

Univerzitet u Sarajevu
Elektrotehnički fakultet u Sarajevu
Odsjek za telekomunikacije

Sistem računarske vizije za analizu rukopisa

**Završni rad
I ciklusa studija**

Mentor:

Red. prof. dr Melita Ahić-Đokić

Sarajevo, 2015.

Kandidat:

Nermina Ahmić

**Red. prof. dr Melita Ahić-Đokić, dipl. el. inž.
Viši asistent mr Emir Sokić, dipl. el. inž.
Odsjek za telekomunikacije**

Tema za završni rad

studenta I ciklusa studija koji studira na ETF-u u skladu sa principima Bolonjskog procesa
na Odsjeku za telekomunikacije (šk. 2014/15)

Tema: Sistem računarske vizije za analizu rukopisa

Student: Ahmić Nermina

Sažetak:

Svaka osoba ima svoj stil pisanja i različit rukopis. Analiza rukopisa može dati mnogo informacija o osobi koja je nešto pisala (spol, dob, psihičko stanje i sl.), i može pomoći pri rješavanju tehničkih zadataka kao što su: verifikacija korisnika, digitalna forenzika itd.

Problem pisane poruke je taj što ona predstavlja statičku informaciju o rukopisu i parametri kao što su pritisak olovke na podlogu, brzina pisanja, kretanje olovke iznad teksta i sl. ne mogu biti jednostavno određeni.

Zadatak ovog završnog rada je razviti algoritam koji omogućava djelimičnu:

- ekstrakciju dinamičkog zapisa teksta iz skeniranih dokumenata,
- ekstrakciju parametara (pritisak, brzina pisanja i sl.)
- ekstrapolaciju 3D kretanja vrha olovke korištenjem analiziranog rukopisa,
- verifikaciju korisnika na osnovu rukopisa.

Polazna literatura:

- [1] Milan Sonka, Vaclav Hlavac, Roger Boyle: „Image Processing, Analysis, and Machine Vision“, Thomson, 2008.
- [2] Edward A. Lee, Pravin Varaiya: „Structure and Interpretation of Signals and Systems“, Electrical Engineering & Computer Science University of California, Berkeley, 2000.
- [3] R. Gonzalez, R. E. Woods, S. L. Eddins: „Digital Image Processing using MATLAB“, PEARSON Prentice Hall, 2009.
- [4] L. da Costa, R. M. Cesar Jr.: „Shape Analysis and Classification: Theory and Practice“, CRC Press LLC, 2001.
- [5] S. Hamid, K. Loewenthal, “Inferring Gender from Handwriting in Urdu and English”, The Journal of Social Psychology, vol. 136, pp. 778–782, 1996.
- [6] V. Burr, “Judging Gender from Samples of Adult Handwriting: Accuracy and Use of Cues”, The Journal of Social Psychology, vol. 142, pp. 691–700, 2002.
- [7] S. N. Srihari, S. Cha, H. Arora, S. Lee, “Individuality of Handwriting”, J Forensic Sci, vol. 47, pp. 856–72, 2002.
- [8] Y. Mingqiang, K. Kidiyo and R. Joseph, “A Survey of Shape Feature Extraction Techniques”, Pattern Recognition, Peng-Yeng Yin (Ed.), pp. 43–90, 2008.
- [9] S. Loncaric, “A Survey of Shape Analysis Techniques”, Pattern Recognition, vol. 31, pp. 983–1001, 1998.

- [10] B. Zhang, S. N. Srihari, "Analysis of Handwriting Individuality Using Word Features", 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, pp. 1142–1146, 2003.
- [11] M. Liwicki, A. Schlapbach, P. Loretan, and H. Bunke, "Automatic Detection of Gender and Handedness from On-line Handwriting", Proc. 13th Conf. of the Int. Graphonomics Society, pp. 179–183, 2007.
- [12] D. Zhang and G. Lu, "Content-based Shape Retrieval Using Different Shape Descriptors: A Comparative Study", pp. 1139–1142, 2001.
- [13] S. Impedovo, B. Marangelli, A. Fanelli, "A Fourier Descriptor Set for Recognizing Nonstylized Numerals", IEEE Transactions on systems, man and cybernetics, vol. 8, pp. 640–645, 1978.

Mentor:

Red. prof. dr Melita Ahić-Đokić

Sažetak

Rukopis, kao jedinstvena odrednica svake osobe, čest je predmet analize u svrhu otkrivanja karakternih osobina iste. Mogućnosti koje nudi savremena tehnologija doprinose tome da se sve češće podliježe provođenju tih analiza računarskim putem. Cilj ovog rada je formiranje sistema za računarsku viziju rukopisa, što prvenstveno obuhvata ekstrakciju dinamičkog zapisa teksta iz skeniranih dokumenata i ekstrapolaciju 3D kretanja vrha olovke korištenjem analiziranog rukopisa. Analiza je provedena korištenjem programskog paketa MATLAB, u kom su implementirane metode za pretpresisiranje i segmentaciju slike, te ekstrakciju relevantnih parametara pomoću funkcije cilja, čiji se proračun bazira na odrednicama latiničnog pisma. Pokazano je da se na ovaj način može doći do traženih rezultata, zbog čega ovaj rad može biti polazna osnova za dalju analizu.

Abstract

Individuality of handwriting is the reason why handwriting is common base element for analysis with the purpose of detecting person's character traits. Modern technology offers the benefit of computational analysis, which makes the process much easier. The goal of this work is creating a system for computer vision of handwritten text, which includes extraction of dynamic text entry from scanned documents and 3D extrapolation of pen tip, using the analyzed text. The analysis was performed in MATLAB program package, where the methods of preprocessing, segmentation and extraction of the relevant parameters were implemented. This process included computing the value of the objective function, based on Latin alphabet features. It is shown that this method leads to the desired results, so this paper can be used as a basis for further studies.

Sadržaj

Sadržaj	4
1. Uvod.....	5
1.1. Grafologija.....	6
1.2. Računarska vizija digitalne slike	8
2. Priprema slike za ekstrakciju parametara	10
2.1. Preprocesiranje slike	11
2.1.1. Normalizacija podataka	12
2.1.2. Odstranjivanje šuma.....	13
2.1.3. Morfološko procesiranje slike.....	17
2.2. Segmentacija podataka.....	20
2.2.1. Određivanje praga.....	21
2.2.2. Detekcija tačaka, linija i rubova	22
3. Praktična realizacija.....	25
3.1. Preprocesiranje slike	25
3.2. Segmentacija podataka.....	27
3.3. Ekstrakcija parametara.....	29
4. Upute za korištenje napisanih funkcija.....	32
FUNKCIJA 1: Funkcija za analizu rukopisa i ekstrapolaciju 3D kretanja vrha olovke.....	32
FUNKCIJA 2: Funkcija cilja	34
FUNKCIJA 3: Funkcija za normalizaciju	36
5. Zaključak	37
6. Lista funkcija.....	38
7. Lista slika	40
8. Lista tabela.....	42
9. Literatura	43

POGLAVLJE 1

Uvod

Brz razvoj tehnologije doveo je do pojave računarskih algoritama koji nude efikasna rješenja za probleme iz različitih životnih sfera. Kako je još od davnina postojala ideja da se može doći do određenih zaključaka o čovjeku analizom izgleda njegovog rukopisa, tokom historije se ta ideja razvijala, te je začeta nauka čiji je zadatak upravo provođenje ove analize, a koja je poznata pod nazivom grafologija [1]. Kao i većina drugih nauka, i sama grafologija je pronašla mogućnost povezivanja sa informatikom, u cilju dobivanja što bržih i preciznijih odgovora na pitanja od interesa.

Iza sistema koji bi omogućio analizu rukopisa i ponudio tražene rezultate stoji složen proces računarske vizualizacije digitalne slike, što podrazumijeva da se povezivanje rukopisa sa računarom izvodi skeniranjem ili fotografiranjem pisanog teksta. Na taj način dobiva se sirovi podatak, koji je potom potrebno obraditi kako bi se dobili parametri od značaja za dalju analizu. Parametri na osnovu kojih se rukopis analizira su zapravo one karakteristike rukopisa koje su uočljive svakom posmatraču, a na osnovu kojih bi on mogao pretpostaviti dob ili spol osobe čiji rukopis posmatra.

Rukopis svake osobe je jedinstven i rezultat je podsvijesti ili navike, baš kao i ostali pokreti tijela. Pokreti ruke koji kreiraju rukopis omogućavaju prenošenje električnih impulsa iz mozga na papir, na taj način odajući informacije o psihičkom stanju, emocijama i brojnim osobinama osobe koja piše. Pokazano je da se više uzoraka rukopisa iste osobe međusobno razlikuje, iako su te razlike manje u odnosu na slučaj kada se porede rukopisi različitih osoba [2]. Uprkos jedinstvenosti rukopisa, postoje odrednice na osnovu kojih je moguće izvršiti procjenu da li je posmatrani tekst napisalo dijete ili odrasla osoba, ljevak ili dešnjak, osoba muškog ili ženskog spola i sl. Cilj sistema za računarsku analizu rukopisa je izdvajanje karakterističnih parametara i njihova usporedba sa dosadašnjim spoznajama – ženski rukopis je uredniji od muškog, slova su veća i imaju više zaobljenja, u rukopisu muškarca se može uočiti veći pritisak olovke na papir i samopouzdanje pri pisanju, ali najčešće i veća brzina pisanja u odnosu na žene [3]. Također, ljevaci i dešnjaci neka slova, poput slova *o*, pišu u različitom smjeru [4], a postoji i čitav niz drugih prepoznatljivih odrednica u pisanom tekstu, o kojima će biti više riječi u nastavku rada.

Cilj ovog poglavlja je prezentiranje osnovnih zadataka grafologije i principa računarske vizije digitalne slike, koji će poslužiti kao uvod u dalju razradu teme i formiranje sistema računarske vizije za analizu rukopisa.

1.1. Grafologija

Grafologija je nauka koja se bavi izučavanjem rukopisa u cilju procjene karakternih osobina osobe čiji se rukopis analizira [1]. Zahvaljujući rezultatima koje nudi, ova nauka primijenjena je u psihologiji, medicini, forenzici, ali i u marketingu. Najčešće analize koje se vrše su analize oproštajnih pisama osoba koje su počinile samoubistvo, te analize potpisa. U medicinske svrhe se grafologija koristi kao pomoć pri dijagnosticiranju i praćenju razvoja bolesti kao što su Parkinsonova i Alzheimerova bolest, pa čak i rak [5]. Interesantni su rezultati do kojih su grafolozi došli kada je u pitanju područje marketinga. Naime, izvršena je analiza rukopisa zaposlenika, u cilju testiranja njihove iskrenosti, a niti u jednoj od 18 kompanija u kojima je analiza provedena, nijedna iskrena osoba nije klasificirana kao neiskrena [1]. Ovakvi rezultati veliki su poticaj da se grafologija još više primjenjuje kao pomoćna disciplina u različitim naukama.

Postoji nekoliko teoretskih pretpostavki na kojima je grafologija zasnovana:

- rukopis nije samo proizvod jednostavnog pokreta ruke, već je aktivnost koja je rezultat podsvijesti i osobnosti,
- iako su svi ljudi koji razumiju neko pismo (latinično, cirilično, arapsko i sl.) učili pisati slova na isti način, svaki je rukopis jedinstven i na taj način predstavlja individualno svojstvo osobe,
- promjene u rukopisu aludiraju na promjene u osobnosti,
- kontekst u kom se neka riječ ili slovo koristi nije od interesa, već se posmatra samo način na koji je tekst napisan,
- od interesa nisu ni pojedinačni znakovi ili slova, već kombinacija slova unutar jedne cjeline, koja se posmatra kao takva,
- rukopis osobe zavisi od njene dobi i učestalosti pisanja,
- s obzirom na to da potezi u rukopisu imaju različit intenzitet i frekvenciju, razmatranjem samo jednog uzorka rukopisa, dobiveni rezultati nemaju istu vrijednost kao rezultati koji se dobiju analizom više uzoraka rukopisa iste osobe,
- postoji prirodna granica varijabilnosti poteza u rukopisu jedne osobe, osim u specijalnim, uglavnom psihopatskim slučajevima [1].

Uzimajući u obzir prethodno pobrojane pretpostavke, grafolozi vrše analizu uzoraka rukopisa, koji bi, radi što preciznijih i tačnijih rezultata, u idealnom slučaju trebali biti sastavljeni od pedeset do stotinu riječi, u kojima bi bila sadržana i velika i mala slova, te većina slova alfabeta. Osim toga, pokazano je da se više informacija može dobiti ukoliko je tekst pisan na bijelom papiru bez linija, jer se na taj način, u zavisnosti od nagiba pisanog teksta, također dolazi do zaključaka o osobnosti i emocionalnom stanju osobe.

Uprkos tome što jednostavna analiza, koju podrazumijeva grafologija, pruža mogućnost procjene širokog opsega karakternih osobina, od kojih su neke prikazane u tabeli 1.1., postoje određeni nedostaci i problemi koji se javljaju u grafologiji. Pouzdanost i validnost grafoloških rezultata najveći je problem s kojim se grafolozi susreću, a javlja se kao posljedica nepostojanja materijalnih dokaza izvedenih zaključaka. Osim toga, potreban je pristanak osobe na podvrgavanje analizi rukopisa, a pred grafologa se stavlja, često neugodan, zadatak da sve rezultate analize podijeli sa osobom kojoj pripadaju.

Svojstvo		Značenje	Svojstvo		Značenje
Veličina slova	Mala	Sposobnost koncentracije	Prekidi u pisanju	Povezanost	Objektivnost, racionalnost, logičnost
	Velika	Ambicioznost, dalekovidnost		Nepovezanost	Intuitivnost, osjetljivost, nesigurnost
	Srednje veličine	Samopouzdanost, tradicionalnost, realnost			
	Promjenjive veličine	Neodlučnost, brze promjene raspoloženja			
Osnovna linija pisanja	/	Pesimizam, obeshrabrenost	Brzina pisanja	Brzo	Pamet, nekomunikativnost
	\	Optimizam, vjera u budućnost		Sporo	Lijenost, nespretnost, neiskrenost
	—	Ujednačen temperament			
Pritisak na olovku	Blag	Neodlučnost, sposobnost brzog praštanja	Razmak između riječi	Veoma širok	Odbojenost od stvarnosti
	Srednje jačine	Prosječan nivo emocionalnosti		Uzak	Štedljivost
	Jak	Dugotrajno pamćenje negativnih događaja		Širok	Mentalna spretnost, objektivnost
	Promjenjiv	Promjenjiva narav		Ujednačen	Ustrajnost, sistematično planiranje
Nagib	Udesno (BC)	Sklonost procjenjivanju	Margine	Veliki razmak s lijeve strane	Hrabrost u suočavanju sa životom
	Ulijevo (FA)	Opreznost, introvertnost		Veliki razmak s desne strane	Izbjegavanje budućnosti, rezervisanost
	Vertikalno (AB)	Razum upravlja emocijama		Nema margina	Nesigurnost, pričljivost
	Promjenjiv	Brze promjene raspoloženja, nepredvidivost		Veliki razmak od početka papira	Formalnost, povučenost
	Udesno (CD)	Ekstrovertnost, orijentiranost ka budućnosti		Veliki razmak do kraja papira	Suzdržanost, površnost
	Udesno (DE)	Duboko proživljavanje svih situacija		Ujednačene	Samodiscipliniranost, samosvjesnost
	Udesno (E+)	Velika ekspresivnost			

Tabela 1.1: Osobine rukopisa i objašnjenja pripadajućih karakternih osobina.¹¹ Tabela preuzeta iz [5].

1.2. Računarska vizija digitalne slike

Ljudsko oko je složen organ koji obezbjeđuje razumijevanje slike u mozgu pretvaranjem svjetlosti u nervne impulse. Da bi se upustili u detaljnije izučavanje računarske vizije digitalne slike, potrebno je razumjeti način na koji ljudi percipiraju ono što vide i napraviti razliku između ljudske i računarske vizije.

