

УНИВЕРЗИТЕТ У БАЊАЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

Алексеј Аврамовић

СЕГМЕНТАЦИЈА ДИГИТАЛНИХ СЛИКА
ПРИМЕНОМ MEAN SHIFT АЛГОРИТМА

Дипломски рад

Бања Лука, јул 2007.

Тема: СЕГМЕНТАЦИЈА ДИГИТАЛНИХ СЛИКА ПРИМЕНОМ
MEAN SHIFT АЛГОРИТМА

Кључне речи:

Mean shift алгоритам
Филтрирање
Сегментација
Дигитална обрада сигнала

Комисија:

проф. др Ђемал Колонић, председник
проф. др Зденка Бабић, ментор
мр Владимир Рисојевић, члан

кандидат:
Алексеј Аврамовић

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
КАТЕДРА ЗА ОПШТУ ЕЛЕКТРОТЕХНИКУ

Предмет: ДИГИТАЛНА ОБРАДА СИГНАЛА

Тема: СЕГМЕНТАЦИЈА ДИГИТАЛНИХ СЛИКА ПРИМЕНОМ MEAN SHIFT АЛГОРИТМА

Задатак: Потребно је дефинисати појам сегментације слика и одредити њено место и значај међу техникама дигиталне обраде слике. Известити математички модел и објаснити све параметре mean shift алгоритма. Дефинисати његове карактеристике и могућности примене. Дати преглед основних функција заснованих на mean shift алгоритму које се користе при робусној анализи простора обележја. Практично реализовати филтрирање слика применом mean shift алгоритма. Практично реализовати сегментацију применом mean shift алгоритма.

Ментор: **проф. др Зденка Бабић**

Кандидат: **Алексеј Аврамовић, 73/02**

Бања Лука, јул 2007.

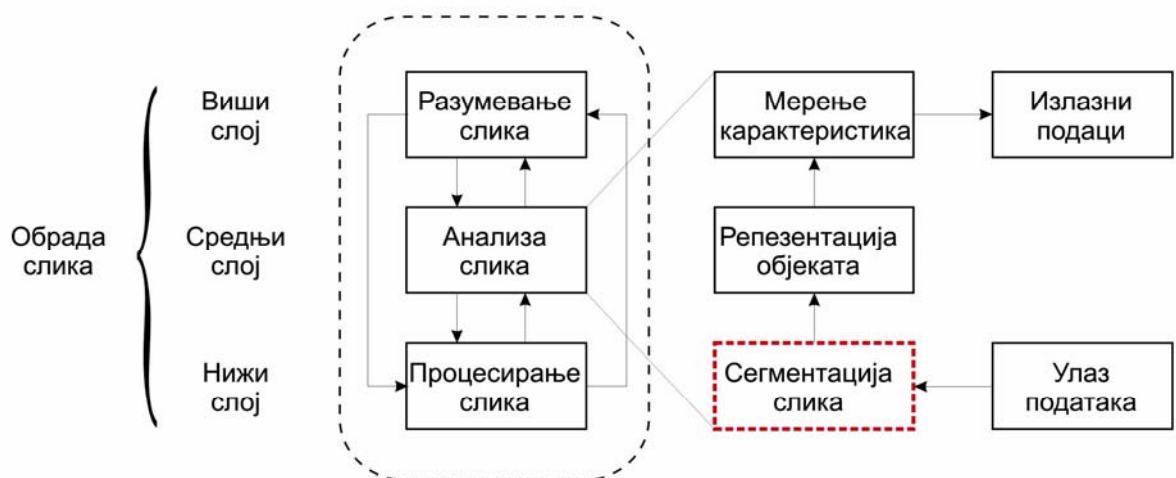
Садржај

1. Увод	1
1.1. Дефиниција сегментације	1
1.2. Основне карактеристике процеса сегментације слика и проширење дефиниције	4
1.3. Нивои истраживања сегментације слика	6
1.4. Организација рада	6
2. Математички модел mean shift алгоритма	7
2.1. Mean shift алгоритам и анализа простора обележја	8
2.1.1. Естимација функције густине вероватноће методом Парзенових прозора	9
2.1.2. Mean shift алгоритам	11
2.1.3. Избор кернела	16
2.1.4. Избор пропусног опсега	17
2.2. Карактеристике mean shift алгоритма	19
3. Робусна анализа простора обележја	21
3.1. Дефинисање простора обележја	22
3.2. Налажење модова функције	23
3.3. Филтрирање са чувањем ивица	24
3.4. Сегментација мирних слика	25
3.5. Сегментација видеа и праћење објеката	26
4. Реализација филтрирања и сегментације	29
4.1. Филтрирање слика	30
4.2. Сегментација слика	34
5. Закључак	37
Додатак А	39
Додатак Б	41
Литература	43

1. УВОД

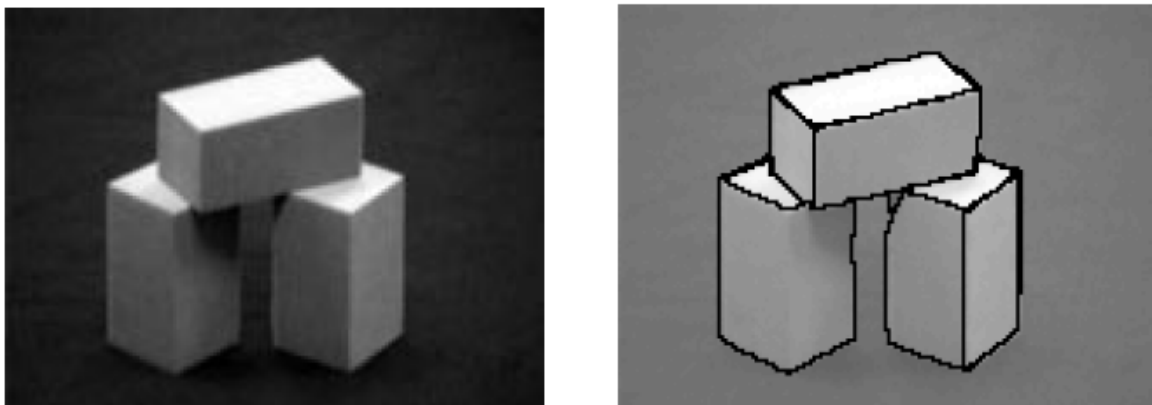
1.1 Дефиниција сегментације

Дигиталне слике, уопштено посматрано, обухватају широк спектар медија, који служе за визуелно представљање података. Под појмом медија подразумевамо све начине визуализације према људској перцепцији, дакле, појам дигиталне слике обухвата дигиталне фотографије, мирне слике, анимације, видео записе, графике, а поједини аутори додају и текст [1]. У савременој литератури усаглашена су три основна нивоа поделе техника за обраду дигиталних слика: нижи слој - обрада дигиталних слика, средњи слој - анализа дигиталних слика и виши слој - разумевање или интерпретација дигиталних слика. Средњи слој, анализа дигиталних слика (у даљем тексту слика), има за циљ екстракцију информација помоћу сегментације слика, репрезентације слика и мерења карактеристика.



Слика. 1.1: Сегментација

Дакле, сегментација слика је први и најкритичнији корак анализе слика, и то из најмање два разлога. Први разлог је очигледан утицај тачности сегментације, као првог корака, на тачност мерења карактеристика које се одвија на основу информација које се добијају сегментацијом. Други разлог је то, што се и поред више од четрдесет година истраживања у области сегментације слика, није дошло до уопштеног модела, који ће дати задовољавајуће резултате за све врсте дигиталних слика. Сегментација слика се често описује као процес који треба да издвоји одређене делове слика и да издвоји одређене објекте на слици. Дефиниција каже да сегментација има за циљ кластеризацију, односно груписање, пиксела у смислене регионе на слици, тј. регионе који одговарају појединим површинама на слици, објектима или деловима природних објеката на слици [7]. Другим речима, можемо рећи да сегментација треба да препозна семантички смислене целине на сликама. Наведена дефиниција је врло захтевна са аспекта реализације, јер су алгоритми за сегментацију развијани током дужег периода у већој или мањој мери успевали да се на адекватан начин изборе са шумовима на слици али не да притом сачувају семантичке ивице на задовољавајући начин. Пример сегментације је дат на Слици 1.2, где можемо да уочимо један од начина како се могу издвојити објекти на слици.



Слика. 1.2: Пример сегментације

Алгоритми за сегментацију слика се могу користити у системима за препознавање објеката, за компресију слика, у системима за видео надзор и праћење објеката преко видео надзора, за обраду слика, у системима за претраживање слика на основу садржаја, мултимедијалним апликацијама и генерално свим системима где је потребно уочити семантичке целине на сликама. Приликом дефиниције самог појма сегментације, постављају се и нека дискутабилна питања [7]. Какав садржај на слици одређује сегмент и како можемо очекивати да ће алгоритам препознати управо одређени сегмент дефинисан појединим вредностима пиксела са слике? Да ли можемо тај проблем да дефинишемо математички и како можемо да га решимо? На крају, на основу којег критеријума можемо прецизно да утврдимо да ли је сегментација била коректна?

Са развојем мултимедијалних и телекомуникационих мрежа и апликација, долази и до повећаног интересовања за објектно оријентисану сегментацију слика, при чему је било веома важно да се сегменти на обрађиваним сликама и видеу, састоје од семантички смислених целина. Овај задатак није тешко испунити када су у питању једноставни објекти, јер се тада сегментација врши једноставним груписањем пиксела са сличним обележијма ниског нивоа (као што је на пример боја). Са друге стране, сегментација слика са сложеним објектима обично захтева неку врсту надгледаног

процеса или обезбеђивање одређене информације високог нивоа, да би се издвојили објекти који одговарају људској перцепцији. У поређењу са сегментацијом слика, сегментација видеа је обично подржана додатним информацијама о кретању објекта, па је реализација система за праћење објеката мање рачунски захтевна.

Алгоритме за сегментацију можемо да поделимо према садржају на основу којег се врши сегментација, на основу аутоматизма сегментације, на основу постојања придружених модела или некон другом критеријуму [1]. Према садржају на основу којег се врши сегментација алгоритми могу да буду засновани на пикселима, регионима, садржају, ивицама, објектима, семантици итд. На пример, алгоритми засновани на садржају пиксела користе обележја ниског нивоа, као што су боја, текстура, интезитет или информацију о разликама између узастопних пиксела. Алгоритми за сегментацију засновани на региону користе информације о ивицама, полигонским границама, облицима и информације о статистици текстуре за поједине објекте. Објектно оријентисани методи користе придружене карактеристике за сваки објекат [1]. Подаци о карактеристичним обележјима могу бити преузети, тј. екстраховани са различитих нивоа, али да би се очувала информација о семантичком смислу на слици, потребно је очувати информације о семантичким ивицама на слици приликом обраде.

Алгоритме за сегментацију слика можемо да поделимо према аутоматизму, и то на аутоматске и полуаутоматске. Полуаутоматски алгоритми подразумевају вођење процеса од стране корисника преко постављања одређених почетних услова. То су нпр. издвајање ивица помоћу активних контура, где је потребно да корисник зада неку иницијалну тачку. Аутоматски алгоритми за сегментацију се извршавају без задавања почетних услова, какви су нпр. алгоритми засновани на теорији графова. Такође, још један од начина да поделимо алгоритме за сегментацију је да ли се користи придружени модел (нпр. функције густине вероватноће) или не. Да би се проценили непознати параметри могу се користити методе засноване на минимизацији математичког очекивања или Бајсовог учења. За сегментацију сложених слика се често користи стохастичко моделовање, као што је Марковљево случајно поље које је еквивалентно са Гибсовом дистрибуцијом која решава проблем минимизације енергије [1]. Овакви стохастички модели захтевају естимацију функције густине вероватноће. Понекад се уместо стохастичког моделовања може употребити физички модел, нпр. приликом сегментације на основу термалних карактеристика код инфрацрвених слика, а такође, користе се и модели рефлексије светлости приликом сегментације на основу боје.

Генерално гледано, приликом издвајања семантичких целина на слици, можемо да кажемо да су ивице, односно дисконтинуитети, од највећег значаја за издвајање природних објеката, док су варијације обележја (у боји, интензитету, итд.) непожељне сметње. Стога, приликом сегментације, потребно је уконтити поменуте варијације обележја или, другим речима, груписати елементе слике (нпр. пикселе) у једну групу ако су варијације (нпр. интензитета) тих пиксела, унутар неког опсега. Такву врсту сегментације можемо применити у просторном (енг., *spatial*) домену, у домену одређеног обележја (колор простора¹), у временском домену или у здруженом домену. Даље, ако таква сегментација слике издвоји велики број сегмената, сегменти се могу даље спајати према неком одређеном критеријуму (у пракси углавном према просторним координатама).

¹ У литератури на енглеском језику, *range domain*

У последње време развија се велики број алгоритама за сегментацију у видео снимцима, првенствено због све веће потребе за аутоматским праћењем објеката преко различитих врста видео надзора. Постоје три основна модела у сегментацији слика из видео записа. Први модел подразумева сегментацију у *spatial-range* домену, па затим праћење издвојених објеката у временском домену, док други модел подразумева екстракцију трајекторија објеката или региона од интереса, према сличностима кретања у временском домену. Трећи модел подразумева сегментацију у троструком здруженом домену, тј. *spatial-range-time* домену.

Постоје неколико предложених начина за поделу техника за сегментацију слика, од којих се у литератури најчешће сусреће подела у три групе: (1) *thresholding*² или његово вишедимензионално проширење, кластеризација, (2) екстракција региона и (3) детекција ивица. Међутим, оваква подела не може строго да разграничи поједине алгоритме за сегментацију, који према карактеристикама не припадају само једној техници, јер је нпр. проблем екстракције региона само један специјалан случај кластеризације. С обзиром на то можемо да кажемо да постоје две основне технике сегментације слика, *кластеризација* и техника *издвајања ивица*. Можемо да се задовољимо претходном поделом, без обзира што у литератури постоје разне дискусије и дефинисане карактеристике резултујућих техника за сегментацију, из разлога што било која друга подела има значајна преклапања, а са друге стране дискусија у том правцу није од значаја у овом раду.

1.2. Основне карактеристике процеса сегментације слика и проширење дефиниције

Ако сегментацију дефинишемо као процес којим делимо слику на непреклапајуће делове, који могу али и не морају да одговарају деловима семантичких целина на слици, можемо да уочимо неке логичне карактеристике:

1. Унија свих непреклапајућих региона треба да буде цела слика
2. Сегменти треба да буду што је могуће више хомогени и униформни, са одређеним карактеристикама
3. Унутрашњост сегмената треба да буде што је могуће униформнија
4. Суседни региони, један у односу на другог, треба да имају значајно другачије карактеристике униформности
5. Границе свих региона, тј. сегмената, треба да буду што једноставније и што прецизније

Сегментацију слике R можемо формално да дефинишемо као раздвајање на непреклапајуће регионе R_i , где је $i = 1, 2, \dots, n$, где је n број непразних скупова. Ако је $P(R_i)$ неки индикатор униформности свих елемената у скупу R_i , онда важи:

1. $\bigcup_{i=1}^n R_i = R$
2. за свако i и j , за које важи $i \neq j$, вреди $R_i \cap R_j = \emptyset$
3. за свако $i = 1, 2, \dots, n$, мора да важи $P(R_i) = 1$

² Treashold, праг – *thresholding* је дакле издајање на основу одређеног прага

4. за све парове $i \neq j$, важи $P(R_i \cup R_j) = 0$
5. за свако $i = 1, 2, \dots, n$, R_i је повезан регион³ [1]

Дакле, претходна формална дефиниција каже да скуп свих региона мора да буде комплетна слика (услов 1), да не сме бити преклапања између региона, тј. да један пиксел може да припада само једном региону (услов 2), затим да регион треба да буде униформан према некој карактеристици и да се према тој карактеристици значајно разликује од осталих региона (услови 3 и 4) и да сви региони треба да буду повезани (нрп. то је неопходно приликом издвајања више објеката са сличним или истим карактеристикама, али није обавезно).

