

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 497

**NEFOTOREALISTIČNA TEHNIKA PRIKAZA
VIDEO ZAPISA**

Matija Forko

Zagreb, lipanj 2009.

Sadržaj

Uvod.....	1
1. Pregled temeljnih postupaka za ostvarivanje nefotorealističnog prikaza.....	2
2. Boje i slikovni formati.....	4
2.1. Modeli boja.....	4
2.2. RGB model.....	4
2.3. Lab model ($L^*a^*b^*$, CIELAB).....	5
2.4. Bitmap format slike.....	6
3. Filtriranje slike.....	8
3.1. Operacije na slikovnim elementima.....	8
3.1.1. Invertiranje slike - negativ.....	8
3.1.2. Pretvaranje boja u sive tonove.....	8
3.1.3. Gamma transformacija.....	9
3.1.4. Normalizacija.....	9
3.1.5. Spajanje slojeva slike.....	10
3.2. Prostorne operacije.....	11
3.2.1. Gaussov filtar.....	12
3.3.1. Selektivni Gaussov filtar.....	15
3.3.2. Filtar razlike Gaussovih filtara (DoG).....	17
3.3.3. Sobelov filtar.....	17
4. Opis temeljnih postupaka za ostvarivanje nefotorealističnog prikaza.....	19
4.1. Pretvorba iz RGB modela u $L^*a^*b^*$ model.....	19
4.2. Pretvorba iz $L^*a^*b^*$ modela u RGB model.....	20
4.3. Otklanjanje šuma i apstrakcija slike.....	21
4.4. Kvantizacija svjetline.....	22

4.5.	Detekcija i naglašavanje rubova.....	24
5.	Programska implementacija.....	27
5.1.	Komponenta VideoCog	27
5.2.	Rad sa slikama u programskom jeziku C#	28
5.3.	Osnovne klase	29
6.	Rezultati.....	31
7.	Zaključak	33
8.	Literatura	34

Uvod

Za razliku od tradicionalne računalne grafike koja teži fotorealizmu, odnosno što vjernijem prikazu na računalu, nefotorealistična tehnika prikaza (NPR, engl. *Non-photorealistic rendering*) područje je računalne grafike koje je inspirirano raznim umjetničkim stilovima. Tako su razvijene tehnike nefotorealističnog prikaza koje oponašaju različite slikarske pravce (primjerice impresionizam ili ekspresionizam), tehničke crteže, crtane filmove, koje daju zanimljive rezultate, kao što prikazuje slika 0.1. Ovo područje našlo je primjenu, između ostaloga u filmskoj industriji (npr. film „Scanner Darkly“) i računalnim igrama (npr. „Team Fortress II“).



Slika 0.1 Usporedba obične slike i slike obrađene jednom od tehnika nefotorealističnog prikaza.

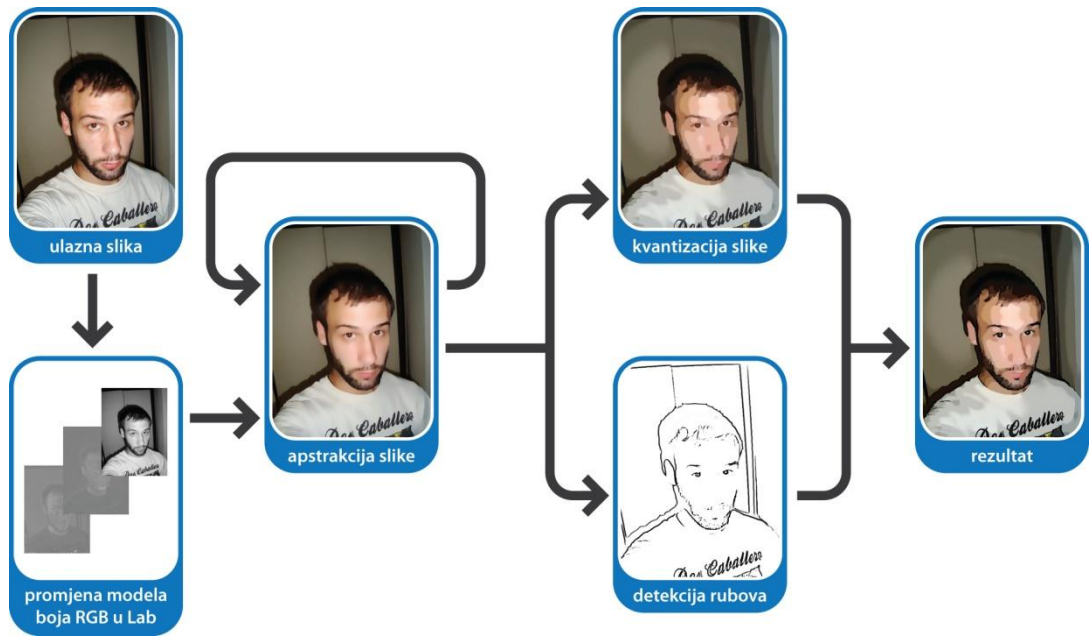
U radu su objašnjeni postupci za ostvarivanje nefotorealističnog prikaza video zapisa i slika u stilu crtanog filma, koji su u praktičnom dijelu i implementirani.

1. Pregled temeljnih postupaka za ostvarivanje nefotorealističnog prikaza

S obzirom na to da postoji mnogo različitih tehnika nefotorealističnog prikaza, postoji i mnogo postupaka za ostvarivanje takvog prikaza. Temeljni postupci za ostvarivanje nefotorealističnog prikaza u stilu crtanog filma uključuju sljedeće radnje [1]:

- Pretvorbe modela boja ($RGB \rightarrow L^*a^*b^* \rightarrow RGB$)
- Otklanjanje šuma i apstrakciju slike
- Kvantizaciju boja
- Detekciju i naglašavanje rubova
- Spajanje više slojeva slike u jednu sliku

Budući da se obrada video zapisa svodi na izvođenje iste operacije za svaku sličicu video zapisa, svi daljnji postupci odnose se na obradu slika i podrazumijeva se da se video zapis obrađuje na isti način. Dijagram toka obrade slike prikazuje slika 1.1. U sljedećem poglavlju dan je pregled korištenih modela boja, dok će u narednim poglavljima detaljno biti opisani postupci filtriranja slike i svaki od navedenih koraka obrade.



Slika 1.1 Dijagram toka obrade slike.

2. Boje i slikovni formati

Prije detaljnijeg opisa postupaka potrebnih za ostvarivanje nefotorealistične tehnike prikaza video zapisa, potrebno je opisati osnovne korištene modele boja i njihove prednosti i mane.

2.1. Modeli boja

Model boja je apstraktni matematički model koji opisuje kako se boje mogu prikazati kao uređene n -torke brojeva, koje se obično sastoje od tri do četiri vrijednosti ili komponente boje. Svaka n -toraka predstavlja jednu boju te se kombiniranjem različitih vrijednosti dobivaju različite boje. Ovisno o modelu, možemo prikazati veći ili manji broj boja.

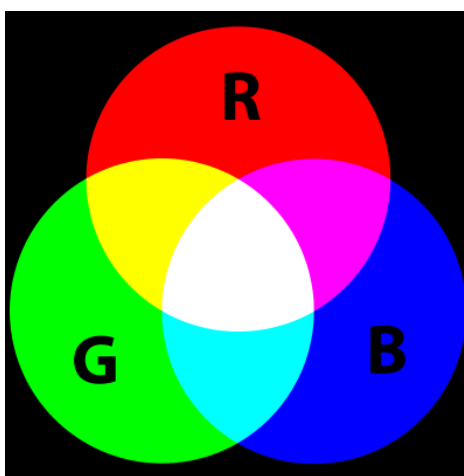
Dva temeljna modela boja su aditivni i subtraktivni. Razlikuju se po načinu dobivanja boje iz komponenti. Kod aditivnog modela boja se dobiva zbrajanjem pojedinih komponenti (npr. RGB model), a kod subtraktivnog (npr. CMYK model) oduzimanjem.

2.2. RGB model

Prvi i osnovni model koji je nastao iz pokušaja da se imitira način na koji čovjek vidi boje je RGB model. Akronim RGB predstavlja riječi *Red* (crvena), *Green* (zelena) i *Blue* (plava). Čovjek ima osjetila za tri osnovne boje: crvenu, zelenu i plavu. Zbog toga se pretpostavilo da se sve boje mogu prikazati kao linearna kombinacija tih triju osnovnih boja. Da bi opisali jednu boju, potrebno je poznavati tri informacije: koliki je udio crvene, zelene i plave komponente. Svaka komponenta prikazuje se nizom od 8 bitova, što nam daje $2^8 = 256$ različitih vrijednosti. Budući da postoje tri komponente, možemo prikazati 256^3 različitih boja.

RGB je aditivni model, što znači da se boje dobivaju zbrajanjem komponenti (Slika 2.1). Ako sve tri komponente iznose nula, dobit ćemo crnu boju, a ako sve komponente iznose maksimum (255), dobit ćemo bijelu boju. Različitim kombinacijama pojedinih komponenata dobit ćemo različite boje. Ako su komponente jednake po iznosu, dobit ćemo nijanse sive boje.