Prepoznavanje objekta, čak i u slučaju kada on nije potpuno u vidnom opsegu, jedna je od najznačajnijih mogućnosti ljudskog oka i mozga. Zahvaljujući spoznajama koje imamo i stečenom iskustvu, na osnovu minimalnog broja raspoloživih podataka možemo prepoznati neki oblik, kao što je naprimjer geometrijski lik kome vidimo samo rubove. S druge strane, procjene različitih vrijednosti, kao što je intenzitet boje, nisu precizne, zbog čega se one najčešće vrše računarskim putem. Boja sivog kruga na podlozi tamnije sive nijanse i istog kruga na podlozi svjetlijе sive nijanse procijenit će se na različite načine, što predstavlja samo optičku varku koju je oku teško opaziti. Ako se analiziraju iste slike na računaru, do greške u procjeni nijanse boje nikada neće doći. Međutim, uprkos određenim prednostima koje računarska vizija slike ima nad ljudskom percepcijom iste, računari su još uvijek daleko od mogućnosti poimanja slike na onaj način na koji je ljudi poimaju.

Računarska vizija digitalne slike nije jednostavna, a kao uzrok tome javlja se nekoliko faktora: gubitak informacije zbog prelaska iz trodimenzionalnog u dvodimenzionalni prostor, šum, previše ulaznih podataka, izmjereno osvjetljenje, te potreba za globalnim razumijevanjem [6].

Gubitak informacije koji se javlja kao posljedica prelaska iz trodimenzionalnog prostora u prostor sa dvije dimenzije čest je problem, koji se ne javlja samo u slučaju računarske vizije slike, već i u samoj ljudskoj percepciji trodimenzionalnog objekta koji posmatra. Upravo iz ovog razloga, ljudi nemaju mogućnost precizne procjene udaljenosti na kojoj se nalazi objekat, ali ovaj problem znatno je izraženiji kada je računarska procjena udaljenosti u pitanju. Ako se analizira fotografija objekta malih dimenzija, koji je u zabilježenom trenutku bio blizu fotoaparata, te fotografija objekta istog oblika, ali većih dimenzija, pozicioniranog na proporcionalno velikoj udaljenosti od fotoaparata, računar će ova objekta „vidjeti“ isto, uslijed nedostatka informacije o trećoj dimenziji.

Prisustvo šuma na slikama je neizbjježno, jer se šum pojavljuje u svim realnim sistemima, što dodatno otežava računarsku analizu slike i može dovesti do oštećenja, ili gubitka dijela sadržanih podataka. Da bi se poboljšala vjerodostojnost analize, koriste se matematski alati kojima se nastoji otkloniti šum u što većoj mjeri.

Prevelika količina ulaznih podataka još jedan je od problema koji se javljaju kada je u pitanju računarska vizija. Stranica A4 formata, skenirana u crno-bijelom modu, pri 300 dpi (dots per inch) i sa 8 bita po pikselu, ima težinu od 8.5 MB (megabyte). Ovako velika slika zahtijeva i jako dug proces obrade.

Osvjetljenje slike zavisi od izvora svjetlosti, njegove pozicije i intenziteta, kao i od lokalne geometrije i fenomena poput refleksije. Svi ovi podaci nisu dostupni računaru, zbog čega je proces mjerjenja osvjetljenja složen.

Računar sliku vizualizira na isti način na koji oko vidi gledajući kroz ključaonicu. Algoritmi za analizu slike najčešće vrše analizu pojedinih sastavnih dijelova digitalne

slike – piksela i njihovih lokalnih „susjeda“. U tu svrhu se definiraju osnovne relacije među pikselima: udaljenost, mogućnost povezivanja i susjedni pikseli, te linearne i nelinearne operacije koje je moguće izvršiti nad pikselima [7]. Na taj način, teško je dobiti informacije na globalnom nivou [6].

Za opis digitalnih slika, koriste se matematski modeli. Dvodimenzionalni prikaz, koji omogućava računarsku vizualizaciju statične slike, podrazumijeva postojanje funkcije dvije promjenjive: x i y , koje odgovaraju prostornim koordinatama u ravni. Ukoliko u funkciji figurira i vrijeme t kao treća varijabla, omogućen je prikaz pokretnih slika. Od posebnog interesa za ovaj rad su statične slike, koje su obično predstavljene u vidu matrice, a prostornim koordinatama odgovaraju prirodni brojevi. Opseg vrijednosti koje mogu poprimati funkcije slike je ograničen. Po konvenciji, za slike u nijansama sive boje vrijedi da najnižoj vrijednosti odgovara crna boja, dok najvišoj odgovara bijela [6].

Osim slika u tonovima sive boje, koje su u literaturi označene terminom monohromatskih slika, razlikujemo i slike u boji. Svjetlost se na slikama u boji mjeri na način da se svakoj tački uređenog para prostornih koordinata (x, y) pridružuje uređena trojka (r, g, b) , pomoću koje je moguće predstaviti sve boje iz spektra. Oznake r , g i b odnose se na crvenu (eng. *red*), zelenu (eng. *green*) i plavu (eng. *blue*) boju, respektivno [8].

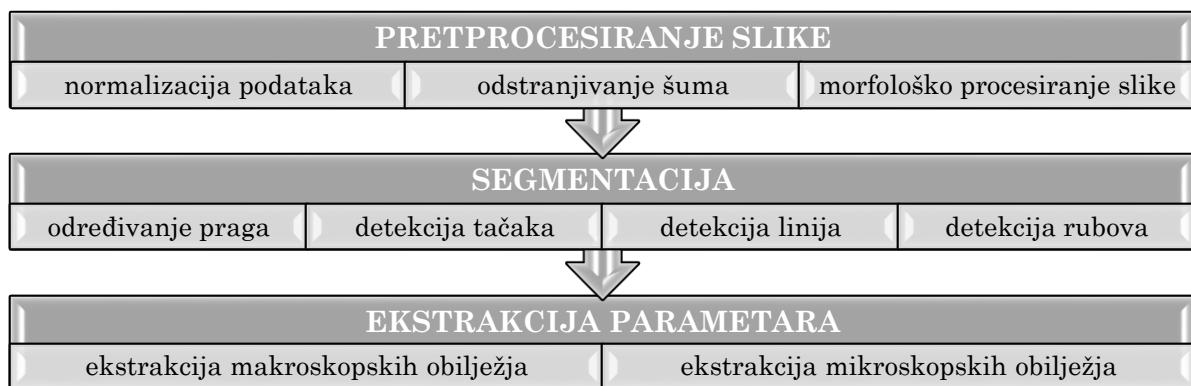
Prije nego što postupak računarske analize slike može započeti, potrebno je izvršiti digitalizaciju kontinualne slike. Proces digitalizacije zahtijeva prvobitno uzorkovanje i kvantizaciju. Uzorkovanjem se funkcija $f(x, y)$, kojom je slika predstavljena, dijeli u matricu sa M redova i N kolona, a potom se svakom uzorku dodjeljuje cjelobrojna vrijednost u procesu kvantizacije. Aproksimacija slike je tim bolja što je više uzoraka i nivoa kvantizacije raspoloživo [6].

Uzimajući u obzir sve probleme koji se javljaju u procesu računarskog predstavljanja digitalne slike, može se doći do zaključka da je ovaj proces vrlo složen. U prethodnom tekstu navedeni su osnovni principi na kojima počiva računarska vizija slike, uz samo površan osvrt na kompleksnu proceduru koju računar provodi, a koja nije tematika ovog rada. Uvod u spomenute koncepte proveden je radi razumijevanja poglavlja koja slijede.

POGLAVLJE 2

Priprema slike za ekstrakciju parametara

Svi procesi u računarskoj obradi slike hijerarhijski su podijeljeni u tri kategorije. Razlikuju se procesi nižeg, srednjeg i višeg nivoa. Procesi nižeg nivoa često se u literaturi nazivaju preprocesiranjem, a obuhvataju operacije kao što su odstranjanje šuma, povećanje kontrasta i izoštravanje slike. Pod procesom srednjeg nivoa podrazumijeva se segmentacija, tj. podjela slike na oblasti ili objekte, prevođenje tih objekata u formu pogodnu za računarsku analizu i njihova klasifikacija. Procesi višeg nivoa imaju ulogu da, primjenom kognitivnih funkcija, skup klasificiranih objekata iz prethodnog stepena procesiranja povežu u smisao cijelinu [7], što se može postići ekstrakcijom relevantnih parametara. U slučaju analize pisanih teksta, parametri kao što su pritisak olovke na papir, nagib olovke pri pisanju, osnovna linija pisanja i veličina slova predstavljaju makroskopska obilježja čiju je ekstrakciju potrebno izvršiti. S druge strane, zaobljenost slova, razmak između slova i razmak između riječi spadaju u mikroskopske parametre pisanih teksta. *Slika 2.1* prikazuje prethodno pobrojane korake u procesu računarske obrade slike.

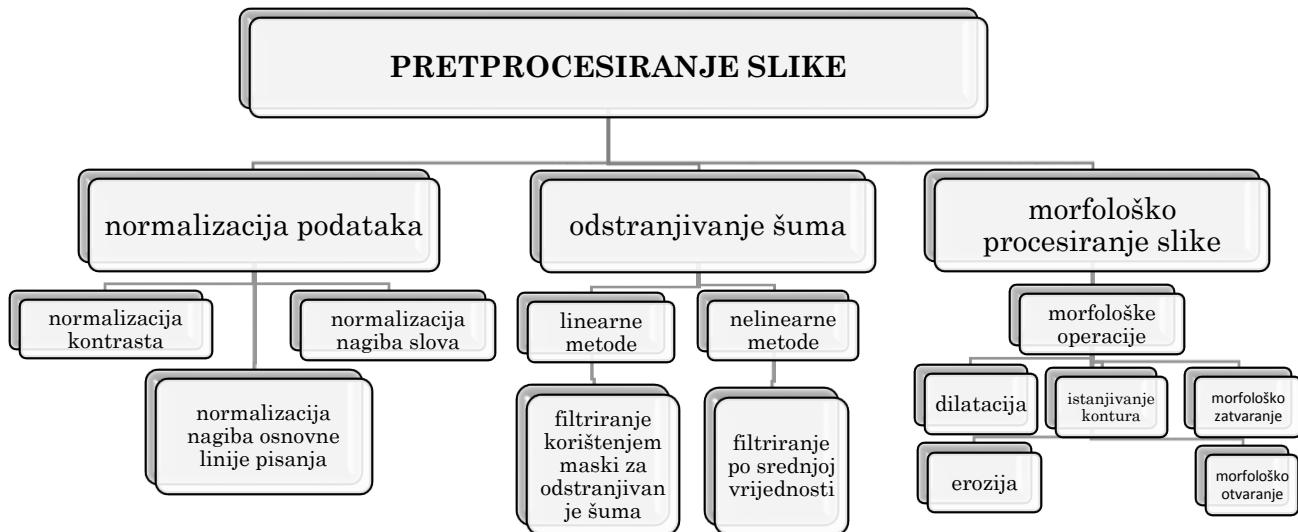


Slika 2.1: Koraci u procesu računarske obrade slike.

U ovom poglavlju će detaljnije biti obrađeni procesi nižeg i srednjeg nivoa, odnosno, preprocesiranje i segmentacija slike.

2.1. Preprocesiranje slike

Prvi korak u digitalnoj obradi slike je preprocesiranje. Ovaj postupak podrazumijeva provođenje niza različitih metoda, u zavisnosti od tipa podataka sadržanih u analiziranoj slici. Kada je u pitanju analiza slike pisanih teksta, jedna od varijanti kombinacije metoda koje je potrebno izvršiti prikazana je na *slici 2.2*, pri čemu je za proces odstranjivanja šuma dat izbor između linearnih i nelinearnih metoda.



Slika 2.2: Koraci u preprocesiranju slike pisanih teksta.

Odstranjanje šuma, normalizacija i kompresija podataka su uobičajeni postupci koje podrazumijeva preprocesiranje slike. Međutim, ovi pojmovi zahtijevaju opširniji opis i ne preciziraju operacije koje se vrše nad slikom. Sama operacija otklanjanja šuma, koja se najčešće vrši filtriranjem, zahtijeva izbor vrste filtra u zavisnosti od krajnjeg cilja (zagladivanje, izoštravanje, povećanje kontrasta ili odstranjanje blago teksturirane/obojene podloge), što jasno pokazuje da je skup operacija u preprocesiranju različit za različite sadržaje slike. Osim filtriranja, reduciranje nivoa šuma može se vršiti primjenom određenih morfoloških operacija ili metodom modeliranja šuma [9]. Pod pojmom morfoloških operacija podrazumijevaju se one operacije koje se izvode u svrhu izdvajanja osobina slike relevantnih za opis oblika objekta prikazanog na slici. Normalizacija podataka obično obuhvata metode rotacije slike ili određenih dijelova slike, kao i promjenu njihove veličine, dok se za kompresiju podataka može koristiti širok spektar morfoloških operacija, od kojih su osnovne dilatacija (eng. *dilation*) i erozija (eng. *erosion*). Dilatacija je operacija koja vrši podebljavanje objekata, a erozija podrazumijeva suprotan proces – istanjivanje objekata na slici [10]. Ove dvije operacije najčešće se kombiniraju u algoritmu koji vrši preprocesiranje slike, kako bi se postigao željeni rezultat.

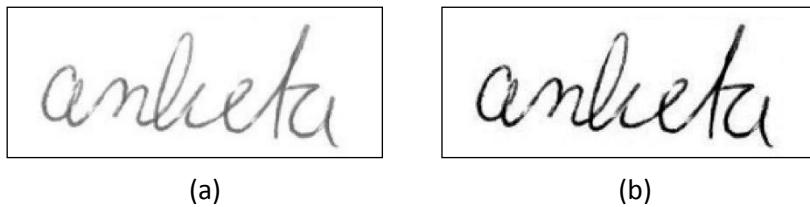
Potrebno je naglasiti da je preprocesiranje postupak koji kao ulazni parametar prima sliku, a kao izlazni parametar daje modificiranu verziju iste. Modifikacija se vrši radi prilagodbe slike za segmentaciju, koja predstavlja naredni korak u procesu digitalne obrade slike. U nastavku teksta, nešto detaljnije će se razmotriti koraci u preprocesiranju slike pisanih teksta, prikazani na *slici 2.2*.

2.1.1. Normalizacija podataka

Normalizacija podataka je proces koji se vrši radi otklanjanja varijacija u pisanju i dobivanja standardiziranih podataka [9]. Izvršavanje ovog koraka u preprocesiranju slike omogućava efikasnu primjenu algoritma za analizu rukopisa nad širim skupom slika. Nagib slova u rukopisu i osnovne linije pisanja, kao i pritisak na olovku bitni su podaci za klasifikaciju korisnika na osnovu rukopisa, ali se najčešće u algoritmima za analizu prvobitno vrši normalizacija kontrasta na slici i nagiba pisanog teksta, radi optimizacije metoda koje će se poslije primjenjivati. Treba napomenuti da ovim postupcima nisu nepovratno izgubljena prvobitna svojstva slike, potrebna za dalju analizu.

Normalizacija kontrasta

Naglašavanje korisnih informacija sadržanih u slici predstavlja važan segment preprocesiranja, koji olakšava dalju obradu slike. Proces raspoznavanja i odstranjivanja šuma znatno je jednostavniji ukoliko je na slici izražen kontrast. Stoga je korisno da algoritam za računarsku analizu rukopisa vrši njegovo podešavanje. Ako se analiza vrši nad slikom u sivim nijansama, povećanje kontrasta podrazumijeva povećanje vrijednosti svjetlih i istovremeno smanjenje vrijednosti tamnijih piksela. Vrijednosti pridružene pikselima kreću se u granicama od 0 do 255, kojima su predstavljene crna i bijela boja, respektivno. Primjer rezultata normalizacije kontrasta prikazan je na *slici 2.3*.

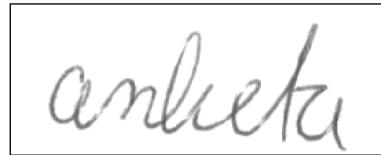


Slika 2.3: Normalizacija kontrasta; (a) originalna slika; (b) slika nakon povećanja kontrasta.

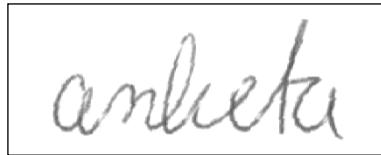
Normalizacija nagiba osnovne linije pisanja

Usljed skeniranja slike, ili zbog samog načina pisanja, rukopis može biti blago iskošen, što može utjecati na efikasnost daljeg rada algoritma za analizu rukopisa. Normalizacija nagiba osnovne linije pisanja zahtijeva računanje međusobne korelacije (kroskorelacija) pojedinih linija i korištenje Houghove transformacije. Ova transformacija predstavlja tehniku detekcije krivih, a bazira se na korištenju postojećeg dualiteta između tačaka krive i njenih parametara [9]. Primjenom Houghove transformacije vrši se prelazak iz prostora podataka u prostor modeliranja.

