Analiza tekstura u slici

Ronta, Hrvoje

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:747449

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

ANALIZA TEKSTURA U SLICI

Završni rad

Hrvoje Ronta

Osijek, 2016.

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. OSNOVNI POJMOVI 2
2.1. Važnost percepcije tekstura iz perspektive psihofizike i računalnog vida 4
2.2. Problemi analize teksture 6
2.2.1. Segmentacija teksture 6
2.2.2. Klasifikacija teksture 7
2.2.3. Sinteza teksture 8
2.3.4. Dobivanje oblika iz teksture 9
3. MATEMATIČKI MODEL 11
3.1. Strukturne metode 12
3.2. Statističke metode 13
3.2.1. Statističke metode prvog reda 13
3.2.2. Statističke metode drugog reda 15
3.3. Metode temeljene na modelu 17
3.4. Transformacijske metode 18
4. EKSPERIMENTALNI DIO 20
4.1. Segmentacija tekstura u slici metodom Gaborovih filtera 20
4.1.1. Učitavanje i prikaz slike 22
4.1.2. Dizajniranje reda Gaborovih filtera 23
4.1.3. Naknadna obrada slika Gaborovih magnituda u Gaborove značajke 25
4.1.4. Klasterizacija ili grupiranje algoritmom *k*-srednjih vrijednosti 28
4.1.5. Prikaz rezultata segmentacije 29
4.2. Klasifikacija tekstura u slici implementacijom algoritma za fraktalnu analizu tekstura 32
5. ZAKLJUČAK 37
LITERATURA 38
POPIS SLIKA 40
SAŽETAK 42
ŽIVOTOPIS 43
PRILOZI 44

1. UVOD

Iako ne postoji stroga definicija tekstura one su čovjeku lako uočljive, intuitivno ih razumijemo te predstavljaju bogat izvor vizualnih informacija o prirodi i trodimenzionalnom obliku promatranog objekta. Teksture su važan sastojak mnogih slika i koriste se za analizu u raznim područjima ljudskog djelovanja, primjerice pri snimanju iz zraka, za sustav automatizirane provjere, procesiranje dokumenata ili u biomedicini.

U samim početcima znanstvenih proučavanja fokus je bio na određivanju korisnih svojstava ili značajki kao što su finoća, oštrina, kontrast, uniformnost, boje, gustoća, granulacija, faza, frekvencija, orijentacija i regularnost teksture. Kasnija istraživanja su dala dublji uvid i razumijevanje kroz uporabu matematičkih modela. Najvažnija četiri problema kod obrade i analize tekstura svode se: na problem klasifikacije pri čemu je potrebno nepoznatu teksturu svrstati u jednu od N poznatih klasa; na problem sinteze; na problem diskriminacije tj. određivanja granica za sliku koja sadrži više područja raznih tekstura (rezultira segmentacijom tekstura) i na problem dobivanja oblika iz teksture odnosno rekonstrukciju 3D geometrijske podloge. Unatoč široj primjeni fundamentalnih matematičkih modela važno je napomenuti da se znanstveno područje koje se bavi računalnim vidom te digitalnom obradom i analizom slike još uvijek razvija kako bi se postiglo točnije modeliranje, klasifikacija i segmentacija teksture.

Cilj ovog rada je dati osnovni uvid u problematiku analize teksture u slici te riješiti praktični zadatak koji se sastoji od eksperimentalnih postupaka segmentacije i klasifikacije. Kod segmentacije analizirano je nekoliko slika kako bi se na njima odredile regije s različitim teksturama, dok je kod klasifikacije cilj bio implementirati algoritam koji bi omogućio točno svrstavanje ulazne slike s teksturama u jednu od poznatih klasa koja sadrži istovjetnu teksturu.

Nakon prvog uvodnog poglavlja, u drugom poglavlju ukratko su opisane teorijske osnove i spomenuta četiri problema. U trećem poglavlju opisane se metode za kvantitativni opis teksture te definirani neki od poznatijih matematičkih modela. U četvrtom poglavlju detaljno su opisani provedeni eksperimenti.

1.1. Zadatak završnog rada

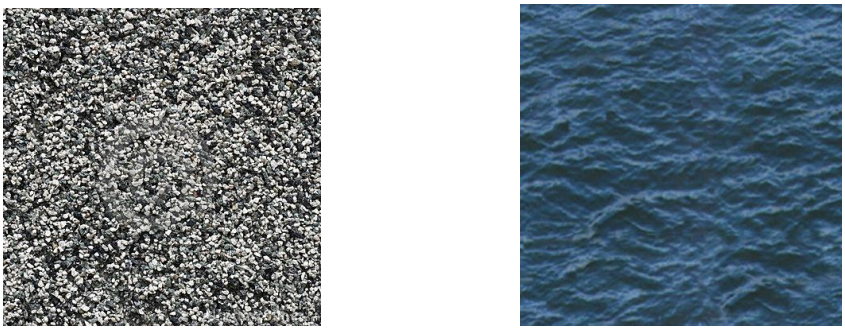
Zadatak ovog završnog rada temelji se na provođenju eksperimenata analize tekstura u slici odnosno opisu postupaka segmentacije i klasifikacije te dobivenih rezultata. U tu svrhu korišten je MATLAB programski paket.

2. OSNOVNI POJMOVI

U mnogim algoritmima za računalni vid i procesiranje slike koriste se pojednostavljene pretpostavke o uniformnosti intenziteta lokalnih regija slike. Unatoč tome slike objekata koje nalazimo u stvarnosti često ne posjeduju regije uniformnog intenziteta npr. slika drvene površine nije uniformna ali sadrži određene ponavljajuće uzorke. Takav uzorak može biti posljedica svojstava površine kao što je hrapavost ili orijentacija, a može biti i razlika u refleksiji zbog boja na površini. Može se ustvrditi da iako lako i intuitivno prepoznajemo teksture kada ih vidimo vrlo teško ih je definirati. To potvrđuje i niz raznih definicija u stručnoj literaturi. Većina predloženih definicija ističe samo neko svojstvo teksture koje je zanimljivo s obzirom na metodu analize slike za koju se takva definicija koristi. Neke od definicija su:

- Tekstura je regija slike u kojoj je skup lokalnih statističkih ili drugih lokalnih svojstava slike konstantan, sporo-promjenjivi ili približno periodičan. [1]
- Tekstura je svojstvo koje opisuje prostorno uređenje nivoa sivog piksela u jednoj regiji. [1]
- Pojam teksture se odnosi na ponavljanje osnovnih elemenata teksture, teksela. Jedan teksel sadrži više piksela čiji raspored može biti periodični, aperiodični ili slučajan. Prirodne teksture se odlikuju slučajnim rasporedom dok je za umjetne više karakterističan periodični raspored piksela. Tekstura može biti gruba, fina, glatka, zrnasta, neravnomjerna, regularna, neregularna ili linearna. [1]

Za definiciju teksture bitno je istaknuti da ona ima sljedeće značajke: postoji osnovni dio *teksel* koji se ponavlja unutar veće cjeline, *teksel* se sastoji od ne slučajno organiziranih elemenata te je konstantne veličine koja se ne mijenja unutar cjeline.

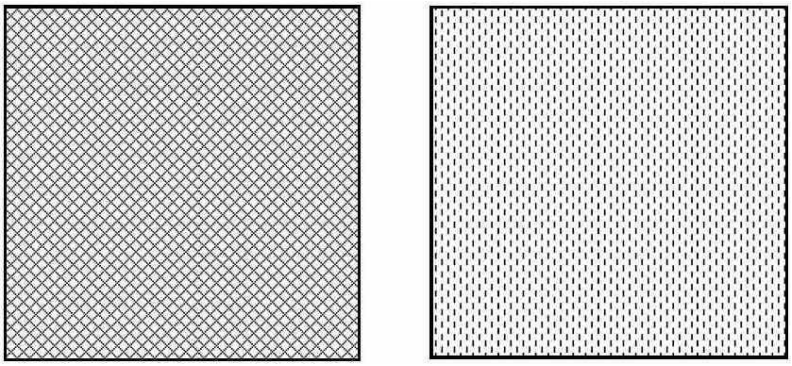


(a) (b)

Sl. 2.1. *Primjer tekstura dobivenih iz stvarnih objekata. (a) šljunak (b) voda*

Teksture je moguće svrstati na jednodimenzionalne (jedan red *teksela*), dvodimenzionalne (tj. slike) i trodimenzionalne volumne teksture. Dvodimenzionalne teksture služe za opis vanjskog izgleda površine objekta dok se trodimenzionalnim može opisati i izgled unutrašnjosti objekta. S

obzirom na osnovne regije definirane svjetlinom (engl. *gray level primitives*) postoje mikroteksture i makroteksture. Kada su osnovne regije male i prostorna interakcija je mala onda se rezultantna tekstura zove mikrotekstura, a kada osnovne regije imaju određeni oblik i organizaciju tekstura se zove makrotekstura. Primjer mikroteksture je kada se nezavisni Gaussov šum doda vrijednostima točaka u nekom području slike koja ima približno konstantnu svjetlinu dok se kod makroteksture slici dodaje Gaussov šum koji je koreliran. Teksture je moguće podijeliti i na umjetne i prirodne. Umjetne se sastoje od točaka, krugova, linija itd. postavljene na neutralnu podlogu. Prirodne teksture su slike prirodnih scena koje sadrže uzorke koji se ponavljaju tipa zid od cigle, vuna, trava, pijesak, voda i slično.



Sl. 2.2. *Primjer umjetnih tekstura*.

Teksture se također mogu urediti u spektar od regularnih do stohastičkih kao što se može vidjeti na slici 2.3. (a) regularne, (b) bliske regularnim, (c) iregularne, (d) bliske stohastičkim, (e) stohastičke. Stohastičke teksture izgledaju kao šum odnosno obojene točkice koje su nasumično raspoređene na slici, slabo specificiranih značajki. Mnoge teksture tako izgledaju ako se gledaju iz udaljenosti.



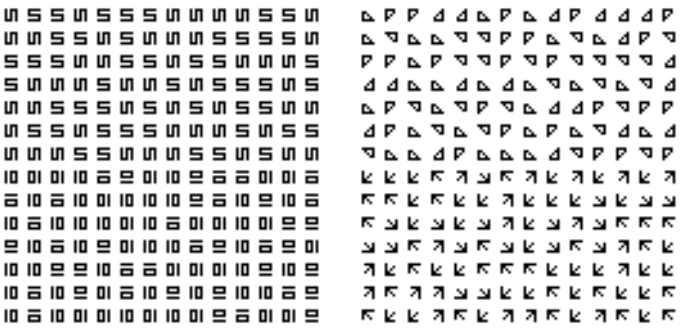
(a) (b) (c) (d) (e)

Sl. 2.3. *Spektar tekstura.*

2.1. Važnost percepcije tekstura iz perspektive psihofizike i računalnog vida

Analiza tekstura važno je i korisno područje istraživanja računalnog vida. Većina prirodnih površina posjeduje teksture pa stoga uspješan sustav vida mora biti sposoban obraditi svijet tekstura koji ga okružuje. Važan aspekt u razumijevanju percepcije tekstura nalazi se upravo iz perspektive čovjekova vida odnosno psihofizike, graničnog područja psihologije, fizike i fiziologije koje istražuje odnos između fizičkih podražaja i psihičkih doživljaja.

Uočavanje tigra skrivenog u gusto lišće zadatak je naše percepcije koji nosi životne posljedice ako pokušavamo preživjeti u šumi. Uspjeh tigra u kamuflaži s druge strane neuspjeh je našeg sustava vida u promatranju tj. u nemogućnosti odvajanja figure od pozadine. Problematika separacije figure od pozadine predstavlja značajan interes proučavanja za psihofizičare. Ta separacija se može temeljiti na raznim znakovima kao što su svjetlina, boja, tekstura i slično. U primjeru tigra u šumi, tekstura igra veliku ulogu. Kamuflaža je uspješna zbog sustava vida promatrača koji ne može diskriminirati (segmentirati) dvije teksture, lišće i tigrovu kožu. Upravo je pitanje koji vizualni procesi dopuštaju separaciju figure od pozadine s obzirom na teksturu osnovna motivacija među psiholozima za proučavanje percepcije tekstura. [1]



(a) (b)

Sl. 2.4. *Parovi tekstura jednake statistike drugog reda.*

Drugi razlog zbog kojega je proučavanje psihofizike percepcije tekstura važno nalazi se u evaluaciji performansi raznih algoritama tekstura u odnosu na čovjekov sustav vida koji ima isti zadatak. Primjer nam je slika 2.4. s parovima tekstura koje je prvi opisao Bela Julesz. Oba primjera na slici 2.4. (a) i (b) sastoje se od dvije regije koje sadrže drugačije teksture. To će biti jasno svakom promatraču nakon pažljivog ispitivanja no ako samo letimično pogledamo tada naša percepcija slike neće rezultirati dvama različitim regijama teksture nego jednom uniformnom regijom. Julesz govori da takav par tekstura nije moguće diskriminirati bez napora jer mozak u ovom slučaju ne može nesvjesno akumulirati točne informacije iz okoliša. Ovakve umjetne teksture nam pomažu u formiranju hipoteze o tome koja su svojstva slike važna u

čovjekovoj percepciji ali i pitanju kako procijeniti performanse računalnog algoritma u analizi teksture. Julesz je opsežno proučavao percepciju diskriminacije tekstura tj. kada je par tekstura moguće diskriminirati u slučaju da imaju jednaku svjetlinu, kontrast i boju. Usredotočio se na prostornu statistiku sivih nivoa slike zadržavajući ostala svojstva osvjetljenja nepromijenjenim. Za razumijevanje Juleszovog temeljnog rada potrebno je definirati koncepte prostorne statistike prvog i drugog reda:

I. Statistika prvog reda mjeri vjerojatnost uočavanja sive vrijednosti na slučajno odabranoj lokaciji slike. Statistika prvog reda može se izračunati iz histograma (grafički prikaz distribucije numeričkih vrijednosti odnosno nivoa sivog) za intenzitet piksela na slici. Oni ovise o pojedinačnoj vrijednosti piksela, a ne o interakciji ili združenom pojavljivanju susjednih vrijednosti piksela. [1]
II. Statistika drugog reda definirana je kao vjerojatnost uočavanja para sivih vrijednosti koji se pojavljuje na krajnjim točkama dipola slučajne duljine smještenih na slici na slučajnoj lokaciji i orijentaciji. To su svojstva parova vrijednosti piksela. [1]

Julesz je tako došao do pretpostavke da je dvije teksture teško diskriminirati ako su im statistike drugog reda identične. Kako bi objasnio ovaj fenomen predložio je "teoriju *tekstona*". *Tekstoni* su vizualni događaji (kao kolinearnost, završetci, zatvaranje i sl.) čija se prisutnost detektira i koristi u diskriminaciji tekstura. Završetci su krajnje točke linijskih segmenata i kuteva. U primjeru (a) na slici 2.4. broj završetaka je jednak (informacija o *tekstonima* se u obje regije ne razlikuje) što dokazuje da čovjekov sustav vida ne može diskriminirati takve teksture. U primjeru (b) pak imamo slučaj gdje je broj završetaka u gornjoj regiji tri, a u donjoj četiri. Zbog razlike u *tekstonima* moguće je diskriminirati regije.

