

**УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ**

Нена Дабих

СЕГМЕНТАЦИЈА СЛИКА У БОЈИ

дипломски рад

Бања Лука, март 2011.

Тема: СЕГМЕНТАЦИЈА СЛИКА У БОЈИ

Кључне ријечи:

Сегментација

Алгоритам нормализованих пресека

Сопствени вектор

Сопствена вриједност

Комисија: проф. др Петар Марић, председник

проф. др Зденка Бабић, ментор

мр Владимир Рисојевић, члан

Кандидат:

Нена Дабић

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
КАТЕДРА ЗА ОПШТУ ЕЛЕКТРОТЕХНИКУ

Предмет: ДИГИТАЛНА ОБРАДА СЛИКЕ

Тема: СЕГМЕНТАЦИЈА СЛИКА У БОЈИ

Задатак: Дати преглед основних приступа сегментацији слика у боји. Имплементирати одабрани алгоритам за сегментацију и тестирати га на примјерима. Анализирати утицај параметара алгоритма на његове перформансе.

Ментор: проф. др Зденка Бабић

Кандидат: Нена Дабић (35/05)

Бања Лука, март 2011.

Садржај

1.	УВОД	2
1.1.	Дефиниција сегментације	2
1.2.	Основне категорије метода за сегментацију слика у боји	3
1.3.	Преглед различитих приступа сегментације слика	4
1.3.1.	Сегментација помоћу прага	5
1.3.2.	Сегментација помоћу кластеризације	6
1.3.3.	Сегментација помоћу региона	6
1.3.4.	Сегментација помоћу граница региона	7
1.3.5.	Сегментација помоћу текстуре	8
1.4.	Пост-обрада сегментираних слика	8
1.4.1.	Означавање региона	9
1.4.2.	Бинарне операције	9
1.4.3.	Морфолошке операције	9
1.5.	Организација рада	9
2.	АЛГОРИТАМ НОРМАЛИЗОВАНИХ ПРЕСЈЕКА	11
2.1.	Формулација $Ncut$ критеријума	12
2.2.	Израчунавање минималне вриједности <i>нормализованог пресјека</i>	15
2.3.	Алгоритам груписања	17
2.3.1.	Реализација тежинског графа	18
2.3.2.	Рјешавање система сопствених вриједности	19
2.3.3.	Подјела графа кориштењем сопственог вектора са другом најмањом сопственом вриједности	20

2.3.4. Рекурзивно понављање подјеле графа	20
3. СЕГМЕНТАЦИЈА СИВИХ СЛИКА	24
3.1. Примјена алгоритма <i>нормализованих пресјека</i> на сиве слике	25
3.2. Утицај параметара алгоритма <i>нормализованих пресјека</i> на резултате сегментације сивих слика	28
4. СЕГМЕНТАЦИЈА СЛИКА У БОЈИ	31
4.1. HSI колор простор	32
4.1.1. Примјена алгоритма <i>нормализованих пресјека</i> на слике у боји представљене у HSI колор простору	34
4.1.2. Примјена алгоритма <i>нормализованих пресјека</i> на микробиолошке и хистолошке слике	38
4.2. Утицај параметара алгоритма <i>нормализованих пресјека</i> на резултате сегментације слика у боји	42
4.3. Вријеме извршавања сегментације	44
4.4. L*a*b колор простор	44
4.4.1. Примјена алгоритма <i>нормализованих пресјека</i> на слике представљене у L*a*b колор простору	45
5. ЗАКЉУЧАК	48
6. ЛИТЕРАТУРА	49
7. ДОДАТАК	51
Уз рад је приложен CD	

Захваљујем Николи Савићу

1. УВОД

Дигитална обрада слике је мултидисциплинирана област која прекрива различите аспекте оптике, електронике, математике, фотографије и рачунара. Области примјене дигиталне обраде слике су многобројне па се каже да је данас практично немогуће замислити било коју област људског инересовања гдје она није присутна.

У посљедњих неколико деценија, дошло је до наглог пораста броја техника и апликација у области дигиталне обраде слике. Међутим, дигитална обрада слике у боји је још увијек спорадично покривена, упркос чињеници да је уобичајна појава да потрошачи бирају погодности слика у боји над традиционалним сивим сликама [2]. С обзиром да сваким даном долази до све већег развоја технологије у скоро свим областима, самим тим расту и захтјеви у погледу развоја дигиталне обраде слика у боји, па у складу с тим настају многобројна рјешења везана за проблематику из различитих области. Талас настајања нових метода и апликација указује да ће се захтјеви за рјешавање проблема везаних за слике у боји, наставити и у наредним деценијама.

Основне области дигиталне обраде слике су репрезентација и моделирање слике, побољшање квалитета слике, рестаурација слике, анализа слике, реконструкција слике из пројекција и компресија слике. Поред наведених, као посебне области које су уско везане за дигиталну обраду слике, могу се издвојити рачунарска графика и рачунарски вид¹ [1]. У област анализе слике спада сегментација слика, која ће се детаљно обрађивати у овом дипломском раду, при чему је потребно одабрати одговарајући алгоритам на основу којег ће се реализовати код за сегментацију слика. Анализа реализованог алгоритма ће се вршити на синтетичким сликама, сликама из природе (пејзажи), те на неким хистолошким сликама. Већина тих анализа ће се односити на слике у боји, што је и тема самог дипломског рада.

Мотив због којег се детаљно приступа проблему сегментације слика је тај што он представља један од најактуелнијих проблема у дигиталној обради слике, те због многобројних области гдје се сегментација може примјенити. Неке од области примјене су у медицини, нпр. приликом пребројавања броја једара на хистолошким сликама, у војсци гдје се сегментација користи при препознавању летјелица или неких других објеката, у полицији гдје се користи за идентификацију особа на основу отисака прстију или слике, у метеорологији, индустријским и многим другим примјенама.

¹ Рачунарски вид (енгл. *computer vision*) - визуелна инспекција помоћу рачунара. Наука и технологија машине која „види“, гдје „види“ у овом случају подразумијева да је машина у могућности да извуче информације из слике које су потребне да би се ријешио неки задатак. Односно, она се бави развојем система који су у могућности да интерпретирају садржај природних сцена. Рачунарски вид има многобројне примјене (нпр. у пољу роботике се користи за реализацију очију робота) [1].

1.1. Дефиниција сегментације

У литератури се могу наћи различите дефиниције сегментације слика. У општем случају сегментација слика се дефинише као поступак којим се издвајају објекти од интереса од остатка слике. Главни циљ сегментације слике је партиционисање слике у различите регионе који су хомогени у односу на неке особине слике [2]. То је комплексан поступак који укључује компоненте које се односе на анализу боје, облика, покрета или текстуре објеката приказаних на слици. Поступак сегментације слика је први и најважнији корак у анализи слике и препознавању облика. Она представља критичну и суштинску компоненту у системима анализе слика и препознавања облика, и подразумијева један од најтежих задатака при обради слике, с обзиром да утиче на квалитет крајњих резултата у поступку обраде [3].

Већина поступака сегментације су хеуристичког карактера. Иако је сегментација најважнија фаза у анализи слике, до дан данас не постоји јединствена и опште прихваћена дефиниција сегментације. Осим тога, не постоји ни једначина за квантитативну процјену колико је поступак сегментације добар. Сегментацију слике R можемо формално да дефинишемо као поступак којим дијелимо слику на непреклапајуће регионе R_i , гдје је $i = 1, 2, \dots, n$, при чему n представља број непразних скупова. Уколико са $P(R_i)$ означимо предикат који се односи на униформност свих елемената унутар региона R_i , треба да буду задовољени сљедећи услови:

1. $\bigcup_{i=1}^n R_i = R$
2. $R_i \cap R_j = \emptyset$, за свако i и j , при чему важи $i \neq j$
3. $P(R_i) = TRUE$, за свако $i = 1, 2, \dots, n$
4. $P(R_i \cup R_j) = FALSE$, за свако $i \neq j$
5. R_i је повезан регион, за свако $i = 1, 2, \dots, n$ [4]

На основу претходних особина може се закључити да при поступку сегментације унија свих региона треба да буде цијела слика, затим да је пресјек два региона празан скуп, да регион треба да буде униформан према некој карактеристици, као и да се сваки регион разликује од осталих по тој истој карактеристици и да сви региони буду повезани.

Региони сегментоване слике имају многе интересантне карактеристике као што су боја, облик, текстура и друго. С обзиром да се слика може интерпретирати на различите начине (помоћу статистике, диференцијалне геометрије интензитета поља, парцијалних диференцијалних једначина, итд.), самим тим постоје многобројни приступи у рјешавању проблема сегментације, који се могу посматрати из различитих перспектива [5]. Проблем сегментације слике је врло распрострањено подручје у области дигиталне обраде слике које подразумијева стотине различитих алгоритама. Свака од техника сегментације заснована је на различитим математичким алатима, међутим ни за једну од тих метода не може се рећи да је у потпуности успјешна и да даје добре резултате за све класе слика. Према томе, сваки од датих алгоритама се примијењује у различитим апликацијама, а то значи да неки алгоритми дају боље резултате када је потребно извршити издвајање ивица, неки су

намијењени за рад са медицинским сликама, док неки од алгоритама су модификовани да дају добру сегментацију на основу текстуре, боје итд.

1.2. Основне категорије метода за сегментацију слика у боји

Иако постоје различите подјеле, међу свим постојећим методама сегментације колор слика, издвајају се четири основне категорије: технике на основу пиксела, технике на основу региона, технике на основу контура и хибридне технике [2]. Посљедња категорија се састоји од метода које интегришу двије различите технике, нпр. технике на основу пиксела и региона, као и од метода које истовремено користе како регионе тако и контуре. У том случају се реализују одвојене категорије техника сегментације које користе специјалне математичке алате (нпр. технике графова, *fuzzy* технике, математичка морфологија, или технике засноване на вјештачким неуронским мрежама).

С обзиром да су уређаји за аквизицију колор слика извори шума, врло је битно да се примјене предобрадни алгоритми засновани на различитим техникама за редукацију шума прије саме сегментације слике. Главни задатак предобrade је да се сузбије шум а да се истовремено очувају ивице. Нажалост, већина најчешће кориштених линеарних *smoothing*² филтара углађују слике, али исто тако врше замућење ивица. Због тога, најбољи резултати предобrade се добијају помоћу нелинеарних филтара који раде у просторном домену. Овакви филтри омогућавају очување ивица и детаља као и уклањање Гаусовог и импулног шума. Добри примјери оваквих филтара су медиан и Кувахара филтар и њихова примјена може значајно да побољша резултате сегментације слика. Поређење ових филтара је најчешће базирано на визуелном процјењивању или израчунавању различитих фактора квалитета.

У поступку сегментације, потребно је користити технике које ће давати задовољавајуће резултате. На самом почетку се поставља питање како изабрати „прави“ метод, при чему „прави“ метод треба да препозна одређени регион који се дефинише на основу одређених униформних обиљежја на слици. Такође, неопходно је да се тај метод може описати математички, као и да постоји критериј на основу којег ће се оцијенити квалитет извршене сегментације. Сам избор метода зависи и од онога ко приступа рјешавању датог проблема, односно колико је његово познавање неке технике, па у зависности од тога у различитим публикацијама се фаворизују одређене методе.

² *Smoothing* – глаткост, изравнавање, равнање

1.3. Преглед различитих приступа сегментације слика

У различитим апликацијама обраде слика у боји, велика пажња се придаје техникама за сегментацију слика. За њу се каже да је прва и најбитнија операција која се примјењује у анализи слике и рачунарском виду, односно сматра се да је сегментација први корак и кључно питање у областима препознавања објеката и разумијевања сцене и слика. Области примјене варирају од контроле индустријског квалитета, преко медицине, навигације робота, геофизичких истраживања, до примјена у војсци итд. У свим овим областима квалитет резултата у даљој обради слике великим дијелом зависи управо од квалитета извршене сегментације [6].

Проблем сегментације у општем случају је до дан данас остао битно истраживачко поље и многе методе сегментације су уназад неколико деценија предложене у разним публикацијама. Први научни радови у области сегментације слика у боји датирају још из осамдесетих година прошлог вијека. Већ у првим књигама које се односе на обраду слика у боји, могу се пронаћи поглавља која су намјењена искључиво сегментацији колор слика. Скоро све технике за сегментацију сивих слика, које су се развиле у ранијем периоду, нашле су примјену и у сегментацији колор слика. Свака таква експанзија на сликама у боји је повезана са избором колор простора. Интересантно је поменути да се у неким ситуацијама, када се врши сегментација слика у боји, добијају знатно бољи резултати него када се иста та техника примјењује на сивим сликама, Слика 1.1. На основу овог се може закључити колике су све могућности обраде колор слика.

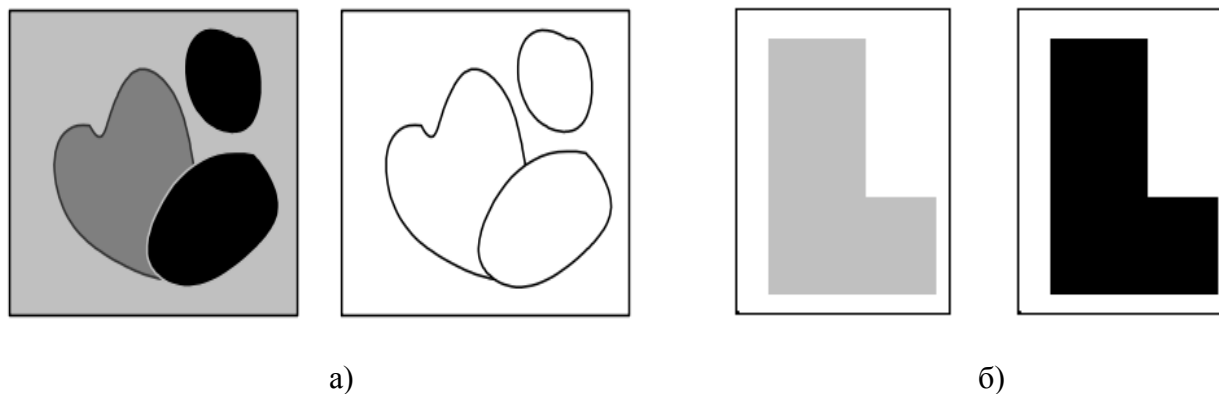


Слика 1.1 – а) издвојени региони на сивој слици, б) издвојени региони на колор слици

У случају када се након сегментације добије много малих региона, који одговарају хомогеним објектима у оригиналној слици, кажемо да је дошло до претјеране сегментације. Термин који се користи да опише ову појаву назива се *oversegmentation*. С друге стране, уколико слика након сегментације садржи неколико великих региона, при чему сваки регион одговара двама или више објеката у оригиналној слици, долази до појаве која се назива *undersegmentation*, која подразумева да сегментација није извршена до краја. Грешке до којих долази при сегментацији слике (*oversegmentation* и *undersegmentation*) су узрок неправилности које настају у даљој анализи и обради слике. Али упркос томе, појава претјеране сегментације је погоднија у даљој обради слике, јер се она може

„поправити“ кориштењем одговарајућих постобрадних техника након којих се број региона на слици може смањити.

У литератури се могу наћи различите подјеле када су у питању технике које се примјењују при сегментацији слика. Међутим, када се говори о основној подјели, издвајају се двије најчешће кориштене технике: проналажење ивица – издвајање само оних пиксела који припадају рубовима објеката и издвајање региона – издвајање цијелог објекта од позадине придружујући пикселе чија је свјетлина испод неког прага позадини а остале објекту и обрнуто, Слика 1.2.



Слика 1.2 - а) издвајање ивица б) издвајање објеката од позадине

Неке подјеле су опширнијег карактера, и узимају у обзир знатно већи број приступа сегментације колор слика. У једну такву подјелу спадају сљедеће методе: сегментација помоћу прага, сегментација помоћу кластеризације, сегментација помоћу региона, сегментација помоћу граница региона и сегментација текстуре [7].

1.3.1. Сегментација помоћу прага

Сегментација помоћу прага представља групу метода заснованих на поређењу освјетљености пиксела са једним или више прагова, па у том случају имамо сегментацију са једним и сегментацију са више прагова. Уколико имамо слике у боји тада се врши поређење боје између пиксела. Сегментација са једним прагом представља најједноставнији вид сегментације и његова основна примјена је за одвајање објекта од позадине, уколико позадина има униформну освјетљеност (боју) која се разликује од објекта. Овакав начин сегментације се може примјенити, нпр., код издвајања писаног или штампаног текста, анализе неких биомедицинских слика, препознавања типа авиона који лети, итд. Сегментација са више прагова је метода коју је погодно користити у случају када имамо сцене са више различитих објеката на позадини.

1.3.2. Сегментација помоћу кластеризације

Сегментација помоћу кластеризације³ је једна од најстаријих техника сегментације, с обзиром да се може користити за сегментацију сивих и колор слика. Нека је дат вектор $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$, који представља мјерења карактеристика једног пиксела слике. Мјерене величине могу бити три компоненте боје пиксела, или нека друга обиљежја израчуната у малом прозору око посматраног пиксела. Поступак сегментације се у том случају састоји од подјеле N -димензионалног простора у узајамно искључиве области, при чему свака та област обухвата групу података која се односи на један регион слике. Такав процес подјеле према сличности вектора података се назива *кластеризација*. У првој фази сегментације врши се израчунавање и издвајање неких обиљежја слике. Сљедећа фаза подразумијева одређивање оптималног броја региона (кластера), као и одређивање скупа вектора који представљају центре сваког кластера. Након тога се врши придруживање сваког пиксела једном од кластера. Придруживање се врши на основу сличности вектора који описује пиксел и вектора који описује центар кластера. Ова метода је са доста успјеха примјењивана у сегментацији мултиспектралних сателитских или авионских снимака, гдје је основни циљ био издвајање региона на површини земље који имају исте карактеристике, као што су нпр., исти тип пољопривредних култура, исти састав земљишта, иста надморска висина, итд. Недостатак ове методе је тај што има велику рачунску сложеност. Алгоритам који ће се обрађивати у овом дипломском раду представља један тип сегментације помоћу кластеризације. Он се заснива на теорији графова, према којој се слика посматра као граф, док су пиксели слике чворови графа. Циљ кориштења овог алгоритма је да се испита његова примјењивост за сегментацију слика општег карактера као и неких медицинских слика.