Регионе или сегменте је најлакше схватити и интерпретирати приликом сегментације мирних сивих (grayscale) слика, где сегменти могу бити области приближно исте светлине на слици, који задовољавају ставке из претходне формалне дефиниције. Непомичне сиве слике се обично представљају као функција координата, односно светлина (ниво сивила) се представља као функција координата $f(x,y)$. Проширење 2D на 3D слике се дефинише пресликавањем $f(x,y) \rightarrow f(x,y,z)$, док се проширење мирних на покретне слике или низове мирних слика дефинише пресекавањем $f(x,y) \rightarrow f(x,y,t)$, док је проширење 2D на покретне 3D слике дефинисано пресликавањем $f(x,y) \rightarrow f(x,y,z,t)$. Уобичајен случај представља проширење мирних сивих слика на мирне колор слике, које се представља пресликавањем $f(x,y) \rightarrow f(x,y)$. Најопштије, можемо да дефинишемо проширење мирних сивих слика на мултидимензионалне покретне колор слике или низове мирних колор слика, помоћу следећег пресликавања $f(x,y) \rightarrow f(x,y,z,t)$. С обзиром на дефиниције проширења слика, потребно је дефинисати и одговарајуће проширење дефиниције сегментације слика. Приликом дефинисања претходних пресликавања, региони, односно сегменти и сви услови треба да се прошире на вишедимензионалне блобове⁴ (blobs), тако да се поменутих пет услова могу, и у овако уопштеном случају, користити за дефинисање сегментације слика. Приликом проширења дефинисаног пресликавањем $f(x,y) \rightarrow f(x,y,z)$, пиксел се замењује вокселем, тј. основним запреминским елементом, док за пресликавања на више од три димензије, основни елемент није дефинисан. Приликом проширења која се дефинишу пресликавањима која укључују и временску компоненту, постоје две могућности сегментације, у просторном и у временском домену, па се и проширења дефиниције треба извршити у складу са тим. Приликом проширења дефинисаног пресекавањем $f(x,y) \rightarrow f(x,y,z,t)$, карактеристике елемената слике постају вектори и у зависности од дефинисаног пресекавања говоримо о простору обележја или неком другом вишедимензионалном простору.

На основу претходног разматрања, можемо да закључимо да је математички строга дефиниција сегментације неопходна за најпростији случај, мирних сивих слика, док се за све остале случајеве дефинишу неопходна проширења. Из тог разлога приликом извођења било којих формулација за сегментацију, под појмом слика можемо да подразумевамо најопштије случајеве мирних слика, покретних слика, колор и сивих слика, итд.

³ Овај услов није неопходан и понекад се изоставља.

⁴ Израз који се у енглеског литератури користи да означи неки скуп вишедимензионалних вектора, кластер или сегмент.

1.3. Нивои истраживања сегментације слика

Истраживање у области сегментације слика почело је са применом првих техника за сегментацију слика средином шездесетих година прошлог века (Робертсов градијентни алгоритам за детекцију ивица), међутим, и даље не постоји уопштена теорија која би објединила све методе и технике за сегментацију. С обзиром на то у литератури постоје разни правци и приступи проблему сегментације слика, може се приметити да нити један алгоритам за сегментацију не даје задовољавајуће резултате за све врсте и модалитете дигиталних слика, нити су алгоритми на једнак начин примењиви на разне апликације (нпр. издвајање ивица, издвајање објеката од интереса, филтрирање слика, дигитална обрада слика, итд.).

Стога, постоји потреба да се строжије дефинишу критеријуми за међусобно поређење различитих алгоритама за сегментацију, јер са повећањем броја алгоритама за сегментацију, процена њихових карактеристика постаје све више значајна. Технике за процену алгоритама за сегментацију могу да се поделе у два основна типа, технике за карактеризацију и технике за компарацију. Оба типа техника наглашавају процену ефикасности алгоритма за сегментацију, а не развој самог алгоритма. Даље, с обзиром на све већи број метода за процену алгоритама, јавља се потреба за карактеризацијом различитих метода.

Дакле, истраживање у области сегментације слика можемо да посматрамо на три основна нивоа. Први и најважнији је ниво реализације алгоритама за сегментацију, средњи ниво подразумева развој метода за процену алгоритама, док највиши ниво подразумева систематско истраживање метода за процену алгоритама [1]. Други и трећи ниво истраживања су од великог значаја приликом општег разматрања проблема сегментације слика, па ће се овај рад углавном односити на реализацију алгоритма и оцену његових карактеристика.

1.4. Организација рада

У раду је описан начин примене mean shift алгоритма приликом сегментације слика, описане су његово основне карактеристике, затим основне функције које можемо да извршимо помоћу њега и извршена је практична реализација филтрирања и сегментације слика. У уводу су дефинисан појам сегментације и одређено је њено место меду техникама дигиталне обраде слика. Такође, дефинисани су основни нивои истраживања сегментације слика. У другој глави, Математички модел mean shift алгоритма, извршено је математичко извођење алгоритма, дате је преглед и опис његових основних параметара, пропусног опсега и облика кернела, затим описана је адаптивна верзија алгоритма. Такође, наведене су карактеристике и могућности примене алгоритма приликом анализе простора обележја. У трећој глави, Робусна анализа простора обележја, детаљно су описане основне функције које могу да се реализују применом mean shift алгоритма и дати су одговарајући псеудокодови. Четврта глава, Реализација филтрирања и сегментације, даје резултате извршавања филтрирања и сегментације слика према описаном алгоритму и дискусију око избора пропусних опсега на извршених функција. У закључку се врши сумирање добијених резултата и наводе се најновије методе које се ослањају на овај алгоритам. У додатку је дат код којим је извршена реализација филтрирања и сегментације.

2. МАТЕМАТИЧКИ МОДЕЛ MEAN SHIFT АЛГОРИТМА

Као што је већ наглашено у уводној глави, сваки метод за сегментацију слика треба да даје одговоре на одређена питања. Пре свега, треба да се дефинише каква тачно област на слици одређује један сегмент. Са семантичког становишта, можемо да кажемо да један сегмент треба да буде таква област на слици која је ограничена јасно израженим ивицама, без обзира на евентуално сенчење, тако да варијације боје или сивила буду минималне унутар области. Дакле, за правилно издвајање објеката на слици, потребно је применити такав алгоритам који ће сачувати информације о израженим ивицама, а ипак, извршити глачање односно смутинг (smoothing). Касније ће бити доказано да mean shift алгоритам управо ради усредњавање на претходно описани начин, односно да врши глачање са очувањем ивица⁵.

Математички модел mean shift алгоритма је релативно једноставан, јер представља непараметарску естимацију градијента функције густине вероватноће, што омогућава проналажење модела функције густине вероватноће. Под појмом мод подразумевамо моделе, односно изражене врхове функције густине вероватноће. Овај модел се лако може имплементирати на слике, где се подразумева да нпр. вектори у простору обележја одговарају узорцима одговарајуће функције густине вероватноће, па региони са већом густином вектора обележја представљају максимуме функције густине вероватноће. Одговор на треће питање о коректности сегментације се може најлакше добити уочавањем сегмената на слици који одговарају семантички смисленим целинама.

⁵ Discontinuity Preserving Smoothing – глачање са очувањем ивица.

2.1. Mean shift алгоритам и анализа простора обележја

Mean shift алгоритам су оригинално предложили Fukunaga и Hostetler 1975. године [1], [2] за детекцију модова функције густине вероватноће. Међутим, алгоритам остаје заборављен све до објављивања радова о његовој могућој примени у процесима сегментације слика и уопште робусне анализе простора обележја, Cheng 1995. [11] и Meer, Comaniciu 1999. [12] Због својих добрих карактеристика, једноставности и пре свега могућностима примене у разним апликацијама, mean shift алгоритам је брзо привукао пажњу научника и инжењера из области дигиталне обраде слике. С обзиром да је показао добре резултате приликом обраде и широке могућности примене и даље се ради на усавршавању овог алгоритма. Примена mean shift алгоритма може да се посматра као ефикасно решење проблема компјутерског вида (computer vision) ниског нивоа (low-level) у најопштијем случају. Наиме, многе методе које се користе за решавање поменутих проблема, нпр. за естимацију функције густине вероватноће су често неефикасне, јер су засноване на претпоставкама које углавном нису задовољене у практичним задацима. У области дигиталне обраде слике овакву врсту проблема можемо да нађемо при анализи простора обележја. Простор обележја добијамо пресликавањем или мапирањем улазних података, при чему се за сваки пиксел, обележје (карактеристика) од интереса пресликава у вишедимензионални вектор у простору тог обележја [2]. Нпр. код обраде слика, вишедимензионални простор обележја можемо да замислимо као скуп вектора који за сваки пиксел представља репрезентацију тог пиксела у неком колор домену. Након што се комплетан улазни податак (у овом случају слика) преслика у одређени колор простор, места са највећом густином представљају модове и циљ анализе је да се раздвоје вектори обележја који припадају различитим модovima у засебне кластере. Наведени пример анализе простора обележја где се пресликавање врши за сваки узорак (пиксел) је један од могућих, а постоје и друга пресликавања приликом којих се користе само одабране тачке према случајном закону. У овом раду ће се разматрати само први пример, који се практично често употребљава.

Мотивација за анализу у простору обележја је очигледна, јер се иначе приликом решавања математичких проблема, проблеми које је немогуће или тешко решити у једном простору пресликавају у други простор где је њихова интерпретација много једноставнија. Највећа предност анализе у простору обележја је то што се све карактеристике пресликавају уз велику толеранцију на сметње, а такође, предност је то што је анализа у простору обележја независна од апликације. То значи да овим алгоритмом можемо да извршавамо различите функције, нпр. филтрирање, детекцију ивица, итд.. Мана је то што је могуће извршити такво пресликавање које ће појединим карактеристикама дати мањи значај, па се оне могу једноставно изгубити при анализи. Међутим, овај недостатак се може избећи, што се често ради нпр. додавањем просторних параметара (у нашем случају координата пиксела на слици) [2]. У пракси су развијене и у литератури описане разне методе за анализу простора обележја које подразумевају а priori знање о броју кластера или расподели густине узорака, какав је нпр. k-means метод за кластеризацију. У литератури се овакве методе називају параметарским и сама чињеница да се параметри задају без довољно неопходног предзнања о карактеристикама простора обележја, представља огроман недостатак. Нпр. у поменутом k-means методу за кластеризацију k је улазни параметар који представља број претпостављених модова, који никако не мора да одговара стварном броју модова. Такође, у неким верзијама алгоритма неопходно је да буде задовољена претпоставка о Гаусовој расподели вектора обележја. Можемо да закључимо да се за анализу простора обележја структурираних по случајном закону морају користити

непараметарске методе које немају претпоставке о структури простора. У литератури су предложене бројне непараметарске методе за кластеризацију, које се могу поделити у две групе, хијерархијска кластеризација и методе засноване на естимацији функције густине. Приликом кластеризације помоћу метода заснованих на естимацији функције густине, једна од основних предности је та што простор обележја може третирати као функција густине вероватноће, па региони са највећом густином вектора обележја одговарају модovima функције густине вероватноће. Корелација је очигледна, налажење модова или врхова функције густине вероватноће у општем случају одговара локацији региона са највећом густином вектора обележја. Груписањем оваквих региона односно вектора обележја који припадају тим регионима у засебне кластере врши непараметарска кластеризација простора обележја, која детектује број модова без претходне претпоставке о њиховом броју. Једна од таквих непараметарских метода заснована на mean shift алгоритму, даје поуздане резултате за велики опсег апликација. Као што је већ напоменуто, овај алгоритам припада групи алгоритама заснованих на естимацији функције густине помоћу прозора, тј. кернела. У литератури је овај поступак познат и као метода Парзенових прозора [3].

2.1.1. Естимација функције густине вероватноће методом Парзенових прозора

Један од основних закона у математичкој вероватноћи каже да је вероватноћа да се неки вектор \mathbf{x} нађе унутар региона R дата интегралом функције густине вероватноће $p(\mathbf{x})$ на том региону R , што се исказује следећом једначином:

$$P = \int_R p(\mathbf{x}) d\mathbf{x} \quad (2.1)$$

Ова једначина, такође, представља основу за све методе за естимацију непознате функције густине вероватноће. Наиме, вероватноћа P , представља *поравнату* или усредњену вредност функције густине на задатом региону R , па тако естимацијом вероватноће P , такође, вршимо естимацију и функцију густине вероватноће $p(\mathbf{x})$. Претпоставка да имамо случај од n узорака $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, најбоље одговара нашем примеру анализе простора обележја пресликаних са слика, јер се ради о коначном броју пиксела. Вероватноћа да се унутар региона R нађе k узорака од укупних n , одговара биномној расподели:

$$P_k = \binom{n}{k} P^k (1-P)^{n-k} \quad (2.2)$$

док је математичко очекивање вредности броја k узорака унутар региона:

$$E(k) = nP \quad (2.3)$$

За довољно велики број узорака n , однос k/n даје добру естимацију вероватноће P , па самим тим и функције густине вероватноће $p(\mathbf{x})$. Ако је функција густине вероватноће континуална и ако је регион R довољно мали⁶ да се може рећи да је функција густине вероватноће приближно константна, тада је:

⁶ Регион треба да буде и довољно велики да увек садржи довољно узорака за математички коректну естимацију.

$$\int_R p(\mathbf{x}) d\mathbf{x} = p(\mathbf{x})V \quad (2.4)$$

где је са V означена запремина посматраног региона R . Делењем леве и десне стране са запремином региона, претходну једнакост можемо да искористимо да проценимо нормализовану, односно просторно усредњену вредност вероватноће P , па наравно и функције густине вероватноће $p(\mathbf{x})$. С обзиром да ни овде нисмо увели ограничења, тј. да је теорија општа, простор можемо да изаберемо произвољно, односно може да се ради о било којем колор простору, или неком другом простору обележја. Комбиновањем претходних једначина добијамо израз за нормализовану усредњену вредност вероватноће P :

$$\frac{P}{V} = \frac{\int_R p(\mathbf{x}) d\mathbf{x}}{\int_R d\mathbf{x}} \quad (2.5)$$

Да бисмо извршили естимацију функције густине вероватноће, извршићемо анализу према следећој процедури. Формирамо низ региона R_1, R_2, \dots и нека је V_n запремина n -тог региона, а k_n број узорака унутар n -ог региона, који се према неком закону повећава са редом естимата. n -ти естимат функције густине вероватноће, на основу претходних једначина добијамо у облику:

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n} \quad (2.6)$$

Да би естимација конвергирала стварној вредности функције, потребно је да буду испуњени неки услови. Пре свега, запремина V_n треба да се приближава нули када ред естимата тежи бесконачности, чиме се осигурава конвергенција ка стварној вредности функције. Даље, број узорака k_n треба да тежи бесконачности када $n \rightarrow \infty$ да би се осигурала конвергенција односа k_n/n према P . Трећи услов, неопходан за конвергенцију претходне једначине, каже да однос k_n/n треба да тежи нули када $n \rightarrow \infty$ [3]. Претходни услови су важни само са аспекта строгог математичког доказа конвергенције, па се о њима неће детаљно расправљати.