Studije na području psihofiziologije, discipline koja podrazumijeva istraživanje fizioloških temelja psihičkih procesa, sugeriraju da mozak obavlja višekanalne, frekvencijske i orijentacijske analize formirane na mrežnici oka. Psihofizički eksperimenti na raznim uzorcima sugeriraju da sustav vida razlaže sliku na filtrirane slike različitih frekvencija i orijentacija. Eksperiment na mozgu makaki majmuna, za kojeg se pretpostavlja da ima vizualno procesiranje slično čovjeku, zabilježio je da je odziv jednostavnih stanica u vidnoj kori mozga sinusoidna rešetka različitih frekvencija i orijentacija. Zaključak je da su stanice prilagođene uskim frekvencijskim i orijentacijskim pojasevima. Navedene studije motivirale su istraživače na području vida da koriste višekanalno filtriranje u analizi tekstura.

2.2. Problemi analize teksture

Pri analizi teksture u praksi postoje četiri osnovna problema. Kao sto je spomenuto u uvodu radi se o problemima segmentacije, klasifikacije, sinteze i dobivanju oblika iz tekstura.

2.2.1. Segmentacija teksture

Prema opisnoj kvalitativnoj definiciji dobre segmentacije kaže se da regije dobivene segmentacijom moraju biti uniformne i homogene u odnosu na neke karakteristike, kao što su nivoi sivog ili teksture. Unutrašnjost regija mora biti jednostavna i bez malih rupa, a granice regija jednostavne, glatke i prostorno točne. Susjedne regije moraju biti dovoljno različite prema značajkama po kojima su uniformne.

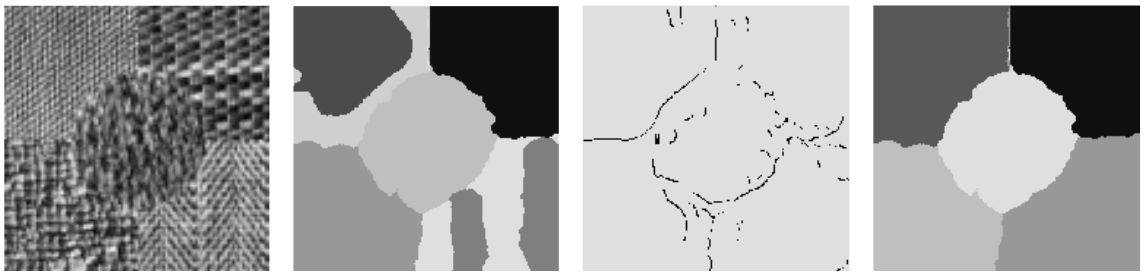
Segmentacija teksture je težak i kompleksan problem, a razlozi su u tome što se ne može *a priori* odrediti kakav tip teksture se nalazi na slici, koliko različitih tekstura ima i kakve teksture posjeduju određene regije slike. Kako bi se provela segmentacija nije niti potrebno znati odrediti specifične teksture na slici već samo pronaći način pomoću kojega bismo znali da se dvije teksture razlikuju, obično u susjednim regijama slike. Dva opća pristupa u izvođenju segmentacije tekstura analogna su metodama segmentacije slike i odnose se na pristup zasnovan na regijama i pristup zasnovan na granicama. Svaki od pristupa ima razvijene brojne metode za segmentaciju. [2]

U pristupu zasnovanom na regijama pokušavaju se odrediti regije slike koje imaju uniformnu teksturu. Pikseli ili male lokalne regije spajaju se s obzirom na sličnost značajki teksture. Nakon toga nastaju uniformne regije koje imaju različite teksture i smatraju se segmentiranim regijama. Prednost ove metode je u tome što su granice regija uvijek zatvorene pa su regije s drugačijim teksturama uvijek dobro odvojene. Nedostatak je što u slučaju segmentacije slika s mnogo regija moramo unaprijed specificirati broj različitih tekstura prisutnih na slici.

Pristup koji se zasniva na granicama nastoji detektirati razlike u teksturama među susjednim regijama. Na mjestima na kojima postoje razlike u teksturi detektiraju se granice. U ovom pristupu nije potrebno unaprijed znati broj regija teksture na slici. Nedostatak je u tome što granice mogu imati praznine i biti isprekidane, a zbog toga dvije regije s različitim teksturama nisu identificirane kao odvojene zatvorene regije. Može se zaključiti da granične metode rezultiraju segmentacijom samo kada sve detektirane granice formiraju zatvorenu krivulju.

U većini slučajeva integracija rezultata dobivenih segmentacijom s obzirom na granice i onoj s obzirom na regije može poboljšati ukupni rezultat segmentacije. To se može vidjeti na slici 2.5. koja upravo prikazuje rezultate takve integracije primjenom metode Gaborovog filtriranja. Na

slici 2.5. (a) imamo originalnu sliku koja se sastoji od pet prirodnih tekstura, pod (b) se nalaze rezultati segmentacija s obzirom na regije, pod (c) procesiranje zasnovano na rubovima i detektirane rubove tekstura, pod (d) nova segmentacija dobivena kombiniranjem rezultata zasnovanih na regijama i onih zasnovanih na rubovima.



(a) (b) (c) (d)

Sl. 2.5. *Segmentacija*.

2.2.2. Klasifikacija teksture

Pojam klasifikacije se odnosi na pridruživanje fizičkog objekta ili slučaja u jednu od nekoliko predefiniranih kategorija. Problem klasifikacije tekstura nalazi se u odlučivanju kojoj od poznatih klasa tekstura pripada promatrani uzorak. Kako bi se to postiglo potrebno je *a priori* znanje o klasama koje treba prepoznati. Kada je to znanje dostupno, a značajke teksture određene tada se koriste metode klasične klasifikacije uzorka kako bi se obavila klasifikacija.

Klasifikacija teksture je proces koji uključuje dvije faze; fazu učenja i fazu raspoznavanja. U fazi učenja cilj je izraditi model za sadržaj tekstura iz svih klasa tekstura, prisutnih u *trening*-podatcima (engl. *training data*). *Trening*-podatci uobičajeno sadrže *trening*-slike tekstura s poznatim oznakama njihovih klasa. Sadržaj tekstura *trening*-slika dobiva se pomoću odabrane metode za analizu tekstura i donosi skup značajki tekstura za svaku *trening*-sliku. Te značajke, koje mogu biti skalari, diskretni histogrami ili empirijske distribucije, opisuju dobivene značajke teksture kao što su prostorna struktura, kontrast, hrapavost, orijentacija itd. U fazi raspoznavanja sadržaj tekstura nepoznatog uzorka se prvo opisuje s istom metodom za analizu. Nakon toga se značajke tekstura uzorka uspoređuju s onim od *trening*-slika koristeći klasifikacijski algoritam ili klasifikator, a uzorak se pridružuje onoj kategoriji s kojom se najbolje podudara. Ako najbolja odabrana kombinacija nije dovoljno dobra prema nekim predefiniranim kriterijima tada se po želji nepoznati uzorak može odbaciti. [3]



Sl. 2.6. *Primjer klasifikacije uzorka teksture trave.*

Na slici 2.6. može se vidjeti primjer klasifikacije teksture trave u odgovarajuću klasu tekstura od šest mogućih koje su poslužile kao uzorak za treniranje klasifikatora, a to su od vrha prema dnu klase tekstura lišće, drvo, trava, folija, baršun i slama. Kvačica simbolizira točan rezultat klasifikacije jer je ulazna slika stavljena u točnu klasu. Praktični primjer gdje se koristi klasifikacija tekstura kao pogodna metoda procesiranja je klasifikacija regija u satelitskim slikama u kategorije uporabe zemljišta, u medicini za dijagnostiku, a primijenjena je i za automatiziranu kontrolu boja. Kategorije su bile ocijene kvalitete boje koje su davali stručnjaci i korištene su kao uzorak za treniranje nadzirane klasifikacije boja.

2.2.3. Sinteza teksture

Sinteza je kompleksna cjelina sastavljena od ujedinjenih elemenata. Problem sinteze tekstura se uglavnom susreće u radu s računalnom grafikom kod renderiranja i predstavlja alternativni način dobivanja tekstura. Može se definirati kao problem gdje je potrebno sintetizirati novu teksturu s obzirom na zadani uzorak teksture, koja bi kada bi je gledao čovjek izgledala kao da je generirana od istog temeljnog postupka. Veličina dobivene teksture određuje se po želji ali ona mora biti što je moguće sličnija ulaznim vrijednostima, ne smije imati vidljive artefakte (šavove, šupljine, krive rubove) i ne smije sadržavati ponavljanje. Osim sinteze dvodimenzionalnih tekstura iz dvodimenzionalnih uzoraka također je moguće sintetizirati volumne teksture iz dvodimenzionalnih uzoraka pri čemu dobivena tekstura također mora biti slična uzorku na svakom dvodimenzionalnom presjeku teksture.



(a) (b)

Sl. 2.7. *Prikaz sinteze*.

Na slici 2.7. (a) može se vidjeti prikaz sinteze dvodimenzionalne teksture iz dvodimenzionalnog uzorka, (b) prikaz sinteze volumne teksture iz dvodimenzionalnog uzorka. Za postupak sinteze važni su pojmovi susjedstva i udaljenosti između dva susjedstva. Susjedstvo je skup slikovnih elemenata, piksela koji su smješteni u kvadratnom prozoru određene veličine. Na susjedstvo se može gledati kao na općenito susjedstvo ili kao na susjedstvo nekog piksela pri čemu se upravo taj piksel nalazi u središtu prozora. Pri određivanju udaljenosti često se traži najbliže susjedstvo i to primjenom udaljenosti dva vektora. Neki matematički modeli koji se koriste za sintezu su model slučajnih Markovljevih polja, model fraktala i mozaik modeli. [4]

2.3.4. Dobivanje oblika iz teksture

Dobivanje oblika iz teksture je tehnika računalnog vida kojom se trodimenzionalni objekt rekonstruira iz dvodimenzionalne slike. Iako je čovjekova percepcija pomoću tekstura u slici sposobna uočiti uzorke, procijeniti dubinu i prepoznati objekte na slici, napraviti sustav koji bi bio sposoban imitirati takvo ponašanje je vrlo zahtjevan problem.

Općenito govoreći na slikama se nalaze brojni tragovi koji promatraču omogućuju uočavanje trodimenzionalnih oblika objekata i površina prisutnih na slici. Primjer takvih tragova su varijacije sjenčanja na površini objekata ili konfiguracija granica i tipovi čvorišta koji dopuštaju uočavanje trodimenzionalnih oblika iz linijskih crteža objekata. Na slici 2.8. može se vidjeti kako varijacija u veličini, obliku i gustoći elemenata teksture daje dojam trodimenzionalnosti.



Sl. 2.8. *Dojam trodimenzionalnosti.*

Prva osoba koja je predložila kako se oblik može percipirati iz teksture bio je američki psiholog James Jerome Gibson u svojem djelu iz 1950. godine pod nazivom *"Percepcija vizualnog svijeta"*. Gibson je upotrijebio pojam gradijenta teksture kako bi naznačio da područja površine koja imaju slične teksture sa susjednim područjima promatrač percipira drugačije zbog razlike u orijentaciji površina i udaljenosti od promatrača. Gibson je upotrijebio pojam gradijenta jer je želio sugerirati diferencijalni proces. Slična istraživanja na polju dobivanja oblika iz teksture nastavljena su u dva glavna pravca, mjerenju i uporabi gradijenta teksture te statističkom zaključivanju temeljenom na probabilističkom modelu tekstura. Istraživači su formalizirali pojam gradijenta teksture kao pronalaženje gradijenta određene skalarne funkcije kao što je gustoća, skraćenje, kompozicija ili skaliranje. Matematički odnos između tih gradijenata i geometrije scene razvijen je i za ravne i za zakrivljene površine. Poteškoće u korištenju ovog pristupa nalaze se u tome što jednostavne distorzije gradijenata ne sadrže dovoljno informacija za određivanje mjere čitave površine krivulje i u tome što u mnogim situacijama nije jasno kako bi se ti gradijenti mogli izmjeriti na slici. Kod drugog važnog pristupa koji se temelji na probabilističkom modelu tekstura za problem određivanja oblika površine ili orijentacije koristi se statističko zaključivanje i estimacija parametara na slici pomoću Bayesove statistike ili maksimalne vjerojatnosti. Nedostatak ovog pristupa je u tome što je nemoguće odrediti usmjerene teksture poput trave ili tkanina i u tome što se dostupne informacije koriste tek djelomično. [5]

3. MATEMATIČKI MODEL

Na sam izgled teksture puno veći utjecaj ima prostorna organizacija i raspored piksela ili grupa piksela nego njihov intenzitet pa se za dobivanje korisnih vrijednosti teksture formira skup lokalnih obilježja slike. Lokalna obilježja slike odnose se na bilo kakvu karakteristiku koja se izračunava na malenom dijelu slike. Ovisno o tome koja lokalna obilježja se izračunavaju metode za kvantitativni opis teksture mogu se kategorizirati na:

- strukturnu metodu
- statističku metodu
- metodu temeljenu na modelu
- metodu procesiranja signala ili transformacijsku

Navedene metode se međusobno ne isključuju pa se u praksi za analizu teksture često koriste u kombinaciji.