Slika 2.4 prikazuje rezultat korekcije nagiba osnovne linije pisanja.



(a)



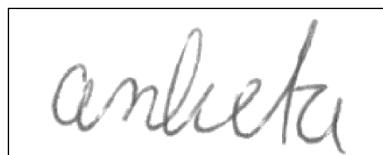
(b)

Slika 2.4: Normalizacija nagiba osnovne linije pisanja; (a) originalna slika; (b) slika nakon normalizacije osnovne linije pisanja.

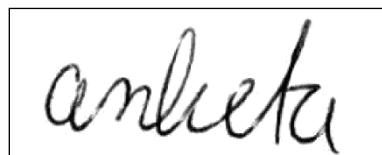
Normalizacija nagiba slova

Nagib slova u rukopisu je jedna od mjerljivih karakteristika pisanih teksta. Računanje nagiba svodi se na mjerjenje ugla između slova koja sadrže duže vertikalne linije (običajno su to slova: *b, d, f, h, k, l i t*, ali s obzirom na raznolikost stilova pisanja, ovo ne mora biti pravilo) i normale povučene na donju liniju slike. Međutim, kako se posmatraju riječi, a ne zasebna slova, proces normalizacije se usložnjava. Najčešće korišteni osnov po kom se vrši normalizacija je procjena prosječnog ugla nagiba slova računanjem aritmetičke sredine svih izmjerjenih uglova, i rotacija slova za taj ugao.

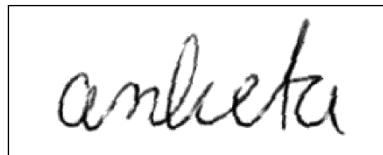
Kombiniranjem tri prethodno navedene metode normalizacije, dobivamo rezultate kao na slici 2.5.



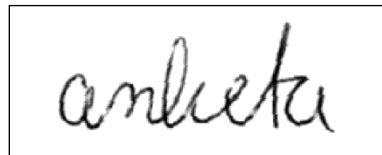
(a)



(b)



(c)



(d)

Slika 2.5: Normalizacija podataka; (a) originalna slika; (b) slika nakon povećanja kontrasta; (c) slika nakon normalizacije nagiba osnovne linije pisanja; (d) slika nakon normalizacije nagiba slova.



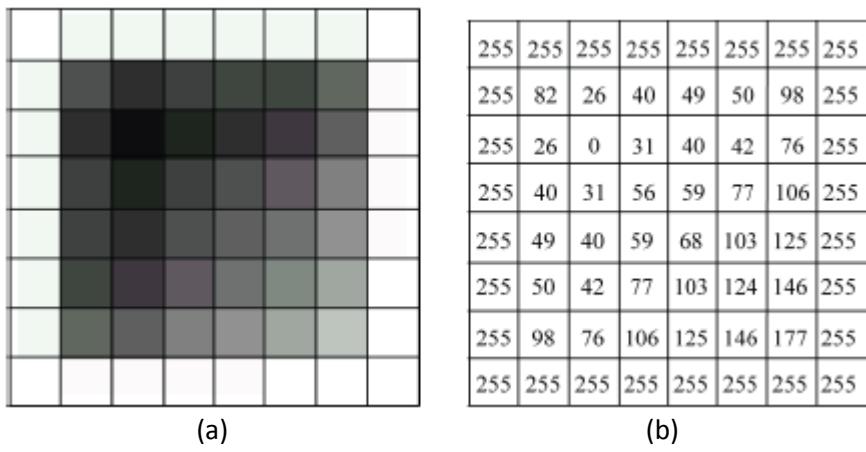
2.1.2. Odstranjanje šuma

Šum se na slikama pisanih teksta javlja kao posljedica skeniranja, a uzrokuje degradaciju kvaliteta podataka sadržanih u slici. Da bi se omogućila kvalitetna ekstrakcija relevantnih parametara, potrebno je potisnuti šum u što većoj mjeri. Tehnike

koje to omogućavaju dijele se u dvije grupe: linearne i nelinearne. Obje vrste tehnika najčešće primjenjuju statističko filtriranje u procesu otklanjanja šuma. Statističkim filtriranjem se, na osnovu mjere povezanosti (korelacije) posmatranog piksela sa okolnim pikselima, odstranjuju pikseli procijenjeni kao šum. Spomenuti pikseli imaju nizak nivo korelacije sa svojim susjedima. Vizuelno je moguće uočiti ovaj fenomen, jer pikseli šuma imaju uočljivo drugačiju nijansu boje u odnosu na susjede i obično su izolirani. Upravo na toj činjenici zasnovana je većina algoritama za odstranjivanje šuma.

Zaglađivanje slike

Zaglađivanje (eng. *smoothing*) slike je jedna od tehnika koja u velikoj mjeri omogućava potiskivanje šuma. Svakom pikselu slike je, prethodno izvršenim postupcima uzorkovanja i kvantizacije, dodijeljena odgovarajuća vrijednost, u zavisnosti od njegove svjetline (*slika 2.6*). Zahvaljujući tome, moguće je izvršiti usrednjavanje vrijednosti piksela koji se nalaze u proizvoljno velikoj okolini razmatranog, što je osnov postupka zaglađivanja slike. Nedostatak ove metode leži u tome da se osim šuma degradiraju i oštiri rubovi na slici, koji nose značajne informacije. Kako bi se spriječila velika degradacija, pri usrednjavanju se uzimaju u obzir samo one tačke koje imaju slične karakteristike kao posmatrana tačka. Treba napomenuti i da primjena metode zaglađivanja daje dobre rezultate samo ako slika sadrži šum u obliku tankih linija, ali je neprimjenjiva ako se šum na slici javlja u obliku većih grumena ili debljih linija. U tom slučaju, potrebno je primijeniti složeniju tehniku restauracije slike. [6]



Slika 2.6: (a) Slika izdijeljena na piksele kojima su postupcima uzorkovanja i kvantizacije dodijeljene odgovarajuće nijanse sive boje; (b) digitalni prikaz slike u obliku niza cijelih brojeva kojima je opisana boja svakog pojedinog piksela.²

Kada je dostupno više slika koje nose istu informaciju, tada se zaglađivanje slike može izvršiti bez njenog zamagljenja (eng. *blurring*) [6]. Ova mogućnost posljedica je osobina funkcije raspodjele srednjih vrijednosti, koja je predmet izučavanja statističke teorije. Naime, ako se na više dostupnih slika izvrši usrednjavanje vrijednosti koje odgovaraju istom skupu piksela, dobiva se raspodjela srednjih vrijednosti za posmatrani

² Slika preuzeta iz [12].

skup. Primjena statističkih spoznaja nad ovim skupom podataka kao rezultat daje sliku sa odstranjениm šumom i jako malom degradacijom korisnih informacija sadržanih u slici. Ova metoda korištena je pri analizi provedenoj u [13] i [14]. Nažalost, u većini slučajeva nije dostupno više nosioca istih informacionih sadržaja, pa tada zaglađivanje slike daje nešto lošije rezultate.

Kada je na raspolaganju samo jedna slika za analizu, dolazi do izražaja degradacija ivica uzrokovana primjenom metode zaglađivanja. U ovom slučaju, vrši se lokalno usrednjavanje, koje podrazumijeva traženje srednje vrijednosti samo onih piksela koji su u dodiru sa posmatranim, ili u njegovoj vrlo bliskoj okolini. Kada se vrši linearno odstranjivanje šuma u prostornom domenu, definiraju se tzv. maske za odstranjivanje

šuma, koje predstavljaju filtre u vidu matrica čije su dimenzije 3×3 . S obzirom na to da više frekvencije odgovaraju šumu, potrebno je izvršiti filtriranje kojim će biti propuštene samo komponente koje se nalaze na nižim frekvencijama. U tu svrhu, dizajniraju se niskopropusni filtri. Uvažavajući prethodno navedena ograničenja, maske za odstranjivanje šuma mogu se definirati kao matrice:

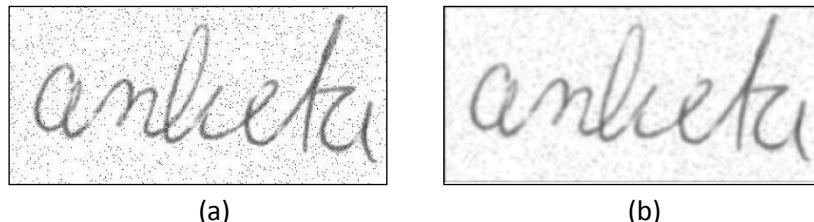
$$\text{- maska 1: } H_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad (2.1)$$

$$\text{- maska 2: } H_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad (2.2)$$

$$\text{- maska 3: } H_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \quad (2.3)$$

Proces filtriranja pomoću matrica H definiranih jednačinama (2.1)-(2.3) sastoji se u pomjeranju centra maske od tačke do tačke na slici koja se obrađuje. Odziv filtra u svakoj tački jednak je sumi proizvoda koeficijenata filtra sa odgovarajućim vrijednostima piksela iz područja obuhvaćenog maskom filtra [6]. Format matrice ne mora nužno biti 3×3 , već je dozvoljeno i njegovo povećanje. Međutim, što je veći format maske, to je veće i zamagljenje, a samim tim i gubitak informacija o detaljima koje slika sadrži [11]. Za masku dimenzija $m \times n$ obično je $m = 2a + 1$ i $n = 2b + 1$, gdje su a i b nenegativni cijeli brojevi. Dakle, uobičajeno je da su dimenzije maske neparni brojevi (isključujući trivijalni slučaj dimenzija 1×1), jer u tom slučaju maska ima jedinstvenu centralnu tačku.

Sukcesivna primjena maske zadane jednačinom (2.3) pokazala je najbolje performanse na skupu obrađivanih slika, a jedan od rezultata prikazan je na *slici 2.7*.

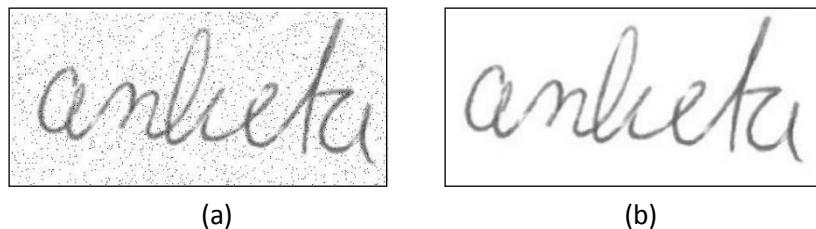


Slika 2.7: Primjena maske za odstranjivanje šuma; (a) originalna slika na koju je dodan tzv. „Salt and Pepper“ šum; (b) slika nakon filtriranja pomoću maske za odstranjivanje šuma.

Najčešće primjenjivana metoda iz skupine nelinearnih tehniki odstranjuvanja šuma je filtriranje po srednjoj vrijednosti (eng. *median filtering*). Naziv potječe iz statističke teorije, u kojoj se srednja linija (eng. *median*) definira kao linija koja dijeli raspodjelu vjerovatnoća na dva dijela, na taj način jasno razdvajajući vjerovatnoće viših i nižih vrijednosti. Za slučajnu varijablu x , srednja linija M je vrijednost za koju je vjerovatnoća da je $x < M$ jednaka 0,5. Kao i prethodno razmatrana linearna tehniki zamagljenja slike, i ova metoda počiva na ideji da se posmatrana tačka na slici, prepoznata kao šum, zamjeni pikselom čija je vrijednost jednak srednjoj vrijednosti piksela iz njegove okoline. Ipak, razlika između ove dvije metode postoji – degradacija ivica u procesu zamagljenja slike manja je ukoliko se primijeni filtriranje po srednjoj vrijednosti. Zahvaljujući ovoj osobini, metoda se može i iterativno ponavljati. [6]

Nedostatak filtriranja po srednjoj vrijednosti je degradacija vrlo tankih linija i oštih rubova na slici (iako u manjoj mjeri u odnosu na degradaciju koja se javlja primjenom linearnih tehniki). Ove neželjene posljedice mogu se izbjegći ukoliko se umjesto skupa piksela koji su na slici raspoređeni tako da čine pravougaonu strukturu, razmatra skup piksela u drugaćijem prostornom rasporedu (krug, elipsa i sl.).

Metoda filtriranja po srednjoj vrijednosti primijenjena nad istim skupom slika nad kojim je primjenjivano i filtriranje pomoću maske za odstranjuvanje šuma pokazala je znatno bolje performanse u odnosu na prvobitno izloženu tehniku. Rezultat je prikazan na *slici 2.8.*



Slika 2.8: Primjena filtriranja po srednjoj vrijednosti u svrhu odstranjuvanja šuma; (a) originalna slika na koju je dodan tzv. „Salt and Pepper“ šum; (b) slika nakon filtriranja po srednjoj vrijednosti.

Pored prethodno obrađenih tehniki za odstranjuvanje šuma postoje i brojne druge, od kojih treba izdvojiti homomorfno i *wavelet* filtriranje. Homomorfno filtriranje je linearne tehniki reduciranja nivoa šuma, koja se primjenjuje u slučaju postojanja multiplikativnih smetnji, dok *wavelet* filtriranje spada u nelinearne tehniki, zasnovane na istoimenoj transformaciji [11]. Spomenute tehniki neće biti detaljnije razmatrane u ovom poglavlju, jer su za provođenje ciljne analize metode filtriranja po srednjoj vrijednosti i pomoću maske za odstranjuvanje šuma dale dovoljno dobre rezultate.

2.1.3. Morfološko procesiranje slike

Morfološke operacije nad slikom vrše se u cilju njenog prilagođenja za dalje procesiranje. Termin „morfološke operacije“ podrazumijeva skup postupaka koji se vrše radi ekstrakcije karakterističnih parametara objekta prikazanog na slici, kao što su granice objekta i njegov skelet. U pretpresiranju i postprocesiranju slike, neke od morfoloških operacija koje se često koriste su filtriranje (eng. *filtering*), istanjivanje (eng. *thinning*) i izrezivanje (eng. *cropping*) [10]. Za proces analize rukopisa, najvažnije operacije obuhvaćene kompresijom podataka su dilatacija („podebljavanje“ objekta prikazanog na slici) i istanjivanje konture. Podrazumijeva se da je proces filtriranja izvršen kroz prethodne korake pretpresiranja – normalizaciju podataka i odstranjivanje šuma.

Dilatacija

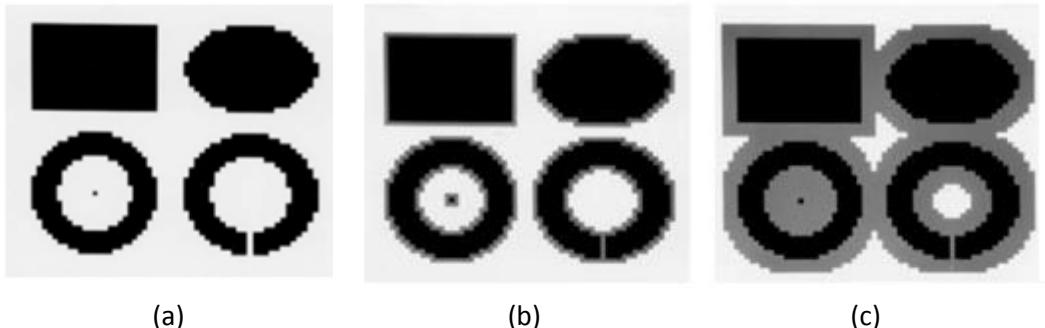
Dilatacija (eng. *dilation*) je prva matematska morfološka operacija, čiji je zadatak povećati (proširiti) oblike prikazane na slici. Proses se može objasniti definiranjem tzv. strukturnog elementa, tj. elementa od čijeg oblika, veličine i težišta (ili neke druge, unaprijed zadane referentne tačke) zavisi kakav će biti rezultat dilatacije. Translacijom strukturnog elementa preko svakog sastavnog piksela slike, dobiva se modifikacija slike, na kojoj su prikazani objekti podebljani. Modificiranu verziju čine pikseli nastali presjekom piksela originalne slike i transliranog strukturnog elementa.