Постоји више начина се одреди низ региона који ће да задовољи претходне услове, а једна од њих представља Парзенов метод прозора за естимацију функције. Основна идеја методе је да се запремина иницијалног региона смањује према некој функцији од n , па се мора доказати и конвергенција, односно поменута функција се мора изабрати тако да претходна три услова буду задовољена. Да бисмо избегли егзактан математички доказ, можемо да усвојимо да за конвергенцију треба да се усвоји однос $V_n \sim n^{-1/2}$ или нека још спорија конвергенција запремине ка нули [3].

Основне карактеристике методе Парзенових прозора можемо једноставно да изведемо из следећег примера. Претпоставимо да имамо d -димезионални простор, и да је R_n хиперсфера чији је радијус по свим димензијама h_n , тада можемо да проценимо број узорака k_n који се налазе унутар хиперсфере дефинисањем следеће јединичне прозорске функције:

$$f(\mathbf{u}) = \begin{cases} 1, & |u_j| \leq 1/2, \quad j = 1, 2, \dots, d \\ 0, & \text{inace} \end{cases} \quad (2.7)$$

Пошто се дефинисана прозорска функција користи за естимацију функције густине вероватноће, потребно је да има исти карактеристике као и та функција, тј. треба да буде ненегативна и интеграбилна на јединицу:

$$f(\mathbf{u}) \geq 0, \quad \int f(\mathbf{u}) d\mathbf{u} = 1 \quad (2.8)$$

Очигледно је да је број узорака, по једној димензији, који се налази унутар хиперсфере чија је запремина h_n^d , исти као и број узорака који се налази унутар опсега према тој димензији $f((\mathbf{x}-\mathbf{x}_i)/h_n)$, па је број узорака унутар хиперсфере једнак:

$$k_n = \sum_{i=1}^n f\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) \quad (2.9)$$

Комбинацијом једначина (2.6) и (2.9) добијамо израз за естимат функције густине вероватноће на месту \mathbf{x} :

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_n} f\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) \quad (2.10)$$

У претходном случају изабрали смо јединичну прозорску функцију која свим узорцима који се налазе унутар хиперсфере даје једнак значај. Међутим, с обзиром да нисмо увели никаква ограничења, прозорску функцију можемо да дефинишемо и као тежинску, тако да ће сваки узорак који се налази унутар хиперсфере дати другачији допринос естимацији с обзиром на удаљеност од центра хиперсфере.

Из претходног излагања је јасно какав утицај ће имати пропусни опсег, односно радијус h_n . Ако је пропусни опсег превелики поравнање ће да се изврши на већем опсегу и могу се изгубити информације, а са друге стране ако је пропусни опсег врло мали, естимација ће да задржи велике варијације.

2.1.2. Mean shift алгоритам

Mean shift алгоритам је заснован на естимацији градијента функције густине вероватноће и самим тим чини један посебан случај претходно описане методе Парзенових прозора. Сада ћемо посматрати вишедимензионалну прозорску функцију односно кернел K_H која, у општем случају, зависи од матрице пропусног опсега \mathbf{H} . Дакле, ради се о уопштеној хиперсфери која може имати различите пропусне опсеге за сваку димензију. Естимација функције густине вероватноће у овом уопштеном примеру, има следећи облик [2]:

$$\widehat{f}_k(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K_H(\mathbf{x}-\mathbf{x}_i) \quad (2.11)$$

која написана другачије, са нормализованим кернелом K , има облик [1]:

$$\widehat{f}_k(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{|\mathbf{H}|^{1/2}} K(d(\mathbf{x} - \mathbf{x}_i)) \quad (2.12)$$

У претходној једначини $d(\mathbf{x} - \mathbf{x}_i)$ означава дистанцу између два вектора, коју можемо да дефинишемо као нормализовану Еуклидову дистанцу у складу са дискусијом о методи Парзенових прозора:

$$d(\mathbf{x} - \mathbf{x}_i) = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{|\mathbf{H}|^{1/2}} \quad (2.13)$$

Комбинацијом једначина (2.12) и (2.13) и с обзиром на једначину (2.6) добијамо коначан израз за естимацију функције:

$$\widehat{f}_k(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{|\mathbf{H}|^{1/2}} K\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{|\mathbf{H}|^{1/2}}\right) \quad (2.14)$$

Наравно, кернел или прозор треба да задовољава карактеристике које смо дефинисали у претходном поглављу, тј. кернел треба да буде интеграбилан на јединицу и да обезбеди конвергенцију ка стварној вредности функције густине вероватноће $f(\mathbf{x})$. Такође, додаје се услов симетричности кернела, који олакашава математичку анализу и најчешће се користи у пракси. Кернел треба да задовољи следеће карактеристике [2]:

$$\text{интеграбилност на јединицу} \quad \int_{R^d} K(\mathbf{x}) d\mathbf{x} = 1 \quad (2.15)$$

$$\text{симетричност} \quad \int_{R^d} \mathbf{x}K(\mathbf{x}) d\mathbf{x} = 0 \quad (2.16)$$

$$\text{конвергенција} \quad \lim_{\|\mathbf{x}\| \rightarrow \infty} \|\mathbf{x}\|^d K(\mathbf{x}) = 0 \quad (2.17)$$

$$\int_{R^d} \mathbf{x}\mathbf{x}^T K(\mathbf{x}) d\mathbf{x} = c_k \mathbf{I} \quad (2.18)$$

Да бисмо дефинисали симетричан вишедимензионални кернел, довољно је да се дефинише кернел по једној димензији, па да се дефинише његово проширење. Вишедимензионални кернел се може генерисати помоћу једнодимензионалног на два начина, као производ једнодимензионалних кернела или ротацијом једнодимензионалног кернела око оса других димензија чиме се добија радијално симетричан кернел. Дакле, можемо да имамо следећа два случаја:

$$K_P(\mathbf{x}) = \prod_{i=1}^d K_1(x_i), \quad K_S(\mathbf{x}) = \alpha_d K_1(\|\mathbf{x}\|) \quad (2.19)$$

Вишедимензионални кернели генерисани на оба начина треба да задовољавају услове (2.15 - 2.18). Практично, чешће се користе радијално симетрични кернели и то њихов посебан облик који се може изразити следећом једначином:

$$K(\mathbf{x}) = c_k k(\|\mathbf{x}\|^2) \quad (2.20)$$

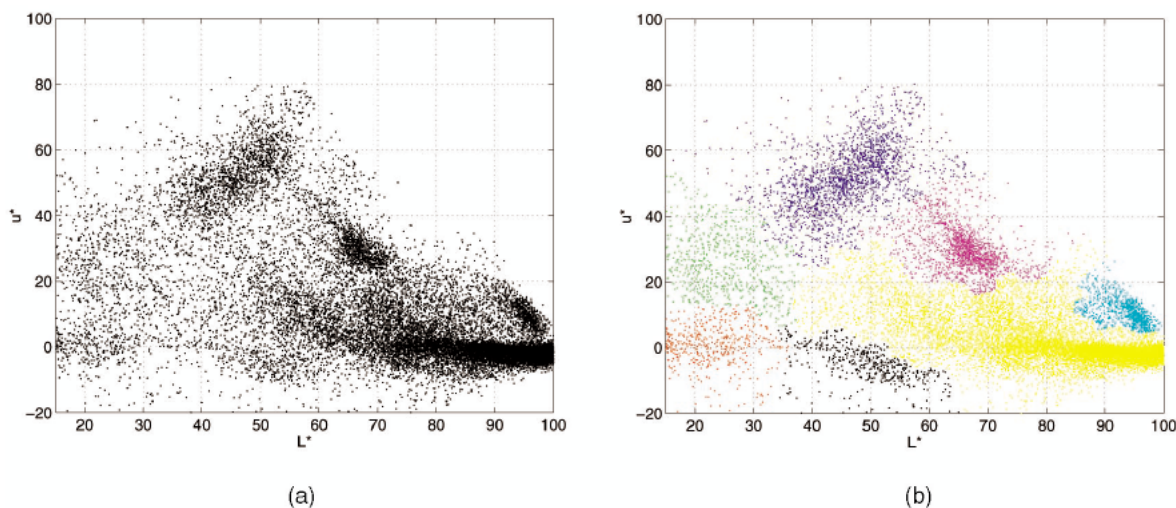
у ком случају је довољно да се дефинише функција $k(x)$ и то само за $x > 0$. Функција $k(x)$ се назива профил кернела. Нормализациона константа c_k треба да гарантује неопходан услов интеграбилности кернела на јединицу. Још је потребно да размотримо матрицу пропусног опсега и њен утицај на естимацију. Могуће је дефинисати различите пропусне опсеге за сваку димензију, али због једноставнијег рачуна и шире практичне примене користе се или само дијагоналне матрице $\mathbf{H} = \text{diag}(h_1^2, h_2^2, \dots, h_d^2)$, или дијагоналне матрице у облику $\mathbf{H} = h^2 \mathbf{I}$. Последњи случај је најједноставнији али и практично најкориснији. Уврштавањем израза за матрицу пропусног опсега у једначину (2.14) добијамо израз за естимацију функције у следећем облику [2]:

$$\hat{f}_k(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (2.21)$$

или када се уврсти израз (2.20), добија се [1]:

$$\hat{f}_k(\mathbf{x}) = \frac{c_k}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \quad (2.22)$$

Већ је споменуто да метода Парзенових прозора врши естимацију функције густине вероватноће, а да mean shift алгоритам ради на принципу естимације градијента функције густине вероватноће. На тај начин се лако проналазе модови или врхови где је функција густине максимална. Из аналогије која је показана на почетку ове главе, можемо да закључимо да детекцијом мода функције густине уствари вршимо локализацију региона са највећим бројем узорака унутар простора обележја. Такве регионе треба да издвојимо у посебне кластере, као што је приказано на слици 2.1. Видимо да алгоритам не захтева претходно дефинисање броја модова, већ их све сам налази, а такође, нема никакве претпоставке о структури простора обележја.



Слика 2.1: Кластеризација у простору обележја

Естимацију градијента функције густине добијамо деривацијом једначине (2.22), и то ако важи да је градијент естимације функције једнак естимацији градијента, што је испуњено због особине линеарности градијента [2]:

$$\widehat{\nabla} f_k(\mathbf{x}) \equiv \nabla \widehat{f}_k(\mathbf{x}) = \frac{2c_k}{nh^{d+2}} \sum_{i=1}^n (\mathbf{x} - \mathbf{x}_i) k' \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \quad (2.23)$$

Ако сада дефинишемо функцију (односно нови профил керенла) као $g(x) = -k'(x)$, под условом да је функција $k(x)$ диференцијабилна за свако $x \in [0, \infty)$ осим за коначан број тачака. Ако су испуњени ти услови функција $g(x)$ представља профил који дефинише нови кернел, за које важе сви поменути услови за кернеле, једначине (2.15 - 2.18). Можемо да напишемо следеће једначине:

$$G(\mathbf{x}) = c_g g \left(\|\mathbf{x}\|^2 \right) \quad (2.24)$$

$$\widehat{f}_g(\mathbf{x}) = \frac{c_g}{nh^d} \sum_{i=1}^n g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \quad (2.25)$$

Једначине (2.24) и (2.25) су потпуно аналогне и имају исто значење као и једначине (2.20) и (2.22), са једином разликом што имамо другачији кернел. Ако важи једнакост $g(x) = -k'(x)$ тада се кернел K назива сенком кернела G . У пракси се најчешће користе униформни, Гаусов кернел и кернел Епанечњикова, што ћемо продискутовати у следећем поглављу. Комбиновањем претходних једначина добијамо следећу зависност:

$$\widehat{\nabla} f_k(\mathbf{x}) = \frac{2c_k}{nh^{d+2}} \left[\sum_{i=1}^n g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^n \mathbf{x}_i g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^n g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right] \quad (2.26)$$

Израз у првој угластој загради, уз одговарајуће константе, представља естимацију функције густине вероватноће помоћу кернела G , као што је написано у једначини (2.25), док израз у другој угластој загради представља померај средње вредности (mean shift), односно разлику између средње вредности рачунате на основу узорака који се налазе унутар хиперсфере и центра хиперсфере:

$$\mathbf{m}_k(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}_i g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^n g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \quad (2.27)$$

Први члан у претходној једначини можемо да упоредимо са изразом (2.5) који даје нормализовану вредност вероватноће на посматраном прозору, дакле средњу вредност. Други члан је иницијални центар прозора или средња вредност из претходне итерације, па је њихова разлика померај средње вредности. Можемо да видимо да овај алгоритам

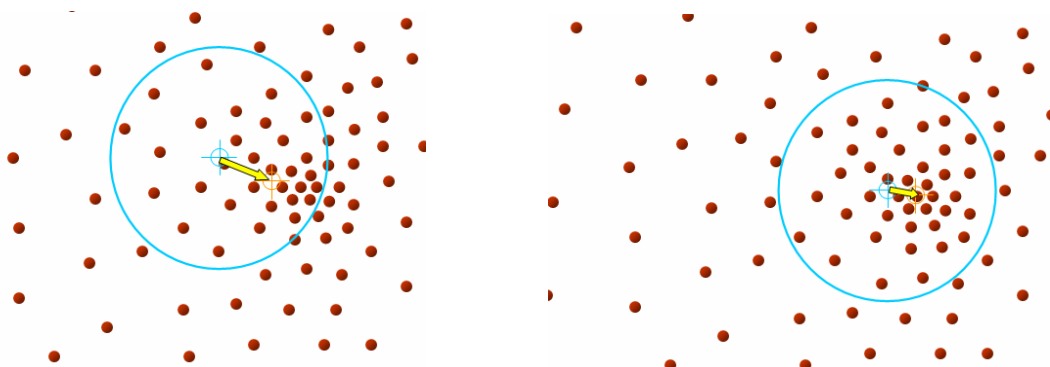
ради на принципу померања према најгушћем региону, и било би коректно доказати конвергенцију. Међутим, већ смо нагласили у којим случајевима метода Парзенових прозора увек конвергира, што наравно зависи од облика кернела. За поменуте кернеле који се најчешће користе конвергенција је гарантована [2]. Комбиновањем једначина (2.23), (2.26) и (2.27), добијамо следећи израз [1], [2], [5]:

$$\widehat{\nabla} f_k(\mathbf{x}) = \frac{2c_k}{h^2 c_g} \widehat{f}_g(\mathbf{x}) \cdot \mathbf{m}_k(\mathbf{x}) \quad (2.28)$$

коју можемо да напишемо и у следећем, једноставнијем, облику [5]:

$$\mathbf{m}_k(\mathbf{x}) = \frac{h^2 c}{2} \frac{\widehat{\nabla} f_k(\mathbf{x})}{\widehat{f}_g(\mathbf{x})} \quad (2.29)$$

Видимо да је вектор помераја средње вредности еквивалентан односу естимације градијента функције са кернелом K и естимације функције са кернелом G . Другим речима, вектор помераја средње вредности је еквивалентан нормализованој естимацији градијента функције са кернелом K при чему је нормализација извршена естимацијом функције кернелом G . Можемо да закључимо да је mean shift алгоритам заснован искључиво на естимацији градијента и да је на основу једначине (2.27) његова практична реализација релативно једноставна.