1.3.3. Сегментација помоћу региона

Сегментација помоћу региона представља скуп метода које покушавају да издвоје подручја слике која су хомогена са гледишта одређених карактеристика. У ову групу спадају: сегментација помоћу раста региона и сегментација помоћу раздвајања и спајања региона.

Основна идеја сегментације помоћу раста региона је да се изврши груписање сусједних пиксела сличних освјетљености (боја), на основу чега се формирају региони. Поступак груписања почиње спајањем по два пиксела истих карактеристика, при чему настаје *атомски регион*. Након тога се посматрају два сусједна региона R_1 и R_2 , чији су обими (број ивичних пиксела) означени са P_1 и P_2 . Нека C представља дужину заједничке границе региона, а D дужину заједничке границе гдје је разлика између карактеристика

³ *Cluster* – група, скуп *clustering* - груписање

пиксела са обје стране границе мања од унапријед дефинисане вриједности ε_1 . Кажемо да ће се региони R_1 и R_2 спојити уколико важи:

$$\frac{D}{\min(P_1, P_2)} > \varepsilon_2$$

гдје је ε_2 константа чија је вриједност најчешће једнака 0.5. Затим се врши испитивање осталих *атомских региона*, након чега се прелази на испитивање региона већих димензија све док је спајање могуће. На овај начин се омогућава прикључивање мањих региона већим, међутим спајање региона сличних димензија није дозвољено. Да би се омогућило спајање два региона сличних величина који су раздвојени тзв. „слабом“ границом, мора бити испуњен следећи услов:

$$\frac{D}{C} > \varepsilon_3$$

гдје је ε_3 константа чија је вриједност најчешће једнака 0.75. Метода раста региона је нарочито погодна код сегментације простих сцена са малим бројем објеката и слабом текстуром.

Код сегментације помоћу раздвајања и спајања региона, слика се дијели на четири квадранта, након чега се испитује да ли су добијени подсегменти униформни по обиљежју на основу којег се врши сегментација. Униформност се може дефинисати као разлика између најмање и највеће освјетљености пиксела у региону, преко варијансе освјетљености или преко неке друге статистичке мјере. Сваки подсегмент, за који није задовољен услов униформности, се даље дијели на четири нова подсегмента. Након што се изврши раздвајање, прелази се на обрнути поступак – спајање региона. Региони који имају исто униформно обиљежје спајају се у већи регион. Недостатак ове методе је у томе што се јавља видљива блоковска структура региона.

1.3.4. Сегментација помоћу граница региона

Методе које врше издвајање ивица генеришу испрекидане границе објеката а не затворене криве. Због тога се у процесу сегментације који користи ивице мора примјенити и неки поступак за спајање ивица, како би се формирале непрекидне границе региона. У ту сврху се могу користити: метода спајања ивица фитовањем криве, спајање ивица хеуристичким методама и спајање ивица Хафовом трансформацијом.

Методом фитовања кривих се могу реконструисати недостајући сегменти уколико мапа ивица садржи прекинуте ивице. Фитовање кривих је најједноставније када нам је познат облик криве, па се недостајући сегмент добија фитовањем нпр. правих или кружних облика. Када су границе региона сложеније, врши се разлагање границе на једноставније дијелове. Главна предност овог метода је њена изузетна једноставност и она даје добре резултате у случају сегментације једноставних сцена. Међутим, када имамо

слике на којима постоји већи број објеката који се преклапају, мапа ивица садржи тачке гранања, па у том случају ова метода не даје добре резултате.

Код хеуристичких метода за спајање ивица се прво формира градијентна слика примјеном одговарајућих маски. Ивице које се добијају могу бити испрекидане, или се може десити да двије ивичне тачке буду спојене на више начина. Кратки прекиди се могу премостити правим линијама, а вишеструке везе се могу елиминисати одбацавањем дуже странице, ако је вишеструка веза у виду троугла, или замјеном четвороугла његовом дужом дијагоналном, када је вишеструка веза четвороугао.

Хафова трансформација (*Hough*) је пресликавање линије из правоуглог координатног сиситема у тачку у поларном координатном систему, па се права линија може представити у параметарском облику:

$$\rho = x\cos\theta + y\sin\theta$$

гдје је ρ растојање од координатног почетка до праве линије дуж нормале на линију, а θ је угао између нормале и x -осе. Прву примјену Хафове трансформације за спајање ивица су описали *Duda* и *Hart*, која је детаљније објашњена у литератури [7].

1.3.5. Сегментација помоћу текстуре

Сегментација помоћу текстуре се заснива на израчунавању неке мјере текстуре за сваки пиксел слике. Основни проблем ове методе је у томе што текстура није локална карактеристика, већ се увијек израчунава унутар прозора у чијем се центру налази пиксел који се посматра, па због тога израчунавање мјера текстуре на граници два региона са различитим текстурама није поуздано. Сегментација текстуре се може заснивати и на методама за издвајање ивица, тако што се прво истакну разлике између текстура, што одговара израчунавању градијента, а затим се врши класификација пиксела. У поступку сегментације текстуре могу се користити и модели текстуре. Најважније примјене сегментације слике помоћу текстуре су у анализи авионских и сателитских слика. На таквим сликама постоје региони са израженом текстуром који представљају планине, шуме или обрадиве површине. Ова метода такође има велику примјену при анализи рендгенских и ултразвучних снимака у медицинској дијагностици.

1.4. Пост-обрада сегментираних слика

Након извршене сегментације, некад је потребно примјенити пост-обрадне поступке помоћу којих се побољшавају резултати сегментације. Неки од пост-обрадних поступака су наведени у наредном тексту.

1.4.1. Означавање региона

Када се у поступку сегментације добије бинарна слика, врши се означавање пиксела, односно потребно је одредити којем региону они припадају. Најједноставнија метода у том случају је да се сваком пикселу придружи ознака или индекс сегмента. Поред овог, постоји много ефикаснији метод који се огледа у томе да се специфицира затворена контура која ограничава неки регион, па да се након тога изврши обиљежавање сваког пиксела унутар те контуре. Постоји неколико различитих метода за спецификацију затворене контуре, које су детаљније објашњене у литератури [7].

1.4.2. Бинарне операције

Након што се у поступку сегментације добије бинарна слика, односно слика чији пиксели имају само двије вриједности, приступа се њеној обради у циљу поправљања квалитета слике и отклањања грешака насталих при сегментацији. Бинарне операције се састоје у интеракцији бинарне слике и структурног елемента којом се мијења облик објекта. Структурни елемент чини група пиксела која формира одређени облик (квадрат, круг, троугао, итд.). Један од пиксела представља центар структурног елемента. Бинарне операције код којих се централном пикселу додјељује вриједност 1 (црна боја) називају се адитивне операције, док се бинарне операције код којих се централном пикселу додјељује вриједност 0 (бијела боја) називају субтрактивне операције. Њихово детаљније објашњење је дато у литератури [7].

1.4.3. Морфолошке операције

У обради слике морфолошке операције представљају поступак обраде којим се мијења облик или структура објекта на слици. Овакве операције су засноване на математичким теоријама скупова и топологије. Основне морфолошке операције су ерозија и дилатација, док у сложеније спадају операције отварања и затварања, „Hit and Miss“ операција, условна дилатација и многе друге.

1.5. Организација рада

У раду је имплементиран алгоритам за сегментацију слика, гдје је сегментација описана на основу критеријума који се зове *нормализовани пресјек* [8]. Овај критеријум се заснива на теорији графова, при чему се сегментација слике посматра као проблем партиционисања графа. У уводној глави је дефинисан појам сегментације, њене основне карактеристике, те многобројне могућности примјене. Такође, дат је преглед основних приступа који се могу користити при сегментацији слика.

Глава 2 – У овој глави је представљен алгоритам *нормализованих пресјека*, гдје је детаљно објашњен принцип по којем он врши сегментацију слике, која се посматра као граф. Такође, дато је математичко извођење алгоритма, начин на који се врши минимизација вриједности критеријума за сегментацију, те објашњење да се ова

минимизација може посматрати као проблем уопштеног система сопствених вриједности. Описани су основни параметри овог алгоритма, њихове основне карактеристике као и начин на који се они примјењују. На крају је дато детаљно објашњење свих корака из којих се састоји кориштени алгоритам.

Глава 3 – Приказује примјену алгоритма *нормализованих пресјека* на сивим сликама, при чему се детаљно описују добијени резултати. Наведене су основне карактеристике функција које утичу на формирање тежинске матрице, као и услови који при томе треба да буду задовољени. Такође је објашњен начин на који се мијењају резултати сегментације, у зависности од промјене кориштених параметара.

Глава 4 – Унутар ове главе су дати резултати примјене алгоритма *нормализованих пресјека* на слике у боји, што представља централни задатак дипломског рада. Дефинисане су основне карактеристике кориштених колор простора, те је дат детаљан опис утицаја параметара при сегментацији колор слика. Тестирања су вршена на различитим класама слика, гдје се на основу добијених резултата може донијети закључак о ефикасности алгоритма, као и на којим класама слика се добија најбоља сегментација. Такође су описани и разлози због којих се јавља проблем дугог извршавања алгоритма.

Глава 5 – Даје се закључак рада до којег се дошло на основу свих извршених тестирања. Наводе се могућа рјешења и идеје за побољшање описаног алгоритма.

2. АЛГОРИТАМ НОРМАЛИЗОВАНИХ ПРЭСЈЕКА

Као што је наглашено у претходној глави, при сегментацији слике, потребно је одабрати такав алгоритам који ће за одређене класе слика давати најбоље резултате. Када се приступа неком проблему сегментације, на самом почетку је неопходно дефинисати који су то објекти од интереса на слици, као и одредити њихова основна обиљежја, како би се на основу тих информација одабрао одговарајући алгоритам. У претходној глави су набројани различити приступи сегментације, при чему је наглашено да ће се у овом дипломском раду користити приступ који је заснован на теорији графова, при чему се сегментација слике третира као проблем подјеле графа. Прије него што се пређе на детаљнији опис овог приступа, дефинисаћемо сам појам графа и његове основне карактеристике. Описно говорећи, графови су фигуре састављене од тачака од којих су неке спојене кривим линијама. Графови се сусрећу у различитим гранама науке, нпр. у хемији као структурне формуле молекула, у електротехници као шеме електричних кола, у компјутерским наукама као дијаграми компјутерских програма, итд. Поред свих наведених, графови налазе примјену и у области сегментације слика, што ће се овдје детаљније објаснити. Граф се може дефинисати као апстрактни математички објекат, а фигура састављена од тачака и линија је, у ствари, геометријска представа или цртеж графа. Уколико се појам графа повезује са појмом бинарне релације, онда се граф дефинише на сљедећи начин: *Нека је X непразан скуп и ρ бинарна релација у X . Уређен пар $G = (X, \rho)$ се назива граф. Елементи скупа X су чворови графа, а елементи скупа ρ гране графа [9].*

У графовском приступу сегментације скуп тачака у простору произвољних особина је представљен као тежински неусмјерени граф $G = (V, E)$, гдје су чворови графа тачке у назначеном простору, при чему се између свака два пара чворова формира грана графа. Тежина сваке гране, $w(i, j)$, је функција сличности између чворова i и j . Задатак у поступку груписања је да се скуп чворова подјели у дисјунктне скупове V_1, V_2, \dots, V_m , гдје је према некој мјери сличност између чворова у скупу V_i велика, а између различитих скупова V_i и V_j мала. Да би се извршила подјела графа, потребно је поставити сљедећа питања:

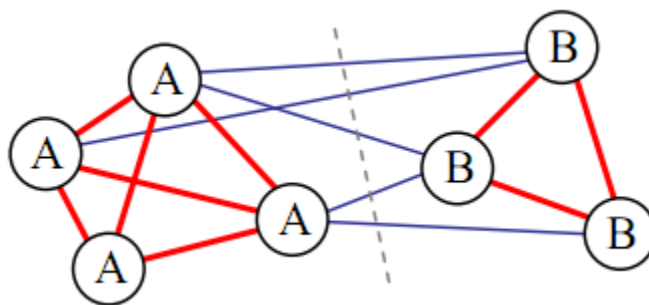
1. Шта је прецизан критеријум за добру подјелу?
2. Како таква подјела може бити ефикасно израчуната?

Овај дипломски рад се бави критеријумом који је заснован на теорији графова и који има задатак да оцјени колико добро је извршена подјела слике. Изворни назив тог критеријума је *нормализовани пресјек*, по коме је име добио и сам алгоритам. Технику сегментације слика, која је заснована на овом критеријуму, су први пут 2000. године представили *Jianbo Shi* и *Jitendra Malik*, чија су примарна истраживања управо била у области сегментације слика, препознавања објеката, анализе покрета и облика, груписања, рачунарског вида,

рачунарског моделовања људског вида и слично. За разлику од неких других алгоритама, као што је нпр. *mean-shift* алгоритам који тражи кластере са сличним пикселима користећи мод за претраживање или нпр. *bottom-up merging* техника која скупља регионе у кохерентне цјелине, алгоритам *нормализованих пресејака* испитује сличност између блиских пиксела и покушава да раздвоји оне групе пиксела између којих постоји слаба веза сличности [10].

2.1. Формулација N_{cut} критеријума

Нека се посматра граф приказан на Слици 2.1. Сви пиксели у скупу А имају велик степен сличности, што је означено црвеном дебелом линијом. Исто вриједи и за пикселе у скупу В. Повезаност између ове двије групе, која је означена плавом тањом линијом, је много слабија. *Нормализовани пресејак* (црткана линија) раздваја скупове А и В на два кластера, односно скупа.



Слика 2.1 – Примјер графа и његов *нормализовани пресејак*

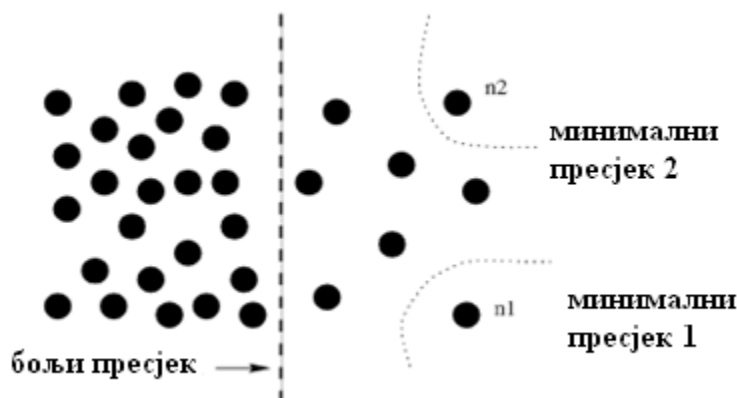
Ако граф са Слике 2.1 означимо са $G = (V, E)$, закључује се да он може бити подијељен у два дисјунктна скупа А и В ($A \cup B = V, A \cap B = \emptyset$) једноставним уклањањем грана које повезују два скупа. Степен различитости између ова два скупа се може израчунати као укупна тежина грана које су уклоњене. У теорији графова, та величина се назива *пресејак* и описује се сљедећом формулом:

$$cut(A, B) = \sum_{i \in A, j \in B} w(i, j) \quad (2.1)$$

Оптимална подјела графа на два дијела је она која минимизира вриједност *пресејака*. Иако постоји велики број оваквих подјела, тражење минималне вриједности *пресејака* графа је веома добро проучен проблем и постоје ефикасни алгоритми за њено рјешавање.

Метод кластеризације заснован на критеријуму минималног *пресејака* су предложили Z. Wu и R. Leahy, који су у ствари покушавали да изврше подјелу графа на k подграфа тако да максимални *пресејак* свих подграфа буде минимизован [8]. Овај проблем може бити ефикасно ријешен рекурзивним проналажењем минималних вриједности *пресејака* који дијеле постојеће сегменте. Z. Wu и R. Leahy су у свом раду показали да кориштење овакавог оптималног глобалног критеријума даје добру

сегментацију на неким сликама. Међутим, недостатак оваквог приступа је у томе што критеријум минималног *пресјека* доводи до издвајања мањих група изолованих чворова у графу. Ово је очекивана последица с обзиром на саму дефиницију вриједности *пресјека* дату изразом (2.1). На основу ове дефиниције се може закључити да се са повећавањем броја грана између два скупа вриједност *пресјека* повећава. На Слици 2.2 је илустрована таква ситуација.



Слика 2.2 – Примјер лоше подјеле у случају кориштења минималног *пресјека*

Уколико се претпостави да су тежине грана инверзно пропорционалне растојању између два чвора, може се примјетити да ће *пресјек*, који дијели чвор n_1 или n_2 , имати веома малу вриједност. У суштини, било који *пресјек* који дијели индивидуалне чворове на десној половини ће имати мању вриједност него *пресјек* који дијели чворове на лијеву и десну половину. Како би се спријечило издвајање мањих скупова тачака, *Shi* и *Malik* су предложили нову мјеру раздвајања између двије групе чворова. Према томе, умјесто да се посматра укупна тежина грана које повезују двије подјеле, нова мјера рачуна *пресјек* као количник укупне повезаности грана према свим чворовима у графу. Ова мјера дисоцијације (раздвајања) се назива *нормализовани пресјек* (N_{cut}), што се математички описује на сљедећи начин:

$$N_{cut}(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \quad (2.2)$$

гдје је $assoc(A, A) = \sum_{i \in A, j \in A} w_{ij}$ асоцијација (сума свих тежина) унутар подграфа, док је $assoc(A, V) = assoc(A, A) + cut(A, B)$ сума свих тежина графа које су повезане са чворовима у скупу A . На идентичан начин се дефинише и $assoc(B, V)$. Са оваквом дефиницијом раздвајања између два скупа, *пресјек* који партиционира мање скупове изолованих чворова више неће имати малу N_{cut} вриједност, с обзиром да ће вриједност *пресјека* скоро сигурно чинити велики проценат укупне повезаности између тог малог скупа и свих осталих чворова.