Glavna primjena ovog modela je u prikazivanju slika na računalnom i televizijskom ekranu. Treba napomenuti da je RGB model ovisan o uređaju na kojem se prikazuje. To znači da se iste boje ne moraju nužno prikazivati na isti način u različitim uređajima (npr. zbog različitih svojstava fosfora u ekranima).



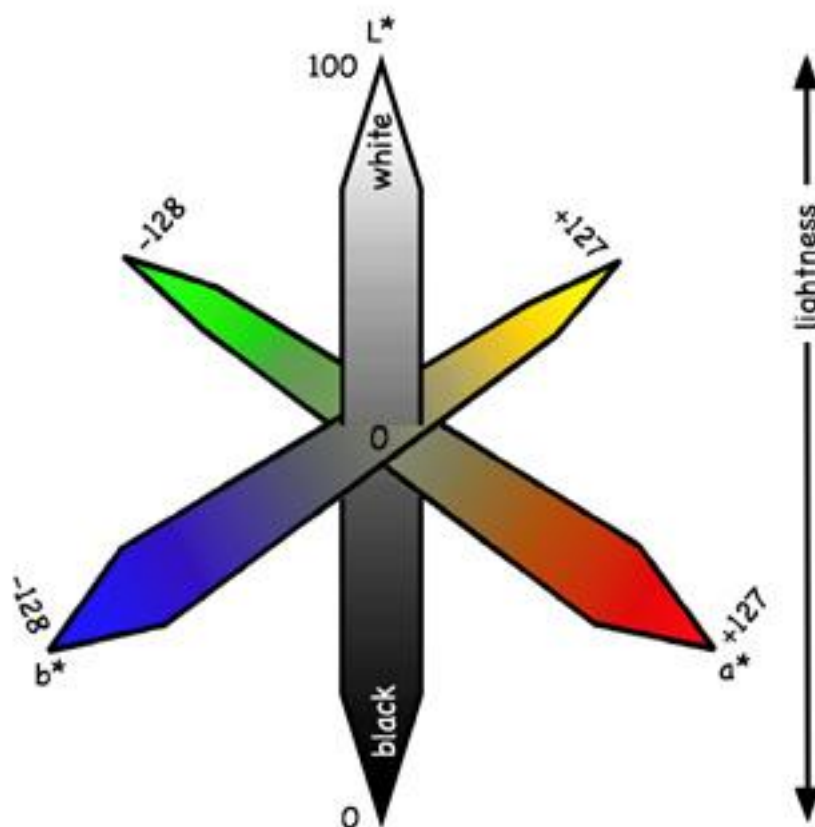
Slika 2.1 Dobivanje boja u RGB modelu.

2.3. Lab model ($L^*a^*b^*$, CIELAB)

Lab model nastao je s ciljem što bolje aproksimacije ljudskog vida i postizanja perceptivne uniformnosti, što znači da bi promjena boje za određenu vrijednost trebala rezultirati jednakom promjenom vizualnog značaja [2].

Kratice Lab, koja se danas uobičajeno koristi umjesto punog naziva $L^*a^*b^*$, označava tri komponente ovog modela koje prikazuje slika 2.2. L^* komponenta određuje svjetlinu (engl. *Lightness*, vrijednost nula kao rezultat daje crnu, a vrijednost 100 bijelu boju), a^* komponenta odnos između crvene i zelene boje (negativne vrijednosti označavaju zelenu, a pozitivne crvenu), a b^* komponenta odnos između žute i plave boje (negativne vrijednosti za plavu, a pozitivne za

žutu). Vrijednosti komponenata kreću se u rasponu od 0 do 100 za L^* i -128 do 127 za a^* i b^* .



Slika 2.2 Prikaz $L^*a^*b^*$ modela boja.

L^* komponenta Lab modela dobro aproksimira ljudsku percepciju svjetline i neovisna je od a^* i b^* komponenti koje određuju boje. Filtriranje a^* i b^* komponenti ima utjecaj samo na boje, dok oština detalja ovisi samo o L^* komponenti. Ovo svojstvo je velika prednost u odnosu na RGB model, zbog toga što omogućuje točnije podešavanje boja, svjetline i kontrasta slike.

2.4. Bitmap format slike

Bitmap (BMP) format je format slike koji se koristi za pohranu slika s metodom opisivanja svakog slikovnog elementa određenim brojem bita. Kod BMP formata slikovni elementi mogu se opisati dubinom boje od 1, 4, 8, 16, 24 ili 32 bita

po slikovnom elementu. BMP je ne-komprimirani format te su datoteke ovog formata u pravilu puno veće od datoteka forma koji se služe nekom vrstom kompresiju.

Ako se koristi jedan bit za određivanje vrijednosti slikovnog elementa, moguće je prikazati samo 2 boje – crnu i bijelu. S 4 bita moguće je prikazati nijanse sive, a s više od 4 bita mogu se prikazati slike u boji. Ako se koriste 24 bita za opis boje, BMP koristi klasični RGB model, dok se za 32 bita koristi RGB model s dodatnom komponentom – prozirnošću.

Podaci u BMP zaglavlju podijeljeni su u četiri bloka:

- BMP zaglavlje (engl. *header*) – općeniti podaci o datoteci (npr. veličina)
- Informacije o Bitmapi – detaljniji opis datoteke
- Tablica boja – popis boja ako se koristi indeksirani pristup opisivanja vrijednosti slikovnog elementa
- Bitmap podaci – vrijednosti svakog pojedinog slikovnog elementa

Svakako najvažniji sadržaj ovog formata su vrijednosti slikovnih elemenata. Ako se koriste 24 bita za opis boje, u datoteci su izravno zapisane vrijednosti za svaki slikovni element u RGB formatu. Vrijednosti se zapisane tako da se počevši od donjeg lijevog kuta slike, redak po redak dolazi do zadnjeg slikovnog elementa koji se nalazi u gornjem desnom kutu.

3. Filtriranje slike

Filtriranje je postupak obrade slike kod kojeg izlazna vrijednost slikovnog elementa ovisi o jednoj ili više ulaznih vrijednosti slikovnih elemenata. Filtriranjem slike mogu se poboljšati određene značajke slike (npr. korekcija boja, podešavanje svjetline i kontrasta), ukloniti nedostaci (npr. otklanjanje šuma slike) ili dobiti razni zanimljivi efekti na slici (npr. detekcija rubova).

Postupci filtriranja slike mogu se podijeliti u dvije osnovne skupine:

- Operacije na slikovnim elementima (engl. *point operations*)
- Prostorne operacije (engl. *spacial operations*)

3.1. Operacije na slikovnim elementima

Zajednička karakteristika ovih operacija je da izlazna vrijednost slikovnog elementa ovisi samo o ulaznoj vrijednosti tog istog slikovnog elementa.

To su operacije koje ulaznu vrijednost slikovnog elementa preslikavaju u izlaznu vrijednost prema transformaciji

$$izlaz = f(ulaz)$$

3.1.1. Invertiranje slike - negativ

Negativ slike dobiva se oduzimanjem ulazne vrijednosti slikovnog elementa od maksimalne moguće vrijednosti. Maksimalna vrijednost kod RGB modela iznosi 255 pa se negativ dobiva oduzimanjem vrijednosti svake komponente od 255.

$$izlaz = f(ulaz) = maxIntenzitet - ulaz$$

3.1.2. Pretvaranje boja u sive tonove

RGB sliku u boji pretvaramo u sive tonove tako da vrijednosti R, G i B komponenti postavimo na istu vrijednost. Ta vrijednost jednaka je sumi umnožaka

R, G i B komponenti s odgovarajućim težinskim faktorima koji su određeni zbog različite osjetljivosti oka na komponente boja. Faktori iznose 0.299 za crvenu komponentu, 0.587 za zelenu i 0.114 za plavu komponentu.

$$izlaz_{R,G,B} = f(ulaz) = ulaz_R \cdot 0.299 + ulaz_G \cdot 0.587 + ulaz_B \cdot 0.114$$

Puno je jednostavnije pretvoriti L*a*b* sliku u boji u sive tonove. Budući da samo a* i b* komponente sadrže podatke o boji slikovnog elementa, potrebno je njihove vrijednosti postaviti na vrijednost nula.

3.1.3. Gamma transformacija

Gamma transformacija je transformacija linearno rastućih ulaznih vrijednosti u eksponencijalno rastuće ili padajuće izlazne.

$$izlaz = f(ulaz) = maxIntenzitet \cdot \left(\frac{ulaz}{maxIntenzitet} \right)^{\frac{1}{\gamma}}$$

Gdje je γ koeficijent koji određuje izlazne razine boja. Za vrijednosti manje od jedan, boje će biti svjetlije, a za vrijednosti veće od jedan tamnije.

3.1.4. Normalizacija

Normalizacija je linearno proširivanje raspona vrijednosti boja slike. Ako se neka od komponenti slike ne pojavljuje u punom rasponu, tada se proširenjem na puni raspon povećava kontrast slike. Normalizacija se obavlja za svaku komponentu posebno.