Strukturni pristup predstavlja teksture prema jasno definiranim primitivnim elementima (mikroteksture) i hijerarhiji prostornog uređenja (makroteksture) tih elemenata. Kako bi se opisala tekstura potrebno je definirati mikroteksture i pravila po kojima su raspoređene. Prednost strukturnog pristupa je u tome što pruža dobar simbolički opis slike iako je to svojstvo korisnije za zadatak sinteze. Apstraktni opis nije pogodan za prirodne teksture zbog velike varijabilnosti mikrostrukture i makrostrukture i teško uočljive razlike između njih.

Kao kontrast strukturnoj metodi, statistički pristup se ne bavi hijerarhijskom strukturom. Umjesto toga indirektno predstavlja teksturu prema ne deterministčkim značajkama koje određuju distribucije i odnose između sivih nivoa slike. S obzirom na broj piksela koji definiraju lokalno obilježje statističke metode se mogu klasificirati na statistiku prvog reda (jedan piksel), drugog reda (dva piksela) i višeg reda (tri i više piksela). Osnovna razlika je u tome što se kod statistike prvog reda estimiraju značajke (npr. srednja vrijednost ili varijanca) pojedinačnih vrijednosti piksela pri čemu se ignorira prostorni međuodnos piksela. Kod statistike drugog i višeg reda estimiraju se značajke dvije i više vrijednosti piksela koji se u međuodnosu pojavljuju na specifičnoj lokaciji. Istraživanjima na području diskriminacije tekstura došlo se do saznanja o važnosti statističke metode drugog reda. Najpoznatija statistička metoda drugog reda za analizu tekstura koja se temelji na statistici parova piksela nastala je prema takozvanoj *co-occurrence* matrici (matrica združenog pojavljivanja). Drugi poznati pristupi su pomoću autokorelacijske funkcije i statistike pojavljivanja niza piksela (engl. *Gray Level Run Length*) no kod primjene nisu zabilježene dobre performanse. Pristup koji se temelji na višedimenzionalnim matricama

združenog pojavljivanja pokazao je bolje rezultate od *valićne* (engl. *wavelet*) transformacije pri klasifikaciji tekstura.

Analiza koja se temelji na modelu koristi fraktale i stohastičke modele s ciljem interpretacije teksture u slici. Procijenjeni parametri modela služe kao lokalna obilježja teksture. Modeli teksture mogu se podijeliti na deskriptivne i generativne (autoregresivni, engl. *moving average*, engl. *time series* modeli u 2D, Markovljevi i mozaik modeli). U prvu grupu spadaju modeli izvedeni na osnovu statističkih principa, a u drugu modeli koji predstavljaju teksturu opisom modela i skupom slučajnih brojeva. Generativni modeli su zanimljivi jer omogućuju sintezu tekstura. Model fraktala se pokazao korisnim u računalnoj grafici za predstavljanje površina s teksturom i za modeliranje nekih prirodnih tekstura ali ima ograničen domet. Osobine prirodnih tekstura i neizbježan šum najčešća su prepreka za pouzdano određivanje parametara fraktalnog modela. Također je ustanovljeno da vizualno potpuno različite teksture mogu imati istu fraktalnu dimenziju koja predstavlja osnovni deskriptor teksture.

Metode procesiranja signala ili transformacijske metode uglavnom koriste Fourierovu, *valićnu* i Gaborovu transformaciju. One predstavljaju sliku u prostoru čiji koordinatni sustav ima interpretaciju povezanu s karakteristikama teksture (npr. frekvencija). Fourierova transformacija pokazuje nedostatke u prostornoj lokalizaciji gdje su korisniji Gaborovi filtri. Njihova je pak uporaba ograničena u praksi jer ne postoji jedinstvena rezolucija filtra na kojoj se mogu lokalizirati prostorne strukture u prirodnoj teksturi. U usporedbi s Gaborovom transformacijom, *valićna* transformacija ima nekoliko prednosti. Variranjem prostornih rezolucija omogućena joj je reprezentacija tekstura u najprikladnijem mjerilu. Također postoji širok raspon u izboru *valićnih* funkcija tako da se lako može odabrati najprikladnija u specifičnoj aplikaciji. *Valićna* transformacija je pogodna za korištenje kod segmentacije tekstura, a nedostatak je u tome što nije translacijski invarijantna.

3.1. Strukturne metode

Strukturne metode za opis teksture zasnivaju se na pretpostavci da se tekstura sastoji od osnovnih elemenata ili primitiva, koji se ponavljaju na gotovo regularan način. U strukturnim metodama cilj je izdvojiti primitivne elemente koji čine teksturu i opisati vezu između njih. Pri karakterizaciji teksture kod ovog pristupa često se vrši podjela na slabe i jake teksture. Slabe teksture imaju slabe prostorne interakcije primitivnih elemenata i da bi se razlikovale dovoljno je odrediti koliko često se primitive pojavljuju u nekoj lokalnoj regiji. Jake teksture pak imaju ne slučajne prostorne interakcije primitivnih elemenata i za njihovu karakterizaciju potrebno je za

svaki par primitivnih elemenata odrediti koliko često se taj par pojavljuje u određenoj prostornoj vezi.

Osnovni element teksture ili primitiv je skup piksela okarakteriziran popisom značajki. Najjednostavniji primitiv je piksel sa značajkom nivoa sivog ali najčešće se za primitiv uzima maksimalno povezani skup piksela koji imaju neku posebnu značajku. Kao značajke mogu se koristiti i mjere oblika povezanih regija ili mjere homogenosti pri čemu se mora izvršiti segmentacija slike prije procesa karakterizacije. Za izdvajanje primitiva iz slabih tekstura postoji više metoda. Najčešće se kao značajka koristi zajednički nivo sivog svih piksela koji pripadaju primitivu, a koriste se i gustoća ruba, statistika pojavljivanja niza piksela. gustoća ekstrema, relacijska stabla i tako dalje. Za izdvajanje primitiva iz jakih tekstura koriste se statistike združenog pojavljivanja primitiva odnosno matrice združenog pojavljivanja za piksele ili grupu piksela.

Kada se iz slike izdvoje primitivni elementi dostupan je popis primitiva, koordinate njihovog središta i njihove značajke. Nekad su dostupni i podatci o obliku i susjedstvu primitiva. Na temelju tih podataka uspostavljaju se prostorne veze kao što su susjedstvo, blizina i brojnost primitiva koja pokazuje koliko se primitiva neke vrste pojavljuje u nekoj prostornoj vezi. Ponekad se izdvajaju i složenije veze kao što je udaljenost između primitiva. Strukturni opis teksture je sličan rečenicama jer se sastoji od primitivnih elemenata povezanih gramatičkim pravilima. U analizi teksture takav pristup naziva se sintaktički. Iako su se pojavile prije nekoliko desetljeća, strukturne metode opisa teksture u praksi se rijetko koriste, a razlog nalazimo u problemu računalne implementacije strukturnih algoritama i velikoj složenosti.

3.2. Statističke metode

Jedna od definirajućih značajki teksture je prostorna distribucija sivih vrijednosti. Teksture koje imaju slučajnu strukturu pogodne su za karakterizaciju pomoću statističkih metoda koje koriste prostorne statističke zavisnosti između piksela u slici. Uporaba statističkih značajki predstavlja neke od ranijih metoda predloženih u literaturi za računalni vid.

3.2.1. Statističke metode prvog reda

Statističke metode prvog reda zasnivaju se na statistikama prvog reda odnosno na raspodjelama vjerojatnosti značajki pojedinačnih piksela. Primjeri takvih statistika su srednja vrijednost i

varijanca intenziteta. Složenije metode prvog reda za ekstrakciju obilježja teksture zasnovane su na histogramu slike. Histogram slike predstavlja raspodjelu nivoa sivog definiranu izrazom:

$$p_u[u_i] = \text{Prob}\{u = u_i\} = \frac{n_i}{N}, \quad u_i = 0,1,\dots,L-1 \qquad (3-1)$$

Gdje je n_i broj piksela u slici čiji je intenzitet u_i, a N ukupan broj piksela na slici. Na osnovu izračunatog histograma koji je jednodimenzionalna funkcija može se izračunati veći broj obilježja (npr. momenti, apsolutni momenti, entropija) slike koristeći definicije:

Momenti:
$$m_l = E\{u^l\} = \sum_{u_i=0}^{L-1} (u_i)^l p_u[u_i], \quad l = 1,2,\dots \qquad (3-2)$$

Apsolutni momenti:
$$\hat{m}_l = E\{|u|^l\} = \sum_{u_i=0}^{L-1} |u_i|^l p_u[u_i], \quad l = 1,2,\dots \qquad (3-3)$$

Centralni momenti:
$$\mu_l = E\{[u - E(u)]^l\} = \sum_{u_i=0}^{L-1} (u_i - m_1)^l p_u[u_i], \quad l = 1,2,\dots \qquad (3-4)$$

Apsolutni centralni momenti:
$$\hat{\mu}_l = E\{|u - E(u)|^l\} = \sum_{u_i=0}^{L-1} (u_i - m_1)^l p_u[u_i], \quad l = 1,2,\dots \qquad (3-5)$$

Entropija:
$$H = E\{-\log_2 p_u\} = -\sum_{u_i=0}^{L-1} p_u[u_i]\log_2(p_u[u_i]) \qquad (3-6)$$

Od navedenih obilježja najčešće se koriste srednja vrijednost m_1, srednja kvadratna vrijednost m_2, disperzija $\hat{\mu}_1$, varijanca μ_2, asimetrija (engl. *skewness*) μ_3, spljoštenost (engl. *kurtosis)* $\mu_4 - 3$ i entropija. Osim ovih obilježja za karakterizaciju se koriste mod i medijan histograma. Mod histograma je ona vrijednost nivoa sivog za koju histogram ima maksimum. Medijan histograma je ona vrijednost nivoa sivog za koju kumulativni histogram ima vrijednost približno 0.5.

3.2.2. Statističke metode drugog reda

U analizi teksture mogu se koristiti i združene vjerojatnosti drugog reda izračunate na regiji slike odnosno matrica združenog pojavljivanja čiji su elementi definirani izrazom:

$$p_u[u_i,u_j] = p_{u_1,u_2}[u_i,u_j] = \text{Prob}\{u_1 = u_i, u_2 = u_j\} = \frac{n_{i,j}}{N}, \quad u_i,u_j = 0,1,\dots,L-1 \qquad (3-7)$$

gdje je $n_{i,j}$ broj parova piksela u_1 i u_2 koji zadovoljavaju određenu relaciju u dijelu slike od interesa. Ako se izračunaju združene vjerojatnosti $p_u[u_i,u_j]$ za sve moguće vrijednosti u_1 i u_2, dobiva se matrica združenog pojavljivanja $\{p_{u_1,u_2}[u_i,u_j]\}$, dimenzija $L \times L$.

Svaki element matrice združenog pojavljivanja $p[i,j,\delta,G]$ predstavlja procjenu združene vjerojatnosti pojavljivanja para točaka u regiji dimenzija $G \times G$, s intenzitetima i i j, na udaljenosti $\delta = (\Delta x,\Delta y) = (r,\Theta)$ pri čemu r predstavlja udaljenost pomaka, a za specifični kut Θ se uzimaju vrijednosti kuteva od $\Theta = 0°,\ 45°,\ 90°,\ 135°$. Ova statistička metoda drugog reda dobila je naziv statistika združenog pojavljivanja nivoa sivog ili GLC (engl. *gray level co-occurrence*).

