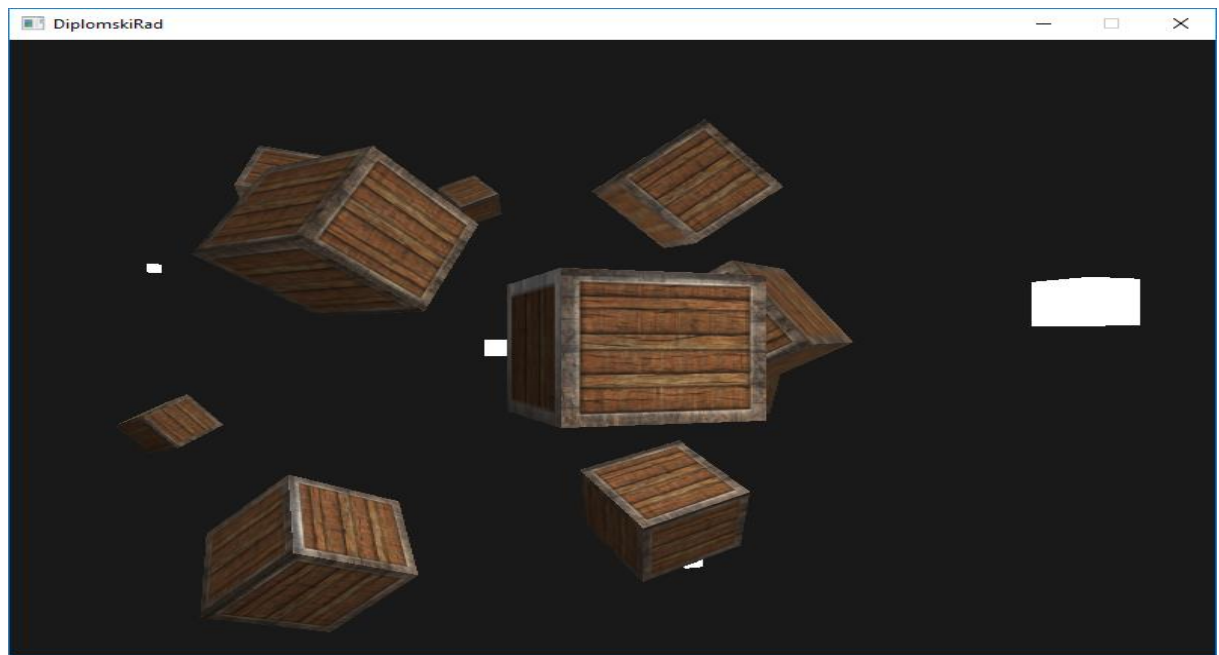
Jednom kada imamo xyY vrijednost, pomnožimo osvjetljenje s faktorom veličine.

$$xyYy = xyYy * L_d(x, y)$$

Konačno, još treba napraviti konverziju iz xyY u XYZ pa u RGB

$$XYZ = \left(\frac{Y}{y}x, Y, \frac{Y}{y}(1.0 - x - y) \right)$$

$$RGB = \begin{pmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{pmatrix} \begin{matrix} X \\ Y \\ Z \end{matrix}$$



Slika 5.2. Izgled Reinhard preslikavanja tonova boje.