Izlazna vrijednost slikovnog elementa ovisi o ulaznoj vrijednosti, minimalnoj i maksimalnoj vrijednosti komponente slikovnog elementa koja se pojavljuje na slici i željenom izlaznom rasponu [3].

$$izlaz = f(ulaz) = minIzlaz + \frac{ulaz - minUlaz}{maxUlaz - minUlaz} \cdot (maxIzlaz - minIzlaz)$$

Gdje je:

- minUlaz – minimalna vrijednost komponente na ulaznoj slici
- maxUlaz – maksimalna vrijednost komponente na ulaznoj slici
- minIzlaz – minimalna vrijednost komponente koju želimo na izlazu

- $maxI_{zlaz}$ – maksimalna vrijednost komponente koju želimo na izlazu

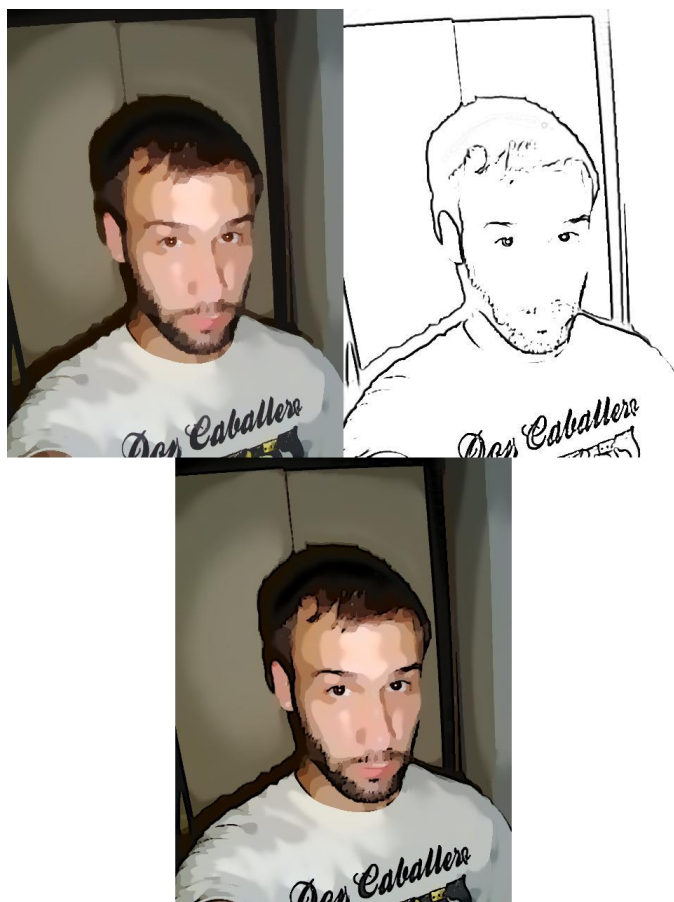
3.1.5. Spajanje slojeva slike

Kad je potrebno više slojeva slike spojiti u jednu sliku, koristi se neka od operacija spajanja slojeva. Na primjer, ako se na jednom sloju slike nalaze naglašeni rubovi, a na drugom izvorna slika, rezultat spajanja ta dva sloja je izvorna slika s naglašenim rubovima (Slika 3.1).

Najjednostavniji način spajanja dva sloja slike je međusobnim množenjem slikovnih elemenata slika i dijeljenjem s maksimalnom mogućom vrijednošću. Kod RGB modela, međusobno se množe komponente i dijele sa 255.

$$izlaz = f(ulaz1, ulaz2) = \frac{ulaz1 \cdot ulaz2}{maxIntenzitet}$$

Budući da je izlazna vrijednost obično manja od obje ulazne vrijednosti, slika dobivena kao rezultat ove operacije je malo tamnija od ulaznih slika.



Slika 3.1 Spajanje dva sloja slike u jednu sliku.

3.2. Prostorne operacije

Zajednička karakteristika prostornih operacije je da izlazna vrijednost slikovnog elementa ovisi o ulaznim vrijednostima u nekoj okolini (susjedstvu) tog slikovnog elementa. Ove operacije izvode se računanjem konvolucije slike s jezgrom filtra kojom je određeno ponašanje filtra.

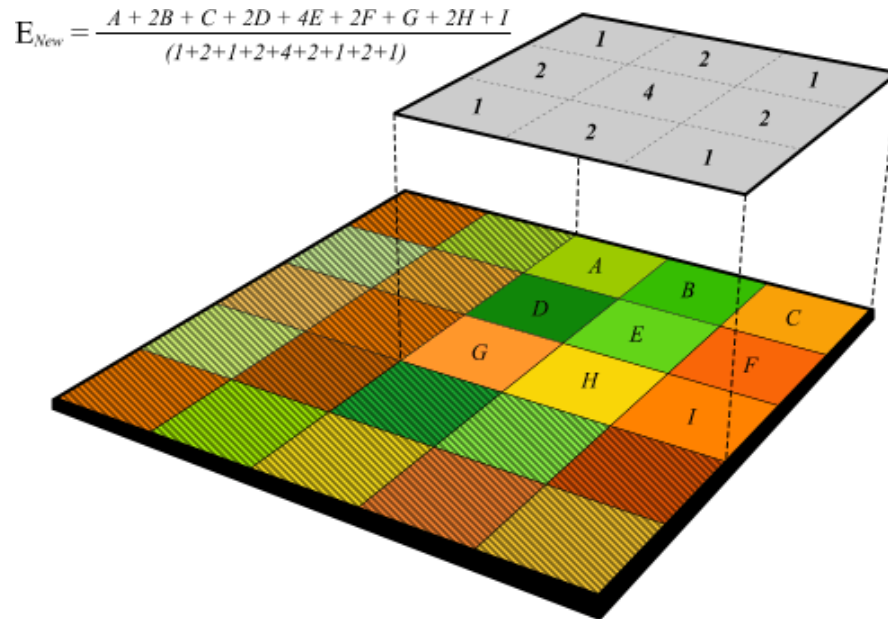
Jezgra filtra je matrica koja sadrži težinske faktore kojima se množe vrijednosti slikovnih elemenata kod računanja nove vrijednosti pojedinog slikovnog elementa. Vrijednost izlaznog slikovnog elementa određena je formulom:

$$slikaIzlaz[x, y] = \sum_{i=-m}^m \sum_{j=-n}^n jezgraFiltra[i, j] \cdot slikaUlaz[x + i, y + j]$$

Ponekad je operaciju konvolucije slike s matricom jezgre filtra (Slika 3.2), moguće rastaviti na dvije uzastopne operacije konvolucije s vektorom jezgre filtra. Ovo svojstvo naziva se separabilnost i ovisno o veličini jezgre filtra, rezultira većim ili manjim ubrzanjem računanja. Vrijednost izlaznog slikovnog elementa tada je određena ovim formulama:

$$slikaIzlazX[x, y] = \sum_{i=-m}^m jezgraFiltra[i] \cdot slikaUlaz[x + i, y]$$

$$slikaIzlaz[x, y] = \sum_{j=-n}^n jezgraFiltra[j] \cdot slikaIzlazX[x, y + j]$$



Slika 3.2 Ilustracija konvolucije.

Prva operacija konvolucije obavlja se u smjeru x-osi, a druga u smjeru y-osi. Važno je primijetiti da se druga operacija konvolucije ne računa ponovno s ulaznom slikom, nego sa slikom koja je dobivena kao rezultat prve operacije konvolucije.

Kod definiranja jezgre filtra važno je pridržavati se sljedećih pravila:

- Dimenzije jezgre filtra moraju biti neparne kako bi se jezgra mogla centrirati na određeni slikovni element.
- Jezgra filtra mora biti normalizirana, što znači da zbroj svih elemenata jezgre mora biti jednak jedan. Ako je zbroj elemenata manji od jedan, nova slika bit će tamnija, a ako je zbroj veći od jedan, nova slika će biti svjetlija. Ponekad je zbog jednostavnosti jezgra filtra definirana tako da zbroj elemenata nije jednak jedan, u tom slučaju se normalizacija provodi tako da se rezultat konvolucije podijeli sa zbrojem vrijednosti svih elemenata jezgre, kao što i prikazuje slika 3.2.