Neka je posmatrani piksel p zadan koordinatama $p = (p_1, p_2)$ i neka je g_p strukturalni element koji se translira. Tada se dilatacija slike δ_g može matematski predstaviti relacijom:

$$\delta_g(f) = \{p | g_p \cap f \neq \emptyset\}. \quad (2.4)$$

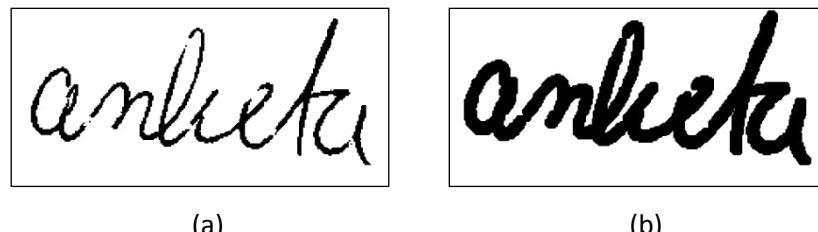
Dakle, rezultat dilatacije slike je skup piksela za koje je presjek strukturnog elementa g_p i piksela f originalne slike neprazan skup. Kako se dilatacija primjenjuje i na unutrašnje i na vanjske granice oblika prikazanog objekta, manje šupljine se još više smanjuju, ili čak nestaju, što može predstavljati problem [12].

Primjeri primjene dilatacije prikazani su na *slici 2.9* i *slici 2.10*.



Slika 2.9: Dilatacija; (a) originalna slika; (b) slika nakon primjene dilatacije (jedna iteracija); (c) slika nakon primjene dilatacije (tri iteracija).³

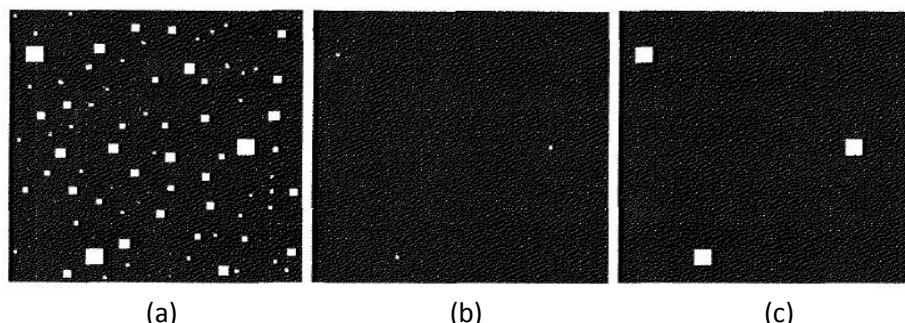
³ Slika preuzeta iz [11].



Slika 2.10: Dilatacija; (a) originalna slika; (b) slika nakon višestruke primjene dilatacije.

Erozija

Erozija (eng. *erosion*) je morfološka operacija suprotna dilataciji. Njena primjena nad binarnom slikom podrazumijeva translaciju strukturnog elementa preko sastavnih piksela slike, što sada uzrokuje smanjenje veličine prikazanih objekata. Najčešća svrha primjene erozije je eliminiranje nebitnih detalja sa binarne slike [7]. Naprimjer, ako za analizu slike 2.11 (a) nisu relevantni kvadrati manjih dimenzija od najvećih prikazanih kvadrata, moguće je primjenom erozije izdvojiti samo one najveće, kao na slici 2.11 (b). Uočljivo je da će izdvojeni kvadrati biti manjih dimenzija u odnosu na originalne. Međutim, moguće je izdvojenim objektima vratiti početne dimenzije. Kako je već ranije spomenuto, u morfološkoj analizi slike kombinacija erozije i dilatacije daje najbolje rezultate. Stoga, ako se nakon izdvajanja relevantnih elemenata erozijom primijeni dilatacija, moguće je vratiti originalne dimenzije prikazanog objekta. Rezultat je prikazan na slici 2.11 (c).



Slika 2.11: Erozija; (a) originalna slika; (b) izdvajanje najvećih kvadrata korištenjem strukturnog elementa odgovarajućih dimenzija - erozija; (c) primjena dilatacije nad (b), uz korištenje istog strukturnog elementa.⁴

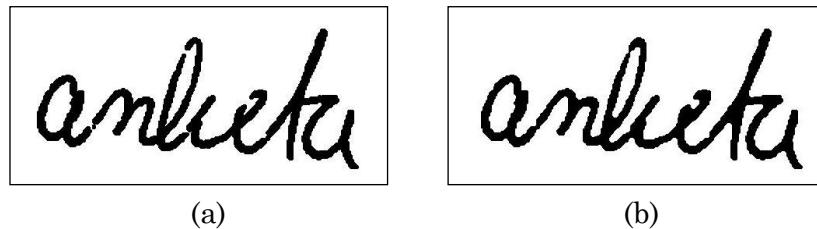
Morfološko otvaranje i zatvaranje

Primjena morfološkog otvaranja (eng. *morphological opening*) nad slikom rezultira u zaglađenijim konturama prikazanog objekta, uski prolazi, tj. „pukotine“ na slici se još više razdvajaju, dok se tanke izbočine u potpunosti eliminiraju. Iako morfološko zatvaranje (eng. *morphological closing*) podrazumijeva zaglađivanje kontura objekta, kao i morfološko otvaranje, ove dvije operacije su međusobno suprotne. Zatvaranje ima ulogu spajanja uskih prolaza na slici i eliminiranja malih šupljina, odnosno, popunjavanja

⁴ Slika preuzeta iz [7].

praznina unutar kontura objekta [15]. Postupak podrazumijeva korištenje struktturnog elementa, na sličan način kao u slučaju primjene dilatacije.

Kako bi se omogućila što bolja ekstrakcija konture pisanog teksta, korisno je primijeniti operaciju morfološkog zatvaranja (*slika 2.12*), s obzirom na to da su isprekidane konture česta pojava uslijed skeniranja slika pisanog teksta. Međutim, degradacija skeniranog teksta je često prevelika da bi se ispunile sve neželjene praznine unutar konture. To je slučaj i na *slici 2.12*, gdje je uočljiv prekid konture slova *e*. Problem se u nekim slučajevima može riješiti povećanjem dimenzija struktturnog elementa, ali nekada ovaj princip dovodi do spajanja međusobno odvojenih kontura, zbog čega nije primijenjen ni na prikazanom primjeru.



Slika 2.12: Morfološko zatvaranje; (a) originalna slika; (b) slika nakon primjene morfološkog zatvaranja.

Istanjivanje kontura

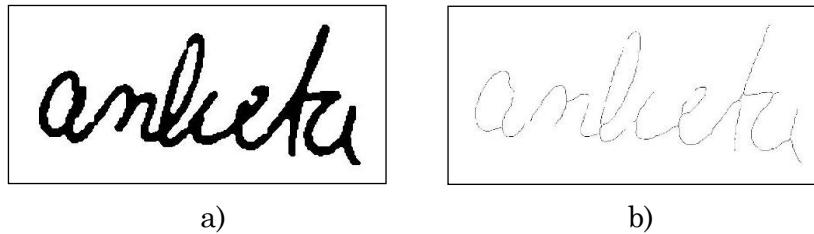
Pod pojmom istanjivanja kontura na binarnoj slici podrazumijeva se brisanje crnih piksela, na način da se oblik objekta bez šupljina svodi na liniju formiranu od minimalnog broja međusobno povezanih piksela pozicioniranih tako da se nalaze na jednakim udaljenostima od njima najbližih vanjskih granica konture [11]. Relacija (2.5) pokazuje primjer istanjivanja konture kada je objekat bez šupljina prikazan matricom dimenzija 3×4 . U slučaju objekta sa šupljinama, oblik objekta se svodi na krug sačinjen od minimalnog broja međusobno povezanih piksela, pozicioniranih na sredini između šupljine i najbližih vanjskih granica konture [11].

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \xrightarrow{\substack{\text{istanjivanje} \\ \text{konture}}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.5)$$

Istanjivanje slike može se vršiti u više iteracija, a povećanje broja iteracija kao rezultat daje tanje konture. Ako se istanjuju konture objekta bez šupljina, prevelik broj iteracija uzrokovat će da kontura postane tačka.

Proces skeletonizacije, odnosno traženja „kostura“ objekta prikazanog na slici, zasniva se na istanjivanju kontura posmatranog objekta. Skelet se koristi za opisivanje strukture objekta, a definira se kao skup tačaka jednako udaljenih od dvije najbliže granične tačke objekta [11].

Rezultat istanjivanja nakon primjene dilatacije i morfološkog zatvaranja nad crno-bijelom verzijom originalne slike prikazan je na *slici 2.13*. Na ovaj način dobiven je skelet početne slike.



Slika 2.13: Istanjivanje kontura; (a) originalna slika; (b) slika nakon istanjivanja kontura.

Treba napomenuti da pored prethodno izloženih morfoloških operacija postoje još neke, koje se većinom zasnivaju na osnovnim – dilataciji i eroziji. Naime, odgovarajućim kombinacijama ovih operacija, izvedeno je istanjivanje, ali i morfološko podebljavanje (suprotna operacija u odnosu na istanjivanje). Osim toga, na dilataciji i eroziji zasnovane su i neke transformacije.

2.2. Segmentacija podataka

Segmentacija je proces dijeljenja slike na njene sastavne objekte ili regije [10]. Nivo do kog se vrši podjela zavisi o problemu od interesa. Algoritmi koji vrše segmentaciju monohromatskih slika baziraju se na analizi vrijednosti intenziteta boje na slici. Na osnovu toga, može se doći do zaključaka o sličnosti između piksela i prisutnim diskontinuitetima. Ako algoritam za segmentaciju počiva na podacima o sličnosti piksela, slika se dijeli na regije klasificirane kao slične, prema unaprijed zadanim kriterijima. S druge strane, ako algoritam raspolaže podacima o diskontinuitetima na slici, tada se podjela vrši na osnovu naglih promjena vrijednosti intenziteta boje, koje se javljaju npr. na ivicama objekta koji je prikazan. *Slika 2.14* ukazuje na korake koje obuhvata segmentacija podataka.



Slika 2.14: Koraci u segmentaciji podataka.

2.2.1. Određivanje praga

Najjednostavnije i najčešće korištene metode za segmentaciju slike zahtijevaju prvo bitno određivanje praga (eng. *thresholding*). Određivanje praga služi kao osnov za transformaciju originalne slike f , koja se obraduje, u izlaznu binarnu sliku g , što se može prikazati relacijom:

$$g(x, y) = \begin{cases} 1, & \text{za } f(x, y) \geq T \\ 0, & \text{za } f(x, y) < T \end{cases}, \quad (2.6)$$

gdje je T vrijednost praga, $g(x, y) = 1$ za one piksele kojim je prikazan objekat na slici i $g(x, y) = 0$ za piksele koji odgovaraju podlozi slike [6]. Algoritmi koji koriste metodu određivanja praga funkcioniraju na način da se prvo izvrši pretraga svih piksela $f(x, y)$ slike f , a potom se, u zavisnosti od toga da li je vrijednost piksela manja ili veća (eventualno jednaka) od praga T , pikseli klasificiraju kao sastavni dijelovi podloge ili objekta, respektivno. Prethodno opisana procedura oslanja se na koncept histograma. Histogram je grafički prikaz koncentracije pojedinih boja na slici, odnosno, koncentracije nijansi sive boje, u slučaju monohromatskih slika [12]. Vrijednost praga određuje se upravo na osnovu podataka koje nudi histogram. Međutim, modifikacija veoma malog broja slika daje dobre rezultate za jedinstvenu vrijednost praga, koja se može dobiti tzv. globalnim određivanjem praga (eng. *global thresholding*). U većini slučajeva, potrebno je tokom procesa segmentacije koristiti više različitih referentnih vrijednosti za odlučivanje da li je posmatrani piksel sastavni dio objekta ili podloge. Adaptivno određivanje praga (eng. *adaptive thresholding*) omogućava prilagođenje te referentne vrijednosti u skladu sa promjenjivim karakteristikama slike. Ova tehnika naziva se još i lokalno određivanje praga (eng. *local thresholding*). Dakle, ako se vrijednost praga određuje na globalnom nivou, tj. na osnovu cijele slike, tada vrijedi:

$$T = T(f), \quad (2.7)$$

dok je na lokalnom nivou T funkcija dvije promjenjive:

$$T = T(f, f_c), \quad (2.8)$$

pri čemu je f_c onaj dio slike u kom se vrši određivanje vrijednosti praga [6]. Za lokalno određivanje praga, slika f se prvo bitno izdijeli na nekoliko dijelova f_c , nakon čega se određuje prag za svaki od tih dijelova ponaosob. Ukoliko nije moguće odrediti vrijednost praga za neki od pripadajućih dijelova slike, dodjeljuje mu se vrijednost na osnovu pragova određenih za susjedne dijelove.

Bitno je spomenuti i tehniku optimalnog određivanja praga. Metoda koja se bazira na aproksimaciji histograma slike, koristeći težinsku sumu⁵ dvije ili više gustoća vjerovatnoće sa normalnom (Gaussovom) raspodjelom naziva se optimalnim određivanjem praga [6]. Posmatraju se maksimumi vjerovatnoća dvije ili više normalnih raspodjela, na osnovu kojih se prag određuje kao ona vrijednost čija je vjerovatnoća najmanja u skupu posmatranih. Na ovaj način se postiže najmanja vjerovatnoća greške u procesu segmentacije.

⁵ Suma koja uzima u obzir važnost faktora za koji se računa.

2.2.2. Detekcija tačaka, linija i rubova

Postoje tri osnovna tipa diskontinuiteta koja se pojavljuju na digitalnim slikama: tačke, linije i rubovi. Najlakši način njihove detekcije je primjena filtra, tj. maske u vidu matrice određenih dimenzija (najčešće 3×3 , tako da se osim posmatranog piksela uzmu u obzir i njemu najbliži susjedni pikseli) na isti način kako je to opisano u procesu primjene maske za odstranjivanje šuma u poglavljiju 2.1.2. Ako se primjenjuje maska dimenzija

3×3 , algoritam za detekciju diskontinuiteta na slici vrši računanje sume proizvoda koeficijenata maske sa vrijednostima piksela originalne slike nad kojim se maska primjenjuje. Primjena maske čija je pripadna matrica:

$$\begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \\ \omega_4 & \omega_5 & \omega_6 \\ \omega_7 & \omega_8 & \omega_9 \end{bmatrix}, \quad (2.9)$$

kao rezultat daje:

$$R = \omega_1 z_1 + \omega_2 z_2 + \dots + \omega_9 z_9 = \sum_{i=1}^9 \omega_i z_i, \quad (2.10)$$

gdje su z_i vrijednosti pridružene pikselima u zavisnosti od njihove nijanse, a ω_i koeficijenti maske [7].

Detekcija tačaka

Detekcija tačaka vrši se jednostavnom primjenom uslova:

$$|R| \geq T, \quad (2.11)$$

gdje je T prethodno određena vrijednost praga, o čemu je više riječi bilo u *paragrafu 2.2.1.* Relacija (2.11) ukazuje na to da se vrši poređenje vrijednosti dodijeljene razmatranom pikselu i vrijednosti njemu susjednih piksela. Ideja se zasniva na tome da se izolovana tačka, tj. ona tačka čija se nijansa znatno razlikuje od podloge koja je homogena, ili barem približno homogena, ujedno razlikuje i od susjednih tačaka, zbog čega je omogućena njena detekcija primjenom maske [7].

Detekcija linija

Detekcija linija na digitalnoj slici, kao i detekcija tačaka, zahtijeva primjenu maski. U tu svrhu, mogu se definirati četiri maske tipične za proces detekcije linija:

horizontalna:

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}, \quad (2.12)$$

$+45^\circ$:

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}, \quad (2.13)$$

vertikalna:

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}, \quad (2.14)$$

-45°:

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}. \quad (2.15)$$

Maska zadana matricom (2.12) najbolje rezultate daje ako se primjeni nad skupom piksela od kojih određeni broj formira horizontalnu liniju. U tom slučaju će pikseli kojim je prikazana linija biti podebljani. Analogno vrijedi i za ostale maske. Rezultat primjene maske (2.13) najbolji je ako slika sadrži linije pod uglom od +45°, maska (2.14) omogućava detekciju vertikalnih linija, a (2.15) onih linija koje su pod uglom od -45°. Koeficijent 2 u matricama (2.12)-(2.15) ukazuje na povećanje vrijednosti odgovarajućih piksela nad kojim se primjenjuje maska. Treba napomenuti da ovaj koeficijent može poprimati i neku drugu vrijednost, veću od 1. Međutim, to nije jedino ograničenje. Suma svih članova maske mora biti jednaka nuli, čime se ukazuje na to da maska nema utjecaja kada se primjeni nad skupom piksela konstantne nijanse sive boje.

