UNIVERZA V MARIBORU

FAKULTETA ZA ELEKTROTEHNIKO,

RAČUNALNIŠTVO IN INFORMATIKO

Božidar POTOČNIK

UPORABA SEGMENTACIJE PRI ANALIZI MEDICINSKIH SLIK

Magistrsko delo

MARIBOR, MAREC 1998

Avtor: Božidar Potočnik.

Naslov: Uporaba segmentacije pri analizi medicinskih slik.

UDK: 681.327.8:61

Ključne besede: digitalna obdelava slik, segmentacija, pragovne operacije, segmentacijske metode na osnovi robov, rast regij, razpoznavalni sistem, slika robov, klasifikacija, Houghova transformacija, razpoznavanje elips, ultrazvok, pegasti šum, jajčni mešički.

Število izvodov: 8.
Razmnoževanje: Laboratorij za sistemsko programsko opremo, FERI Maribor.

Na Univerzi v Mariboru,
Fakulteti za elektrotehniko, računalništvo in informatiko

je dne 16. marca 1998

Božidar POTOČNIK, dipl. inž. rač.

uspešno zagovarjal magistrsko delo

z naslovom:

UPORABA SEGMENTACIJE PRI ANALIZI MEDICINSKIH SLIK

pred komisijo:

red. prof. dr. Viljem ŽUMER, predsednik

izr. prof. dr. Damjan ZAZULA, mentor-član

izr. prof. dr. Franc SOLINA, komentor-član

red. prof. dr. Nikola GUID, član

ZAHVALA: Za strokovno pomoč v celotnem času magistrskega študija in še posebej pri izdelavi magistrskega dela se iskreno zahvaljujem svojemu mentorju prof. dr. Damjanu Zazuli, ki me je s svojimi idejami in nasveti pri raziskovanju vedno hrabril, vzpodbujal in pomagal. Za nasvete in komentarje se zahvaljujem tudi komentorju prof. dr. Francu Solini in prof. dr. Nikoli Guidu, prav tako pa se zahvaljujem tudi vsem sodelavcem iz Laboratorija za sistemsko programsko opremo, ki so mi z drobnimi nasveti velikokrat sneli iz ramen velika bremena.

Velika zahvala gre tudi prof. dr. Veljku Vlaisavljeviću iz Oddelka za človeško reprodukcijo in endokrinologijo, Splošna bolnišnica Maribor, kakor tudi Mauvis A. Gore z Instituta za reproduktivno biologijo, German Primate Centre Göttingen, ki sta s svojimi premnogimi nasveti ter nenehnim evalviranjem dobljenih rezultatov bistveno pripomogla k nastanku pričujočega dela.

Za pomoč v vseh letih mojega študija, predvsem za njeno izredno potrpežljivost, razumevanje in ljubezen, se zahvaljujem svoji dragi ženi Mateji. Zahvaljujem se tudi svojim staršem, ki so me pri mojem študiju vedno podpirali in bodrili.

Ne nazadnje se zahvaljujem tudi Ministrstvu za znanost in tehnologijo, ki mi je s štipendijo omogočilo raziskovalno delo.

UDK: 681.327.8:61

Ključne besede: digitalna obdelava slik, segmentacija, pragovne operacije, segmentacijske metode na osnovi robov, rast regij, razpoznavalni sistem, slika robov, klasifikacija, Houghova transformacija, razpoznavanje elips, ultrazvok, pegasti šum, jajčni mešički.

UPORABA SEGMENTACIJE PRI ANALIZI MEDICINSKIH SLIK

POVZETEK: Magistrska naloga opisuje uporabo segmentacije pri analizi medicinskih slik. Podrobno nas seznani s problemi, na katere naletimo pri konstruiranju takšnih segmentacijskih algoritmov. Vidimo tudi, da konstrukcija kontekstno odvisnih medicinskih segmentacijskih algoritmov ni možna brez tesnega sodelovanja ekspertov, kar se kaže predvsem pri evalviranju dobljenih segmentacijskih rezultatov. Konstruiranje takšnih algoritmov ima predpisano obliko, in sicer se je treba najprej seznaniti z medicinsko domeno, zatem izbrati ustrezno kontekstno neodvisno segmentacijsko metodo, na koncu pa je treba dobljene segmentacijske rezultate še ovrednotiti. Magistrsko delo podrobno opisuje te korake na primeru razvijanja segmentacijskega algoritma za avtomatsko določanje jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov.

V začetku dela nas tako najprej seznani z ultrazvokom, ultrazvočnimi pretvorniki in sondami ter njihovo uporabo v medicini, predvsem pri opazovanju sprememb v menstrualnem ciklusu. Ob tem dobimo osnovno znanje o poteku menstrualnega ciklusa, jajčnikih in jajčnih mešičkih. Ultrazvočne slike se izkažejo za izredno šumne (pegasti šum), zato so v nadaljevanju opisane tudi metode in operatorji za predobdelavo ultrazvočnih slik, še posebej operatorji za odstranjevanje peg. V najobsežnejšem delu je podan pregled obstoječih kontekstno neodvisnih segmentacijskih metod, pri čemer je uvedena razvrstitev le-teh v štiri skupine: pragovne operacije, segmentacijske metode na osnovi robov in regij ter ujemanje vzorcev. Simulacija desetih segmentacijskih metod iz različnih skupin, ki je opravljena na 50 umetnih slikah (približkih realnih ultrazvočnih slik), kot dve najuspešnejši metodi izdvoji pragovno operacijo OPTIMAL ter segmentacijsko metodo rasti regije UNION. Zatem sledi opis Houghove transformacije, ki je uporabljena za neposredno določanje položaja jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov. Jajčni mešički so aproksimirani z elipsami, kar pa se izkaže za neprimerno, saj je odstotek razpoznanih mešičkov le okrog 30 %.

Nadaljevanje je zato namenjeno razpoznavanju mešičkov po ustaljeni poti – s klasičnim razpoznavalnim sistemom. Podrobno je opisan razpoznavalni sistem "xultra" za avtomatsko določanje položaja jajčnih mešičkov, v katerem je združeno vso pridobljeno znanje o ultrazvoku, mešičkih in segmentacijskih metodah. Algoritem "xultra" je preizkušen na 30 realnih ultrazvočnih posnetkih jajčnikov, dobljeni rezultati pa so primerjani s pravilnimi rezultati, ki so jih posredovali eksperti – ginekologi. S tem algoritmom je odstotek razpoznavanja za okrog 40 % višji kot pri razpoznavanju s Houghovo transformacijo, pri iskanju le dominantnih jajčnih mešičkov pa je odstotek razpoznavanja 86,8 %. Zatem je podana kritična ocena algoritma "xultra", predlagane pa so tudi nekatere izboljšave. Na koncu magistrskega dela je opisan vpliv ekspertov s področja medicine pri konstruiranju segmentacijskih medicinskih algoritmov, naštete so prednosti in slabosti avtomatskega razpoznavanja ter stvari, na katere je treba še posebej paziti pri izbiri kontekstno neodvisnih segmentacijskih metod, ki naj bi bile vključene v segmentacijski algoritem za medicinske slike.

UDK: 681.327.8:61

Keywords: digital image processing, segmentation, thresholding, edge based segmentation methods, region growing, recognition system, edge image, classification, Hough transform, ellipse recognition, ultrasound, speckle noise, ovarian follicles.

APPLICATION OF SEGMENTATION FOR MEDICAL IMAGE ANALYSIS

ABSTRACT: In this work the application of segmentation for medical image analysis is described. Problems encountered when constructing such segmentation algorithms are described in detail. It can be seen that the construction of context-dependent segmentation algorithms without tight collaboration of experts is not possible, this can especially be seen when evaluating the segmentation results. The construction of these algorithms follows a well-defined principle: first the medical domain should be studied, then the appropriate context-independent segmentation method should be chosen, and finally, all the segmentation results obtained should be evaluated. This work describes in detail these steps dealing with the development of a segmentation algorithm for automated detection of follicles in ultrasound images of ovaries.

At the beginning, an introduction to ultrasound, ultrasound transducers and probes, and their application in medicine, especially for monitoring the variations in menstrual cycle, is given. At the same time, we get a basic knowledge about menstrual cycle, ovaries and follicles. Ultrasound images are very noisy (speckle noise), thus the methods and operators for preprocessing the ultrasound images, especially the operators for speckle noise reduction, are described. In the most extensive part of this work, an overview of existing context-independent segmentation methods is given, where the classification of these methods into four groups is introduced: thresholding, edge and region based segmentation methods, and pattern matching. A simulation of ten segmentation methods from different groups, which has been performed on 50 artificial images (approximations of real ultrasound images), pointed out thresholding operation OPTIMAL and region growing method UNION as the two most successful. A description of Hough transform follows, applied, afterwards, for direct follicle detection in ultrasound images of ovary. The follicles are approximated by ellipses, what proves to be unsuitable, because the recognition rate of follicles was only around 30 %.

The sequel of this work is devoted to the follicle recognition using classical approach – with classical recognition system. The recognition system "xultra" for automated detection of the follicles, in which the knowledge obtained about ultrasound, follicles and segmentation methods is gathered, is described in detail. The algorithm "xultra" was tested on 30 real ultrasound images of ovary, results obtained were compared to the correct ones which were provided by the experts – gynaecologists. With this algorithm the recognition rate is around 40 % higher than the recognition using Hough transform, however, when searching only the dominant follicles the recognition rate is 86,8 %. Then the critical evaluation of algorithm "xultra" is given and also some improvements are proposed. At the end of the work, an influence of medical experts on the construction of medical segmentation algorithms is described, some advantages and disadvantages of automated recognition are listed, and finally, the issues to which we should particularly pay attention when selecting the context-independent segmentation methods to be included in segmentation algorithm for medical images are listed.

Kazalo

1. Uvod

Računalnik je postal eno izmed najpomembnejših orodij danes. Računalniška obdelava in avtomatizacija dela je zajela že skoraj vsa področja življenja. Tako nam računalniki že dolgo časa aktivno pomagajo pri namiznem založništvu, z njimi lahko tvorimo in vzdržujemo velike podatkovne zbirke, ali pa načrtujemo oz. projektiramo razne arhitekturne in gradbeniške probleme, krmilimo in simuliramo industrijske procese, v zadnjem času pa se vse aktivneje uporabljajo tudi na področju zabave, če naštejemo le nekatera izmed mnogih področij njihove vsestranske uporabnosti. Računalnik ima posebno mesto tudi pri obdelavi slik, saj lahko z njegovo pomočjo na razne načine spreminjamo vsebino slik ter jih tako pripravljamo za učinkovitejšo nadaljnjo uporabo (npr. za tisk, shranitev v bazo slik, odkrivanje struktur ipd.).

Računalniki so digitalne naprave, saj v osnovi operirajo z biti, zato so tudi vsi podatki v njih predstavljeni digitalno, tudi slike. Zato imamo pri računalniški obdelavi slik dejansko opravka z digitalnimi slikami. Digitalne slike so ponavadi predstavljene kot matrike realnih števil oz. t. i. pikslov (možna predstavitev slike je tudi s šesterokotno mrežo pikslov). Piksli, ki so osnovni gradniki slik, nosijo, v primeru črno-belih oz. sivih slik, informacije o sivini posameznega koščka slike (informacije o jakosti oz. intenzivnosti svetlobe), v primeru barvnih slik pa vsebujejo informacije o barvi (npr. v predstavitvi barv z modelom RGB je piksel polje treh elementov, kjer vsak element nosi podatke o posamezni komponenti barve, in sicer rdeči, zeleni ter modri). V tako predstavljenih digitalnih slikah pa po nekih pravilih, ki so odvisna od vrste naloge, transformiramo piksle, jih združujemo v grupe, ali iz njih izločamo bistvene značilnosti, ali jih obdelujemo na kakšen drug način. Standardne metode oz. algoritmi za računalniško obdelovanje digitalnih slik so dobro znane in s teorijo podprte metode [7, 27, 31, 60, 63]. V zadnjem času se mnogo avtorjev, npr. [19, 28, 32, 37, 49], ukvarja z razpoznavanjem objektov v digitalnih slikah. Pri razpoznavanju objektov moramo konstruirati takšne postopke (algoritme), ki iz koraka v korak reducirajo podatke v slikah, na koncu pa se na osnovi nekih postavljenih kriterijev sprejme odločitev, ali je objekt razpoznan ali ne. Avtomatsko razpoznavanje objektov ali le izločanje objektov je v praksi zelo uporabno, saj nam zelo poenostavi rutinsko delo. V zadnjem času dobiva vse večjo veljavo tudi na področju medicine.

Eksperti s področja medicine se pri svojem delu srečujejo z zelo raznovrstnimi medicinskimi slikami. Če smo bolj eksaktni, moramo reči, da eksperti pri svojem delu uporabljajo raznovrstne medicinske naprave, ki omogočajo vizualizacijo oz. prikaz posameznih telesnih struktur na zaslon (ali kakšen drug prikaz). Šele iz prikaza na zaslonu pa lahko dobimo digitalno sliko. Slike lahko zajemamo tudi direktno iz medicinske naprave, vendar pa pri tem ponavadi digitaliziramo signale, ki so tako že namenjeni za krmiljenje zaslona, zato bomo v nadaljevanju oba omenjena termina enačili.

Zdravniki, predvsem diagnostiki, morajo pri postavljanju diagnoz analizirati kompleksne medicinske slike in iz njih izluščiti ustrezne podatke. Takšno analiziranje je po eni strani zelo rutinsko, npr. pregled ogromne množice slik, kjer je treba obkrožiti oz. izmeriti določeno lastnost neke telesne strukture (npr. jajčne mešičke v jajčnikih), hkrati pa je tudi časovno zelo zahtevno in naporno. Slednje velja še posebej v primerih, ko je vsaka pomota lahko usodna ter je zato treba opraviti analizo še posebej skrbno (npr. odkrivanje rakastih celic in tumorjev). Čeprav predstavlja ročno analiziranje takšnih slik za zdravnike rutino, pa to še ne pomeni, da pri svojem delu ne delajo napak. Prav zaradi prepričanosti o svojem znanju in želje po čim hitrejšem analiziranju teh slik vnesejo pri svojem delu veliko subjektivnih napak (npr. zaradi površnega določanja meje pljuč je izmerjeni volumen napačen). Problem časovne kompleksnosti ročne analize se še stopnjuje, ko preidemo iz analize 2D medicinskih slik na analizo 3D ali celo na 4D medicinskih slik.

Razbremenitev ekspertov pri njihovem rutinskem delu je glavni razlog, ki je pospešil razvoj mnogih aplikacij za analizo medicinskih slik. Mnogokrat vlada zmotno prepričanje, da z uvedbo računalniške obdelave in analize medicinskih slik zdravnik ne bo več potreben. Vendar temu ni tako, saj je na koncu vsake računalniške analize nujno potreben ekspert, ki zna interpretirati dobljene rezultate in jih tudi uporabiti pri postavljanju diagnoz. V literaturi lahko najdemo mnogo medicinskih aplikacij, ki s pristopom računalniške obdelave slik rešujejo zelo različne medicinske probleme. Tako smo v [8] zasledili aplikacijo za merjenje glave zarodka v ultrazvočnih posnetkih, v [59] je opisan sistem za avtomatsko sledenje aortam v kardiovaskularnih magnetno resonančnih slikah MRI (angl. Magnetic Resonance Imaging), v [68] so se lotili avtomatičnega določanja kontur eksterne lobanjske lupine v slikah CT (angl. Computed Tomography), v [34] so se ukvarjali s štetjem mišičnih vlaken v slikah, dobljenih z elektronskim mikroskopom, v [40] pa z izračunom prekatnega volumna v ultrazvočnih slikah lobanje. Našteli smo le nekaj takšnih aplikacij. Videli smo, da res pokrivajo zelo različna medicinska področja, hkrati pa imajo vse zelo različno zgradbo in pristop k obdelovanju slik. Vsem tem in tudi ostalim medicinskim aplikacijam pa je skupno, da v določenem trenutku obdelovanja opravijo segmentacijo slike. Segmentacija slik pomeni grupiranje oz. združevanje pikslov v množice, in sicer glede na neke lastnosti oz. kriterije (npr. sivino pikslov, teksturo).

Segmentacija je kritičen faktor pri obdelovanju in analizi slik. Za medicinske aplikacije je segmentacija zajetih slik nujen korak, saj nam omogoča kvantitativno merjenje podatkov, kot so npr. ploščina ali volumen posameznih struktur [59]. Zdi se, da je segmentacija še posebej težavna v primerih medicinskih slik, to pa predvsem zaradi variacij in raznih situacij, ki se tukaj pojavljajo [68]. Problem konstrukcije neke univerzalne segmentacijske metode, uporabne za vse medicinske slike, je še toliko večji, ker imamo v medicini opravka z zelo različnimi tipi slik, ki se v določenih primerih tudi bistveno razlikujejo med seboj. Tako se v biomedicinskih aplikacijah srečujemo s slikami MRI, slikami, dobljenimi z emisijsko tomografijo in projekcijo z X-žarki, slikami CT, ultrazvočnimi slikami, slikami, dobljenimi z impedančno tomografijo in električnim ter magnetnim valovanjem, slikami, dobljenimi s posebnimi fotografskimi metodami [67]. Glavni namen segmentacije medicinskih slik je redukcija podatkov, ki jo v slikah opravimo z inteligentnim modificiranjem slik z nizkih nivojev (pikslov) v višje nivojske reprezentacije (regije). Moramo se namreč zavedati, da slike v splošnem vsebujejo ogromno količino podatkov, izmed katerih je mnogo redundantnih. V večini biomedicinskih aplikacij je končni cilj segmentacije tvorba pravil za razvrščanje dobljenih regij, ta pravila pa pomagajo ekspertom pri postavljanju diagnoz.

Konstruiranje segmentacijskih algoritmov, posebej še algoritmov, namenjenih za medicinske probleme, je izredno zahtevna naloga. Pri razvijanju takšnega algoritma moramo upoštevati tip slik, ki jih obdelujemo. Zato je treba podrobno analizirati njihove lastnosti, posebnosti, kakšen šum se lahko pojavi, kakšna je resolucija slik oz. kakšna je največja ločljivost itd. Naslednja stvar, ki jo moramo analizirati, je, kako veliki so oz. kakšen delež ploščine glede na celotno sliko zavzemajo objekti, ki jih želimo segmentirati. S to analizo želimo odgovoriti na vprašanje, ali imamo opravka z majhnimi ali velikimi objekti. To je pomembno zato, ker pri velikih objektih lahko dopustimo, da segmentacijski algoritem razpoznava objekte z določeno napako (npr. nekaj deset pikslov), pri iskanju majhnih objektov pa si takšne napake ne smemo privoščiti. Nato moramo v slikah poiskati še določene lastnosti, skupne vsem pikslom, ki tvorijo isti objekt. Na koncu se moramo le še odločiti, s kakšnim postopkom bomo združevali piksle.

Večino odgovorov na zgoraj postavljena vprašanja lahko poda strokovnjak s področja digitalne obdelave slik, le o tipičnih lastnostih, ki ločijo iskani objekt od okolice, mora odgovore vedno posredovati ekspert. Pri tem mislimo na odgovore, kot so npr.: jajčni mešiček je na ultrazvočni sliki vedno temne barve in se nahaja v jajčniku, ali pa npr.: tumor enostavno odkrijemo v barvnih slikah MRI, če opazujemo le modri spekter barve. Iz tega lahko sklenemo, da je kvaliteten segmentacijski algoritem za medicinske slike lahko le rezultat kvalitetnega timskega dela ekspertov s področja obdelava slik in s področja medicine.

Ko je segmentacijski algoritem, oz. če rečemo širše, razpoznavalni sistem skonstruiran, moramo preveriti njegovo učinkovitost in uspešnost ter ga zatem še primerjati s podobnimi algoritmi. Objektivna evaluacija teh algoritmov na veliki množici kliničnih podatkov (medicinskih slik) je eden izmed pomembnih korakov pri ugotavljanju pravilnosti in klinične uporabnosti algoritma. Le malo raziskovalcev je opravilo takšno evalvacijo na veliki množici podatkov. Mnogi so primerjali svoje algoritme le na fantomih (umetno tvorjenih slikah, kjer vladajo podobne razmere kot na originalnih) ali opravili in-vitro študije, ki pa so tipično idealistične reprezentacije realnih podatkov. Predvideti učinkovitost na realnih podatkih, ki temeljijo na takšnih rezultatih, je težko [8]. Celo tisti, ki preverjajo svoje algoritme na realnih kliničnih podatkih, uporabljajo različne kriterije in statistike, kar pa oteži primerjavo učinkovitosti med algoritmi. Razlogi, da je evalvacija segmentacije medicinskih slik tako težka, so naslednji [8]:

- pomanjkanje definiranih zlatih standardov. Pri medicinskih slikah je segmentacija, ki jo je opravil ekspert opazovalec, edini standard, ki je na voljo. Le-ta pa ne more biti zlati standard, ker je preveč popačena zaradi subjektivnosti, ki jo vnaša ekspert (inter in intra opazovalčeva variabilnost);

- problemi pri definiranju metrik. Te metrike je težko definirati predvsem zaradi kompleksne multidimenzionalne narave segmentacijskih podatkov;

- pomanjkanje standardiziranih statističnih protokolov. Sklepi in zaključki o učinkovitosti algoritmov zahtevajo statistično analizo, ki uporablja standardne protokole;

- neprijetnega in časovno zahtevnega zbiranja podatkov. Zbiranje velikega števila podatkov, z ročno definiranimi segmentacijskimi rezultati, je zelo zahtevna in hkrati ekspertom dolgočasna naloga.

Šele ko opravimo vse potrebne analize učinkovitosti medicinskega segmentacijskega algoritma (razpoznavalnega sistema) in smo hkrati zadovoljni z rezultati, ga lahko damo v klinično uporabo (kot orodje, ki aktivno pomaga zdravnikom).

V magistrskem delu bomo opisali konstruiranje segmentacijskega algoritma za medicinske slike. V 2. poglavju se bomo srečali z ultrazvočnimi slikami kot eno izmed tipičnih medicinskih domen. Seznanili se bomo z osnovnimi fizikalnimi lastnostmi ultrazvoka, zatem pa bomo opisali njegovo uporabo v medicini. Predstavili bomo razne vrste medicinskih ultrazvočnih prikazov in s tem področja uporabe. V nadaljevanju bomo prikazali čisto konkreten primer medicinske uporabe ultrazvoka, in sicer kot sredstvo za opazovanje dogajanja znotraj jajčnikov (sprememb v menstrualnem ciklusu). Poglavje bomo zaključili z analiziranjem motenj (šuma), ki se lahko pojavijo v ultrazvočnih slikah, opisali pa bomo tudi filtre in operatorje, s katerimi lahko takšne motnje izločimo. V 3. poglavju bomo opravili kompleksno analizo obstoječih segmentacijskih metod. Vse segmentacijske metode bomo razvrstili v 4 različne skupine, poiskali skupne značilnosti za posamezno skupino, nekaj izmed segmentacijskih algoritmov pa bomo tudi podrobno opisali. Iz tega čisto teoretičnega dela bomo prešli na praktičen del, kjer bomo izbrali nekaj teh metod, jih preizkusili na umetno tvorjenih slikah (nekakšnih fantomih za ultrazvočne posnetke) ter jih na osnovi izbranih statistik tudi razvrstili po uspešnosti. V 4. poglavju se bomo lotili problema razpoznavanja jajčnih mešičkov, pri čemer bomo kot segmentacijsko metodo uporabili Houghovo transformacijo, ki v zadnjem času postaja vse popularnejša in uporabnejša metoda. V 5. poglavju bomo v segmentacijskem algoritmu za ultrazvočne posnetke jajčnikov združili pridobljeno znanje o ultrazvoku, jajčnih mešičkih ter segmentacijskih metodah (poglavji 2 in 3). Hkrati bomo tudi odpravili pomanjkljivosti, ki smo jih zasledili pri segmentiranju s Houghovo transformacijo (poglavje 4). Konstrukcijo tega algoritma (oz. razpoznavalnega sistema) bomo opisali korak za korakom, opozarjali na posebej pomembne stvari, nenehno pa bomo tudi poudarjali pomen sodelovanja z ekspertom – ginekologom. Razviti segmentacijski algoritem bomo zatem preizkusili na množici realnih ultrazvočnih posnetkov, dobljene rezultate pa primerjali z rezultati ročnega segmentiranja, ki ga je opravil ekspert. Delo bomo zaključili s sklepom, kjer bomo opravili kratko primerjavo obeh segmentacijskih algoritmov iz poglavij 4 in 5, zatem pa bomo podali še nekatere sklepe, ki so se oblikovali skozi celoten študij področja segmentacije medicinskih slik.