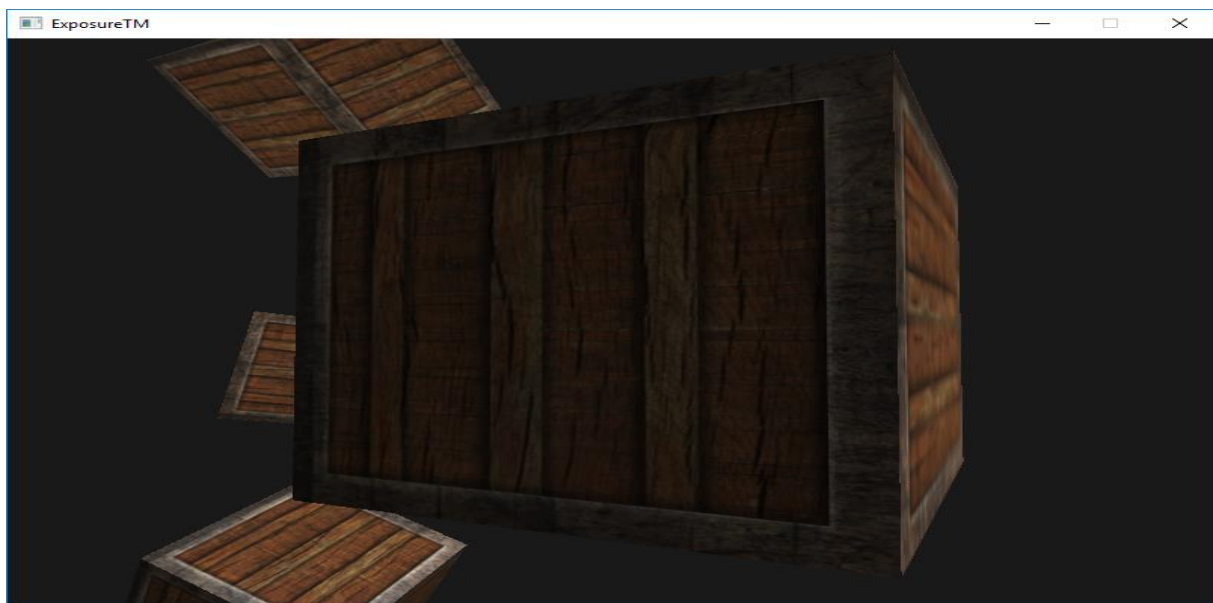
5.2. Utjecaj ekspozicije na scenu



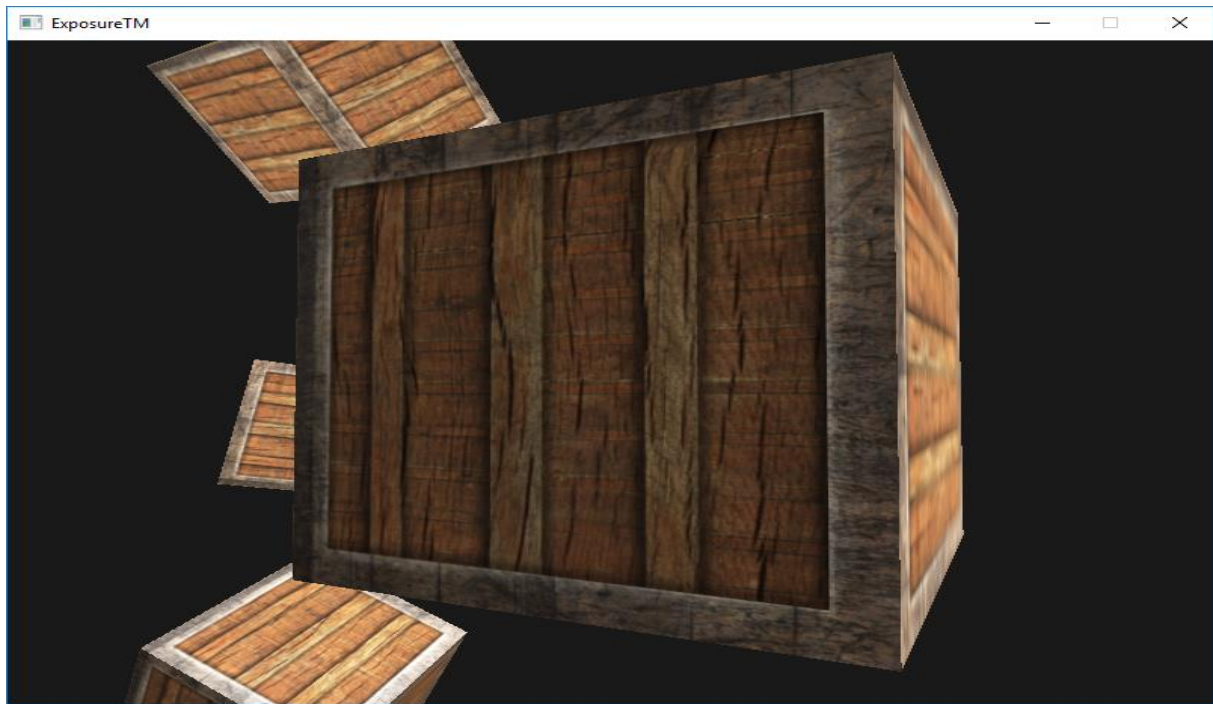
Slika 5.2.1. Izgled scene s ekspozicijom 1



Slika 5.2.2. Izgled scene s ekspozicijom 5.



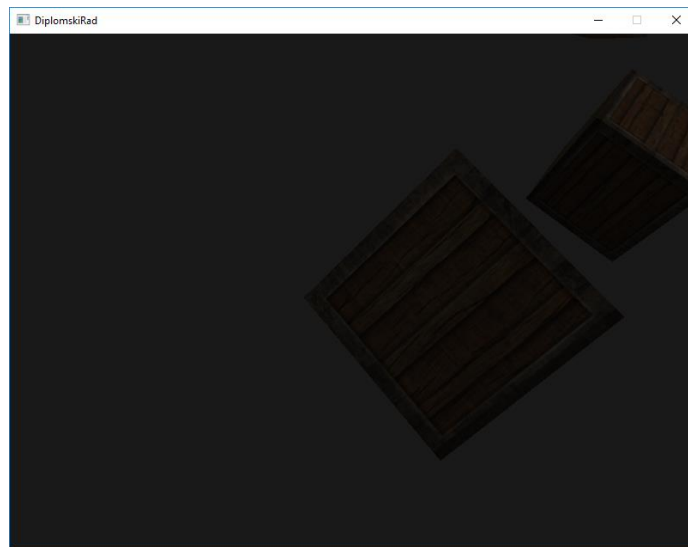
Slika 5.2.3. Izgled kutije s ekspozicijom 1.