Kada je pomak r mali u odnosu na veličinu primitiva (slučaj grubih tekstura), parovi točaka na udaljenosti r će imati slične intenzitete boja pa će vrijednosti elemenata matrice u okolini glavne dijagonale biti znatno više od ostalih. U slučaju kada je pomak r istog reda veličine kao i dimenzija primitiva (slučaj glatkih tekstura), intenziteti točaka na udaljenosti r se poprilično razlikuju pa je raspored elemenata matrice uniformniji. Uporabom GLC matrice moguće je pouzdano procijeniti značajke teksture ako se za više vrijednosti pomaka r izračuna neka mjera disperzije elemenata u odnosu na glavnu dijagonalu. Primjer dobrog pokazatelja disperzije elemenata u odnosu na glavnu dijagonalu je moment inercije:

$$S_I = \sum_{u_i=0}^{L-1}\sum_{u_j=0}^{L-1}(u_i-u_j)^2 p_{u_1,u_2}[u_i,u_j] \qquad (3-8)$$

Osim momenta inercije iz GLC matrice se može izračunati veliki broj drugih obilježja teksture. Najvažnije od tih obilježja su autokovarijanca, energija, entropija, kontrast, homogenost, korelacija itd.:

$$Autokovarijanca: \quad \bar{u}_i = \sum_{u_i=0}^{L-1}\sum_{u_j=0}^{L-1} u_i p_{u_1,u_2}[u_i,u_j] \qquad (3-9)$$

$$\bar{u}_j = \sum_{u_i=0}^{L-1} \sum_{u_j=0}^{L-1} u_j p_{u_1,u_2}[u_i, u_j]$$

Autokorelacija:
$$r_{uu} = \sum_{u_i=0}^{L-1} \sum_{u_j=0}^{L-1} u_i u_j p_{u_1,u_2}[u_i, u_j] \qquad (3-10)$$

Energija:
$$S_G = \sum_{u_i=0}^{L-1} \sum_{u_j=0}^{L-1} (p_{u_1,u_2}[u_i, u_j])^2 \qquad (3-11)$$

Entropija:
$$S_E = -\sum_{u_i=0}^{L-1} \sum_{u_j=0}^{L-1} p_{u_1,u_2}[u_i, u_j] \log_2(p_{u_1,u_2}[u_i, u_j]) \qquad (3-12)$$

Kontrast:
$$CON = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i-j)^2 p[i,j] \qquad (3-13)$$

Homogenost:
$$HOM = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{p[i,j]}{1+|i-j|} \qquad (3-14)$$

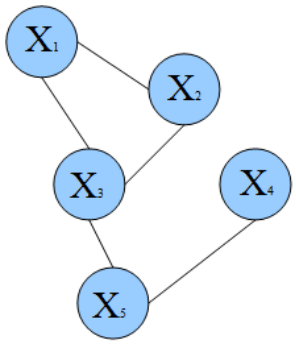
Korelacija:
$$COR = \frac{1}{\sigma_x \sigma_y} \left[\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} ijp[i,j] - \mu_x \mu_y \right] \qquad (3-15)$$

gdje su μ_x i μ_y srednje vrijednosti, a σ_x i σ_y varijance slučajnih vektora dobivenih zbrajanjem GLC matrice. Iako navedeni izrazi mogu poslužiti za ispravnu klasifikaciju različitih tekstura ako ih promatramo odvojeno niti jedan ne sadrži kompletnu informaciju o značajkama i izgledu teksture. Zbog pouzdanosti i dobrih rezultata u velikom broju aplikacija GLC metoda se u praksi često koristi za kvantitativnu karakterizaciju teksture što je u teorijskoj analizi početna točka pri ocijeni novih algoritama.

Nedostatci se mogu naći u tome što je za izračunavanje rezultata iz GLC matrice potreban velik broj aritmetičkih operacija. S obzirom da dimenzije matrice ovise o broju kvantizacijskih nivoa potrebno vrijeme izračuna može se znatno smanjit ako se broj nivoa sivog slike smanji (s 256 na recimo 64 ili 32 nivoa). U slučajevima u kojima to nije moguće koriste se manje pouzdane metode poput statistike razlike intenziteta (engl. *gray level difference*) ili statistike pojavljivanja niza piksela (engl. *gray level run length*).

3.3. Metode temeljene na modelu

Metode opisa teksture na temelju modela vrše modeliranje teksture pomoću stohastičkih metoda ili pomoću fraktala. Značajke teksture se dobiju iz procijenjenih parametara. Na slici 3.1. nalazi se primjer slučajnog Markovljeva polja koje se sastoji od čvorova X_1, X_2, X_3, X_4 i X_5 međusobno povezanih bridovima pri čemu svaki brid predstavlja ovisnost između susjednih čvorova. Slučajna Markovljeva polja modeliraju teksturu na način da se sve sive vrijednosti slike izraze kao funkcija sivih vrijednosti u susjedstvu svakog piksela pri čemu je svaki piksel susjedstva okarakteriziran udaljenošću od središnjeg piksela.



Sl. 3.1. *Graf slučajnog Markovljeva polja.*

U teoriji vjerojatnosti Markovljevo svojstvo označava bezmemorijsko svojstvo slučajnog procesa. Slučajni proces posjeduje Markovljevo svojstvo ako uvjetna gustoća vjerojatnosti nekog budućeg stanja ovisi samo o trenutnom stanju, a ne i o nizu prethodnih stanja. Primjer procesa s Markovljevim svojstvom su Markovljevi lanci koji mogu koristiti za modeliranje jednodimenzionalnih signala tipa zvuk, dok se za modeliranje dvodimenzionalnog signala kakav je slika koriste Markovljeva polja. Model Markovljeva polja predstavlja neko prethodno znanje koje se iskazuje preko vjerojatnosti. Kako bi se lakše mogli modelirati odnosi između elemenata slike koristi se teorija grafova. Ako postoji zajednička funkcija vjerojatnosti:

$$p(X) = p(X_1, X_2, \dots, X_m) \qquad (3-16)$$

i graf $G = (N, E)$ pri čemu je N skup čvorova, a E skup bridova grafa G, svaku varijablu može se pridružiti jednom čvoru grafa $N = \{1,2, \dots, m\}$ i time dobiti slučajno polje na grafu G. Markovljeva polja mogu se primijeniti za sintezu, segmentaciju ili detekciju rubova. Za segmentaciju se konstruira tako da se svakom pikselu pridruži jedna slučajna varijabla, dok se kod detekcije rubova slučajne varijable dodjeljuju parovima piksela i te slučajne varijable poprimaju binarne vrijednosti. [7]

Model fraktala temelji se na fraktalnoj analizi koja se sastoji od skupa matematičkih postupaka za određivanje fraktalne dimenzije. Sam fraktal predstavlja grub ili fragmentiran geometrijski objekt koji se može podijeliti na dijelove pri čemu je svaki umanjena potpuna ili barem približna kopija cjeline (ovo svojstvo naziva se samosličnost), a moguće ga je generirati iteracijom određenih postupaka. Karakterizira ih fraktalna dimenzija koja je strogo veća od topološke dimenzije. Ona se prikazuje brojem i često je jedina mjera fraktala, pokazuje stupanj fragmentacije granica ili neregularnosti u različitim mjerilima. Fraktalna dimenzija također određuje koliko se fraktali razlikuju u odnosu na Euklidske objekte (točka, linija, krug itd.). Upravo na temelju vrijednosti fraktalne dimenzije koja je u korelaciji sa svojstvom hrapavosti tekstura moguće je provesti analizu slike i postupak klasifikacije. [8] Bayesova metoda je korištena i opisana u sklopu eksperimenta klasifikacije u idućem poglavlju kako bi bilo lakše pratiti postupak.

3.4. Transformacijske metode

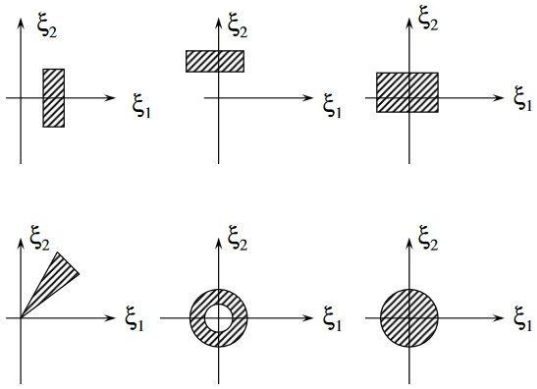
Za kvantitativni opis teksture mogu se iskoristiti i značajke spektra teksture dobivenog primjenom Fourierove, *valićne*, Gaborove ili neke druge transformacije. Značajke mogu biti izdvojene iz transformirane slike uporabom zonalnih maski. Zonalne maske izdvajaju koeficijente koji odgovaraju nekoj značajci. Prvo se linearnoj transformaciji podvrgne područje slike koje nas zanima, a zatim se zonalnom maskom $Z[k,l]$ izdvoje željeni transformacijski koeficijenti:

$$S[k,l] = F[k,l]Z[k,l] \qquad (3-21)$$

koji služe kao značajke za daljnji postupak analize slike. Zonalna maska najčešće predstavlja prozor pravokutnog oblika, kružnog oblika, oblika kružnog isječka ili njihove kombinacije kao što se može vidjeti na slici 3.2.

Kružni segmenti služe za procjenu energije u različitim frekvencijskim opsezima. Maske oblika kružnog isječka su korisne za ocjenu orijentacije karakteristika teksture. Kombinacija kutne i kružne maske je korisna za otkrivanje periodičnih ili aperiodičnih karakteristika. Značajke tekstura u transformacijskom području pronašle su veliku primjenu u praksi, npr. za diskriminaciju vrste zemljišta kod satelitskih snimaka ili detekciju nepravilnosti kod snimki pluća. Kao transformacija se najčešće koristi diskretna Fourierova transformacija. U ovom radu

su za provođenje eksperimenta segmentacije korišteni Gaborovi filteri što je opisano u idućem poglavlju.



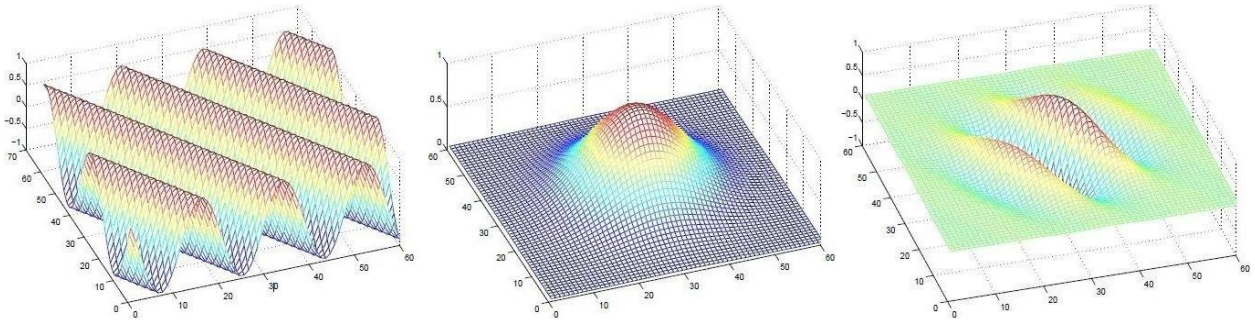
Sl. 3.2. *Najčešći oblici zonalnih maski.*

4. EKSPERIMENTALNI DIO

Kako bi se eksperiment analize tekstura u slici proveo korišten je MATLAB (***mat**rix **lab**oratory*) *R2015b Version 8.6.0.267246* tvrtke MathWorks, programski jezik visokih performansi i interaktivna okolina za numeričko i matrično računanje, razvoj algoritama i aplikacija, analizu, simulaciju, modeliranje i još čitav niz ostalih primjena u području znanosti. Prvi dio eksperimenta sastoji se od implementacije postupka segmentacije tekstura u slici pomoću Gaborovih filtera i metode grupiranja ili klasteriranja algoritmom *k*-srednjih (engl. *k-means*) vrijednosti. Drugi dio eksperimenta čini postupak klasifikacije tekstura na temelju implementacije algoritma za fraktalnu analizu tekstura.

4.1. Segmentacija tekstura u slici metodom Gaborovih filtera

Gaborov filter, nazvan prema Dennisu Gaboru, je linearni filtar koji se koristi za detekciju rubova i ekstrakciju značajki tekstura sa slike, koristeći Gaussovu jezgrenu funkciju moduliranu dvodimenzionalnom sinusoidom. S obzirom na sličnost bioloških osobina vida tj. receptora s frekvencijom i prostornom orijentacijom Gaborovih filtera, oni su se pokazali posebno korisnima za analizu tekstura. Pomoću njega ulazna slika se prostorno i frekvencijski lokalizira tako da ulazni uzorak bude što je moguće bolje vremensko-frekvencijski određen. U prostornoj domeni 2D Gaborov filter je Gaussova funkcija modulirana sinusnim ravnim valom. [9]



(a) (b) (c)

Sl. 4.1. *Vizualizacija Gaborovog filtera.*

Na slici 4.1. može se vidjeti kompozicija Gaborovog filtera: (a) 2D sinusoida orijentiranja 30° obzirom na x-os, (b) Gaussova funkcija, (c) odgovarajući Gaborov filter. Može se jasno uočiti način na koji sinusoida postaje prostorno lokalizirana. Filter sadrži realnu i imaginarnu komponentu pomoću kojih se može formirati kompleksni broj ili se komponente mogu koristiti zasebno. Matematički se to definira na sljedeći način:

Kompleksni zapis:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\exp(i(2\pi\frac{x'}{\lambda}+\psi)) \qquad (4-1)$$

Realna komponenta:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\cos(2\pi\frac{x'}{\lambda}+\psi) \qquad (4-2)$$

Imaginarna komponenta:

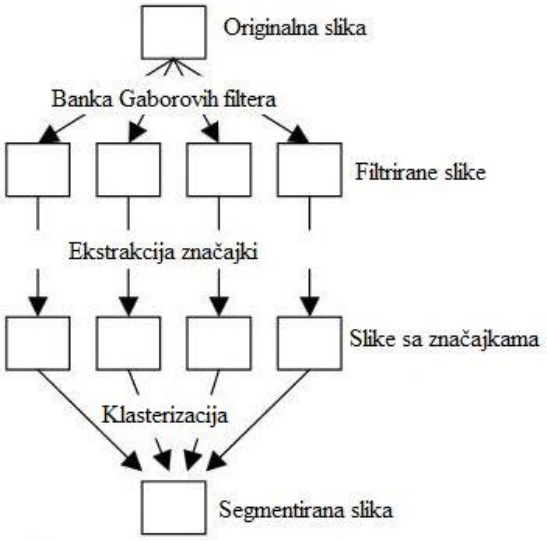
$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\sin(2\pi\frac{x'}{\lambda}+\psi) \qquad (4-3)$$

Pri čemu su:

$$x' = x\cos\theta + y\sin\theta \qquad (4-4)$$

$$y' = -x\sin\theta + y\cos\theta$$

U navedenim matematičkim izrazima, λ predstavlja valnu duljinu sinusoide, θ predstavlja orijentaciju, ψ predstavlja fazni pomak, σ je standardna devijacija Gaussove anvelope ili ovojnice, γ je prostorni omjer dimenzija (eng. aspect ratio).