На исти начин се може дефинисати и мјера за укупну нормализовану асоцијацију унутар групе за дату партицију, при чему она одражава колико чврсто су чворови унутар групе повезани. Ова мјера се описује следећим изразом:

$$N_{assoc}(A, B) = \frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)} \quad (2.3)$$

Битна карактеристика асоцијације и дисоцијације је да су оне међусобно повезане што је описано помоћу следећег израза:

$$\begin{aligned} N_{cut}(A, B) &= \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \\ &= \frac{assoc(A, V) - assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, V) - assoc(B, B)}{assoc(B, V)} \\ &= 2 - \left(\frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)} \right) \\ &= 2 - N_{assoc}(A, B) \end{aligned} \quad (2.4)$$

Према томе, критеријуми N_{cut} и N_{assoc} , на основу којих се формирају двије подјеле при кориштењу алгоритма за сегментацију, су идентични и могу бити задовољени истовремено. Они се могу одредити минимизацијом дисоцијације између група (N_{cut}) или одређивањем максималне асоцијације унутар група (N_{assoc}). Ми смо у овом дипломском раду као критеријум подјеле графа користили *нормализовани пресјек*.

Табеларни приказ асоцијација и *пресјека* се може видјети у Табели 2.1. На основу табеле може се примјетити да се дијелењем елемената табеле са редом или колоном *sum* добијају нормализоване вриједности *пресјека* и асоцијације.

	A	B	sum
A	$assoc(A, A)$	$cut(A, B)$	$assoc(A, V)$
B	$cut(B, A)$	$assoc(B, B)$	$assoc(B, V)$
sum	$assoc(A, V)$	$assoc(B, v)$	

Табела 2.1 – Табеларни изглед асоцијација и *пресјека* за граф на Слици 2.1

2.2. Израчунавање минималне вриједности нормализованог пресека

У циљу добијања минималне вриједности *нормализованог пресека* посматра се граф V , који је подијељен у два скупа A и B . Нека је \mathbf{x} вектор лабела чија је димензија $N = |V|$, и нека је $x_i = 1$ уколико чвор i припада скупу A , а -1 у супротном случају. Нека је \mathbf{W} симетрична матрица реда $N \times N$, при чему је $W(i, j) = w_{ij}$ и нека је $\mathbf{d}(i) = \sum_j w(i, j)$ укупна повезаност између чвора i и свих осталих чворова, односно укупна сума i -тог реда матрице \mathbf{W} . Помоћу овако дефинисаних \mathbf{x} и \mathbf{d} , $N_{cut}(A, B)$ се може записати на другачији начин:

$$\begin{aligned} N_{cut}(A, B) &= \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \\ &= \frac{\sum_{(x_i > 0, x_j < 0)} -w_{ij} x_i x_j}{\sum_{x_i > 0} \mathbf{d}_i} \\ &\quad + \frac{\sum_{(x_i < 0, x_j > 0)} -w_{ij} x_i x_j}{\sum_{x_i < 0} \mathbf{d}_i} \end{aligned} \tag{2.5}$$

Нека је \mathbf{D} дијагонална матрица реда $N \times N$, чији су елементи на дијагонали одређени са \mathbf{d} ,

$$k = \frac{\sum_{x_i > 0} \mathbf{d}_i}{\sum_i \mathbf{d}_i}$$

и нека је $\mathbf{1}$ вектор реда $N \times 1$ који садржи све јединице. Користећи чињеницу да су $\frac{1+x}{2}$ и $\frac{1-x}{2}$ лабеле за $x_i > 0$ и $x_i < 0$, једначина (2.5) се може написати у облику:

$$\begin{aligned} N_{cut}(A, B) &= \frac{(\mathbf{1} + \mathbf{x})^T (\mathbf{D} - \mathbf{W}) (\mathbf{1} + \mathbf{x})}{k \mathbf{1}^T \mathbf{D} \mathbf{1}} + \frac{(\mathbf{1} - \mathbf{x})^T (\mathbf{D} - \mathbf{W}) (\mathbf{1} - \mathbf{x})}{(1 - k) \mathbf{1}^T \mathbf{D} \mathbf{1}} \\ &= \frac{(\mathbf{x}^T (\mathbf{D} - \mathbf{W}) \mathbf{x} + \mathbf{1}^T (\mathbf{D} - \mathbf{W}) \mathbf{1})}{k(1 - k) \mathbf{1}^T \mathbf{D} \mathbf{1}} + \frac{2(1 - 2k) \mathbf{1}^T (\mathbf{D} - \mathbf{W}) \mathbf{x}}{k(1 - k) \mathbf{1}^T \mathbf{D} \mathbf{1}} \end{aligned}$$

Након увођења одговарајућих смјена, при чему се користе одређена математичка правила и трансформације, може се показати да је минимизовање *нормализованог пресека* по свим могућим лабелама вектора \mathbf{x} еквивалентно минимизовању израза $\frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}}$, па у том случају имамо следећу једнакост:

$$\min_{\mathbf{x}} N_{cut}(\mathbf{x}) = \min_{\mathbf{y}} \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}} \quad [8] \quad (2.6)$$

гдје је $\mathbf{y} = (\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x})$ вектор који узима вриједности 1 и $-b$, при чему је $b = \frac{k}{1-k}$. Са овако дефинисаним вектором \mathbf{y} лако се показује да вриједи $\mathbf{y}^T \mathbf{D} \mathbf{1} = 0$. Детаљно извођење и начин добијања минимизоване N_{cut} вриједности, као и остала математичка објашњења *Shi* и *Malik* су описали у свом раду [8].

Уколико се обрати пажња на израз (2.6), може се примјетити да је он идентичан изразу за Рајлијев количник [8]. Водећи се тиме, ако вектор \mathbf{y} узима реалне вриједности, израз (2.6) се може минимизовати рјешавањем уопштеног система сопствених вриједности, који је дат са:

$$(\mathbf{D} - \mathbf{W}) \mathbf{y} = \lambda \mathbf{D} \mathbf{y}. \quad (2.7)$$

Дефиниције сопственог вектора и сопствене вриједности која му одговара су дате у литератури [9].

Ако се вратимо на израз (2.7), закључује се да постоје два ограничења када је у питању вектор \mathbf{y} , и та ограничења произлазе из услова одговарајуће лабеле вектора \mathbf{x} . Прво се узима у обзир ограничење да је $\mathbf{y}^T \mathbf{D} \mathbf{1} = 0$. Лако се може показати да је ово ограничење вектора \mathbf{y} аутоматски задовољено рјешавањем уопштеног система сопствених вриједности. То се може постићи трансформисањем изрази (2.7) у стандардни облик сопственог система и показујући да је одговарајући услов у овом случају задовољен. Према томе, израз (2.7) се може записати на сљедећи начин:

$$\mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} \mathbf{z} = \lambda \mathbf{z}. \quad (2.8)$$

гдје је $\mathbf{z} = \mathbf{D}^{\frac{1}{2}} \mathbf{y}$. Лако се може провјерити да је $\mathbf{z}_0 = \mathbf{D}^{\frac{1}{2}} \mathbf{1}$ сопствени вектор система датог изразом (2.8), чија је сопствена вриједност једнака 0. Осим тога, $\mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}}$ је симетрична позитивна семидефинитна матрица, с обзиром да је и матрица $(\mathbf{D} - \mathbf{W})$, која је још позната и под називом Лапласова матрица, такође позитивна и семидефинитна. Дакле, \mathbf{z}_0 је у ствари сопствени вектор са најмањом сопственом вриједности система (2.8) при чему су сви сопствени вектори овог система нормални један на други. У овом случају то значи да је \mathbf{z}_1 , који представља сопствени вектор којем одговара друга најмања сопствена вриједност, нормалан на \mathbf{z}_0 . Уколико ову тврдњу преведемо у уопштени систем сопствених вриједности (2.7), долази се до сљедећих закључака: 1) $\mathbf{y}_0 = \mathbf{1}$ је сопствени вектор са сопственом вриједности 0 и 2) $0 = \mathbf{z}_1^T \mathbf{z}_0 = \mathbf{y}_1^T \mathbf{D} \mathbf{1}$, гдје је \mathbf{y}_1 сопствени вектор са другом најмањом сопственом вриједности система (2.7).

Сада ћемо се присјетити чињеница везаних за Рајлијев однос: нека је матрица A симетрична матрица. Уз ограничење да је вектор x ортогоналан на $j - 1$ најмањих сопствених вектора x_1, \dots, x_{j-1} , однос $\frac{x^T A x}{x^T x}$ је минимизован сљедећим најмањим сопственим вектором x_j и свака минимална вриједност је одговарајућа сопствена вриједност λ_j . Као резултат, добија се:

$$z_1 = \operatorname{argmin}_{z^T z_0 = 0} \frac{z^T D^{-\frac{1}{2}} (D - W) D^{-\frac{1}{2}} z}{z^T z}$$

одакле слиједи:

$$y_1 = \operatorname{argmin}_{y^T D \mathbf{1} = 0} \frac{y^T (D - W) y}{y^T y}$$

Према томе, сопствени вектор са другом најмањом сопственом вриједности уопштеног система (2.7) представља реално рјешење проблема *нормализованог пресеја*. Друго ограничење вектора y , према којем би $y(i)$ требао да узима двије дискретне вриједности, није аутоматски задовољено. Због тога се и каже да *нормализовани пресејек* није рјешење нашег оригиналног проблема, односно проблема *пресеја*. Међутим, ублажавањем овог ограничења, које подразумијева да се вриједности сопственог вектора трансформишу у дискретан облик, може се постићи да израчунавање минималне вриједности *нормализованог пресеја* буде једноставније.

Водећи се овом логиком може се направити сличан аргумент који ће показивати да је сопствени вектор са трећом најмањом сопственом вриједности такође реално рјешење које оптимално дијели прва два дијела. У ствари, овај дио аргумента може бити проширен на такав начин да показује да се у сваком наредном кораку постојећи графови могу подијелити на два дијела, уколико се користи сопствени вектор са сљедећом најмањом сопственом вриједности. Међутим, у пракси, грешка апроксимације, која настаје због разлике између реалне вриједности рјешења и дискретне вриједности рјешења, се повећава са сваким узетим сопственим вектором и како сви сопствени вектори морају да задовоље ограничење опште међусобне ортогоналности, рјешења базирана на вишим сопственим векторима постају непоуздана. Зато је најбоље да се проблем партиционисања рјешава на сваком подграфу појединачно.

2.3. Алгоритам груписања

У претходна два поглавља је дат детаљан математички опис алгоритма *нормализованих пресеја* који се користи за подјелу графа. У наредном излагању ће се опширније описати кораци из којих се састоји алгоритам који се користи при сегментацији слика, а кога називамо алгоритам груписања. Већ је речено да сопствени вектор коме одговара друга најмања сопствена вриједност представља реално рјешење које оптимално врши подјелу цијелог графа, док је трећа најмања сопствена вриједност

рјешење које оптимално дијели први подграф на два дијела, итд. Односно, постојећи граф се може издијелити уколико се при свакој подјели користи сљедећа најмања сопствена вриједност. Међутим, такав начин подјеле не даје увијек поуздане резултате, па се из тог разлога проблем подјеле графа рјешава посебно за сваки подграф. Овакво рјешење се имплементира кроз рекурзивни алгоритам груписања тако што се сваки пут када се добије подграф опет рјешава уопштени систем сопствених вриједности при чему се добија сопствени вектор са другом најмањом сопственом вриједности на основу којег се врши подјела тог подграфа. Реализација алгоритма груписања је извршена у програмском пакету *Матлаб*.

Алгоритам груписања, који се обрађује у овом дипломском раду се може описати кроз наредна четири корака:

1. Узима се произвољна слика, формира се тежински граф $G = (V, E)$ и одређују се тежине грана које повезују два чвора, при чему су те тежине мјера сличности између два чвора
2. Рјешава се систем $(D - W)x = \lambda Dx$ да би се добио сопствени вектор са најмањом сопственом вриједности
3. Сопствени вектор са другом најмањом сопственом вриједности се користи при подјели графа
4. Одлучује се да ли тренутна подјела графа треба да се даље дијели и уколико је то потребно, врши се поновна подјела сегментованих дијелова.

2.3.1. Реализација тежинског графа

Поступак креирања тежинског графа почиње тако што се прво одабере слика чију сегментацију желимо. Формирање графа подразумијева да се пиксели слике посматрају као чворови, док се веза између свака два пиксела представља граном графа. Тежина те гране треба да представља вјероватноћу да два пиксела припадају истом објекту на слици. За мјеру сличности два пиксела се узима вриједност свјетлине или боје посматраних пиксела, као и њихова просторна локација, па се у том случају тежина графа која повезује два чвора i и j може описати математичом релацијом на сљедећи начин:

$$w_{ij} = e^{-\frac{\|F_i - F_j\|^2}{\sigma_f^2}} \cdot e^{-\frac{\|X_i - X_j\|^2}{\sigma_x^2}} \quad (2.9)$$

при чему мора бити задовољен услов $\|X_i - X_j\| < r$. Функција F_i код сивих слика дефинише вриједност свјетлине пиксела, а код колор слика представља вриједност боје, која је одређена са три координате, а функција X_i одређује просторну локацију пиксела. При одређивању разлике просторне локације два пиксела, користи се Еуклидова функција удаљености.

Уколико услов $\|X_i - X_j\| < r$ није задовољен, вриједност тежине између пиксела i и j ће бити једнака нули. r представља параметарску вриједност која се поставља произвољно и

од ње зависи који ће се све пиксели са слике узимати у обзир при сегментацији, а исто тако зависи и брзина извршавања самог поступка. Што је већи параметар r , сегментација ће се дуже извршавати, јер је потребно да се изврши већи број поређења пиксела. Математички гледано, свака слика се посматра као матрица, па се према томе тежински граф такође представља у виду матрице, коју називамо тежинска матрица. Уколико је наша улазна слика димензија $n \times n$, то значи да ће тежинска матрица бити реда $n^2 \times n^2$. За слике величине, нпр., 100×100 тежинска матрица ће бити врло великих димензија, па је потребно да се она реализује као ријетко попуњена матрица, јер се једино у том случају може алоцирати потребна меморија матрице. Већина елемената у ријетко попуњеној матрици ће имати вриједност 0, с обзиром да мора бити задовољен услов $\|X_i - X_j\| < r$.

2.3.2. Рјешавање система сопствених вриједности

Када се реализује тежинска матрица, са свим вриједностима тежина, потребно је добити дијагоналну матрицу D , након чега се прелази на одређивање сопствених вектора и њима одговарајућих сопствених вриједности. Да би се добиле ове вриједности, потребно је ријешити сљедећи систем:

$$(D - W)y = \lambda Dy \quad (2.10)$$

Као што је већ напоменуто у претходној глави, уопштени систем, одређен са изразом (2.10), се може трансформисати у стандардни облик проблема сопствених вриједности:

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}z = \lambda z \quad (2.11)$$

У *Матлабу* постоји готова функција (*eigs*) која се користи при израчунавању сопствених вектора и њихових сопствених вриједности. Недостатак ове функције је у томе што се веома споро извршава, с обзиром да је ријетко попуњена матрица великих димензија. Све то доводи до успоравања укупног поступка сегментације. Да не бисмо одређивали све сопствене векторе, јер је нама потребан само сопствени вектор са другом најмањом сопственом вриједности, постоји могућност да се позове функција *eigs* тако што се назначи да се при израчунавању добијају само двије најмање сопствене вриједности. На овај начин се донекле убрзава поступак извршавања функције *eigs*.

Рјешавање стандардног проблема сопствених вриједности за све сопствене векторе захтјева $O(n^3)$ операција, гдје је n број чворова у графу. На основу овога се може закључити да се са повећањем броја чворова у графу, повећава и број операција, што постаје непрактична околност у апликацијама гдје се врши сегментација слика, када се посматра са аспекта времена извршавања целокупног поступка. Добра околност груписања које се обрађује у овом дипломском раду је у томе што оно има сљедеће особине: 1) графови су углавном само локално повезани, због чега је резултујући сопствени систем проријеђен, односно формира се тежинска матрица која има велики број нултих вриједности, 2) само првих неколико сопствених вектора су потребни при подјели графа и 3) подјела сопственог вектора на два сегмента се врши на основу знака вектора,

односно уколико је вриједност сопственог вектора позитивног предзнака она ће се придружити једном сегменту, а у случају негативног предзнака припаће другом сегменту. Ове посебне особине датог проблема могу у потпуности бити искориштене уколико се примјени *Lanczos*⁴ метод, са којим се број операција смањује на $O(n)$. Детаљнији опис *Lanczos* метода као и његов утицај на вријеме израчунавања сопствених вектора су *Shi* и *Malik* дали у свом раду [16].

2.3.3. Подјела графа кориштењем сопственог вектора са другом најмањом сопственом вриједности

Након рјешавања уопштеног система сопствених вриједности и добијања свих сопствених вектора и њихових сопствених вриједности, приступа се избору сопственог вектора којем одговара друга најмања сопствена вриједност. Овај вектор омогућава подјелу графа на два дијела. Односно, прво се нађе средња вриједност одговарајућег сопственог вектора, а затим се све вриједности које су мање од те средње придружују једном сегменту, а остале другом сегменту. У идеалном случају, сопствени вектор би требао да узима двије дискретне вриједности, при чему би се на основу знака добијених вриједности одређивао начин подјеле графа. Међутим, у пракси имамо другачију ситуацију, односно сопствени вектор узима континуалне вриједности и у том случају је потребно изабрати тачку подјеле на основу које ће се граф подјелити на два дијела. Постоје различити начини бирања тачке подјеле. Један од начина је узимање нуле, средње или *медиан* вриједности, или се евентуално може тражити тачка подјеле тако да $N_{cut}(A, B)$ вриједност буде минимизована. Минимизована N_{cut} вриједност се добија помоћу израза (2.6), при чему је њен начин израчунавања детаљно описан у поглављу 2.2. Након одређивања N_{cut} вриједности, слика се дијели на два сегмента.