Слика 2.2: Проналажење региона са највећом густином

Суштина алгоритма је приказана на слици 2.2 где је приказан упроштени једнодимензионалан случај. Изабере се иницијална полазна тачка и посматрају се сви узорци који се налазе у дефинисаној хиперсфери са центром у тој тачки. На слици иницијалана полазна тачка је означена плавом бојом, као и круг који у овом 1D случају чини пропусни опсег или прозор. У следећем кораку се рачуна средња вредност на основу података који се налазе унутар прозора, која ће да се налази у оном делу хиперсфере у којој је већа концентрација узорака, односно већа густина. То можемо да видимо на слици, где је нова средња вредност означена црвеном бојом. У следећој итерацији центар хиперсфере се помера у тачку одређену с израчунатом средњом вредношћу из предходне итерације. Вектор помераја средње вредности је означен жутом стрелицом. Тада се нова средња вредност проглашава за центар прозора и поступак се понавља. Дакле, поново се посматрају само узорци који се налазе унутар хиперсфере, али овога пута са новим центром. Средња вредност ће поново да се налази у оном делу хиперсфере где је густина узорака највећа и што се центар сфере приближава региону са највећом густином помераји су све мањи. Алгоритам се

зауставља онда када померај буде мањи од неког предефинисаног прага, односно врло мали или теоретски једнак нули. То може да се потврди и из једначине (2.29) односно чињенице да је померај средње вредности еквивалентан нормализованој естимацији градијента. Када је градијент једнак нули знамо да смо дошли на место највеће локалне густине узорака, односно да смо детектовали мод. Тада се узорци који се налазе недалеко од трајекторије којом центар хиперсфере долази до максимума функције обележавају и издвајају као једна кернел. Поступак кластеризације се наставља све док постоје узорци који нису придружени нити једном кластеру.

2.1.3. Избор кернела

Већ је наведено да је за успешну сегментацију слика потребно да се филтрирају мале варијације на релативно униформним областима слике, а да се ивице сачувају током тог процеса, да се не би нарушила семантичка структура слике. Да би се постигао овакав ефекат, врло је битан избор кернела или прозора приликом претпроцесирања (филтрирања) слике. У пракси се најчешће користе два типа кернела, Гаусов кернел и кернел Епанечњикова, који је изведен према критеријуму минимизације средње интегрисане квадратне грешке (minimum mean integrated square error, MISE) [1],[2],[6]. Профил Гаусовог кернела је:

$$k_g(x) = e^{-x/2} \quad (2.30)$$

из којег се добија радијално симетричан кернел облика [1],[2]:

$$K_g(\mathbf{x}) = \frac{1}{(2\pi h^2)^{d/2}} e^{-\|\mathbf{x}\|^2/2h^2} \quad (2.31)$$

Очигледно је да је сенка Гаусовог кернела, такође, Гаусов кернел. У пракси се често, ради лакшег рачуна Гаусов кернел ограничи на пропусни опсег, чиме се одбацују изразито мале вредности проузроковане експоненцијалном функцијом, што је допустиво јер је грешка занемарива. Профил кернела Епанечњикова је облика:

$$k_e(x) = \begin{cases} 1-x, & |x| \leq 1 \\ 0, & |x| > 1 \end{cases} \quad (2.32)$$

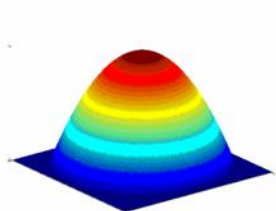
Овакав профил генерише радијално симетричан кернел у облику:

$$K_e(\mathbf{x}) = \begin{cases} \frac{d+2}{2c} (1-\|\mathbf{x}\|^2), & \|\mathbf{x}\| \leq 1 \\ 0, & \text{in case} \end{cases} \quad (2.33)$$

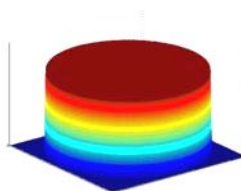
Кернел Епанечњикова је сенка униформном кернелу, који узима све узорке унутар пропусног опсега са једнаким значајем. С обиром на профил униформног кернела:

$$k_u(x) = \begin{cases} 1, & |x| \leq 1 \\ 0, & |x| > 1 \end{cases} \quad (2.34)$$

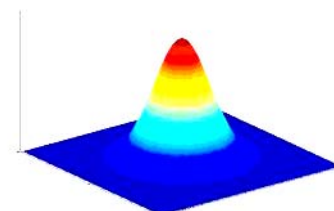
може да се закључи да је употребом кернела Епанечњикова, поред тога што се минимизира грешка, рачун поприлично упроштен. Генерисање вишедимензионалних кернела помоћу набројаних профила, методом којом се добијају радијално симетрични кернели представља најједноставнији, али не и једини случај. Наиме, јасно је да радијално симетрични кернели одговарају изотропним кернелима који имају једнаке пропусне опсеге по свим димензијама. У том случају је потребно задати један пропусни опсег, односно један радијус хиперсфере.



кернел Епанечњикова



униформни кернел



Гаусов кернел

Постоје још два случаја која се често користе у пракси, то су анизотропни кернел и кернел са адаптивним пропусним опсегом. Анизотропни кернел има различите пропусне опсеге по различитим димензијама. Са друге стране, показало се да је за структуру видеа приликом сегментације оптималније користити кернел који прилагођава свој облик, величину и оријентацију структури података, па су развијени адаптивни анизотропни кластери [4]. Наиме, често се дешава да нпр. приликом сегментација видеа или праћења објеката неприлагођеност кернела облику или локалној структури простора обележја узрокују временску некохерентност између узастопних фрејмова, што узрокује лошије перформансе алгоритма. Такође, јасно је да се за специјалне облике простора обележја и у случају мирних слика добијају бољи резултати када се употребљава адаптивни кернел.

Дакле, адаптивни кернел је анизотропни кернел који се прилагођава локалној структури простора обележја и стога пружа додатне апликационе могућности за сегментацију. Нпр. алгоритам се може прилагодити да за мирне позадинске регионе врши сегментацију са мање уочљивих детаља док се у региону од интереса могу сачувати и најситнији детаљи [4].

2.1.4. Избор пропусног опсега

Пропусни опсег или нормализационе константе често представљају највећи проблем приликом сегментације, јер генерално не постоји метод који би одредио задовољавајуће пропусне опсеге за шири опсег слика. Такође, приликом дискусије о естимацији функције густине вероватноће методом Парзенових прозора, већ су поменуте последице лошег избора пропусног опсега, тј. избора превеликог или премалог пропусног опсега. Са друге стране поменуто је да кернел Епанечњикова минимизира средњу интегрисану квадратну грешку, међутим, потребно је нагласити да је анализа изведена под претпоставком да је простор обележја нормалне структуре, што никако не можемо да потврдимо у практичним случајевима [6]. Стога, оптималне пропусне опсеге је потребно одређивати на основу структуре података за сваки случај посебно [5]. Ипак, та чињеница не ограничава примену mean shift алгоритма, јер је његова конвергенција и даље гарантована, а са друге стране оптимални пропусни опсег се лако може одредити експериментално. Увођење адаптивног пропусног опсега узрокује и увођење променљивог кернела, јер ако се пропусни опсег мења по некој димензији, сасвим је логично и да ће се кернел мењати у зависности од тог пропусног

опсега. Кернели са промењивим пропусним опсегом дају добре резултате у случајевима вишедимензионалних улазних података, што и јесте случај приликом обраде слика [6].

Постоје два приступа како може да се дефинише промењиви пропусни опсег. У првом случају избором различитог пропусног опсега $h = h(\mathbf{x})$, за сваку тачку естимације \mathbf{x} чиме се добија тзв. балонски естиматор функције густине (balloon density estimator), при чему је естимат функције у тачки \mathbf{x} једнак просечној вредности једнако скалираних кернела свакој тачки. Овакав приступ није дао побољшање у односу на фиксни пропусни опсег нити задовољава интегралност кернела на јединицу, па је стога одбачен.

У другом приступу бира се промењиви пропусни опсег за сваки вектор обележја \mathbf{x}_i и то као $h = h(\mathbf{x}_i)$, чиме се добија тзв. естиматор густине узорака (sample point density estimator). У овом случају је естимат функције густине вероватноће у тачки \mathbf{x} једнак просечној вредности различито скалираних кернела у свакој тачки. Овакав приступ задовољава све потребне услове кернела и показало се да има побољшане карактеристике у односу на случај фиксног пропусног опсега [6]. Исти аутори су предложили следећу зависност промењивог пропусног опсега:

$$h(\mathbf{x}_i) = h_0 \sqrt{\frac{\lambda}{f(\mathbf{x}_i)}} \quad (2.35)$$

У овом случају је потребно да се на основу структуре података одреде почетна вредност пропусног опсега h_0 и естимат функције густине $f(\mathbf{x}_i)$ у датој тачки, која се назива пилот функција. Када се примени адаптивни пропусни опсег, на основу једначине (2.21) добијамо естимат функције према адаптивном mean shift алгоритму:

$$\widehat{g}_k(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h(\mathbf{x}_i)^d} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h(\mathbf{x}_i)}\right) \quad (2.36)$$

С обзиром на једначине (2.35) и (2.36) можемо да уочимо на који начин структура простора обележја утиче на естимацију функције. С обзиром на обрнуту пропорционалност, узорак који одговара малој вредности функције, се узима са малом тежинском вредношћу, док се узорци који одговарају великој вредности функције узимају са великом тежинском вредношћу. На тај начин се облик кернела прилагођава структури простора обележја.

Анализу настављамо као и у случају фиксног пропусног опсега, с том разликом што се уместо фиксног пропусног опсега h уводи промењиви пропусни опсег h_i за сваки узорак. Знамо да је естимација градијента једнака градијенту естимације функције:

$$\widehat{\nabla} g_k(\mathbf{x}) \equiv \nabla \widehat{g}_k(\mathbf{x}) = \frac{2c_k}{n} \sum_{i=1}^n \frac{(\mathbf{x} - \mathbf{x}_i)}{h_i^{d+2}} k\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h_i}\right\|^2\right) \quad (2.37)$$

и у овом случају, сређивањем добијамо адаптивни померај средње вредности:

$$\mathbf{m}_g(\mathbf{x}) = \frac{\sum_{i=1}^n \frac{\mathbf{x}_i}{h_i^{d+2}} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h_i}\right\|^2\right)}{\sum_{i=1}^n \frac{1}{h_i^{d+2}} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h_i}\right\|^2\right)} - \mathbf{x} \quad (2.38)$$

У претходној једначини, разлика у односу на случај фиксног пропусног опсега јесте узимање у обзир различитог пропусног опсега за сваку тачку. Комбинацијом једначина (2.35), (2.36) и (2.38) добијамо коначан израз за померај средње вредности у адаптивном mean shift алгоритму:

$$\mathbf{m}_g(\mathbf{x}) = \frac{\lambda}{\frac{1}{n} \sum_{i=1}^n \tilde{f}(\mathbf{x}_i)} \frac{h_0^2 c_g \widehat{\nabla} f_k(\mathbf{x})}{2 \widehat{f}_g(\mathbf{x})} \quad (2.39)$$

Потребно је нагласити да се све константе у претходној једначини попуцарају са случајем фиксног пропусног опсега, осим нормализационе естимације функције густине са кернелом G , која има облик:

$$\widehat{f}_g(\mathbf{x}) = c_g \frac{\sum_{i=1}^n \frac{\tilde{f}(\mathbf{x}_i)}{h_i^d} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h_i}\right\|^2\right)}{\sum_{i=1}^n \tilde{f}(\mathbf{x}_i)} \quad (2.40)$$

Један од недостатака представља потреба за естимацијом пилот функције која даје вредност функције густине на месту тачке \mathbf{x}_i . Друга дискутабилна чињеница је зависност норме вектора помераја средње вредности од константе пропорционалности λ , па за велико λ може да се деси да алгоритам заобилази мод, док за мале вредности проналажење мода може дуго да траје. Такође, из једначине (2.39) уочавамо да ако константу пропорционалности изаберемо тако да буде једнака пилот функцији за сваку тачку, добијамо случај фиксног пропусног опсега. Можемо да закључимо, да је адаптивни mean shift алгоритам уопштени случај алгоритма са фиксим пропусним опсегом [6].

Други аутори [4] су предложили решење према којем се пропусни опсег у range домену мења као функција анизотропне матрице пропусног опсега у домену простора \mathbf{H}_i^s . Та матрица се процењује на основу стандардне дијагоналне матрице пропусног опсега и номиналне вредности пропусног опсега у range домену. Поређењем резултата дошло се до закључка да тако дефинисан анизотропни кернел много боље сегментира слике, нарочито са издуженим униформним областима.

2.2. Карактеристике mean shift алгоритма

Иако је већ поменуто доста тога што може да се нађе у овом поглављу, овде ће се на једном месту набројати све предности и мане овог алгоритма. Пре свега, mean shift је непараметарски робусни алгоритам за кластеризацију који ради на принципу естимације градијента функције густине вероватноће. Дакле, овај алгоритам можемо да

користимо за анализу простора обележја, јер не претпоставља ништа око структуре тог простора и налази све модове без обзира на то колико их има. У питању је итеративни алгоритам који свој померај прилагођава у односу на стрмину функције, па се може рећи да се ради о модификованом градијентном процесу при чему је предност то што се корак аутоматски модификује на основу структуре података. Приликом наилаaska на регионе са великом густином, корак се смањује и на тај начин омогућава лакше проналажење мода. Даље, конвергенција се гарантује за све случајеве који су важни за практичну примену, с обзиром на наведене карактеристике методе Парзенових прозора на коју се овај алгоритам ослања. Такође, предност је и то што не подразумева облик кластера, већ се и он прилагођава структури простора обележја. Врло битно је и то што је алгоритам независан од апликације, тј. може да се користи за све могуће апликације које се срећу у дигиталној обради слика, нпр. за издајање ивица, издвајање региона, сегментацију, праћење објеката, обраду слике (убацивање друге позадине или неког другог објекта), филтрирање, итд. Предност је и то што је алгоритам погодан за обраду података у реалном времену, па је с обзиром на све то нашао веома широку примену у области рачунског вида. У самом алгоритму, потребно је задати само један параметар, то је ширина пропусног опсега, али у адаптивној верзији система могуће је проценити оптималан пропусни опсег за конкретну структуру простора обележја. И на крају важно је поменути да је алгоритам робустан, што значи да је отпоран на евентуалне уљезе или другим речима да на конвергенцију неће утицати евентуални узорци са изразито другачијим карактеристикама.