Sl. 4.2. *Proces segmentacije slike*.

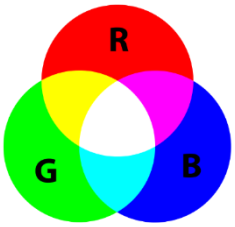
Proces segmentacije se sastoji od nekoliko osnovnih koraka kao što se može vidjeti na slici 4.2. Ulazna slika se propušta kroz banku Gaborovih filtera, određuju se značajke Gaborovih magnituda i klasterizacijom se dobiva segmentirana slika.

4.1.1. Učitavanje i prikaz slike

Prvo se učitava slika s naredbom *imread* i konvertira iz RGB modela u sliku sivih tonova (engl. *grayscale*) pomoću naredbe *rgb2gray*. Ako je slika već u ispravnom formatu nije potrebno konvertirati.

```
ulazna_slika = imread('naziv_slike.jpg');
img_sivi_tonovi = rgb2gray(ulazna_slika);
figure
imshow(ulazna_slika)
```

RGB je kratica za engleski pojam "*Red* (crvena) *Green* (zelena) *Blue* (plava)", a RGB model se definira kao aditivni model boja u kojem crvena, zelena i plava svjetlost zajedničkim dodavanjem na razne načine reproduciraju širok spektar boja poznat čovjekovom sustavu vida. Čovjek ima osjetila za tri osnovne boje na temelju čega se pretpostavilo da se sve boje mogu prikazati kao linearna kombinacija triju osnovnih komponenti. Tako npr. za opis jedne boje potrebno je poznavati udio crvene, zelene i plave komponente.



Sl. 4.3. *Aditivno miješanje osnovnih boja*

Svaka komponenta se prikazuje nizom od 8 bitova što u konačnici iznosi 256 (2^8) različitih vrijednosti. Obzirom da postoje 3 komponente moguće je prikazati 256^3 različitih boja. RGB model glavnu je svrhu pronašao u reprezentaciji i prikazu slika u sustavima poput televizora i računala. Pretvorba slike u boji u sive tonove provodi se tako da se vrijednosti crvene, zelene i plave komponente postave na istu vrijednost koja je jednaka sumi umnožaka crvenih, zelenih i plavih komponenti s težinskim faktorima. Faktori su određeni na osnovu osjetljivosti oka i iznose 0.299 za crvenu, 0.587 za zelenu i 0.114 za plavu komponentu.

$$Agray = 0.299R + 0.587G + 0.114B \qquad (4-5)$$

Naredbom *figure* otvara se novi prozor za prikaz slike, a s *imshow* prikazuje ulazna slika. Na slici 4.4. (a) nalaze se prirodne teksture pustinjskog pijeska i neba, slika (b) prikazuje teksture slova i pozadine, a na (c) su Brodatzove teksture zida lijevo i pletiva od palme rafije desno. Ove

dvije Brodatzove teksture su odabrane slučajno i spojene u jednu sliku pomoću jednostavnog programa za obradu slike.



(a) (b) (c)

Sl. 4.4. *Ulazne slike u smanjenom mjerilu koje je bilo potrebno segmentirati*.

4.1.2. Dizajniranje reda Gaborovih filtera

Jedan od početnih koraka je dizajn reda Gaborovih filtera podešen za različite frekvencije i orijentacije. U MATLAB-u se koristi naredba *gabor* za dobivanje Gaborovih filtera. Ona sadrži punu parametrizaciju Gaborovih filtera i definira sve veličine korištene pri radu. Parametri:

Orientation - orijentacija (u stupnjevima)

Wavelength - valna duljina sinusoide (u broju piksela po ciklusu)

SpatialAspectRatio - prostorni omjer dimenzija koji određuje eliptičnost Gaborove funkcije

SpatialFrequencyBandwidth - prostorna širina filtera (u oktavama)

SpatialKernel - kompleksna prostorna konvolucijska jezgra

Za stvaranje reda Gaborovih filtera sa specificiranim valnim duljinama i orijentacijama koristi se MATLAB funkcija *g = gabor(WAVELENGTH,ORIENTATION)*. *WAVELENGTH* opisuje valnu duljinu sinusoide s rasponom vrijednosti u intervalu [2, beskonačno) piksela/ciklusu. *ORIENTATION* je orijentacija filtera i definira se kao kut između normale i sinusnog ravnog vala te može poprimiti vrijednosti u intervalu [0, 360] stupnjeva. U slučaju kada su *WAVELENGHT* i *ORIENTATION* vektori, *g* predstavlja red Gaborovih objekata koji sadrže sve jedinstvene kombinacije valnih duljina i orijentacija.

Kako bi se pronašle informacije na učitanoj slici odabire se odgovarajući set frekvencija i orijentacija. U praksi su to najčešće orijentacije između [0, 150] stupnjeva u koracima od 45° i valna duljina uzorka s rastućom potencijom broja 2 počevši od $\sqrt[4]{2}$ do duljine hipotenuze zadane

slike. Odabir većeg broja orijentacija može rezultirati točnijom segmentacijom ali također povećati vrijeme obrade računalom. U eksperimentu su korištene vrijednosti orijentacija od 0° do 150° u koracima od 30°. Slično se može reći i za vrijednosti valnih duljina, izborom većeg broja valnih duljina dizajnirat će se veća banka filtera, povećati vrijeme obrade ali rezultat u konačnici može biti puno lošiji nego s manjim brojem valnih duljina. U eksperimentu za slike 4.4. (a) i (b) korišten je manji broj filtera i valna duljina od 2 piksela/ciklusu kako bi se postigao dobar rezultat, dok je za sliku (c) zbog kompleksnosti uzorka korišten veći broj filtera s valnim duljinama 2, 4, 8, 16, 32 i 64 piksela/ciklusu.

```
velicina_slike = size(ulazna_slika);

numRows = velicina_slike(1);

numCols = velicina_slike(2);
```

Naredba *numRows* vraća broj redova, a *numCols* vraća broj stupaca zadane slike i koriste se za izračun vrijednosti hipotenuze.

```
valna_duljina_Min = 4/sqrt(2);

valna_duljina_Max = hypot(numRows,numCols);

n = floor(log2(valna_duljina_Max/valna_duljina_Min));

wavelength = 2.^(0:(n-2)) * valna_duljina_Min;

deltaTheta = 30;

orientation = 0:deltaTheta:(180-deltaTheta);

g = gabor(wavelength,orientation);
```

Nakon što su definirane vrijednosti valne duljine i orijentacije odredi se red Gaborovih filtera *g* koji se još naziva i banka filtera jer sadrži sve kombinacije valnih duljina i orijentacija. Tako npr. za valne duljine 2 i 4, te kuteve 0°, 45°, 90° i 135° napravit će se red od osam filtera. Naredba *floor* zaokružuje broj na najbližu cjelobrojnu vrijednost prema minus beskonačno.

Zatim je potrebno izvući Gaborove magnitude iz slike. Rad s magnitudama odziva svakog filtera je tipičan kod ovog postupka jer daje dobre rezultate.

```
gabormag = imgaborfilt(img_sivi_tonovi,g);
```

Naredbom *imgaborfilt* primjenjuje se Gaborov filter ili vise njih na 2D sliku. Ovdje je naredba *imgaborfilt* iskorištena za računanje magnituda i faza odziva Gaborovih filtera s ulaznom slikom prethodno konvertiranom u sliku sivih tonova.

4.1.3. Naknadna obrada slika Gaborovih magnituda u Gaborove značajke

Postupak naknadne obrade se provodi kako bi se mogli iskoristiti odzivi Gaborovih magnituda. To uključuje Gaussovo zaglađivanje (engl. *Gaussian smoothing*), dodavanje dodatnih prostornih informacija u set značajki, njegovo preoblikovanje u matrični oblik kakav se očekuje od *pca* i *kmeans* funkcija i normalizacija informacija značajki u zajedničku varijancu i srednju vrijednost. Svaka slika Gaborove magnitude sadrži neke lokalne varijacije čak i unutar dobro segmentiranih regija koje mogu poremetiti postupak segmentacije. To najčešće rezultira pojavom točkastih artefakata ili pogrešno detektiranim granicama u segmentiranoj slici. Kako bi se te varijacije kompenziralo primjenjuje se postupak zaglađivanja informacija Gaborovih magnituda primjenom jednostavnog Gaussovog niskopropusnog filtriranja. Odabrana vrijednost *sigma* se podudara sa Gaborovim filterom koji je izvukao svaku značajku. Uvodi se koeficijent zaglađivanja *K* koji kontrolira koliko točno zaglađivanja je upotrijebljeno na odzive Gaborovih magnituda. Za sve tri ulazne slike vrijednost *K* je postavljena na šest kako bi se postigao dobar rezultat.

```
for i = 1:length(g)
sigma = 0.5*g(i).Wavelength;
K = 6;
gabormag(:,:,i) = imgaussfilt(gabormag(:,:,i),K*sigma);
end
```

Kod konstruiranja Gaborovog seta značajki korisno je dodati mapu prostornih lokacija za *X* i za *Y*. Ove dodatne informacije omogućuje odabir grupiranja koja su prostorno bliža. Naredbom *meshgrid* stvara se Kartezijev koordinatni sustav, a naredbom *cat* lančano spajaju polja (*X, Y, gabormag, featureSet*) duž dimenziju veličine tri.

```
X = 1:numCols;
Y = 1:numRows;
[X,Y] = meshgrid(X,Y);
```

featureSet = cat(3,gabormag,X);

featureSet = cat(3,featureSet,Y);

numPoints = numRows*numCols;

X = reshape(featureSet,numRows*numCols,[]);

Podatci se oblikuju u matricu **X** u formu koja se očekuje od *pca* i *kmeans* funkcije. Svaki piksel u rešetki slike je odvojena podatkovna točka (engl. *datapoint*), a svaka *x*, *y* ravnina u varijabli *featurSet* je zasebna značajka. Prema tome svaki filter u Gaborovoj banci filtera predstavlja zasebnu značajku uz dvije dodatne značajke dobivene iz prostorne informacije. Tako ukupno postoji specifičan broj Gaborovih značajki ovisno o ulaznoj slici i parametrima filtera te dvije dodatne prostorne značajke za svaki piksel slike.

Idući važan korak je normalizacija značajki pomoću *bsxfun* naredbe koja primjenjuje specificirane binarne operacije element po element na redcima matrica.

X = bsxfun(@minus, X, mean(X));

X = bsxfun(@rdivide,X,std(X));

U ovom slučaju kako bi se provela normalizacija cilj je oduzeti srednju vrijednost od značajke i podijeliti vrijednost novo dobivene značajke (od koje je srednja vrijednost prethodno oduzeta) sa standardnom devijacijom. Kod matrica naredba *mean(X)* vraća srednju vrijednost tj. vektor redak koji sadrži srednje vrijednosti svakog stupca, a naredba *std(X*) vraća standardnu devijaciju tj. vektor redak koji sadrži standardne devijacije svakog stupca.

Kako bi se dobio jasan dojam kako značajke Gaborovih magnituda doista izgledaju može se upotrijebiti PCA metoda (engl. *Principal Component Analysis*) ili analiza glavnih komponenti te se premjestili iz višedimenzionalne reprezentacije svakog piksela ulazne slike u 1-D vrijednost intenziteta za svaki piksel. Ona se primjenjuje kada je velik broj varijabli u skupu redundantan, odnosno kada se više varijabli odnosi na istu dimenziju i kada ne pružaju nikakvu dodatnu informaciju koja već nije obuhvaćena nekom drugom varijablom. Cilj PCA metode je kreiranje *p* linearnih kombinacija izvornih varijabli koje se nazivaju glavne komponente (engl. *principal components*). Primjenom analize glavnih komponenti nije uvijek moguće veliki broj originalnih varijabli reducirati na mali broj izvedenih varijabli. Ako originalne varijable nisu u korelaciji tada se analizom neće postići nikakav rezultat. Ako su originalne varijable u visokoj korelaciji (bilo pozitivnoj ili negativnoj) tada se postižu najbolji rezultati. Mogućnosti ove metode prvi

puta je opisao Karl Pearson 1901. godine, a praktične računske metode razradio Hotelling 1933. no zbog velike kompleksnosti metoda nije bila učinkovita na velikom broju varijabli. Široka primjena uslijedila je razvojem brzih računala.

```
coeff = pca(X);

znacajke_2Dslike = reshape(X*coeff(:,1),numRows,numCols);

figure

imshow(znacajke_2Dslike,[])
```

Coeff = pca(X) vraća koeficijente glavnih komponenti za **N x P** podatkovnu matricu **X**. Redci matrice **X** odgovaraju opažajima, a stupci varijablama. Svaki stupac **coeff** matrice sadrži koeficijente za jednu glavnu komponentu, a stupci su u padajućem redoslijedu obzirom na varijancu komponente. Pomoću naredbe *reshape* vrši se pretvorba matrice kako bi se dobio njen oblik u kojem je moguće prikazati sliku naredbom *imshow*. Kao argument je uzeta matrica **X*coeff(:,1)**, dok varijable *numRows* i *numCols* predstavljaju dimenziju slike. Tako preoblikovana matrica sprema se u *znacajke_2Dslike*. Naredba *figure* otvara novi prozor za prikaz slike.