2.3.4. Рекурзивно понављање подјеле графа

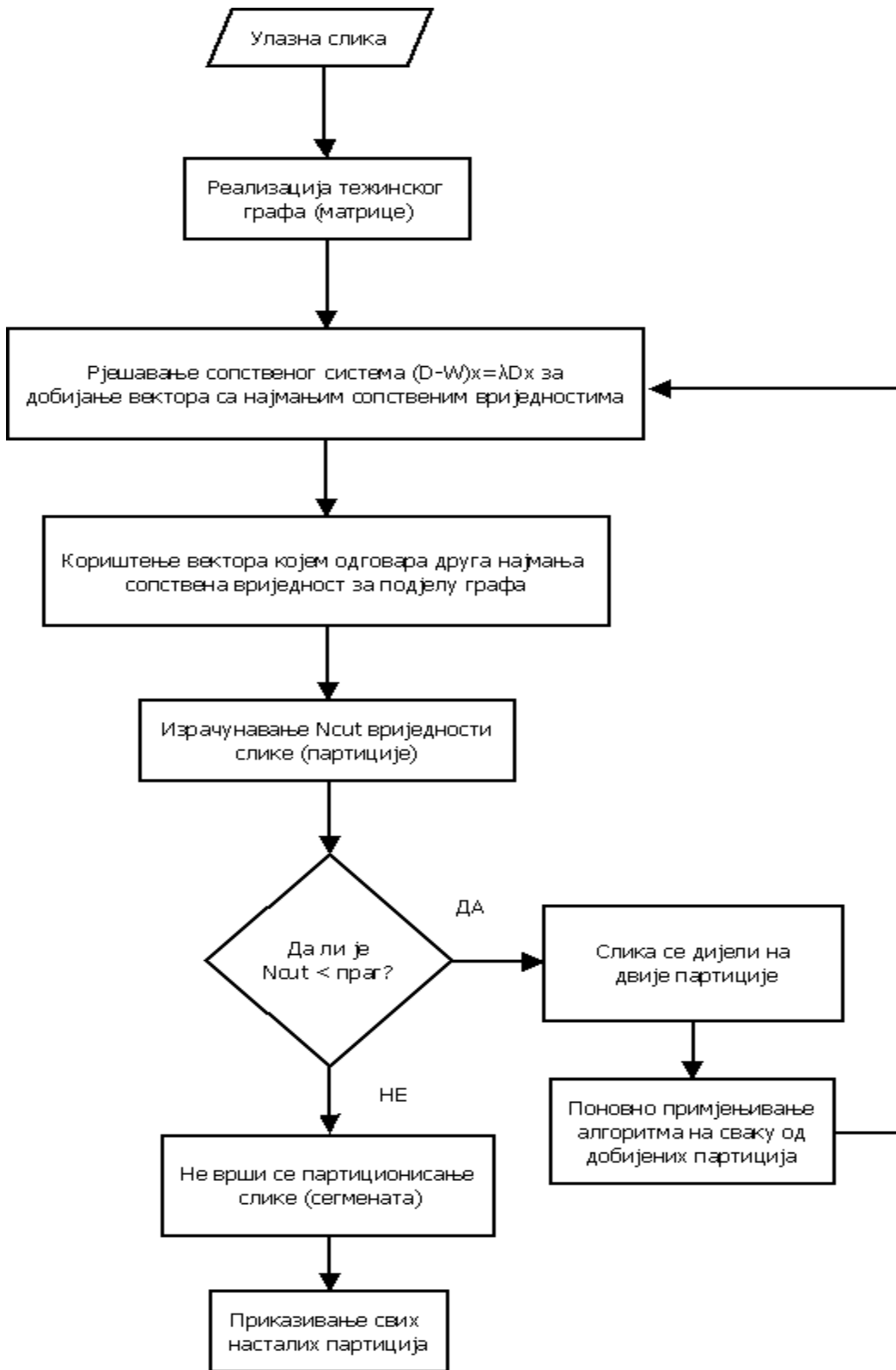
Одабиром тачке подјеле, граф се дијели и као резултат се добијају два подграфа. Затим се на добијене подграфове опет примјењује дати алгоритам и тај поступак се понавља све док N_{cut} вриједност не пређе одређени лимит.

При вршењу подјеле, потребно је поставити критеријум стабилности. О чему је у ствари ријеч? *Shi* и *Malik* су у својим истраживањима дошли до закључка да сопствени вектор понекад може прије узети облик континуалне функције, него дискретне вриједности која се тражи. Са аспекта сегментације, то значи да ће такав сопствени вектор покушати да подјели неки регион слике, код којег подјела није потребна. У ствари, уколико смо приморани да извршимо подјелу слике која има овакав сопствени вектор, показаће се да постоје многе тачке подјеле које имају сличне N_{cut} вриједности. Због тога ће таква подјела бити непоуздана и нестабилна. Да бисмо избјегли овакву сегментацију,

⁴ *Lanczos* алгоритам – итеративни алгоритам који се користи када је потребно наћи сопствене вриједности и векторе квадратне матрице. Нарочито је користан у случају веома великих *sparse* матрица.

уводи се критеријум стабилности који има задатак да спријечи даљу подјелу слике уколико та подјела није потребна, па кажемо да он представља неки вид прага. Према томе, уколико је N_{cut} вриједност слике мања од вриједности тог прага, опет се позива алгоритам који се сада примјењује на сваки подграф (подсегмент) појединачно. Поступак се наставља све док N_{cut} вриједност не постане већа од прага, и тада престаје даље партиционисање сегмената.

За боље разумијевање алгоритма груписања на Слици 2.3 је приказан дијаграм тока који приказује све потребне кораке при извршавању поступка сегментације.



Слика 2.3 - Дијаграм тока алгоритма груписања

Принцип подјеле слике кориштен у овом дипломском раду се још зове и рекурзивни *Two-way N_{cut}* , којим се подразумијева да се слика прво подијели на два сегмента, а затим се на сваки од добијених сегмената опет примјењује алгоритам, при чему долази и до њихове подјеле на два дијела. Недостатак овог приступа је у томе што се користи само други сопствени вектор, иако неколико сљедећих сопствених вектора такође садрже корисне информације за подјелу. Умјесто овог, може се користити приступ који се заснива на критеријуму *k – way нормализованог пресјека*, који дијели слику на k сегмената. И у овом случају за дату слику, фомирамо одговарајући граф $G = (V, E, W)$. Претопставићемо да је укупан број чворова једнак N . Сваки пар чворова је повезан са граном графа E . Тежина $W(i, j)$ представља сродност између чворова i и j , и она мјери вјероватноћу да чворови i и j припадају истом подграфу. За подјелу графа на k подграфа, при чему вриједи $V = V_1 \cup V_2 \cup \dots \cup V_k$, $\forall V_i \cap V_j = \emptyset$, $i \neq j$, *k – way нормализовани пресјек* је дефинисан на сљедећи начин:

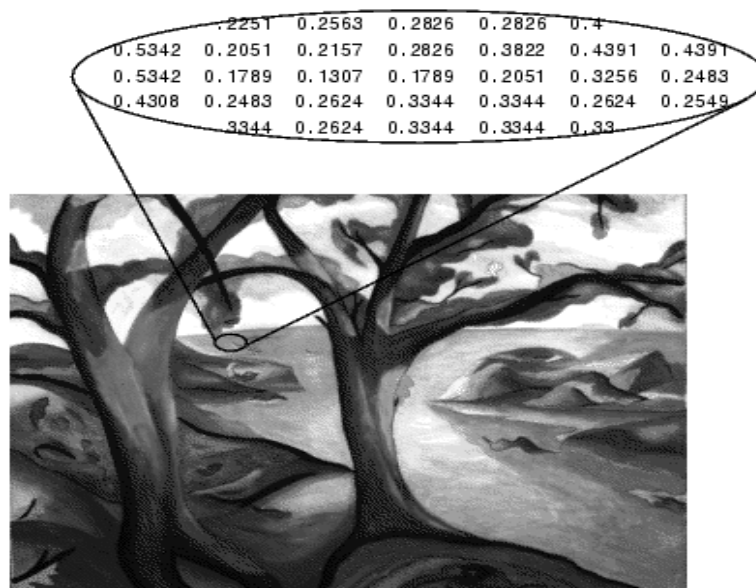
$$\min k N_{cuts}(V) = \frac{1}{k} \sum_{i=1}^k \frac{cut(V_i, V \setminus V_i)}{assoc(V_i, V)}$$

гдје $cut(V_i, V \setminus V_i) = \sum_{i \in V_i, j \in V \setminus V_i} W(i, j)$ представља суму свих тежина грана које повезују чворове из подграфа V_i са чворовима који не припадају том подграфу, а $assoc(V_i, V) = \sum_{i \in V_i, j \in V} W(i, j)$ је сума свих тежина грана које повезују чворове из подграфа V_i са свим осталим чворовима из графа V . Иако директна оптимизација *k – way нормализованог пресјека* представља *NP*-комплетно рјешење, трансформацијом лабеле матрице за подјелу у континулани домен омогућава се да се овај проблем ријешити помоћу система сопствених вриједности. Повећавањем броја k , долази до монотоног раста *k – way нормализованог пресјека*. Овим се показује да *k – way нормализовани пресјек* не може адаптивно одредити број сегмената k за дату слику. Због тога, како одабрати број k остаје отворен проблем [11]. Овај дипломски рад се базира само на кориштењу рекурзивног *Two-way N_{cut}* поступка.

3. СЕГМЕНТАЦИЈА СИВИХ СЛИКА

У претходној глави је детаљно описан математички модел алгоритма *нормализованих пресјека*, његове основне особине, као и начин како се он примјењује при сегментацији слика. У овој глави ће се вршити примјена датог алгоритма на сиве слике, те ће се анализирати како поједини параметри утичу на резултате сегментације. С обзиром на проблем спорог извршавања алгоритма, који је споменут у претходној глави, слике на којима се тестира алгоритам су мањих димензија, тачније ниједна слика не прелази величину 128×128 пиксела. Сегментација на сликама већих димензија је такође могућа, међутим оне нису кориштене јер је акценат првенствено стављен на резултате који се добијају при самом поступку, па тек онда на брзину добијања тих резултата. Тестирања су вршена на рачунару слједећих карактеристика: *Microsoft Windows XP*, брзина процесора 1.70 GHz и 1 GB радне меморије. При томе је кориштен програмски пакет *Матлаб*, верзија 2007.

Прије него што се почне са тестирањем слика, прво ћемо дати дефиницију сиве слике. Сива слика је слика у којој вриједност сваког пиксела представља појединачан узорак, који носи само информацију интензитета. Ова врста слика је сачињена од нијанси сиве боје, при чему оне варирају од оне која има најслабији интензитет (црна) до оне са најјачим интензитетом (бијела). Црно-бијеле слике, за разлику од сивих, имају само двије боје, црну и бијелу, па се у литератури често сусрећу под називом бинарне слике. Интензитет пиксела сивих слика је изражен у опсегу од 0 до 255, гдје је 0 црно, а 255 бијело, или од 0 до 1, гдје 0 представља црну, а 1 бијелу боју, док остале бројчане вриједности представљају различите нивое сивила [12]. На Слици 3.1. је приказана типична сива слика са њеном нумеричком репрезентацијом.



Слика 3.1 – Нумеричко представљање сивих слика у опсегу од 0 до 1

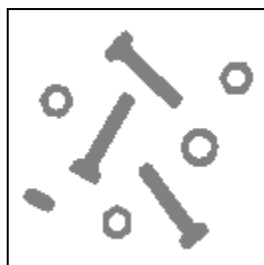
3.1. Примјена алгоритма *нормализованих пресека* на сиве слике

Као што је већ напоменуто, алгоритам је примјењен на сивим сликама, гдје се као мјерило сегментације узима вриједност свјетлине на слици. У случају сивих слика, формира се тежинска матрица, чији су елементи дефинисани сљедећим изразом:

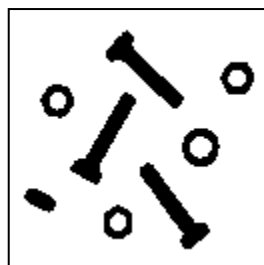
$$w_{ij} = e^{-\frac{\|F_i - F_j\|^2}{\sigma_I^2}} \cdot e^{-\frac{\|X_i - X_j\|^2}{\sigma_X^2}} \quad (3.1)$$

при чему мора бити задовољен услов $\|X_i - X_j\| < r$. X_i представља просторну локацију пиксела i , док је F_i вектор који описује особине пиксела које се базирају на информацијама везаним за интензитет, боју или текстуру. Код сивих слика, та информација се искључиво односи на интензитет пиксела на слици, што се математички може описати као $F_i = I_i$. То значи да ће тежина између два пиксела бити једнака разлици њихових освјетљености, односно интензитета. Када је у питању услов $\|X_i - X_j\| < r$, може се примјетити да у случају када он није испуњен тежина w_{ij} ће бити једнака 0. Односно, израчунаваће се само оне тежине код којих су пиксели i и j удаљени мање од r пиксела.

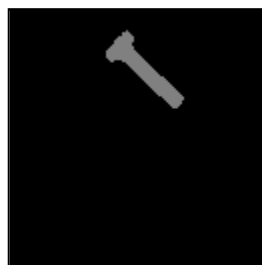
Прва тестирања су вршена на синтетичким сликама које имају само двије нијансе сиве боје, при чему једна нијанса одређује позадину, а друга нијанса објекат. Слика „sarafi.tif“ величине 128×128 пиксела је типичан примјер такве слике. Резултати сегментације су приказани на Слици 3.2. На основу добијених резултата, може се примјетити да је сегментација извршена без грешке, односно да је сваки објекат на слици издвојен као посебан регион, при чему позадина слике такође представља један регион. Вријеме које је потребно да се изврши сегментација износи око 105 секунди.



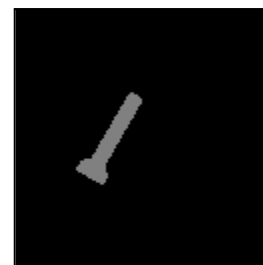
а)



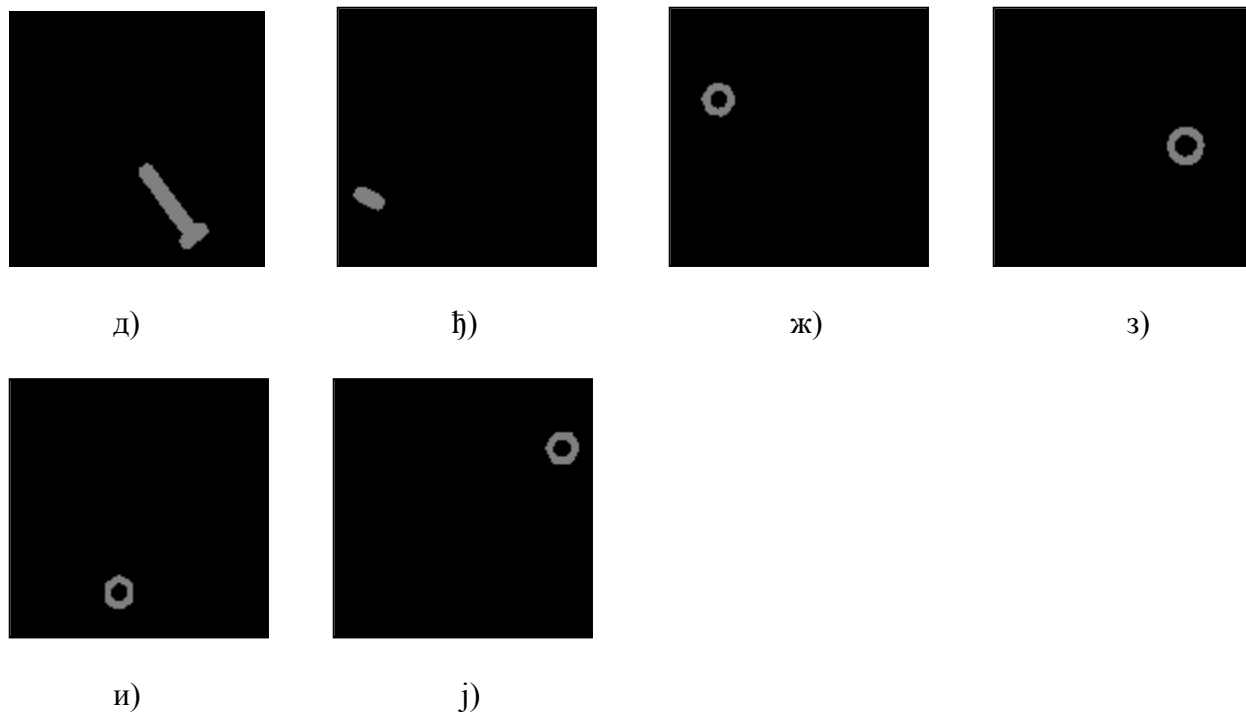
б)



в)



г)



Слика 3.2 – а) оргинална слика (128×128). Сlike б) - ј) показују добијене регионе, гдје је N_{cut} вриједност мања од 0.006. Вриједности параметара: $\sigma_I = 0.08$, $\sigma_X = 4$, $r = 5$

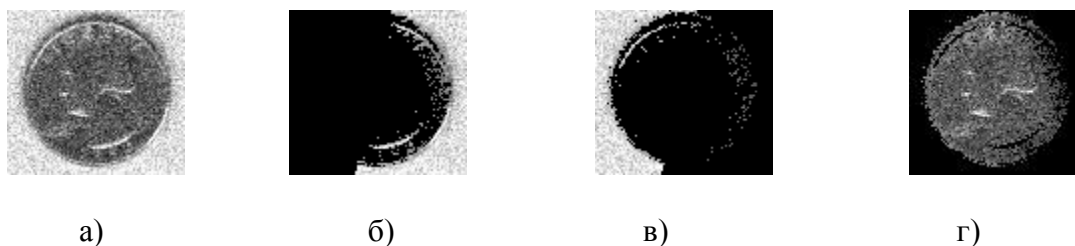
Слика „*dinar.png*“ је примјер сиве слике код које имамо присутне различите нијансе сиве боје. Међутим, и у овом случају је сегментација добро извршена, односно добијају се два региона, при чему једном одговара објекат (динар), а другом позадина. Након извршене сегментације, мали објекти динара су придружени позадини, међутим, уз помоћ одређених пост-обрадних поступака, овај недостатак се може кориговати тако да резултати сегментације буду без грешке. Величина слике је 82×89 пиксела, па је вријеме извршавања знатно мање (нешто мање од 20 секунди). Резултати сегментације су приказани на Слици 3.3.