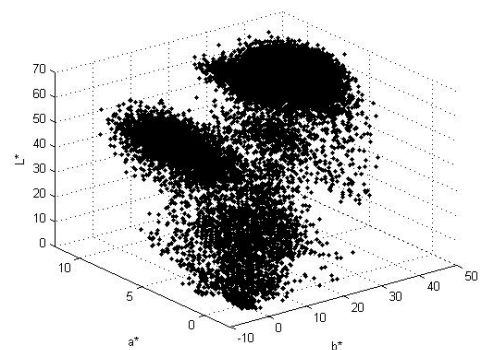
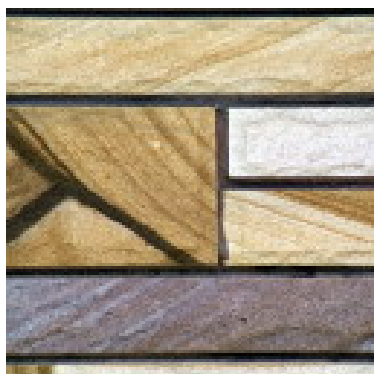
Једина мана овог алгоритма је немогућност генералног одређивања пропусног опсега за шири спектар улазних сигнала, тј. у нашем случају више различитих врста слика. Овај недостатак се може уклонити увођењем адаптивног пропусног опсега, међутим, тада се уводи проблем са естимацијом пилот функције. На основу претходних карактеристика може да се закључи да mean shift алгоритам представља један веома користан начин за дигиталну обраду слике. У следећој табели су дате основне карактеристике алгоритма:

Табела 2.1: Карактеристике алгоритма

<i>предности</i>	<i>недостаци</i>
<ul style="list-style-type: none"> ○ непараметарски алгоритам ○ не зависи од апликације ○ нема а priori услова око облика кернела ○ може да анализира случајно структуриране просторе ○ робустан алгоритам ○ може да ради у реалном времену ○ задаје се само један параметар који има физичко значење 	<ul style="list-style-type: none"> ○ проблематично одређивање оптималног пропусног опсега ○ лоше изабран пропусни опсег може да проузрокује појаву непостојећих модова или да споји два мода

3. РОБУСНА АНАЛИЗА ПРОСТОРА ОБЕЛЕЖЈА

Основне карактеристике простора обележја су вишедимензионалност и структура која је случајног карактера [2]. Овакви простори могу да се анализирају само помоћу непараметарских алгоритама који немају никакве а priori претпоставке о структури поља. Mean shift јесте алгоритам који задовољава те услове, међутим, главни разлог због кога се он наметнуо јесте његова примењивост независно од апликације. У литератури је описан велики број могућих примена, налажење модела, филтрирање, адаптивно филтрирање, сегментација мирних слика, сегментација покретних слика или видеа, праћење објеката на видеу [21], [27], [28], итд. Нарочито је практичну примену нашла апликација за праћење објеката у системима за надзор заснована на mean shift алгоритму [20], [27]. Веома важна дискусија око простора обележја јесте избор метрике којом ће се одређивати удаљеност између два вектора. Познато је да RGB колор простор није најпогоднији за примену Еуклидове метрике, али бројни примери из шире литературе о дигиталној обради слике наводе како се при анализи RGB колор простора може користити Еуклидова метрика као код линеарних $L^*u^*v^*$ и $L^*a^*b^*$ колор простора [31], Слика 3.1.



Слика 3.1: Представљање слике у $L^*a^*b^*$ простору

Други приступ је да се примени друга метрика у RGB простору, нпр. Махаланобисова метрика [2] која би са друге стране захтевала посебно дефинисање матрице пропусног опсега. Међутим, како се обично настоји минимизирати број улазних параметара остало се при Еуклидовој метрици и једном параметру за дефинисање пропусног опсега. Простор обележја се може анализирати тако што ће свака тачка (врх вектора) да се процесуира и да се за сваку тачку тражи тачка конвергенције, или у другом приступу да се примени тзв. basin of attraction метод, према коме се тачке које су близу трајекторије, којом се из неке тачке путује до њеног мода, аутоматски приписују том истом моду, под претпоставком да би и оне конвергирале према истом моду.

Проблем може да представља неадекватан избор пропусног опсега, јер ако је он превише мали добићемо непостојеће модове односно локалне максимуме, који се могу елиминисати постпроцесирањем и то спајањем свих кластера чији су модови, тј. тачке конвергенције, на мањој удаљености од ширине пропусног опсега. Примена овог принципа могла би да изазове проблем ако се нека тачка налази близу граници па самим тим буде посећена више пута. И то може да се реши на више начина, нпр. применом принципа првог посетиоца коме се приписује тачка, док се накнадне посете занемарују. Следећи важан детаљ који треба да се размотри пре анализе простора обележја јесте облик кернела. И у овом случају се применом mean shift алгоритма одлази корак даље у компромису између жељеног и потребног, јер се у пракси показало да најједноставнији случај, употреба униформног профила, даје најбоље резултате.

3.1. Дефинисање простора обележја

У практичној примени приликом обраде слика, слике се представљају конкатенацијом матрица просторне расподеле тј. места сваког пиксела на слици и вектора боја за сваки пиксел на слици. У случају сивих слика имамо матрицу димензија $m \times n \times 1$, док у осталим случајевима имамо матрице димензија $m \times n \times p$, где је $p=3$ за колор слике или $p>3$ у другим вишедимензионалим случајевима, нпр. код покретних слика. Просторна расподела пиксела се у литератури на енглеском језику назива spatial (просторни) домен, док се колор простор назива range (област) домен⁷. Да би се извршила анализа слике неком непраметарском методом и успут сачувала информација о простором односу пиксела, потребно је формирати тзв. здружени spatial-range домен. За оба домена посебно подразумева се Еуклидова метрика, док се евентуалне разлике у вредностима превазилазе одговарајућом нормализацијом. Практично, сада је анализу потребно вршити здруженим кернелом. Пошто производ радијално симетричних кернела дозвољава дефинисање посебног пропусног опсег за сваки кернел, проширење на здружени кернел се дефинише као производ два кернела сваког домена посебно [2]. За сваки кернел посебно и даље важе правила која кернели треба да задовољавају, па самим тим и за резултујући кернел, јер смо поменули да је један од начина генерисања вишедимензионалног кернела уствари производ више једнодимензионалних кернела. Здружени кернел, дакле, има следећи облик;

$$K(\mathbf{x}) = \frac{c_{k1}}{h_s^2} k\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|^2\right) \frac{c_{k2}}{h_r^p} k\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|^2\right) \quad (3.1)$$

⁷ Услед недостатка литературе на српском језику, користиће се термини на енглеском језику.

Све константе и друге вредности у претходној једначини имају иста значења као што је дефинисано у предходној дискусији, с тим што је у овом случају познато да spatial домен има димензије $m \times n \times 2$, док range домен има димензије $m \times n \times p$. Такође, анализу за различите домене вршимо са различитим деловима вектора обележја. Вектор обележја у овом случају има сложену структуру која се састоји од дела spatial домена и дела range домена.

Нпр. вектор обележја за мирну колор слику има следећу структуру $\mathbf{x} = [\mathbf{x}^s, \mathbf{x}^r]$, где је вектор spatial домена $\mathbf{x}^s = [x, y]$, а вектор range домена је функција вектора spatial домена и ако је изабран стандардни RGB колор простор, има следећу структуру $\mathbf{x}^r = [R(\mathbf{x}^s), G(\mathbf{x}^s), B(\mathbf{x}^s)]$. На овај начин се дефинише вишедимензионални простор обележја који ће да сачува информацију о просторној расподели модова. Приликом анализе кернел spatial домена ће узети у обзир само прва два члана, који се односе на просторне координате, док ће кернел range домена узети у обзир само остале чланове вектора обележја који се односе на боју.

3.2. Налажење модова функције

Основна анализа која може да се изврши помоћу mean shift алгоритма јесте налажење модова функције. Пошто је сам алгоритам апликацијски независан, ова функција може да се укључи у друге и да се тако формира сложенија процедура. Тако се, нпр. ова функција користи приликом филтрирања слика, а филтрирање се даље користи приликом сегментације слике. С обзиром да алгоритам заснован на естимацији градијента, врло лако је имплементирати алогритам за детекцију модова. Даље, пошто је конвергенција подразумевано загарантована, знамо да вектор помераја средње вредности конвергира ка нули, Итеративни процес описан једначином:

$$\mathbf{m}(\mathbf{y}_j) = \mathbf{y}_{j+1} - \mathbf{y}_j \quad (3.2)$$

Итеративни алгоритам за проналажење модова (псеудокод)

формира се сет улазних података у простору обележја $S = \{\mathbf{x}_i, i=1, 2, \dots, n\}$, па се за $i=1, 2, \dots, n$ врши анализа:

1. **иницијализација**; поставља се иницијални центар кернела као $\mathbf{y}_1 = \mathbf{x}$
2. **итерација**; рачуна се нови центар кернела према формули:

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}$$

3. рачуна се померај средње вредности $\mathbf{m}(\mathbf{y}_j) = \mathbf{y}_{j+1} - \mathbf{y}_j$
 4. **конвергенција**; понављају се кораци 2 и 3 док се не задовољи услов $\|\mathbf{m}(\mathbf{y}_j)\| \leq \varepsilon$
 5. **крај**; поставља се тачка конвергенције $\mathbf{y}_c^i = \mathbf{y}_j$ за избрани улазни узорак \mathbf{x}_i
-

се зауставља у оном тренутку када буде $\mathbf{y}_{j+1} = \mathbf{y}_c$, односно када буде испуњен услов:

$$\mathbf{m}(\mathbf{y}_c) = \mathbf{y}_c - \mathbf{y}_c = 0 \quad (3.3)$$

То значи да се итеративни процес зауставља када градијент буде једнак нули, а пошто градијент представља вектор највеће промене функције у датој тачки, анулирањем његове вредности долазимо на максимум функције. У овом случају, такође, можемо да применимо basin of attraction метод и да не рачунамо тачку конвергенције за сваки улазни вектор. Практична реализација ове функције може да се оствари у два корака, у првом кораку проналажење стационарних тачака које одговарају врховима функције, и у другом кораку провера да ли се ради о локалним максимумима или превојним тачкама. Ако се ради о локалним максимумима или превојним тачкама те тачке не одговарају модовима. Према математичким дефиницијама, да ли је нека тачака мод или не, најлакше се може утврдити тако што се тачка првобитне конвергенције помери за вектор мале норме и посматра да ли ће се вратити у исту тачку конвергенције. Ако се не врати у првобитну тачку конвергенције, значи да се радило о локалном максимуму или превојној тачки [2].

3.3 Филтрирање са чувањем ивица

Карактеристика конвергенције према најгушћем региону се може искористити за реализацију филтра који ће да врши усредњавање области са малом варијацијом а да при томе не нарушава ивице на слици. Ова процедура може да се искористи као претпроцесирање слике приликом кластеризације вектора обележја

Филтрирање са чувањем ивица (псеудокод)

формира се сет улазних података $S = \left(\left[\mathbf{x}_i^s f(\mathbf{x}_i^s) \right], i = 1, 2, \dots, n \right)$, према оригиналној слици

$f = f(\mathbf{x}^s)$; за сваки вектор из формираног простора обележја врши се итеративни поступак:

1. **иницијализација**; за $j = 1$ поставља се иницијални центар кернела као $\mathbf{y}_1 = \mathbf{x}_i$

$$\sum_{i=1}^n \mathbf{x}_i g_s \left(\left\| \frac{\mathbf{y}_j^s - \mathbf{x}_i^s}{h_s} \right\|^2 \right) g_r \left(\left\| \frac{\mathbf{y}_j^r - \mathbf{x}_i^r}{h_r} \right\|^2 \right)$$

2. **итерација**; рачуна се следећи центар: $\mathbf{y}_{j+1} = \frac{\sum_{i=1}^n \mathbf{x}_i g_s \left(\left\| \frac{\mathbf{y}_j^s - \mathbf{x}_i^s}{h_s} \right\|^2 \right) g_r \left(\left\| \frac{\mathbf{y}_j^r - \mathbf{x}_i^r}{h_r} \right\|^2 \right)}{\sum_{i=1}^n g_s \left(\left\| \frac{\mathbf{y}_j^s - \mathbf{x}_i^s}{h_s} \right\|^2 \right) g_r \left(\left\| \frac{\mathbf{y}_j^r - \mathbf{x}_i^r}{h_r} \right\|^2 \right)}$

3. рачуна се померај средње вредности $\mathbf{m}(\mathbf{y}_j) = \mathbf{y}_{j+1} - \mathbf{y}_j$

4. **конвергенција**; понављају се кораци 2 и 3 док се не задовољи услов $\|\mathbf{m}(\mathbf{y}_c)\| \leq \varepsilon$,

где је $\mathbf{y}_c = \begin{bmatrix} \mathbf{y}_c^s \\ \mathbf{y}_c^r \end{bmatrix}$

5. **крај**; поставља се излазни податак, тј. филтрирана вредност $\mathbf{F}_i = \mathbf{y}_c^r$

6. поступак се понавља за сваки улазни податак и чува се вредност тачака конвергенције у простору области за сваку улазну тачку, као излазни податак

Излаз је филтрирана слика \mathbf{F}

Анализа се врши у здруженом домену, као што је описано у поглављу о дефинисању простора обележја. Здружени домен узима у обзир и геометријску удаљеност и сличност према карактеристикама боја, па ако су карактеристике боја сличне, примењује се нископропусно филтрирање. Са друге стране, ако су карактеристике боја различите, што је случај у областима ивица, то узрокује да кернел обухвата регион који је празан па уопште не долази до филтрирања, односно, чува се информација о постојању ивица [1]. Посебно интересантан случај када користимо кернел Епанечњикова, пошто видимо да се приликом рачунања новог центра користи само профил коме је кернел Епанечњикова сенка, а то је профил униформног кернела. У том случају имамо дословно рачунање средње вредности, без тежинских функција [5]. Тада алгоритам из претходне табеле постаје знатно једноставнији, јер се нови центар кернела рачуна према следећој формули:

$$\mathbf{y}_{j+1} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i, \quad \mathbf{x}_i \in S(\mathbf{y}_j) \quad (3.4)$$

У претходној једначини у обзир се, као што је познато, узимају само подаци који се налазе унутар пропусног опсега, што је означено као да припадају сфери S са центром у тачки \mathbf{y}_j . Излазни подаци представљају матрицу димензија $m \times n \times p$, где су подаци који одговарају позицијама $p=1,2$ у ствари вредности у spatial домену простора према којима конвергира пиксел са сваке од позиција из матрице $m \times n$, и њих можемо да искористимо приликом кластеризације. Подаци који одговарају позицијама $p=3:end$ су вредности у range домену којима конвергира вредност пиксела са сваке позиције из матрице $m \times n$, и то су вредности које се односе на филтрирану слику.

3.4. Сегментација мирних слика

За потребе сегментације слика користимо модификовану и дорађену методу за филтрирање описану у претходном поглављу. Наиме, након што се изврши филтрирање, потребно је извршити кластеризацију тако што ће се придружити лабела свим модовима који су лоцирани приликом филтрирања. Наиме, ако неколико пиксела конвергира према једној тачки сви ти узорци се обележавају да припадају једном кластеру. Када тај кластер пресликамо назад у неки колор простор, можемо да видимо који сегмент смо добили.