(a) (b) (c)

Sl. 4.5. *Prikaz značajki Gaborovih magnituda pomoću PCA metode.*

Naredbom *imshow* prikazuju se značajke Gaborovih magnituda što se može vidjeti na slici 4.5. za (a) prvu ulaznu sliku, (b) drugu ulaznu sliku i (c) treću ulaznu sliku. Vrijednosti minimuma su prikazane crnom bojom, a vrijednosti maksimuma bijelom. Također se može jasno vidjeti koja područja slike će biti problematična za segmentaciju. Na slici 4.5. (a) to je rubno područje koje odvaja teksture pijeska i neba. Na slici 4.5. (b) to su vanjska područja koja obrubljuju slova odnosno konture, a nastala su jer font koristi sjenčanje. Na slici 4.5. (c) to je također rubno područje koje odvaja dvije teksture te donji dio desne regije koji je tamniji.

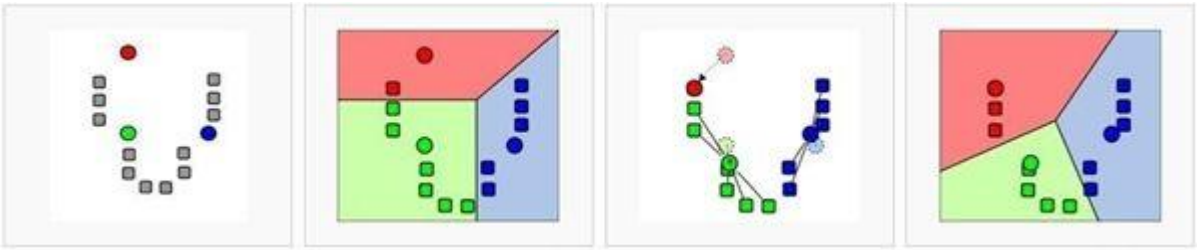
4.1.4. Klasterizacija ili grupiranje algoritmom *k*-srednjih vrijednosti

Klasterizacija metodom *k*-srednjih vrijednosti predstavlja metodu vektorske kvantizacije koja se originalno razvila za potrebe procesiranja signala ali je postala popularna u analizi grupa ili klastera i rudarenju podataka (engl. *data mining*). Prema općenitoj definiciji cilj klasterizacije je particioniranje *n* opažaja u *k* grupa ili klastera pri čemu je:

a) svaka grupa ili klaster homogena obzirom na određene varijable (značajke), tj. da su svi opažaji u jednoj grupi slični jedni drugima

b) svaka grupa različita od druge obzirom na te iste varijable (značajke), tj. da se opažaji u jednoj grupi moraju razlikovati od opažaja u drugoj grupi.

Geometrijski svaki opažaj se može prikazati kao točku u *p* dimenzionalnom prostoru pri čemu je *p* broj varijabli (značajki) koje opisuju opažaj. Svaku grupu predstavlja centroid (srednja vrijednost grupe), a za udaljenost između klastera koristi se kvadrat Euklidske udaljenosti između njihovih centroida. Algoritam *k*-srednjih vrijednosti radi na principu slučajnog ili heurističkog odabira *k* točaka koje će biti početne vrijednosti centara i iterativno se izvršava dok ne konvergira. [12]



(a) (b) (c) (d)

Sl. 4.6. *Slikovni prikaz rada kmeans algoritma*.

Na slici 4.6. može se vidjeti slikovni prikaz rada algoritma *k*-srednjih vrijednosti: (a) slučajno se odabire *k* točaka koje će biti početne vrijednosti centara, (b) formiraju se *k* klasteri tako da se točke dodjeljuju najbližem centru, (c) centroid svakog *k* klastera postaje novi centar, (d) ponavlja se korak (a) i (b) dok se ne postigne konvergencija. Faza izvršavanja algoritma sastoji se od dva iterativna koraka, u prvom se formiraju grupe tako da se točke iz skupa dodjeljuju najbližem centru, a u drugom se izračunaju nove vrijednosti centra tako da se izračuna srednja vrijednost svih točaka u grupi. Algoritam konvergira kada se više ne mijenja raspodjela točaka po grupama. Cilj algoritma je dobiti takve grupe u kojima je varijanca između točaka unutar svake grupe što manja. Osim složenosti u izračunu kod velikog broja varijabli koja rezultira jako sporom obradom kao nedostatke ovog algoritma mogu se izdvojiti unaprijed zadan parametar za broj

srednjih vrijednosti (može se izbjeći primjenom heurističkog postupka) i uporabu Euklidske udaljenosti kao kriterija za određivanje pripadnosti pojedinoj grupi što se u nekim slučajevima pokazalo lošim za kvalitetnu segmentaciju.

rezultat_segm = kmeans(X,broj različitih regija slike,'Replicates',5);

Klasterizacija metodom *k*-srednjih vrijednosti ponavlja se pet puta kako bi se minimizirala pogreška pri čemu je potrebno točno definirati koliko različitih regija tekstura je prisutno na slici koja se segmentira. Sve tri ulazne slike imaju dvije različite regije. Naredba *kmeans* particionira točke **N x P** matrice podataka **X** u *k* klastere pri čemu redci predstavljaju točke, a stupci predstavljaju varijable.

4.1.5. Prikaz rezultata segmentacije

Kako bi se vizualizirala segmentacija prvo se preoblikuje matrica u prikladan oblik te koristi naredba *label2rgb* koja konvertira tako dobivenu matricu u sliku RGB modela boja. U ovom slučaju tamno plava boja ispunjava jednu regiju slike dok svijetlo plava ispunjava drugu regiju što se može vidjeti na slici 4.7. za sve tri ulazne slike.

rezultat_segm = reshape(rezultat_segm,[numRows numCols]);

figure

imshow(label2rgb(rezultat_segm))



(a) (b) (c)

Sl. 4.7. *Vizualizacija segmentacije.*

Za jednostavniji i pregledniji usporedni prikaz ulazne slike i segmentirane slike korišteno je transparentno preklapanje rezultata preko ulazne slike. S naredbom *hold on* zadržava se ulazna slika kako bi se preko nje prikazao rezultat segmentacije. Količina prozirnosti regulira se pomoću vrijednosti *AlphaData*.

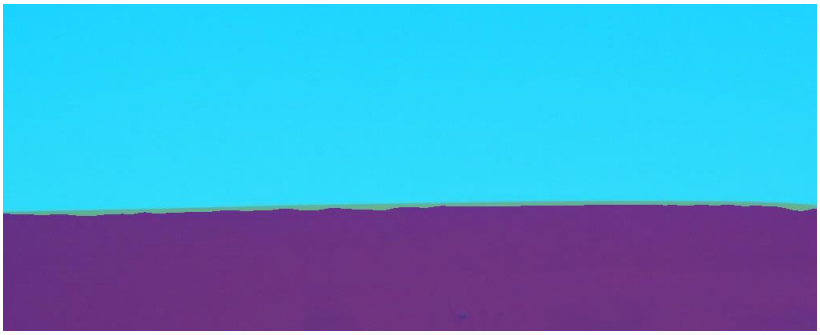
figure

imshow(ulazna_slika)

hold on

preklopi = imshow(label2rgb(rezultat_segm));

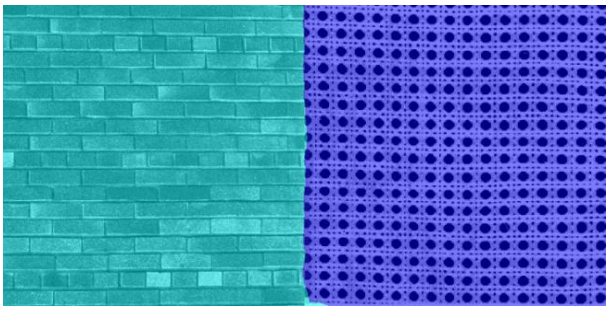
preklopi.AlphaData = 0.5;



(a)



(b)



(c)

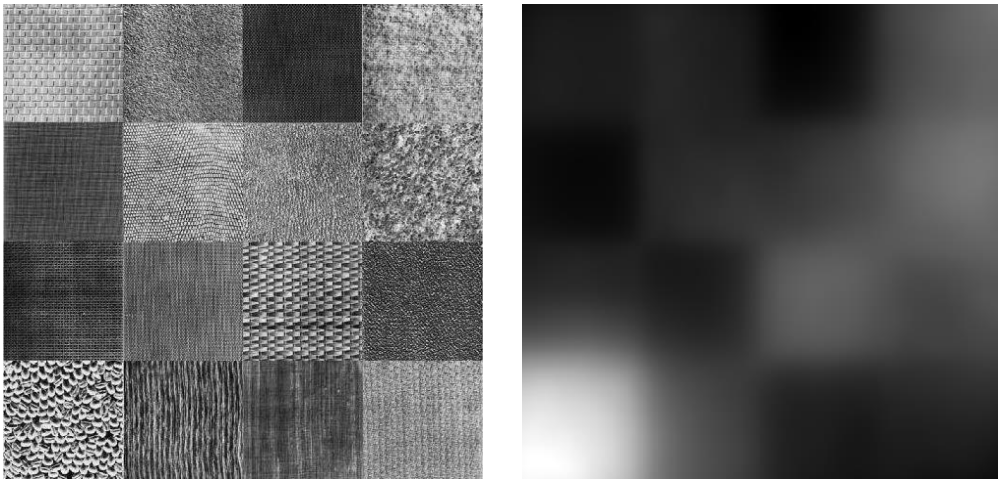
Sl. 4.8. *Usporedni prikaz ulazne slike i rezultata segmentacije.*

Na slici 4.8. može se vidjeti usporedni prikaz ulazne slike i rezultata segmentacije za sve tri ulazne slike. Na slici 4.8. (a) regije su dobro segmentirane, bez vidljivih artefakata ali granično područje za koje je prethodno uspostavljeno da bi moglo biti problematično nije u potpunosti točno segmentirano jer regija s teksturom neba ulazi u regiju s teksturom pijeska. Do toga je došlo jer je područje horizonta gdje se sunčeva svijetlost odbija od pješčane podloge kod fotografiranja svijetlije boje što je otežalo segmentaciju. Na slici 4.8. (b) zbog sjenčanja koje koristi font, konture su svijetlije i segmentirane su zajedno s pozadinom što je očekivani rezultat. Na slici 4.8. (c) iako se radi o najtežem uzorku segmentacija je dala zadovoljavajući rezultat. Regije su točno segmentirane, a najvećim dijelom i granično područje. Greška se javila u donjem dijelu graničnog područja gdje dolazi do malenog prijelaza jedne regije u drugu.

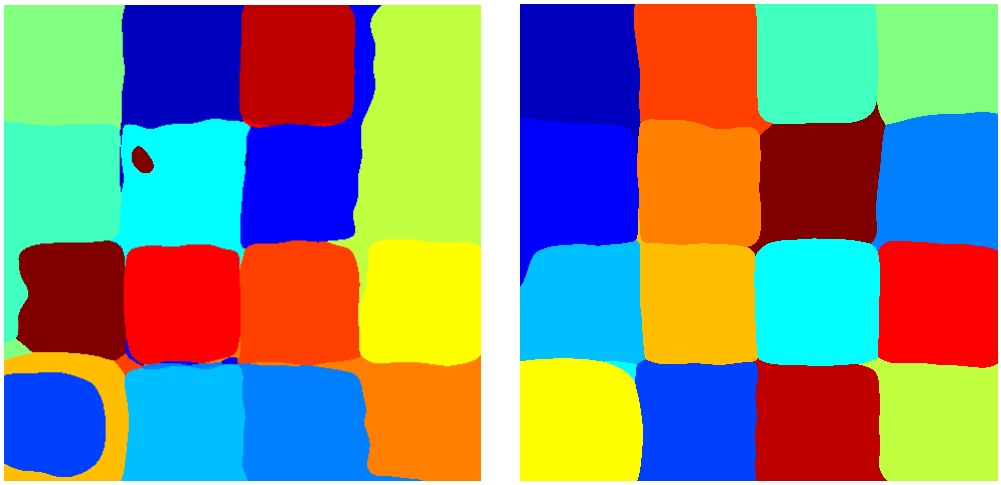
Isti postupak je proveden za sliku 4.9. (a) s većim brojem različitih regija tekstura ali nažalost nije pokazao zadovoljavajući rezultat niti nakon mnogobrojnih izmjena parametara filtera u MATLAB-u.

ulazna_slika = imresize(ulazna_slika,0.3);

Nakon što je slika učitana bilo je potrebno smanjiti njene dimenzije od 1578x1574 piksela za 70% pomoću *imresize* naredbe jer zbog početnih dimenzija vrijeme obrade bi trajalo satima, a i potreba za računalnim resursima, prvenstveno memoriji mogli bi dovesti do rušenja sustava.