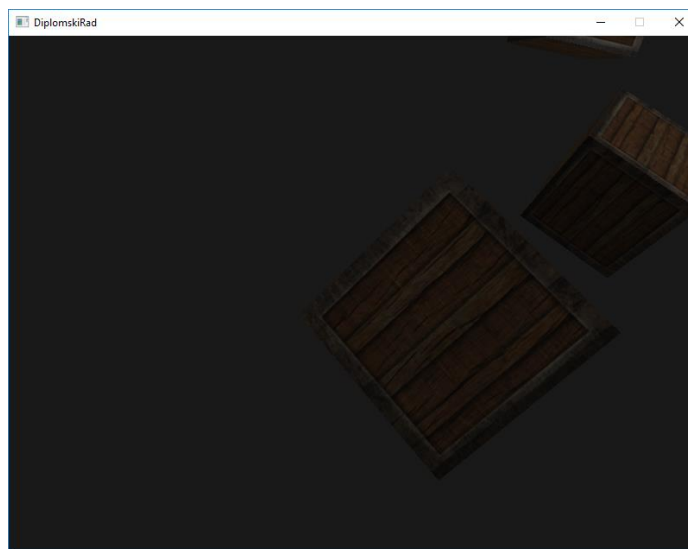
Slika 5.2.4 Izgled kutije s ekspozicijom 5

Na slikama se vidi očita promjena s mijenjanjem ekspozicije koja je neophodna za efektivan rad grafičkih proizvoda. Inače se to u HDR algoritmima radi samopodešavajuće, ovisno o prosječnom osvjetljenju cijele scene. Od metoda koje sam implementirao, Reinhard metoda je samopodešavajuća i podešava osvjetljenje na temelju ukupnog osvjetljenja scene.

5.3.Reinhardov samopodešavajući doprinos



(a)

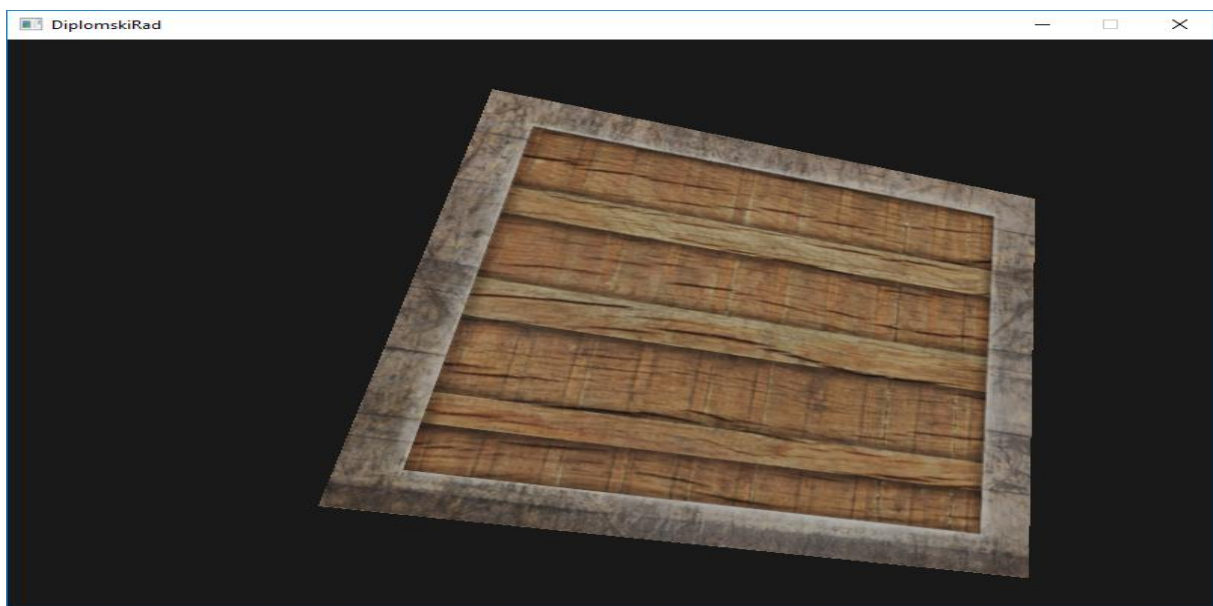
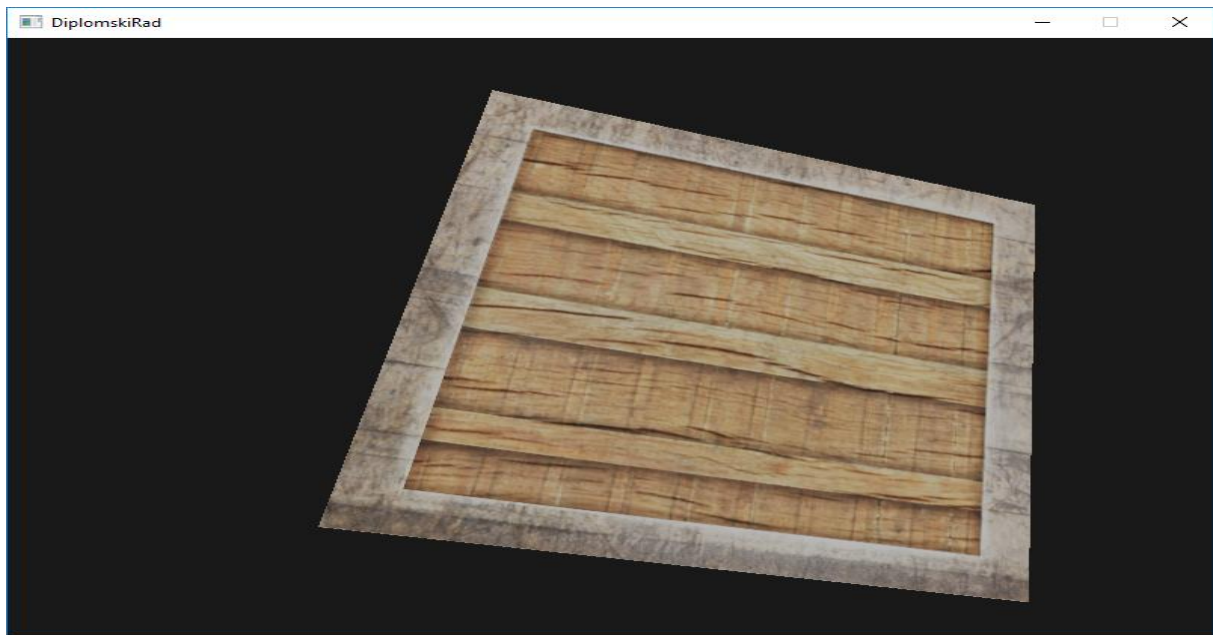