Потребно је придружити лабеле свим пикселима у простору обележја и на тај начин извршити сегментацију комплетне слике. На овај начин ћемо добити велики број сегмената на слици, нарочито ако је пропусни опсег у spatial домену мали. Ипак, међу тим сегментима ће бити и велики број лажних сегмената, јер се приликом кластеризације у простору обележја занемарила чињеница да је један број тачака конвергенције међусобно ближи од пропусног опсега у оба домена, што уствари говори да се ради о једном кластеру. Такође, број сегмената је могуће смањити ако се уклоне сви сегменти који имају мање пиксела од неке минималне вредности. То су обично сегменти од по неколико пиксела чије вредности у range домену дефинишу засебан кластер, а обично не носе значајну семантичку информацију. Њихово уклањање се врши придруживањем неком од суседних кластера. У том случају имамо три улазна податка, пропусне опсеге за оба домена и минималан број пиксела који један сегмент треба да садржи.

Сегментација мирних слика (псеудокод)

формира се сет улазних података у простору обележја $S = \left(\left[\mathbf{x}_i^s f(\mathbf{x}_i^s) \right], i = 1, 2..n \right)$,

према оригиналној слици $f = f(\mathbf{x}^s)$;

1. **филтрирање**; врши се филтрирање улазних података према описаном алгоритму и чувају се подаци о тачкама конвергенције $\mathbf{y}_c = \left[\mathbf{y}_c^s \mathbf{y}_c^r \right]$
2. **кластеризација**; врши се кластеризација излазних података тако што се у један кластер придружују сви узорци који конвергирају ка једном моду, а такође, спајају се они кластери чији су модови на мањој удаљености од пропусног опсега
3. **спајање**; сви кластери који садрже мање од M пиксела се спајају са једним од суседних кластера

излаз је матрица димензија $m \times n$, која на месту сваког пиксела има лателу која означава којем кластеру тај пиксел припада

3.5. Сегментација видеа и праћење објеката

У претходним поглављима описане су основне апликације које могу да се изведу помоћу mean shift алгоритма. Наравно, оне могу да се искористе приликом извршавања сложенијих операција. Нпр. резултати сегментације мирних слика се могу искористити приликом сегментације, екстракције објеката и алгоритама за праћење објеката у видеу. У том случају имамо и временску променљиву, па се анализа обавља у здруженом spatial-range-time домену. Ту постоје два могућа приступа, интерфрејмски и интрафрејмски. Интерфрејмски приступ користи временску редунантност између узастопних фрејмова, док интрафрејмски приступ врши сегментацију слика за сваки фрејм посебно у случају да се појављује нови објекат на видеу. Приликом реализације алгоритма за сегментацију видеа, користе се оба приступа у зависности од тренутне потребе.

Интерфрејмски метод врши поређење објеката, односно сегмената између узастопних фрејмова и на тај начин може да се реализује праћење објеката, при чему се посебно обраћа пажња на покретне објекте. То се реализује помоћу једноставне идеје поклапања модова, према којој мод неког објекта, тј. кластера, на следећем фрејму, треба да буде на истом или приближно истом месту као и на претходном фрејму. Да би могли да извршимо овакво поређење, као што је већ речено, потребно је да се примени здружени spatial-range-time домен. У овом случају имамо сет улазних података $S = [\mathbf{x}^s(t), f(\mathbf{x}^s(t))]$, при чему је $\mathbf{x}^s(t) = [x(t), y(t)]$, док вектор домена области може да се дефинише као $f(\mathbf{x}^s(t)) = [R(\mathbf{x}^s(t)), G(\mathbf{x}^s(t)), B(\mathbf{x}^s(t))]$. Вектор промене средње вредности у овако дефинисаном простору обележја има следећи облик [1]:

$$\mathbf{m}(\mathbf{x}_t, \mathbf{x}_{t-1}) = \frac{\sum_{i=1}^n \mathbf{x}_{i,t-1} g_s \left(\left\| \frac{\mathbf{y}_t^s - \mathbf{x}_{i,t-1}^s}{h_s} \right\|^2 \right) g_r \left(\left\| \frac{\mathbf{y}_t^r - \mathbf{x}_{i,t-1}^r}{h_r} \right\|^2 \right)}{\sum_{i=1}^n g_s \left(\left\| \frac{\mathbf{y}_t^s - \mathbf{x}_{i,t-1}^s}{h_s} \right\|^2 \right) g_r \left(\left\| \frac{\mathbf{y}_t^r - \mathbf{x}_{i,t-1}^r}{h_r} \right\|^2 \right)} - \mathbf{x}_t \quad (3.5)$$

Из претходне једначине видимо да се померај вектора средње вредности дефинише као померај центара средње вредности у временском домену, тј. видимо да померај средње вредности има нешто измењену улогу у односу на претходне случајеве. Овде није циљ да се досегне мод функције, већ се прати промена мода у току времена и на тај начин реализује праћење објекта (који је представљен одговарајућим модом) у току времена. Све карактеристике које је mean shift алгоритам показао приликом реализације проналажења мода могу да се пресликају и на овај случај.

Сегментација покретних слика (видеа) (псеудокод)

1. **сегментација првог фрејма**; врши се сегментација првог фрејма видеа описаног са простором обележја $\mathbf{x}(t) = [\mathbf{x}^s(t) f(\mathbf{x}^s(t))]$, у тренутку $t=1$ према описаном алгоритму, па на основу улазне слике $\mathbf{f}(\mathbf{x}(t=1))$ добија сегментирана слика првог фрејма $\mathbf{F}(\mathbf{x}(t=1))$
 2. **претпроцесирање**; врши се претпроцесирање за фрејм $t=t+1$ и то извршавањем филтрирања о range домену по временској оси са дефинисаним пропусним опсегом
 3. **spatial-range-time mean shift филтрирање**;
 - поставља се центар кернела у тачку $\mathbf{x}(t) = [\mathbf{x}^s(t) f(\mathbf{x}^s(t))]$, тј. одабира се пиксел из текућег фрејма и постављају се иницијалне вредности $j=1$ и $\mathbf{y}_j = \mathbf{x}$, такође се прикупаља сет узорака из претходне итерације $\mathbf{x}_i(t-1) = [\mathbf{x}_i^s(t-1) f(\mathbf{x}_i^s(t-1))]$ са центром у $\mathbf{y}_j = \mathbf{x}$ који се налазе унутар пропусног опсега h_s
 - поставља се центар кернела у тачку $\mathbf{x}(t) = [\mathbf{x}^s(t) f(\mathbf{x}^s(t))]$, тј. одабира се пиксел из текућег фрејма и постављају се иницијалне вредности $j=1$ и $\mathbf{y}_j = \mathbf{x}$, такође се прикупаља сет узорака из претходне итерације $\mathbf{x}_i(t-1) = [\mathbf{x}_i^s(t-1) f(\mathbf{x}_i^s(t-1))]$, $i=1,2,..n$ са центром у $\mathbf{y}_j = \mathbf{x}$ који се налазе унутар пропусног опсега h_s
 - рачуна се тачка конвергенције на основу једначине из корака два за сегментацију мирних слика \mathbf{y}_c
 - ако је задовољен услов $\|\mathbf{x}^r(t) - \mathbf{y}_c^r(t)\|^2 \leq h_r$, тада је пронађен одговарајући мод за изабрани пиксел и поставља се $\mathbf{F}(\mathbf{x}(t)) = \mathbf{F}(\mathbf{x}(t-1))$, у супротном прелази се на интрафрејмски модел рада
 4. понављају се кораци 2 и 3 за све тачке у текућем фрејму
 5. **сегментација**; сегментира се филтрирани низ слика $\mathbf{F}(\mathbf{x}(t))$ и резултат се смешта у $\hat{\mathbf{F}}(\mathbf{x}(t))$
 6. **повезивање**; повезују се сегменти из суседних фрејмова
 7. понављају се кораци од 2 до 6 за све фрејмове
- Излаз је низ сегментираних слика $\hat{\mathbf{F}}(\mathbf{x}(t))$, $t=1,2,..$

Исто тако алгоритам за праћење објеката ће да прилагођава своје карактеристике у случају да се објекат који се прати на видеу креће брже (што подразумева већи размак мода између два суседна фрејма) или спорије (што узрокује мањи померај мода између два фрејма), па се норма вектора помераја мења сразмерно брзини објекта.

Приликом сегментације видеа, важно је скренути пажњу да је у једначини (3.5) сет података простора обележја узет из претходног фрејма $x_i(t-1)$, $i = 1, 2, \dots, n$, док се обележја иницијалног кернела узимају из текућег фрејма $x(t)$. Сада се може мало детаљније објаснити на који начин је сегментација мирних употребљена у случају сегментације видеа. У новом тренутку вектор обележја $x(t)$ који је део неког објекта или региона, се дефинише као део неког региона из претходног тренутка ако његова удаљеност од тачке конвергенције тог региона у претходном фрејму $y_c(t-1)$ мања од дефинисаног пропусног опсега $d(x^r(t) - y_c^r(t-1)) < h_r$. Ако је удаљеност већа од дефинисаног попусног опсега, сматра се да је одабрани узорак на текућем фрејму део неког новог објекта који се појавио са текућим фрејмом и тада се прелази на интрафрејмски поступак естимације локалних модова.

Интрафрејмски поступак се користи када се за узорак из текућег фрејма не пронађе припадајући регион у претходном фрејму. За издвајање нових објеката користи се искључиво информација подударна мирној слици тј. само текући фрејм.

Дакле, алгоритам за сегментацију видеа можемо да опишемо на следећи начин. У првој итерацији се обичан spatial-range алгоритам за сегментацију мирних слика да би се извршила сегментација првог фрејма, а почев од другог фрејма користи се здружени spatial-range-time домен за сегментацију. Узимањем у обзир и временске варијабле, између текућег и претходног фрејма се врши процес "придруживања модова", тј. за сваки пиксел из текућег фрејма се проналази одговарајући мод из претходног фрејма и ако се тај мод нађе на место одговарајућег пиксела се уписује вредност конвергенције тог подударајућег мода из претходне итерације. Међутим, ако се такав мод не пронађе, у том случају се за тај пиксел примењује алгоритам сегментације мирних слика у текућем фрејму. Да би се повећала стабилност сегментације у времену, може да се обави претпроцесирање у домену области у времену, што отклања мање варијације у времену које могу да изазову нестабилност приликом придруживања модова.

4. РЕАЛИЗАЦИЈА ФИЛТРИРАЊА И СЕГМЕНТАЦИЈЕ

У претходним главама детаљно је описан математички модел mean shift алгоритма, могућности његове примене и начини на које се он користи за сегментацију слика. Поред изведених теоретских карактеристика дати су и псеудокодови за основне апликације. У овој глави ће бити приказани резултати добијени извршавањем филтрирања и сегментације према описаном алгоритму и биће извршено поређење резултата добијених са различитим параметрима. У литератури [1], [2], [5], [6], [12] су приказани резултати експеримената са различитим параметрима, такође, описане су бројне методе за унапређење перформанси алгоритма [10], [18], [22].

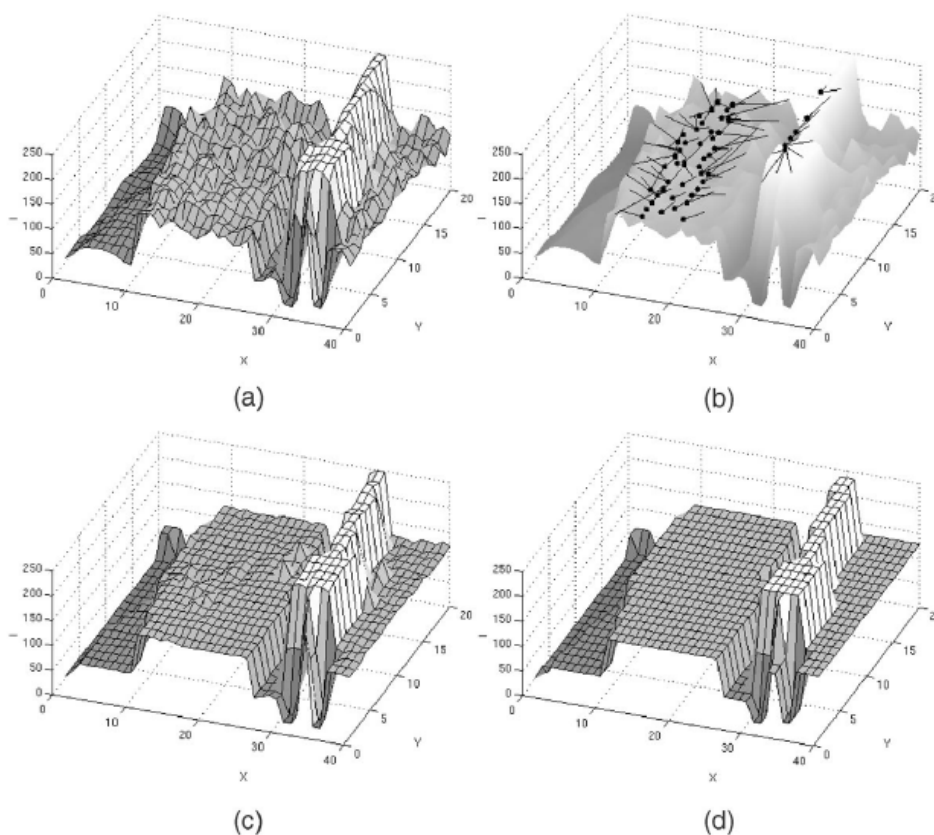
Већ су изведене све потребне теоретске карактеристике функције за филтрирање и сегментацију засновано на mean shift алгоритму. Знамо да ће се филтрирањем изравнати вредности на областима са малим варијацијама, док ће се сачувати све потребне информације о ивицама на слици. На Слици 4.1 је приказан начин на који се врши филтрирање. На Слици 4.1a је један део слике који приказује ниво интензитета светлине у односу на координате пиксела на слици. На Слици 4.1b су приказане трајекторије по којима се долази до локалних модова и на крају Слика 4.1c приказује резултате филтрирања слично као на првој слици. Може да се види да су мање варијације усредњене, док није било утицаја на ивице.

Сегментација врши изједначавање светлина у областима са усредњеним варијацијама светлине, као што је приказано на Слици 4.1d. Такве нивое, изједначене према њиховој средњој вредности, називамо регионима. Можемо да уочимо да је приликом сегментације очувана информација о ивицама, што је потребно да се очува семантички смисао на слици.

4.1. Филтрирање слика

У овом поглављу је описан поступак кориштен за филтрирање слика према *mean shift* алгоритму. Улазни подаци су слика, која се филтрира и пропусни опсези у *spatial* домену и *range* домену, а излаз је матрица која се добија конкатенацијом матрице конвергенције у *spatial* домену и матрице конвергенције у *range* домену. Дакле, за сваки пиксел је дат вектор чија прва два елемента представљају координате којима тај пиксел конвергира у *spatial* домену, а остали чланови су вредности којој се конвергира у *range* домену. Произвољно се као улазни параметри могу задати и максималан број итерација, који зауставља тражење мода после одређеног броја итерација, па евентуално скраћује време рачунања, затим праг тачности, који дефинише максималну удаљеност између вектора обележја у узастопним итерацијама када можемо усвојити да се ради о нултој вредности градијента, односно која дефинише да ли смо пронашли мод. Такође, преко улазног параметра можемо да изаберемо кернел Епанечњикова или Гаусов кернел. У наведеним експериментима користио се искључиво кернел Епанечњикова, због своје једноставности⁸ и веће брзине у односу на Гаусов кернел.