(a) (b)



(c) (d)

Sl. 4.9. *Prikaz rezultata segmentacije kompleksnog uzorka od 16 različitih tekstura.*

Na slici 4.9. (c) i (d) može se vidjeti rezultat segmentacije slike sa šesnaest različitih Brodatzovih tekstura spojenih u jednu sliku. Kako bi se dobio rezultat na slici 4.9. (c) korišten je manji broj valnih duljina (2, 4 i 8 piksela/ciklusu) i orijentacija (0°, 45°, 90° i 135°), dok je veća točnost postignuta s većim brojem valnih duljina (2, 4, 8, 16, 32, 64 i 128 piksela po ciklusu) i orijentacija (0°, 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120° i 135°). Slika 4.9. (b) prikazuje Gaborove magnitude za rezultat na slici (d). Unatoč određenoj razini točnosti u prepoznavanju granica i regija tekstura očekivano je došlo do značajnih pogrešaka zbog velike složenosti uzorka što se može vidjeti u prelijevanju boja i pogrešno definiranim granicama i regijama. Točnost postupka u zahtjevnijim situacijama može se poboljšati smanjenjem dimenzija uzorka, korekcijom količine zaglađivanja ili nekim kompleksnijim kombiniranim metodama segmentacije u odnosu na osnovnu metodu korištenu u ovom eksperimentu.

4.2. Klasifikacija tekstura u slici implementacijom algoritma za fraktalnu analizu tekstura

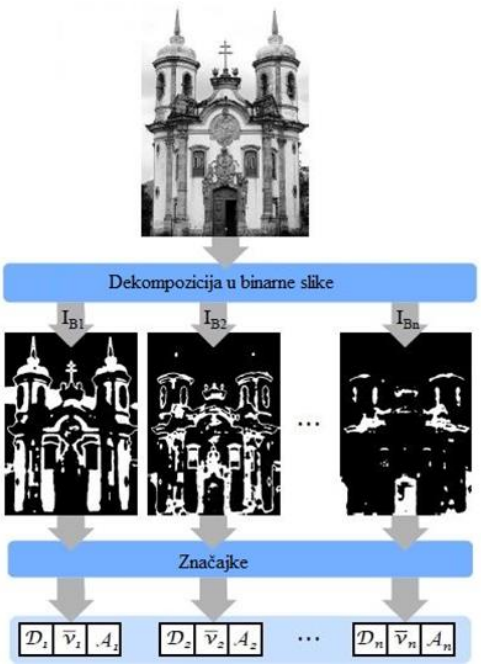
Kao sto je već spomenuto klasifikacija se odnosi na pridruživanje fizičkog objekta ili slučaja u jednu od nekoliko predefiniranih kategorija. Cilj ovog eksperimenta je provesti postupak pridruživanja jedne slike teksture odgovarajućoj klasi tekstura. On se u osnovi temelji na učenju klasifikatora pomoću seta označenih slika s prikazom tekstura te njegovog korištenja za dobivanje oznake kod neoznačene slike. Za provođenje eksperimenta korišten je već izrađen i dostupan za preuzimanje *SFTA* (engl. *Segmentation-based Fractal Texture Analysis*) algoritam za fraktalnu analizu tekstura zasnovan na segmentaciji. Klasifikaciju pomoću *SFTA* algoritma može se podijeliti u 3 koraka:

(1) ekstrakcija značajki tekstura

(2) treniranje klasifikatora

(3) klasifikacija neoznačenih slika s teksturama

Algoritam se koristi za ekstrakciju značajki tekstura iz slike pomoću funkcije *sfta (I, nt)* pri čemu *I* predstavlja ulaznu sliku u sivim tonovima, a *nt* je parametar koji definira veličinu vektora značajki. Značajke se vraćaju kao vektor dimenzija 1 x (6*nt-3). Postupak ekstrakcije se sastoji od dekompozicije ulazne slike u set binarnih slika iz kojeg se računaju fraktalne dimenzije regija kako bi se opisali segmentirani uzorci teksture. Fraktalna dimenzija je vrijednost koja nam daje uvid u to u kojoj mjeri neki fraktal ispunjava prostor u kojem se nalazi. [13]



Sl. 4.10. *Ekstrakcija značajki.*

Na slici 4.10. može se vidjeti postupak dekompozicije ulazne slike i dobivanja *sfta* vektora značajki kao rezultat veličine binarne slike $A_1, A_2, \dots A_n$, srednje vrijednosti nivoa sivog $\overline{v}_1, \overline{v}_2, \dots \overline{v}_n$ i fraktalne dimenzije rubova $D_1, D_2, \dots D_n$. Za početak postupka potrebno je pronaći odgovarajuću set slika s teksturama koje su podijeljene prema istovjetnim značajkama u odgovarajuće kategorije i koji će poslužiti za treniranje klasifikatora. Za tu potrebu korišten je set od tisuću slika s raznim teksturama (staklo, drvo, voda, tkanina, cigla, granit, tapeta, krzno, sag itd.), a izradila ga istraživačka grupa prof. Jeana Poncea te je dostupan za slobodno preuzimanje preko interneta pod uvjetom da se citira njihov rad. [14] Slike su podijeljene u 25 klasa, svaka sadrži 40 uzoraka dimenzija 640x480 piksela u *.jpg* formatu sivih tonova. Nakon toga potrebno je raspakirati preuzete komprimirane datoteke u jedan folder nazvan *dataset* i kreirati tekstualnu datoteku naziva *slike_klase.txt* koja sadrži popis svih slika unutar *dataset* foldera pri čemu je u svakom pojedinačnom retku napisan naziv pojedinačne slike i klase kojoj pripada, tako npr. u prvom retku piše *T07_40.jpg, water*. Za treniranje klasifikatora postoje mnoga već implementirana rješenja za klasifikaciju poput *Weka* algoritma ili *LIBSVM*-a. U MATLAB-u za tu svrhu se može koristiti *naivan* Bayes klasifikator. Kako bi se on trenirao potrebno je organizirati vektore značajki u jednu **N x M** matricu gdje **N** predstavlja ukupan broj slika u uzorku za treniranje, a **M** veličinu vektora značajki. Također je potrebno definirati i polje ćelija **N x 1** u kojem se nalaze oznake klasa slika. Prema definiciji *naivan* Bayesov klasifikator predstavlja jednostavan probabilistički klasifikator koji se temelji na primjeni Bayesova teorema s jakim (*naivnim*) nezavisnim pretpostavkama između značajki. Bayesov teorem se definira na sljedeći način:

$$P(c_i|A) = \frac{P(c_i)P(A|c_i)}{P(A)} \qquad (4-6)$$

pri čemu su c_i hipoteze i također klase, A predstavlja ulaznu klasu za koju se provjerava pripada li određenoj klasi c_i, $P(c_i|A)$ je aposteriorna vjerojatnost za ulaznu klasu odnosno vjerojatnost hipoteze c_i ukoliko imamo klasu A, $P(c_i)$ je apriorna vjerojatnost hipoteze, $P(A)$ je vjerojatnost pojavljivanja klase A, $P(A|c_i)$ je uvjetna vjerojatnost klase A ukoliko vrijedi hipoteza c_i. [15]

Naivni Bayesov klasifikator prema formuli (4-7) definiran je kao umnožak apriorne vjerojatnosti $P(c_i)$ neke klase c_i i uvjetnih vjerojatnosti $P(A_j|c_i)$. Uvjetna vjerojatnost predstavlja vjerojatnost pojavljivanje neke značajke iz klase A u klasi c_i. Ovaj klasifikator se naziva *naivnim* jer pretpostavka o uvjetnoj nezavisnosti u praksi uglavnom ne vrijedi.

$$P(c_i|A) = P(c_i)\prod_{j=1}^{n} P(A_j|c_i) \qquad (4-7)$$

Kao prednosti *naivnog* Bayesovog klasifikatora može se navesti da mu je dovoljan manji uzorak podataka za treniranje da bi se provela klasifikacija, jednostavan je i skalabilan, ima dobre performanse, otporan je na šum i može se prilagoditi za gotovo bilo koji problem klasifikacije. Problem ovog klasifikatora je što ne može raditi s velikim brojem značajki zbog računanja uvjetnih vjerojatnosti za svaku značajku. Pokretanjem sljedećeg koda u MATLAB-u:

```
slikePopis = fopen('slike_klase.txt', 'r');
F = zeros(1000, 21); L = cell(1000, 1);
redak = fgetl(slikePopis); trenutniRedak = 1;
while ischar(redak)
   disp(redak)
   razdvoji = regexp(redak, ',[ ]*', 'split');

      lokacija = fullfile('dataset', razdvoji{1});
   I = imread(lokacija);
   f = sfta(I, 4);
   F(trenutniRedak, :) = f;
   L{trenutniRedak} = razdvoji{2};
   redak = fgetl(slikePopis);
   trenutniRedak = trenutniRedak + 1;
end;
fclose(slikePopis);
```

čita se datoteka *slike_klase.txt* te generira matrica ***F*** s vektorima značajki i polje ćelija ***L*** s oznakama slika. S naredbom *fopen* otvara se datoteku u modu za čitanje *'r'*. Naredba *zeros* kreira nul-matrice, a naredba *cell* kreira prazno polje ćelija. S *while* petljom prolazi se kroz datoteku sve dok *ischar* naredba vraća vrijednost 1, a činit će to sve dok redak sadrži polje sa znakovima. S *disp* naredbom ispisuju se redci u MATLAB naredbenom prozoru sve dok se ne prođe kroz cijelu datoteku i izvrše sve naredbe unutar petlje. Obzirom na uzorak od tisuću slika to može

potrajati, ovisno na kakvom *hardwareu* se izvodi. U ovom slučaju je bilo potrebno 1476 sekunda ili 24,6 minuta (izmjereno pomoću naredbe *tic* i *toc*). S *regexp* razdvaja se oznaka klase od naziva slike u retku. Naredba *fullfile* se koristi za dobivanje cjelovite lokacije gdje je slika (npr. *lokacija = C:\User\Documents\dataset\slika.jpg*) čiji naziv je dobiven unutar datoteke *slike_klase.txt* i retka u kojem se trenutno nalazimo u petlji. U idućoj iteraciji prelazi se u novi redak te se dohvaća slika koja se u njemu nalazi. Naredba *imread* čita sliku i stavlja je u matricu nad kojom se provodi ekstrakcija vektora značajki pomoću *SFTA* algoritma i zatim formira matrica ***F*** svih vektora značajki. Nakon toga koriste se već poznate naredbe za formiranje polja ćelija s oznakama slika i prelazak u novi redak u datoteci *slike_klase.txt*. Kada se izvrši izvođenje petlje odnosno formira matrica ***F*** zatvara se datoteka naredbom *fclose*. Nakon generiranja matrice ***F*** i polje ćelija ***L*** sljedeći korak je treniranje *naivnog* Bayes klasifikatora *nb_klasifikator* naredbom:

nb_klasifikator = NaiveBayes.fit(F, L);

Trening podatci su sadržani u matrici ***F***, polje ćelija ***L*** sadrži popis imena klasa za svaku sliku pri čemu svaki redak u ***L*** definira kojoj klasi pripada odgovarajući redak u matrici ***F***. Važno je napomenuti da zbog toga ***F*** i ***L*** moraju imati identičan broj redaka. Na samom kraju postupka provodi se klasifikacija željene ulazne slike korištenjem *nb_klasifikatora*.

ulazna_slika = imread('ulazna_slika.jpg')

znacajke = sfta(ulazna_slika, 4)

predict(nb_klasifikator, znacajke)

Za početak testirana je ispravnost algoritma s ulaznim slikama koje su se već nalazile u setu slika korištenom za treniranje klasifikatora i on je bez greške određivao kojoj klasi slike pripadaju. Zatim je uzeto nekoliko nasumičnih slika iz seta i u jednostavnom programu za obradu slike promijenjen raspored elemenata na slici kako bi se dobio uzorak koji se ne nalazi u originalnom setu ali tako da dobiveni uzorak ne bi previše odstupao od klase kojoj pripada. Primjer je slika 4.11. s kariranim uzorkom. I u ovom slučaju algoritam je točno klasificirao ulaznu sliku. Na slici 4.12. u crvenom pravokutniku nalazi se rezultat uspješne klasifikacije kariranog uzorka sa slike 4.11. u MATLAB-u gdje je navedeno da ulazna slika pripada klasi karirano (engl. *plaid*). Nakon toga uslijedilo je klasificiranje još deset slika tekstura preuzetih s interneta koje se uvelike razlikuju od uzorka za treniranje i algoritam je očekivano pokazao nisku razinu točnosti. Tek su dvije od deset slika točno klasificirane. Uzrok tome može se pronaći u

velikom odstupanju ulaznih slika od onih koje su korištene za treniranje klasifikatora te upravo u nedovoljno velikom uzorku podataka za treniranje.