Слика 3.3 - (а) оргинална слика (82×89). Сlike б) - в) показују добијене регионе, гдје је N_{cut} вриједност мања од 0.008. Вриједности параметара: $\sigma_I = 0.08$, $\sigma_X = 15$, $r = 7$

Уколико бисмо на претходне слике додали шум, сегментација би се у том случају поприлично нарушила. Примјењивањем Гаусовог шума на слику „*sarafı.tif*“ било је

немогуће добити било какву разумну сегментацију која би издвајала објекте од интереса, па из тог разлога ти резултати овдје нису ни приказани. Средња вриједност примјењеног Гаусовог шума је била једнака 0, а вриједност варијансе 0.01. Ситуација са сликом „*dinar.png*“ је нешто другачија, међутим и у том случају се не добијају најбољи резултати, Слика 3.4. Може се примјетити да су приликом сегментације добијена три региона, при чему се позадина састоји из два дијела, а пиксели динара нису придружени само региону објекта, него неки од њих припадају и позадини. Разлог због којег се добијају овакви резултати је тај да се усљед шума мијењају вриједности пиксела на слици, што утиче на вриједност тежине између два пиксела, па према томе долази и до промјене вриједности сопствених вектора, као и њихових сопствених вриједности. То за последицу има промјену N_{cut} вриједности, па се због тога формирају другачији региони, у односу на оне који се добијају при сегментацији слике без шума. Ипак, без обзира што долази до претјеране подјеле позадине, добијену сегментацију можемо сматрати прихватљивом.

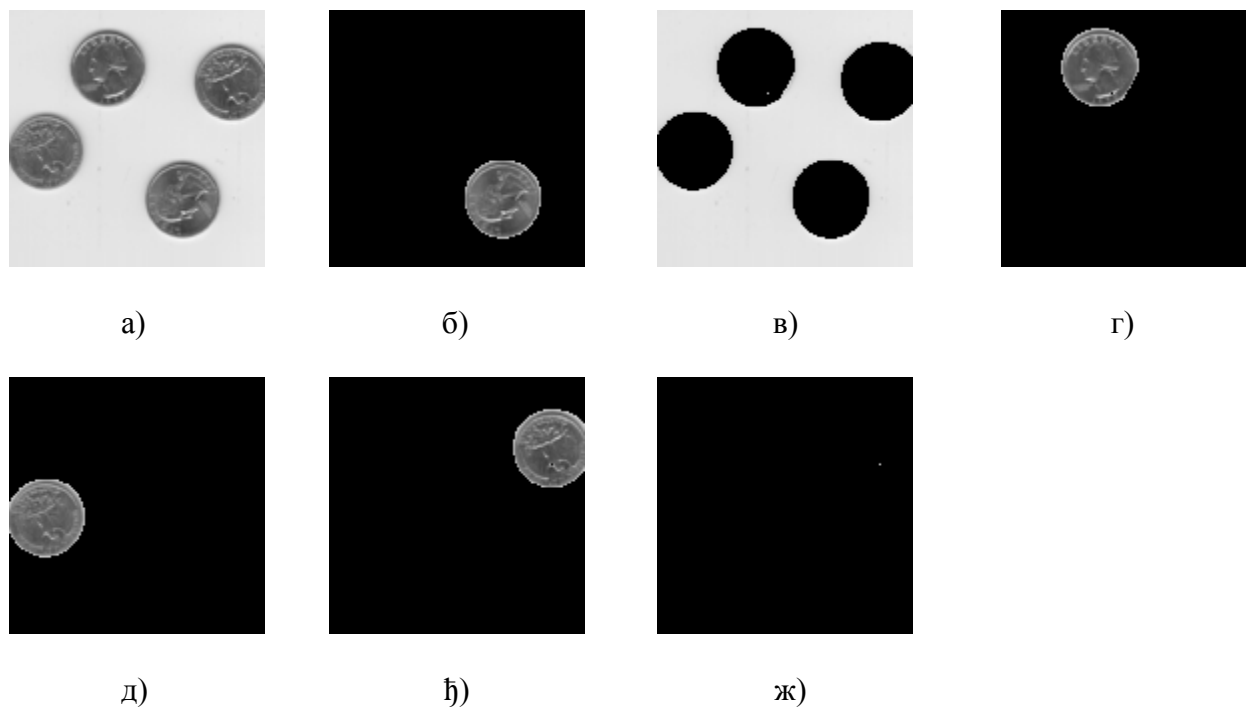


Слика 3.4 – (а) оргинална слика са Гаусовим шумом (82 × 89). Сlike б) - г) показују добијене регионе, гдје је N_{cut} вриједност мања од 0.075. Вриједности параметара: $\sigma_I = 0.075$, $\sigma_x = 15$, $r = 7$

У општем случају, када се на слике примјени Гаусов шум може се закључити сљедеће: да би се добили задовољавајући резултати потребно је да Гаусов шум претјерано не утиче на карактеристике саме слике. Колико ће се слика промјенити након што јој се дода Гаусов шум, зависи од тога какви објекти се налазе на њој, колико их има и које су нијансе. Иако се на први поглед чини да ће код синтетичких слика резултати бити добри, с обзиром на њихову једноставност, то није случај. Код слике „*sarafi.tif*“ немогуће је добити добру сегментацију, јер присуство Гаусовог шума у потпуности мијења карактеристике слике, па умјесто двије нијансе сиве, имамо мноштво различитих вриједности пиксела, што директно утиче на добијање сопствених вектора, па самим тим и на коначне резултате.

Интересантну ситуацију имамо у случају слике „*dinari128.png*“ Слика 3.5. На основу задатих параметара, закључујемо да су се у поступку сегментације издвојили сви објекти од интереса, као и позадина с тим да имамо један регион вишка, односно регион на којем се издвојио само један бијели пиксел. Параметар који се односи на праг, од којег зависи да ли ће се сегментација даље извршавати, може да се креће у опсегу од 0.002 до 0.02, а да се при томе не мијењају резултати сегментације. Уколико се вриједност тог параметра мијења ван овог опсега, односно ако се повећава или смањује, број региона ће бити већи, односно мањи. Са промјеном осталих параметара, резултати сегментације ће се промијенити у толикој мјери да неки региони садрже објекат, али и дио позадине, или да

се позадина састоји од више региона, при чему су објекти коректно сегментирани. Вријеме извршавања цјелокупне сегментације износи 125 секунди, а величина слике је 128×128 .



Слика 3.5 – а) оригинална слика (128×128). Сlike б) – ж) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.02. Вриједности параметара: $\sigma_I = 0.04$, $\sigma_X = 15$, $r = 7$

3.2. Утицај параметара алгоритма *нормализованих пресека* на резултате сегментације сивих слика

Слика „*helikopter.jpg*“ је такође једна од тестних слика, на којој ћемо детаљније описати утицај параметара на поступак сегментације. Једна од могућих сегментација, односно резултати који се добијају кориштењем назначених параметара, је приказана на Сlici 3.6.



Слика 3.6 – а) оригинална слика (84×147). Сlike б) – г) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.03. Вриједности параметара: $\sigma_I = 0.08$, $\sigma_X = 10$, $r = 5$

У идеалном случају би се требале добити само два региона, један који би означавао објекат (хеликоптер), и други који би представљао позадину. Са овако дефинисаним параметрима смо добили три региона, при чему је први регион обухватио дио позадине и дио елиса хеликоптера, други регион већим дијелом чини хеликоптер и дио позадине, док трећи регион чини остатак позадине. Може се закључити да поједини детаљи објекта, као што је дио елиса хеликоптера, као и неки дијелови свјетлијих нијанси нису придружени истом региону на којој се налази већи дио објекта. Уколико би се промјенила вриједност првог параметра σ_1 , чија вриједност утиче на први члан тежинске функције, добиће се резултати приказани на Слици 3.7.



Слика 3.7 – Сlike а) – в) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.03. Вриједности параметара: $\sigma_1 = 0.04$, $\sigma_x = 10$, $r = 5$

Смањивањем вриједности овог параметра, долази до знатно бољег издвајања објекта, односно у овом случају једним регионом је обухваћен скоро цијели хеликоптер осим његових елиса. Друга два региона чини позадина и елисе хеликоптера. У овом случају резултати су много бољи, па се може закључити да параметар σ_1 има велики утицај у одређивању сопствених вектора па самим тим и у одређивању N_{cut} вриједности. Иако је промјена овог параметра релативно мала, тј. гледајући први и други случај сегментације његова вриједност се разликује само за 0.04, међутим та промјена доводи до тога да се ред првог члана тежинске функције смањи и до 4 пута. Нпр, уколико је разлика у интензитету између два пиксела једнака 0.5 и ако је $\sigma_1 = 0.08$ први члан ће бити реда 10^{-17} , а за $\sigma_1 = 0.04$ ред ће се смањити на 10^{-68} , што је у ствари показатељ колики је утицај овог параметра на цјелокупну сегментацију. Према томе, може се доћи до закључка да се са смањивањем овог параметра боље издвајају поједини детаљи објекта, гдје ће се нпр. за $\sigma_1 = 0.02$ издвојити цијели хеликоптер, али без елиса. Међутим, у том случају ће се јавити већи број партиција које чине позадину, као и региони који ће садржати само по један пиксел. Ово је проблем који се јављао и код слике „*dinari128.png*“.

Уколико се мијења параметар r , нпр. ако се он повећа, при прорачуну тежинске функције у обзир се узима већа област пиксела, односно један пиксел ће се поредити са већим бројем пиксела у његовој близини. Нпр., уколико се узме да је $r = 10$, добиће се резултати приказани на Слици 3.8.



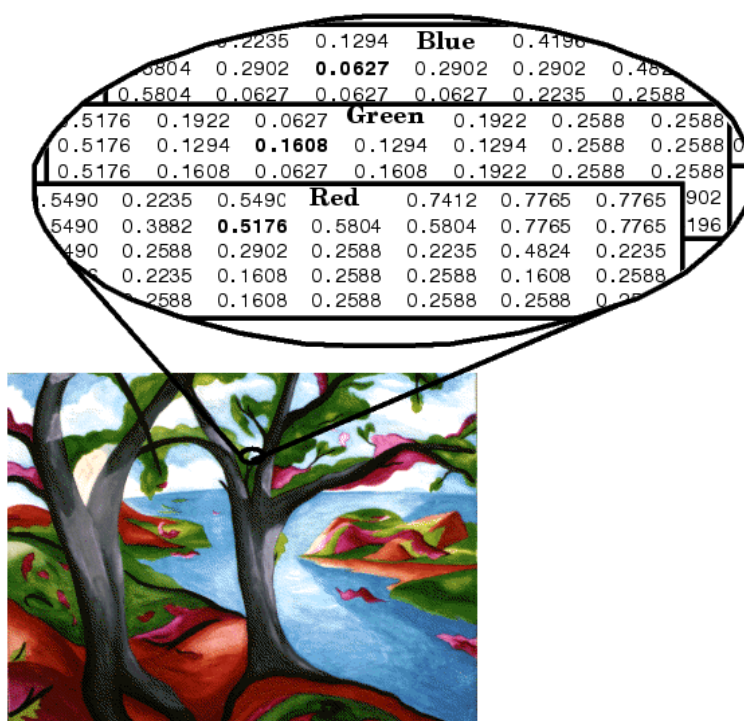
Слика 3.8 – Сlike а) – в) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.008. Вриједности параметара: $\sigma_1 = 0.02$, $\sigma_x = 10$, $r = 10$

У овом случају је регион, којег чини објекат, обухватио и дио елиса. Позадина је издјељена на два региона, док четврти регион чини само једна бијела тачка, односно један бијели пиксел. Очигледно је да су при прорачуну сопствених вектора њихове вриједности биле такве да је слика прво издијељена на регион са бијелим пикселима и сегмент који чини остатак слике који се у даљим позивањима алгоритма опет дијели на одговарајуће подсегменте. При овој сегментацији узели смо и другачију N_{cut} вриједност у односу на претходни примјер, како ће се избјегао већи број непотребних подјела позадине. Нажалост, немогуће је подесити такву вриједност N_{cut} параметра који би спријечио појаву региона који садржи само један пиксел.

4. СЕГМЕНТАЦИЈА СЛИКА У БОЈИ

Претходна глава се бавила анализом резултата који се добијају примјеном алгоритма *нормализованих пресека* на сиве слике и она је представљала увертиру у главни задатак овог дипломског рада, а то је сегментација слика у боји. Прије него што се почне са анализом слика у боји, потребно је да се дефинише шта је то слика у боји, у којим све колор просторима се она може представити, те навести основне карактеристике оних колор модела који су конкретно кориштени у овом дипломском раду. Сlike које су кориштене при тестирању су различитих карактеристика, с обзиром да је циљ дипломског рада да покаже, не само ефикасност алгоритма при сегментацији, него и да се он може примјењивати на што више различитих класа слика. Тестне слике нису прелазиле величину од 160 × 130 пиксела.

Слика у боји је слика која садржи информације о боји за сваки пиксел. За разлику од бинарних и сивих слика које имају само једну вриједност за сваку свјетлину у свакој тачки, слике у боји имају три вриједности свјетлине, по једну за црвену, зелену и плаву како је приказано на Слици 4.1. Ове три вриједности представљају интензитет свјетлости у различитим спектралним опсезима које око види као различите боје [1].



Слика 4.1 – Нумеричка репрезентација слике у боји

За визуелно прихватљиве резултате, неопходно је да се обезбједе три узорка (колор канала) за сваки пиксел, при чему се сваки пиксел тумачи као координата у неком колор простору. Најчешће кориштен колор простор јесте RGB⁵ простор, с обзиром да се слике из овог простора уобичајно користе код фотографије, камера и монитора. Међутим, постоје и други колор простори, као што су нпр. СМЈК⁶, HSI, YCbCr⁷ и L*a*b. Овај рад се конкретно базира на анализи слика у HSI и L*a*b колор простору, па ће према томе они бити детаљније објашњени.

Мотивација због које се у обради слике користи боја потиче од чињенице да је боја моћан дескриптор који поједностављује идентификацију објеката и њихово издвајање, као и да човјек може разликовати много више нијанси боја него нијанси сивога. Оно што представља основне карактеристике које се користе да би се једна боја разликовала од друге су свјетлина, доминантна боја и засићеност. Свјетлина подразумијева интензитет боје, доминантна боја се односи на одговарајућу таласну дужину свјетлости, док је засићеност обрнуто пропорционална количини бијеле свјетлости која је додата доминантној боји [1].

4.1. HSI колор простор

HSI колор простор је веома важан и атрактиван колор модел за обраду слика јер представља боје на сличан начин на који људско око доживљава боју. HSI колор модел представља сваку боју са три компоненте: *hue* - доминантна боја, *saturation* - засићеност, *intensity* – интензитет [1]. На Слици 4.2 је приказан примјер како су представљене боје у HSI колор простору.

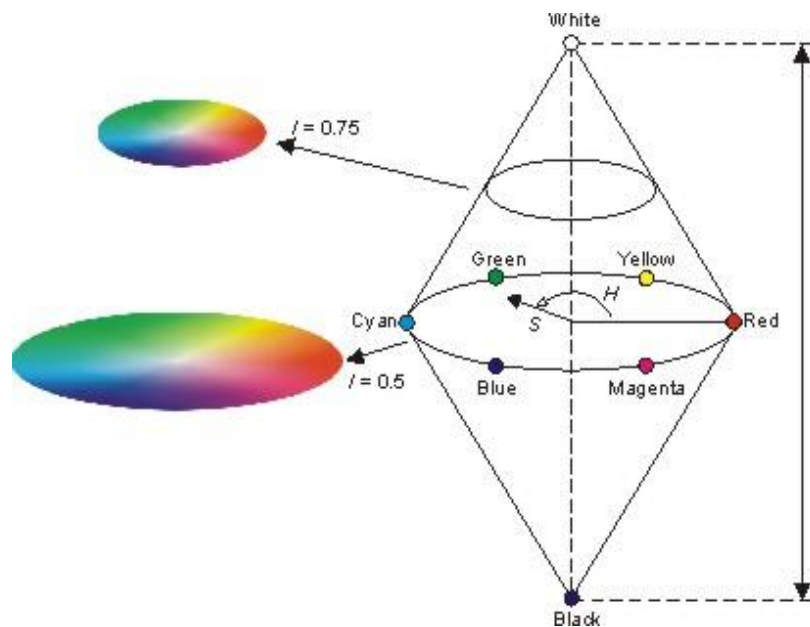
Доминантна боја је компонента која описује саму боју. Она се представља у форми угла између 0 и 360 степени. 0 степени одговара црвеној боји, зелена боја се налази на 120, а плава боја на 240 степени. Компонента засићености одговара полупречнику, и њена вриједност варира од 0 до 1. Интензитет је компонента која дефинише колика је свјетлина

⁵ RGB простор је колор простор базиран на RGB колор моделу код којег се боја формира сабирањем три основне боје различитог интензитета: црвене (red), зелене (green) и плаве (blue). За R=G=B добију се све сиве нијансе од црне до бијеле [1].