3.2.1. Gaussov filter

Gaussov filter je niskopropusni filter koji se često koristi kod obrade slika. Svojstva Gaussovog filtra su da smanjuje šum u slici (zbog toga što se šum

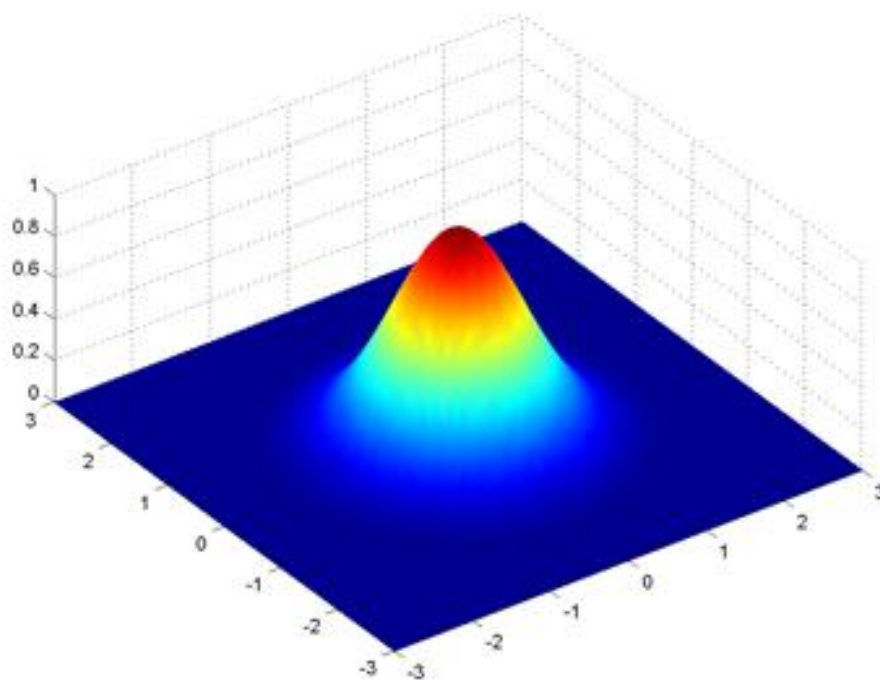
uglavnom sastoji od visokih frekvencija koje Gaussov filtar ne propušta) i smanjuje količinu detalja, odnosno zamućuje sliku [4].

Jezgra filtra popunjava se vrijednostima koje su izračunate po 2D Gaussovoj funkciji (Slika 3.3):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Gdje je:

- σ – standardna devijacija Gaussove funkcije
- x – udaljenost od centralnog slikovnog elementa po x osi
- y – udaljenost od centralnog slikovnog elementa po y osi

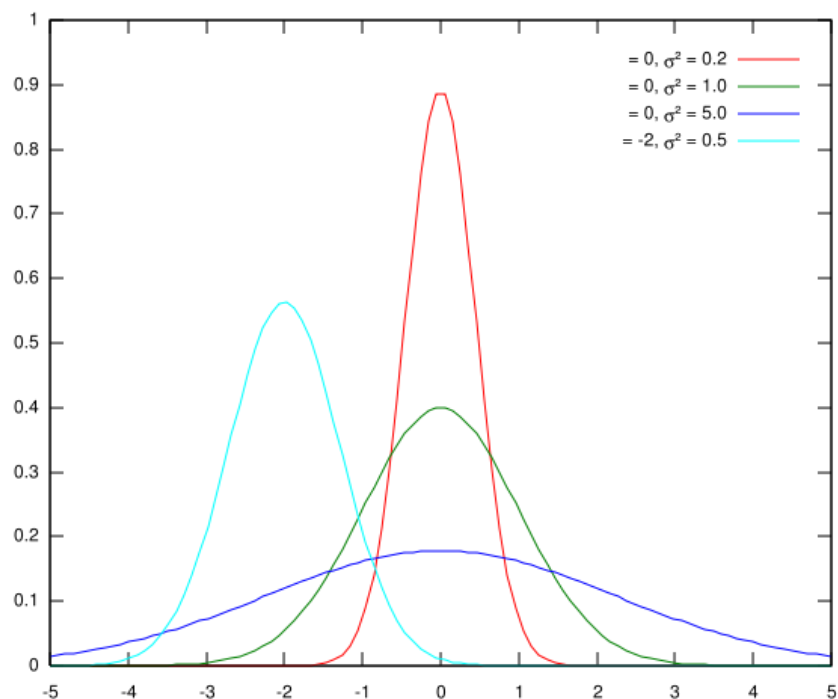


Slika 3.3 2D Gaussova funkcija.

2D Gaussova funkcija zapravo je umnožak dviju 1D Gaussovih funkcija (Slika 3.4) pa je filtriranje, umjesto matricom jezgre popunjenom vrijednostima 2D Gaussove funkcije, moguće obaviti u dva prolaza filtriranjem vektorom jezgre popunjenim vrijednostima 1D Gaussove funkcije:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

Svojstvo separabilnosti Gaussovog filtra omogućava vrlo učinkovitu implementaciju, jer je uzastopno filtriranje vektorom jezgre puno brže od filtriranja matricom jezgre.

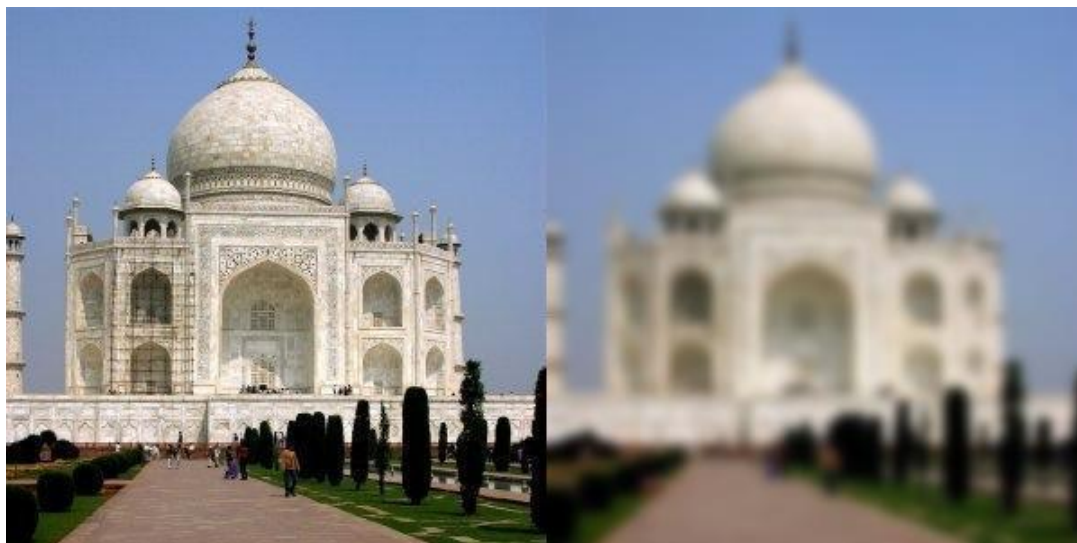


Slika 3.4 1D Gaussove funkcije.

S obzirom na to da je Gaussova funkcija definirana na beskonačnom intervalu, imat će vrijednost različitu od nule za svaku točku na slici. To znači da bi za izračun nove vrijednosti svakog slikovnog elementa morali uzeti u obzir sve slikovne elemente. U praksi je potrebno ograničiti broj slikovnih elemenata koji ulaze u proračun kako bi se povećala brzina izvođenja. Dovoljno je uzeti u obzir slikovne elemente unutar intervala 3σ sa svake strane, što osigurava točnost Gaussove razdiobe od 99.73% [5]. Duljina vektora jezgre koja je potrebna da bi se obavilo filtriranje tada iznosi:

$$duljinaVektora = 2 \cdot 3\sigma + 1$$

Kako je prilikom definiranja parametara filtriranja poželjno znati koliko će okolnih slikovnih elemenata ulaziti u izračun vrijednosti izlaznog slikovnog elementa, zadaje se polumjer djelovanja filtra, odnosno 3σ . Polumjer je jedini parametar koji je potrebno zadati kako bi se obavilo filtriranje Gausovim filtrom i brzina filtriranja ovisi samo o toj vrijednosti.

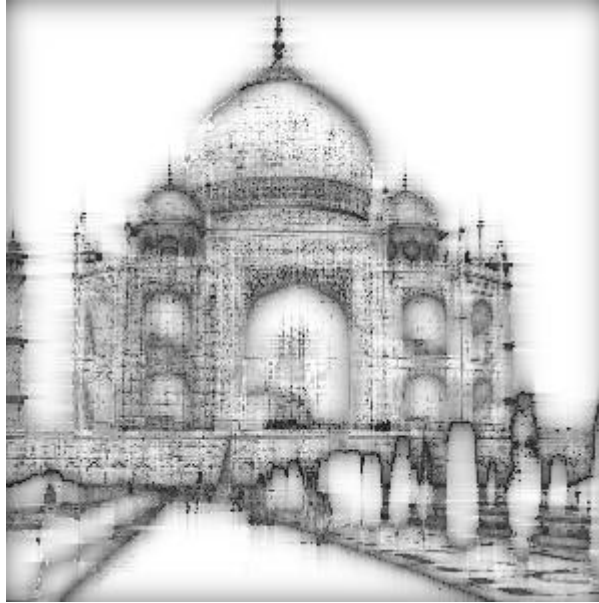


Slika 3.5 Originalna slika i slika filtrirana Gausovim filtrom (polumjer 10).

3.3.1. Selektivni Gaussov filtar

Kod filtriranja slike selektivnim Gausovim filtrom, za razliku od običnog Gaussovog filtra koji kod računanja nove vrijednosti slikovnog elementa uzima u obzir sve slikovne elemente unutar odabranog intervala, u obzir se uzimaju samo oni slikovni elementi unutar odabranog intervala koji se vizualno ne razlikuju od centralnog slikovnog elementa više od neke odabrane granice, odnosno slikovni elementi slični centralnom [6].