2.

Ultrazvok in medicina

Ultrazvok se uporablja v medicinski diagnostiki že od sredine petdesetih let tega stoletja. Za začetnika ultrazvočne diagnostike štejemo Iana Donalda, zdravnika iz Velike Britanije, ki je prvi objavil temeljne principe ehografije. Ultrazvok se danes uporablja za zelo raznovrstne medicinske probleme (ginekologija, porodništvo, analiza srca itd.) in ga zato lahko upravičeno štejemo kot eno izmed najpomembnejših orodij medicine.

V tem poglavju bomo predstavili ultrazvok in njegovo uporabo. Tako bomo v podpoglavju 2.1 opisali bistvene fizikalne lastnosti ultrazvoka in izvedbe ultrazvočnih naprav, ki se uporabljajo pri medicinski diagnostiki. V podpoglavju 2.2 se bomo nato srečali s problemom opazovanja jajčnih mešičkov kot z eno izmed pomembnih aplikacij ultrazvoka v medicini. Poglavje pa bomo zaključili čisto v smislu digitalne obdelave slik, saj bomo opisali probleme, s katerimi se lahko srečamo pri posnetkih, dobljenih iz ultrazvočnih naprav, ter kako jih lahko s posebnimi operatorji za filtriranje digitalnih slik omilimo oz. odstranimo.

2.1 Ultrazvok v medicini

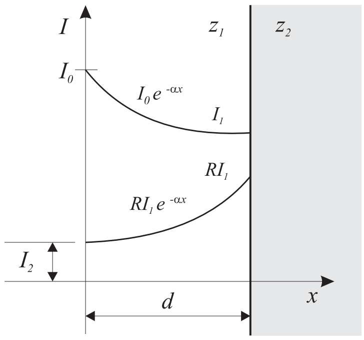
Zvok povzroča vibriranje delcev zraka ali nekega drugega medija, skozi katerega se razširja. Gibanje delcev lahko prikažemo s periodično funkcijo različnih frekvenc. Če so frekvence gibanja delcev v slišnem področju, to je med 16 Hz in 16 kHz do 20 kHz, potem imenujemo takšno gibanje zvok, ultrazvok pa predstavlja enako mehansko vibriranje delcev zraka, vendar v področju nad 20 kHz. V medicinski diagnostiki se uporablja ultrazvok, kjer se frekvence tipično gibljejo od 1 MHz do 15 MHz (klasičen ultrazvok), v posameznih primerih pa celo do 100 MHz [65].

Hitrost širjenja ultrazvoka, kakor tudi zvoka, je različna za različne snovi in je odvisna od gostote snovi, skozi katero se razširja (npr. zvok se skozi vodo pri 20^o C pri normalnem tlaku širi s hitrostjo 1480 m/s, skozi zrak 331 m/s, mišice 1585 m/s, kri 1570 m/s in aluminij 6400 m/s) [65].

Če se posamezni delci snovi (molekule ali skupki molekul) premikajo vzdolž smeri širjenja, potem takšnemu valu pravimo longitudinalni ali vzdolžni val. Longitudialni val se lahko razširja skozi poljubno snov, ki je vsaj malo stisljiva. V medicinski uporabi ultrazvoka prevladuje longitudinalni način širjenja valov (poznamo še t. i. transverzalni oz. prečni način širjenja).

Ko se ultrazvok širi po homogeni snovi z določeno akustično impedanco z_1 in ko na svoji poti naleti na snov z drugačno akustično impedanco z_2, pride na tem mestu do odboja, del vala pa se lomi ter nadaljuje pot. Koeficient odboja R je odvisen od vrste snovi oz. njene akustične impedance ter ga izračunamo kot $\left(\frac{z_1-z_2}{z_1+z_2}\right)^2$. V zraku je koeficient odboja 0,0004 Ns/m^3, v vodi 1,48 Ns/m^3, v mišicah 1,70 Ns/m^3 ter v aluminiju 17,4 Ns/m^3 [65].

Velja tudi, da se jakost ultrazvočne energije pri prehodu skozi homogeno snov eksponentno zmanjšuje glede na prehojeno pot, in sicer po zakonu $I_0\,\mathrm{e}^{-\alpha x}$, kjer je I_0 začetna energija, α je koeficient dušenja in x je prehojena pot [65]. Na sliki 2.1 je prikazan odboj ultrazvočnega vala in eksponentno zmanjševanje njegove energije. Jakost ultrazvoka se izraža z energijo po enoti površine. V medicinski diagnostiki se energija meri v milivatih na kvadratni centimeter (mW/cm^2). V terapevtske namene pa se uporablja tudi t. i. močni ultrazvok [56].



Slika 2.1: Dušenje in odboj ultrazvočne energije

Vse opisane lastnosti ultrazvoka so izredno pomembne pri njegovi uporabi v medicinski diagnostiki, saj se tukaj meri velikost odbite energije. V medicini se uporablja oddajnik hkrati tudi kot sprejemnik ultrazvočnih valov. Sprejete (odbite) valove prikažemo na zaslonu.

Sposobnost razlikovanja detajlov na ultrazvočnem prikazu je odvisna od ločljivosti oz. razdvojne moči, pri čemer je ločljivost definirana kot najmanjša oddaljenost, na kateri še lahko vidimo ločeno dva sosednja elementa opazovane strukture. Možno je pokazati, da mora biti ta oddaljenost (d) večja od valovne dolžine λ, torej $d \geq \lambda = \frac{c}{f}$ (npr. hitrost valovanja c je v tkivu 1500 m/s in frekvenca valovanja f je 2 MHz, potem bomo lahko razlikovali detajle, večje od 0,75 mm). To pravilo velja v primeru, ko ima snop sevanja manjši premer od λ (ultrazvok se v praksi ne pojavlja kot en sam žarek, temveč kot snop), v nasprotnem primeru pa določa ločljivost širina snopa. Zaradi teh razlogov se za preiskavo manjših organov uporablja ultrazvok višjih frekvenc, za večje pa ultrazvok nižjih frekvenc. Prav tako se je treba zavedati, da je dušenje ultrazvoka pri višjih frekvencah večje, saj koeficient dušenja α raste s frekvenco. Zato moramo, če se želimo izogniti prevelikemu dušenju signala, za preiskave organov na večjih globinah organizma uporabiti ultrazvok nižjih frekvenc.

V porodništvu in ginekologiji se uporablja ultrazvok s frekvencami od 2 MHz do 5 MHz [65]. Šele v poznih 80-ih letih so razvili tudi pretvornike, primerne za medicinsko uporabo, ki vibrirajo z zelo visokimi frekvencami (od 30 do 100 MHz). Takšen ultrazvok uporabljamo za pregled oči (mnogo struktur v očesu je premajhnih za klasičen ultrazvok, CT ali MRI), za pregled kože in intravascularno vizualizacijo [45].

2.1.1 Ultrazvočni pretvorniki

Ultrazvok najdemo v naravi izredno redko. Najbolj znan primer so netopirji, ki uporabljajo ultrazvok za koordiniranje svojega gibanja. Ultrazvok lahko generiramo tudi umetno, npr. s ploščico, ki niha s frekvenco v področju ultrazvoka.

Če bi neko ploščico premikali pravokotno na njeno površino z visoko frekvenco v področju ultrazvoka, bi dobili ultrazvočno vibriranje oz. tvorjenje ultrazvočnega vala. Takšno nihanje lahko ustvarimo s piezoelektričnimi in magnetostrikcijskimi materiali. Piezoelektrični materiali so primerni za uporabo v medicini, ker lahko pošiljajo zelo visoke frekvence ultrazvoka, večje tudi od 1 GHz, medtem ko se magnetostrikcijski materiali uporabljajo najpogosteje v področju nižjih frekvenc od 10 do 100 kHz, večinoma ne v medicinske namene [65].

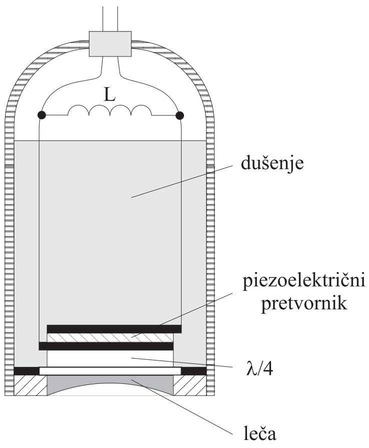
V medicinski ultrazvočni tehniki se uporabljajo sintetični in keramični piezoelektrični materiali, kot so barijev titanat ($BaTiO_3$), svinčev cirkonat ($PbZrO_3$), svinčev titanat ($PbTiO_3$) in drugi. Piezoelektrični materiali imajo sposobnost piezoelektričnega efekta (angl. pressure-electric effect). Piezoelektrični efekt pomeni, da posamezni kristali pod vplivom mehanskega pritiska tvorijo na svoji površini električne potenciale. Po drugi strani pa z obrnjenim piezoelektričnim efektom nastane ultrazvok. Torej, če na kristal privedemo električni impulz, potem to privede do hitrih sprememb debeline kristala, ki se krči in širi ter tvori ultrazvočne valove. Hkrati pa je ultrazvok sposoben na polih takšnega kristala povzročiti tudi električno napetost. Na ta način je krog sklenjen: en in isti kristal – pretvornik lahko služi ne samo kot oddajnik valov, ampak tudi kot sprejemnik odbitih valov [56].

Piezoelektrična ploščica, ki jo najpogosteje izdelajo v obliki diska ali pravokotne ploščice, lahko s spreminjanjem napetosti (sile) na njeni površini spremeni svojo debelino ($\delta \pm \triangle\delta$). Če piezoelektrično ploščico od zunaj vzbudimo z napetostnim impulzom, bo začela vibrirati ter se bo mehansko iznihala po krajšem oz. daljšem času, odvisno od koeficienta mehanskega dušenja. V medicinskem diagnostičnem ultrazvoku želimo imeti impulze s kratkim trajanjem in kratkotrajno vibriranje ploščice, ki ne bi pokvarilo ločljivosti dobljenega prikaza. Če impulz traja manj kot 1 μs, tkivo pa leži na globini, večji od 5 cm, potem pretvornik porabi le 2 % časa za pošiljanje impulza, ostalih 98 % pa za sprejem odmevov [56].

Mehanska frekvenca f_0, na kateri bo takšna ploščica vibrirala, je odvisna od debeline ploščice δ in od hitrosti širjenja ultrazvoka c skozi material ploščice. Frekvenco f_0 določimo kot $\frac{c}{2\delta}$ [65].

2.1.2 Ultrazvočne sonde

Ultrazvočni pretvornik (ploščica) se uporablja hkrati kot oddajnik in kot sprejemnik ultrazvoka. Ploščica iz piezoelektričnega materiala, ki je ponavadi okrogle oblike, je najpomembnejši del ultrazvočne sonde (slika 2.2). Preko elektrod, ki so priključene na ploščico, lahko dovajamo kratkotrajne napetostne impulze. Ploščica, ki jo vzbudimo s takšnim impulzom, oddaja ultrazvok na obe strani, pri tem je zadnja stran, ki gleda v notranjost sonde, napolnjena z materialom za dušenje (npr. s sintetično smolo). Na sprednji strani se lahko ultrazvok prosto širi, vendar pa moramo ta snop pravilno usmeriti oz. fokusirati. Za to nalogo se uporabljajo plankonkavne oz. bikonkavne zbiralne leče (zvok se širi skozi lečo hitreje kot skozi okoliški medij). Včasih dodamo v sondo še ploščico debeline $\lambda/4$, ki dobro prepušča ultrazvok, namenjena pa je za utrjevanje piezoelektričnega kristala in boljše prilagajanje akustičnih impedanc. Za generiranje ultrazvočnega impulza se uporabljajo naprave, ki lahko generirajo napetostni impulz od 150 V do 500 V v času le nekaj deset nanosekund [65].



Slika 2.2: Ultrazvočna sonda

2.1.3 Različni tipi ultrazvočnih prikazov

Ultrazvočne valove, ki se odbijajo od posameznih struktur v telesu, zaznamo s sondo in jih zatem na neki način prikažemo. V medicini imamo opravka z raznimi vrstami prikazov in s tem tudi različnimi ultrazvočnimi napravami. Poglejmo si nekatere.

A-prikaz

A-prikaz (angl. amplitude) je enodimenzionalen prikaz. Najenostavnejša ehoskopska naprava, ki daje A-prikaz, je ehoencefalograf. Najpogosteje se uporablja v neurologiji za preiskavo glave.

A-prikaz dobimo tako, da iz sonde pošljemo kratkotrajen ultrazvočni impulz. Ko ta usmerjen val naleti na strukture različnih akustičnih impedanc, se val na tej meji odbije (del potuje naprej). Ta odbita energija pride nazaj do piezoelektričnega pretvornika v sondi, kjer se ta ultrazvočni signal zaradi piezoelektričnega efekta ploščice pretvori v napetostni signal. Dobljeni napetostni signal predstavlja odmev, ki se po ojačanju prikaže na zaslonu katodne cevi kot vertikalni odmik.

B-prikaz

B-prikaz (angl. brightness) je dvodimezionalen prikaz. Lahko je statičen ali dinamičen.

1. Pri statičnem B-prikazu se ultrazvočni impulzi pošiljajo na enak način kot pri A-prikazu, in sicer v enem snopu, pri tem pa se odmevi prikazujejo vzdolž te smeri na zaslonu s svetlimi točkami, ki so tem bolj svetle, čim večja je amplituda odmeva. Zatem premaknemo sondo pravokotno glede na opazovani predmet (smer x) ter ponovimo postopek. Postopek ponavljamo, dokler ne zgeneriramo celotne slike. Statičen prikaz se danes redko uporablja, predvsem zato, ker se slika z ročnim premikanjem sonde tvori prepočasi. Kljub temu pa se lahko posamezni detajli na takšen način mnogo bolje preučijo.

2. Dinamičen prikaz imenujemo tudi dvodimenzionalen prikaz v realnem času. V tem primeru je za tvorjenje dvodimenzionalne slike treba premikati sondo na takšen način, da v sekundi dobimo vsaj 25 do 30 polnih slik. To lahko dosežemo na dva načina: ali z mehanskim prebiranjem z eno ali največ štirimi ultrazvočnimi sondami ali pa z nizom sond, postavljenih v linijo, kjer je število teh elementov med 50 in več sto elementi.

   a. Mehanske rotirajoče in nihajoče sonde. Mehanski prebirniki tvorijo sektorsko sliko (trikotna oblika slike). Primer takšnega prebirnika je ultrazvočna sonda z enim pretvornikom, ki lahko niha kot zvon. V primeru sonde z enim samim nihajočim pretvornikom se ultrazvočni snopi neenolično emitirajo, to pa zaradi tega, ker kotna hitrost takšne sonde ni konstantna (ko pretvornik zaniha do krajnih položajev, je kotna hitrost 0). Zato raje damo več pretvornikov na rotirajoči cilinder (od 2 do 4 pretvornike), ki imajo vsi konstantno kotno hitrost in oddajajo enolične snope. Kakršnakoli je že struktura sonde (ali zvon ali rotirajoče sonde), vedno so pretvorniki položeni v tekočino, npr. v olje ali vodo, ki omogoča prenos ultrazvočne energije s čim manj odboja in izgube na robnih področjih [65]. Mehanske sektorske sonde imajo pomanjkljivost [61], da je kvaliteta slik bližnjih področij zaradi reverberacije (angl. reverberation) slaba (podpoglavje 2.1.4).

   b. Polja pretvornikov. Polja pretvornikov so lahko linearne ali kvadratne oblike, lahko pa so pretvorniki organizirani koncentrično v obliki kroga. Najpogostejša so linearna polja, ki so tipično široka 1 cm in dolga od 10 do 15 cm s 64 do 512 elementi [61]. Te pretvornike naredimo iz enega keramičnega traka, ki ga z utori razdelimo na piezoelektrične ploščice. Pretvorniki se vključujejo (prižigajo) eden za drugim ter brez mehanskega premikanja tvorijo paralelno premikanje ultrazvočnega snopa. Prižiganje posameznih pretvornikov v oddajni in sprejemni smeri izvaja multipleksor. Na ta način dobimo pravokotni format slike [65].

M-prikaz

M-prikaz (angl. move) se uporablja za prikaz karakterističnih gibov srčnih struktur (določimo lahko npr. dimenzijo aorte). Pregled se izvaja tako, da postavimo sondo med 3. in 4. rebro. Ultrazvoči snop se pomika gor in dol v smeri y, s tem dobimo prvo dimenzijo prikaza, druga dimenzija je čas. Tudi v tem načinu se odmevi beležijo kot svetle točke na katodni cevi. M-prikaz daje zelo koristne informacije o delovanju srca, zato je znan pod imenom ehokardiografija.

Ultrazvočne naprave na osnovi Dopplerjevega efekta

S pomočjo ultrazvočnih naprav na osnovi Dopplerjevega efekta (Dopplerjev efekt: če se izvor približuje sprejemniku, je sprejeta frekvenca višja od oddane frekvence signala, če se izvor oddaljuje, velja obratno), lahko iz sprememb frekvenc sklepamo o hitrosti kroženja krvi, ob znanem preseku krvnih žil pa še o pretoku krvi. Tako lahko diagnosticiramo spremembe v krvnih žilah (zoožitev oz. zamašitev), oskrbo kostnega mozga s krvjo in nepravilnosti perifernega krvnega toka. Te naprave se uporabljajo tudi v porodništvu, npr. za odkrivanje bitja srca 12-tedenskega zarodka.

Pomanjkljivost Dopplerjevih naprav s kontinuiranim oddajanjem ultrazvoka je nezmožnost določanja mesta, kjer se nahaja gibajoči se objekt (v tem primeru kri v žili). Ta problem rešujemo z uporabo impulznih Dopplerjevih naprav, ki oddajajo ultrazvok le v kratkih impulzih. Prikaz impulznega dopplerja ponavadi kombiniramo z B-prikazom (prikaz lokacije združimo s posebnim diagramom, ki daje prikaz o hitrosti kroženja krvi v odvisnosti od časa). Impulzni doppler pa lahko tudi direktno dodamo na 2D prikaz, in sicer z različnimi barvami (npr. rdeča oz. oranžna barva za kroženje krvi v eni smeri in modra barva za nasprotno smer).

3D ultrazvočne naprave

Klasični 2D ultrazvok ima kar nekaj omejitev [15]. Te so:

- subjektivnost konvencionalnega pregleda. Ta se pojavi zaradi različnih izkušenj in znanja diagnostika pri manipuliranju z ultrazvočno sondo. Pri pregledu diagnostik dejansko miselno transformira 2D sliko v 3D strukturo tkiva (sam prilagaja pregled) in nato postavlja diagnozo;

- težko je locirati tanke 2D ultrazvočne slikovne ravnine v organih;

- težko je po že opravljenem pregledu reproducirati določeno lokacijo v sliki;

- pacientova anatomija oz. orientacija včasih omejuje kot, pod katerim lahko pošljemo ultrazvočni snop. Zaradi tega ne moremo doseči optimalnih slikovnih ravnin, potrebnih za diagnozo.

Vse te probleme lahko odpravimo z vpeljavo 3D ultrazvoka. 3D ultrazvočni pogled lahko dobimo tako, da s klasičnimi sondami ročno zajemamo 2D ultrazvočne slike, podatek o tretji dimenziji pa dobimo iz dodatnih naprav, ki merijo odmik sonde od neke referenčne točke. 3D ultrazvočne slike pa lahko dobimo tudi s 3D sondami.

2.1.4 Napake v ultrazvočni diagnostiki

Pri postavljanju diagnoz na osnovi ultrazvoka se lahko pojavijo subjektivne in objektivne napake [56]. Subjektivne napake, kot so npr. napačna identifikacija posameznega dela organa, so rezultat neizkušenosti diagnostika (pojavijo se lahko celo optične iluzije) in se dajo odpraviti. Bolj pomembne so objektivne napake, ki jih ne moremo popolnoma odstraniti, hkrati pa moramo na njih vedno računati. Objektivne napake bomo opisali na primeru 2D ultrazvočnih posnetkov. V teh posnetkih se pojavljajo neobstoječe strukture, ki jih v organih ne najdemo. Neobstoječe strukture se pojavijo zaradi:

- pegastega šuma (angl. speckle noise). Pegasti šum se pojavi, ko so strukture v določenem organu (telesa) premajhne, da bi jih slikovni sistem (ultrazvočna naprava), še lahko zaznal. Zato povzročijo interferenco med različnimi deli sprejetega vala, ki se odbijejo od regije objekta [1];

- reverberacije. Ta pojav nastane z odbijanjem odbitih valov od površine sonde in je pogostejši pri visokofrekvenčnih pretvornikih. Tako nastali ultrazvočni valovi ponovno potujejo v globino k tkivu in od tega tkiva se še drugič odbijejo. Ultrazvočni valovi lahko večkrat potujejo med sondo in tkivom (reflektorjem) ter dajo ta neželeni efekt;

- lomljenja ultrazvočnega snopa.

Ob neobstoječih strukturah se lahko v ultrazvočnih posnetkih pojavi tudi sprememba odtenka sivine. Ta se pojavi pri t. i. distalnem ojačanju. Pri tem se oddaljenemu tkivu, ki se nahaja pod nekim tkivom dobrih akustičnih lastnosti (t. i. akustično okno), zaradi ojačanega odmeva spremeni odtenek sive skale, kar se na prikazu vidi kot zelo svetel del. V določenih primerih se spremeni tudi izgled tkivne strukture (tekstura) in ne le odtenek. V nasprotnem primeru pa lahko posamezna tkiva toliko oslabijo – zmanjšajo amplitudo signala, da oddaljeno tkivo postane bistveno temnejše in nejasno. Ta pojav imenujemo senčenje in lahko povzroči, da posamezne distalne strukture postanejo popolnoma nevidne.

2.2 Medicinski problem iz ginekologije

V prejšnjem podpoglavju smo že omenili, da se ultrazvok lahko uporablja pri zelo različnih medicinskih problemih. V tem podpoglavju si bomo ogledali konkreten primer uporabe ultrazvoka v ginekologiji, in sicer za opazovanje rasti jajčnih mešičkov znotraj jajčnikov. Pred tem pa bomo na kratko opisali bistvene spremembe, ki se dogajajo znotraj jajčnikov in so temeljne za nadaljnje razumevanje.