⁶ СМЈК простор је колор простор базиран на СМЈК колор моделу. На подручјима гдје се боје RGB модела преклапају настају цијан (Cyan), магента (Magenta), жута (Yellow) и бијела (White). За разлику од адитивног RGB модела, по СМЈК моделу боје се добијају одузимањем. Ако се искомбинују чиста цијан, магнета и жута, апсорпција ће бити максимална што ће произвести црну боју. Како приликом штампе није могуће постићи чисту боју, умјесто црне се добија смеђа, па се са ове три основне боје комбинује и црна (K) [1].

⁷ YCbCr простор је колор простор базиран на YCbCr колор моделу. Овај модел се састоји од луминентне (Y) и двије хроминентне компоненте: Cb, која прекрива спектар од плаве до жуте, и Cr, за спектар од зелене до црвене. Луминентна компонента даје сиву верзију слике, док двије хроминентне компоненте обезбјеђују конвертовање сиве слике у слику у боји [1].

пиксела и његове вриједности се налазе у опсегу између 0 и 1, при чему 0 одговара црној, а 1 бијелој боји.



Слика 4.2 – Представљање боја у HSI колор простору.

При кориштењу HSI колор простора, не мора да се зна колики проценат плаве или зелене боје је потребан да би се добила одређена боја. Да бисмо добили боју коју желимо, само се треба подесити компонента доминантне боје. Уколико желимо да од бордо добијемо пинк боју, подешава се компонента засићености. А да бисмо неку боју направили свјетлијом или тамнијом, мијењамо интензитет. Када је компонента засићености једнака нули, боја је сива, а уколико је засићеност једнака јединици, боја се налази на крајњим границама базе конуса. Према томе, што је веће засићење то ће боја бити више удаљена од централне осе. Када је интензитет једнак нули, боја је црна, и у том случају је компонента засићености недефинисана. Уколико одржавамо вриједност засићености на јединици, при чему мијењамо вриједност интензитета креирају се различите нијансе боје [13]. Формула која врши конверзију слике из RGB у HSI модел и обрнуто је много компликованија него код других колор модела. Међутим, захваљујући *Матлабу*, у којем постоји готова функција која врши дату конверзију, тај проблем је рјешен. Та функција гласи *rgb2hsv* или у обрнутом случају *hsv2rgb* [14]. HSI колор модел се врло често користи у рачунарском виду и анализи слике за детекцију особина или за сегментацију слика.

4.1.1. Примјена алгоритма *нормализованих пресјека* на слике у боји представљене у HSI колор простору

С обзиром да се овдје алгоритам примјењује на слике у боји, тежинска функција, која се користи при формирању тежинске матрице, се разликује у односу на ону коју смо користили код сивих слика. Њена форма је иста, односно описана је помоћу исте формуле:

$$w_{ij} = e^{-\frac{\|F_i - F_j\|^2}{\sigma_f^2}} \cdot e^{-\frac{\|X_i - X_j\|^2}{\sigma_x^2}} \quad (4.1)$$

Међутим, у овом случају је функција F_i дефинисана на другачији начин. Наиме, F_i се овдје базира на информацијама везаним за боју и описује се сљедећом формулом:

$$F_i = [v, v \cdot s \cdot \sin(h), v \cdot s \cdot \cos(h)]_i \quad (4.2)$$

гдје су h , s и v вриједности HSI колор модела, које се користе при сегментацији слике. Може се примјетити да израз (4.2) није одређен само са h , s и v компонентом, него се појављују и функције \sin и \cos , и то у таквој форми да је потребно одредити $\sin(h)$ и $\cos(h)$, након чега се врши множење тих фактора са одговарајућим компонентама кориштеног колор простора. Битно је напоменути да се при одређивању функције F_i треба обратити пажња приликом тражења разлике ($F_i - F_j$) између два пиксела, јер се код слика у боји одузимање мора вршити по све три компоненте, како би се узеле у обзир све три вриједности пиксела. Одређивање функције која се односи на просторну локацију пиксела остаје исто као и код сивих слика, и у том случају вриједи исти услови.

Да бисмо показали ваљаност алгоритама, примјенићемо га на слици на којој су све присутне боје јасно дефинисане без примјеса било каквих нијанси или сметњи. На Слици 4.3 су приказани резултати извршене сегментације, са којих се може видјети да се свака обојена површина слике издвојила као засебан регион. Вријеме трајања сегментације износи 95 секунди.



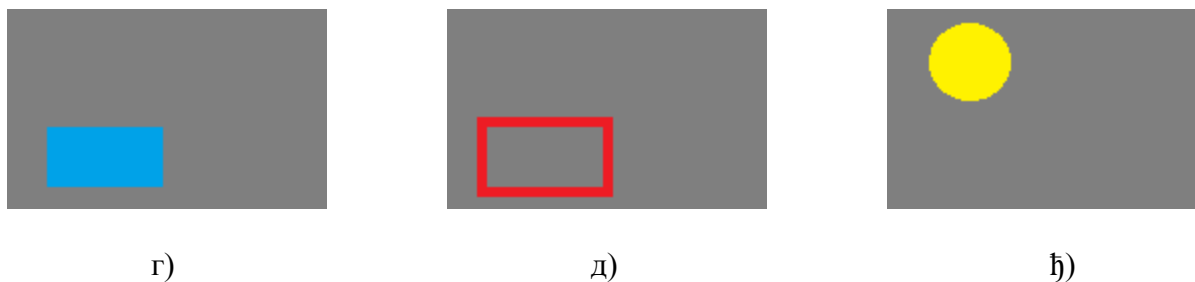
а)



б)

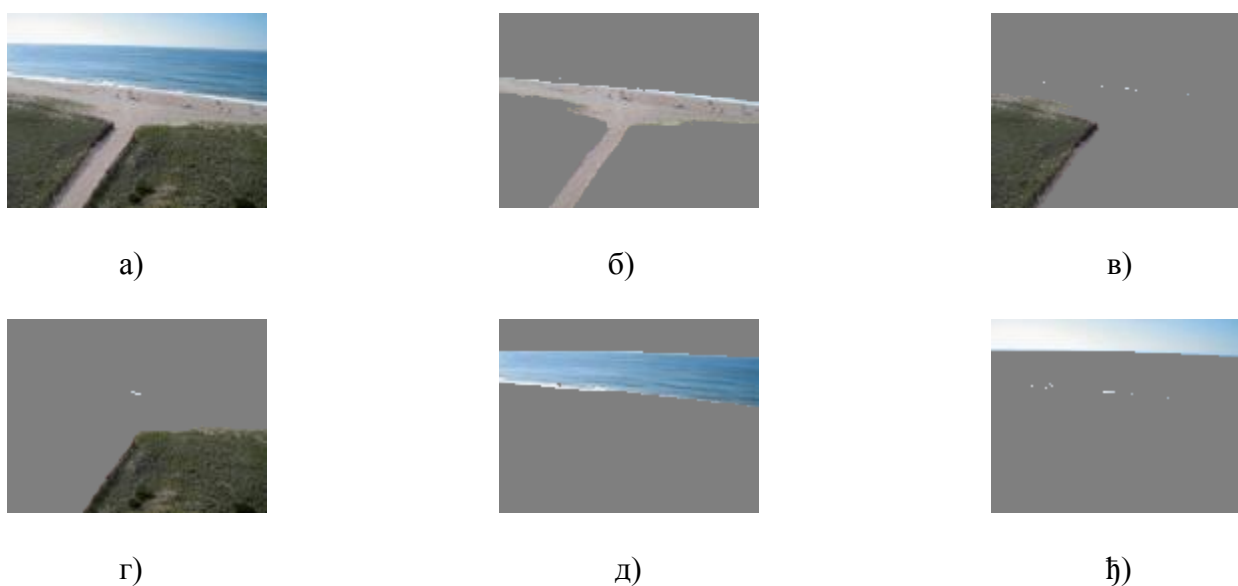


в)



Слика 4.3 – а) оригинална слика (160×100). Сlike б) – њ) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.02. Вриједности параметара: $\sigma_I = 0.08$, $\sigma_X = 4$, $r = 5$

Сlike из природе (различите врсте пејзажа) су у општем случају врло погодне када се врши сегментација на основу боје. Према томе, одређени број тестирања је вршен управо на таквим сликама. На Слици 4.4 је приказана сегментација једне такве слике „plaza.jpg“.

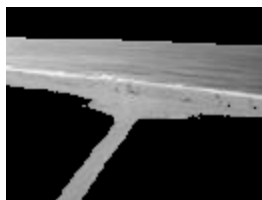


Слика 4.4 – а) оригинална слика (130×97). Сlike б) – њ) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.02. Вриједности параметара: $\sigma_I = 0.04$, $\sigma_X = 20$, $r = 7$

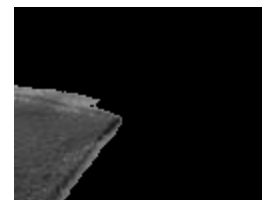
На основу добијених резултата, може се закључити да је сегментација коректно извршена, с тим да неки региони садрже издвојене пикселе који не припадају објекту од интереса који се налази конкретно на том региону. Оно што је карактеристично на овим резултатима је чињеница да је зелена површина подјељена на два региона, иако они имају исте особине, односно исте су боје. Разлог је тај што је зелена површина на слици раздвојена, што алгоритам није у стању да препозна, те је посматра као два одвојена објекта. Уколико се на ову слику примјени функција *rgb2gray*, помоћу које се слика у боји конвертује у сиву слику, а да се при томе вриједности параметара не мјењају, добијају се резултати приказани на Слици 4.5.



а)



б)



в)



г)



д)

Слика 4.5 - а) конвертована сива слика (130×97). Сlike б) – д) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.02. Вриједности параметара: $\sigma_1 = 0.04$, $\sigma_x = 20$, $r = 7$

У овом случају као резултат сегментације се добијају четири региона, односно један мање него у случају када имамо слику у боји. Овдје су два региона из претходног примјера приказана као један. То је посљедица тога да море и плажа имају скоро исте нијансе сиве након извршене конверзије. Овај примјер је узет само као илустративни како би се показало колико боја представља битан фактор при сегментацији слика. Вријеме извршавања алгорита износи око 60 секунди.

На Сликама 4.5 и 4.6 су приказани још неки резултати сегментације слика сличих карактеристика.



а)



б)



г)



д)

Слика 4.5 – а) оргинална слика (154×123). Сlike б) – г) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.015. Вриједности параметара: $\sigma_1 = 0.05$, $\sigma_x = 4$, $r = 7$



а)



б)



в)



г)



д)

Слика 4.6 – а) оргинална слика (130×97). Сlike б) – д) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.02. Вриједности параметара: $\sigma_1 = 0.04$, $\sigma_x = 20$, $r = 7$

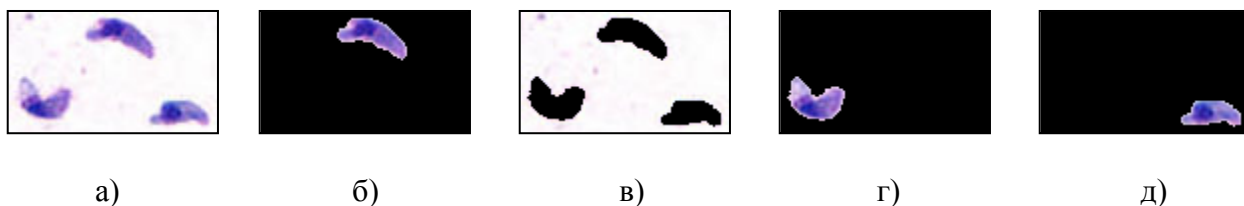
На основу добијених резултата се примјећује да је сегментација успјешно извршена, при чему свака подјела приказује по један објекат од интереса, у овом случају по једну обојену површину. Код слике „pejzaz.jpg“, једном региону припадају и небо и облаци иако су они различитих особина, односно различите боје. Да бисмо добили два региона, потребно је повећати параметар N_{cut} . Међутим, повећавањем овог параметра, долази до подјеле не само неба и облака него и до подјеле зелене површине на два региона, иако знамо да та подјела није разумна. Разлог је чињеница да је зелена боја с десне стране тамније нијансе у односу на лијеву страну слике, па алгоритам то препознаје као два различита објекта. Очигледно да разлика у нијансама зелене боје, доводи до такве промјене при израчунавању сопствених вектора и њихових сопствених вриједности, да прво настаје тај регион, па тек онда при сљедећем позивању алгоритма настају региони који садрже небо,

односно облаке, тако да је немогуће подесити параметре на такав начин да добијемо подјелу слике која ће у потпуности дати добру сегментацију.

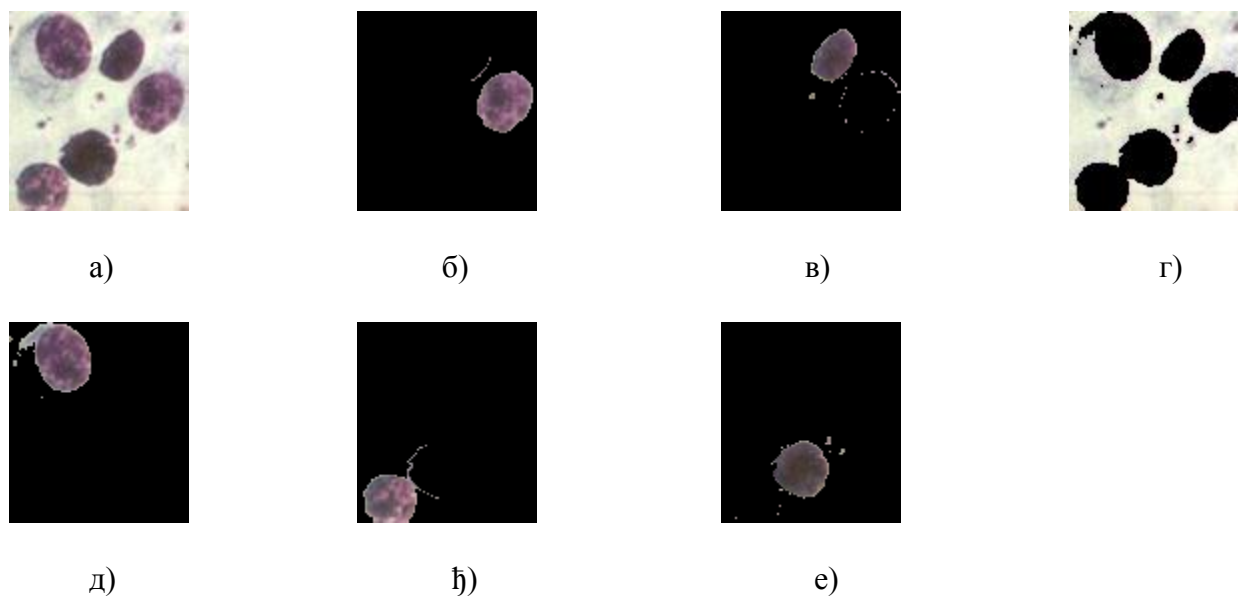
4.1.2. Примјена алгоритма *нормализованих пресека* на микробиолошке и хистолошке слике

Микробиологија је наука која се бави пручавањем микроорганизама. Микроорганизми су од великог значаја, с обзиром да су неопходни за прављење неких основних животних намирница, као што су хљеб, сир, вино итд., а исто тако се користе при прављењу антибиотика, вакцина, витамина и многих других битних продуката. Врло су важни за еколошки систем, јер захваљујући њима постоје азотни и угљеникови циклуси који се одвијају дубоко у земљишту и у дубоким водама. Међутим, постоји и негативна страна микроорганизама, а она се огледа у томе да су кроз историју микроорганизми били изазивачи многих болести, при чему се са неким од њих и дан данас сусрећемо [15]. Зато је потребно вршити разна микробиолошка истраживања, да би се стекао што бољи увид о њиховом присуству. У тим микробиолошким истраживањима примјену је нашла и дигитална обрада слике, или прецизније речено сегментација слике.

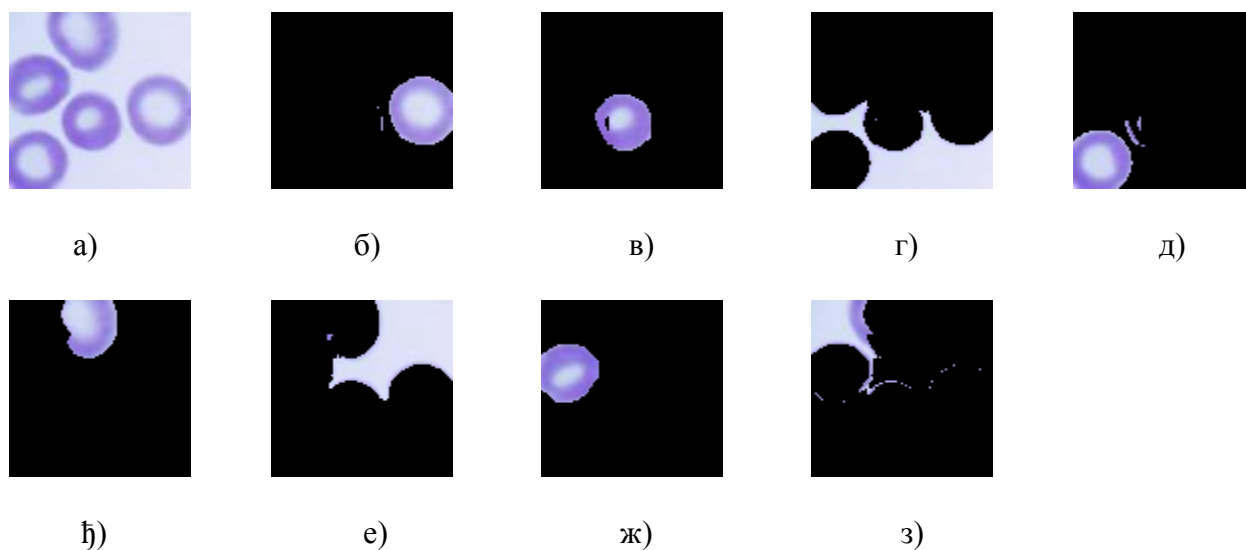
На Сликама 4.7, 4.8 и 4.9 су приказани само неки од резултата добијених приликом тестирања алгоритма *нормализованог пресека* на овој класи слика. Може се примјетити да је сегментација у случају слике „*sl_1.jpg*“ коректно извршена, односно сваком региону припада по један објекат, укључујући и позадину која представља један регион. Што се тиче слике „*sl_2.jpg*“, такође можемо рећи да су резултати задовољавајући јер је сваки објекат представља засебан регион, с тим да сваки од региона садржи поједине пикселе и других објеката. У случају слике „*sl_3.jpg*“ сегментација је успјешно извршена уколико се само посматра издвајање објеката. Међутим, проблем представља позадина која је издјелена на више региона. Са промјеном параметра може се добити мањи број региона, али таквих да су једном региону придружена два објекта, или евентуално објекат и дио позадине, што не представља боље рјешење.