Ako se koristi RGB model boja, mjera sličnosti može biti relativni intenzitet slikovnog elementa (zbroj komponentata podijeljen s tri), ali bolji rezultati postižu se ako se filtriranje obavlja korištenjem $L^*a^*b^*$ modela boja, zbog perceptivne uniformnosti istoga. Dovoljno je kao mjeru sličnosti slikovnih elemenata računati razliku vrijednosti L^* komponente i uspoređivati sa zadanom granicom. Slika 3.6 prikazuje područje selekcije filtra, svjetliji su slikovni elementi koji se više zamućuju, a tamniji su manje zamućeni slikovni elementi.



Slika 3.6 Područje selekcije selektivnog Gaussovog filtra.

Kao i Gausovim filtrom, filtriranje selektivnim Gausovim filtrom može se obaviti u dva prolaza konvolucijom slike s vektorom jezgre filtra, prvo u smjeru x-osi, a zatim u smjeru y-osi.

Kao rezultat filtriranja, na novoj slici su područja sličnog intenziteta zamućena i ujednačena, a područja gdje je razlika intenziteta veća (rubovi) ostaju oštra, kao što prikazuje slika 3.7. Budući da se filtriranjem selektivnim Gausovim filtrom otklanja šum, a rubovi ostaju oštri, slika koju smo dobili nakon filtriranja pogodna je za detekciju rubova.



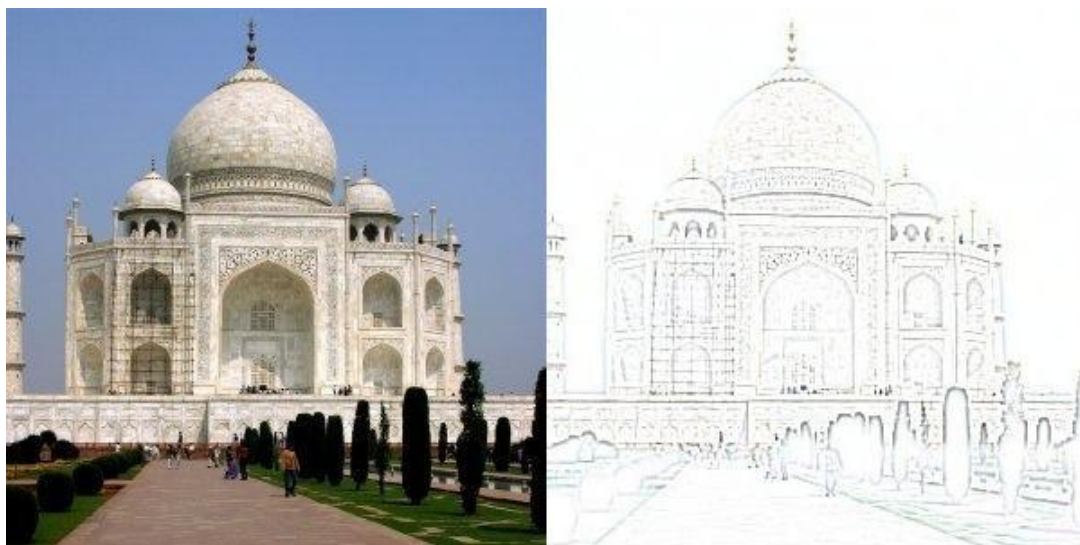
Slika 3.7 Originalna slika i slika filtrirana selektivnim Gausovim filtrom.

3.3.2. Filtar razlike Gaussovih filtara (DoG)

Filtar razlike Gaussovih filtara DoG (engl. *Difference of Gaussians*) koristi se za detekciju rubova slike, na način da oduzme rezultate dva Gaussova filtra različitih polumjera primijenjenih na sliku. Budući da je Gaussov filtar niskopropusni filtar, rezultat oduzimanja dva filtra različitih polumjera bit će područje frekvencija koje se nalazi između područja djelovanja ta dva filtra, što znači da je DoG filtar pojasno propusni filtar.

Većina filtara koji se koriste za detekciju rubova pojačava visoke frekvencije, ali kako te frekvencije često sadrže šum, takvi filtri detektiraju i nepoželjan šum. *Difference of Gaussians* filtar učinkovito rješava ovaj problem „rezanjem“ visokih frekvencija i zbog toga daje dobre rezultate uz prisutnost šuma.

Rezultat primjene ovog filtra prikazuje slika 3.8. Kako u većini slučajeva nakon primjene filtra rubovi nisu dovoljno naglašeni, potrebno je sliku dodatno obraditi (npr. normalizirati i napraviti gamma korekciju).



Slika 3.8 Originalna slika i slika s detektiranim rubovima.

3.3.3. Sobelov filtar

Još jedan filtar koji se često koristi za detekciju rubova slike jest Sobelov filtar, koji zapravo računa gradijent intenziteta u svakoj točki slike. Gradijent je veći na područjima nagle promjene intenziteta slike i ta područja često odgovaraju

rubovima, kao što prikazuje slika 3.9. Sobelov filtar definiran je s dvije jezgre veličine 3x3, jedna kojom se dobiva gradijent u smjeru x-osi, i druga kojom se dobiva gradijent u smjeru y-osi.

$$jezgraX = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad jezgraY = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$gradijentX = jezgraX * slika$$

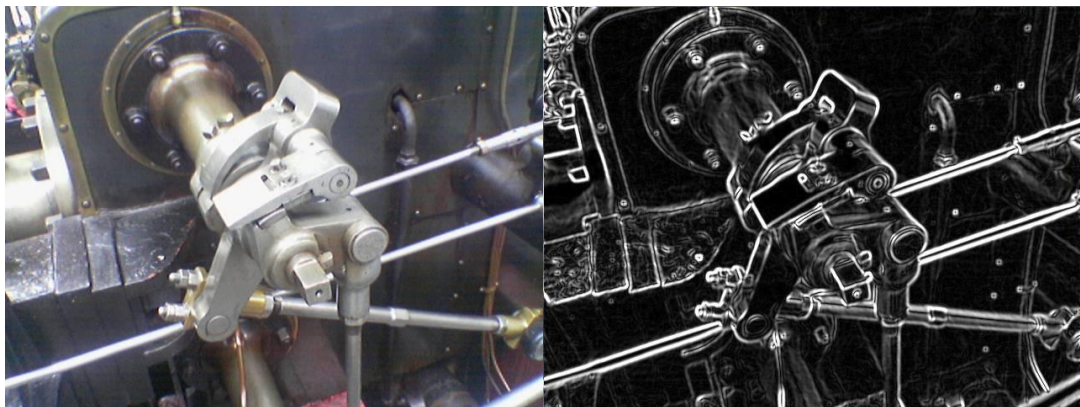
$$gradijentY = jezgraY * slika$$

Ukupan iznos gradijenta za svaki slikovni element računa se po formuli:

$$gradijent = \sqrt{gradijentX^2 + gradijentY^2}$$

A smjer gradijenta po formuli:

$$\theta = \tan^{-1} \left(\frac{gradijentY}{gradijentX} \right)$$



Slika 3.9 Originalna slika i slika dobivena filtriranjem Sobelovim filtrom.

4. Opis temeljnih postupaka za ostvarivanje nefotorealističnog prikaza

Nakon opisa filtera potrebnih za ostvarivanje nefotorealističnog prikaza u stilu crtanog filma, slijedi opis navedenih postupaka i način njihovog ostvarenja korištenjem opisanih filtera.

4.1. Pretvorba iz RGB modela u L*a*b* model

Ne postoje formule za direktnu pretvorbu boja iz RGB u L*a*b* model, zbog toga je potrebno boje prvo pretvoriti u XYZ, a zatim u L*a*b* model.

Formule za prijelaz iz RGB modela u XYZ model, pri čemu su R, G i B oznake za RGB vrijednosti prebačene u interval od 0 do 1:

$$R' = 100 \cdot f(R)$$

$$G' = 100 \cdot f(G)$$

$$B' = 100 \cdot f(B)$$

$$f(x) = \begin{cases} \left(\frac{x + 0.055}{1.055} \right)^{2.4}, & \text{za } x > 0.04045 \\ \frac{x}{12.92}, & \text{inače} \end{cases}$$

$$[X \ Y \ Z] = [R' \ G' \ B'] \cdot [M]$$

$$[M] = \begin{bmatrix} 0.4124 & 0.2126 & 0.0193 \\ 0.3576 & 0.7152 & 0.1192 \\ 0.1804 & 0.0722 & 0.9503 \end{bmatrix}$$

Formule za prijelaz iz XYZ modela u L*a*b* model, pri čemu su X_n , Y_n i Z_n referentne vrijednosti za bijelu boju:

$$L^* = 116 \cdot f\left(\frac{Y}{Y_n}\right) - 16$$

$$a^* = 500 \cdot \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right]$$

$$b^* = 200 \cdot \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right]$$

$$f(x) = \begin{cases} x^{\frac{1}{3}}, & \text{za } x > 0.008856 \\ 7.787 \cdot x + \frac{16}{116}, & \text{inače} \end{cases}$$

4.2. Pretvorba iz L*a*b* modela u RGB model

Isto kao i kod pretvorbe iz RGB u L*a*b* model, kod inverzne pretvorbe također je potrebno boje prvo pretvoriti u XYZ, a zatim u RGB model.