У литератури [2], [5] је наведено да је немогуће одредити одговарајуће пропусне опсеге за широки опсег врста слика, па је за њихово аутоматско одређивање потребан сложенији алгоритам, о чему овде неће бити речи. Ради одређивања оптималног пропусног опсега извршено је филтрирање слике *cameraman.tif* са пропусним опсезима од вредности (8,8) до вредности (16,16), што је приказано на сликама од 4.3 до 4.11.



Слика 4.1: Илустрација филтрирања и сегментације *mean shift* алгоритмом

⁸ Кернел Епанечњикова је сенка униформном кернелу.



Слика 4.2: Оригинална слика



Слика 4.3: Филтрирана слика (8,8)



Слика 4.4: Филтрирана слика (8,12)



Слика 4.5: Филтрирана слика (8,16)



Слика 4.6: Филтрирана слика (12,8)



Слика 4.7: Филтрирана слика (12,12)



Слика 4.8: Филтрирана слика (12,16)



Слика 4.9: Филтрирана слика (16,8)



Слика 4.10: Филтрирана слика (16,12)



Слика 4.11: Филтрирана слика (16,16)

Из претходних примера може да се примети да је резултат филтрирања поравнање области са мањим варијацијама, и то у зависности од пропусног опсега. Такође, може да се примети да већи пропусни опсег у *range* домену има више утицаја на филтрирање него повећање пропусног опсега у *spatial* домену. То се може објаснити чињеницом да се у *range* домену уствари врши *права* селекција података који улазе у здружени кернел. Ако је пропусни опсег у *range* домену мањи, мало података ће да се нађе у кернелу чак и ако је пропусни опсег у *spatial* домену већи. Тако, на сликама од 4.3 до 4.11 можемо да видимо да је пропусни опсег (12,12) Слика 4.7 приближно ефикасан као и пропусни опсег (16,12) Слика 4.10. Можемо да закључимо да за ову врсту слика пропусни опсег (12,16) даје најбоље резултате, уз напомену да повећање пропусног опсега у *spatial* домену значајно продужава трајање филтрирања.

Приликом реализације, пре свега, потребно је проверити шта је задано од произвољних параметара и ако неки нису задани, потребно је да се поставе на подразумевајуће вредности. Такође, потребно је проверити да ли је задано са којим кернелом се врши филтрирање, и ако није да се постави кернел Епанечњикова. Даље, врши се иницијализација улазних података, односно слике, врши се постављање мреже, односно матрице величине слике са редним бројевима свих пиксела, па се на основу тога и улазне слике креира простор обележја. Филтрирање се врши тако што се за

сваки пиксел нађу сви подаци који се налазе на удаљености од иницијалне тачке мањој од пропусних опсега у оба домена и на основу тих података рачуна средња вредност. Удаљеност између два вектора у простору обележја рачунамо за сваки домен посебно, и то по Еуклидовој метрици. Узорци који се налазе у пропусном опсегу и у једном и у другом домену се налазе у области коју обухвата здружени кернелу и тада се рачуна нова средња вредност. Након рачунања средње вредности, проверамо да ли је дошло до конвергенције, и ако није поступак се понавља. Када дође до конвергенције, као излазни податак се, за сваки пиксел, уписује тачка конвергенције у оба домена, односно вектор конвергенције у здруженом домену. Код којим је реализована функција у програмском пакету MATLAB се може наћи у Додатку А.

Такође, из следећег примера, где је извршено филтрирање колор слике *football.jpg* и сиве слике *coins.png*, са пропусним опегом (12,16), можемо да закључимо да се за тај пропусни опсег добијају задовољавајући резултати, тј. поравнање мањих варијација уз чување ивица:



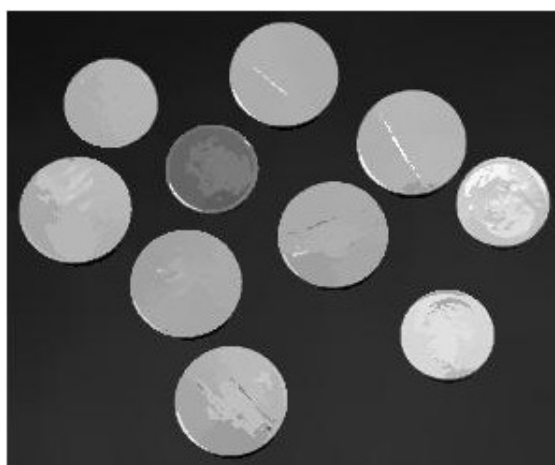
Слика 4.12: Оригинална слика



Слика 4.13: Филтрирана слика (12,16)



Слика 4.14: Оригинална слика



Слика 4.15: Филтрирана слика (12,16)

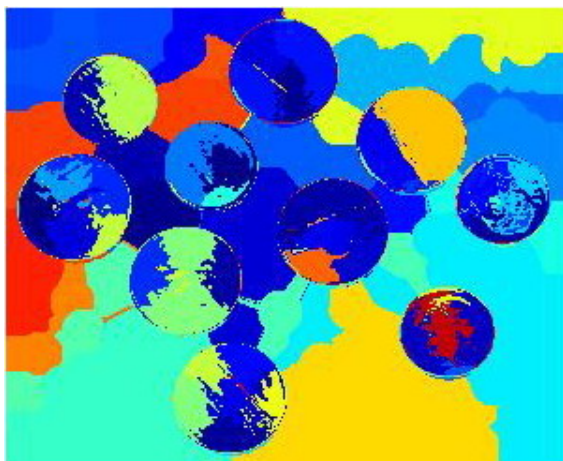
4.2. Сегментација слика

Сегментацију слика вршимо према резултатима филтрирања извршеног према mean shift алгоритму, и то груписањем (кластеризацијом) свих узорака (вектора обележја) који се налазе на удаљености која је мања од ширине пропусних опсега у оба домена, чиме се добија поравнање на областима са малим варијацијама, као на слици 4.1d. Такође, може да се имплементира и "избацавање" оних кластера који садрже мање од предвиђеног броја пиксела. Излазни податак је матрица димензија величине слике, која на месту сваког пиксела садржи лабелу која означава кластер коме тај пиксел припада. Сегментирана слика уместо оригиналне вредности боје пиксела има усредњену вредност пиксела према кластеру коме тај пиксел припада. Резултати сегментације најбоље могу да се интерпретирају приликом издвајања објеката на слици, где се лако уочава и квалитет издвајања семантички смислених објеката. Код слика са мање јасним семантичким целинама, сегментација, такође, има врло важну улогу, нарочито са све учесталијом имплементацијом система за претраживање база слика на основу садржаја [30]. Од кључног значаја је избор пропусног опсега од кога највише зависи број сегмената. На изабраним примерима, на сликама су дати просторни распоред сегмената према лабелама и сегментиране слике. Алгоритам је реализован тако што су, након кластеризације, спојени суседни кластери ако се налазе на малом растојању у range домену.

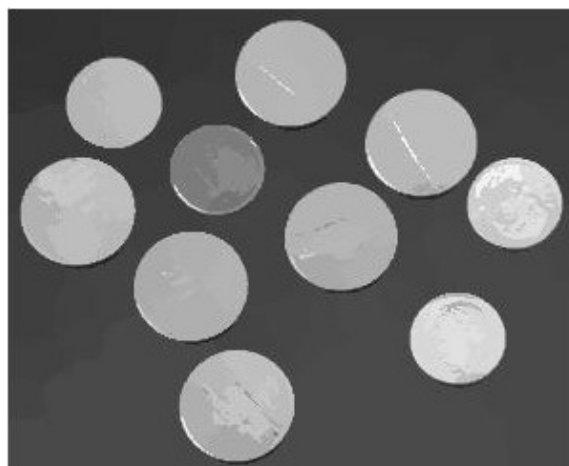
Реализација функције кластеризације је, такође, урађена у програмском пакету MATLAB и код је доступан у Додатку Б. Приликом иницијализације се креира нулта матрица величине слике у коју ће се уписивати лабеле на место сваког пиксела. Улазни подаци су матрица конвергенције у здруженом простору, коју смо добили применом филтрирања и пропусни опсеги у оба домена, и то исти они које смо користили приликом филтрирања. Да би се добио што једноставнији поступак врши се конкатенација улазне матрице са истом матрицом мреже коју смо користили да означимо положај узорака приликом филтрирања. На тај начин се лакше издвајају елементи који нису додељени нити једном кластеру. Ова функција, такође, пребројава колико који кластер садржи елемената, што је потребно знати ако се жели реализовати уклањање кластера са малим бројем елемената.

Други корак представља функција која спаја суседне сегменте у случају да имају приближно исте карактеристике у range домену, односно реализује се спајање свих кластера који се налазе на мањој удаљености од пропусног опсега у spatial домену и који се налазе на малом растојању у range домену. Овде је, такође, од највећег значаја избор модификованог пропусног опсега у оба домена. У spatial домену је потребно одабрати такав пропусни опсег да може да споји задовољавајући број нпр. сегмената позадине, али да притом не споји различите објекте са различитих делова слике који имају приближно исте вредности у range домену.

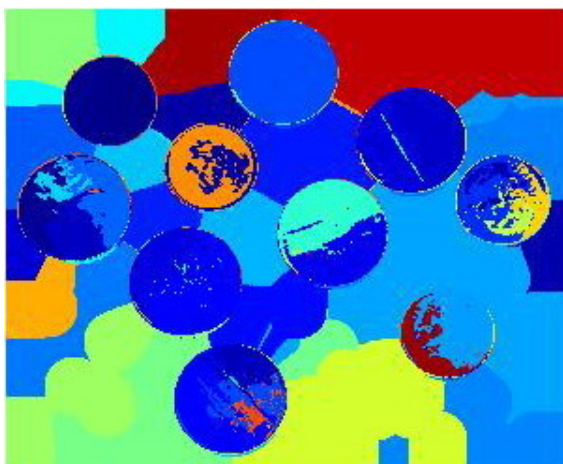
Улазне вредности ове функције су матрица конвергенције добијена филтрирањем слике, затим матрица лабела добијена у првом кораку кластеризације, пропусни опсеги који се бирају према објашњеним правилима. Излазне вредности су поправљена матрица лабела и сегментирана слика, те број кластера пре и после спајања. Резултати сегментације су на сликама од 4.16 до 4.21, и то за пропусне опсеге (12, 16) и (16,16):



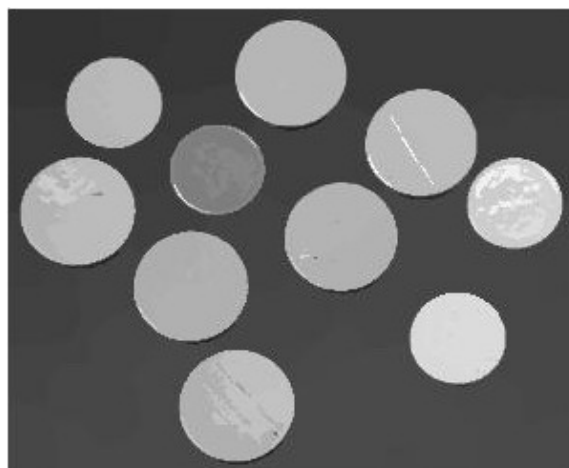
Слика 4.16: Слика лабела (12, 16)



Слика 4.17: Сегментирана слика (12, 16)



Слика 4.18: Слика лабела (16, 16)



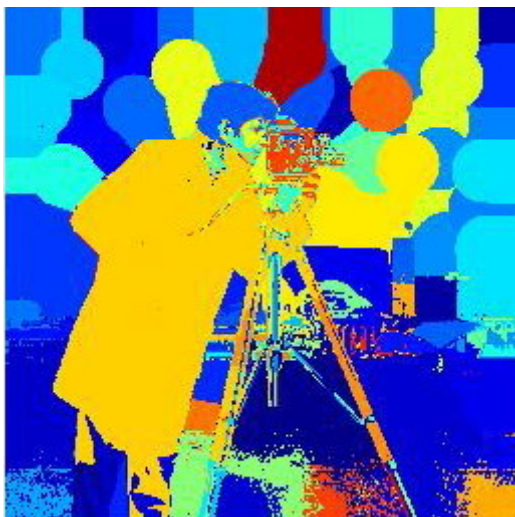
Слика 4.19: Сегментирана слика (16, 16)



Слика 4.20: Слика лабела (12, 16)



Слика 4.21: Сегментирана слика (12, 16)

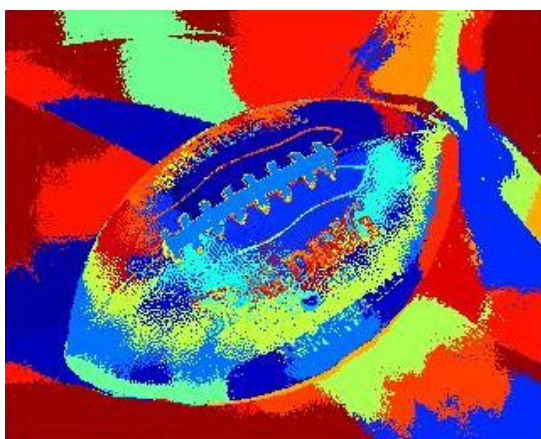


Слика 4.22: Слика лабела



Слика 4.23: Сегментирана слика

Можемо да видимо да су и на сложенијој слици *cameraman.tif* издвојени семантички смислени објекти, и у овом случају у више сегмената. То је и логично, јер се користио пропусни опсег у *spatial* домену знатно мањи од величине слике, па и после спајања суседних сегмената не можемо очекивати да ће небо на слици представљати један сегмент. Међутим, применом у системима са интегрисаним упаривањем региона [30], адекватним поређењем сегменти који припадају само нпр. небу требали би да се посматрају као један сегмент. На сликама 4.22 и 4.23 су приказани резултати кластеризације и сегментације слике *football.jpg* и то са пропусним опсегом (12,16). Можемо да закључимо да је сегментација са семантичког становишта задовољавајућа, јер се лопта издваја из позадине.



Слика 4.24: Слика лабела



Слика 4.25: Сегментирана слика (12,16)

ЗАКЉУЧАК

У раду је дата детаљна математичка основа mean shift алгоритма, представљене су основне карактеристике апликација заснованих на овом алгоритму и представљене су основне могућности примене алгоритма. Практична реализација је изведена у програмском пакету MATLAB и представља дословну програмску примену алгоритма описаног у литератури. Ипак, приликом практичне реализације аутори у свету дају предност програмском језику C++, првенствено јер је извршавање алгоритма знатно брже. Недостатак реализације у MATLAB-у је то што је извршавање функција релативно споро, што није случај са реализацијом у програмском језику C++. На интернету је доступан EDISON систем⁹ [2], [5], [12], који реализује све основне функције са mean shift алгоритмом које се баве обрадом мирних слика. Међутим, одабрана је реализација у MATLAB-у, јер је програмирање брже и једноставније.