2.2.1 Jajčniki in jajčni mešički

Jajčnika (angl. ovary) sta ploščati, ovalni, približno 3 do 4 cm dolgi tvorbi, belkaste neravne površine in sta vpeta v zadnjo stran podvojka ob strani maternice (vsak na svojo stran – levo in desno). V skorji jajčnika so številni jajčni mešički (angl. follicle) v različnih fazah zorenja. Pramešički so najbolj nezreli. Tvorijo jih jajčne celice, ki jih obdaja enoslojna ploščata vrhnjica (epitel oz. vrhnja plast sluznice). Najzrelejši so Grafovi mešički, ki se ciklično pojavljajo vsak mesec, praviloma po eden pri spolno zrelih ženskah. V sredini jajčnika pa so krvne žile, mezgovnice in živčne niti [36].

V zreli dobi, ki traja od pubertete do menopavze, si zaporedoma sledijo maternični in ovarialni ciklusi. Za začetek posameznega ciklusa štejemo dan, ko nastopi mesečno perilo. Od takrat naprej lahko opazimo v jajčniku razvoj enega izmed mešičkov, ki približno v 14 dneh dozori (Grafov mešiček). Zrel mešiček lahko zaznamo na površini jajčnika kot okroglasto izboklino velikosti graha. Mešiček je napolnjen s tekočino, obdajata pa ga zunanja in notranja ovojnica, proti tekočini pa še vrhnjica. Skupek celic vrhnjice sega v notranjost zrelega mešička v obliki grička, ki vsebuje jajčno celico, obdano s prosojno ovojnico. Jajčna celica je največja celica človeškega telesa in meri 0,2 mm. Ko mešiček dozori, poči, jajčece pa se osvobodi (zato mešičkom tudi rečemo mehurčki). Temu dogodku rečemo ovulacija. Na mestu razpočenega mešička nastane po kratkotrajni manjši krvavitvi t. i. krvavo telo, zatem pa še rumeno telo, ki izloča hormon progesteron. V primeru, ko ni prišlo do nosečnosti, približno 14 dni po ovulaciji rumeno telo zakrni, na njegovem mestu pa nastane brazgotina – belo telo. Po propadu rumenega telesa prične rasti v jajčniku nov mešiček, s tem pa se začne nov ciklus.

Dogajanja znotraj jajčnikov smo opisali res zelo v grobem. V resnici poteka v jajčniku pravi boj za prevlado med mešički. Ta boj se kaže v nenehnem spreminjanju koncentracij raznih hormonov v telesu, dokler posamezen mešiček ne prevlada in ne uniči ostalih. Lahko se tudi zgodi, da v posameznem ciklusu ne pride do ovulacije [36, 41, 47, 56].

2.2.2 Ultrazvok in menstrualni ciklus

Spremljanje sprememb (podpoglavje 2.2.1), ki se dogajajo v menstrualnem ciklusu, je koristno za vse pacientke, posebno pa še v primerih zakonske neplodnosti. Z razvojem ultrazvočnih pretvornikov in samih ultrazvočnih naprav lahko te spremembe opazujejo tudi s pomočjo ultrazvoka. Z ultrazvokom spremljajo dogajanja v maternični sluznici ter dozorevanje jajčnih mešičkov v jajčnikih.

Skupni imenovalec vseh sprememb v menstrualnem ciklusu je njihova dobro znana cikličnost, ki jo povzročajo hormoni hipotalamo-hipofizne-ovarialne osi. Ne smemo pa pozabiti, da lahko ultrazvočna slika samo v grobem prikazuje intenzivne transformacije, ki se dogajajo na celičnem in podceličnem nivoju teh biološko aktivnih tkiv [56].

Spremembe v maternični sluznici

Maternična sluznica (endometrij) lahko zaradi zelo občutljivih receptorjev spremlja hormonska dogajanja ter pri tem kaže zelo kompleksne spremembe svoje ultrastrukture [56]. V zgodnji folikularni fazi izgleda endometrij pri ultrazvočnem pregledu kot ena ehogena črta, ki se razteza od vrha maternice (fundusa) do cerviksa.

V naslednjih nekaj dneh, v sredini folikularne faze – 8. do 10. dan ciklusa, se endometrij debeli (debelina je že več kot 2 mm), njegova ehogenost pa se postopoma izgublja. Dva do tri dni pred ovulacijo, v pozni folikularni fazi, se ehogeni endometrij ponovno zoži na tanko linijo, medtem ko se okoli njega nahaja širša hipoehogena cona. Neposredno pred ovulacijo endometrij ne daje več močnega odmeva. V naslednjih dneh pa se endometrij ponovno debeli ter postaja enakomerno ehogen.

Ultrazvočno opazovanje maternične sluznice ne zadostuje za natančno opazovanje menstrualnega ciklusa in se ponavadi kombinira še s kakšno drugo metodo, npr. z opazovanjem sprememb v jajčnikih.

Dozorevanje jajčnih mešičkov

Mnogo pomembnejše od opazovanja maternične sluznice je spremljanje rasti jajčnih mešičkov. Mešičke ponavadi opazujemo od sredine folikularne faze pa do ovulacije. Porast koncentracije hormona FSH povzroči selekcijo in rast določene skupine mešičkov. Običajno se okoli 9. dne menstrualnega ciklusa en mešiček po svoji velikosti loči od ostalih (preovularni, dominantni mešiček). Dominantni mešiček je od ostalih mešičkov neodvisen in potrebuje za svoj nadaljnji razvoj manjše koncentracije hormona FSH, to pa povzroči, da ostali, v razvoju zaostali mešički, degenerirajo [47]. Pri pacientkah z inducirano ovulacijo se lahko razvije več preovularnih mešičkov.

Število in velikost posameznih mešičkov moramo pri vsakem pregledu zabeležiti, saj šele s tem lahko opazimo dinamiko njihove rasti ter ocenimo čas ovulacije. Za rast preovularnega mešička rečemo, da je linearen [56].

V normalnem ciklusu doseže mešiček 10. dan velikost od 8 do 10 mm [41]. Pri konstantnem dnevnem povečanju premera za 2,4 mm lahko predvidimo čas ovulacije s standardnim odklonom napake 1,37 dni. Povprečna vrednost premera mešička v maksimalni fazi razvoja je med 22 in 23 mm. Pokazalo se je tudi, da ni razlike v velikosti med mešički v spontanih in induciranih ciklusih [56].

Na koncu poglejmo še, kako vse to opazimo na ultrazvočnem prikazu. Ponavadi so jajčniki ovalne oblike, njihova velikost pa se spreminja s časom in hormonskim stanjem. Jajčnik je zaradi mnogo majhnih mešičkov hipoehogen in se na sliki opazi kot ovalna oblika, ki je temnejša od okolice [41]. Znotraj jajčnika zaznamo majhne strukture, ki so še temnejših barv. Te strukture so lahko ali mešički ali žile. Mešički niso povsem okrogli, ampak bolj ovalni, medtem ko v času ovulacije postanejo bolj okrogli. Zaradi tekočine, ki jo vsebujejo, so na ultrazvočni sliki temnih barv.

2.3 Predobdelava ultrazvočnih medicinskih slik

Čeprav eksperti (zdravniki) razberejo vse potrebne informacije že kar iz samega ultrazvočnega prikaza (originalnih – neobdelanih posnetkov), lahko trdimo, da so takšni posnetki skoraj neuporabni za računalniško analizo. To pomeni, da so motnje v ultrazvočnih posnetkih tako velike (podpoglavje 2.1.4), da je sposobnost odkrivanja objektov (organov oz. določenih struktur) bistveno zmanjšana, hkrati pa je zmanjšana tudi učinkovitost določenih računalniških algoritmov (npr. odkrivanje robov), posebej prirejenih za avtomatsko analizo slik. Posebej moteč je pegasti šum, ki ga vedno najdemo pri vseh ultrazvočnih posnetkih.

S predobdelavo ultrazvočnih posnetkov želimo izboljšati predvsem kvaliteto oz. vizualne lastnosti posnetkov ter jih tako ustrezno pripraviti za računalniško analizo. V teh slikah večinoma popravljamo kontraste sivin (izenačujemo histograme sivin ipd.), hkrati pa tudi odstranjujemo prisotnost šuma, predvsem pegastega. Za popravljanje kontrastov in drugih vizualnih lastnosti slik uporabljamo dobro znane histogramske tehnike [7, 31, 60, 63, 64]. Pegasti šum lahko delno odstranimo s klasičnimi operatorji za odstranjevanje šuma, npr. z nizkim sitom, medianim in Gaussovim filtrom [63], ali pa s kombinacijo osnovnih filtrov, npr. tako, da sliko najprej filtriramo z medianim filtrom z masko 3x3 piksle, zatem pa še z medianim filtrom velikosti 5x5 pikslov [40]. Če želimo sistematično reducirati pegasti šum, pa sta v literaturi znana dva pristopa [1, 38]: inkoherentna tehnika obdelave, to je povprečenje več slik, kjer spreminjamo parametre ultrazvočne naprave, in filterske tehnike, ki jih uporabljamo, ko so slike že posnete.

2.3.1 Odstranjevanje pegastega šuma

Inkoherentne tehnike temeljijo na povprečenju več slik iste strukture objekta, ki jih dobimo tako, da spreminjamo enega ali več parametrov ultrazvočnega instrumenta (npr. čas oddajanja ultrazvočnih valov). S tem postanejo pegasti vzorci v sliki bistveno bolj nekorelirani, hkrati se reducira varianca peg in tako postaneta povprečna vrednost ter tekstura objektov lažje zaznavni. Varianco peg pa lahko reduciramo tudi s filtrskimi tehnikami, ki jih uporabimo na že posnetih ultrazvočnih slikah. Pri tem se moramo zavedati, da preprosti filtri zameglijo informacije o objektih (npr. robove), ki bi jih radi ohranili [1]. V ultrazvočnih B-slikah želimo po filtriranju ohraniti naslednje značilnosti [33]: mejo med dvema regijama z majhno razliko v sivinah, majhne strukture, ki so velike približno tako kot pege, veliko paleto sivin itd.

Za nas so veliko bolj uporabne filtrske tehnike, saj ponavadi dobimo ultrazvočne posnetke od ekspertov in dejansko nimamo vpliva na to, kako so bili ti pridobljeni (nimamo vpliva na nastavitev parametrov ultrazvočnih inštrumentov). Poglejmo si nekaj teh metod.

Geometrijski filter [12] je bil v osnovi narejen za redukcijo pegastega šuma v slikah SAR (angl. synthetic aperture radar). Je enodimenzionalen filter, ki pa ga zaporedno uporabimo v 4 različnih smereh v 2D sliki: horizontalni, vertikalni ter dveh diagonalnih smereh. Avtorji poročajo o dobri redukcijski sposobnosti tega filtra, hkrati pa ohranja tudi prostorske informacije v sliki. Avtor Bamber [1] je naredil nov filter za odstranjevanje pegastega šuma. Glavna ideja tega algoritma je, da želi v sliki poiskati regije, ki so po določenem kriteriju podobne popolnoma razviti pegi (pega je popolnoma razvita, ko so ultrazvočni žarki naključni in fino razporejeni ter jih je mnogo znotraj resolucijske celice slikovnega sistema) ter jih zamenjati z regijami, ki bodo imele povprečno sivino. Hkrati pa bo ta algoritem pustil regije, ki niso podobne pegam, nedotaknjene. Dejansko je to nekakšen inteligenten filter, ki na osnovi kriterija podobnosti določa, koliko glajenja (filtriranja) potrebujejo posamezni deli slike. Novo filtrirano vrednost $g(x, y)$ določimo iz stare nefiltrirane vrednosti $f(x, y)$ za piksel na mestu (x, y) kot:

$$g(x, y) = M + k(f(x, y) - M),$$

kjer je M lokalno povprečje starih vrednosti v 3x3-okolici trenutnega piksla. Konstanto k uravnavamo z uporabljeno mero podobnosti p, ki je definirana kot ulomek med lokalno varianco σ^2 in lokalnim povprečjem M.

V zadnjem času se je pojavilo mnogo adaptivnih filtrskih tehnik (maska filtra se za vsak piksel spreminja), vendar moramo pri uporabi teh tehnik paziti predvsem na naslednje stvari:

- težko je določiti ustrezne uteži filtra, ki bodo veljavni za celotno ultrazvočno sliko,
- ni sistematičnega pravila za določanje velikosti lokalne regije za dano ultrazvočno B-sliko,
- odstranjevanje pegastega šuma in ohranjevanje robov sta nasprotujoči si zahtevi.

Avtor Koo [38] je predlagal nov adaptivni algoritem. Algoritem deluje tako, da najprej izberemo začetno regijo (ISR – angl. initial seed region), ki je kvadratne oblike in centrirana okoli piksla (FP), ki ga filtriramo. Zatem izračunamo povprečje M in varianco σ^2 za to začetno regijo. Sledi izračun lokalne homogenosti H, ki jo določimo kot

$$H = \frac{\sigma^2}{M}.$$

Če je izračunana homogenost manjša ali enaka pragovni vrednosti H_0 (H_0 je parameter ultrazvočne naprave, ki ga lahko eksperimentalno določimo), potem rečemo, da je začetna regija ISR homogena ter jo označimo za končno začetno regijo. Če ta pogoj ni izpolnjen, pa rečemo, da je v tej začetni regiji vsebovana strukturno nehomogena regija, ki jo moramo odstraniti iz začetne regije. Zato opravimo t. i. postopek krčenja začetne regije. Krčimo tako, da odstranjujemo piksle na meji regije. Najprej odstranimo vse piksle na eni strani začetne regije (piksli z meje so na štirih straneh začetne regije), ponovno izračunamo mero lokalne homogenosti ter opravimo primerjavo s H_0. Če je nova mera H manjša od H_0, potem postopek krčenja končamo, saj smo našli homogeno lokalno regijo – končno začetno regijo, sicer pa postopek ponovimo z odstranjevanjem pikslov na drugi strani. V ekstremnem primeru krčenja se lahko začetna regija skrči na eno samo točko – ta primer obravnavamo posebej.

Naslednji korak je rast končne začetne homogene regije. To opravimo tako, da k začetni regiji zaporedno dodajamo sosede, ki imajo enake statistične karakteristike kot začetna regija. Sosedi oz. sosednje regije so definirane na enak način kot pri krčenju (imamo 4 sosednje regije velikosti 1 x dolžina ene strani začetne regije). Pri rasti definiramo relativno varianco ${\sigma_N}^2$ sosednje regije kot

$$ {\sigma_N}^2 = \frac{1}{D} \sum_N \left[f(k,l) - M_C^2 \right], $$

kjer je D velikost sosednje regije, M_C pa je povprečna vrednost trenutne regije (ne združene regije). Zatem izračunamo mero podobnosti H_N med sosednjo regijo in trenutno regijo kot razmerje med σ_N^2 in M_C. Ponovno opravimo primerjavo med H_N ter H_0 in, če je izpolnjen pogoj manjše ali enako, potem to sosednjo regijo, ki je statistično podobna trenutni regiji, dodamo k trenutni regiji. Rast ponavljamo, dokler še lahko dodamo kakšnega soseda. Ko je rast regije končana, zamenjamo vrednost piksla, ki ga filtriramo (FP) z aritmetičnim povprečjem te nove regije.

Omenili smo že, da se lahko začetna regija skrči na en sam piksel. To se lahko zgodi v primeru, ko je piksel, ki ga filtriramo, impulz ali pa se nahaja na določenem robu. V tem primeru vrednost piksla FP zamenjamo z aritmetičnim povprečjem osmih obdajajočih pikslov. Avtorji poročajo o veliki učinkovitosti tega filtra, hkrati pa tudi o njegovi zmožnosti, da ohranja robove regij.

V [33] smo zasledili še en adaptivni filter za odstranjevanje pegastega šuma, ki filtrira B-slike tako, da uporablja lokalne statistike. Sestavljen je iz treh korakov: izračuna lokalnih statistik, rasti regije in uporabe operatorja za glajenje.

V literaturi [24, 71] smo zasledili, da je za odstranjevanje pegastega šuma primerna tudi diskretna valčna transformacija (DWT). Avtorji Guo in ostali [24] ter Wei in ostali [71] predlagajo naslednji postopek za izločanje pegastega šuma v slikah SAR: najprej originalno sliko transformiramo z DWT, nad transformiranko opravimo mehko pragovno operacijo, zatem pa z inverzno DWT dobimo filtrirano sliko.

V tem podpoglavju smo opisali nekaj filtrov, s katerimi lahko odstranimo pegasti šum iz ultrazvočnih slik. Na koncu moramo le še dodati, da lahko že tudi s preprostimi filtri, kot je npr. nizko sito, ustrezno pripravimo medicinsko ultrazvočno sliko za analizo. To dejansko pomeni, da je izbira ustreznega filtra za predobdelavo v veliki meri odvisna od tipa naloge, katero rešujemo.

3.

Segmentacija

Segmentacija je eden izmed najpomembnejših korakov obdelave, ki pripravi sliko za podrobno analizo. Njen glavni cilj je razdeliti sliko na dele, ki kar najverjetneje ustrezajo objektom ali pa območjem iz realnega sveta [7, 26, 27, 31, 60, 63]. Ločimo dve vrsti segmentacije: popolno in delno. Popolna segmentacija vrne kot rezultat množico nezdruženih regij, ki ustrezajo objektom v vhodni sliki, medtem ko se pri delni segmentaciji regije direktno ne ujemajo z objekti. Za popolno segmentacijo je v proces segmentacije ponavadi treba vključiti določeno vnaprejšnje znanje o problemu. Kljub temu pa obstaja cela vrsta segmentacijskih problemov, ki jih lahko rešimo na nizkem nivoju obdelovanja – takšno obdelovanje je kontekstno neodvisno, uporabljeni so modeli, ki niso usmerjeni k objektom in nobeno vnaprejšnje znanje o pričakovanih segmentacijskih rezultatih ni vključeno. Ogledali si bomo segmentacijske metode, ki delujejo na nizkem nivoju obdelovanja, kontekstno odvisno segmentacijo pa najdemo v [63].

Kakšna je dobra segmentacija slike? Regije, dobljene s segmentacijo, naj bodo enotne in homogene glede na določene karakteristike, kot so npr. sivine ali teksture. Notranjost regij naj bo preprosta in brez mnogih lukenj. Sosednje regije naj imajo izrazito različne vrednosti glede na karakteristike, po katerih so enotne. Meje vsakega segmenta naj bodo preproste, nenazobčane (raskave) in prostorsko natančne. Zadostiti vsem tem želenim lastnostim segmentacije je težko, ker so striktno enotne in homogene regije polne majhnih lukenj in imajo nazobčane meje. Če vztrajamo, da naj imajo sosednje regije velike razlike v vrednostih, lahko povzročimo, da se regije zlijejo in meje med posameznimi segmenti izginejo.

Kot ne obstaja nobena teorija pri grupiranju (angl. clustering), tudi ne obstaja teorija pri segmentiranju slik [26]. Segmentacijske tehnike so večinoma "ad hoc" in se razlikujejo v načinu, po katerem poudarijo eno ali več želenih lastnosti, in v načinu, kako uravnavajo in delajo kompromise med želenimi lastnostmi.

V nadaljevanju bomo opisali segmentacijske metode, ki jih bomo glede na njihovo dominantno značilnost, ki jo uporabljajo, razdelili v tri velike skupine. Prva skupina metod – pragovna operacija – uporablja globalno znanje o sliki, ki ga ponavadi predstavimo s histogramom sivin. Segmentacija na osnovi robov tvori drugo veliko skupino metod, zadnja skupina metod pa je segmentacija na osnovi regij. V podpoglavju 3.4 se bomo seznanili tudi z nekakšno kvazi segmentacijsko metodo – ujemanjem, s katerim v sliki iščemo znan vzorec (objekt). Poglavje pa bomo zaključili s preizkusom izbranih segmentacijskih metod na umetnih slikah.

Vse segmentacijske metode bomo opisali na vhodni sliki sivin f, vendar ni nobene razlike v segmentaciji, če funkcija f predstavljala tudi druge veličine, npr. velikost gradienta, mero za opis vzorcev, parametre za opis gibanja v slikah itd.

3.1 Pragovna operacija

Pragovna operacija (angl. thresholding) je najpreprostejša segmentacijska metoda. Je računsko enostavna in hitra, izredno lahko pa jo je uporabiti tudi za realnočasovne aplikacije. Za segmentacijo objekta in ozadja moramo določiti konstanto svetlosti oz. prag. Pragovna operacija je transformacija vhodne slike f v izhodno segmentirano (binarno) sliko g po predpisu:

$$g(i,j) = \begin{cases} 1, & \text{če } f(i,j) \geq T \\ 0, & \text{sicer} \end{cases}, \tag{3.1}$$

kjer je T konstanten prag (konstantna vrednost sivine oz. svetlosti), $g(i,j) = 1$ označuje piksel objekta, $g(i,j) = 0$ pa piksel ozadja.

Če se objekti v sliki ne dotikajo (tudi ne prekrivajo) in če se vrednosti sivin objektov bistveno razlikujejo od sivin ozadja, potem je pragovna operacija z globalnim pragom (en prag veljaven za vso sliko – enačba (3.1)) primerna segmentacijska metoda. V praksi je to zelo redko, saj celo pri zelo enostavnih slikah (scenah) pride do velikih odstopanj v sivinah objekta in ozadja (npr. odstopanja zaradi neenakomerne osvetlitve scene, neenakomerne vhodne naprave), kar naredi globalen prag neprimeren. Zato v današnjih aplikacijah raje uporabljamo variabilen prag, ki se spreminja s položajem v sliki [63]. Globalen prag je ponavadi rezultat neke funkcije $T = T(f)$ nad celotno vhodno sliko f, medtem ko je variabilen prag $T(i,j) = T(f, f_{lokalna})$ določen lokalno glede na del slike, v katerem se prag določa. S tako definiranim variabilnim pragom lahko reformuliramo pragovno operacijo iz enačbe (3.1) ter zapišemo njeno splošno obliko:

$$g(i,j) = \begin{cases} 1, & \text{če } f(i,j) \geq T(i,j) \\ 0, & \text{sicer} \end{cases}. \tag{3.2}$$

Glavna razlika med tako definirano pragovno operacijo in definicijo iz enačbe (3.1) je, da se sedaj odločamo o posameznem pikslu, ali pripada objektu ali ne, in to na osnovi primerjave s pragom, ki je točno določen za ta piksel (vsak piksel ima svoj prag).

Osnovna oblika pragovne operacije ima mnogo različic [31, 60, 63]. V praksi se zelo pogosto uporablja operacija polovičnega praga (angl. semitresholding), s katero maskiramo piksle ozadja (postavimo na 0), medtem ko vrednosti objektov pustimo nespremenjene. Pomembna je tudi operacija pasovnega praga, ki ima obliko

$$g(i,j) = \begin{cases} 1, & \text{če } f(i,j) \in D \\ 0, & \text{sicer} \end{cases} . \tag{3.3}$$

Ta operacija segmentira sliko v skupino pikslov, ki imajo vrednosti sivin iz množice sivin D, ter v piksle ozadja. V praksi velikokrat srečamo tudi večpragovno operacijo:

$$g(i,j) = \begin{cases} 0, & \text{če } f(i,j) < T_1 \\ 1, & \text{če } T_1 \leq f(i,j) < T_2 \\ 2, & \text{če } T_2 \leq f(i,j) < T_3 \\ \dots & \\ n-1, & \text{če } T_{n-1} \leq f(i,j) < T_n \\ n, & \text{sicer} \end{cases} , \tag{3.4}$$

s katero razdelimo sliko na več regij, pri čemer ima vsaka regija svojo številko. Segmentirana slika sedaj ni več binarna, ampak ima več nivojev sivin.