(b)

Slika 5.3.1. (a) osvjetljenje u mračnoj sceni bez korištenja Reinhard metode; (b) osvjetljenje u mračnoj sceni s korištenjem Reinhard metode.

Utjecaj samopodešavajućeg aspekta Reinhard algoritma se vidi na slici 5.3.1. Na slici 5.3.1. (a) je prikazana scena kada je osvjetljenje jako slabo i ne koristi se Reinhard. No pošto Reinhardov algoritam se bazira na prosječnoj osvjetljenosti scene, on automatski pojačava vrijednosti kako bi se bolje vidjelo. Slično kao i čovjek kada uđe u mračnu prostoriju pa nakon nekog vremena može bolje vidjeti. Utjecaj je sličan i za

jako svjetle scene.



Slika 5.3.2. Gornja slika: osvjetljenje u jako svjetloj sceni bez korištenja Reinhard metode;
Donja slika: osvjetljenje u jako svjetloj sceni s korištenjem Reinhard metode.

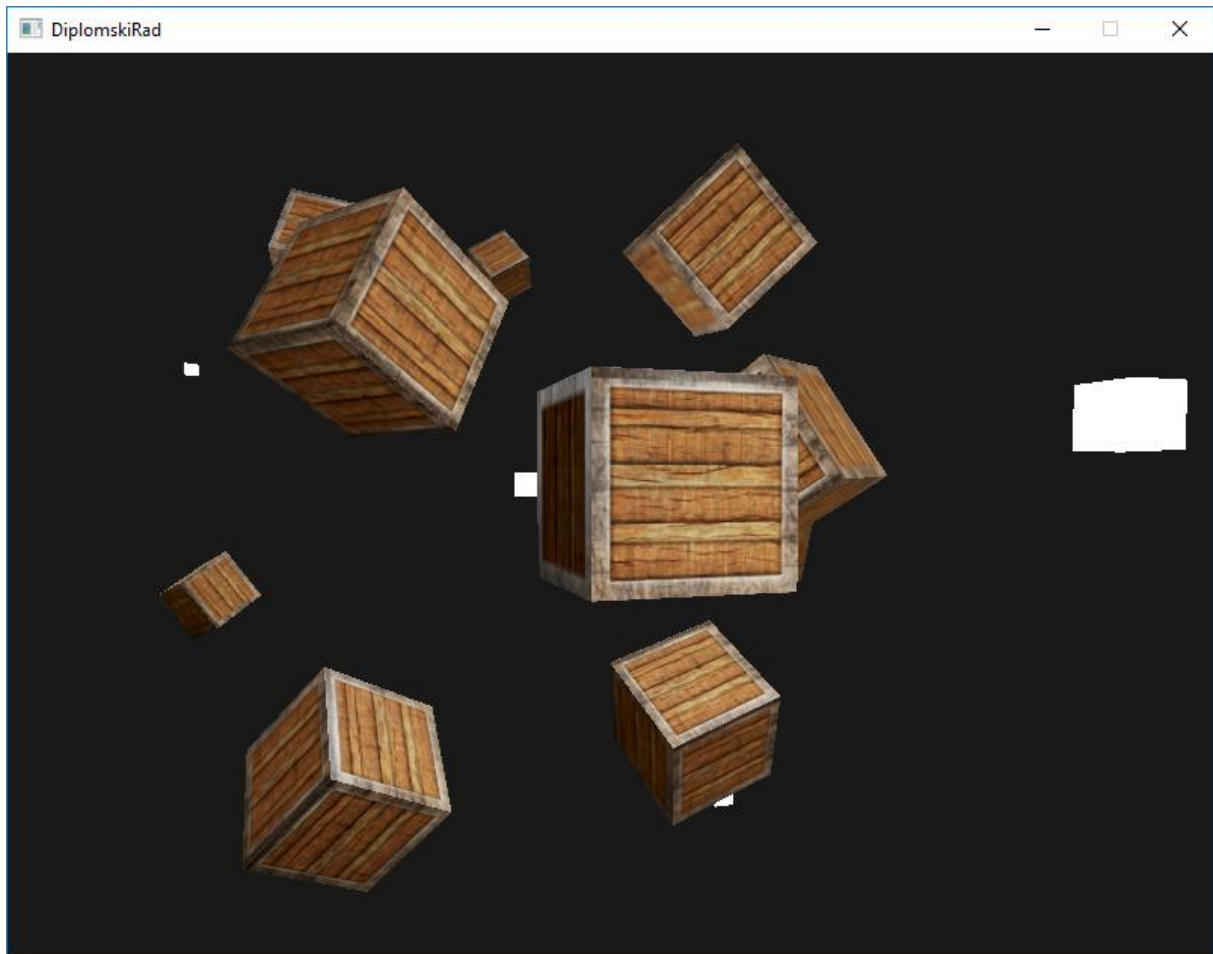
Kada je osvjetljenje prejako, metoda će se prilagoditi i smanjiti vrijednosti osvjetljenja. Također isto kao i čovjekove oči. U tome leži najveća ljepota HDR algoritma.