Sl. 4.11. *Karirani uzorak.*



Sl. 4.12. *Rezultat klasifikacije u MATLAB-u.*

5. ZAKLJUČAK

U ovome radu opisani su osnovni pojmovi u području istraživanja koje se bavi analizom tekstura u slici. To uključuje problematiku same definicije pojma teksture te opis četiri glavna problema kod obrade i analize tekstura, klasifikaciju, segmentaciju, sintezu i dobivanje oblika iz teksture. Teorijski su objašnjene metode za kvantitativni opis teksture čija svrha je izračunavanje lokalnih obilježja slike. One uključuju strukturnu metodu, statističke metode, metode koje se temelje na modelu i transformacijske metode. U sklopu statističke metode obrađene su statističke metode 1. i 2. reda, kod metoda temeljenih na modelu ukratko je dan pregled Markovljevih polja, modela fraktala i Bayesove metode, a kod transformacijskih metoda opisani su Gaborovi filteri. Unatoč mnoštvu metoda koje danas imamo na raspolaganju i od kojih su neke našle i široku praktičnu primjenu još uvijek ne postoji konzistentna teorija pa se može zaključiti kako je u području analize tekstura potrebno provoditi daljnja istraživanja i razvijati nove efikasnije metode. U eksperimentalnom dijelu rada riješena su dva problema, prvi se zasniva na segmentaciji, a drugi na klasifikaciji. Eksperimenti su provedeni pomoću MATLAB-a s rezultatima za koje se može ustvrditi da su očekivani i zadovoljavajući. Osnovna metoda Gaborovih filtera i grupiranja algoritmom *k*-srednjih vrijednosti je učinkovita u eksperimentalnoj primjeni i daje dobre rezultate kod segmentiranja tekstura u slici ali zbog nedostataka u slučajevima kada je potrebno segmentirati kompleksan uzorak i vremenskog trajanja čitavog postupka osnovna metoda nije prikladna za praktičnu uporabu. Kod ovog postupka veliki nedostatak je i u tome što je za svaku sliku kod segmentacije potrebno unaprijed definirati broj regija. Razvijene su i brojne složenije kombinirane metode također temeljene na Gaborovim filterima koje daje dobre rezultate i kod kompleksnih uzoraka uz optimalno vrijeme računalne obrade ali zbog ograničenog opsega ovog rada nisu implementirane i opisane. Jedan od problema kod segmentacije bila je evaluacija rezultata jer se ona ne može egzaktno definirati već ovisi o direktnoj analizi promatrača. Naime u sva tri primjera regije nisu bile potpuno točno segmentirane ali nije bilo moguće kvantificirati u kojoj mjeri rezultat odstupa od potpuno točne segmentacije. Problem klasifikacije riješen je pomoću algoritma za fraktalnu analizu tekstura. Algoritam se pokazao uspješnim u klasifikaciji bilo koje slike koja se već nalazila u uzorku ali i onih slika koje su po svojim značajkama bile bliske. U slučajevima kada je bilo potrebno klasificirati teksturu koja tematski odgovara jednoj od 25 klasa ali se po značajkama znatno razlikuje (npr. valovi i mirna voda) algoritam nije uspješno proveo klasifikaciju. To bi se moglo popraviti povećanjem uzorka za treniranje novim slikama tekstura.

LITERATURA

[1] C. H. Chen, L. F. Pau, P. S. P. Wang, The Handbook of Pattern Recognition and Computer Vision (2nd Edition), World Scientific Publishing Co., str. 207.-248., Singapur, 1998.
Izvor: http://slipguru.disi.unige.it/readinggroup/papers_vis/textureanalysis_jain.pdf; pristup ostvaren 1.4.2016.

[2] Vaijinath V. Bhosle, Vrushsen P. Pawar, Texture Segmentation: Different Methods, International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, br. 5, sv. 3, str. 69.-74., Studeni 2013.
Izvor: http://www.ijsce.org/attachments/File/v3i5/E1893113513.pdf; pristup ostvaren 1.4.2016.

[3] T. Ojala, M. Pietikäinen, Texture classification, Machine Vision and Media Processing Unit University of Oulu Finland
Izvor: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OJALA1/texclas.htm; pristup ostvaren 1.4.2016.

[4] Matija Forko, Postupci sinteze volumnih tekstura i teksturiranja objekata, Diplomski rad, ZEMRIS, FER, Zagreb, Lipanj 2011.
Izvor: http://www.zemris.fer.hr/predmeti/ra/Magisterij/11_Forko/Files/Diplomski_Matija_Forko.pdf; pristup ostvaren 5.4.2016.

[5] J. Malik, R. Rosenholtz, Computing Local Surface Orientation and Shape from Texture for Curved Surfaces, International Journal of Computer Vision, Kluwer Academic Publishers, str. 149.-168., Nizozemska, 1997.
Izvor: http://people.eecs.berkeley.edu/~malik/papers/malik-rosen97.pdf; pristup ostvaren 5.4.2016.

[6] A. Materka, M. Strzelecki, Texture Analysis Methods – A Review, Technical University of Lodz, Institute of Electronics, Bruxelles 1998.
Izvor: http://www.eletel.p.lodz.pl/programy/cost/pdf_1.pdf; pristup ostvaren 8.4.2016.

[7] Krunoslav Dropučić, Vedran Sabadoš, Iva Mandić, Primjena Markovljevih polja u obradi slike, Projektni zadatak, FER, Zagreb, Siječanj 2012.
Izvor: https://www.fer.unizg.hr/_download/repository/Primjena_Markovljevih_polja_u_obradi_slike.pdf; pristup ostvaren 20.4.2016.

[8] Fractal, Wikipedia članak
Izvor: https://en.wikipedia.org/wiki/Fractal; pristup ostvaren 20.4.2016.

[9] V. S. N. Prasad, J. Domke, Gabor Filter Vizualization, Materijali za predavanje, University of Maryland, Maryland, Ožujak 2005.

Izvor: https://wwwold.cs.umd.edu/class/spring2005/cmsc838s/assignment-projects/gabor-filter-visualization/report.pdf; pristup ostvaren 10.5.2016.

[10] Gabor, MATLAB dokumentacija

Izvor: http://www.mathworks.com/help/images/ref/gabor.html?searchHighlight=gabor; pristup ostvaren 10.5.2016.

[11] Texture Segmentation Using Gabor Filters, MATLAB dokumentacija

Izvor: http://www.mathworks.com/help/images/texture-segmentation-using-gabor-filters.html; pristup ostvaren 10.5.2016.

[12] k-means clustering, Wikipedia članak

Izvor: https://en.wikipedia.org/wiki/K-means_clustering; pristup ostvaren 10.5.2016.

[13] A. F. Costa, An Efficient Algorithm for Fractal Analysis of Textures, SIBGRAPI Conference on Graphics, Patterns and Images, IEEE, str. 39.-46., Brazil, Kolovoz 2012.

Izvor: https://drive.google.com/file/d/0ByOACbBQiCq4Ymh6WnJsMWhaUWs/edit; pristup ostvaren 15.5.2016.

[14] S. Lazebnik, C. Schmid, J. Ponce, A Sparse Texture Representation Using Local Affine Regions, IEEE Transactions on Pattern Analysis and Machine Intelligence, br. 8, sv. 27, str. 1265.-1278., Kolovoz 2005.

Izvor: http://www-cvr.ai.uiuc.edu/ponce_grp/data/; pristup ostvaren 15.5.2016.

[15] Naive Bayes classifier, Wikipedia članak

Izvor: https://en.wikipedia.org/wiki/Naive_Bayes_classifier; pristup ostvaren 15.5.2016.

POPIS SLIKA

Slika 2.1. Primjer tekstura dobivenih iz stvarnih objekata (a) šljunak (b) voda, str. 2.

Izvor: (a) http://thumbs.dreamstime.com/x/gravel-14927545.jpg); pristup ostvaren 7.4.2016.
(b) http://textures101.com/textures/Water/Plain/2011/6/12/tn1_water512_prekf.jpg; pristup ostvaren 7.4.2016.

Slika 2.2. Primjer umjetnih tekstura, str. 3.

Izvor: https://www.fer.unizg.hr/_download/repository/08AnalizaTekstura.pdf; pristup ostvaren 8.4.2016.

Slika 2.3. Spektar tekstura, str. 3.

Izvor: https://upload.wikimedia.org/wikipedia/commons/0/02/Texture_spectrum.jpg; pristup ostvaren 8.4.2016.

Slika 2.4. Parovi tekstura jednake statistike drugog reda, str. 4.

Izvor: http://slipguru.disi.unige.it/readinggroup/papers_vis/textureanalysis_jain.pdf; pristup ostvaren 8.4.2016.

Slika 2.5. Segmentacija, str. 7.

Izvor: http://slipguru.disi.unige.it/readinggroup/papers_vis/textureanalysis_jain.pdf; pristup ostvaren 8.4.2016.

Slika 2.6. Primjer klasifikacije uzorka teksture trave, str. 8.

Izvor: http://www.robots.ox.ac.uk/~vgg/research/texclass/figs/intro.jpg; pristup ostvaren 10.4.2016.

Slika 2.7. Prikaz sinteze, str. 9.

Izvor: (a) http://image.slidesharecdn.com/fasttexturesynthesisusingtree-structuredvectorquantization-121127142238-phpapp01/95/fast-texture-synthesis-using-tree-structured-vector-quantization-2-638.jpg; pristup ostvaren 10.4.2016.
(b) http://research.microsoft.com/en-us/um/people/yuedong/project/lazysyn/teaser.png; pristup ostvaren 10.4.2016.

Slika 2.8. Dojam trodimenzionalnosti, str. 9.

Izvor: http://jov.arvojournals.org/data/Journals/JOV/932796/jov-10-5-17-fig009.jpeg; pristup ostvaren 10.4.2016.

Slika 3.1. Graf slučajnog Markovljeva polja, str. 17.

Izvor: https://upload.wikimedia.org/wikipedia/en/f/f7/Markov_random_field_example.png; pristup ostvaren 20.4.2016.

Slika 3.2. Najčešći oblici zonalnih maski, str. 19.

Izvor: https://www.fer.unizg.hr/_download/repository/04EkstrakcijaZnacajki.pdf; pristup ostvaren 20.4.2016.

Slika 4.1. Vizualizacija Gaborovog filtera, str. 20.

Izvor: https://wwwold.cs.umd.edu/class/spring2005/cmsc838s/assignment-projects/gabor-filter-visualization/report.pdf; pristup ostvaren 10.5.2016.

Slika 4.2. Proces segmentacije slike, str. 21.

Izvor: http://watnow.uwaterloo.ca/pub/hammouda/sd775-paper.pdf; pristup ostvaren 10.5.2016.

Slika 4.3. Aditivno miješanje osnovnih boja, str. 22.

Izvor: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/AdditiveColor.svg/330px-AdditiveColor.svg.png; pristup ostvaren 10.5.2016.

Slika 4.4. Ulazne slike u smanjenom mjerilu koje je bilo potrebno segmentirati, str. 23.

Izvor: (a) https://skittledip.files.wordpress.com/2012/03/img_1354.jpg; pristup ostvaren 10.5.2016.
(b) https://cms-assets.tutsplus.com/uploads/users/166/posts/25556/image/Step_05_2.jpg; pristup ostvaren 10.5.2016.
(c) http://www.ux.uis.no/~tranden/brodatz/D101.gif, http://www.ux.uis.no/~tranden/brodatz/D94.gif; pristup ostvaren 10.5.2016.

Slika 4.5. Prikaz značajki Gaborovih magnituda pomoću PCA metode, str. 27.

Izvor: MATLAB

Slika 4.6. Slikovni prikaz rada kmeans algoritma, str. 28.

Izvor: https://en.wikipedia.org/wiki/K-means_clustering; pristup ostvaren 12.5.2016.

Slika 4.7. Vizualizacija segmentacije, str. 29.

Izvor: MATLAB

Slika 4.8. Usporedni prikaz ulazne slike i rezultata segmentacije, str. 30.

Izvor: MATLAB

Slika 4.9. Prikaz rezultata segmentacije kompleksnog uzorka od 16 različitih tekstura, str. 31.

Izvor: (a) Brodatzov album; pristup ostvaren 12.5.2016.
(b), (c), (d) MATLAB

Slika 4.10. Ekstrakcija značajki, str. 32.

Izvor: https://drive.google.com/file/d/0ByOACbBQiCq4Ymh6WnJsMWhaUWs/edit; pristup ostvaren 15.5.2016.

Slika 4.11. Karirani uzorak, str. 36.

Izvor: http://www-cvr.ai.uiuc.edu/ponce_grp/data/; pristup ostvaren 15.5.2016.

Slika 4.12. Rezultat klasifikacije u MATLAB-u, str. 36.

Izvor: MATLAB

SAŽETAK

Analiza tekstura u slici

U ovome radu opisani su osnovni pojmovi analize tekstura u slici te četiri glavna problema kod obrade i analize tekstura, klasifikacija, segmentacija, sinteza i dobivanje oblika iz teksture. Teorijski su objašnjene metode za kvantitativni opis teksture, strukturna metoda, statističke metode, metode koje se temelji na modelu i transformacijske metode. U eksperimentalnom dijelu rada riješena su dva problema pomoću MATLAB-a, prvi se zasniva na segmentaciji, a drugi na klasifikaciji. Segmentacija je riješena metodom Gaborovih filtera i grupiranjem algoritmom *k*-srednjih vrijednosti, a klasifikacija algoritmom za fraktalnu analizu tekstura i *naivnim* Bayesovim klasifikatorom. Provedeni eksperimenti dali su zadovoljavajuće rezultate.

Ključne riječi: teksture, klasifikacija, segmentacija, sinteza, dobivanje oblika iz teksture, regije, metode, Gaborov filter, klasterizacija, *kmeans*, fraktal, Bayes

ABSTRACT

Texture analysis in the image

This Bachelor thesis describes the basic concepts of texture analysis in the image and four major problems in processing and analysis of textures, classification, segmentation, synthesis and obtaining shape from texture. It also explains methods for quantitative description of textures, structural method, statistical methods, model based methods and transform methods. The experimental part of the paper contains solution for two problems made in MATLAB, the first is based on segmentation and the other on classification. Segmentation is solved by the method of Gabor filters and k-means clustering, while classification is solved by segmentation-based fractal texture analysis algorithm and *naive* Bayes classifier. Conducted experiments gave satisfactory results.

Keywords: textures, classification, segmentation, synthesis, shape from texture, regions, methods, Gabor filter, clustering, *kmeans*, fractal, Bayes

ŽIVOTOPIS

Hrvoje Ronta rođen je 9. listopada 1987. u Osijeku. Nakon završene Opće gimnazije u Donjem Miholjcu 2006. godine upisao je preddiplomski sveučilišni studij računarstva na Elektrotehničkom fakultetu u Osijeku koji privodi kraju.

Potpis: _____

PRILOZI

Na DVD-R mediju se nalaze slike i algoritmi koji su poslužili za izradu eksperimentalnog dijela rada.