Pragovna operacija je v vseh svojih različicah zelo preprosta. Najpomembnejši del te metode je izbira ustreznega praga. Prag lahko izberemo sami (ga npr. interaktivno vnesemo ali pa fiksiramo na določeno vrednost), obstaja pa tudi veliko metod za določanje pragov. Nekatere izmed teh avtomatično določajo prag, pri nekaterih pa lahko uporabnik z nastavljanjem določenih parametrov uravnava (nadzoruje) izbiro pragov. Mnogi avtorji [22, 25, 55, 66] so pokazali, da je odločitev za določeno metodo izbire pragov v prvi vrsti odvisna od tipa slik, s katerimi imamo opravka (ni vsaka metoda primerna za vsako sliko), to pa zaradi tega, ker pri določitvi praga vsaka metoda uporablja drugačno značilnost slike (globalni histogram, lokalni histogram, velikost gradienta itd.).

V splošnem lahko razdelimo vse metode določanja pragov na metode, ki določajo globalni prag, in na adaptivne metode, ki rezultirajo v variabilnem pragu. V nadaljevanju si bomo ogledali obe vrsti metod.

3.1.1 Metode za določanje globalnih pragov

Čeprav v splošnem za segmentiranje slike en globalni prag ne zadostuje, pa obstaja cela vrsta aplikacij (nalog), kjer je z enim samim pragom možno doseči zadovoljivo razdelitev slike. To je tudi glavni razlog, da obstaja cela vrsta takšnih metod, ki so hkrati podprte tudi s teorijo. Večina izmed teh metod določa prag na osnovi histograma sivin slike, zato si bomo v nadaljevanju najprej ogledali te metode.

Metode na osnovi histograma

Glasbey je v svojem članku [20] podal natančen pregled metod določanja pragov na osnovi histograma sivin. Preden bomo opisali nekaj izmed teh metod, podajmo potrebne definicije. Naj bo $h = (h_0, h_1, \dots, h_n)$ histogram sivin, pri čemer naj element h_i pomeni število pikslov[1] v sliki z vrednostjo sivine i, kjer je n maksimalna sivina v sliki (ponavadi 255), T pa bo predstavljal prag.

[1] Element h_i je lahko definiran tudi kot verjetnost, da ima piksel sivino i.

Prva metoda se imenuje MINIMUM. Algoritem privzema bimodalen histogram, prag T je določen kot minimum h_i med dvema maksimumoma. Če histogram ni bimodalen, potem ga moramo gladiti (filtrirati), dokler ne postane bimodalen. Metoda INTERMODES poišče maksimuma h_j in h_k, prag T pa postavi na vrednost $(j+k)/2$.

Za opis naslednjih algoritmov bomo vpeljali dodatne oznake:

$$A_j = \sum_{i=0}^{j} h_i, \quad B_j = \sum_{i=0}^{j} i h_i, \quad C_j = \sum_{i=0}^{j} i^2 h_i, \quad D_j = \sum_{i=0}^{j} i^3 h_i, \quad j = 0, \ldots, n.$$

Metoda MEDIAN izbere prag T tako, da je ulomek A_T/A_n kar se da blizu vrednosti $0,5$ (metoda p: deli, ko je ulomek A_T/A_n čim bližje vrednosti p), medtem ko je v metodi MEAN prag določen kot B_n/A_n. Ob vrsti preprostih metod pa obstaja kar nekaj zelo kompleksnih metod določanja pragov.

Ideja metode MOMENTI je, da mora segmentirana binarna slika imeti prve tri momente enake kot originalna slika. Prag T izberemo tako, da je A_T/A_n najbližji x_0, pri čemer so:

$$x_0 = \frac{1}{2} - \frac{B_n/A_n + x_2/2}{\sqrt{x_2^2 - 4x_1}}, \quad x_1 = \frac{B_n D_n - C_n^2}{A_n C_n - B_n^2}, \quad x_2 = \frac{B_n C_n - A_n D_n}{A_n C_n - B_n^2}.$$

Iterativna metoda INTERMEANS(I) oz. OPTIMAL ([63]) potrebuje na vhodu začetni približek praga (npr. MEAN). Prag iterativno izračunavamo kot $T = (\mu_T + \nu_T)/2$, kjer sta $\mu_T = \frac{B_T}{A_T}$ in $\nu_T = \frac{B_n - B_T}{A_n - A_T}$, dokler ne dosežemo konvergence (npr. $T^{(k+1)} = T^{(k)}$). Tudi metoda MINERROR(I) je iterativna in potrebuje začetni približek. Za njen opis potrebujemo naslednje oznake:

$$p_T = \frac{A_T}{A_n} \quad q_T = \frac{A_n - A_T}{A_n}, \quad \sigma_T^2 = \frac{C_T}{A_T} - \mu_T^2, \quad \tau_T^2 = \frac{C_n - C_T}{A_n - A_T} - \nu_T^2.$$

Celi del večje rešitve kvadratne enačbe

$$x^2 \left\{ \frac{1}{\sigma^2} - \frac{1}{\tau^2} \right\} - 2x \left\{ \frac{\mu}{\sigma^2} - \frac{\nu}{\tau^2} \right\} + \left\{ \frac{\mu^2}{\sigma^2} - \frac{\nu^2}{\tau^2} + \log\left(\frac{\sigma^2 q^2}{\tau^2 p^2}\right) \right\} = 0 \tag{3.5}$$

nam da nov prag T (seveda, če rešitev obstaja).

Kapur je v članku [32] predstavil novo metodo iskanja pragov, ki temelji na konceptu entropije (ENTROPIJA). Naj histogram h predstavlja verjetnosti posameznih sivin. Tedaj lahko definiramo dve verjetnostni porazdelitvi (ena za objekt, druga za ozadje): prva (A) je definirana za diskretne vrednosti od 0 do T, druga (B) pa od $T+1$ do n, in sicer:

$$A : \frac{h_0}{P_T}, \frac{h_1}{P_T}, \ldots, \frac{h_T}{P_T}, \quad P_T = \sum_{i=0}^{T} h_i;$$
$$B : \frac{h_{T+1}}{1-P_T}, \frac{h_{T+2}}{1-P_T}, \ldots, \frac{h_n}{1-P_T}.$$

Obema porazdelitvama priredimo entropiji, in sicer:

$$H(A) = -\sum_{i=0}^{T} \frac{h_i}{P_T} \ln(\frac{h_i}{P_T}) = \ln P_T + \frac{H_T}{P_T}, \quad H_T = -\sum_{i=0}^{T} h_i \ln h_i;$$
$$H(B) = -\sum_{i=T+1}^{n} \frac{h_i}{1-P_T} \ln(\frac{h_i}{1-P_T}) = \ln(1-P_T) + \frac{H_n - H_T}{1-P_T}.$$

Zatem definiramo kriterijsko funkcijo $\Psi(T)$ kot vsoto $H(A)$ in $H(B)$. Kapur je pokazal, da po maksimizaciji kriterijske funkcije dobimo maksimalno informacijo med porazdelitvijo objekta in porazdelitvijo ozadja v sliki. Diskretna vrednost, ki maksimizira $\Psi(T)$, je prag.

Idejo entropije je uporabil tudi Li v [43], kjer je minimiziral navzkrižno entropijo med originalno in segmentirano sliko. Izhajal je iz naslednjih omejitev:

(i) Segmentirana slika $G = (g_1, g_2, \ldots, g_N)$ ima nivoja μ_1 in μ_2 (originalna in segmentira slika sta predstavljeni kot en velik vektor),

(ii)
$$\sum_{f_i<T} f_i = \sum_{f_i<T} \mu_1,$$

(iii)
$$\sum_{f_i\geq T} f_i = \sum_{f_i\geq T} \mu_2,$$

Te omejitve omogočajo, da μ_1 in μ_2 določimo kot:

$$\mu_1 = \frac{\sum_{f_i<T} f_i}{N_1} \quad \mu_2 = \frac{\sum_{f_i\geq T} f_i}{N_2},$$

pri čemer sta N_1 in N_2 število pikslov v obeh regijah. Iz zgornjih omejitev in z uporabo entropije dobimo naslednjo kriterijsko funkcijo:

$$\eta(T) = \sum_{f_i<T} f_i \log(\frac{f_i}{\mu_1(T)}) + \sum_{f_i\geq T} f_i \log(\frac{f_i}{\mu_2(T)}). \tag{3.6}$$

Prag dobimo z minimizacijo funkcije $\eta(T)$. Podobna ideja je v Otsujevi in Brinkovi metodi [3] (ki je posplošitev Otsujeve metode), kjer pa minimiziramo medrazredno varianco z metodo najmanjše kvadratne razdalje pri istih omejitvah. Kriterijska funkcija ima obliko:

$$\theta(T) = \sum_{f_i<T} (f_i - \mu_1(T))^2 + \sum_{f_i\geq T} (f_i - \mu_2(T))^2. \tag{3.7}$$

Enačbi (3.6) in (3.7) lahko izrazimo s histogramsko funkcijo h in ju tako poenostavimo.

Glasbey [20] je testiral te algoritme na množici histogramov, ki so bili mešanica dveh Gaussovih porazdelitev (umetno tvoril bimodalne histograme). Prišel je do naslednjih sklepov: ENTROPIJA in MOMENTI vračata prenizke pragove, podobno velja v določenih primerih za MEAN in MEDIAN, INTERMEANS(I) pa skuša razbiti daljšo komponento, če so razmerja med obema porazdelitvama neenaka. Algoritma MINIMUM in MINERROR(I) vračata podobne rezultate. MINERROR(I) je ocenil kot najboljšega, in sicer, ker se preprosto izračuna, redkokdaj ne uspe in je relativno neobčutljiv na variabilnost vzorcev. V praksi so histogrami vse prej kot mešanica dveh Gaussovih porazdelitev, vendar pa bi bilo preoptimistično upati, da bo metoda, ki se je slabo odrezala pri Gaussovi mešanici, dobro delala tudi v bolj kompleksnih situacijah. V resnici pa se lahko zgodi obratno, saj je npr. Hannah [25] zelo uspešno uporabil slabo ocenjeni Kapurjevo in Otsujevo metodo pri odkrivanju majhnih tujkov v vrečkah z zamrznjeno hrano, o podobnem uspehu entropijske metode, ki jo je uporabil pri detektiranju dima iz industrijskih dimnikov, pa je poročal Tseng [66], če naštejemo samo nekatere.

Vse zgoraj opisane metode, razen MEAN, lahko enostavno razširimo na iskanje dveh ali več pragov.

Prednosti teh algoritmov so, da uporabljajo globalne lastnosti histograma. Kljub temu pa ostajamo še vedno pred problemom: kaj če imata dve različni sliki enak histogram in s tem tudi isti prag? Je ta prag primeren za obe sliki? Navedeni problem nas opozarja, da moramo vedno preveriti pravilnost rezultatov, če segmentiramo s pragovno operacijo, pri čemer smo prag določili iz histograma.

Hibridne metode za določanje globalnih pragov

Srečali smo se že z najpreprostejšo metodo določanja pragov, ko prag določimo kar sami. Takšnih kvazi metod bi si lahko izmislili še veliko, vendar o njih ne bomo govorili. Globalni prag je vedno določen na osnovi neke globalne informacije v sliki. Histogram sivin najbolj popolno zajema informacije o porazdelitvi sivin v sliki, o povprečju, varianci itd., zato ga pri določanju pragov večinoma uporabljajo tudi hibridne metode. Pri hibridnih metodah pravimo, da se uporaba histograma sivin dopolnjuje še s kakšnim drugim konceptom.

V nadaljevanju bomo predstavili hibridno parametrično metodo. Videli smo, da mnoge metode za izbiro pragov težijo k avtomatizaciji izbire pragov, vendar to v nekaterih primerih vodi do slabih segmentacijskih rezultatov. Če se odpovemo avtomatičnemu določanju pragov in če vpeljemo uglaševalni faktor, ki ga lahko preprosto nadzoruje uporabnik, lahko dobimo segmentacijski sistem velike zanesljivosti. To idejo je uporabil Morii [51] v svoji nadzorovani – iterativni metodi, kjer so pragi izbrani po kriteriju minimalnega obteženega kvadratičnega popačenja. Najprej določimo dva poskusna praga y_1 in y_2 kot:

$$y_1^{(k)} = \frac{\sum_{i=0}^{t^{(k)}} ih_i}{\sum_{i=0}^{t^{(k)}} h_i}, \qquad y_2^{(k)} = \frac{\sum_{i=t^{(k)}+1}^{n-1} ih_i}{\sum_{i=t^{(k)}+1}^{n-1} h_i}.$$

S pomočjo poskusnih pragov pa določamo prag T po naslednji enačbi:

$$T^{(k+1)} = \begin{cases} \lfloor (y_1^{(k)} + y_2^{(k)})/2 \rfloor, & \text{če } \omega_1 = \omega_2 \\ \lfloor \frac{\omega_1 y_1^{(k)} - \omega_2 y_2^{(k)} + \sqrt{\omega_1 \omega_2} |y_1^{(k)} - y_2^{(k)}|}{\omega_1 - \omega_2} \rfloor, & \text{sicer} \end{cases}, \tag{3.8}$$

kjer sta ω_1 in ω_2 uteži (parametra), zanju pa velja $\omega_1 + \omega_2 = 1$ in $\omega_1, \omega_2 > 0$. Za $T^{(0)}$ lahko izberemo $\lfloor n/2 \rfloor$, izračun pa ponavljamo do konvergence. Utež ω_1 lahko dobimo iz analize verjetnosti histogramskega modela ali pa po analizi množice realnih slik, kjer po njihovi ročni obdelavi dobimo $\omega_{1,min}$ in $\omega_{1,max}$, ti vrednosti pa potem uporabimo na originalnih slikah.

3.1.2 Metode za določanje variabilnih pragov

Z enim samim globalnim pragom v splošnem ne moremo uspešno segmentirati kompleksnejših slik ali slik, pridobljenih pod slabimi osvetlitvenimi pogoji. Takšen primer je slika, kjer je objekt na enem delu slike svetlejši (oz. temnejši) kot ozadje v drugem delu. V takšnih primerih si pomagamo z variabilnimi pragi, ki se spreminjajo od piksla do piksla (enačba (3.2)). Variabilni prag lahko določimo za vsak piksel posebej kot neko funkcijo okolice obravnavanega piksla, lahko pa sliko najprej razdelimo na nekaj podslik, določimo globalne pragove, ki veljajo za središčne piksle teh podslik, pragove za vse ostale piksle pa dobimo po interpolaciji. Te metode imenujemo tudi (lokalno) adaptivna ali dinamična pragovna operacija. V nadaljevanju bomo predstavili nekaj tipičnih metod.

Marino [46] je predstavil lokalno adaptivno metodo, sestavljeno iz prednosti hiperboličnega in histereznega principa pri določanju pragov. Po hiperboličnem zakonu določimo lokalni prag T kot:

$$T = Gain \frac{Q}{povp + 1}, \tag{3.9}$$

kjer je Q kvantizacijski nivo slike (npr. 256), $povp$ je povprečje 8 sosednjih pikslov, $Gain$ pa je parameter, ki ga določimo glede na tip naloge (ponavadi 70-80 % vrednosti MEDIAN) in je odvisen od histograma vhodne slike. Ta vrsta segmentacije tvori močne erozije v neposredni bližini robov. Pri histereznem principu pa imamo opravka z dvema pragoma, in sicer odreznim pragom T_1 in varnostnim pragom T_2, ki je višji kot prvi. Ideja tega principa je, da je piksel, ki ima sivino večjo od T_1, lahko obravnavan kot visok le, če je povezan s semenom, npr. takšnim pikslom, ki ima sivino višjo od T_2. Prednosti so omejena erozija in odstranitev izoliranega šuma, omejitev pa je slaba učinkovitost v homogenih področjih. Hibridna metoda združuje prednosti lokalne hiperbolične pragovne operacije in histereznega principa. Implementacija te metode je podobna kot pri histereznem principu, razlika je le ta, da se prag T_2 izračuna lokalno z uporabo hiperboličnega zakona. S pomočjo praga T_2 pa izračunamo T_1 kot:

$$T_1 = T_2(1 - \frac{1}{h}), \tag{3.10}$$

kjer je

$$\frac{1}{h} = \frac{\lfloor a-b-c+d \rfloor + \lfloor a-b+c-d \rfloor + \lfloor a+b-c-d \rfloor}{3Q}$$

in so a, b, c in d sivine vogalnih pikslov iz maske 3x3, centrirane okoli trenutno obravnavanega piksla.

Glavna omejitev večine lokalno adaptivnih metod je, da je za njihovo implementacijo treba opraviti več prehodov skozi sliko (najprej določimo npr. lokalni prag, ki ga moramo še interpolirati po vsej sliki) – s tem postanejo neuporabne za realnočasovne aplikacije. Perez [55] je razvil rekurzivno pragovno metodo na osnovi modela tvorbe slike iz svetilnosti i ter odbojne jakosti r ($f(x, y) = r(x, y)\, i(x, y)$). V enodimenzionalnem primeru se prag T_x določi z zaporednim pregledovanjem vektorja (vrstica ali stolpec slike), in sicer tako, da staremu pragu T_{x-1} prištejemo popravek. V dvodimenzionalnem primeru pa seštejemo obtežena praga, dobljena po vrstici in stolpcu:

$$T_{xy} = \omega_1 T_{xy}^{vrstica} + \omega_2 T_{xy}^{stolpec}. \tag{3.11}$$

Ta algoritem se odlično izkaže tudi pri slikah s slabo osvetljeno sceno in pri manjših sencah. S pragovno operacijo v slikah s slabo osvetljeno sceno se je ukvarjal tudi Parker [53]. Ideja njegove lokalne metode je podobna kot pri operatorjih za poudarjanje robov, in sicer, da objekt v sliki tvori majhne regije z relativno velikim gradientom intenzivnosti (ponavadi na robu objekta), medtem ko je na robu slike ta gradient majhen. Ta metoda najprej iz gradientne slike poišče piksle, ki zagotovo pripadajo objektu, o vsakem neklasificiranem pikslu pa se na osnovi primerjave z zanj določenim pragom odloči, ali pripada objektu ali ne. V [44] najdemo lokalno adaptivno metodo, ki je posebej primerna za segmentiranje majhnih in nizkokontrastnih objektov iz šumnega okolja. Zanimivo se je problema segmentacije lotil Rodriquez [58], ki z adaptivno pragovno metodo izloča ozadje v dveh fazah. V prvi fazi določi homogene regije ozadja, v drugi fazi pa izboljša izločanje ozadja v nehomogenih regijah. Prednost te metode je, da dovoljuje prisotnost objektov, ki so temnejši in/ali svetlejši kot lokalno ozadje, v nekaterih primerih pa celo več kot eno ozadje.

3.1.3 Metode za določanje več globalnih pragov

V praksi velikokrat želimo segmentirati sliko na več kot dve regiji (ozadje in objekt), saj so npr. v sliki različni objekti, ali pa nas morda zanimajo detajli na objektu (ali ozadju) ipd. To nalogo lahko opravimo z večpragovno operacijo, ki je definirana z enačbo (3.4). Tudi pri večpragovni operaciji je bistvenega pomena pravilna oz. ustrezna izbira pragov.

V podpoglavju 3.1.1 smo zapisali, da lahko večino metod za določanje globalnih pragov enostavno razširimo na iskanje dveh ali več globalnih pragov. Trivialna posplošitev metode za določanje globalnih pragov na iskanje več pragov je, da najprej s to metodo za originalno sliko določimo globalen prag. S tem nam histogram slike razpade na dva dela. Zatem pa za vsak del histograma (slike) ponovno, s to isto metodo, določimo globalen prag. Ta postopek ponavljamo, dokler je v obeh delih še dovolj pikslov. Takšna posplošitev je uporabna za vsako metodo, vendar se poraja vprašanje o kvaliteti tako določenih pragov.

Zato pri posploševanju metode raje upoštevamo že pri izpeljevanju formul, da imamo opravka z več pragi [20, 32, 43], npr. pri Kapurjevi metodi [32] nimamo več opravka z dvema verjetnostnima porazdelitvama (podpoglavje 3.1.1), ampak z več porazdelitvami in s tem tudi z več entropijami, kriterijska funkcija, ki jo moramo maksimizirati, pa je funkcija več spremenljivk.

Wang [70] je predstavil algoritem za avtomatsko izbiro več pragov. Algoritem je zasnovan na ideji, da so gradienti v bližini robov objektov višji. Zato potrebujemo natančno opredeljeno sliko robov, kajti vsi piksli v sliki robov nimajo iste vloge v relativni svetlosti slike f. Eni robni piksli so lahko svetli (vrednost 2), drugi temni (vrednost 1) ali pa sivi. Zato tvorimo klasificirano sliko robov C po naslednjem hevrističnem postopku:

$$C(x,y) = \begin{cases} 1, & \text{če je } (x,y) \text{ robni piksel in } f(x,y) \in [gmin(x,y), \mu(x,y) - \Delta(x,y)] \\ 2, & \text{če je } (x,y) \text{ robni piksel in } f(x,y) \in [\mu(x,y) + \Delta(x,y), gmax(x,y)] \\ 0, & \textit{sicer} \end{cases},$$

kjer sta $gmin(x,y)$ in $gmax(x,y)$ minimalna oz. maksimalna sivina v 3x3-okolici piksla (x,y), medtem ko sta μ in Δ definirana kot:

$$\mu(x,y) = \frac{(gmax(x,y) + gmin(x,y))}{2} \quad \text{in} \quad \Delta(x,y) = \frac{(gmax(x,y) - gmin(x,y))}{2N},$$

parameter N pa je konstanta (npr. 3). Iz klasificirane slike C tvorimo dva histograma, in sicer histograma svetlih pikslov h_b ter temnih pikslov h_d:

$$\begin{aligned} h_b(sivina) &= \text{število elementov v } \{(x,y) | f(x,y) = sivina, C(x,y) = 2\}, \\ h_d(sivina) &= \text{število elementov v } \{(x,y) | f(x,y) = sivina, C(x,y) = 1\}. \end{aligned}$$

Sedaj lahko zapišemo algoritem avtomatskega določanja več pragov:

g_1 = minimalna sivina v originalni sliki f
g_2 = maksimalna sivina v originalni sliki f
$j = 0$
REPEAT
1. *$j = j + 1$.*
2. *Tvori h_d in h_b.*
3. *Če je število populacije v h_d ali h_b premajhno, potem END.*
4. *Poišči sivini g_b in g_d, kjer sta $h_b(g_b)$ in $h_d(g_d)$ maksimalna.*
5. *Na osnovi verjetnosti $P_b(T)$, da dobimo svetle regije, če opravimo pragovno operacijo pri T:*

$$P_b(T) = \frac{\sum_{i=T}^{g_2} h_b(i)}{\sum_{i=T}^{g_2} h_d(i) + \sum_{i=T}^{g_2} h_b(i)},$$

in na osnovi verjetnosti $P_d(T)$, da dobimo temne regije:

$$P_d(T) = \frac{\sum_{i=g_1}^{T} h_d(i)}{\sum_{i=g_1}^{T} h_d(i) + \sum_{i=g_1}^{T} h_b(i)},$$

se odločimo o pragu. Pri tem uporabljamo naslednje pravilo:

$$T(j) = \begin{cases} g_d, & \text{če } P_d(g_d) \geq P_b(g_b) \\ g_b, & \text{sicer} \end{cases}.$$

6. Če $T(j) = g_d$, potem $g_1 = g_d + 1$, sicer pa $g_2 = g_b - 1$.
END.

Algoritem 1: Algoritem za avtomatsko določanje več pragov.

Pri pragovni operaciji predvidevamo, da bomo s pragom g_d dobili temne regije, pri g_b pa svetle regije.