Ako se maske prikazane matricama (2.12)-(2.15) pojedinačno primijene nad istom slikom, može se vršiti poređenje njihovih performansi za detekciju linija na slici. Neka su rezultati, tj. odzivi maski označeni sa R_1, R_2, R_3 i R_4 , respektivno. Uvažavajući relaciju (2.10) poređenje se vrši tako da se za svako $i, j \in \{1, 2, 3, 4\}$, $i \neq j$ posmatra uslov $|R_i| > |R_j|$. Ukoliko je ovaj uslov ispunjen, smatra se da je veća vjerovatnoća da je maska sa odzivom R_i pogodna za primjenu. Naprimjer, u slučaju da je $i = 1, j \in \{2, 3, 4\}$ i uslov $|R_i| > |R_j|$ je uvijek zadovoljen, smatra se da je posmatrana tačka povezana sa tačkama koje se nalaze s njene lijeve i desne strane, na taj način formirajući horizontalnu liniju [7].

Detekcija rubova

Detekcija skupa tačaka koje odgovaraju izraženijim promjenama nijanse sive boje na slici naziva se detekcijom rubova [16]. Ako se razmatranje ograniči na slike u sivim tonovima, proces se zasniva na računanju izvoda prvog ili drugog reda funkcije kojom je slika opisana. Računanje izvoda prvog reda (gradijenta) vrši se u tzv. procesu detekcije rubova korištenjem gradijenta. U tom slučaju, porede se amplitude gradijenta pripadajućih tačaka slike sa unaprijed određenom vrijednosti praga. Tačke koje odgovaraju rubovima imaju veće vrijednosti gradijenta, pa se detekcija rubova vrši jednostavnim izborom onih tačaka čija je vrijednost iznad zadanog praga. S druge strane, kada se algoritam za detekciju rubova zasniva na izvodima drugog reda (laplasijan), rubovi se prepoznaju kao one tačke u kojima funkcija izvoda drugog reda mijenja svoj znak.

Rubovi na binarnim slikama mogu se detektovati zahvaljujući sljedećim činjenicama:

- pikseli rubova pripadaju objektu prikazanom na slici, zbog čega imaju vrijednost različitu od piksela podloge slike,
- postoji barem jedan piksel koji pripada podlozi, a susjedan je posmatranom, koji pripada objektu na slici.

Svaki rub je vektorska varijabla sa dvije komponente – amplitudom i smjerom. Amplituda ruba jednaka je amplitudi gradijenta, a smjer ruba rotira se u skladu sa smjerom gradijenta. Smjer gradijenta daje informaciju o maksimalnom porastu funkcije $f(x,y)$, kojom je opisana slika. Sam gradijent dvokomponentne varijable definira se relacijom:

$$\text{grad}[f(x,y)] = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}, \quad (2.17)$$

njegov modul (amplituda) je:

$$|\text{grad}[f(x,y)]| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}, \quad (2.18)$$

dok je smjer gradijenta jednak uglu između apscisne ose i tačke čije su koordinate određene parcijalnim izvodima funkcije f po komponentama x i y :

$$\psi = \arg\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right). \quad (2.19)$$

Metod detekcije rubova pomoću gradijenta koristi se kada je bitan ugao pod kojim se rub nalazi. Ako to nije slučaj, vrši se detekcija pomoću laplasijana. Ova metoda uzima u obzir samo amplitudu rubova. Laplasijan ima jednaka svojstva u svim smjerovima, zbog čega je invarijantan na rotaciju slike, a definira se sljedećom relacijom:

$$\nabla^2 f(x,y) = \frac{\delta^2 f(x,y)}{\delta x^2} + \frac{\delta^2 f(x,y)}{\delta y^2}, \quad (2.20)$$

gdje je ∇ Laplaceov operator.

Usporedbom metoda detekcije rubova pomoću gradijenta i pomoću laplasijana, dolazi se do zaključka da su rubovi tanji ako se detektuju korištenjem laplasijana. Uprkos ovoj prednosti, zbog velikog broja zatvorenih petlji koje se pojavljaju u rezultatima primjene ove metode, ona se pokazala nepovoljnog. Sam proces traženja tačaka u kojima drugi izvod funkcije mijenja znak, što je koncept na kom se zasniva metoda, zahtijeva mnogo vremena, a to je još jedan razlog koji detekciju pomoću gradijenta čini povoljnijom metodom.

POGLAVLJE 3

Praktična realizacija

3.1. Preprocesiranje slike

Algoritam za analizu rukopisa implementiran je u programskom paketu MATLAB, a testiranje njegovog rada izvršeno je nad bazom skeniranih slika rukopisa, na kojima je grafitnom olovkom napisana riječ „anketa“. Analiza je provedena i nad manjom skupinom ostalih riječi iz rečenice „Ovo je najzabavnija anketa koja se ikada pojavila“, koje su također sastavni dio baze, korištene i u radu [17].

U poglavljiju 2.1. data je teorijska podloga za preprocesiranje slike. Za potrebe implementacije, dovoljno je izvršiti samo neke od operacija koje su razmatrane u spomenutom poglavljju.

Prvi korak u preprocesiranju slike je normalizacija podataka. U okviru ovog segmenta, izvršena je prilagodba kontrasta na slici, dok se normalizacija nagiba osnovne linije pisanja i nagiba slova nisu vršile u ovom dijelu. Ova dva postupka obično se provode prije segmentacije u slučaju kada je cilj algoritma prepoznavanje sadržaja pisanog teksta, što nije tema ovog rada. Međutim, njihov značaj za ekstrakciju relevantnih parametara se ne može osporiti. Stoga će se nakon procesa segmentacije testirati utjecaj normalizacije nagiba slova i osnovne linije pisanja, u cilju dobivanja informacija o uglovima nagiba.

U zavisnosti od veličine originalne slike, ponekad je korisno izvršiti promjenu dimenzija iste. Kako su slike rukopisa same po sebi u većini slučajeva malih dimenzija, prvobitno su, pomoću ugrađene MATLAB funkcije *resize*, dimenzije originalne slike uvećane 4 puta. Nakon toga, izvršena je normalizacija kontrasta, korištenjem tehnike adaptivnog odredivanja praga (o kojoj je više riječi bilo u poglavljju 2.2.1). Ova tehnika može se implementirati kombinacijom MATLAB funkcija *imfilter* i *fspecial*. Funkcija *fspecial* omogućava specifikaciju vrste filtra koji će se koristiti, a mogući filtri primjenjuju se naznakom odgovarajuće ključne riječi: „average“, „disk“, „gaussian“, „laplacian“, „log“, „motion“, „prewitt“ ili „sobel“. Korišten je filter za usrednjavanje, ključne riječi „average“. Treba napomenuti da MATLAB nudi i ugrađenu funkciju *graythresh* za globalno

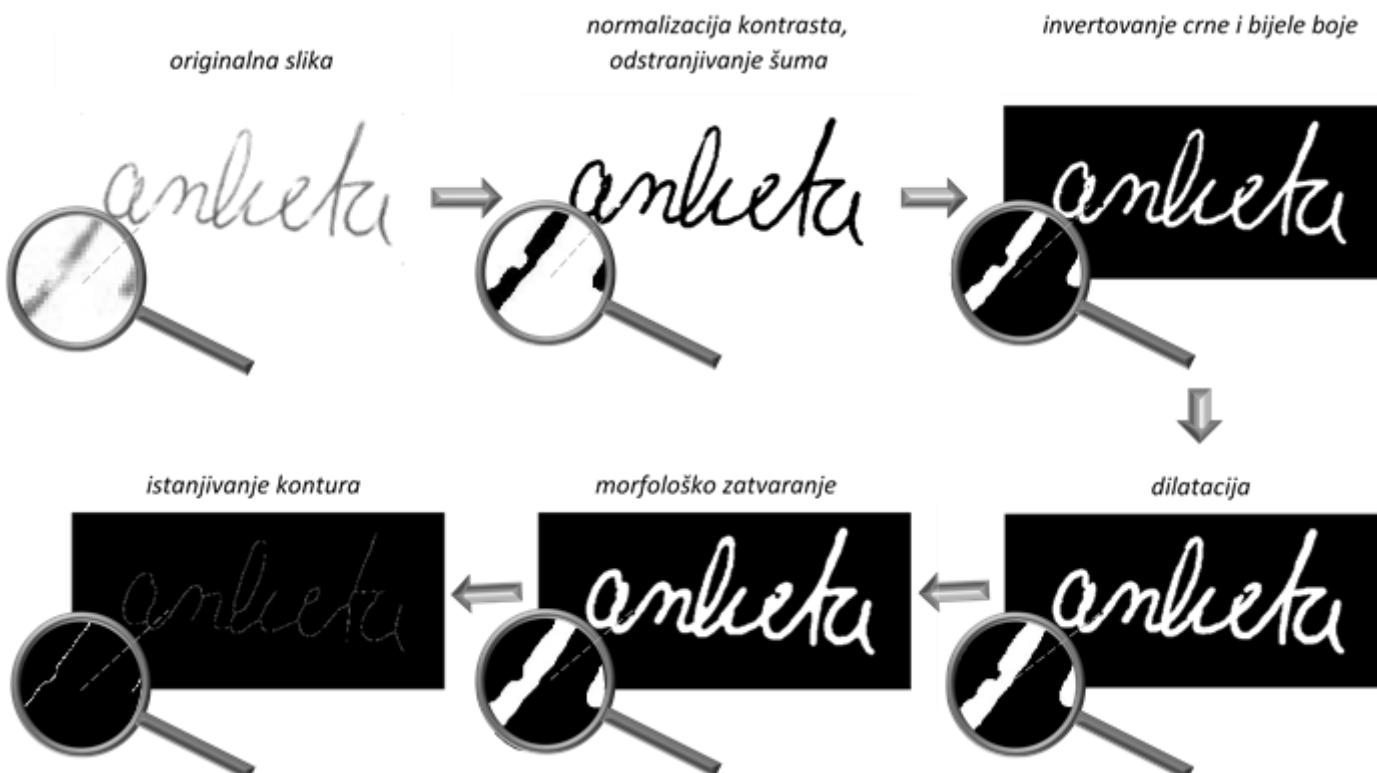
određivanje praga, ali je ova tehnika dala lošije rezultate u odnosu na lokalno, tj. adaptivno određivanje praga.

Filtriranje po srednjoj vrijednosti pokazalo se kao najbolja tehnika za odstranjivanje šuma nad razmatranim skupom slika. Funkcija *medfilt2* u velikoj mjeri odstranjuje šum, te pruža mogućnost otklanjanja crnih piksela u uglovima slike, koji se javljaju kao neželjena posljedica filtriranja.

Uobičajeno je da se pisani tekst na analiziranoj slici predstavi bijelim pikselima, a da podloga bude crne boje. Stoga se, nakon naglašavanja kontrasta i smanjenja nivoa šuma na slici, primjenjuje funkcija *im2bw* kojom se osigurava da svi pikseli poprimaju jednu od dvije vrijednosti: 0 ili 1 (crna i bijela boja, respektivno), odnosno, da nema sivih piksela. Kada se nad ovom funkcijom primjeni negacija, tj. upotrijebi funkcija *not*, dobiva se željeni rezultat – invertovanje crnih i bijelih piksela.

Morfološke operacije koje su primijenjene u okviru preprocesiranja slike su dilatacija (eng. *dilation*), morfološko zatvaranje (eng. *morphological closing*) i istanjivanje (eng. *thinning*). Za prve dvije pobrojane operacije koristi se pomoćna funkcija *strel*, koja omogućava izbor oblika strukturnog elementa za izvršenje odgovarajućeg morfološkog postupka. Neki od mogućih oblika su: „line“, „disk“, „diamond“, „rectangle“, „square“, itd. I u slučaju dilatacije, kao i morfološkog zatvaranja, najboljom opcijom se pokazao strukturni element kružnog oblika – „disk“. Dilatacija je izvedena primjenom funkcije *imdilate*, a morfološko zatvaranje funkcijom *imclose*.

Rezultat svih operacija primijenjenih u preprocesiranju slike prikazan je na *slici 3.1*.



Slika 3.1: Koraci u preprocesiranju slike

Posljednji korak u pretprocesiranju je istanjivanje kontura slike. Ovaj postupak izvršen je primjenom Zhang-Suen algoritma. Algoritam se sastoji u tome da se posmatra svaki crni piksel (piksel podloge) koji ima osam susjednih piksela. Dakle, vrši se izdvajanje matrice dimenzija 3×3 , gdje je posmatrani piksel na poziciji (2, 2) u matrici, tj. u centru. Neka je ovaj piksel označen sa p_1 i neka je raspored piksela u izdvojenoj matrici sljedeći:

$$\begin{bmatrix} p_9 & p_2 & p_3 \\ p_8 & p_1 & p_4 \\ p_7 & p_6 & p_5 \end{bmatrix}. \quad (3.1)$$

Označimo sa A broj prelazaka sa bijelog na crni piksel, u nizu $p_2 - p_3 - p_4 - p_5 - p_6 - p_7 - p_8 - p_9 - p_2$, a sa B broj susjednih piksela koji su crni. Pikselima koji ispunjavaju sljedeće kriterije (u tačno navedenom redoslijedu) postavlja se vrijednost koja odgovara bijelom pikselu:

- piksel je crn i ima osam susjeda,
- $2 \leq B \leq 6$,
- $A = 1$,
- barem jedan od piksela p_2 , p_4 i p_6 je bijel,
- barem jedan od piksela p_4 , p_6 i p_8 je bijel.

Nakon završetka ove faze, ponavlja se ispitivanje prva tri navedena uslova. Ukoliko su oni zadovoljeni, ispituje se i da li je:

- barem jedan od piksela p_2 , p_4 i p_8 bijel,
- barem jedan od piksela p_2 , p_6 i p_8 bijel.

Ako su i ova dva uslova zadovoljena, vrijednost središnjeg piksela izdvojene matrice dimenzija 3×3 postavlja se na vrijednost bijelog piksela. Na ovaj način se omogućava svodenje grupe bijelih piksela na jedan, što rezultira istanjenom konturom.

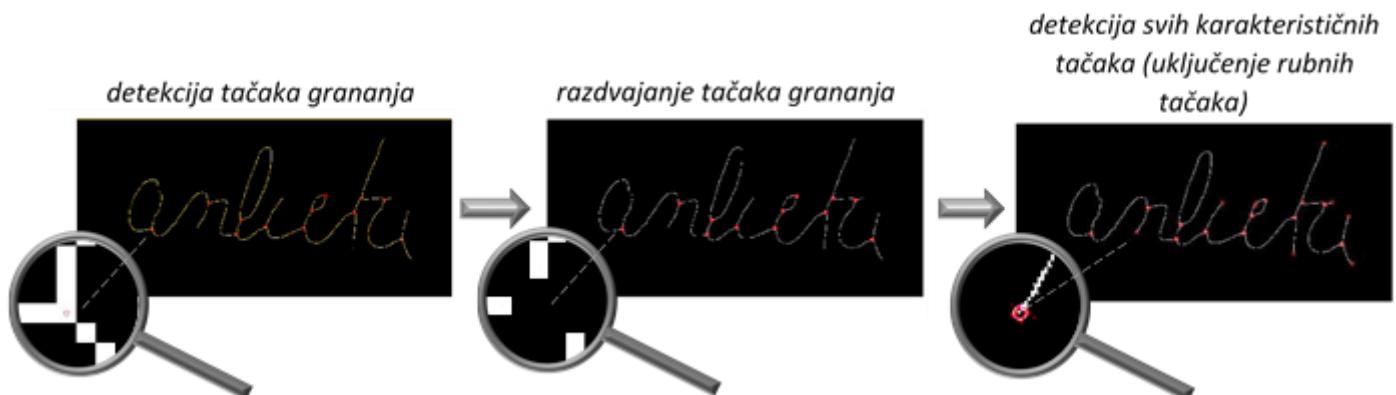
3.2. Segmentacija podataka

Kada je u pitanju analiza rukopisa, najznačajniji dio segmentacije podataka je detekcija tačaka. Postoje dvije grupe karakterističnih tačaka koje je potrebno identificirati: tačke grananja i krajnje tačke konture. Pod pojmom tačke grananja podrazumijeva se ona tačka u kojoj se presijecaju najmanje dvije linije konture, dok je krajnja tačka ona tačka (bijeli piksel) čije sve susjedne tačke izuzev jedne pripadaju podlozi (crni pikseli). S obzirom na to da se u ovom stadiju vrši analiza crno-bijele slike, proces je pojednostavljen iz razloga što nije potrebno vršiti određivanje praga. Umjesto toga, tačke grananja se detektuju pomoću šablona rasporeda u kojima se mogu nalaziti pikseli unutar izdvojene sastavne matrice slike. Dakle, vrši se izdvajanje matrice dimenzija 3×3 i za svaku izdvojenu matricu se provjerava da li je središnji piksel bijele boje, te da li su njemu susjedni bijeli pikseli u nekom od rasporeda iz šablona za detekciju tačaka grananja. Ukoliko je to slučaj, središnji piksel se označava kao tačka grananja.