5.4. Burgess metoda za preslikavanje tonova boje [9]

Za ovu tehniku nisam pronašao objašnjenje matematičkog izvoda, ali sam ju iskoristio i daje zanimljive rezultate u odnosu na druge metode koje sam isprobao.

$$\text{maxColor} = \max(0, \text{color} - 0.004)$$

$$\text{Color} = \frac{(\text{maxColor} * (6.2 * \text{maxColor} + 0.05))}{(\text{maxColor} * (6.2 * \text{maxColor} + 2.3) + 0.06)}$$



Slika 5.3.1. Burgess metoda na sceni s kutijama.

Burgess metoda je dosta kontrastna. Svjetlije boje čini još svjetlijima, a mračne ostavlja mračnima, iako čuva detalje u njima.

5.4. Uncharted2 metoda preslikavanja tonova boje

Uncharted2 metoda se još zove i filmska metoda. Izračun je ovakav:

$$F(x) = \frac{(x * (A * x + C * B) + D * E)}{x * (A * x + B) + D * F} - \frac{E}{F}$$

Gdje su konstante:

$A = 0.22$

$B = 0.30$

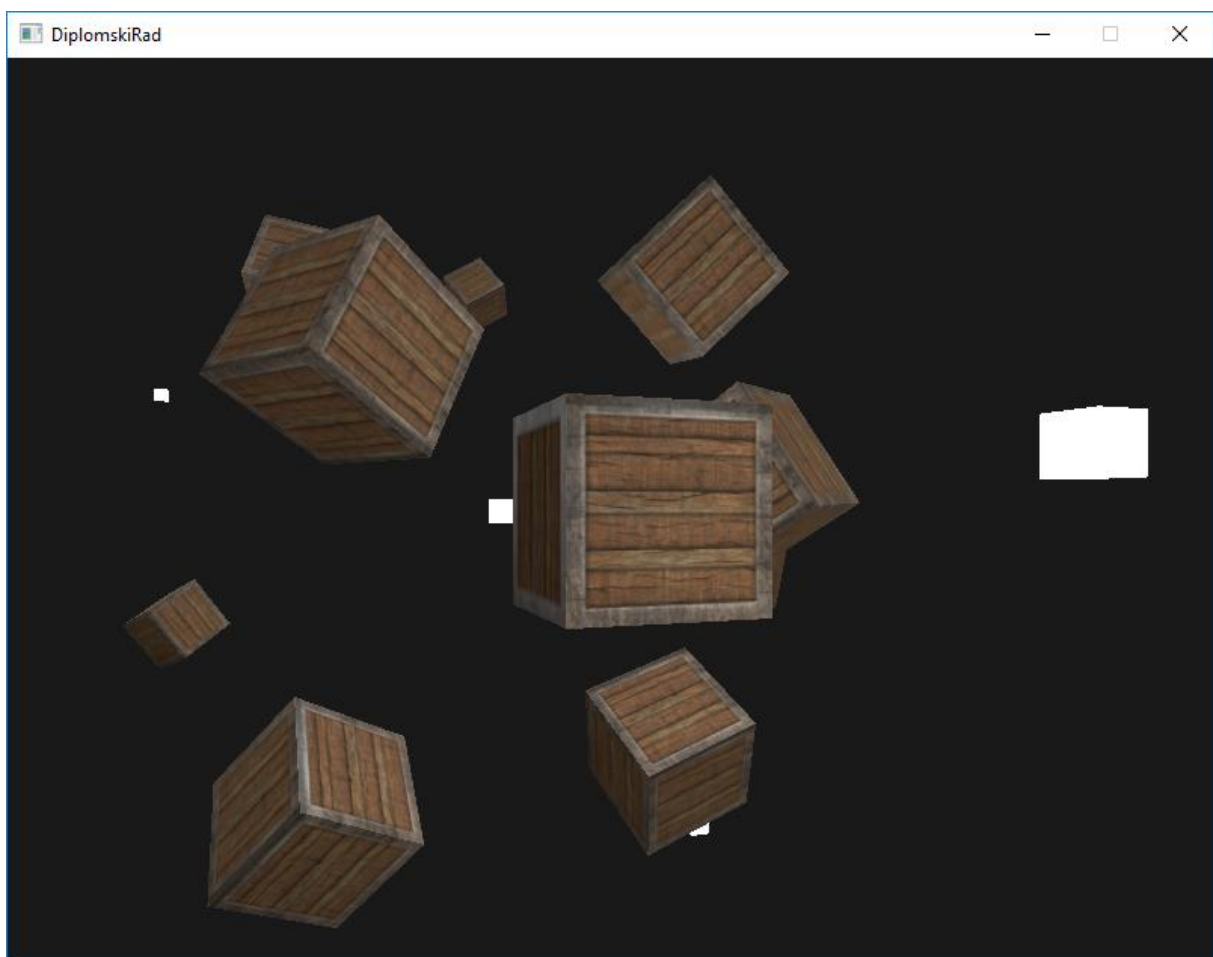
$C = 0.10$

$D = 0.20$

$E = 0.01$

$F = 0.30$

Pošto je funkcija krivulja, sve ove konstante ju oblikuju na željeni način. Mogu se i mijenjati po potrebi. Ovakve je konstante koristila kompanija NaughtyDog kod stvaranja igrice Uncharted 2.