Večina metod za določanje globalnih pragov najprej tvori histogram sivin in iz njega z izračunanjem določenih mer, ki najbolje razdelijo nivoje sivin v histogramu, določijo prage. Slabost tega pristopa je, da je striktno globalen, saj se pri določanju pragov ne uporablja dejstvo, da so regije sestavljene iz sosednjih ali povezanih pikslov, katerih vrednosti sivin so približno enake. Gorman [22] pa je poiskal prage, ki ohranjajo povezanost pikslov znotraj regije. Povezanost je lokalna mera, ki pa jo izračunamo skozi vso sliko. Ideja te parametrične metode je ekvivalentna opravljanju binarizacije pri vsaki sivini, kjer zatem izberemo prage znotraj obsega sivin, kjer je število povezanih regij konstantno. To zagotavlja, da so pragi locirani pri sivinah, kjer je verjetnost, da se bo posamezna regija razdelila, najmanjša. Metoda je posebej primerna za slike z nizkim kontrastom, kjer je tudi sicer težko opraviti pragovno operacijo, ali pa, če število pragov ni vnaprej znano. Tedaj bo izbrala optimalno število pragov.

3.1.4 Pragovne operacije za slike, predstavljene s hierarhičnimi podatkovnimi strukturami

Glavna ideja opravljanja pragovne operacije na hierarhično predstavljenih slikah temelji na navadnih pragovnih operacijah (podpoglavje 3.1.1), cilj pa je, da skušamo odkriti prisotnost regije v sliki pri nižji resoluciji, zatem pa to regijo natančneje popisujemo od najnižje do originalne (najvišje) resolucije [63]. Določene metode delujejo na piramidnih podatkovnih strukturah, kjer sliko z nižjo resolucijo dobimo s povprečenjem določenih pikslov iz slike, predstavljene z višjo resolucijo. Obstajajo še druge hierarhične podatkovne strukture (npr. štiriška drevesa, kjer sliko iterativno delimo na 4 podslike) [31, 60, 63], vendar si bomo ogledali le metode, temelječe na piramidnih strukturah.

Burt [5] je prvi predstavil piramidno strukturo – Gaussovo piramido – za opis slik. Gaussova piramida je zaporedje slik, v katerem vsako sliko dobimo iz njenega predhodnika s filtriranjem z nizkim sitom. Če je F_{l-1} slika na $(l-1)$-tem nivoju in ima dimenzijo MxM, potem je slika F_l dimenzije $M/2$ x $M/2$ in jo dobimo iz:

$$F_l(x,y) = \sum \sum_{m,n=-2}^{2} w(m,n) F_{l-1}(2x+m, 2y+n), \tag{3.12}$$

kjer je w utežna funkcija oz. generacijsko jedro. Za funkcijo w so postavljene naslednje omejitve: 1) mora se dati razcepiti $w(m,n) = \hat{w}(m)\hat{w}(n)$ (po stolpcih in vrsticah), 2) biti mora normalizirana $\sum \hat{w}(m,n) = 1$, 3) simetrična $\hat{w}(m) = \hat{w}(-m)$ in 4) veljati mora princip enakega doprinosa (npr. za 5x5-jedro $[c\, b\, a\, b\, c]$ bi veljalo: a prosta spremenljivka, $b = 1/4$ in $c = 1/4 - a/2$).

Iz Gaussove piramide takoj dobimo Laplaceovo piramido, in sicer tako, da od Gaussove slike F_l (ki jo moramo še prej z interpolacijo spraviti na originalno resolucijo) odštejemo F_{l+1} (prav tako originalnih dimenzij), pri najvišji resoluciji pa sta obe sliki, Gaussova in Laplaceova, enaki.

Ponavadi predstavljamo s piramidnimi strukturami slike, katerih dimenzije so potence števila 2 (npr. 2^L x 2^L). Piramidna struktura porabi le za tretjino prostora več kot originalna slika. Laplaceove, posebej pa še Gaussove piramide so izredno učinkovita računska orodja, uporabna na raznih področjih (npr. pri analizi tekstur in gibanja, pri slikovnem kodiranju). Za nas je kot ena izmed segmentacijskih tehnik posebej zanimivo piramidno povezovanje (angl. pyramid linking). Poteka pa takole:

Korak 0: Naj bo F_0 originalna slika in hkrati najnižji nivo Gaussove piramide. Tvori Gaussovo piramido z generacijskim jedrom dolžine 4 ($a = b = 1/4$), in to do nivoja K.

Korak 1: Vsako vozlišče poveži z očetom (oče je piksel na za ena višjem nivoju, pri katerega izračunu je sodeloval trenutni piksel – sin), ki ima najbližjo vrednost njegovi lastni.

Korak 2: Preračunaj Gaussovo piramido od nivoja 1 do K, in sicer upoštevaj pri izračunu nove vrednosti očeta le tiste sinove, ki so povezani z njim (uporabimo lahko tudi uteži). Koraka 1 in 2 ponavljamo do konvergence.

Korak 3: Priredi vsem vozliščem vrednost, ki jo ima koren poddrevesa, v katerem so tudi sami (koreni so vozlišča na nivoju K).

Algoritem 2: Algoritem piramidnega povezovanja.

Z določanjem nivoja piramide K povemo, na koliko regij bomo sliko segmentirali. Popolnoma drugače pa se je lotil segmentiranja slik, predstavljenih z Gaussovo piramido, Song [62]. Začetno sliko najprej razdelimo na 4 podslike, v teh podslikah pa se na osnovi variance odločimo, ali je v podsliki rob ali ne. Podsliko, v kateri je rob, zatem predstavimo na višjem nivoju (Gaussova piramida) in postopek ponovimo. Na najvišjem nivoju piramide določimo prag za vse podslike s prisotnimi robovi, za vse ostale pa ta prag ocenimo. Zatem pa opravimo pragovno operacijo na osnovi teh robov.

V [63] najdemo splošen koncept segmentiranja hierarhično predstavljenih slik. Najprej uporabimo eno izmed dosedaj opisanih metod določanja pragov (podpoglavje 3.1.1) pri sliki, predstavljeni z najnižjo ločljivostjo (vrh piramide). V naslednjem koraku pa preciziramo segmentacijski rezultat – piksle v bližini mej vključimo ali v regije objekta ali ozadja. Ta korak ponavljamo za vsak nivo piramide do najvišje ločljivosti, kjer dobimo končno segmentacijo. Velika prednost takšnih metod je izredno majhen vpliv šuma na segmentacijske rezultate, saj imamo pri takšni segmentaciji opravka s filtriranimi slikami, kjer je šum izločen.

Burt je v [4] opozoril na nekaj stvari, ki jih moramo pri takšni segmentaciji pazljivo preučiti. Segmentacija je proces, kjer sliko razdelimo na regije, ki imajo bolj ali manj homogene lastnosti. Izračun teh lastnosti pa je odvisen od izida segmentacije. Problem v tej obojestranski odvisnosti se pojavi z izbiro operatorja w (enačba (3.12)) in velikostjo okna za izračun lastnosti. Okna morajo biti v primerjavi z elementi vzorca, ki prispevajo k izračunu lastnosti, dovolj velika, hkrati pa ne prevelika, da bi se ne prekrivala in tako zameglila meje med segmenti. V splošnem elementi vzorcev in velikosti segmentov niso znani vnaprej, velikost oken pa ponavadi tudi ni primerna za vsa področja v sliki. Burt predlaga, da naj bodo hierarhične metode zasnovane tako, da bodo iterativno preračunavale lastnosti in segmentirale sliko.

3.2 Segmentacijske metode na osnovi robov

To so najstarejše segmentacijske metode in temeljijo na veliki množici informacij o robovih v sliki. Segmentacija na osnovi robov deluje na robovih, ki jih dobimo iz slike z operatorji za detektiranje robov (Sobelov, Prewittov, Kirschev, Laplaceov, Cannyjev operator itd.). Dober opis teh operatorjev najdemo v [7, 27, 31, 60, 63]. Slike robov (to je slika, ki nastane kot rezultat uporabe detektorja robov) ne moremo direktno uporabiti kot segmentacijski rezultat, zato moramo te robove nekako še povezati v verige robov, ki bodo bolje ustrezale mejam v sliki. Končni cilj teh metod je doseči vsaj delno segmentacijo, kar pomeni, da bi združili lokalne robove v sliko, kjer bi verige robov z veliko verjetnostjo ustrezale objektom ali drugim delom slike.

Metode so izredno občutljive na količino vnaprejšnjega znanja, ki ga lahko vključimo v posamezen algoritem – več vnaprejšnjega znanja v splošnem zagotavlja boljše segmentacijske rezultate. Če je na voljo le malo takšnega znanja, potem ne moremo preveriti kvalitete segmentacijskih rezultatov in hkrati ni nobene povratne informacije za popravljanje teh rezultatov.

Glavna problema segmentacijskih metod na osnovi robov, ki nastaneta zaradi šuma ali nezanesljive informacije v sliki, sta prisotnost robov na področjih, kjer ni mej, in odsotnost robov, kjer te meje obstajajo.

3.2.1 Pragovna operacija na sliki robov

V sliki robov obstajajo majhne (neničelne) vrednosti robov, ki nastanejo zaradi nepomembnih sprememb v sivinah zaradi kvantizacijskega šuma, majhnih nepravilnosti v osvetlitvi itd. S pragovno operacijo (podpoglavje 3.1) lahko odstranimo te majhne vrednosti. Pragovna operacija nam lahko tudi klasificira robne piksle na pomembne (močne) in nepomembne (šibke), ali na svetle in temne (glej algoritem 1).

3.2.2 Sproščanje robov

Meje, ki jih dobimo z metodo, opisano v prejšnjem podpoglavju, so zaradi šuma v sliki zelo pokvarjene (velikokrat manjkajo pomembni deli meje). Če upoštevamo značilnosti robov v kontekstu njihove medsebojne sosednosti, potem lahko izboljšamo kvaliteto rezultirajoče slike. Vse značilnosti slike, vključno s tistimi, potrebnimi za nadaljnji obstoj robov, iterativno izračunavamo, dokler potek robov ni popolnoma jasen.

Metoda sproščanja robov (angl. edge relaxation) [63], ki si jo bomo ogledali, temelji na prelomnici (angl. crack edge) – to so robovi med dvema piksloma. Vrednost robnega piksla upoštevamo na obeh koncih prelomnice e, s tem pa hkrati definiramo minimalno sosesko roba (imejmo sliko 2x3 piksle, potem je rob med piksloma $(1, 2)$ in $(2, 2)$ prelomnica e, iz njega pa na obe strani izhajajo druge tri prelomnice – na levo: prelomnica med piksloma $(1, 1)$ in $(1, 2)$, $(1, 1)$ in $(2, 1)$ ter $(2, 1)$ in $(2, 2)$, podobno na desno). Vsaki prelomnici e priredimo tip in ga označimo s parom $i-j$, kar pomeni, da i prelomnic (z veliko vrednostjo) izhaja iz prelomnice e na levo, podobno za j. Metoda sproščanja robov, ki je iterativna metoda, pa na osnovi tipa prelomnice in vrednosti zaupanja roba $c(e)$ konvergira k nič (rob se je končal) ali k ena (rob tvori mejo). Zapišimo algoritem:

1. *Določi vrednost zaupanja $c^{(1)}(e)$ za vse prelomnice e v sliki (npr. kot normalizirano velikost prelomnice).*

2. *Poišči tip prelomnice na osnovi vrednosti zaupanja robov v njeni soseski. Vozlišče e je tipa i, če:*

   $$tip(i) = max_k(tip(k)), \qquad k = 0, 1, 2, 3, \qquad kjer$$

   $$\begin{aligned} tip(0) &= (m-a)(m-b)(m-c), \\ tip(1) &= a(m-b)(m-c), \\ tip(2) &= ab(m-c), \\ tip(3) &= abc, \end{aligned}$$

   pri čemer so a, b in c vrednosti prelomnic, ki izhajajo iz e na levo (desno), m pa je izbran kot $max(a; b; c; 0,1)$.

3. *Ažuriraj vrednost zaupanja $c^{(k+1)}(e)$ za vsak rob e glede na njegov tip in prejšnjo vrednost zaupanja $c^{(k)}(e)$. Tip 0-0 pomeni izoliran rob: negativen vpliv na vrednost zaupanja; 0-2 in 0-3 je mrtev konec: negativen*

vpliv, 0-1 pomeni nejasno: ni vpliva; 1-1 nadaljevanje: močan pozitiven vpliv; 1-2 in 1-3 pomenita nadaljevanje do sekanja meje: srednje pozitiven vpliv; 2-2, 2-3 ali 3-3 pomenijo most med mejami: ni vpliva. Vrednost zaupanja modificiramo:

$$\begin{array}{ll} \textit{vrednost naraste:} & c^{(k+1)}(e) = min(1, c^{(k)}(e) + \delta), \\ \textit{vrednost se zmanjša:} & c^{(k+1)}(e) = max(0, c^{(k)}(e) - \delta), \end{array}$$

kjer je δ majhna pozitivna konstanta, ponavadi med $0,1$ *in* $0,3$.

4. *Če $c^{(k+1)}(e) > T_1$, potem priredi $c^{(k+1)}(e) = 1$, če pa $c^{(k+1)}(e) < T_2$, potem priredi $c^{(k+1)}(e) = 0$.*

5. *Končaj, če so vse vrednosti zaupanja konvergirale ali k 0 ali k 1, sicer pa ponovi korake 2 in 3.*

Algoritem 3: Algoritem sproščanja robov.

Korak (4) je dodan za uravnavanje hitrosti sproščanja robov in natančnosti mej. Ta metoda je zelo učinkovita, hkrati pa omogoča tudi večkratno označevanje robov. Slabost te metode je velika časovna zahtevnost, zato v praksi raje izberemo hitrejše metode – npr. dinamično programiranje (podpoglavje 3.2.3).

3.2.3 Sledenje meji

Če meje regij niso znane, regije pa so v sliki definirane (npr. s pragovno operacijo), potem lahko meje enolično določimo. Pri mejah regije imamo opravka z več koncepti (definicijami) mej [31, 63]. Notranja meja regije je dejanska meja regije in je vedno del regije. Zunanja meja regije pa je meja ozadja in ni nikoli del regije (pomembna je pri opisu regij z raznimi značilnostmi). Podajmo psevdokod sledenja notranji meji regije:

1. *Preiskuj sliko iz levega zgornjega kota vrstico za vrstico, dokler ne naletiš na piksel, ki pripada novi regiji. Označi ta piksel s P_0. Piksel P_0 je začetni piksel meje regije. Spremenljivka dir hrani smer prejšnjega pomika vzdolž meje iz prejšnjega piksla do trenutnega piksla. Priredi:*

   a) *$dir = 3$, če odkrivamo mejo v 4-sosedstvu;*

   b) *$dir = 7$, če odkrivamo mejo v 8-sosedstvu.*

2. *Preišči* 3x3*-sosedstvo trenutnega piksla v protiurni smeri (možne smeri so podane na sliki 3.1), začni pri pikslu, ki je v smeri:*

   a) *$(dir + 3)$ mod 4 ali*

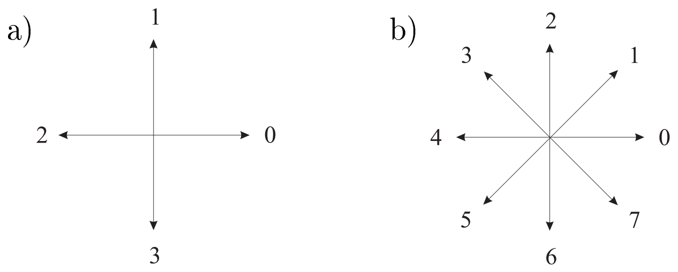
   b) *$(dir + 7)$ mod 8, če je dir lih in*
      $(dir + 6)$ mod 8 sicer.

Prvi piksel, ki ima enako vrednost, kot jo ima trenutni piksel, je novi element meje in ga označimo s P_n. Ažuriraj spremenljivko dir.

3. *Če je trenutni element meje P_n enak elementu meje P_1 in če je prejšnji element meje P_{n-1} enak P_0, potem KONEC, sicer ponovi korak 2.*

4. *Zunanja meja regije je določena s piksli $\{P_0, \ldots, P_{n-2}\}$.*

Algoritem 4: Algoritem sledenja notranji meji regije.



Slika 3.1: Vnaprej določene smeri za sledenje meji v a) 4- in b) 8-sosedstvu.

Algoritem sledenja (algoritem 4) lahko uporabimo tudi za sledenje zunanji meji. Psevdokod je naslednji:

1. *Sledi notranji meji regije v 4-sosedstvu (algoritem 4).*

2. *Zunanja meja je sestavljena iz vseh pisklov, ki ne pripadajo regiji in ki smo jih testirali med iskanjem notranje meje. Če je bil kakšen piksel testiran več kot enkrat, potem vse te ponovitve piksla vnesemo v izhodni seznam.*

Algoritem 5: Algoritem sledenja zunanji meji regije.

Oba algoritma za sledenje meji delata dobro za vse regije, ki so večje od enega piksla – kar pa je trivialen problem. S tema algoritmoma pa ne moremo najti mej lukenj znotraj regije.

Z obema tako definiranima mejama (notranjo in zunanjo) naletimo na problem pri sosednjih regijah, saj takšni regiji ne bosta imeli skupne meje (čeprav jo dejansko imata), kar lahko povzroča velike probleme na višjih nivojih obdelovanja (opis regij s parametri, zlivanje regij ipd.). To je bil razlog za uvedbo novega koncepta – razširjena meja, ki pa lahko definira mejo med sosednjimi regijami. Razširjena meja ima isto obliko in velikost kot naravna meja objekta, le da je premaknjena za 1/2 piksla navzdol in desno.

Videli smo, da je zelo preprosto slediti meji pri že definiranih regijah. Veliko težje pa je slediti meji, če imamo na vhodu le sliko robov (brez dodatnega znanja). V takšnih primerih moramo povezovati robove po nekem kriteriju (npr. podobne velikosti in smeri robov) v verige – npr. enojno povezovanje (glej uvod v podpoglavje 3.3). V mnogo realnih primerih imamo podano začetno in končno točko meje, takrat lahko sledimo meji z iskanjem po obteženem grafu, pri čemer so vozlišča grafa robni piksli (ki imajo velikost večjo od nekega praga), vozlišči pa sta povezani med seboj, če njuna smer ustreza lokalni smeri meje [63]. Uporabimo lahko razne strategije iskanja: iskanje po grafu na osnovi neke hevristične funkcije (algoritem A), razvijanje v širino ali globino, dinamično programiranje pa lahko uporabimo tudi, če informacija o začetni in končni točki ni na voljo ali pa če želimo poiskati več mej iz različnih začetnih točk.

Če slika sestoji iz objektov znanih oblik in velikosti, lahko na segmentacijo gledamo kot na problem iskanja določenih objektov v sliki (neke vrste ujemanje – glej podpoglavje 3.4). Učinkovita metoda na osnovi robov za iskanje določenih oblik (premic, krogov, elips) v slikah je Houghova metoda (poglavje 4).

Čim več vnaprejšnjega znanja o položaju ali obliki meje imamo, tem boljše lahko segmentiramo sliko. Znani so modeli aktivnih kontur (kače) [60], ki pričnejo iskanje meje tako, da izkoristijo vnaprejšnje znanje (ali znanje, ki ga posreduje uporabnik) o približnem položaju in obliki konture, optimizacijska metoda pa izboljša začetni približek konture in konvergira k želeni konturi.

3.2.4 Konstrukcija regije iz meje

Segmentacijske metode na osnovi robov le redko popolnoma segmentirajo sliko oz. popolnoma natančno določijo meje regij. Tako imamo velikokrat opravka z delno segmentacijo, kjer so definirane le delne meje regij. Obstajajo pa tudi metode za konstrukcijo regije iz delne meje [63].

Metoda, s katero se bomo seznanili, upošteva pri konstrukciji regije verjetnost za odločitev o tem, ali je piksel znotraj regije, ki jo opisuje delna meja. Piksel meje opišemo s položajem in smerjo roba $\Phi(x)$. Zaprto mejo konstruiramo iz para nasprotnih pikslov meje, pri čemer najbližji nasprotni piksel meje iščemo pravokotno glede na vsak pomemben rob v sliki. Piksel je potencialni element regije, če je na ravni črti, ki povezuje dva nasprotna piksla meje. Končna odločitev o potencialnem pikslu temelji na verjetnosti. Algoritem konstruiranja regije iz delnih mej:

1. *Za vsak piksel meje x poišči nasprotni piksel z roba znotraj maksimalne razdalje M. Če nasprotnega piksla nismo našli, poizkusi z naslednjim. Če smo nasprotni piksel našli, potem označi kot potencialni element regije vsak piksel, ki je na liniji, ki povezuje oba piksla meje.*

2. *Izračunaj število označb za vsak piksel (število označb pove, kolikokrat je bil piksel na linijah med dvema nasprotnima piksloma). Z $b(x)$ označi število označb za piksel x.*

3. Določi utež $B(x)$ po kriteriju:

$$
\begin{aligned}
B(x) &= 0,0, && \textit{če } b(x) = 0; \\
&= 0,1, && \textit{če } b(x) = 1; \\
&= 0,2, && \textit{če } b(x) = 2; \\
&= 0,5, && \textit{če } b(x) = 3; \\
&= 1,0, && \textit{če } b(x) > 3;
\end{aligned}
$$

Vrednost zaupanja, da je piksel x element regije, določimo z vsoto $\sum B(x_i)$ v 3x3-sosedstvu piksla x. Če je vrednost zaupanja 1 ali več, potem označi x kot element regije, drugače pa kot piksel ozadja.

Algoritem 6: Algoritem konstruiranja regije iz delnih mej.

3.3 Segmentacijske metode na osnovi regij

Cilj segmentacijskih metod, opisanih v prejšnjem podpoglavju, je bil poiskati meje med regijami, metode, opisane tukaj, pa konstruirajo regije direktno. Preprosto je skonstruirati regijo iz njene meje oz. poiskati mejo za obstoječo regijo. Segmentacijski rezultati metod na osnovi robov (podpoglavje 3.2) in metod rasti regij niso ponavadi popolnoma enaki, zato v [63] predlagajo, da bi združili rezultate obeh metod. Tehnike rasti regij so v splošnem primernejše za šumne slike, kjer še posebej težko detektiramo robove.

Segmentacijske metode na osnovi regij imenujemo tudi metode rasti regije (angl. region growing). Pri teh metodah sliko razdelimo na regije na osnovi nekega homogenostnega kriterija (npr. na osnovi povprečij sivin znotraj regij). Segmentacijo slike f za homogenostni kriterij H definiramo kot delitev slike f na nepovezane in neprazne podmnožice $P = \{R_1, R_2, \ldots, R_S\}$ tako, da velja [57, 63]:

$$
\begin{array}{lll}
(1) & f = \bigcup_i (R_i), & \\
(2) & \text{Regija } R_i \text{ je povezana}, & \forall i = 1, 2, \ldots, S, \\
(3) & H(R_i) = TRUE, & \forall i = 1, 2, \ldots, S, \\
(4) & H(R_i \, U \, R_j) = FALSE, & \forall i \neq j, \text{ kjer sta } R_i \text{ in } R_j \text{ sosednji regiji.}
\end{array}
\tag{3.13}
$$

Prvi in drugi pogoj pomenita, da mora vsak piksel pripadati regiji in da mora biti regija povezana. Tretji pogoj določa, da morajo regije biti homogene glede na neki kriterij, medtem ko s četrtim pogojem zahtevamo, da je velikost vsake regije maksimalna. Če delitev slike izpolnjuje vse štiri pogoje, potem je ta delitev optimalna segmentacija slike. Monga [50] trdi, da definicije (1) do (4) ne definirajo popolne segmentacije, zato je dodal še eno definicijo, in sicer:

(5) Izmed vseh segmentacij Seg, ki ustrezajo pogojem (1) do (4), poiščimo takšno segmentacijo Seg^*, ki optimizira pozitivno funkcijo kvalitete segmentacije C:

$$C(Seg^*) \leq C(Seg), \quad \forall Seg \in Seg_p(f),$$

kjer je $Seg_p(f)$ množica vseh možnih delitev slike f.