Због својих предности, апликације засноване на mean shift алгоритму су последњих година преузеле водећу улогу у области дигиталне обраде слике. Развијене су разне модификације алгоритма, тако да он може боље да интерпретира случајне просторе обележја и да им се прилагоди. Такође, развијене су разне технике за примену алгоритма приликом обраде покретних слика и то првенствено у системима за видео надзор. Апликације засноване на mean shift алгоритму за праћење објеката, могу да се поделе на два типа, први тип који користи адаптивни кернел, такозвани CAMShift (Continuously Adaptive Mean Shift) алгоритми и други тип алгоритма који користи комбинацију mean shift и Kalmanовог филтра. Mean shift алгоритам је показао добре резултате и приликом примене у апликацијама за детекцију ивица на сликама. Уопште апликације засноване на овом алгоритму могу да се користе за велики опсег практичних проблема у области дигиталне обраде слике.

Уопштено, можемо да кажемо да је mean shift алгоритам један универзалан алат за робусну анализу простора обележја, који има значајну предност јер се резултати анализе контролишу само обликом кернела, односно пропусним опсегом. Стога, све

⁹ <http://www.caip.rutgers.edu/riul/research/code.html>

апликације засноване на овом алгоритму могу да се лако уграде у неке сложеније процедуре и да се аутоматски прилагођавају на основу обележја вишег нивоа. Сложеније апликације се програмирају тако да се са mean shift алгоритмом комбинују процедуре за аутоматско подешавање пропусног опсега, као што је то описано код адаптивног алгоритма или према неком другом принципу. Дакле, и даље се врше истраживања у смеру побољшања експлоатације и бољег подешавања карактеристика апликација заснованих на овом алгоритму. Може да се закључи да је карактеристика mean shift алгоритма да буде привучен максимумима модела веома вредан практичан модел чија је прилагодљивост веома битна за практично решавање проблема компјутерског вида [2].

Најбоља илустрација претходне тврдње може да буде једна од последњих апликација за сегментацију заснована на mean shift алгоритму [10], која као критеријум засуштавања, односно проналаска модела користи минимум ентропије. На овај начин се у сегментационој слици чувају природни детаљи и поред врло прецизне сегментације.

Поред детаљног описа алгоритма и његових могућности, циљ овог рада је био да искористи mean shift алгоритам да изврши сегментацију слика која може да се користи у системима за претраживање база слика на основу сличности региона. Приказани примери потврђују да се постављени задатак успешно извршио, јер су приликом сегментације издвојене све семантичке целине на сликама, са релативно малим бројем сегмената.

Додатак А

Филтрирање према mean shift алгоритму, извршено у МАТЛАВ-у се реализује према следећем коду:

```
function M = msfilter(X, sig_spt, sig_rng, softflag, maxiter, mindelta)

% M = msfilter(X, sig_spt, sig_rng, softflag, maxiter, mindelta) vrsi
% filtriranje ulazne slike X u zdruzenom spatial/range domenu prema mean
% shift algoritmu. Propusni opsezi su sig_spt i sig_rng, a maxiter
% predstavlja maksimalan broj iteracija, dok mindelta predstavlja prag
% konvergencije. Postavljanjem softflag na 1 vrsi se filtriranje sa
% Gausovim kernelom, dok je podrazumevajuci kernel Epanecnjikova. Izlazni
% podaci su predstavljeni matricom M koja predstavlja konkatenciju
% konvergentih tacaka u oba domena.

[sig_spt,er] = checknumericargs( sig_spt, 1, 0, 1 ); error(er);
if(nargin<4 || isempty(softflag))
    softflag = 0;
end;
if(nargin<5 || isempty(maxiter))
    maxiter = 100;
end;
if(nargin<6 || isempty(mindelta))
    mindelta = .01;
end;

[mrows, ncols, p] = size(X);
p = p+2;
[grid_rs grid_cs] = ndgrid( 1:mrows, 1:ncols );
data = cat( 3, cat( 3, grid_rs, grid_cs), X);

%%% Glavna petlja

if( softflag )
    radius = sig_spt*2;
else
    radius = sig_spt;
end;

for i=1:mrows
    for j=1:ncols
        Mij = data(i,j,:);
        Mij = Mij(:)';
        itercount = 0;
        diff = 1;

        while(itercount < maxiter && diff>mindelta)

            % Prikupljanje mogucih korisnih podataka u domenu prostora

            r = round( Mij(1));
            c = round( Mij(2));
            br = max(1,r-radius):min(mrows,r+radius);
            bc = max(1,c-radius):min(ncols,c+radius);
            dw = data(br, bc, : );
            dwr = reshape( dw, [], p );

            % Racunanje sledece srednje vrednosti

            Mij_old = Mij;
```

```

n = size( df, 1);
D = sum(((dwr(:,1:2) - ones(n,1)*Mij(:,1:2))/sig_spt).^2,2);

if(softflag)
    D1 = sum(((dwr(:,3:end) - ones(n,1)*Mij(:,3:end))/sig_rng).^2,2);
    R = max(D,D1);
    P = D.*D1;
    T = P(R<=1);
    dwr = dwr(D <= 1, : );
    n1 = size( dwr, 1);
    D1 = sum(((dwr(:,3:end) - ones(n1,1)*Mij(:,3:end))/sig_rng).^2,2);
    dwr = dwr(D1 <= 1, : );
    S = exp( -T );
    sumS = sum(S);
    Srep = S(:,ones(1,p));
    Mij = sum(dwr .* Srep, 1 )/sumS;
else
    dwr = dwr(D <= 1, : );
    n1 = size( dwr, 1);
    D1 = sum(((dwr(:,3:end) - ones(n1,1)*Mij(:,3:end))/sig_rng).^2,2);
    dwr = dwr(D1 <= 1, : );
    Mij = sum(dwr, 1 ) / size(dwr,1);
end;

% Provera da li se nasla tacka konvergencije
diff = sum((Mij_old-Mij).^2,2);
itercount = itercount+1;

end;

M(i,j,:) = Mij(:);

end;

end;

figure, imshow(M(:, :, 3:end),[])

```

msfilter.m

Додатак Б

Кластеризација према mean shift алгоритму, извршена у MATLAB-у се реализује према следеће две функције, чији су кодови:

```
function [out,nL] = mscluster(X, sig_spt, sig_rng)

% [out,nL] = mscluster(X, sig_spt, sig_rng) je funkcija koja realizuje
% klasteriaciju filtrirane slike i mapirane u prostor obelezja. Koristi
% se nakon funkcije msfilter i kao izlaz daje matricu labela. Na mestu
% svakog piksela se nalazi labela koja oznacava kojem klasteru taj piksel % pripada.

[mrows, ncols, p] = size(X);
p = p+2;
[grid_rs grid_cs] = ndgrid( 1:mrows, 1:ncols );
M = cat( 3, cat( 3, grid_rs, grid_cs), X);
out = zeros(mrows, ncols);

count = 1;
us = 1;
s2=sig_spt^2;
r2=sig_rng^2;
tic

while(us>0)
    datat = out==0;
    g1=1;
    for i=1:mrows;
        for j=1:ncols;
            if(datat(i,j)==1)
                datap(g1, :, :) = M(i, j, :);
                g1 = g1+1;
            end;
        end;
    end;
    o3 = ceil(rand*g1);
    e(:, :, :) = datap(o3, :, :);
    x = (e(1));
    y = (e(2));

    if(out(x,y)==0)
        out(x,y)=count;
        c=0;

        for i=1:mrows
            for j=1:ncols

                if ((sum((M(x,y,3:4)-M(i,j,3:4)).^2,3)<s2))
                    if(sum((M(x,y,5:end)-M(i,j,5:end)).^2,3)<r2)
                        out(i,j)=count;
                        c=c+1;
                    end;
                end;
            end;
        end;

        nL(count)=c;
        count = count +1;

    end;

    us=0;
    for i=1:mrows;
```

```

        for j=1:ncols;
            if(out(i,j)==0)
                us = us +1;
            end;
        end;
    end;

end;

toc
figure, imshow(label2rgb(out))

```

mscluster.m

```

function [out,tuo,numberC,newNumberC] = msconnect(M,label,s,r)

% function [out,tuo] = msconnect(M,label,s,r) vrši spajanje bliskih segmenata
% u domenu prostora, koji imaju slične karakteristike u domenu oblasti.
% Također, vrši se vizuelno predstavljanje segmentirane slike.

out = label;
s2=s^2;
r2=r^2;
numberC = max(max(out));
[mrows, ncols,p] = size(M);

for c = 1:numberC
    for i=1:p
        a = M(:, :, i);
        X(c, i) = mean2(a(out == c));
    end;
end;

for c = 1:numberC
    for w = 1:numberC
        if(sum((X(c,1:2)-X(w,1:2)).^2,2)<s2)
            if ((sum((X(c,3:end)-X(w,3:end)).^2,2)<r2))
                found = out==w;
                out(found) = c;
            end;
        end;
    end;
end;

figure, imshow(label2rgb(out))

%Prebrojavanje klastera nakon spajanja

newNumberC=0;
out1 = zeros(mrows, ncols);

for i =1:numberC
    if(max(max(out==i)))
        newNumberC = newNumberC + 1;
        foundj = out==i;
        out1(foundj)=newNumberC;
    end;
end;

figure, imshow(label2rgb(out1))

tuo = zeros(mrows, ncols, p-2);
for c = 1:newNumberC
    for i=3:p
        tuo1 = zeros(mrows, ncols);
        a = M(:, :, i);
        X1(c, i-2) = mean2(a(out1 == c));
        tuo1(out1 == c) = X1(c, i-2);
        tuo(:, :, i-2) = tuo(:, :, i-2) + tuo1;
    end;
end;

figure, imshow(tuo)

```

msconnect.m

Литература

- [1] YU – JIN ZHANG, *Advances in Image and Video Segmentation*, Tsinghua University, Beijing, China, IMR Press, 2006.
- [2] DORIN COMANICIU, PETER MEER, Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Transactions on pattern analysis and machine intelligence, May 2002.
- [3] RICHARD O. DUDA, PETER E. HART, DAVID G. STORK, *Pattern Classification*, Second Edition, John Wiley & Sons, Inc., 2001.
- [4] JUE WANG, BO THIESSON, YINGQING XU, MICHAEL COHEN, Image and Video Segmentation by Anisotropic Kernel Mean Shift, Microsoft Research
- [5] DORIN COMANICIU, PETER MEER, Mean Shift Analysis and Applications, Department of Electrical and Computer Engineering Rutgers University, Piscataway, USA
- [6] DORIN COMANICIU, PETER MEER, VISVANATHAN RAMESH, The Variable Bandwidth Mean Shift and Data-Driven Scale Segmentation, Princeton and Piscataway, USA
- [7] A.D. JEPSON, D.J. FLEET, Image Segmentation, 2004.
- [8] MIGUEL Á. CARREIRA-PERPIÑÁN, Fast Nonparametric Clustering with Gaussian Blurring Mean Shift, Department of Computer Science & Electrical Engineering, Oregon Health & Science University, USA, 2006.
- [9] MIGUEL Á. CARREIRA-PERPIÑÁN, Acceleration Strategies for Gaussian Mean-Shift Image Segmentation, Department of Computer Science & Electrical Engineering, Oregon Health & Science University, USA
- [10] ROBERTO RODRÍGUEZ, ANA G. SUAREZ, A New Algorithm for Image Segmentation by Using Iteratively the Mean Shift Filtering, Institute of Cybernetics, Mathematics and Physics, La Habana, Cuba, 2006.
- [11] YIZONG CHENG, Mean Shift, Mode Seeking and Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, August 1995.
- [12] DORIN COMANICIU, PETER MEER, Robust Analysis of Feature Spaces: Color Image Segmentation, Department of Electrical and Computer Engineering Rutgers University, Piscataway, USA

- [13] KIL-MOO LEE, PETER MEER, RAE-HONG PARK, Robust Adaptive Segmentation of Range Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, February 1998.
- [14] BOGDAN GEORGESCU, ILAN SHIMSHONI, PETER MEER, Mean Shift Based Clustering in High Dimensions: A texture Classification Example, Rutgers University, Piscataway, USA and Technion – Israel Institute of Technology
- [15] RAGHAV SUBBARAO, PETER MEER, Nonlinear Mean Shift for Clustering over Analytic Manifolds, Department of Electrical and Computer Engineering, Rutgers University, Piscataway, USA, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006.
- [16] RAGHAV SUBBARAO, PETER MEER, Discontinuity Preserving Filtering over Analytic Manifolds, Department of Electrical and Computer Engineering, Rutgers University, Piscataway, USA
- [17] PETER MEER, Robust Techniques for Computer Vision, Electrical and Computer Engineering Department, Rutgers University, USA, 2004
- [18] JUE WANG, YINGQING XU, HEUNG-YEUNG SHUM, MICHAEL F. COHEN, Video Tooning, University of Washington and Microsoft Research
- [19] CHANGJIANG YANG, RAMANI DURAISWAMI, AHMED ELGAMMAL, LARRY DAVIS, Real-Time Kernel-Based Tracking in Joint Feature-Spatial Spaces, Perceptual Interfaces and Reality Laboratory, University of Maryland and Department of Computer Science, Rutgers University, Piscataway, USA
- [20] C. BELEZNAI, B. FRÜHSÜCK, H. BISCHOF, Human Tracking by Fast Mean Shift Mode Seeking, Journal of Multimedia, April 2006.
- [21] ROBERT T. COLLINS, Mean-shift Blob Tracking through Scale Space, Carnegie Mellon University
- [22] HANZI WANG, DAVID SUTER, False-Peaks-Avoiding Mean Shift Method for Unsupervised Peak-Valley Image Segmentation, Department of Electrical and Computer System Engineering, Monash University, Australia, 2003.
- [23] YAN KE, RAHUL SUKTHANKAR, MARTIAL HEBERT, Efficient Temporal Mean Shift for Activity Recognition in Video, School of Computer Science, Carnegie Mellon and Intel Research Pittsburgh
- [24] SZE HON YAN, SOMNUK PHON-AMMUAISUK, A Case Study of Mean Shift in Image Segmentation, Faculty of Information Technology, Multimedia University, Jalan Multimedia, Selangor Darul Ehsan, Malaysia
- [25] GREG MICHAELSON, ANDY WALLACE, KEVIN HAMMOND, IAIN WALLACE, ARMELLE BONENFANT, ZEZHONG CHEN, Towards Resource Certified Image Processing Software, University of York, York, 2006.
- [26] ARIEL SHAMIR, Feature-Space Analysis of Unstructured Meshes, The Interdisciplinary Center, Herzliya

- [27] DORIN COMANICIU, PETER MEER, VISVANATHAN RAMESH, *Real-Time Tracking of Non-Rigid Objects using Mean Shift*, Princeton and Piscataway, USA, 2000.
- [28] MARK A. KECK JR., JAMES W. DAVIS, AMBRISH TYAGI, *Tracking Mean Shift Clustered Point Clouds for 3D Surveillance*, Ohio State University Columbus, USA, 2006.
- [29] YAKOV KESELMAN, EVANGELIA MICHELI-TZANAKOU, *Extraction and Characterization of Regions of Intrest in Biomediacal Images*, Departments of Computer Science and Biomediacal Engineering, Rutgers University, USA
- [30] ВЛАДИМИР РИСОЈЕВИЋ, Претраживање база слика на основу сличности региона, магистарска теза, Електротехнички факултет Баља Лука, 2006.
- [31] R. GONZALEZ, R. WOODS, S. EDDINS, *Digital Image Processing Using MATLAB*, Prentice Hall, 2004.