Da bi algoritam za analizu rukopisa imao sposobnost predviđanja putanje pisana, potrebno je konturu rukopisa razdvojiti na više segmenata. Razdvajanje svake tačke

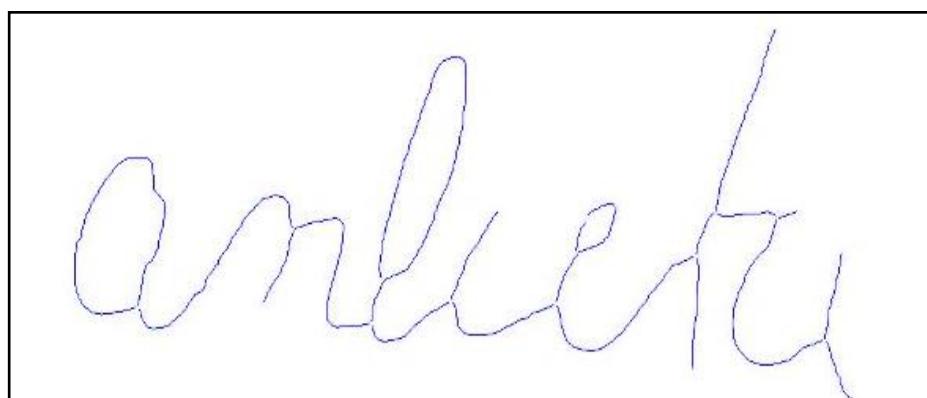
grananja na više krajnjih tačaka implicira razdvajanje segmenata konture koji su se presijecali u tim tačkama. Najjednostavniji način implementacije opisanog postupka sastoji se u tome da se vrijednosti svih piksela matrice dimenzija 3×3 koja kao središnji piksel ima tačku grananja postave na vrijednosti crnih piksela. Nakon toga, potrebno je izvršiti detekciju novonastalih krajnjih tačaka konture. Krajnje tačke nastale na ovaj način jednostavno se detektuju izdvajanjem matrice dimenzija 5×5 oko piksela koji je predstavljao tačku grananja. Ovakva matrica može imati bijele piksele samo u prvom i posljednjem redu, te u prvoj i posljednjoj koloni, s obzirom na to da su svi pikseli njene unutrašnje matrice (3×3) prebrisani – postavljeni na vrijednost crnih piksela. Moguće krajnje tačke su upravo ovi bijeli pikseli. Ukoliko unutar jednog reda ili jedne kolone postoji više od jednog bijelog piksela, kao krajnji piksel se bira onaj koji je bliži središnjem.

Nakon određivanja krajnjih tačaka nastalih razdvajanjem tačaka grananja, potrebno je odrediti i krajnje tačke originalne konture. One se jednostavno određuju sumiranjem svih elemenata matrice dimenzija 3×3 , izdvojene oko bijelog piksela, uz provjeru uslova da je suma jednak 2 (bijeli pikseli imaju vrijednost 1, što implicira da je još samo jedan piksel pored središnjeg bijelog).



Slika 3.2: Detekcija karakterističnih tačaka konture.

Detekcija linija konture znatno je lakša ako su poznate krajnje tačke svih sastavnih segmenata. Korištenjem podataka o koordinatama krajnjih tačaka, vrši se „iščitavanje“ bijelih piksela između dvije krajnje tačke svakog segmenta. Na taj način se dobiva linija segmenta. Rezultat iscrtavanja linija svih segmenata prikazan je na slici 3.3.



Slika 3.3: Izdvojena linija konture.

3.3. Ekstrakcija parametara

Za svaki mogući redoslijed određenog broja segmenata konture i smjer kretanja po istim, algoritam za analizu rukopisa vrši proračun funkcije cilja. Ova funkcija uzima u obzir faktore relevantne za predviđanje putanje pisanja, množeći ih sa odgovarajućim težinskim koeficijentima. Osim što moraju biti izabrani tako da naglašavaju nivo relevantnosti svakog pojedinog faktora za ciljnu funkciju, težinski koeficijenti ujedno trebaju omogućiti i usklađivanje svih vrijednosti koje funkcija obuhvata.

Svaki segment konture ima definirane po dvije krajnje tačke, koje služe kao osnov za određivanje smjera kretanja. Ako se krajnje tačke označe brojevima 1 i 2, moguće sheme kretanja po segmentu su: 1 – 2 i 2 – 1 (putanja bez povratka), te 1 – 2 – 1 i 2 – 1 – 2 (putanja sa povratkom). Manja vrijednost funkcije cilja za posmatrani redoslijed segmenata i smjer kretanja po istim indikator je veće vjerovatnoće da je stvarna putanja pisanja upravo onakva kao razmatrana.

Relevantni faktori su:

- w_1 : ukazuje na poziciju početne tačke konture po apscisnoj osi,
- w_2 : uzima u obzir ugao kretanja kroz tačke grananja,
- w_3 : sadrži informaciju o dužini pređenog puta,
- w_4 : sadrži informaciju o skokovitoj putanji, tj. prelazima između segmenata koji se ne nalaze na bliskom rastojanju,
- w_5 : sadrži informaciju o kretanju olovke u vazduhu,
- w_6 : provjerava da li je početna tačka pisanja ujedno i tačka grananja.

Funkcija cilja se sada može predstaviti jednostavnom relacijom:

$$f = \sum_{i=1}^6 w_i, \quad (3.2)$$

Kako je već spomenuto, svaki segment konture sadrži po dvije krajnje tačke, ali nije definirano koja od njih je početna, a koja završna tačka. Faktor w_1 uzima u obzir poziciju obje krajnje tačke u odnosu na apscisnu osu i daje prednost onoj tački koja ima manju vrijednost koordinate x . Drugim riječima, u ovom slučaju je kriterij da početna tačka bude pozicionirana što više ulijevo. Razlog tome leži u činjenici da je u latiničnom pismu smjer pisanja slijeva nadesno. Ovaj kriterij naročito je važan u slučaju da se analizira početak jedne riječi, tj. konture, dok je njegov značaj manji ukoliko se analizira proizvoljan segment konture, koji nije početni.

Faktor w_2 nosi informaciju o uglu kretanja kroz tačke grananja. Za svaka dva izdvojena segmenta konture, algoritam prvo provjerava da li su završna tačka jednog i početna tačka narednog segmenta ujedno i tačke koje su nastale kao posljedica razdvajanja tačke grananja na više krajnjih tačaka. Ukoliko jesu, računa se ugao između pravca kretanja po prvom i pravca kretanja po narednom segmentu, u odnosu na identificirane krajnje tačke. Vjerovatnoća ispravnog poretku segmenata i redoslijeda kretanja po njima veća je što je promjena ugla kretanja kroz tačku grananja manja.

Informacija o dužini pređenog puta pri kretanju po konturi sadržana je u faktoru w_3 . Dužina putanje jednog segmenta računa se na osnovu ukupnog broja sastavnih

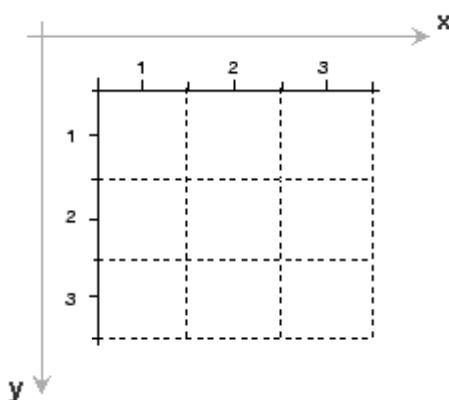
piksela segmenta. Počevši od jednog kraja segmenta, vrši se vrednovanje prelazaka iz tačke u tačku, sve do drugog kraja segmenta. Prelazak iz jednog piksela na njemu susjedni piksel, pozicioniran dijagonalno u odnosu na posmatrani, ima vrijednost jednaku $\sqrt{2}$, dok prelasci u ostale susjedne piksele imaju vrijednost 1. Ove vrijednosti odgovaraju gradijentu prave određene sa dvije tačke – posmatranim pikselom i onim u koji se prelazi.

Treba napomenuti i da se, u slučaju kretanja po putanji sa povratkom na početnu tačku, izračunati pređeni put množi sa koeficijentom 2, što indicira dvostruko pređen put.

Skokovita putanja pisanja podrazumijeva postojanje velikog broja diskontinuiteta pisanog teksta, što je indikator čestog podizanja olovke s papira. Svrha faktora w_4 je smanjenje diskontinualnog kretanja po segmentima konture. Za svaka dva izdvojena segmenta, algoritam vrši provjeru postojanja kraće putanje prelaska s jednog na drugi segment. Ukoliko takva putanja postoji, postoji ujedno i pogodniji redoslijed kretanja po posmatranim segmentima, za koji će vrijednost faktora w_4 biti manja.

Faktor w_5 sadrži informacije o kretanju olovke u vazduhu. Prvi korak pri proračunu ovog faktora je detekcija podizanja olovke s papira, što se ostvaruje provjerom rastojanja između dva susjedna segmenta konture. Ukoliko je najmanje rastojanje među njima veće u odnosu na maksimalno rastojanje na kom se mogu nalaziti dva segmenta nastala razdvajanjem tačke grananja na više krajnjih tačaka, detektuje se podizanje olovke. Sljedeći korak je proračun udaljenosti između prethodnog i narednih segmenata, u odnosu na apscisnu osu. Manje udaljeni segmenti više su vjerovatni. Osim toga, relevantan je i položaj narednih segmenata u odnosu na ordinatu. Naime, uobičajeno je da se nakon podizanja olovke s papira nastavlja pisati u smjeru odozgo prema dolje, zbog čega se prednost daje onim segmentima sa manjom vrijednosti na ordinatnoj osi. Razlog zbog kog je manja vrijednost koordinate y od interesa leži u tome da je koordinatni sistem koji odgovara slici učitanoj u programskom paketu MATLAB reflektovan u odnosu na horizontalnu osu, što ima za posljedicu da tačke na vizuelno višoj poziciji na slici imaju manju vrijednost na ordinatnoj osi.

Konačno, faktor w_6 nosi informaciju o tome da li je početna tačka razmatranog segmenta konture ujedno i tačka grananja ili ne. Implementirani algoritam vrši provjeru krajnjih tačaka potencijalnih početnih segmenata. Prednost se daje onim segmentima čija se krajnja tačka ne grana. Ovaj faktor je relevantan samo za određivanje početne tačke konture.



Slika 3.4: Prikaz koordinatnog sistema programskog paketa MATLAB.

Primjena prethodno opisane funkcije u algoritmu za računarsku analizu rukopisa omogućava dobivanje moguće najvjeroatnije putanje pisanja. Jedan od rezultata prikazan je na *slici 3.5*. Crvena boja označava kretanje po konturi, a zelena boja je indikator da se putanja dva put prelazi (putanja sa povratkom).



Slika 3.5: Prikaz rezultata predviđanja putanje pisanja pomoću sistema za analizu rukopisa.

U okviru poglavlja 2.1.1 obrađene su metode obuhvaćene postupkom normalizacije. Neke od njih, kao što su normalizacija nagiba slova i osnovne linije pisanja, nisu bile izvršene prije ekstrakcije dinamičkog zapisa teksta iz skeniranih dokumenata i ekstrapolacije 3D kretanja vrha olovke, kako se ne bi narušile informacije koje nosi originalni zapis. Analiza je provedena naknadno, a numerički rezultati (za razmatrani primjer) su prikazani u tabeli 3.1. Vizuelni prikaz rezultata dat je u okviru spomenutog poglavlja. Treba napomenuti da izvršenje precizne klasifikacije korisnika zahtijeva i implementaciju ostalih metoda razmatranih u poglavlju 2.

Svojstvo	Numerička vrijednost	Značenje
Broj prekida unutar riječi	0	Objektivnost, racionalnost, logičnost
Nagib osnovne linija pisanja	1.0137°	Pesimizam, obeshrabrenost
Nagib slova	24.2312°	Sklonost procjenjivanju

Tabela 3.1: Rezultati ekstrakcije nekih od relevantnih parametara za verifikaciju korisnika i njihova značenja izvedena iz tabele 1.1.

DODATAK 1

Upute za korištenje napisanih funkcija

■ FUNKCIJA 1: Funkcija za analizu rukopisa i ekstrapolaciju 3D kretanja vrha olovke

Prototip funkcije je:

analizirajRukopis(slika).

Ulazni parametar *slika* predstavlja sliku pisanog teksta čija se analiza treba izvršiti. Funkcija nema izlaznih parametara, a kao rezultat njenog pozivanja vrši se iscrtavanje predviđene putanje pisanja po ekstraktovanoj konturi. Iako nema izlaznih parametara, nakon pozivanja funkcije moguć je rad sa konturom, te uvid u proračunate vrijednosti, kao što su najvjerojatniji redoslijed segmenata i kretanja po istim, te vrijednosti svakog od faktora relevantnih za funkciju cilja. Ove varijable snimljene su kao *.mat* datoteke i može se izvršiti njihovo učitavanje u bilo koji drugi MATLAB kod.

Prvi korak u analizi je pretprocesiranje slike. U ovom slučaju, ono podrazumijeva konverziju iz slike u boji u sliku sivih tonova, primjenom ugradene MATLAB funkcije *rgb2gray* nad originalnom slikom. Osim toga, obuhvata i promjenu dimenzija originalne slike, koje se povećavaju korištenjem funkcije *imresize*. Za određivanje praga korištena je lokalna metoda, implementirana funkcijom *odrediPrag*, čiji su ulazni parametri razmatrana slika, veličina lokalnog „prozora“, tj. dijela slike po kom će se vršiti usrednjavanje, te granična srednja vrijednost piksela u razmatranom „prozoru“. Nakon određivanja praga, vrši se filtriranje po srednjoj vrijednosti, pozivom ugradene funkcije *medfilt2*. Iako ova funkcija vrši (djelimično) odstranjivanje crnih piksela sa rubova slike, u tu svrhu se primjenjuje i funkcija *removeBlackCorners*, preuzeta sa web stranice *MathWorks-a*.

Uobičajeno je da se morfološke operacije vrše nad slikom sa invertovanom crnom i bijelom bojom, zbog čega se primjenjuje funkcija *not* nad *im2bw*, koja vrši konverziju slike proslijedene kao ulazni parametar u crno-bijelu sliku. Obje navedene funkcije su već postojeće, ugrađene MATLAB funkcije. Za izvršenje morfoloških operacija dilatacije i

zatvaranja također su korištene ugrađene funkcije, i to *imdilate* i *imclose*, a odgovarajući strukturni elementi su kreirani primjenom *strel* funkcije. Posljednji korak u preprocesiranju je smanjenje dimenzija slike (ponovnom primjenom funkcije *imresize*), čime se obezbjeđuje veća brzina izvršavanja algoritma. Nakon toga se vrši istanjivanje bijele konture prikazane na slici, na način opisan u *poglavlju 2*.

Funkcija *nadjipresjecneTacke* zadužena je za pronalaženje presječnih, tj. tačaka grananja istanjene konture koja joj se prosljeđuje kao ulazni parametar. Tačke se određuju izdvajanjem matrice dimenzija 3×3 oko svakog od bijelih piksela i poređenjem pozicije bijelih piksela unutar izdvojene matrice sa shemom rasporeda u kojima se mogu nalaziti pikseli u slučaju da je središnji piksel (oko kog je izdvojena matrica) tačka grananja. Rubne tačke konture nalaze se na sličan način, provjerom broja bijelih piksela koji su susjedni središnjem pikselu.