Izkaže se, da tudi ta novi pogoj ni dovolj, da bi dobili enolično segmentacijo, saj lahko ima več segmentacij isto minimalno vrednost funkcije C, kljub temu pa ta pogoj zagotavlja veliko bolj natančno definicijo segmentacije. Hitro lahko vidimo, da je optimizacijski problem, definiran s točko (5), problem nepolinomske zahtevnosti (NP), zato se v praksi zadovoljimo s podoptimalno rešitvijo.

Obstaja cela vrsta segmentacijskih metod rasti regije, ki razdelijo sliko na regije tako, da izpolnjujejo točke (1) do (4) iz enačbe (3.13). Zelo dobro je te metode razvrstil Haralick [26]. Najpreprostejše metode – enojno povezovanje – gledajo na vsak piksel kot na vozlišče v grafu. Sosednji piksli, katerih lastnosti (sivina) so si dovolj podobne, so povezane z lokom. Te metode so privlačne zaradi njihove preprostosti, vendar pa imajo problem z veriženjem, saj je za združitev dveh velikih regij dovolj le en lok (sosednja piksla sta si dovolj podobna, ali sta si zato podobni tudi obe regiji?). Nadgradnja te metode so tehnike hibridnega povezovanja, kjer skušamo vsakemu pikslu prirediti vektor značilnosti, pri čemer je ta vektor odvisen od KxK-okolice. Podobnost (in tudi povezovanje) je tako funkcija vrednosti sosednjih pikslov, zato se te metode obnašajo boljše pri šumnih podatkih. Pri centroidnem povezovanju, kjer sliko preglejujemo po vnaprej določenem vrstnem redu (npr. levo-desno, gor-dol), pa se vrednost piksla ne primerja več s sosednjim pikslom, ampak s povprečjem že obstoječega (ne nujno zaključenega) sosednjega segmenta. Piksel dodamo k najbližji regiji in ažuriramo njeno povprečje (ob tem lahko pride tudi do zlivanja dveh najbližjih sosednjih regij), če pa nobena regija ni dovolj blizu, lahko tvorimo nov segment s tem pikslom. Metode prostorskega grupiranja (angl. spatial clustering) določajo slikovne segmente s kombiniranjem tehnik grupiranja v merskem prostoru (histogramu sivin) in tehnik prostorske rasti regije. Tako npr. metoda iz te skupine najprej poišče vse vrhove v histogramu, zatem pa na teh pikslih opravi rast regije. Zadnja vrsta metod pa so metode delitev in zlitij, kjer najprej sliko iterativno delimo, dokler ne dobimo homogenih regij, zatem pa podobne regije zlijemo.

V nadaljevanju si bomo ogledali nekaj skupin segmentacijskih metod rasti regije, vendar pa bomo pri tem privzeli preprostejšo razdelitev metod [63]. Najprej si bomo ogledali metode zlitja regij (podpoglavje 3.3.1), kjer bomo zelo podrobno predstavili posamezne koncepte, saj so le-ti uporabni za vse metode rasti regije. Sledili pa bosta kratki podpoglavji z opisom metod delitev regij ter metod delitev in zlitij regij.

3.3.1 Zlitje regij

Vse segmentacijske metode zlitja regij lahko popišemo s splošnim algoritmom [63]:

1. *Definiraj začetno metodo, ki bo segmentirala sliko v mnogo majhnih regij, tako da je izpolnjen kriterij (3) iz enačb (3.13).*

2. *Definiraj kriterij za zlitje dveh sosednjih regij.*
3. *Zlij vse sosednje regije, ki zadovoljujejo kriterij za zlitje. Če ne moremo več zliti nobenega para regij, potem KONEC.*

Algoritem 7: Splošen algoritem zlitja regij.

Vse metode zlitja regij imajo enako ogrodje (algoritem 7), razlikujejo se le v definiciji začetne segmentacijske metode in kriterija zlitja. Rezultat metod zlivanja regij je ponavadi odvisen od vrstnega reda, po katerem zlivamo regije – tako je odvisno, ali segmentacijo začnemo v zgornjem levem kotu ali pa v spodnjem desnem kotu (v nadaljevanju si bomo pogledali tudi način zlitja, ki je neodvisen od vrstnega reda). Preprostejše metode zlivanja regij začnejo segmentacijo z začetnimi regijami velikosti 2x2 piksla (nekatere tudi le z enim pikslom, lahko pa tudi 3x3 piksle in več), pri tem pa opišejo regije s preprostimi statističnimi vrednostmi sivin (uporabimo lahko tudi lokalni histogram ipd.), na osnovi katerih naredimo preproste homogenostne kriterije. Kompleksnejše metode pa uporabljajo za segmentacijo zapletene algoritme, kot so npr. iskanje v prostoru stanj, nekatere pa tudi posebne podatkovne strukture – supermreže [63].

Homogenostni kriteriji

Dve sosednji regiji lahko zlijemo, če nova združena regija izpolnjuje homogenostni kriterij. Najpreprostejše metode uporabljajo kot kriterij zlitja primerjavo povprečnih sivin obeh regij, ki ju želimo zliti. Če sta obe povprečji dovolj blizu (potreben je segmentacijski prag), potem ti regiji zlijemo. Namesto kriterija vrednosti sivin lahko uporabimo razne statistike. Naj bo N število pikslov v regiji R s povprečjem $\overline{X}$ in standardnim odklonom S^2/N, y pa je sivina trenutnega piksla. Potem lahko definiramo statistiko T [26]:

$$T = \left(\frac{(N-1)N}{N+1} (y - \overline{X})^2 / S^2 \right)^{1/2}.$$

Če je T dovolj majhen, potem y dodamo k regiji in hkrati ažuriramo povprečje ter standardni odklon regije. Spet druga statistika ima obliko:

$$T = |y - \overline{X}_{nova}| - (1 - S/\overline{X}_{nova})\Theta,$$

kjer je Θ uporabniški parameter. Če je $T < 0$ za sosednjo regijo R, za katero velja, da je $|y - \overline{X}|$ minimalna, potem y dodamo k R, sicer pa začnemo novo regijo. Obe opisani statistiki je možno preprosto dopolniti za primere, kadar zlivamo dve pravi regiji (obe večji od enega piksla).

Odločitev za zlitje regij večinoma temelji na primerjavi razlike njihovih značilnosti z vnaprej definirano vrednostjo (segmentacijski prag) [9]. Določiti pravi segmentacijski prag je bistvenega pomena za segmentacijske metode na osnovi regij. V dosedaj znanih metodah se prag določi ali na osnovi vnaprejšnjega znanja, analize histograma ali pa je rezultat napornega poizkušanja (angl. trial-and-error process). Pri kompleksnejših slikah pa je za uspešno segmentacijo potrebnih celo več pragov. Chang [9] je razvil adaptivni homogenostni test za zlitje regij. Najprej si poglejmo idejo pri metodi s fiksnim pragom. Naj bosta R_i in R_j dve sosednji regiji, sestavljeni iz n oz. m primitivnih regij (npr. 2x2 pikslov). To pomeni, da je $R_i = \{X_1, \ldots, X_n\}$ in $R_j = \{Y_1, \ldots, Y_m\}$, kjer X_i in Y_j predstavljata vrednosti značilnosti primitivnih regij. Naj bosta $\overline{X}$ in $\overline{Y}$ povprečji značilnosti za R_i in R_j. Fiksni homogenostni test s pragom ϵ je definiran kot:

$$FIX(R_i \, U \, R_j) = \begin{cases} \text{TRUE,} & \text{če } |\overline{X} - \overline{Y}| < \epsilon \\ \text{FALSE} & \text{sicer} \end{cases} . \tag{3.14}$$

Če optimalen prag ϵ obstaja in ga je možno poiskati, potem lahko fiksni homogenostni test preprosto implementiramo.

Segmentacijski prag specificira, za koliko lahko značilnosti odstopajo znotraj regije. Zato je prag (parameter) odvisen od značilnosti in regije. V adaptivnem testu pa se odločamo na osnovi porazdelitve značilnosti dveh obravnavanih regij. Najprej za podan parameter λ (ponavadi $\lambda \in [0,80; 0,85]$) določimo adaptiven obseg (l_{i1}, l_{i2}) za regijo R_i, za katerega velja, da delež vseh značilnosti regije, določen z λ, leži znotraj njega, ali zapisano s formulami:

$$P(l_{i1} < X_j < l_{i2}) = \lambda \quad \text{in} \quad P(l_{i2} \leq X_j) = P(X_j \leq l_{i1}) = \frac{1-\lambda}{2}.$$

Ko imamo določena oba adaptivna obsega (za R_i in R_j), lahko opravimo adaptivni homogenostni test:

$$ADAP(R_i \, U \, R_j) = \begin{cases} \text{TRUE,} & \text{če } l_{j1} < \overline{X} < l_{j2} \text{ in } l_{i1} < \overline{Y} < l_{i2} \\ \text{FALSE} & \text{sicer} \end{cases} . \tag{3.15}$$

Načini zlivanja

Kocher [35] je predstavil zlivanje regij s pomočjo posebne podatkovne strukture grafa sosednosti regij RAG (angl. region adjacency graph), ki ga dobimo tako, da začetno sliko razdelimo na množico sosednjih homogenih kvadratov (pravokotnikov) (npr. velikosti 3x3 pikslov). Ti kvadrati predstavljajo vozlišča v grafu RAG. Povezavam pripišemo vrednosti mer različnosti regij. Zatem iterativno združujemo vozlišča, kjer je vrednost na povezavi minimalna. Združevanja je konec, ko je izpolnjen zaključni pogoj (npr. število regij v segmentirani sliki nad določenim pragom ali najmanjša mera različnosti, večja od določenega praga). Končni graf RAG preslikamo nazaj v matrično predstavitev – vsak piksel iz istega vozlišča v grafu RAG oštevilčimo z isto številko.

Kocher je v eni iteraciji združil vozlišči iz grafa RAG, med katerima je strošek zlitja minimalen. Popović [57] pa je predstavil še dva druga pristopa zlitja. V prvem pristopu ponovno združimo vozlišči z najmanjšo ceno zlitja, vendar sedaj povezave med preostalimi vozlišči in tem združenim odstranimo (v prvem primeru smo te povezave ažurirali). V tem manjšem grafu postopek ponavljamo, dokler ne moremo združiti več nobenega vozlišča. S tem se konča prva iteracija. V drugi iteraciji ponovno izračunamo vse vrednosti povezav in ves postopek ponavljamo, dokler ne dosežemo konvergence. Pri tem pristopu lahko združimo v eni iteraciji več vozlišč. Tretji pristop se razlikuje od prejšnjih dveh v tem, da se vozlišči za zlitje ne izbereta po vrednosti povezave med njima. Ko določimo potencialni par za zlitje, preverimo, ali vsaj eno izmed obeh vozlišč pripada drugemu potencialnemu paru za zlitje, že prisotnemu v grafu. Če to ne drži, potem pripišemo povezavi med vozliščema ceno zlitja, sicer pa se lahko pojavita dve možnosti:

- le eno od obeh vozlišč pripada potencialnemu paru, že prisotnemu v grafu. Če je vrednost nove povezave manjša od vrednosti povezave para, že prisotnega v grafu, potem obdržimo novo vrednost, staro pa zavrženo, sicer pa obratno;

- obe vozlišči pripadata potencialnima paroma, že prisotnima v grafu. Če je vrednost nove povezave manjša od obeh vrednosti povezav, že prisotnih v grafu, potem obdržimo novo povezavo, obe stari pa zavržemo, sicer pa obratno.

Po koncu tega prirejanja (za ves graf) vozlišča združimo. Tudi tukaj lahko v eni iteraciji združimo več vozlišč. Drugi pristop zmanjša potreben procesorski čas glede na konzervativen Kocherjev pristop. Obe metodi dajeta rezultate, ki so neodvisni od vrstnega reda, po katerem opravljamo segmentacijo, medtem ko je tretji pristop, ki je sicer najbolj primeren za paralelizacijo in je tudi sicer časovno najmanj zahteven, odvisen od vrstnega reda, po katerem združujemo pare. Ti pristopi v splošnem ne dajejo istih rezultatov, izkaže pa se, da so te razlike izredno majhne, če najprej opravimo delitev regije (podpoglavje 3.3.2) na originalni sliki, šele zatem pa zlitje.

Seveda pa vse metode zlitja regij ne uporabljajo tako kompleksnih podatkovnih struktur, kot je graf RAG, in tudi sicer niso vse metode tako zapletene. Spoznali smo že, da obstajajo metode, ki preglejuje sliko vrstico za vrstico in dodajajo po en piksel k že obstoječim regijam (torej zlivajo en piksel z regijo ali pa tvorijo novo regijo). Iz tega sklopa metod si bomo ogledali algoritem Fioria [16], ki je opravil rast regije s strategijo združi-išči, kjer podprogram Združi zliva oz. združuje dve regiji, podprogram Išči pa identificira možni podmnožici za zlitje. Prednost te strategije (v njeni preprosti izvedbi) je izredno preprosta implementacija in linearna časovna zahtevnost, odvisna od števila pikslov. Za popis regije uporablja drevesno strukturo – vsi piksli, ki pripadajo isti regiji, so v istem drevesu. Nerekurzivna oblika algoritma:

```
VHOD:  Slika f, velikosti k x l pikslov
       Posebej obdelaj prvo vrstico.
       FOR i = 2 TO k DO BEGIN
          Posebej obdelaj prvi piksel v vrstici i.
          FOR j = 2 TO l DO BEGIN
             levi = VrniKoren(f[i, j − 1]);
             zgornji = VrniKoren(f[i − 1, j]);
             ta = VrniKoren(f[i, j]);
             IF Homogen(levi,ta) THEN ta=Zdruzi(levi,ta);
             IF Homogen(zgornji,ta) THEN ta=Zdruzi(zgornji,ta);
```

END.
Vse piksle iz vrstice i direktno poveži z očeti.
END.

Algoritem 8: Zlitje regij s strategijo združi-išči.

3.3.2 Delitev regij

Metode delitve regije začnejo proces segmentacije, za razliko od metod za zlivanje regij (podpoglavje 3.3.1), s celotno sliko. Začetna slika je na začetku ena sama regija in ne izpolnjuje kriterija homogenosti (točka (3) enačb (3.13)). Zatem regijo (sliko) zaporedno delimo, dokler vsaka izmed podregij ne izpolnjuje kriterija homogenosti, hkrati pa mora biti izpolnjena še točka (4) iz enačb (3.13). Čeprav izgleda ta pristop dualen zlitju regij, se izkaže, da se rezultati segmentiranja z obema metodama lahko zelo razlikujejo, četudi uporabimo iste kriterije homogenosti. To se zgodi zaradi tega, ker so nekatere regije lahko homogene med procesom delitve in jih zato ne delimo več, medtem ko nekaterih homogenih regij s procedurami zlitja ne moremo konstruirati, ker majhnih podregij ne moremo zliti.

3.3.3 Delitev in zlitje regij

Te metode so kombinacija metod delitve in zlitja regij ter združujejo prednosti obeh metod. Pri segmentiranju slike uporabljajo njeno hierarhično predstavitev (piramide, štiriška drevesa) – regije pa imajo kvadratno obliko (pravokotno) in ustrezajo elementom na ustreznem nivoju hierarhične podatkovne strukture. Metode začnejo s celotno sliko kot začetnim segmentom. Če trenutni segment ni homogen, ga iterativno delimo na 4 kvadrate (pravokotnike) – ena delitev pomeni za ena višji nivo v hierarhični strukturi. Homogenostni kriteriji so lahko preprosti kriteriji na osnovi sivin ali kompleksnejši na osnovi določenih statistik (glej podpoglavje 3.3.1). Po končanem deljenju vseh segmentov (vse regije na vseh hierarhičnih nivojih so homogene) so meje med segmenti zelo kvadrataste oz. izgledajo zelo nenaravne (umetne). Zato sledi še drugi korak te metode – zlitje. Zlivamo po dve podobni sosednji regiji, pri čemer ni nujno, da obe regiji pripadata istemu očetu, niti ni nujno, da sta na istem hierarhičnem nivoju. Zlivanje končamo, ko ne moremo več zliti nobenih dveh sosednjih regij. V [63] najdemo opisanih nekaj metod delitev in zlitja regij.

Slike, segmentirane z metodami rasti regije, velikokrat vsebujejo zaradi neoptimalne nastavitve parametrov ali preveč regij (premala rast) ali pa premalo regij (prevelika rast). Za izboljšanje klasifikacijskih rezultatov je bilo razvitih mnogo metod poobdelovanja. Preprosto poobdelovanje temelji na splošni hevristiki in zmanjša število majhnih regij v segmentirani sliki, ki jih nismo mogli zliti glede na originalen kriterij homogenosti. Te majhne regije ponavadi niso pomembne v nadaljnjem obdelovanju in jih lahko obravnavamo kot segmentacijski šum. Odstranjevanje teh majhnih regij poteka tako, da jih vse združimo z najbolj podobnimi velikimi regijami. Nekatere bolj kompleksne metode poobdelovanja pa združujejo segmentacijske informacije, dobljene iz rasti regij in z metodami na osnovi robov. Primer takšnega obdelovanja je v [19], kjer najprej k začetni krpi, ki mora biti znotraj prave regije, dodajamo sosednje piksle, ki zadovoljujejo kriterij homogenosti. Rast regije kontroliramo s povprečnim gradientom meje trenutne regije. Ker se ta kriterij izračunava sproti (on-line), gre ta rast predaleč, zato opravimo še skrčitev regije, ki nam vrne optimalno regijo.

3.4 Segmentacija na osnovi ujemanja vzorcev

Ujemanje vzorcev je zelo važen koncept v segmentaciji. Uporablja se za lociranje znanih objektov in iskanje določenih vzorcev v sliki. Segmentacija na osnovi ujemanja je zelo preprosta tehnika, če imamo v sliki, ki jo obdelujemo, popolnoma enak vzorec, kot ga iščemo (ali kakšno izmed njegovih rotacij). V praksi je skoraj vedno določen del vzorca pokvarjen zaradi šuma in geometrijskih deformacij, zato ni možno iskati absolutnega ujemanja, ampak si raje pomagamo s kriteriji, ki nam vrnejo mesto maksimalnega ujemanja.

Kriterije ujemanja lahko definiramo na mnogo načinov. Največkrat uporabimo za kriterij korelacijo med vzorcem in sliko. Naj bo f vhodna slika, h vzorec in V množica pikslov v sliki. Potem lahko kriterij ujemanja na mestu (u, v) zapišemo:

$$
\begin{aligned}
C_1(u,v) &= \frac{1}{max_{(i,j)\in V}|f(i+u,j+v)-h(i,j)|}, \\
C_2(u,v) &= \frac{1}{\sum_{(i,j)\in V}|f(i+u,j+v)-h(i,j)|} \quad \text{ali} \\
C_3(u,v) &= \frac{1}{\sum_{(i,j)\in V}(f(i+u,j+v)-h(i,j))^2}.
\end{aligned}
$$

Za odkrivanje vzorcev, ki so v sliki rotirani, povečani ali pomanjšani, bi morali z metodami ujemanja pregledati sliko glede na vzorec v vseh možnih rotacijah in velikostih. To je časovno zelo zahtevna naloga. V takšnih primerih obstajajo nekateri posebni prijemi [60, 63].

3.5 Primerjava segmentacijskih metod

V prejšnjih podpoglavjih (od 3.1 do 3.4) smo se podrobno seznanili s segmentacijskimi metodami. V tem podpoglavju pa bomo med seboj primerjali nekaj segmentacijskih algoritmov iz posameznih skupin segmentacijskih metod. Vse algoritme bomo najprej preizkusili na umetno tvorjenih slikah, kjer bomo skušali razpoznati temne elipse na svetlem ozadju, zatem pa še na realnih, ultrazvočnih posnetkih jajčnika, kjer bomo iskali jajčne mešičke. Na koncu tega podpoglavja pa bomo podali kratko oceno o primernosti posameznega segmentacijskega algoritma.

3.5.1 Testno zaporedje slik

Preden se lotimo primerjave segmentacijskih algoritmov, moramo izbrati testne slike, na katerih bomo preizkušali njihovo uspešnost. V splošnem lahko izberemo določen nabor realnih slik in na njih ugotavljamo uspešnost posameznega algoritma. Vendar je takšno analiziranje algoritmov izredno nesistematično, saj za realne slike v splošnem ne vemo veliko o prisotnem šumu (tip in jakost šuma), ne poznamo točnega položaja objektov itd. Zato bi po takšnem analiziranju lahko dali le približno oceno o vsakem algoritmu.

Naše testno zaporedje je sestavljeno iz 50 umetno tvorjenih slik dimenzije 256x256 pikslov z 256 nivoji sivin. V vsaki sliki so temne elipse na svetlem (belem) ozadju. Elipse se ne prekrivajo med seboj, niti se ne dotikajo. Za vsako elipso imamo natančne podatke o položaju težišča, ploščini, velikosti obeh osi in sivini. Položaj težišča elipse, velikost obeh osi ter barvo (sivino) smo določili naključno. Velikosti obeh osi sta izbrani naključno iz intervala $[5, 64]$ pikslov, medtem ko je sivina elipse iz intervala $[0, 85]$. Maksimalno število elips v posamezni sliki smo omejili na 10. Tako tvorjenemu zaporedju slik (vsaka slika je brez šuma) smo zatem dodajali Gaussov šum in dobili novo zaporedje šumnih slik. Velikost dodanega šuma smo podajali z razmerjem med signalom in šumom (SNR) po naslednji formuli:

$$SNR = 10 \log \frac{E_{slika}}{E_{šum}}, \tag{3.16}$$

kjer E pomeni energijo slike oz. šuma. Iz osnovnega zaporedja slik smo z dodajanjem šuma dobili še 8 šumnih zaporedij slik. Prvo šumno zaporedje smo dobili tako, da je bilo SNR v vsaki sliki 40 dB, analogno pa je za ostala zaporedja SNR še 30 dB, 20 dB, 10 dB, 5 dB, 2 dB, 0 dB in -1 dB.