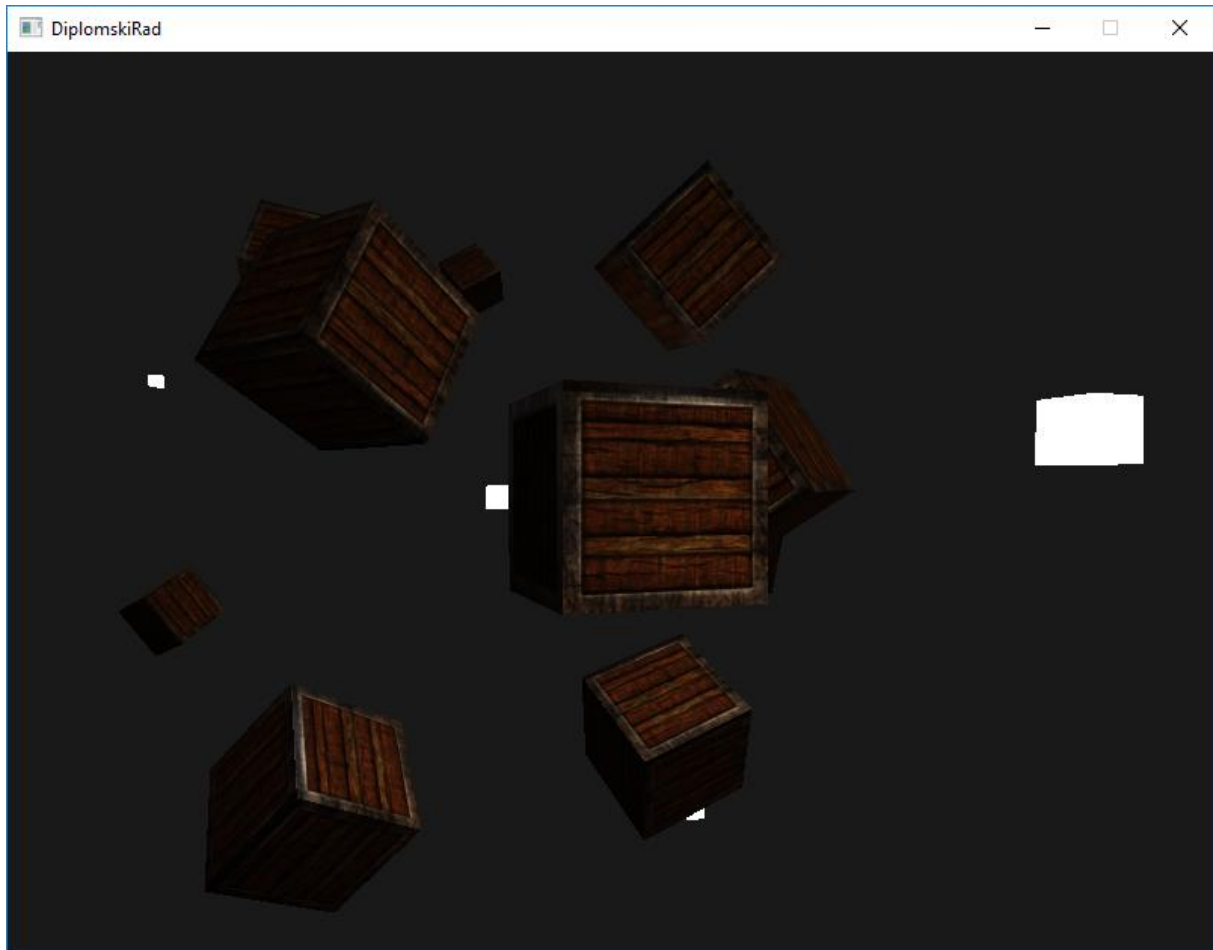


Slika 5.4.1. Uncharted2 metoda preslikavanja tonova boja

Uncharted2 metoda čini boje dosta blijedima za ovakvu scenu, i nekako mračnijima. Ali izgleda ugodno i privlačno.

5.5. Utjecaj gama korekcije

Sve što je dosad prikazano, jest prikazano koristeći gama korekciju, jer bez nje su rezultati jako loši, odnosno jako mračni. Izgled scene bez gama korekcije se vidi na slici 5.5.1.