Formule za prijelaz iz L*a*b* u XYZ model:

$$Y' = \frac{(L^* + 16)}{116}$$

$$X' = Y' + \frac{a^*}{500}$$

$$Z' = Y' - \frac{b^*}{200}$$

$$X = X_n \cdot f(X')$$

$$Y = Y_n \cdot f(Y')$$

$$Z = Z_n \cdot f(Z')$$

$$f(x) = \begin{cases} x^3, & \text{za } x^3 > 0.008856 \\ \frac{\left(x - \frac{16}{116}\right)}{7.787}, & \text{inače} \end{cases}$$

Formule za prijelaz iz XYZ u RGB model, pri čemu su dobivene RGB vrijednosti u intervalu od 0 do 1.

$$X' = \frac{X}{100}$$

$$Y' = \frac{Y}{100}$$

$$Z' = \frac{Z}{100}$$

$$[R' \ G' \ B'] = [X' \ Y' \ Z'] \cdot [M]$$

$$[M] = \begin{bmatrix} 3.2405 & -0.9693 & 0.0557 \\ -1.5371 & 1.8758 & 0.2040 \\ -0.4985 & 0.0416 & 1.0572 \end{bmatrix}$$

$$R = f(R')$$

$$G = f(G')$$

$$B = f(B')$$

$$f(x) = \begin{cases} 1.055 \cdot x^{\frac{1}{2.4}} - 0.055, & \text{za } x > 0.0031308 \\ 12.92 \cdot x, & \text{inače} \end{cases}$$

4.3. Otklanjanje šuma i apstrakcija slike

Filtriranjem slike selektivnim Gausovim filtrom istovremeno otklanjamo šum slike i dobivamo apstrahiranu sliku. Pod pojmom apstrakcije slike misli se na pojednostavljenje nebitnih značajki slike, a pojačavanje bitnih. Nakon prvog filtriranja selektivnim Gausovim filtrom, otklonjen je šum slike i slična područja slike postala su još više ujednačena. Taj efekt još više dolazi do izražaja ako filtriranje ponovimo nekoliko puta.

Slika 4.1 prikazuje utjecaj selektivnog Gausovog filtra na sliku. Zadani parametri su polumjer = 15 i maksimalna razlika intenziteta = 10. U gornjem redu lijevo prikazana je originalna slika, a desno slika nakon prvog prolaza filtra. U donjem redu lijevo prikazana je slika nakon 2 prolaza filtra, a desno nakon četiri prolaza. Svakim prolazom filtra slična područja se sve više ujednačavaju, dok rubovi ostaju oštri. Željeni efekt obično se postiže nakon tri do četiri prolaska filtra, ovisno o parametrima.

Sljedeća dva koraka obrade slike provode se na slici dobivenoj kao rezultat ove operacije.

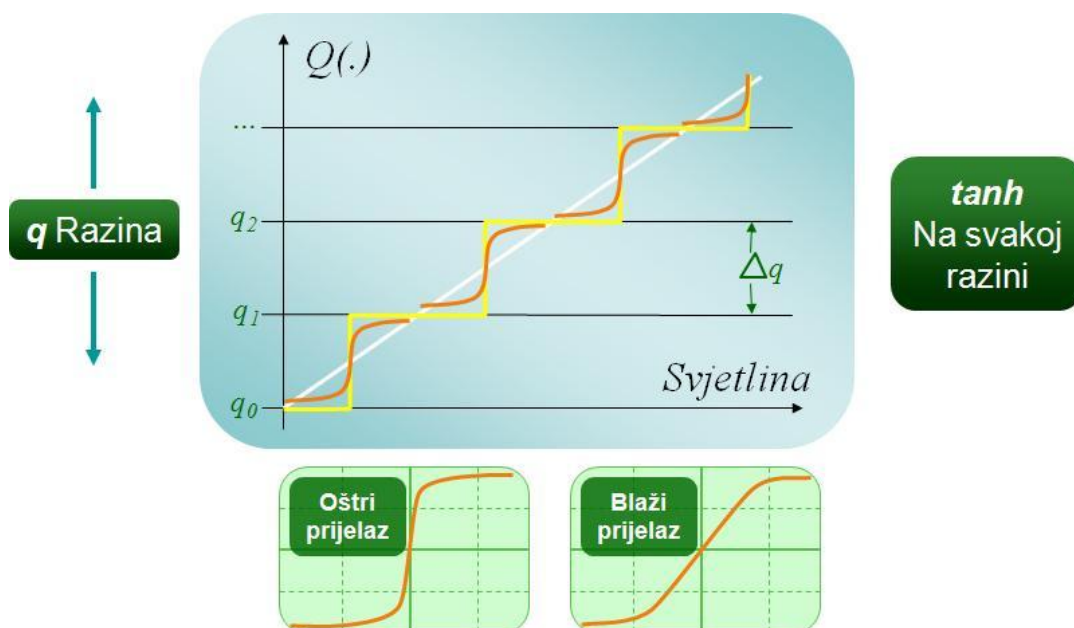


Slika 4.1 Apstrakcija slike.

4.4. Kvantizacija svjetline

Kako bi se postigao efekt crtanog filma, potrebno je kvantizirati L^* komponentu (svjetlinu) slike. Da bi se izbjegli nagli skokovi s jedne na drugu

razinu kvantizacije za male promjene boje, koristi se modificirana kvantizacija koju prikazuje slika 4.2.



Slika 4.2 Prikaz razina kvantizacije.

Izlazna razina svjetline računa se prema formuli:

$$Q(x) = q_{najbliža} + \frac{\Delta q}{2} \cdot \tanh\left(\varphi_q(x - q_{najbliža})\right)$$

Gdje je:

- x – ulazna razina svjetline
- $q_{najbliža}$ – razina kvantizacije najbliža ulaznoj razini svjetline
- Δq – širina pojedine razine kvantizacije
- φ_k – parametar koji određuje oštrinu prijelaza iz jedne razine u drugu. Za veće vrijednosti rezultat su oštri prijelazi, a za manje blaži prijelazi.

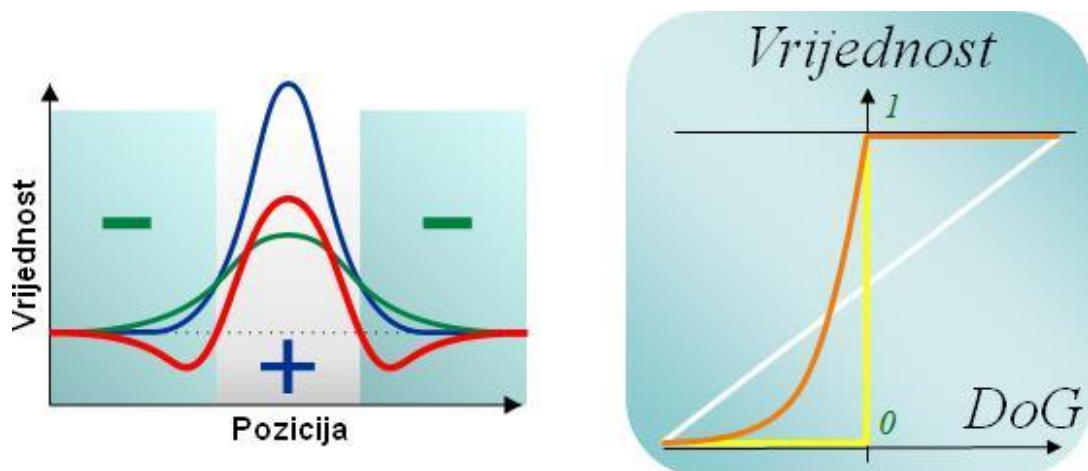
Poželjno bi bilo da nisu svi prijelazi jednako oštri ili blagi, primjerice da u pozadini prijelazi budu blaži, a oštiji na stvarima koje imaju više detalja. To se postiže tako da se parametar φ_k mijenja ovisno o iznosu gradijenta svjetline. Gradijenti svjetline dobiveni su primjenom Sobelovog filtra. Tako su dobiveni oštiji prijelazi u područjima gdje je iznos gradijenta veći (rubovi), a blaži prijelazi u okolnim područjima (Slika 4.3).



Slika 4.3 Kvantizacija svjetline.

4.5. Detekcija i naglašavanje rubova

Kao što je već spomenuto, rubovi se detektiraju DoG filtrom, ali filtar je malo modificiran kako bi se poboljšao vizualni dojam. Razliku dvije Gaussove funkcije (plavi i zeleni graf) prikazane crvenim grafom i zavisnost vrijednosti ruba o toj funkciji prikazuje slika 4.4.



Slika 4.4 DoG funkcija i zavisnost vrijednosti ruba o funkciji.