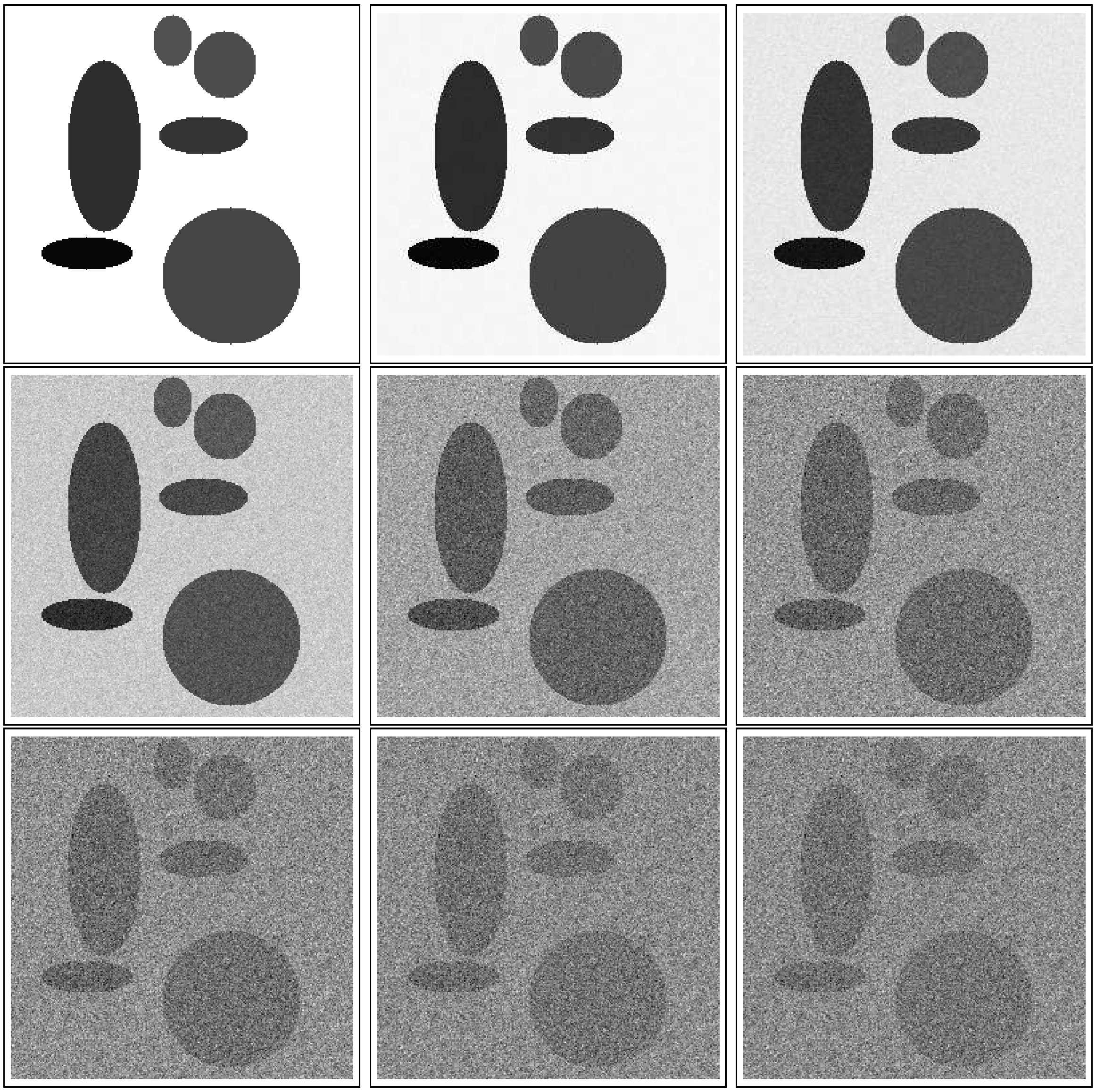
Na sliki 3.2 lahko vidimo primer takšnih umetnih slik. V zgornjem levem kotu je originalna slika, vsako naslednjo sliko dobimo z dodajanjem ustreznega šuma.

3.5.2 Izbrani segmentacijski algoritmi in opis poteka testov

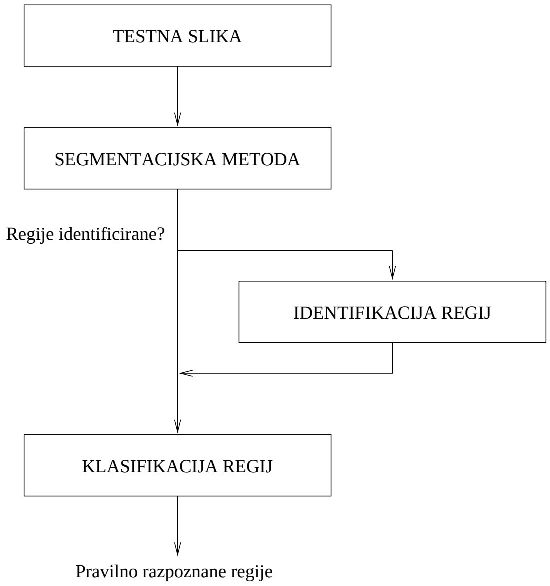
Iz množice segmentacijskih metod smo si za naše preizkušanje izbrali naslednje segmentacijske algoritme: MEAN, MEDIAN, MOMENTI, OPTIMAL, MINERROR(I) in ENTROPIJA, vse pragovne operacije z globalnim pragom (podpoglavje 3.1.1), večpragovno operacijo avtorja Wanga (podpoglavje 3.1.3), pragovno operacijo H2Y z variabilnim pragom avtorja Marina (podpoglavje 3.1.2), ter dve segmentacijski metodi rasti regij, in sicer na osnovi grafa RAG in strategije združi-išči (podpoglavje 3.3.1).

Vse te segmentacijske algoritme smo najprej preizkusili na umetnih slikah, dejansko smo opravili neke vrste simulacijo, saj smo z dodajanjem šuma ustvarjali možne realne situacije, zatem pa smo te algoritme preverili še na realnih ultrazvočnih posnetkih jajčnika.

Ker želimo za posamezen segmentacijski algoritem imeti natančne rezultate (koliko vseh regij nam vrne, koliko je pravilno razpoznanih, za koliko se razlikujejo razpoznane regije od originalnih, kakšen je vpliv šuma), smo vse teste zasnovali po shemi, ki je prikazana na sliki 3.3.



Slika 3.2: Testne slike: v zgornjem levem kotu je originalna testna slika s 6 elipsami (sivine elips: 7, 45, 52, 69, 77 in 80), desno od nje slika, kjer je originalni sliki dodan Gaussov šum s SNR 40 dB, zatem 30 dB, 20 dB, 10 dB, 5 dB, 2 dB, 0 dB, v desnem spodnjem kotu pa je originalni sliki dodan šum s SNR -1 dB. Vse slike so dimenzije 256x256 pikslov v 256 sivinah.



Slika 3.3: Shema testiranja segmentacijskega algoritma

Vsako sliko iz testnega zaporedja slik (slika sivin) najprej segmentiramo z enim izmed segmentacijskih algoritmov. Rezultirajoča segmentirana slika je v primeru pragovnih operacij z globalnim pragom in H2Y binarna slika, v vseh ostalih primerih (večpragovna operacija in metodi rasti regije) pa dobimo sliko sivin. S tem korakom smo opravili segmentacijo in naša naloga bi lahko bila končana, ker pa želimo dobljene segmentacijske rezultate tudi ovrednotiti, smo testiranju dodali še korak poobdelave. V nadaljevanju vse regije iz segmentirane slike identificiramo (označimo) z algoritmom za iskanje temnih pik [63] – rezultat je slika sivin, kjer vsaka sivina pomeni eno regijo. Vsako regijo zatem klasificiramo ali kot pravilno, torej če ustreza objektu na sliki, ali kot napačno. Klasificiranje poteka tako, da regijo najprej opišemo s parametri (ploščina, težišče, meja itd.). Tako opisano regijo primerjamo z najbližjim objektom glede na razdaljo med težiščema (naj še enkrat opozorimo, da so podatki o natančnem položaju objekta – elipse in njegovi ploščini ter težišču vnaprej znani – so v datoteki). O tem, ali je regija pravilno razpoznana, se odločamo na osnovi izračunanega preseka med regijo in objektom. Če je razmerje med tem izračunanim presekom in originalnim objektom, pomnoženo s 100, večje ali enako postavljenemu pragu (pragi oz. kriteriji so 100, 90, 80, 70, 60 in 50) in če je razmerje med presekom in regijo večje od določenega pragu, potem je regija razpoznana kot pravilna (npr. regija je razpoznana pri kriteriju 70, kar pomeni, da regija prekriva 70 % elipse), sicer pa kot napačna.

S tem ko smo definirali, kaj za nas pomeni razpoznana regija, lahko merimo uspešnost in učinkovitost posameznega segmentacijskega algoritma. Za posamezno testno sliko vodimo dve vrsti statistik: 1) odstotek razpoznanih objektov (elips) – to dejansko pomeni razmerje med številom razpoznanih elips in številom vseh elips v posamezni sliki pri določenem kriteriju, 2) odstotek razpoznanih regij – pomeni razmerje med številom vseh pravilno razpoznanih regij in vseh regij. Pri omenjenih statistikah ločimo med regijo in elipso. To razlikovanje smo vpeljali, ker lahko več regij razpozna eno in isto elipso. Ta dodaten podatek je posebej pomemben pri razpoznavanju elips pri nižjih kriterijih (kriterijih pod 50).

Tudi segmentacijske algoritme bomo ovrednotili s tema statistikama, le da bomo prej izračunali povprečje teh statistik skozi celotno testno zaporedje slik. S tem bomo dejansko za posamezne segmentacijske algoritme merili celovito senzitivnost (razmerje med prav razpoznanimi in vsemi obstoječimi objekti) in pa senzitivnost za obliko elipse (razmerje med regijami, ki so elipse in vsemi predlaganimi regijami). Ti statistiki bomo predstavili s krivuljama ROC (angl. receiver operating characteristic), kjer bodo pri obeh krivuljah na abcisni osi kriteriji (razmerje med presekom in originalnim objektom), na ordinatni osi pa bo odstotek razpoznanih elips oz. odstotek razpoznanih regij.

3.5.3 Rezultati

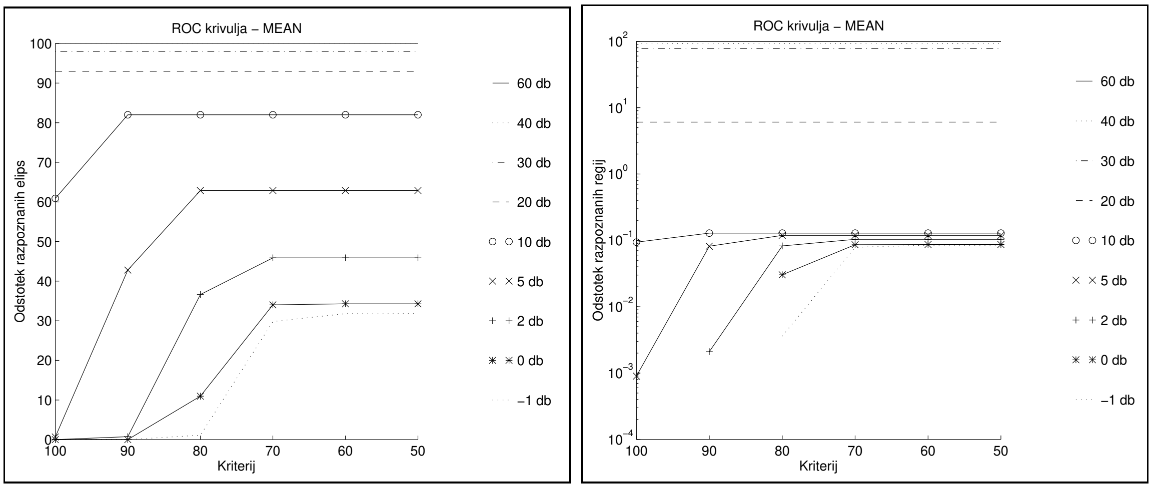
V prejšnjih dveh podpoglavjih smo natančno opisali, s kakšnimi testnimi slikami želimo preizkušati naše segmentacijske algoritme. V tem podpoglavju pa bomo podrobno opisali simulacijske rezultate.

Algoritem MEAN

Najprej bomo predstavili rezultate, dobljene s pragovnimi operacijami, ki pri segmentaciji uporabljajo globalni prag. Najpreprostejša takšna pragovna operacija je MEAN (podpoglavje 3.1.1). Operacija MEAN določa globalni prag kot povprečje vseh pikslov v sliki. Na sliki 3.4 prikazujeta dobljene rezultate krivulji ROC – odstotek razpoznanih elips in regij.

V krivulji ROC, ki prikazuje odstotek razpoznanih elips, vidimo, da je v slikah, kjer je SNR pri dodanem šumu manj kot 10 dB, odstotek razpoznavanja skoraj 100 % (če smo eksaktni: pri slikah s SNR 20 dB je ta odstotek 93 %). Ker imamo opravka z metodo, ki pri določitvi pragov uporablja informacije iz histograma sivin, je smiselno, da si pogledamo tipično obliko histograma za testne slike (pri posameznem šumu). Na sliki 3.5 so pripadajoči histogrami sivin za testne slike s slike 3.2. Iz te slike lahko jasno vidimo, da se elipse (so temne in imajo sivine pod 80) v slikah s SNR do 10 dB zelo dobro ločijo od ozadja (je svetlo in s sivinami okoli 255), oz. povedano drugače, porazdelitveni funkciji za elipse in za ozadje se ne prekrivata. Čeprav smo sklepali le na osnovi enega histograma pri določenem šumu, lahko kljub temu trdimo, da zgornje velja za vsako sliko iz testnega zaporedja.

Algoritem MEAN pa kljub dobri razločitvi elips in ozadja v slikah, kjer je na primer SNR pri dodanem šumu 20 dB, ne razpoznava elipse stoodstotno. Razlaga za to je preprosta. V testnih slikah, kjer imamo le eno majhno elipso (zelo majhno število sivin, različnih od ozadja), pri izračunu globalnega praga prevlada ozadje, to pomeni, da globalni prag pade na začetek porazdelitvene funkcije ozadja. Ko segmentiramo sliko s takšnim



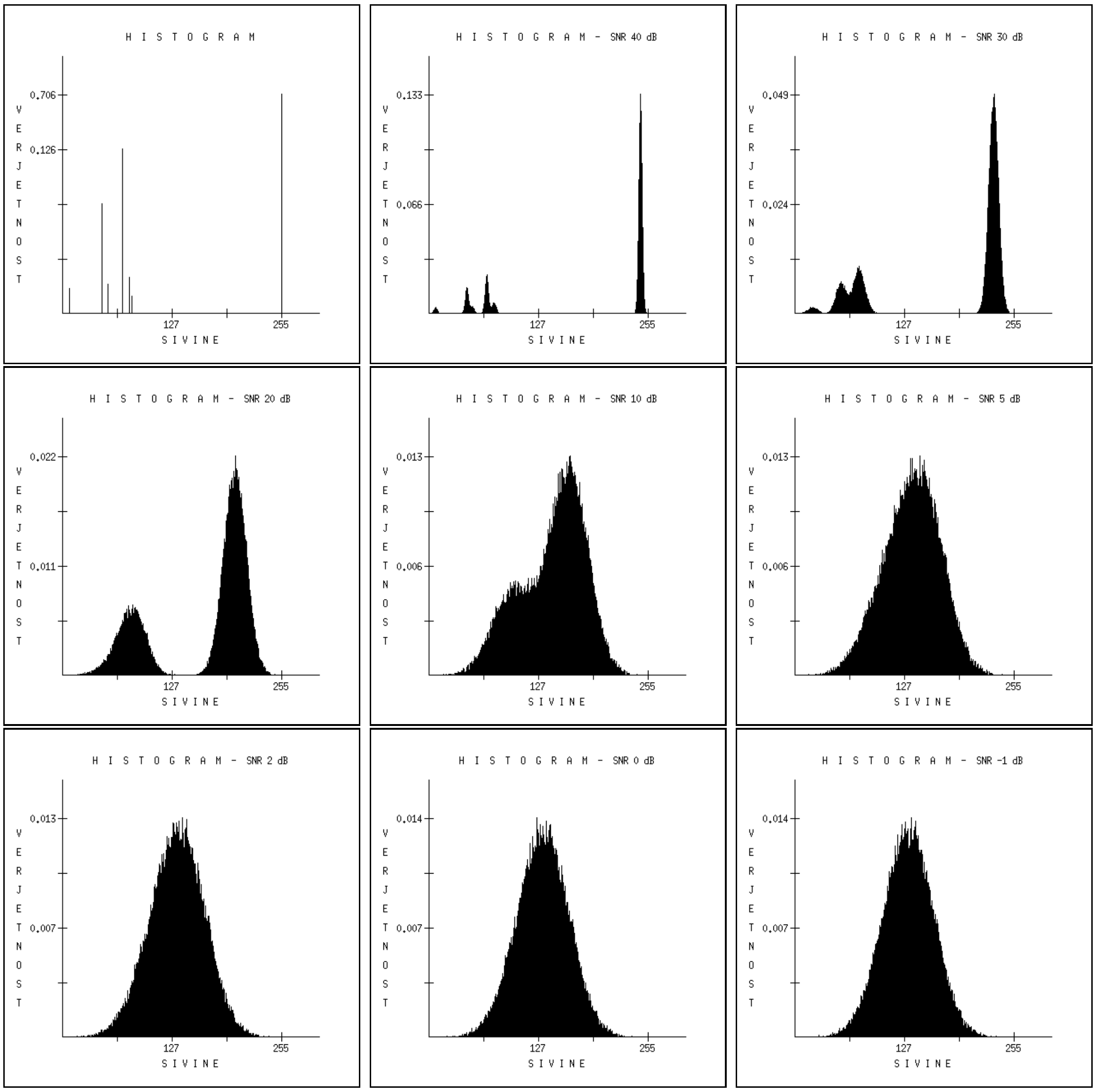
Slika 3.4: Krivulji ROC za segmentacijski algoritem MEAN. Ordinatna os z odstotkom razpoznanih regij je zaradi boljše preglednosti v logaritemski skali.

pragom, nam ostane v rezultirajoči segmentirani sliki še veliko šuma (šum se lahko tudi pripoji k elipsi in deformira njeno obliko), kar lahko povzroči napačno identifikacijo takšne elipse.

Poglejmo si še rezultate (slika 3.4) za šumne testne slike, kjer dodan šum raste (SNR raste od 10 dB do -1 dB). Vidimo, da odstotek razpoznanih elips pada z dodajanjem šuma, kar je seveda logično. V slikah s SNR -1 dB je ta odstotek 31 %. Iz pripadajočih histogramov (slika 3.5) opazimo, da lahko le v slikah, kjer je SNR pri dodanih šumih največ 10 dB, z gotovostjo določimo prag. Z algoritmom MEAN smo dobili dobre rezultate tudi pri višjem šumu, kar pa je bolj posledica izbire naših testnih slik (zahtevali smo, da so elipse temne – podpoglavje 3.5.1). Algoritem MEAN namreč določa za močno šumne slike (SNR od 5 dB do -1 dB) vse globalne prage v okolici sivine 127, kar je ravno polovica histograma, in ker so elipse temnejše, jih s takšnim pragom uspešno segmentiramo.

Na sliki 3.4 lahko vidimo tudi krivuljo ROC, ki prikazuje odstotek pravilno razpoznanih regij. S to krivuljo na neki način popisujemo, koliko šuma je segmentacijski algoritem sposoben odstraniti. Zaradi boljše preglednosti smo ordinatno os (odstotek razpoznanih regij) prikazali v logaritemski skali. Algoritem MEAN uspe šum v slikah, kjer je SNR do 20 dB, skoraj v celoti odstraniti, v slikah, kjer je SNR od 20 dB do -1 dB, pa število identificiranih regij strmo naraste (seveda je večina teh posledica šuma). Pri slikah s SNR -1 dB je tako odstotek razpoznanih regij v povprečju 0,0843 %, kar pomeni, da imamo v povprečju opravka s približno 2000 regijami (v testnih slikah smo omejeli število elips na 10). Opisani rezultati so dobljeni za kriterij 50 (ploščina preseka med regijo in elipso, deljena s ploščino elipse in pomnožena s 100, je večja ali enaka 50).

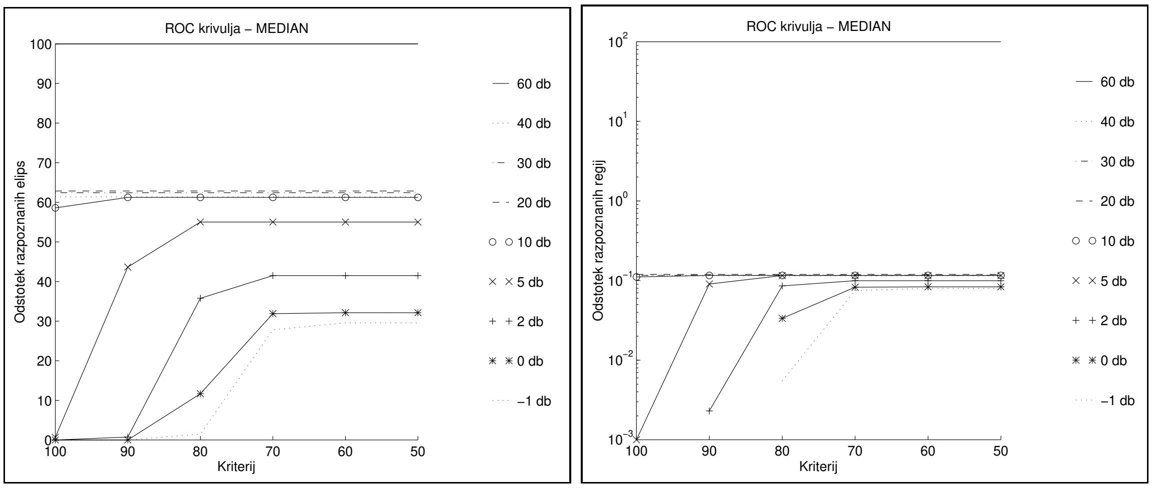
V nadaljevanju bomo podali rezultate še za preostale pragovne operacije. Mnogo trditev, ki smo jih navedli za algoritem MEAN, bo veljalo tudi za te algoritme in jih zato ne bomo ponavljali (sklepi o histogramu ipd.).



Slika 3.5: Pripadajoči histogrami sivin za testne slike s slike 3.2.

Algoritem MEDIAN

Algoritem MEDIAN (podpoglavje 3.1.1) izbere za globalni prag tisto sivino, kjer je vsota pikslov iz histograma sivin (od 0 do trenutnega praga), najbližja polovici vseh pikslov v sliki. V sliki 3.6 so prikazani rezultati za algoritem MEDIAN.



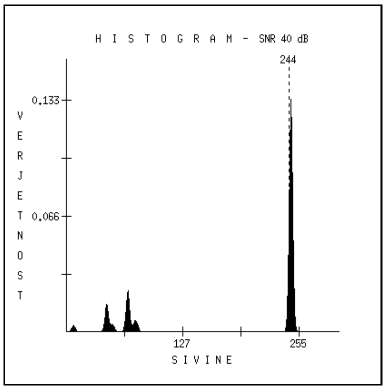
Slika 3.6: Krivulji ROC za segmentacijski algoritem MEDIAN.

Iz krivulje ROC, kjer je prikazan odstotek razpoznanih elips, opazimo, da ima algoritem MEDIAN izredno nizek odstotek razpoznavanja pri zelo šibkem dodanem šumu – v slikah s SNR 40 dB je ta odstotek le 61 %. Razlog za to lahko ponovno iščemo v obliki histograma oz. v strukturi testne slike. Ker vsota ploščin vseh elips v večini primerov ne dosega polovice velikosti slike, je prag določen (globoko) znotraj porazdelitvene funkcije ozadja. Ta primer lahko vidimo na sliki 3.7.

Če dodamo k slikam več šuma, potem algoritem MEDIAN določa globalne prage v okolici sivine 127. Razlog za takšno obnašanje je naslednji: slikam dodajamo Gaussov šum, porazdelitev le-tega pa je simetrična glede na sivino, ki je v okolici 127 (glej histograme na sliki 3.5). V slikah s SNR -1 dB je odstotek razpoznanih elips 29 % pri kriteriju 50. Na sliki 3.6 je tudi ROC-krivulja, ki prikazuje odstotek razpoznanih regij. Vidimo, da ta algoritem ne uspe odstraniti šuma (četudi je zelo majhen), kar se kaže v zelo nizkem odstotku razpoznanih regij. V slikah s SNR -1 dB je ta odstotek 0,0798 %.