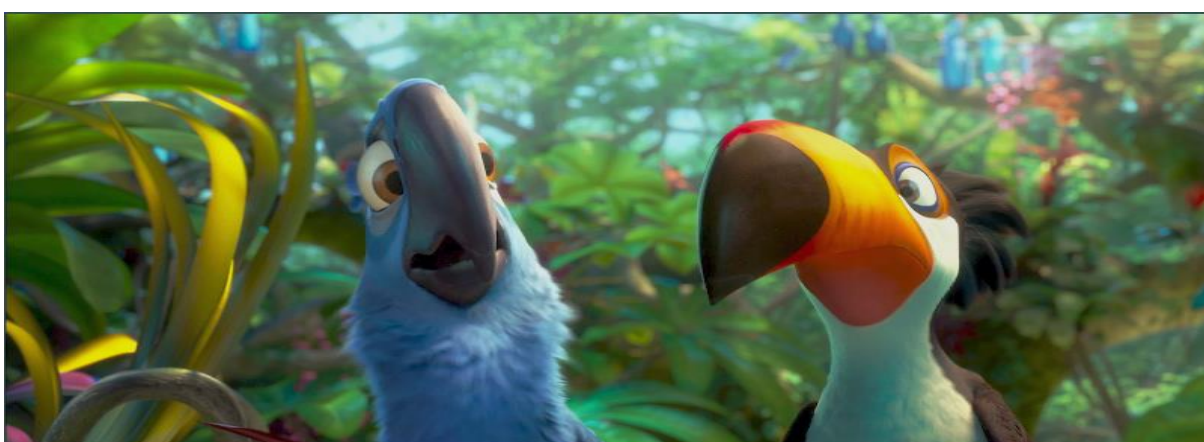
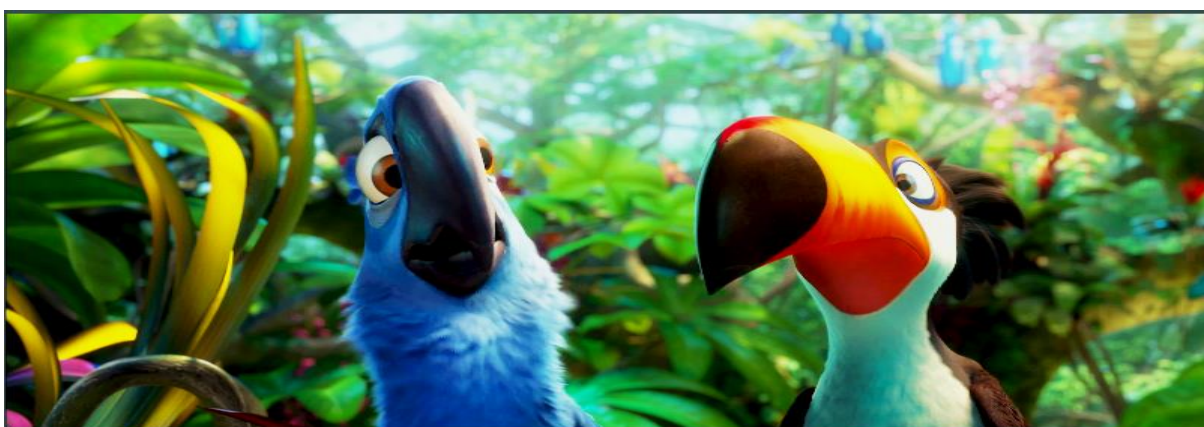
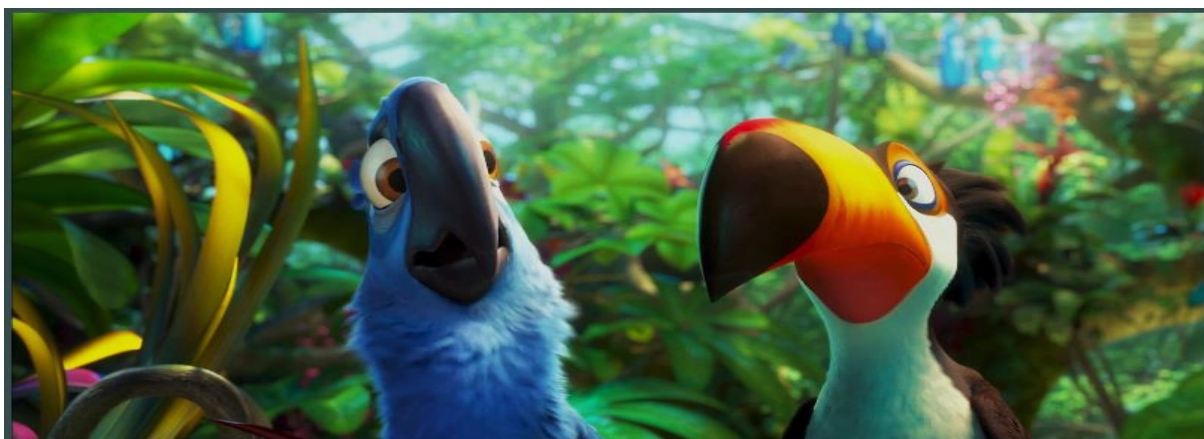


Slika 5.5.1. Scena bez gama korekcije

Kao što se vidi na slici scena izgleda jako mračno i neprivlačno jer monitor pomračuje boje. Zato se svaki put koristi gama korekcija kakva je i opisana u prijašnjim poglavljima da bi se to ispravilo.

5.6. Slika s više boja

Ovdje sam prikazano rad algoritama nad slikom s više boja, pošto je scena s kutijama dosta ograničena po pitanju boja. Na slikama bi se trebao bolje vidjeti rad metoda.



Slika 5.6.1. Najgornja – Reinhard; Sredina – Burgess; Najdonja – Uncharted2

Na slici 5.6.1. se vidi bolja usporedba implementiranih metoda za preslikavanje tonova boja iz razloga što se na slici upravo i nalazi više boja pa je lakše prepoznati razlike. Od ove 3 implementirane metode, najviše mi se sviđa Uncharted2 metoda jer daje najviše filmski i ugodan izgled gdje nema previše odskakanja određenih boja te nema zasićenosti.