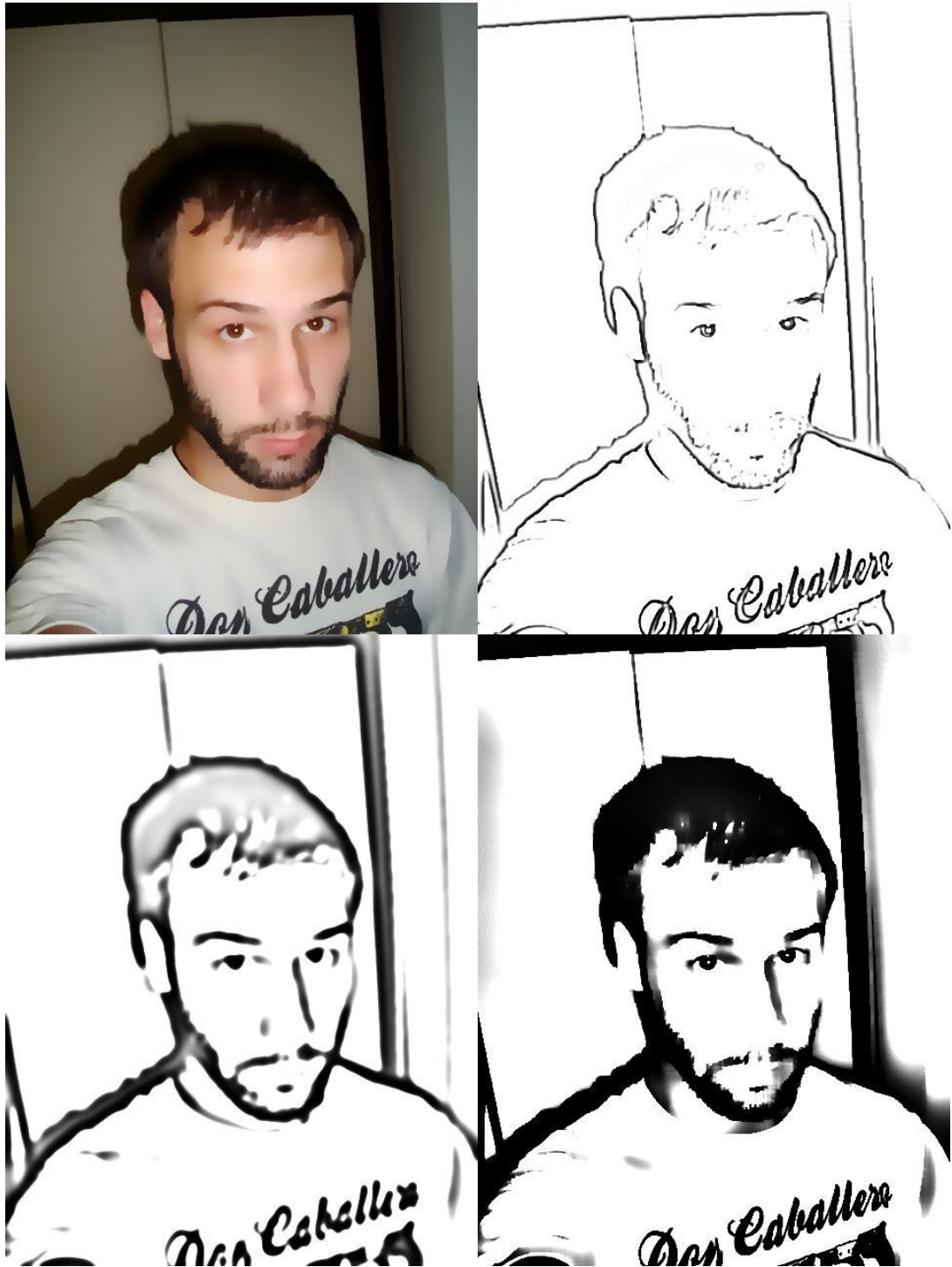
Vrijednost funkcije računa se po formuli:

$$D(x) = \begin{cases} 1, & \text{ako je } (gauss_{manji}(x) - \tau \cdot gauss_{veći}(x)) > 0 \\ 1 + \tanh(\varphi_e \cdot (gauss_{manji}(x) - \tau \cdot gauss_{veći}(x))), & \text{inače} \end{cases}$$

Gdje je:

- $gauss_{veći}(x)$ – slika filtrirana Gausovim filtrom većeg polumjera
- $gauss_{manji}(x)$ – slika filtrirana Gausovim filtrom manjeg polumjera
- τ – parametar koji određuje osjetljivost detekcije rubova. Za male vrijednosti detektira se manje šuma, ali i manje rubova. Najbolji rezultati postižu se za vrijednost 0.98
- φ_e – parametar koji određuje oštrinu prikazanih rubova.

Isprobavanjem različitih vrijednosti ovih parametara mogu se dobiti zanimljivi rezultati, kao što daje slika 4.5. Pažljivim odabirom parametara moguće je dobiti sliku u stilu skice nacrtane i osjenčane olovkom.



Slika 4.5 Rezultati detekcije rubova za razne vrijednosti parametara.

5. Programska implementacija

Svi navedeni postupci obrade slika implementirani su u programskom jeziku C#. Za otvaranje i kompresiju video zapisa, i za dohvaćanje svake sličice zapisa koristi se komponenta VideoCog, dok je sama obrada pojedine sličice izvedena osnovnim konstruktima jezika.

5.1. Komponenta VideoCog

Komponenta VideoCog dostupna je za preuzimanje u probnoj (engl. trial) verziji koja nudi punu funkcionalnost na 30 dana. Kad se komponenta postavi na formu, nudi mnoštvo zanimljivih mogućnosti od kojih su pri izradi rada korištene samo one za učitavanje video zapisa, dohvaćanje svake sličice zapisa i za kompresiju video zapisa.

Otvaranje video zapisa obavlja se jednostavnim upisivanjem puta do video zapisa u polje *Streams*. Na sličan način definira se i put do izlaznog video zapisa.

```
videoCogControl1.Streams[0].Source = inputFileNames;  
videoCogControl1.OutputFile = outputFileNames;
```

Prolazak kroz sve sličice video zapisa obavlja se u petlji koja se ponavlja onoliko puta koliko ima sličica u video zapisu i unutar koje se svaki put poziva metoda *StepForward(int FrameCount)*. Parametar *FrameCount* označava za koliko sličica se radi pomak unaprijed, prilikom poziva funkcije unutar petlje parametar je postavljen na vrijednost 1.

Kako bi se omogućila obrada sličica, potrebno je postaviti svojstvo *GrabberEnabled* na logičku vrijednost istinitosti.

```
videoCogControl1.Streams[0].GrabberEnabled = true;
```

Da bi se omogućilo spremanje sličica u novi video zapis, potrebno je postaviti svojstvo *Recording* na *true* i vratiti na *false* nakon prolaska kroz sve sličice video zapisa.

Najvažnija metoda koja se koristi je `FrameCallback`, koja se automatski poziva nakon što se dohvati sličica video zapisa.

```
private void FrameCallback(double SampleTime, System.IntPtr pBuffer, int BufferSize, int Width, int Height);
```

Parametri koje funkcija prima i koji se koriste za daljnju obradu su:

- *SampleTime* – vrijeme (u milisekundama) od početka video zapisa
- *pBuffer* – pokazivač na polje RGB vrijednosti (polje bajtova u BGR redoslijedu, prva vrijednost odgovara B, druga G, a treća R komponenti)
- *BufferSize* – veličina međuspremnika (u bajtovima)
- *Width* – širina slike (u pikselima)
- *Height* – visina slike (u pikselima)

To su svi podaci koji su potrebni za obradu slike, odnosno video zapisa. Kod manipulacije bajtovima slike treba imati na umu da je stvarna veličina polja u memoriji jednaka $3 * \text{širinaSlike} * \text{visinaSlike}$, zbog toga što je svaki slikovni element zapisan pomoću R, G i B komponente, odnosno tri bajta memorije.

5.2. Rad sa slikama u programskom jeziku C#

U programskom jeziku C# postoji mogućnost jednostavnog pretvaranja slike u bitmapu prilikom njenog učitavanja. Zbog jednostavnosti formata zapisa takvu sliku jednostavno je obrađivati. Iako postoje ugrađene funkcije za dohvaćanje i postavljanje vrijednosti slikovnog elementa na određenim koordinatama, zbog količine vremena potrebne da se slika obradi na takav način, te funkcije nisu prikladne za bilo kakvu ozbiljniju upotrebu. Puno veće brzine obrade postižu se korištenjem pokazivača. Iz zapisa bitmape moguće je dobiti pokazivač na polje RGB vrijednosti i direktno u memoriji mijenjati vrijednosti slikovnih elemenata. Blok koda u kojem se koriste pokazivači potrebno je okružiti ključnom riječi *unsafe*, također, u postavkama projekta potrebno je omogućiti prevođenje takvog koda.