Слика 4.7 – а) оргинална слика „*sl_1.jpg*“ (104 × 60). Сlike б) – д) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.05. Вриједности параметара: $\sigma_1 = 0.05$, $\sigma_x = 10$, $r = 7$



Слика 4.8 - а) оригинална слика „sl_2.jpg“ (90 × 100). Сlike б) – е) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.03. Вриједности параметара: $\sigma_I = 0.05$, $\sigma_X = 50$, $r = 10$

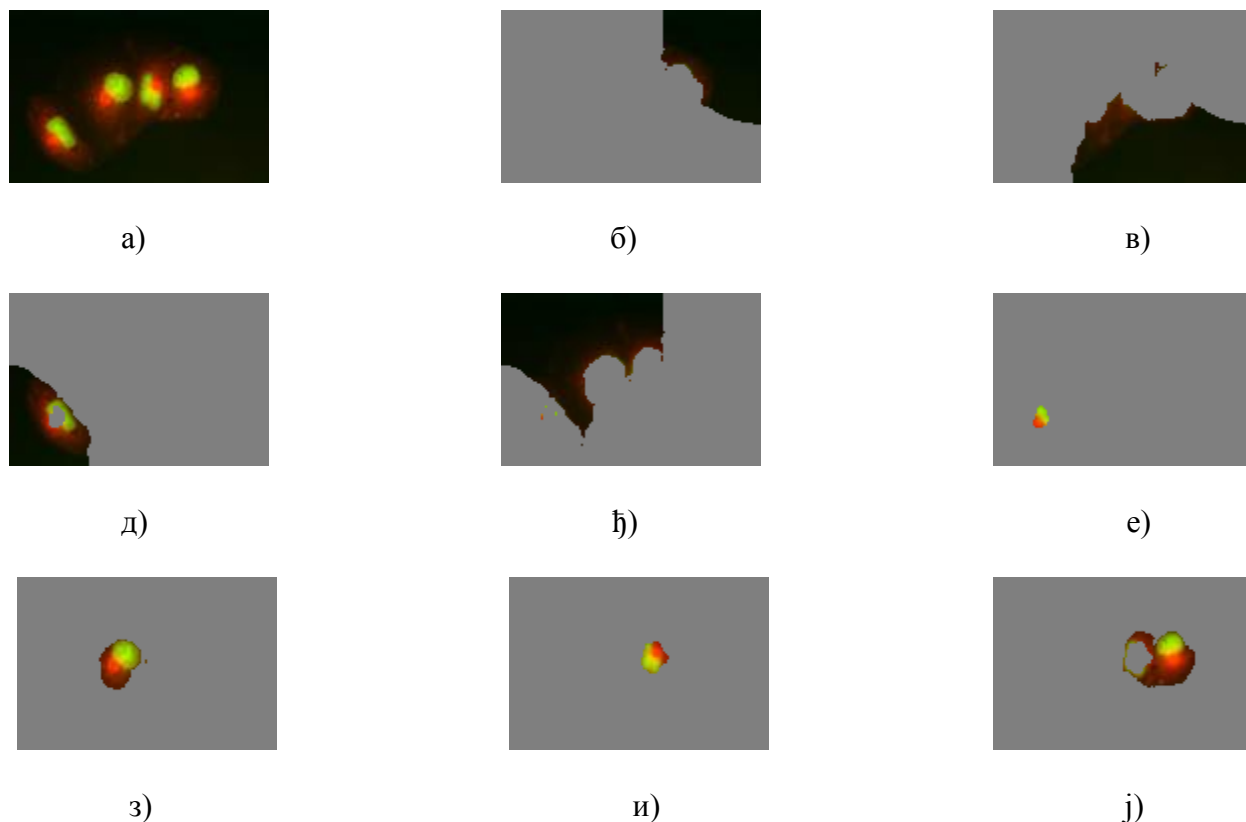


Слика 4.9 - а) оригинална слика „sl_3.jpg“ (91 × 88). Сlike б) – е) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.05. Вриједности параметара: $\sigma_I = 0.045$, $\sigma_X = 10$, $r = 7$

Хистологија је биолошка дисциплина која проучава ткива. Савремена хистолошка методологија се заснива на кориштењу микроскопа. При хистолошкој анализи, добијена ткива се ставе на стакла и фиксирају како би се спријечило изумирање узорка. Затим се узорци сперу са етанолом, како би се узорак дехидрирао, након чега се очисти са

одговарајућим хемикалијама. Овај процес траје од 12 до 16 часова, и у том периоду узорак ткива ће бити у таквом физичком стању да је могуће распознати симптоме обољелог ткива. Да би ткиво са својим ћелијама и органелама било видљиво ономе ко га посматра, неопходно је обојити ткиво одређеним агенсима [16]. При анализи ових ткива корисно је користити сегментацију слика у боји, јер нам она омогућава да издвојимо оне ћелије и једра која су нам од значаја. Издвајање, односно пребројавање ћелија и једара је посао лабораната, међутим они услед умора или неких других спољних фактора могу погријешити. Управо због тога, сегментација постаје све значајнија у овој области и тежи се реализацији алгорита који би вршио ова пребројавања, а да при томе не долази до грешке.

У овом дипломском раду вршена је примјена алгорита и на хистолошке слике, међутим резултати нису тако добри као у случају слика чији су резултати раније наведени. Примјер сегментације хистолошке слике дат је на Слици 4.10. С обзиром да не постоји јасна граница између црвене и зелене боје епитела, алгоритам их издваја као један регион, иако би требало да свака од боја приказаног епитела буде засебно издвојена. Такође, ни сегментација позадине није коректна, јер је она издјељена на више региона, а требало би да припада само једном региону.



Слика 4.10 – а) оригинална слика „epitel.jpg“ (130 × 86). Сlike б) – j) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.04. Вриједности параметара: $\sigma_1 = 0.04$, $\sigma_x = 4$, $r = 4$

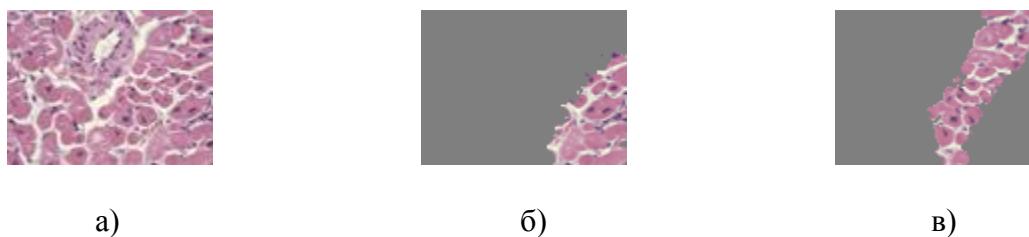
Са промјеном параметара можемо довести до ситуације да се засебно издваја црвени, а засебно зелени регион, мада и у том случају поједини региони садрже обе боје истовремено. Што се тиче подјеле позадине, она и даље остаје иста, односно подијељена на више региона. Оно што је карактеристично при оваквој сегментацији је чињеница да тада настају и региони који садрже само по неколико пиксела црвене, односно зелене боје, и у том случају број региона цјелокупне сегментације је врло велик. Конкретно за параметре: $N_{cut} = 0.3$, $\sigma_I = 0.03$, $\sigma_X = 4$ и $r = 4$, реализована су 24 региона. На Слици 4.11 је, ради илустрације, приказано само неколико таквих региона.



Слика 4.11 – а) зелени регион, б) црвени регион, в) неколико пиксела зеленог региона, г) неколико пиксела црвеног региона

Због чега је толико тешко извршити добру сегментацију једино се може објеснити уколико посматрамо како су боје представљене у HSI колор простору. Наиме, на основу Слике 4.2 може се закључити да се нијансе црвене, жуте и зелене боје налазе једна поред друге, па уколико се при сегментацији параметар σ_i подеси тако да боја нема превелики утицај при подјели, региони који се добију садржаће све три боје. С друге стране, ако је параметар σ_i подешен тако да боја има велики утицај при подјели, добиће се такви региони чији се објекти на сваком од њих међусобно разликују за неколико нијанси исте боје. Зато имамо регионе који садрже само по неколико пиксела боје одређене нијансе.

На Слици 4.12 је приказан још један примјер примјењивања алгоритма на хистолошку слику. У овом случају такође имамо лошу сегментацију, а са промјеном параметара можемо постићи ситуацију као код слике „epitel.jpg“, гдје ћемо имати или превелики број региона на којима ће се издвајати по један или неколико пиксела, или мањи број региона на којима ће се налазити објекти различитих боја.



а)

б)

в)



Слика 4.12 – а) оригинална слика (103×77). Сlike б) – њ) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.08. Вриједности параметара: $\sigma_1 = 0.06$, $\sigma_x = 4$, $r = 7$

4.2. Утицај параметара алгоритма *нормализованих пресека* на резултате сегментације слика у боји

Резултати сегментације слика у боји првенствено зависе од начина на који се изабери вриједности кориштених параметара. Наиме, помоћу параметра N_{cut} се дефинише критеријум стабилности, односно његова вриједност одређује да ли ће се наставити партиционисање већ сегментованих дијелова на слици. Принцип по којем се врши поновна сегментација у зависности од N_{cut} вриједности је већ детаљно описан у 2. Глави. Може се рећи да N_{cut} параметар има највећи утицај при сегментацији, јер управо од њега зависи колико ћемо имати региона након целокупног поступка сегментације. Његова вриједност се, за све слике које су тестиране у овом дипломском раду, кретала у границама од 0.008 до 0.08, што је зависило како од карактеристика слике, тако и од вриједности осталих параметара.

Остали параметри су σ_1 , σ_x и r . Параметар σ_1 је параметар који директно утиче на вриједности елемената унутар тежинске матрице. У зависности од тога колика је његова вриједност, у тежинској функцији већи утицај ће имати први, односно други експоненцијални члан, а то значи да ће се приликом сегментације већи значај придавати боји, односно просторној локацији пиксела. Аналогно вриједи и за параметар σ_x . Параметар σ_1 се налази у називнику експоненцијалног члана који садржи функцију F_i која се базира на информацијама о боји. Вриједност овог параметра се, за слике у боји, кретала од 0.04 до 0.08. Иако се на први поглед чини да је варијација овог параметра мала, она има велики утицај. Вриједности функције F_i се крећу у опсегу од 0 до 1, јер се у том опсегу представљају вриједности пиксела слике. Према томе, што је однос између бројника и називника већи, вриједност експонента ће бити мања. С обзиром да експоненцијални члан има негативан експонент то значи да ће вриједност експоненцијалног члана у том случају бити већа. Вриједности овог експоненцијалног члана знају бити реда 10^{-3} па све до 10^{-12} . На основу свега претходног може се закључити да се са повећавањем вриједности параметра σ_1 повећава вриједност експоненцијалног члана који утиче на сегментацију слике на основу боје. Према томе, уколико бисмо на сликама на којима је сегментација успјешно извршена повећали вриједност само σ_1 параметра, добио би се мањи број партиција, а то значи да се не би вршило партиционисање оних сегмената чија је даља подјела потребна. Закључак је да у том случају боја нема довољно утицаја на сегментацију слике, па тада настају региони на којима се налазе објекти различитих боја. С друге

стране, уколико бисмо смањивали вриједност σ_1 параметра, као резултат сегментације бисмо добили већи број региона него што је потребно, јер би у том случају дошло до подјеле пиксела који се разликују само по нијанси боје на два или више региона иако ти пиксели у ствари припадају једном објекту.

Што се тиче параметра σ_x , његово математичко објашњење је исто као и код параметра σ_1 , односно он на исти начин утиче на експоненцијални члан као и σ_1 . Експоненцијални члан на који он утиче у свом бројнику садржи функцију X_i која дефинише просторну локацију пиксела. А та просторна локација пиксела се одређује на основу Еуклидске удаљености, што је у 2. Глави детаљније објашњено. Вриједности овог параметра су се приликом тестирања кретале од 4 до 20. С обзиром да се бројник експоненцијалног члана израчунава према Еуклидској удаљености, а при томе се мора задовољити услов да је $\|X_i - X_j\| < r$, његове вриједности се крећу у опсегу од 0 до r^2 . Вриједност ће бити једнака нули када се пиксел упоређује сам са собом, а r^2 , односно нешто мањи од ове вриједности, када се пиксел упоређује са себи најудаљенијим пикселом за који вриједи услов $\|X_i - X_j\| < r$. Према томе, на основу претходне анализе се може закључити да је вриједност експоненцијалног члана увијек била већа од нуле и мања или једнака јединици. Стога, њена вриједност је знатно већа у односу на вриједност првог експоненцијалног члана и она може утицати на коначну вриједност тежинске функције. Тежинска функција ће имати мању вриједност уколико је овај експоненцијални члан ближи 0, а већу када је експоненцијални члан близак или једнак јединици. Вриједности параметра σ_x , које су кориштене при тестирању слика нису имале великог утицаја на резултате сегментације, на основу чега се може извући закључак да је параметар σ_1 имао већи значај при сегментацији слика у боји него параметар σ_x .

Параметар r одређује дозвољену удаљеност између два пиксела да би се могло вршити њихово поређење. Што је овај параметар већи, то значи да се сваки пиксел пореди са већим бројем пиксела у његовом сусједству. Избор веће вриједности параметра r има своје позитивне и негативне стране. Позитивна страна је у томе што ће се у том случају узети већи број пиксела приликом поређења, због чега ће и сегментација бити боља. Мада, не мора увијек да значи да већи број пиксела за поређење подразумијева и бољу сегментацију, управо због вриједности осталих параметара као и због особина слике чија се обрада захтијева. Негативна страна велике вриједности r је повезана са брзином извршавања сегментације, односно у том случају тежинска матрица има више ненултих вриједности, па је израчунавање сопствених вектора и њихових сопствених вриједности знатно дужег трајања. Зато је потребно одредити такву вриједност параметра r која ће давати добре резултате у погледу сегментације, али и да ће вријеме извршавања бити у разумним границама. У наредном поглављу ћемо се више позабавити са временом извршавања алгорита.

4.3. Вријеме извршавања сегментације

У 2. Глави смо рекли да је вријеме извршавања алгоритма *нормализованих пресека* једнако $O(n)$ гдје је n број пиксела слике. Иако смо већ нагласили чињеницу да је при сегментацији слика акценат стављен на то колико је успјешна сегментација, ипак је потребно да обратимо пажњу на вријеме које се утроши при примјењивању датог алгоритма на неку слику. Сlike које смо користили при тестирањима нису прелазиле величину од 160×130 пиксела, тако да извршавање сегментације никада није трајало дуже од 3 минуте. Вријеме извршавања смо мјерили помоћу *Matlabових* функција *tic* и *toc*. Колико дуго ће се извршавати сегментација слике зависи од тога колике су вриједности параметара, нарочито када је у питању вриједност параметра r , који одређује са коликом области пиксела ће се поредити сваки пиксел слике. Карактеристике слике такође утичу на брзину извршавања, односно у зависности од сложености слике као и од тога какве објекте она има на себи имаћемо мањи или већи број рекурзивних понављања алгоритма. Још један од фактора који утиче на брзину извршавања алгоритма су и спецификације рачунара на којем се врши сегментација. *Shi* и *Malik* су као могуће ријешење овом проблему у свом раду навели мултирезолуциону имплементацију, због које се вријеме извршавања на сликама већих димензија може знатно смањити. Међутим ми се нисмо бавили овим проблемом у склопу дипломског рада, па ову чињеницу наводимо као могућност усавршавања постојећег кода за сегментацију.

4.4. L^*a^*b колор простор

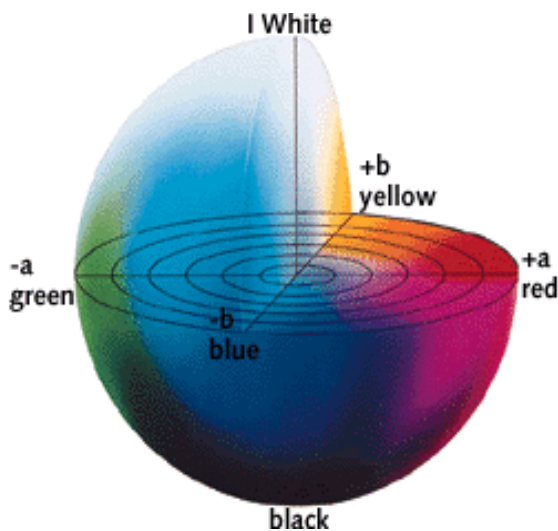
L^*a^*b колор простор је један од најраспрострањенијих и најкориснијих колор простора који је настао математичком трансформацијом CIE XYZ⁸ колор простора. Он се данас често користи као скраћеница за неформални CIE 1976 (L^*a^*b) колор простор, који се још и зове CIE Lab, гдје су L^* , a^* и b^* координате овог простора. L^*a^*b колор простор се користи у системима за управљање бојама, као и у мапирању гамута⁹. Према томе, овај колор простор не представља формат који се директно може приказати на екрану, с обзиром да његов гамут обухвата укупни видљиви спектар [17].