Za potrebe razdvajanje segmenata, svim elementima izdvojene matrice oko bijelog piksela detektovanog kao tačka grananja dodjeljuje se vrijednost crnog piksela. Kako su na ovaj način „izbrisane“ tačke grananja, vrši se detekcija novih rubnih tačaka nastalih ovim postupkom. Međutim, za ispravnu detekciju sada nije dovoljno samo provjeriti broj susjednih piksela, već se proces detekcije vrši korištenjem funkcije *odredirubneTacke*, koja kao ulazne parametre prima matricu dimenzija 5×5 izdvojenu oko tačke koja je bila detektovana kao tačka grananja, te njene koordinate x i y . Uzimajući u obzir da svi pikseli unutrašnje matrice dimenzija 3×3 imaju vrijednost crnih piksela, vrši se detekcija rubnih tačaka koje se nalaze u krajnjim redovima i kolonama matrice, kako je to već opisano u *poglavlju 2*. Sve tačke se sortiraju nalijevo primjenom funkcije *sortirajNalijevo*, koja prima vektor izdvojenih tačaka i sortira ih u rastućem redoslijedu prema x koordinati. Ovaj postupak se vrši radi poretka segmenata slijeva nadesno, za potrebe daljih proračuna.

Kreira se varijabla *kontura* tipa ćelije (eng. *cell*), čiji su atributi *sastavneTacke*, *duzinaPuta*, *karakteristicneTacke*, *kretanje* i *centarMase*. Izdvajanjem jednog po jednog bijelog piksela novodobivene slike, računaju se prethodno navedeni parametri, pri čemu se svi bijeli pikseli koji su uzeti u obzir brišu sa originalne slike, kako bi se onemogućio ponovni povratak na već razmatrane piksele. Parametri *karakteristicneTacke* i *kretanje* su složeni parametri, tipa ćelije, od kojih prvi sadrži informaciju o tome koje su tačke krajnje tačke segmenta i da li se granaju, a drugi nosi informaciju o dužini puta koji se prelazi ukoliko se radi o putanji s povratkom ili bez povratka. Primjena funkcije *sortirajKonture* podrazumijeva proračun parametra *centarMase*, s obzirom na to da se segmenti konture sortiraju prema kriteriju da su im centar mase i jedna krajnja tačka pozicionirane što više ulijevo. Proračun centra mase izvršen je na temelju postupka opisanog u [18]. Ulazni i izlazni parametar funkcije su istog tipa (*cell*), a izlazna ćelija sadrži iste segmente kao i ulazna, poredane u rastućem redoslijedu prema navedenom kriteriju.

Proračun najvjerovaljnijeg redoslijeda segmenata i pravca kretanja po istim vrši se nad reduciranim brojem segmenata konture. Proizvoljan broj segmenata (u aplikaciji je predviđeno da ovaj broj bude jednak 6) se prosljeđuje funkciji *funkcijaCilja* koja obavlja ovaj zadatak, na način koji će biti opisan u nastavku ovog poglavlja. Za konačni poredak se uzima u obzir samo poredak prva dva segmenta (sa pripadajućim redoslijedom kretanja), nakon čega se funkciji prosljeđuje naredni set segmenata, počevši od trećeg segmenta iz prethodnog seta. Pri vršenju proračuna za jedan set segmenata, funkcija

uzima u obzir poredak prethodnog seta. Nakon obrađivanja posljednjeg seta, dobiva se konačni redoslijed segmenata i kretanja po njima, te je omogućen vizuelni prikaz rezultata.

FUNKCIJA 2: Funkcija cilja

Prototip funkcije cilja je:

$$[vrijednost, W] = funkcijaCilja(rKontura, moguciindeks, moguciredoslijed, pockontura).$$

Značenja ulaznih parametara su sljedeća:

- *rKontura*: kontura koja sadrži reducirani broj segmenata originalne konture koja se analizira,
- *moguciindeks*: mogući redoslijed segmenata konture *rKontura*,
- *moguciredoslijed*: mogući redoslijed pravaca kretanja po segmentima, koji može poprimiti jednu od četiri vrijednosti, i to:
 - o 1: označava kretanje od tačke definirane kao prva tačka konture ka drugoj krajnjoj tački (shema 1-2),
 - o 2: označava kretanje od tačke definirane kao druga tačka konture ka prvoj krajnjoj tački (shema 2-1),
 - o 3: označava putanju s povratkom, gdje kretanje po segmentu počinje od tačke definirane kao prva tačka konture, usmjerava se ka drugoj krajnjoj tački, a potom se vrši povratak u početnu tačku po istoj putanji (shema 1-2-1),
 - o 4: označava putanju s povratkom, gdje kretanje po segmentu počinje od tačke definirane kao druga tačka konture, usmjerava se ka prvoj krajnjoj tački, a potom se vrši povratak u početnu tačku po istoj putanji (shema 2-1-2),
- *pockontura*: parametar koji ima vrijednost 0 ako reducirana kontura *rKontura* ne sadrži početni segment ili 1 ako sadrži.

Značenja izlaznih parametara su sljedeća:

- *vrijednost*: konačna vrijednost funkcije cilja, dobivena sumiranjem svih relevantnih parametara,
- *W*: vektor koji sadrži vrijednost svih parametara $w_i, i = [1,6]$, razmatranih u poglavljju 3.

Izbor početne tačke konture zavisi od pozicije krajnjih tačaka početnog segmenta u odnosu na apscisnu osu. Stoga se vrši poređenje vrijednosti koordinata krajnjih tačaka i pridružuju se odgovarajuće vrijednosti u zavisnosti od toga da li su pozicionirane više uljevo ili udesno. Manje vrijednosti odgovaraju većoj vjerovatnoći ispravne pretpostavke. Kako ovaj kriterij vrijeđi i pri izboru narednog segmenta u redoslijedu segmenata koji

predstavljaju putanju, vrši se proračun centra mase za svaki od segmenata i uzima se u obzir pozicija onih njihovih krajnjih tačaka koje su bliže koordinatnom početku (po apscisnoj osi). Prednost se daje onom segmentu čiji centar mase i krajnja tačka imaju manju vrijednost na apscisnoj osi.

Pri izboru početnog segmenta konture i njegove početne tačke od interesa je i da li je potencijalna početna tačka ujedno i tačka grananja. Ukoliko jeste, funkcija cilja poprima veću vrijednost.

Proračun ugla kretanja kroz presječne tačke podrazumijeva provjeru da li su razmatrane tačke klasificirane kao tačke grananja, što se vrši primjenom pomoćne funkcije *grananjeTacke*, koja koristi informacije sadržane u strukturi *rKontura*. U slučaju da je rezultat provjere pozitivan, traženi ugao dobiva se pozivanjem funkcije *ugaoKretanjaKrozPresjecneTacke*. Ova funkcija kao ulazne parametre prima četiri vrijednosti: *kontura1* i *kontura2* predstavljaju dva susjedna segmenta konture u mogućem redoslijedu segmenata koje *funkcijaCilja* prima kao parametre, *endtackak1* predstavlja završnu tačku prvog segmenta, dok je *starttackak2* početna tačka drugog segmenta proslijedenog kao parametar. Proračun ugla zahtijeva definiranje pravih koje zaklapaju ugao od interesa. Za prvi segment se pronalazi virtualna tačka, čije su koordinate jednake aritmetičkoj sredini koordinata prvih 15 tačaka koje su susjedne tački *endtackak1*, a za drugi segment vrijedi analogija, pri čemu su od interesa susjedi tačke *starttackak2*. Prave su određene krajnjim tačkama koje funkcija prima kao ulazne parametre, te njima odgovarajućim virtualnim tačkama. Ugao između njih računa se primjenom kosinusne teoreme.

Parametar koji nosi informaciju o dužini pređenog puta zahtijeva prvobitnu provjeru da li se radi o putanji s povratkom, nakon čega se dužina puta računa jednostavnim pozivanjem atributa *duzinaPuta* koji je sastavni dio svakog elementa strukture *rKontura*.

Kretanje olovke u vazduhu je parametar koji je relevantan isključivo onda kada se detektuje podizanje olovke s papira. Detekcija se vrši provjerom udaljenosti između segmenata od interesa i poređenjem njihove minimalne udaljenosti sa unaprijed zadanim pragom. Pomoćna funkcija koja se koristi u ovu svrhu je *udaljenostMedjuKonturama*, koja kao ulazne parametre prima dva segmenta, za koje računa udaljenosti između svake dvije krajnje tačke segmenata, tražeći njihovo minimalno rastojanje. Funkciji za proračun udaljenosti, nazvanoj *dajUdaljenost*, prosljeđuju se koordinate razmatranih tačaka, na osnovu kojih se traži pravolinjsko rastojanje između njih. Najmanje od tih rastojanja je rezultat koji vraća funkcija *udaljenostMedjuKonturama*. Nakon detekcije podizanja olovke s papira, računa se razlika pozicija susjednih segmenata (po apscisnoj osi). Prednost pri izboru narednog segmenta daje se onom koji ima najveću vrijednost po ordinatnoj osi (razlog je objašnjen u okviru poglavlja 3) i najmanje odstupanje po apscisnoj osi u odnosu na prethodni segment.

Da bi se izvršila eliminacija skokova, računa se udaljenost između svake dvije moguće početne i krajnje tačke segmenata koji se razmatraju. Faktor koji sadrži informaciju o skokovitoj putanji poprima veće vrijednosti za veće udaljenosti između tačaka. Za proračun udaljenosti koristi se ranije spomenuta pomoćna funkcija *dajUdaljenost*.

FUNKCIJA 3: Funkcija za normalizaciju

Prototip funkcije za normalizaciju je:

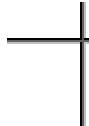
$$[brojPrekida, nagibOsnovneLinije, nagibSlova] = \text{normaliziraj}(slika).$$

Ulagani parametar *slika* predstavlja sliku pisanog teksta čija se normalizacija treba izvršiti. Značenja izlaznih parametara su:

- *brojPrekida*: broj razmaka unutar analizirane riječi,
- *nagibOsnovneLinije*: ugao nagiba osnovne linije pisanja izražen u stepenima, pri čemu pozitivne vrijednosti odgovaraju nagibu prema gore, a negativne nagibu prema dolje,
- *nagibSlova*: ugao nagiba slova pisanog teksta koji se analizira, izražen u stepenima, pri čemu pozitivne vrijednosti odgovaraju nagibu udesno, a negativne ukazuju na zakrenutost teksta u lijevu stranu.

Broj razmaka dobiva se korištenjem ugradene MATLAB funkcije *regionprops*. Prikaz regija unutar kojih se nalaze segmenti pisanog teksta u kojima nema razmaka može se dobiti promjenom vrijednosti parametra *prikaz* na 1.

Ugao nagiba osnovne linije pisanja nalazi se na osnovu crnih piksela slike (koje odgovaraju dijelu slike koja prikazuje pisani tekst) pozicioniranih na najvišim vrijednostima *y*-ose. Normalizacija nagiba slova vrši se na način opisan u poglavljju 2.1.1. Za potrebe proračuna relevantnih uglova, korišteni su dijelovi funkcija preuzetih sa *MathWorks*.



Zaključak

Analiza rukopisa je često sastavni dio istraživanja koja se provode u oblastima forenzičke, psihologije, ali i medicine, te brojnih drugih naučnih disciplina. Digitalna obrada slike omogućila je pojednostavljenje procesa analize u velikoj mjeri. Međutim, iako postoji dug niz pogodnosti koje nudi računarska analiza, pokazuje se da je u nekim segmentima nadmoćna analiza provedene od strane stručnjaka iz oblasti od interesa.

U ovom radu je pokazano kako se može izvršiti digitalna obrada slike rukopisa do te mjeru da se ostvari mogućnost izdvajanja konture pisanog teksta. Ovaj postupak zahtijeva kombiniranje metoda pretprocesiranja, kojim se u velikoj mjeri poboljšava kvalitet originalne slike, što predstavlja osnov za dalju obradu. Primjena morfoloških operacija nad slikom, uz korištenje metoda detekcije tačaka, linija i rubova, te istanjivanja ivica pokazala je dobre rezultate za ekstrakciju konture pisanog teksta. Kvalitet ekstraktovane konture narušen je samo onda kada je originalna slika degradirana šumom u tolikoj mjeri da ga nije moguće otkloniti primjenom univerzalnih metoda koje su obuhvaćene algoritmom za računarsku analizu rukopisa.

Izdvojena kontura predstavlja osnov za ekstrapolaciju 3D kretanja vrha olovke u analiziranom rukopisu. Ključna funkcija za proračun putanje kretanja olovke uzima u obzir poziciju početnih tačaka segmenata konture, vrši minimizaciju kretanja i promjene ugla kretanja vrha olovke, te detekciju podizanja olovke s papira. U radu je pokazano da se na ovaj način može izvršiti predviđanje stvarne putanje slikom prikazanog pisanog teksta. Međutim, postoje i neki karakteristični slučajevi kada algoritam ne daje vjerovatne rezultate. Ovo se prvenstveno odnosi na ona slova latiničnog pisma koja zahtijevaju pisanje zdesna nalijevo (npr. slovo *j*), te slova s tačkom (*i, j*). Da bi se riješili ovi problemi, potrebno je implementirati dodatne metode i ograničenja pri proračunu funkcije cilja.

Ekstrakcija parametara kao što su nagib osnovne linije pisanja, nagib slova pisanog teksta i broj razmaka unutar pisane riječi omogućavaju otkrivanje pojedinih karakterističnih osobina osobe čiji se rukopis analizira, ali je za ispravnu klasifikaciju korisnika potrebno uzeti u obzir i brojne druge parametre (pritisak olovke na papir, brzina pisanja i sl.), jer je bez njih klasifikacija nepotpuna, a samim time može biti i neispravna.

Uzimajući u obzir sve rezultate provedene analize, jasno je da ovaj sistem za računarsku viziju rukopisa, uz nadogradnju dodatnim metodama koje su u ovom radu izostavljene, može omogućiti preciznu klasifikaciju korisnika na temelju poznatih značenja ekstraktovanih parametara.



Lista funkcija

FUNKCIJA 1: analizirajRukopis

PROTOTIP: $\text{analizirajRukopis}(\text{slika})$

ULAZNI PARAMETRI: slika – SLIKA PISANOG TEKSTA PROIZVOLJNIH DIMENZIJA, ZA KOJU SE ŽELI UTVRDITI POTENCIJALNA PUTANJA PISANJA.

IZLAZNI PARAMETRI: FUNKCIJA NEMA IZLAZNIH PARAMETARA, A KAO REZULTAT NJENOG POKRETANJA VRŠI SE ISRČTAVANJE PREDVIĐENE PUTANJE PISANJA PO IZDVOJENOJ KONTURI SLIKE PROSLIJEĐENE KAO ULAZNI PARAMETAR.

FUNKCIJA 2: odrediPrag

PROTOTIP: $\text{prag} = \text{odrediPrag}(I, ws, C)$

ULAZNI PARAMETRI: I – SLIKA KOJA SE ANALIZIRA, ws – VRIJEDNOST KOJOM SE DEFINIRAJU DIMENZIJE FILTRA ZA USREDNJAVANJE, C – GRANIČNA SREDNJA VRIJEDNOST PIKSELA U OKVIRU RAZMATRANIH DIMENZIJA FILTRA.

IZLAZNI PARAMETRI: prag – VRIJEDNOST PRAGA DOBIVENA PRORAČUNOM NA OSNOVU ULAZNIH PARAMETARA.

FUNKCIJA 3: nadjiPresjecneTacke

PROTOTIP: $\text{presjecneTacke} = \text{nadjipresjecneTacke}(Istanjeno)$

ULAZNI PARAMETRI: $Istanjeno$ – OBRAĐENA VERZIJA ORIGINALNE SLIKE, NA KOJOJ SU PRIKAZANE ISTANJENE KONTURE.

IZLAZNI PARAMETRI: presjecneTacke – DETEKTOVANE TAČKE GRANANJA, SPREMLJENE U MATRICU ČJI SU ELEMENTI U PRVOJ KOLONI KOORDINATE PRESJEĆNIH TAČAKA PO x -OSI, A ELEMENTI U DRUGOJ KOLONI KOORDINATE PRESJEĆNIH TAČAKA PO y -OSI.