Primjer implementacije najjednostavnije operacije nad slikom – inverzije:

```
unsafe public void inverzniFiltar( byte* pocetniPixelSlike, int sirinaSlike, int visinaSlike)
{
    byte* pixelSlike = pocetniPixelSlike;
```



```

for ( int pixelY = 0 ; pixelY < visinaSlike ; pixelY++ )
{
    for ( int pixelX = 0 ; pixelX < sirinaSlike ; pixelX++ )
    {
        pixelSlike[0] = (byte)( 255 - pixelSlike[0] );
        pixelSlike[1] = (byte)( 255 - pixelSlike[1] );
        pixelSlike[2] = (byte)( 255 - pixelSlike[2] );

        pixelSlike = pixelSlike + 3;
    }
}

```

PixelSlike je pokazivač na prvu komponentu boje prvog slikovnog elementa u zapisu slike. Komponente su zapisane u obrnutom (BGR) redoslijedu. Naredbom pixelSlike[0] pristupamo prvoj vrijednosti na koju pixelSlike pokazuje, a drugoj i trećoj vrijednosti pristupamo povećavanjem indeksa. Zadnjim retkom koda pomiče se pokazivač na zapis sljedećeg slikovnog elementa.

Svi implementirani filtri imaju jednaku osnovnu strukturu, a razlikuju se u linijama koje određuju novu vrijednost slikovnog elementa.

U nekim slučajevima potrebno je napraviti kopiju bitmape i alocirati memoriju za tu kopiju. U C#-u se blok memorije za bitmapu zadane visine i širine alocira ovako:

```
byte* pocetniPixel = (byte*)Marshal.AllocHGlobal( 3 * sirinaSlike * visinaSlike * sizeof( byte ) );
```

Na kraju korištenja, alociranu memoriju potrebno je i osloboditi:

```
Marshal.FreeHGlobal( (IntPtr)pocetniPixel );
```

5.3. Osnovne klase

U apstraktnoj klasi *Filter* koju nasljeđuju ostale konkretne klase, definirani su atributi zajednički svim implementiranim filtrima i pomoćne metode koje filtri koriste. Npr. metoda za kopiranje bitmape i metode za pretvaranje iz RGB u Lab model boja i obrnuto.

U konkretnoj klasi *OsnovniFiltri*, implementirani su filtri koji obavljaju operacije na slikovnim elementima – inverzni filter, pretvaranje u sive tonove, normalizacija slike i gamma korekcija.

Klasa *ProstorniFiltri* implementira prostorne operacije – Gaussov filtar, selektivni Gaussov filtar, *Difference of Gaussinas* filtar, Sobelov filtar i postupak kvantizacije boja.

U klasi *MultiplyBlendingMode* implementiran je postupak spajanja dva sloja slike u jednu sliku množenjem vrijednosti slikovnih elemenata.

6. Rezultati

Vremena trajanja pojedinih postupaka obrade slike, u ovisnosti o parametrima i dimenzijama slike, prikazuje Tablica 1. Vremena su izražena u sekundama. Oznake pojedinih parametara su:

- r – polumjer selektivnog Gaussovog filtra
- Δ – maksimalna razlika intenziteta
- N – broj kvantizacijskih razina
- r_v – veći polumjer DoG filtra
- r_m – manji polumjer DoG filtra

Tablica 1 Trajanje izvođenja pojedinih postupaka obrade.

		Dimenzije slike (u pikselima)		
		320x240	640x480	1280x768
Apstrakcija slike (4 iteracije)	$r = 5, \Delta = 10$	0.375	1.453	4.672
	$r = 10, \Delta = 10$	0.546	2.141	6.875
	$r = 10, \Delta = 5$	0.548	2.109	6.844
	$r = 20, \Delta = 10$	0.859	3.422	11.531
Kvantizacija	$N = 8$	0.406	1.547	5.125
	$N = 12$	0.391	1.549	5.140
Detekcija rubova	$r_v = 3$ $r_m = 1$	0.203	0.813	2.609
	$r_v = 5$ $r_m = 4$	0.234	0.937	3.016
	$r_v = 10$ $r_m = 8$	0.297	1.172	3.75
	$r_v = 100$ $r_m = 1$	0.797	3.391	11.078
Ukupno trajanje	$r = 10, \Delta = 10, N = 8,$ $r_v = 5, r_m = 4$	1.186	4.535	15.016

Vidljivo je da vrijeme trajanja pojedinih postupaka raste otprilike proporcionalno s dimenzijama slike. Trajanje postupka apstrakcije (mjereno je vrijeme potrebno za 4 prolaza selektivnog Gaussovog filtra) ovisi samo o polumjeru selektivnog Gaussovog filtra, promjena vrijednosti drugog parametra nema utjecaja na brzinu izvođena. Trajanje detekcije rubova ovisi o zadanim polumjerima DoG filtra, a vrijeme trajanja kvantizacije ne ovisi o broju kvantizacijskih razina.

7. Zaključak

Iako je područje nefotorealističnog prikaza relativno novo, našlo je primjenu u mnogim područjima, npr. filmskoj industriji, CAD alatima, računalnim igrama, itd.

Ovaj rad pokazuje kako se kombinacijom jednostavnih temeljnih postupaka može ostvariti zanimljiv efekt, koji će obrađenu sliku ili video zapis sigurno izdvojiti iz mora drugih.

Kako u radu [1], koji je poslužio kao osnova za ovaj rad nisu opisane implementacije pojedinih filtara, najveći dio vremena utrošen je na proučavanje raznih filtara kojima bi se postigao željeni efekt. Također, neki od postupaka obrade nisu detaljno opisani, primjerice postupak kvantizacije, koji je u konačnici ipak implementiran, i postupak pomicanja slikovnih elemenata prema rubovima, koji se zbog premalo napisanih informacija nije našao u finalnoj verziji. Unatoč tome, konačan rezultat gotovo je istovjetan rezultatima rada [1].

Mjesta za poboljšanja naravno ima. Trebalo bi se pozabaviti ubrzanjem postupka obrade kako bi se trajanje obrade video zapisa većih formata svelo na prihvatljivu razinu.

8. Literatura

1. Holger Winnemöler, Sven C. Olsen, Bruce Gooch: „Real-Time Video Abstraction“, 2006., <http://videoabstraction.net/>
2. „Lab color space“, Wikimedia Foundation, 14.6.2009., <http://en.wikipedia.org/wiki/Lab_color_space>
3. Steven Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, San Diego, California, 1999.
4. „Gaussian blur“, Wikimedia Foundation, 7.6.2009., <http://en.wikipedia.org/wiki/Gaussian_blur>
5. Neven Elezović, Normalna razdioba, *Slučajne varijable*, Element, Zagreb, 2007., 39-49.
6. „Selective Gaussian Blur“, <<http://docs.gimp.org/2.6/en/plugin-sel-gauss.html>>

Sažetak

Nefotorealistična tehnika prikaza video zapisa

Ovaj rad opisuje postupke potrebne za ostvarivanje nefotorealistične tehnike prikaza video zapisa u stilu crtanog filma. U prvom poglavlju dan je popis temeljnih postupaka koje potrebno ostvariti. U drugom poglavlju opisuju se modeli boja i BMP slikovni format. U trećem poglavlju objašnjeni su postupci filtriranja slike i opisani su filtri koje je potrebno implementirati. U četvrtom poglavlju opisuju se temeljni postupci i njihova implementacija korištenjem opisanih filtara. U petom poglavlju u kratko su objašnjeni osnovni principi praktične implementacije navedenih postupaka.

Ključne riječi: apstrakcija, detekcija rubova, filtriranje slike, kvantizacija, nefotorealistični prikaz (NPR).

Abstract

Non-photorealistic video rendering

This paper describes basic methods that are required for non-photorealistic video rendering with cartoon-like effects. Basic methods are listed in the first chapter. Second chapter describes different color models and BMP image format. Image filtering methods and filters required to produce cartoon-like effects are explained in chapter three. Fourth chapter describes implementation of basic methods using filtering methods. And brief explanation of practical implementation of mentioned methods is given in chapter five.

Keywords: abstraction, edge detection, image filtering, non-photorealistic rendering (NPR), quantization.

Slike

Slika 0.1 Usporedba obične slike i slike obrađene jednom od tehnika nefotorealističnog prikaza.	1
Slika 1.1 Dijagram toka obrade slike.	3
Slika 2.1 Dobivanje boja u RGB modelu.	5
Slika 2.2 Prikaz $L^*a^*b^*$ modela boja.	6
Slika 3.1 Spajanje dva sloja slike u jednu sliku.	10
Slika 3.2 Ilustracija konvolucije.	12
Slika 3.3 2D Gaussova funkcija.	13
Slika 3.4 1D Gaussove funkcije.	14
Slika 3.5 Originalna slika i slika filtrirana Gausovim filtrom (polumjer 10).	15
Slika 3.6 Područje selekcije selektivnog Gaussovog filtra.	16
Slika 3.7 Originalna slika i slika filtrirana selektivnim Gausovim filtrom.	16
Slika 3.8 Originalna slika i slika s detektiranim rubovima.	17
Slika 3.9 Originalna slika i slika dobivena filtriranjem Sobelovim filtrom.	18
Slika 4.1 Apstrakcija slike.	22
Slika 4.2 Prikaz razina kvantizacije.	23
Slika 4.3 Kvantizacija svjetline.	24
Slika 4.4 DoG funkcija i zavisnost vrijednosti ruba o funkciji.	24
Slika 4.5 Rezultati detekcije rubova za razne vrijednosti parametara.	26

Tablice

Tablica 1 Trajanje izvođenja pojedinih postupaka obrade.....	31
--	----