CIE L^*a^*b (CIE Lab) колор простор описује све боје које су видљиве људском оку. Три координате овог простора представљају свјетлину боје ($L^*=0$ даје црну а $L^*=100$ показује дифузно бијелу), њену позицију између црвене/магенте и зелене (a^* , чије негативне вриједности показују зелену док позитивне вриједности показују црвену/магенту), као и позицију између жуте и плаве боје (b^* , чије негативне вриједности показују плаву а позитивне жуту боју). Нулте вриједности компонената a^* и b^* указују на

⁸ CIE XYZ колор простор је један од првих математички описаних колор простора, настао као посљедица изучавања перцепције боја. CIE XYZ колор простор, који је познат и под називом CIE 1931 колор простор, је креиран од стране International Commission on Illumination (CIE) 1931. године.

⁹ Гамут – дијаграм хроминанси који даје све боје које просјечни посматрач може да види [1].

сивило које се приближава бијелој боји када компонента L расте [18]. Начин њиховог представљања у тродимензионалном простору дат је на Слици 4.13.



Слика 4.13 – Представљање боја у L^*a^*b колор простору

Формула која се врши конверзију између L^*a^*b и RGB колор простора је прилично компликована с обзиром да RGB колор модел зависи од уређаја у којем се користи. Зато је прво потребно да се RGB вриједности трансформишу у одређени апсолутни колор простор, као што су $sRGB$ или $Adobe RGB$. Ово прилагођавање ће такође бити зависно од уређаја, али ће резултујући подаци након трансформације бити независни од уређаја, што ће омогућити да се ти подаци трансформишу у $CIE 1936$ колор простор, након чега слиједи и трансформација у L^*a^*b простор. Захваљујући *Матлабу* овај скуп трансформација се може постићи кориштењем двије функције на сљедећи начин:

```
cform = makecform('srgb2lab');
lab = applycform(I,cform);
```

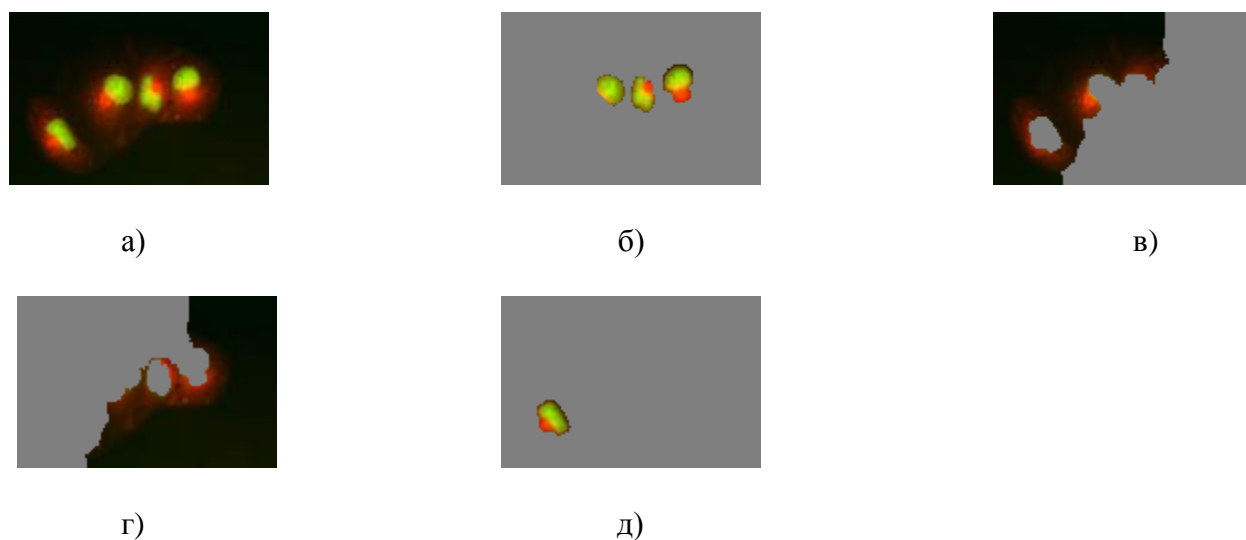
гдје је I слика представљена у RGB колор простору [14].

4.4.1. Примјена алгоритма *нормализованих пресјека* на слике представљене у L^*a^*b колор простору

Да би се реализовала сегментација слика у L^*a^*b колор простору, прво је било потребно извршити трансформације помоћу двије *Матлабове* функције које су наведене у претходном поглављу. Након пребацивања RGB слике у L^*a^*b колор простор, прелази се на реализацију тежинске матрице. У овом случају се вектор F_i не састоји од функција \sin и \cos , него га чине само три компоненте L^*a^*b колор простора. Овакав начин дефинисања вектора F_i доводи до другачијег избора вриједности за параметар σ_1 .

При тестирању алгоритма кориштен је исти скуп слика као у случају када је кориштен HSI колор модел. Након узимања одговарајућих вриједности параметара добијени су веома слични резултати као и у претходним разматрањима. У случају тестирања слика из природе и микробиолошких слика резултати су били идентични, односно број региона као и објекти који су се налазили на њима нису се разликовали од оних који су се добијали код HSI колор слика. Из тог разлога те резултате нећемо опет приказивати. Овдје ћемо се више позабавити резултатима хистолошких слика, јер су оне и биле главни разлог због којег смо се одлучили да у склопу тестирања алгоритма користимо L^*a^*b колор модел.

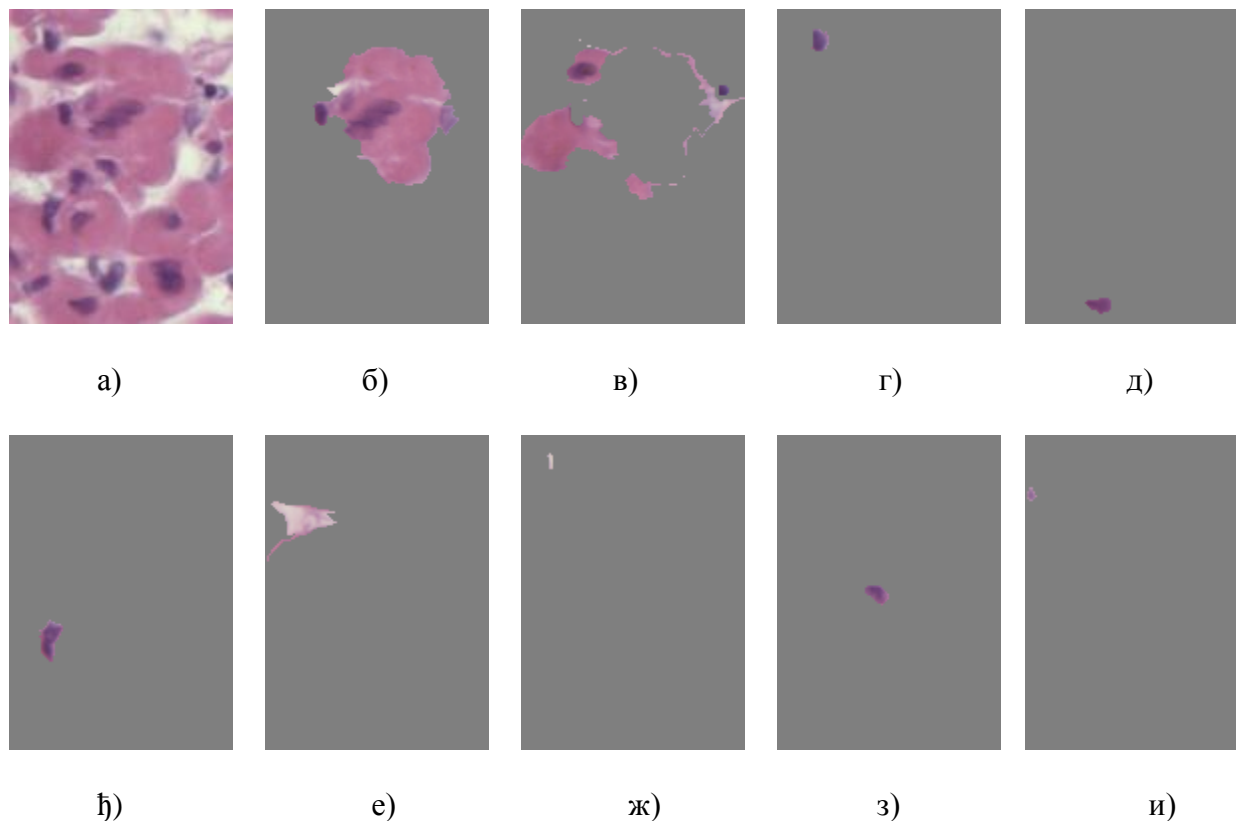
Прије него што прикажемо неке од резултата, потребно је напоменути да су вриједности свих параметара углавном остали у истом домену као и код тестирања у HSI колор простору, осим параметра σ_I чије су вриједности у овом случају били цијели бројеви и кретали су се између вриједности 4 и 10. На Слици 4.14 су приказани резултати слике „epitel.jpg“, када се она представи у L^*a^*b колор простору.



Слика 4.14 – а) оргинална слика „epitel.jpg“ (130×86). Сlike б) – д) приказују добијене регионе, гдје је N_{cut} вриједност мања од 0.04. Вриједности параметара: $\sigma_I = 6$, $\sigma_X = 10$, $r = 10$

На основу добијених региона, видимо да је дошло до издвајања зеленог региона, међутим може се примјетити да ти региони поред зелене садрже и по мали дио црвене боје, што значи да и у овом случају није дошло до потпуног раздвајања ћелија које имају различите боје. Иако сегментација није успјешно извршена, можемо слободно рећи да су добијени резултати бољи него у случају када се користи HSI колор модел. Овдје имамо мањи број региона којима припада позадина, док региони, на којима се налазе објекти, имају много јасније издвојене ћелије епитела. Са промјеном параметара добијамо исту ситуацију као и код HSI колор слика, гдје се реализују региони које садрже по неколико пиксела. Лоши резултати се добијају и у случају хистолошке слике „hist.tif“. Њена оргинална величина је 2048×1536 пиксела, међутим због проблема спорог извршавања било је потребно да се

она смањи. Након смањивања њена величина је износила 130×86 пиксела и у том случају су објекти од интереса (једра) постали мање видљиви, што представља један од разлога због којих се сегментација тешко извршава. Наиме, уколико бисмо са слике оргиналне величине издвојили само један њен дио, и на њему тестирали алгоритам, добили би се региони приказани на Слици 4.15. С обзиром да су са овим параметрима формирана 33 региона, ми смо приказали само неке од њих ради илустрације добијене сегментације.



Слика 4.15 – а) оргинална слика (157×112). Сlike б) – и) приказују неке од добијених региона, гдје је N_{cut} вриједност мања од 0.06. Вриједности параметара: $\sigma_I = 3$, $\sigma_X = 5$, $r = 5$

Може се примјетити да је у овом случају дошло до издвајања појединих једара, мада су резултати сегментације и даље далеко од задовољавајућих. Нарочито из разлога што се и у овом случају формирају такве партиције, које садрже само по један или евентуално по неколико пиксела слике. Међутим, може се закључити да би се у случају бржег извршавања, алгоритам могао примјенити на слике већих димензија, због чега би и резултати сегментације били побољшани у одређеној мјери.

5. ЗАКЉУЧАК

Сегментација слика у боји је област дигиталне обраде слике која сваким даном постаје популарнија, с обзиром на многобројне могућности примјене. Из тог разлога настају различита рјешења која се баве овим проблемом. Постоји велики број алгоритама сегментације слика у боји, а у овом раду је описан један од њих – *нормализовани пресјек*. Наиме, прво је дат детаљан математички опис алгоритама *нормализованих пресјека*, његове основне карактеристике, те начин на који се он примјењује при сегментацији слика.

Кључна ствар овог алгорита је у томе да се са слике која се сегментира прво издвајају веће области слике, након чега слиједи добијање све мањих и мањих сегмената. Односно циљ је да се при сегментацији издвоје добро дефинисани сегменти, који представљају битне карактеристике слике. То се постиже кориштењем критеријума *нормализованог пресјека* који спријечава издвајање изолованих пиксела до чега долази уколико се користи обичан критеријум *пресјека*. Минимизација овог критеријума представља *NP-комплетан* проблем. Међутим, предност алгорита је у томе што се може наћи апроксимативно дискретно рјешење представљањем овог проблема помоћу уопштеног система сопствених вриједности.

Приликом испитивања ефикасности алгорита, кориштене су различите класе слика. При томе се сегментација слика заснивала на интензитету и боји пиксела. На основу добијених резултата можемо рећи да је алгорита испунио очекивања када је у питању издвајање добро дефинисаних сегмената. У случају синтетичких, као и слика из природе сегментација је успјешно извршена. То се не може рећи за класу хистолошких слика, па кажемо да је у том случају алгорита неуспјешан. Приликом поређења резултата који се добијају када се алгорита примијени на исте слике, при чему је једна сива а друга у боји, потврђена је чињеница да боја представља веома битан фактор при сегментацији слика. На резултате сегментације великим дијелом утичу кориштени параметри, с тим да избор параметара зависи од особина слике која се треба сегментирати. Из тог разлога је врло тешко оправдати кориштене вриједности параметара. Према томе, постизање добрих резултата сегментације зависи од упорности аутора при подешавању параметара. Још један проблем који се јављао при извршавању алгорита је тај да су се код неких слика издвајали појединачни пиксели иако сам алгорита користи такав критеријум који спријечава ту ситуацију.

Могућа побољшања реализованог кода за сегментацију слика се првенствено односе на његово вријеме извршавања. Односно, убрзавање рада алгорита подразумијева увођење новог начина израчунавања сопствених вектора, што би омогућило примјену алгорита и на слике већих димензија. Проблем издвајања појединачних пиксела би се могао ријешити уколико бисмо при реализацији алгорита користили још један параметар који би дефинисао величину области пиксела до које би се извршавала сегментација. Једна од идеја за будући рад који би се бавио истом проблематиком је да се више обрати пажња на тестирање хистолошких слика, односно да се детаљније проуче параметри, како би се дошло до вриједности које би давале добру сегментацију ове класе слика.

6. ЛИТЕРАТУРА

- [1] З. Бабић, *Дигитална обрада слике – скрипта*, Бања Лука, 2005.
- [2] R. Lukas, K. N. Plataniotis, *Color Image Processing – Methods and Applications*, Taylor & Francis Group, University of Toronto, Toronto, Ontario, Canada, 2007.
- [3] H. D. Cheng, X. H. Jiang, Y. Sun, J. L. Wang, „Color Image Segmentation – Advances & Prospects“, Dept. of Computer Science, Utah State University
- [4] Y. J. Zhang, „Advances in Image and Video Segmentation“, Tsinghua University, Beijing, China, IMR Press, 2006.
- [5] Terry S. Yoo, *Insight into Images – Principles and Practice for Segmentation, Registration, and Image Analysis*, National Library of Medicine, NIH, Office of High Performance Computing and Communications, 2004.
- [6] X. Cufi, X. Munoz, J. Freixenet, J. Marti, A Review on Image Segmentation Techniques Inegrating Region and Boundary Information, October 31, 2001.
- [7] М. В. Поповић, *Дигитална обрада слике*, Академска мисао, Београд, 2006.
- [8] Jianbo Shi and Jitendra Malik, Normalized Cuts and Image Segmentation, IEEE Transactions on PAMI, Vol. 22, No. 8, Aug. 2000.
- [9] Д. М. Цветковић, И. Б. Лацковић, М. Ј. Меркле, *Математика I – Алгебра*, Академска мисао, Београд, 2004.
- [10] R. Szeliski, *ComputerVision – Algorithms and Applications*, May 17, 2010.
- [11] R. Kimmel, R. Klette, A. Sugimoto, Computer Vision - ACCV 2010, 10th Asian Conference on Computer Vision, Queenstown, New Zeland, Revised Selected Papers, Part III, November 2010.

- [12] S. Janson, *Stephen Janson On Digital Photography*, O'Reilly Media, Inc. 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2006.
- [13] S. Nagabhusana, *Computer Vision and Image Processing*, New Age International Publishers
- [14] R. C. Gonzales, R. E. Woods, S. L. Eddins, *Digital Image Processing Using MATLAB @*, 2004.
- [15] Wikipedia *The Free Encyclopedia* – Mikrobiologija
<http://bs.wikipedia.org/wiki/Mikrobiologija>, задњи пут посјећено 03.03. 2011. године
- [16] Wikipedia *The Free Encyclopedia* – Histologija
<http://sr.wikipedia.org/sr-el>, задњи пут посјећено 03.03. 2011. године
- [17] M. Masoodian, S. Jones, B. Rogers, Computer human interaction: 6th Asia Pacific Conference, APCHI 2004, New Zeland, Juny/July 2004.
- [18] J. Borch, M. B. Lyne, R.E. Mark, C.C. Habeger, Handbook of Physical Testing of Paper, Volume 2, Second Edition, Revised and Expanded.

7. ДОДАТАК

Као што је већ наглашено, за имплементацију алгоритма је кориштен програмски пакет *Матлаб*, при чему је свака од реализованих функција написана у облику `.m` фајла. Умјесто навођења цијелог кода, овдје ћемо само набројати из којих све функција се састоји алгоритам, а њихов детаљан опис и реализација је приложена на CD-у.

Ради боље прегледности резултата, реализовани су `.m` фајлови `proba.m`, `proba1.m` и `proba2.m` у којима је алгоритам по више пута примијењен на различите сиве слике, слике у боји када се користи HSI колор модел и слике у боји када се користи L^*a^*b колор модел.

Матлабови .m фајлови

- `segmentacija.m`
- `tezinska.m`
- `tezinska_color.m`
- `tezinska_color_lab.m`
- `particija.m`
- `sopstveniVektor.m`
- `izracunajNcut.m`