Rezultati

Način na koji je stvorena HDR slika u mojoj implementaciji je dovoljno dobar da se pokaže rad algoritama za preslikavanje tonova boja.

Metode koje sam isprobao sve daju različite rezultate na sceni s kutijama. Reinhardova metoda je dobra kod sažimanja visokih vrijednosti osvjetljenja u niže, ali tamniji dijelovi nisu toliko izraženi. Burgessova metoda je malo živahnija u odnosu na ostale, i sve čini svjetlijim, dok tamnije dijelove ostavlja tamnima te stvara dojam velikog kontrasta. Uncharted2 metoda je zapravo jako izbljedjela i dosta drukčija od svih ostalih metoda za preslikavanje tonova boja koje sam pronašao na Internetu, stvara filmski doživljaj, čak i u računalnoj grafici, vjerojatno zato i pripada pod kategoriju filmskih metoda za preslikavanje tonova. Što se tiče „eye adaptation“, odnosno samopodešavajuće metode, ona efektivno odrađuje ono što bi trebala. Kada je prosječno osvjetljenje scene malo, ekspozicija će se povećati i sve će postati svjetlije. Na taj način možemo vidjeti u mraku. A ako je scensko osvjetljenje prejako, ekspozicija scene će se smanjiti pa ćemo opet moći vidjeti bolje.

Što se tiče gama korekcije, ona se uvijek treba koristiti jer inače scene izgledaju jako nerealistično.

Još postoji puno metoda koje se mogu implementirati. Namjera mi je bila i stvaranje HDR slike iz višestrukih LDR slika korištenjem mapa zračenja, pa me zanima kakve bi bile razlike, vjerojatno ne velike. Veći je utjecaj kod metoda preslikavanja tonova boja. Od metoda preslikavanja tonova boja trebalo bi implementirati neku metodu koja je lokalna pa vidjeti kakvi su tu rezultati. Vjerojatno će biti puno bolji od priloženih rezultata jer je i ljudsko oko osjetljivije na lokalni kontrast pa bi bilo realističnije, ali na račun računalnih resursa.

Zaključak

Slike visoko dinamičnog raspona su velika tema i ima jako puno različitih načina za njihovo preslikavanje u niži raspon pogodan za prikaz. Pošto preslikavanje tonova boja nije egzaktna znanost, nego se radi više o ukusu, ne može se reći da postoji najbolja metoda, nego uvijek se pita promatrača i sve ovisi o željenom učinku. Za razliku od toga, gama korekcija jest uvijek poželjna jer uvijek stvara prirodniji izgled tako da je se uvijek koristi. Mjesta za nove metode ima, treba vidjeti utjecaj lokalnog operatora na scenu s kutijama. Pretpostavljam da bi dijelovi koji nisu izloženi jakom svjetlu, bili puno detaljniji jer bi se zbog lokalnog djelovanja takvi dijelovi mogli puno bolje prilagoditi mračnom okruženju. Nažalost tema je velika i obuhvaća puno novog znanja kojeg je trebalo usvojiti, pa nisam stigao napraviti sve što sam naumio. Nadam se da će netko nastaviti raditi s ovime jer se još da puno napraviti.

Literatura

[1] Visoko dinamični raspon kod slika, wikipedia,

https://en.wikipedia.org/wiki/High-dynamic-range_imaging

[2] Preslikavanje tonova boja, wikipedia,

https://en.wikipedia.org/wiki/Tone_mapping

[3] Gamma korekcija, wikipedia,

https://en.wikipedia.org/wiki/Gamma_correction

[4] Photographic Tone Reproduction for Digital Images, Reinhard,

http://www.cmap.polytechnique.fr/~peyre/cours/x2005signal/hdr_photographic.pdf

[5] Puma: A High-Quality Retinex-Based Tone Mapping Operator,

<https://ipg.fer.hr/download/repository/puma.pdf>

[6] Learn OpenGL,

<https://learnopengl.com/#!Introduction>

[7] Nutty software, Reinhard implementation,

http://www.nutty.ca/?page_id=352&link=hdr

[8] Mipmap, wikipedia,

<https://en.wikipedia.org/wiki/Mipmap>

[9] Filmic tonemapping operators, John Hable

<http://filmicworlds.com/blog/filmic-tonemapping-operators/>

Sažetak

Preslikavanje tonova boja kod slika visoko dinamičnog raspona

U ovom radu obrađena je tema preslikavanja tonova boja kod slika visoko dinamičnog raspona. Dan je uvod u problematiku slika visoko dinamičnog raspona i njihovog preslikavanja u prikladni raspon za prikaz. Opisane su neke metode preslikavanja tonova boje i prikazane implementacije. Također je prikazan i utjecaj gama korekcije na konačnu boju slike.

Ključne riječi: računalna grafika, slike visoko dinamičnog raspona, HDR, gama, preslikavanje tonova boje

Abstract

Tone mapping of high dynamic range images

In this paper the topic of tone mapping of high dynamic range images is analyzed. A introduction to the problem of high dynamic range images and their tone mapping into suitable range is given. Several methods of tone mapping are described and shown in implementation. Also, an effect of gamma correction to the final colors in image is shown

Keywords: computer graphics, High dynamic range imaging, HDR, Tone mapping, gamma