FUNKCIJA 4: odrediRubneTacke

PROTOTIP: $[rx, ry] = \text{odrediRubneTacke}(izdvoji, x, y)$

ULAZNI PARAMETRI: $izdvoji$ – IZDVOJENA MATRICA DIMENZIJA 5×5 OKO TAČKE ČJE SU KOORDINATE x I y , KOJE PREDSTAVLJaju DRUGA DVA ULAZNA PARAMETRA.

IZLAZNI PARAMETRI: rx – VEKTOR KOORDINATA x DETEKTOVANIH RUBNIH TAČAKA, ry – VEKTOR KOORDINATA y DETEKTOVANIH RUBNIH TAČAKA.

FUNKCIJA 5: sortirajNalijevo

PROTOTIP: $g = \text{sortirajNalijevo}(f)$

ULAZNI PARAMETRI: f – NIZ ČIJU SU ELEMENTI KOORDINATE TAČAKA, PRI ČEMU SU U PRVOJ KOLONI SMJEŠTENE KOORDINATE PO x -OSI, A U DRUGOJ KOLONI KOORDINATE PO y -OSI.

IZLAZNI PARAMETRI: g – SORTIRANI NIZ f , PRI ČEMU SE SORTIRANJE VRŠI PO KOORDINATAMA x -OSE, U RASTUĆEM REDOSLIJEDU.

FUNKCIJA 6: sortirajKonture

PROTOTIP: $g = \text{sortirajKonture}(f)$

ULAZNI PARAMETRI: f – KONTURA SLIKE, TIPO CELL, KOJU ČINI GRUPA SVIH NJENIH SASTAVNIH SEGMENTA.

IZLAZNI PARAMETRI: g – SORTIRANA KONTURA f , PRI ČEMU SE SORTIRANJE VRŠI U RASTUĆEM REDOSLIJEDU, NA OSNOVU CENTRA MASE I POZICIJA KRAJNJIH TAČAKA SASTAVNIH SEGMENTA ORIGINALNE KONTURE.

FUNKCIJA 7: funkcijaCilja

PROTOTIP: $[vrijednost, W] = \text{funkcijaCilja}(rKontura, moguciindeks, moguciredoslijed, pockontura)$

ULAZNI PARAMETRI: $rKontura$ – STRUKTURA KOJA SADRŽI REDUCIRANI BROJ SEGMENTA ORIGINALNE KONTURE SLIKE, $moguciindeks$ – MOGUĆI POREDAK INDEKSA SEGMENTA SADRŽANIH U STRUKTURI $rKontura$, $moguciredoslijed$ –

MOGUĆI POREDAK PRAVACA KRETANJA PO SEGMENTIMA, *pockontura* – PARAMETAR KOJI IMA VRJEDNOST 0 AKO REDUCIRANA KONTURA *rKontura* NE SADRŽI POČETNI SEGMENTILI 1 AKO SADRŽI.

IZLAZNI PARAMETRI: *vrijednost* – KONAČNA VRJEDNOST FUNKCIJE CILJA, DOBIVENA SUMIRANJEM SVIH RELEVANTNIH PARAMETARA, W – VEKTOR KOJI SADRŽI VRJEDNOST SVIH PARAMETARA $w_i, i = [1,6]$, RAZMATRANIH U POGLAVLJU 3.

FUNKCIJA 8: *grananjeTacke*

PROTOTIP: *tackaGrananja* = *grananjeTacke(rKonturaSegment, tacka)*

ULAZNI PARAMETRI: *rKonturaSegment* – SEGMENT ČIJA JE PRIPADAJUĆA TAČKA *tacka*, ZA KOJU SE PROVJERAVA DA LI JE TAČKA GRANANJA, *tacka* – PARAMETAR U KOM SU SADRŽANE KOORDINATE x I y TAČKE ZA KOJU SE PROVJERAVA DA LI SE GRANA.

IZLAZNI PARAMETRI: *tackaGrananja* – PARAMETAR KOJI POPRIMA VRJEDNOST 0 UKOLIKO RAZMATRANA TAČKA NIJE TAČKA GRANANJA, ILI VRJEDNOST 1 UKOLIKO JESTE.

FUNKCIJA 9: *ugaoKretanjaKrozPresjecneTacke*

PROTOTIP: *ugao* = *ugaoKretanjaKrozPresjecneTacke(kontura1, kontura2, endtackak1, starttackak2)*

ULAZNI PARAMETRI: *kontura1, kontura2* – DVA Izdvojena segmenta konture slike koja se analizira, *endtackak1* – završna tačka prvog segmenta, *starttackak2* – početna tačka drugog segmenta.

IZLAZNI PARAMETRI: *ugao* – izračunata vrijednost promjene ugla pri kretanju od završne tačke jednog segmenta do početne tačke drugog segmenta.

FUNKCIJA 10: *udaljenostMedjuKonturama*

PROTOTIP: *udaljenost* = *udaljenostMedjuKonturama(kontura1, kontura2)*

ULAZNI PARAMETRI: *kontura1, kontura2* – dva izdvojena segmenta konture slike koja se analizira.

IZLAZNI PARAMETRI: *udaljenost* – izračunata vrijednost udaljenosti između ulaznih segmenata.

FUNKCIJA 11: *dajUdaljenost*

PROTOTIP: *udaljenost* = *dajUdaljenost(x1, y1, x2, y2)*

ULAZNI PARAMETRI: *x1, y1* – koordinate prve tačke, *x2, y2* – koordinate druge tačke.

IZLAZNI PARAMETRI: *udaljenost* – izračunata vrijednost udaljenosti između dviše tačke čije su koordinate ulazni parametri *x1, y1, x2 i y2*.

FUNKCIJA 12: *funkcijaVizualizacije*

PROTOTIP: *funkcijaVizualizacije(kontura, indeksmin, redoslijedmin, delay)*

ULAZNI PARAMETRI: *kontura* – parametar tipa *CELL* koji predstavlja konturu originalne slike čija se analiza vrši, *indeksmin* – vektor koji sadrži indekse segmenata konture u najvjerovatnijem poretku (onom poretku za koji je funkcija cilja minimalna), *redoslijedmin* – vektor koji sadrži redoslijed kretanja po segmentima u najvjerovatnijem poretku, *delay* – opcioni parametar, koji određuje brzinu iscrtavanja, a čija je podrazumijevana vrijednost jednaka 0.01.

IZLAZNI PARAMETRI: funkcija nema izlaznih parametara, a kao rezultat njenog pokretanja vrši se iscrtavanje predviđene putanje pisanja po izdvojenoj konturi slike.

FUNKCIJA 13: *normaliziraj*

PROTOTIP: *[brojPrekida, nagibOsnovneLinije, nagibSlova] = normaliziraj(slika)*

ULAZNI PARAMETRI: *slika* – slika pisano teksta proizvoljnih dimenzija.

IZLAZNI PARAMETRI: *brojPrekida* – broj razmaka unutar slikom prikazanog teksta, *nagibOsnovneLinije* – ugao nagiba osnovne linije pisanja, izražen u stepenima, *nagibSlova* – ugao nagiba slova pisano teksta, izražen u stepenima.

FUNKCIJA 14: *korigujNagibSlova*

PROTOTIP: *ugao* = *korigujNagibSlova(slika)*

ULAZNI PARAMETRI: *slika* – slika pisano teksta proizvoljnih dimenzija.

IZLAZNI PARAMETRI: *ugao* – ugao nagiba slova pisano teksta, izražen u stepenima.

Lista slika

SLIKA 2.1 KORACI U PROCESU RAČUNARSKE OBRADE SLIKE	10
SLIKA 2.2: KORACI U PRETPROCESIRANJU SLIKE PISANOG TEKSTA	11
SLIKA 2.3 NORMALIZACIJA KONTRASTA	12
SLIKA 2.3 (A): ORIGINALNA SLIKA	
SLIKA 2.3 (B): SLIKA NAKON POVEĆANJA KONTRASTA	
SLIKA 2.4: NORMALIZACIJA NAGIBA OSNOVNE LINIJE PISANJA	13
SLIKA 2.4 (A): ORIGINALNA SLIKA	
SLIKA 2.4 (B): SLIKA NAKON NORMALIZACIJE NAGIBA OSNOVNE LINIJE PISANJA	
SLIKA 2.5: NORMALIZACIJA PODATAKA.....	13
SLIKA 2.5 (A): ORIGINALNA SLIKA	
SLIKA 2.5 (B): SLIKA NAKON POVEĆANJA KONTRASTA	
SLIKA 2.5 (C): SLIKA NAKON NORMALIZACIJE NAGIBA OSNOVNE LINIJE PISANJA	
SLIKA 2.5 (D): SLIKA NAKON NORMALIZACIJE NAGIBA SLOVA	
SLIKA 2.6: ZAGLAĐIVANJE SLIKE	14
SLIKA 2.6 (A): PODJELA SLIKE NA PIKSELE KOJIMA SE POSTUPCIMA UZORKOVANJA I KVANTIZACIJE DODJELIJU ODGOVARAJUĆE NIJANE SIVE BOJE	
SLIKA 2.6 (B): DIGITALNI PRIKAZ SLIKE U OBLIKU NIZA CIJELIH BROJAVA KOJIMA JE OPISANA BOJA SVAKOG POJEDINOG PIKSELA	
SLIKA 2.7: PRIMJENA MASKE ZA ODSTRANJIVANJE ŠUMA.....	15
SLIKA 2.7 (A): ORIGINALNA SLIKA NA KOJU JE DODAN TZV. „SALT AND PEPPER“ ŠUM	
SLIKA 2.7 (B): SLIKA NAKON FILTRIRANJA POMOĆU MASKE ZA ODSTRANJIVANJE ŠUMA	
SLIKA 2.8: PRIMJENA FILTRIRANJA PO SREDNJOJ VRIJEDNOSTI U SVRHU ODSTRANJIVANJA ŠUMA	16
SLIKA 2.8 (A): ORIGINALNA SLIKA NA KOJU JE DODAN TZV. „SALT AND PEPPER“ ŠUM	
SLIKA 2.8 (B): SLIKA NAKON FILTRIRANJA PO SREDNJOJ VRIJEDNOSTI	
SLIKA 2.9: DILATACIJA – OBJAŠNJENJE PRINCIPA DILATACIJE.....	17
SLIKA 2.9 (A): ORIGINALNA SLIKA	
SLIKA 2.9 (B): SLIKA NAKON PRIMJENE DILATACIJE (JEDNA ITERACIJA)	
SLIKA 2.9 (C): SLIKA NAKON PRIMJENE DILATACIJE (TRI ITERACIJE)	
SLIKA 2.10: DILATACIJA – REZULTAT PRIMJENE U ALGORITMU.....	18
SLIKA 2.10 (A): ORIGINALNA SLIKA	
SLIKA 2.10 (B): SLIKA NAKON VIŠESTRUKO PRIMJENE DILATACIJE	
SLIKA 2.11: EROZIJA.....	18
SLIKA 2.11 (A): ORIGINALNA SLIKA	
SLIKA 2.11 (B): IZDVAJANJE DOMINANTNIH ELEMENATA KORIŠTENjem STRUKTURNOG ELEMENTA ODGOVARAJUĆIH DIMENZIJA	
SLIKA 2.11 (C): PRIMJENA DILATACIJE NAD (B), UZ KORIŠTENje ISTOG STRUKTURNOG ELEMENTA	
SLIKA 2.12: MORFOLOŠKO ZATVARANJE.....	19
SLIKA 2.12 (A): ORIGINALNA SLIKA	
SLIKA 2.12 (B): SLIKA NAKON PRIMJENE MORFOLOŠKOG ZATVARANJA	

SLIKA 2.13: ISTANJIVANJE KONTURA.....	20
SLIKA 2.13 (A): ORIGINALNA SLIKA	
SLIKA 2.13 (B): SLIKA NAKON ISTANJIVANJA KONTURA	
SLIKA 2.14: KORACI U SEGMENTACIJI PODATAKA	20
SLIKA 3.1: KORACI U PRETPROCESIRANJU SLIKE – PRAKTIČNA REALIZACIJA	26
SLIKA 3.2: DETEKCIJA KARAKTERISTIČNIH TAČAKA KONTURE.....	28
SLIKA 3.3: IZDVAJANJE LINIJE KONTURE – PRAKTIČNA REALIZACIJA	28
SLIKA 3.4: PRIKAZ KOORDINATNOG SISTEMA PROGRAMSKOG PAKETA MATLAB	30
SLIKA 3.5: PRIKAZ REZULTATA PREDVIĐANJA PUTANJE PISANJA POMOĆU SISTEMA ZA ANALIZU RUKOPISA	31



Lista tabela

TABELA 1.1: OSOBINE RUKOPISA UZ OBJAŠNJENJE PRIPADAJUĆIH KARAKTERNIH OSOBINA.....	7
TABELA 3.1: REZULTATI EKSTRAKCIJE NEKIH OD RELEVANTNIH PARAMETARA ZA VERIFIKACIJU KORISNIKA I NJIHOVA ZNAČENJA IZVEDENA IZ TABELE 1.1	31

Literatura

- [1] J. U. McNeal, „Graphology: A New Marketing Research Technique“, Journal of Marketing Research, Vol. IV, pp. 363-367, 1967.
- [2] B. Zhang and S. N. Srihari, “Analysis of Handwriting Individuality Using Word Features”, 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, pp. 1142–1146, 2003.
- [3] V. Burr, “Judging Gender from Samples of Adult Handwriting: Accuracy and Use of Cues”, The Journal of Social Psychology, vol. 142, pp. 691–700, 2002.
- [4] M. Liwicki, A. Schlapbach, P. Loretan, and H. Bunke, “Automatic Detection of Gender and Handedness from On-line Handwriting”, Proc. 13th Conf. of the Int. Graphonomics Society, pp. 179–183, 2007.
- [5] V. Kamath, N. Ramaswamy, P. Navin Karanth, V. Desai and S. M. Kulkarni, „Development of an Automated Handwriting Analysis System“, ARPN Journal of Engineering and Applied Sciences, 2011.
- [6] Milan Sonka, Vaclav Hlavac, Roger Boyle: „Image Processing, Analysis, and Machine Vision“, Thomson, 2008.
- [7] Rafael C. Gonzalez, Richard E. Woods: „Digital Image Processing“, PEARSON Prentice Hall, 2001.
- [8] Edward A. Lee, Pravin Varaiya: „Structure and Interpretation of Signals and Systems“, Electrical Engineering & Computer Science University of California, Berkeley, 2000.
- [9] N. Arica, F. T. Yarman-Vural, „An Overview Of Character Recognition Focused On Off-line Handwriting“, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, pp. 216-233, 2001.
- [10] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins: „Digital Image Processing using MATLAB“, PEARSON Prentice Hall, 2009.
- [11] William K. Pratt, „Digital Image Processing“, PIKS Inside, Third Edition, 2001.
- [12] Luciano da Fontura Costa, Roberto Marcondes Cesar Jr.: „Shape Analysis and Classification: Theory and Practice“, CRC Press LLC, 2001.

- [13] S. Impedovo, B. Marangelli and A. Fanelli, "A Fourier Descriptor Set for Recognizing Nonstylized Numerals", IEEE Transactions on systems, man and cybernetics, vol. 8, pp. 640–645, 1978.
- [14] S. N. Srihari, S.-H. Cha, H. Arora and S. Lee, "Individuality of Handwriting", J Forensic Sci, vol. 47, pp. 856–72, 2002.
- [15] S. Loncaric, "A Survey of Shape Analysis Techniques", Pattern Recognition, vol. 31, pp. 983–1001, 1998.
- [16] Bernd Jahne, „Digital Image Processing“, 6th revised and extended edition, Springer-Verlag Berlin Heidelberg, 2005.
- [17] E. Sokic, A. Salihbegovic, M. Ahic-Djokic, „Analysis of Off-line Handwritten Text Samples of Different Gender using Shape Descriptors“, Telecommunications (BIHTEL), 2012 IX International Symposium, 2012.
- [18] Y. Mingqiang, K. Kidyo and R. Joseph, "A Survey of Shape Feature Extraction Techniques", Pattern Recognition, Peng-Yeng Yin (Ed.), pp. 43–90, 2008.