Algoritem MOMENTI

Nato smo testirali pragovno operacijo MOMENTI (podpoglavje 3.1.1). Ideja te metode je, da mora segmentirana binarna slika imeti prve tri momente enake kot originalna slika. Na sliki 3.8 smo prikazali uspešnost tega algoritma. Iz ROC-krivulje za odstotek razpoznanih elips opazimo, da algoritem MOMENTI skoraj 100 % razpoznava elipse v slikah, kjer je SNR manj kot 10 dB. Presenetljiv pa je rezultat, da je pri nešumnih slikah odstotek

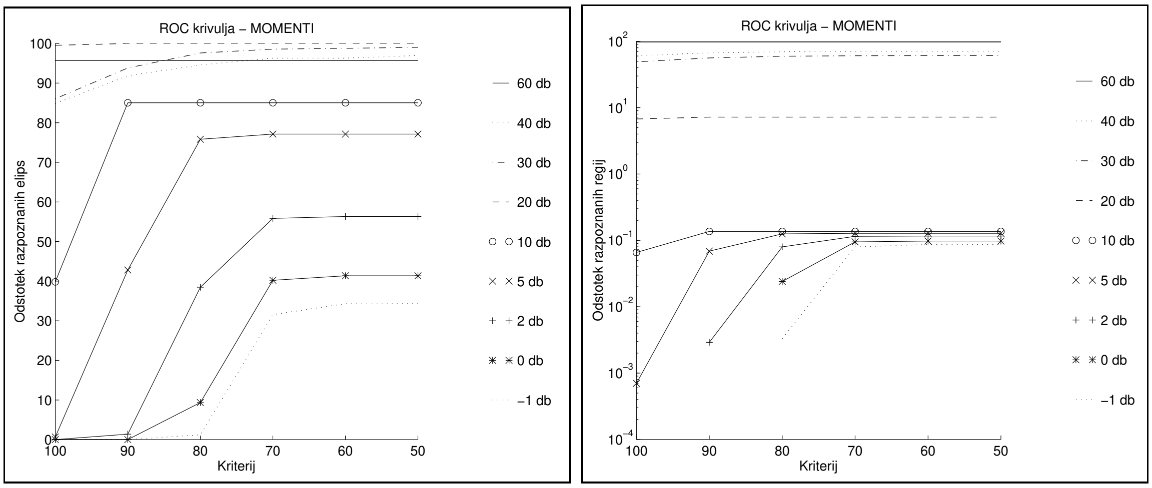


Slika 3.7: S segmentacijskim algoritmom MEDIAN določen globalni prag za testno sliko s slike 3.2. V sliki je razmerje SNR 40 dB.

razpoznavanja le 96 %. Algoritem MOMENTI določa globalen prag podobno kot MEDIAN, le da se tukaj vsota pikslov (od 0 do trenutnega praga) približuje x_0, ki pomeni delež vseh pikslov v sliki, in ne nujno 0,5, kot pri MEDIAN-i. Podrobno smo preučili določanje deleža x_0 in opazili, da dela ta algoritem največje napake ravno v slikah, kjer je le malo različnih sivin (vendar več kot 2) in kjer je število pikslov z določeno sivino majhno. Ta primer ustreza ravno našim brezšumnim testnim slikam – temne elipse (od tega ena izredno majhna) na svetlem ozadju (to da skupaj okrog 10 različnih sivin). V takšnih primerih ima delež x_0 vrednost, ki je ravno na meji med vsoto pikslov iz $n-1$ in n elips (vsoto pikslov do praga T_1 in T_2, ki sta lahko tudi daleč narazen), pri čemer je n število vseh elips v sliki. Pri tem sta razliki med x_0 in prvo vsoto, deljeno s številom vseh pikslov v sliki, ter x_0 in drugo vsoto, prav tako deljeno s številom vseh pikslov, skoraj enaki, vendar prevladuje prva vsota (s tem pa tudi prenizek prag T_1), tako da razpoznamo zadnjo, n-to elipso kot ozadje.

V testnih slikah, kjer se pojavi ta neželeni efekt (ena izmed teh je na sliki 3.2 zgoraj levo), smo spreminjali ploščine elips pri istih sivinah (naredili smo razne kombinacije teh ploščin) in v vseh primerih smo napačno segmentirali najbolj svetlo elipso (ta sivina je bila vedno pod 100). Zatem pa smo pri istih ploščinah spreminjali razliko med najtemnejšo in najsvetlejšo elipso. Z zmanjšanjem te razlike nam je uspelo pravilno segmentirati vse elipse, četudi je bila katera izmed njih izredno majhna (npr. zavzemala je le 4 % ploščine glede na vse elipse v sliki).

Pri šumnih slikah zgoraj opisani problem ne pomeni tako velike nevarnosti za napačno segmentacijo, saj je lahko napaka v določitvi praga velika le 1 sivi nivo, to pa v šumnih slikah, ki pokrivajo celotno območje histograma sivin (vse sivine), ne pomeni skoraj nič. Zato vidimo v sliki 3.8, da je odstotek razpoznanih elips v slikah, kjer je SNR 20 dB, 100 % (pri kriteriju 50). Pri močno šumnih slikah tudi algoritem MOMENTI vrača globalne prage, ki so v okolici sivine 127. Odstotek razpoznanih elips v slikah s SNR -1 dB je 34 %. Krivulja ROC za odstotek razpoznanih regij je tipična (z dodajanjem šuma pada tudi

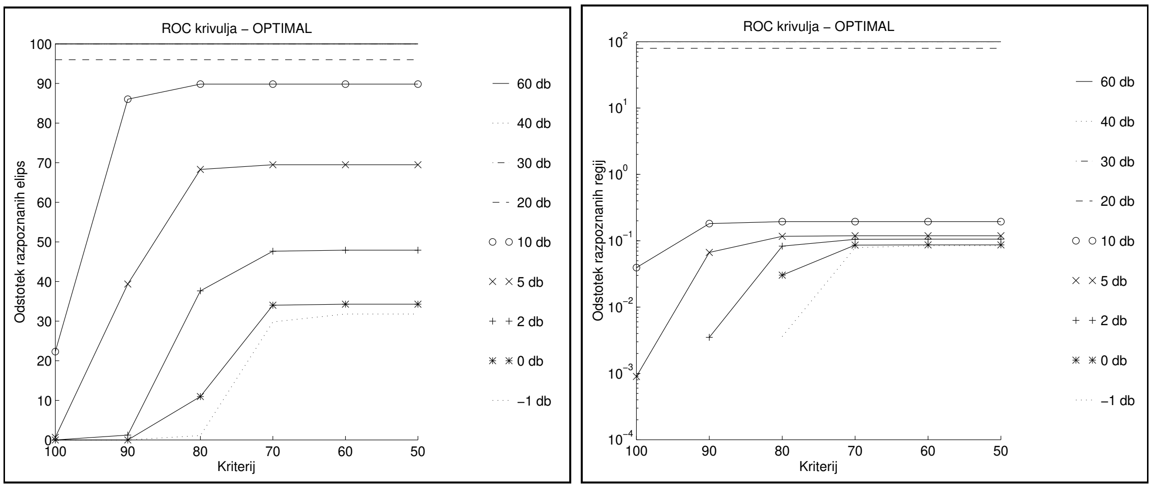


Slika 3.8: Krivulji ROC za segmentacijski algoritem MOMENTI.

odstotek razpoznavanja). V slikah s SNR -1 dB je ta odstotek 0,0976 %.

Algoritem OPTIMAL

Ta algoritem iterativno določa globalni prag na polovici med povprečno sivino objektov in ozadja (podpoglavje 3.1.1). Za začetni približek smo uporabili globalni prag, dobljen z algoritmom MEAN. Rezultati so prikazani na sliki 3.9.



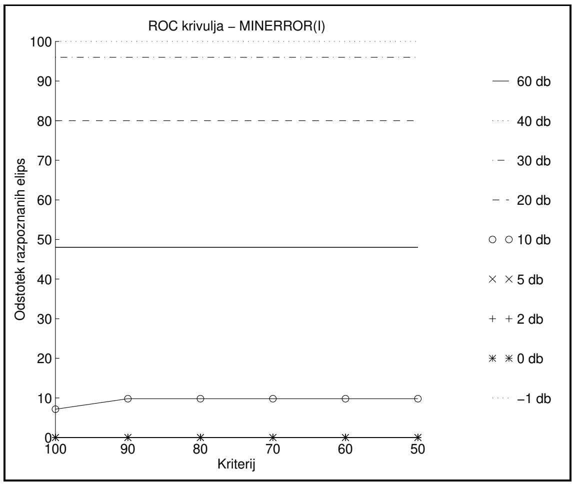
Slika 3.9: Krivulji ROC za segmentacijski algoritem OPTIMAL.

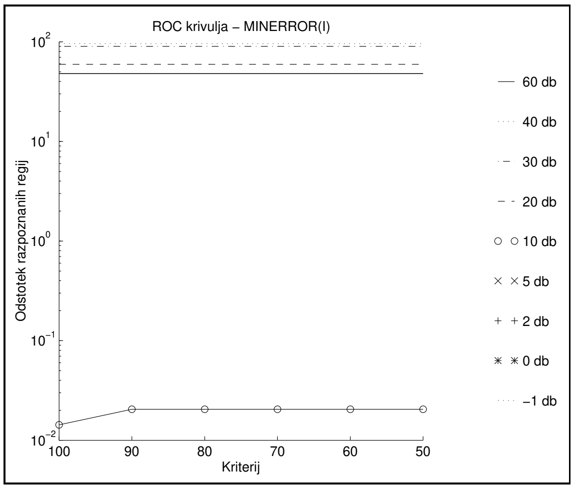
Iz krivulje ROC vidimo, da je odstotek razpoznanih elips za slike, kjer je SNR največ 20 dB, skoraj 100 % (pri SNR 20 dB je 96 %). Tudi pri slikah s SNR 10 in 5 dB dobimo zelo dobre rezultate, pri večjem šumu pa odstotek razpoznavanja strmo pade – pri slikah s SNR -1 dB je le še 32 %. Pojav razlagamo s tem, da sta pri slikah, kjer je SNR 10 dB, obe porazdelitveni funkciji (objektov in ozadja) še zelo dobro ločljivi (glej histograme na sliki 3.5), algoritem OPTIMAL pa določi globalni prag ravno na presečišču obeh funkcij. Pri večjem šumu tega presečišča ni več možno določiti in zato algoritem vrača prage v okolici sivine 127.

Algoritem OPTIMAL nima samo visokega odstotka razpoznanih elips pri majhnem dodanem šumu, ampak dobro opravi celo s samim šumom, saj v slikah, kjer je SNR do 20 dB, s segmentacijo ne pridobimo nobene dodatne regije, nastale zaradi šuma (to pomeni, da ta metoda uspe šum v celoti izfiltrirati). To lahko vidimo na krivulji ROC, kjer so prikazani odstotki razpoznanih regij (slika 3.9). Odstotek z dodajanjem šuma seveda pada, v slikah s SNR -1 dB je samo 0,0843 %.

Algoritem MINERROR(I)

Naslednji algoritem, ki smo ga implementirali in preizkusili, je MINERROR(I). Tudi ta je iterativen in potrebuje začetni približek (za začetni približek smo izbrali rezultat algoritma MEAN). Algoritem iterativno določa prag kot celi del večje rešitve kvadratne enačbe (glej podpoglavje 3.1.1) – ob predpostavki, da rešitev obstaja.





Slika 3.10: Krivulji ROC za segmentacijski algoritem MINERROR(I).

Na sliki 3.10 vidimo dobljene rezultate. Vidimo lahko, da je odstotek razpoznanih elips v slikah, kjer ni šuma, le 48 %. To je izredno slab rezultat, vendar ta rezultat zajema tudi primere, kjer rešitev ne obstaja (v slikah, kjer rešitev obstaja, so elipse razpoznane 100 %). Zaradi tega smo algoritem MINERROR(I) še bolj podrobno preučili. Opazili smo, da ima ta algoritem največje probleme v primerih, kjer sta ali σ^2 (standardni odklon vseh objektov) ali τ^2 (standardni odklon ozadja) enaka 0 – to je takrat, kadar so vsi objekti homogeni ali kadar je homogeno ozadje (enačba (3.5)). V primerih, ko so le objekti homogeni, dobimo smiselne rezultate (seveda, če prej odpravimo ulomke v enačbi (3.5)), in sicer ima globalni prag vrednost sivine objektov. V drugem primeru, ko je ozadje homogeno, ima prag sivino enako kot ozadje; v tem primeru je celotna slika segmentirana kot ozadje (napačna segmentacija). Najtežji primer za ta algoritem pa je (čeprav je sicer najpreprostejši), kadar so tako objekti kot ozadje homogeni (ena elipsa na belem ozadju). Tedaj nam enačba (3.5) vrne neskončno rešitev (obe strani enačbe sta enaki nič). Ta algoritem je treba modificirati za brezšumne slike, in sicer, če je le ozadje homogeno, potem vrniti za ena manjši prag, kot je ozadje, če pa so tako objekti kakor ozadje homogeni, potem je najbolje določiti prag kar na polovici med objekti in ozadjem. Tako spremenjen MINERROR(I) bi elipse razpoznal 100 % tudi v brezšumnih slikah.

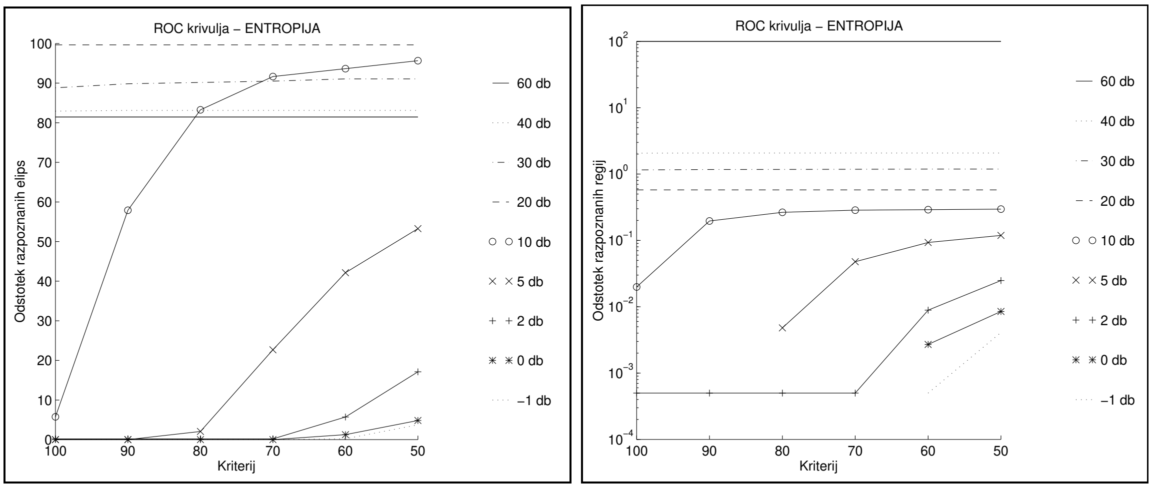
Pri slikah, kjer smo dodali nekaj šuma (SNR od 40 dB do 20 dB), je ta segmentacijski algoritem vrnil zadovoljive rezultate (odstotek razpoznanih elips se giblje od 100 % do 80 %). Ti rezultati so tudi potrdili trditve Glasbeya [20], da je MINERROR(I) najbolj primeren za segmentiranje slik, katerih histogrami imajo obliko dveh Gaussovih krivulj (eno za objekt(e), drugo za ozadje). V slikah, kjer je SNR 10 dB, je odstotek razpoznanih elips le še 9,8 %, medtem ko v slikah s SNR od 5 dB do -1 dB ne razpoznamo nobene elipse več (v teh primerih dve Gaussovi krivulji nista več vidni, MINERROR(I) pa vrača prage, ki so previsoki).

Algoritem ENTROPIJA

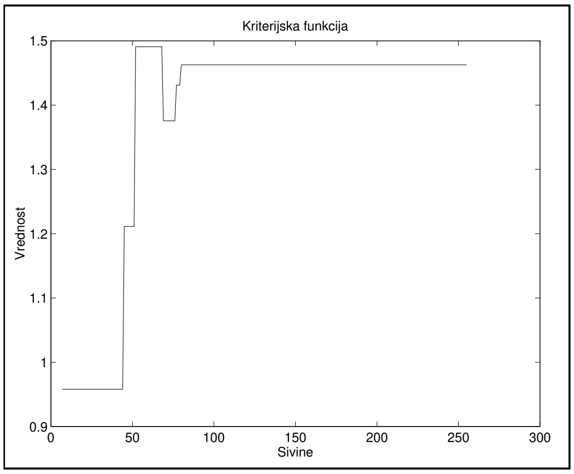
Zadnja, ki si jo bomo ogledali iz sklopa pragovnih operacij, je ENTROPIJA (podpoglavje 3.1.1). Ta operacija določa globalne prage na osnovi entropije (informacije) med verjetnostnima porazdelitvama sivin objektov in ozadja. Kriterijska funkcija Ψ (vsota entropij objektov in ozadja) ima obliko narobe obrnjene črke u (pri polnem histogramu), zato prag določamo na mestu, kjer ima funkcija maksimum. Rezultati, dobljeni s segmentacijskim algoritmom ENTROPIJA, so prikazani na sliki 3.11.

Med preizkušanjem tega algoritma smo videli, da določa nizke prage. To je posledica težnje ENTROPIJE, da bi bila informacija med objekti in ozadjem maksimalna (dvom oz. nejasnost informacije minimalna). Če imajo elipse sivine, ki so si zelo blizu, potem jih ENTROPIJA pravilno segmentira, sicer pa se zgodi, da najsvetlejše elipse razpozna kot ozadje (to velja za brezšumne slike). Ta efekt lahko vidimo na ROC-krivulji za odstotek razpoznanih elips za brezšumne slike, kjer odstotek znaša le 81 %. Na sliki 3.12 vidimo potek kriterijske funkcije za brezšumno testno sliko. Opazimo strm padec pri sivini 69, zatem pa rahel dvig (pri sivini 80). Prag je določen prenizko (52), saj s tem pragom razpoznamo tri elipse kot ozadje (elipse s sivinami 69, 77 in 80). Razpon med najtemnejšo in najsvetlejšo elipso je prevelik (73), zato pri pragu 52 dobimo več informacije kot pri pravilnem pragu (80).

Z dodajanjem šuma (SNR do 10 dB) se razpoznavanje elips izboljša – pri slikah s SNR 10 dB je ta odstotek kar 95 % pri kriteriju 50, medtem ko je ta isti odstotek pri kriteriju 100 le 6 %, kar nam jasno kaže, da ENTROPIJA določa prenizke prage (glej sliko 3.11). Pri zelo velikem dodanem šumu odstotek razpoznanih elips pade in je v slikah, kjer je SNR -1 dB, enak 3,7 %. Prenizko določeni pragi pustijo po segmentaciji v šumnih elipsah



Slika 3.11: Krivulji ROC za segmentacijski algoritem ENTROPIJA.

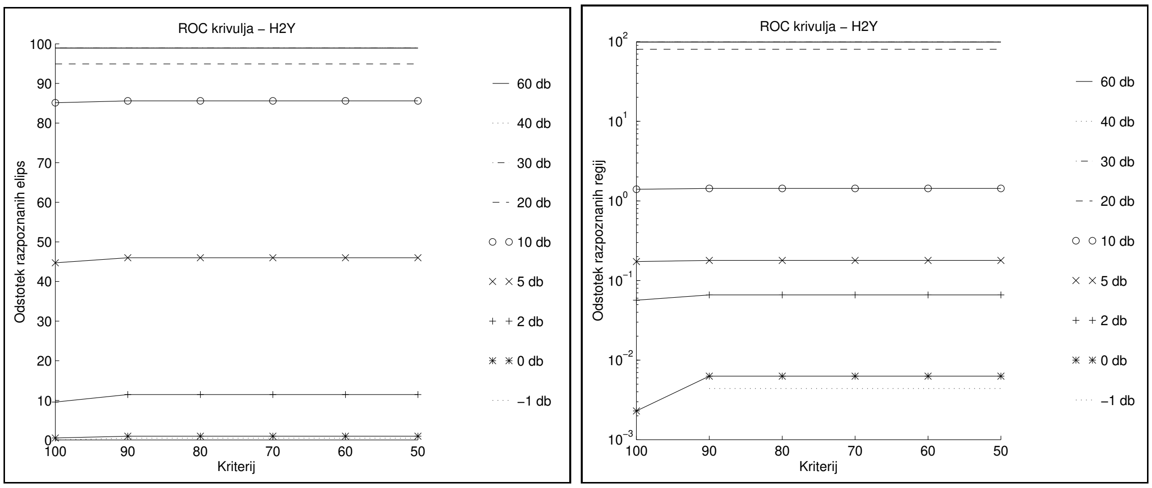


Slika 3.12: Kriterijska funkcija Ψ za brezšumno testno sliko s slike 3.2.

mnogo lukenj ali elipso celo razdelijo na posamezne piksle, zato je odstotek razpoznanih regij za segmentacijski algoritem ENTROPIJA izredno nizek (glej sliko 3.11). Pri slikah, kjer je SNR 40 dB, je ta odstotek 20 %, pri slikah s SNR -1 dB pa le še 0,0041 %.

Algoritem H2Y

Naslednji segmentacijski algoritem H2Y avtorja Marina (podpoglavje 3.1.2) je iz sklopa metod, ki določajo variabilne prage, to pomeni, da za vsak piksel določimo svoj prag. Algoritem H2Y je parametričen, in sicer mu moramo podati prag in parameter *Gain* (glej enačbo (3.9)). Za prag smo izbrali globalen prag, določen po metodi OPTIMAL, medtem ko smo parameter *Gain* določili izkustveno kot $\frac{T_{Opt}^2}{Q}$, kjer je T_{Opt} optimalen prag, Q pa je kvantizacijski nivo slike. Parameter *Gain* bistveno vpliva na segmentacijske rezultate. Če je bil ta parameter določen prenizko (med 0,1 in 0,3 optimalnega praga), smo sicer res odstranili ves šum, vendar pa tudi velik del elips (kar je bilo pred segmentacijo elipsa, je po segmentaciji bila le še množica izoliranih in nepovezanih pikslov), če pa je bil previsok (več kot 0,6 optimalnega praga), smo ves šum segmentirali kot objekte (že v slikah s SNR 20 dB je bil odstotek razpoznanih elips zelo nizek). Na sliki 3.13 so podani rezultati, dobljeni z algoritmom H2Y. Odstotek razpoznanih elips je pri malem šumu skoraj 100 % (v slikah s SNR 10 dB je 86 %). Algoritem H2Y se pri šibkem šumu obnaša kot nekakšen filter, saj uspe odstraniti ves impulzni šum (izolirane šumne piksle), medtem ko v slikah z večjim šumom (SNR večji od 10 dB) šum ne nastopa več izolirano in ga zato H2Y ne odstrani v celoti (posamezni deli so dodani k objektom). To trditev lahko podkrepimo z ROC-krivuljo za odstotek razpoznanih regij (slika 3.13), kjer vidimo, da je ta odstotek v slikah z majhnim dodanim šumom okrog 100 %, v slikah s SNR 10 dB pa je še zmeraj 1,4 %.



Slika 3.13: Krivulji ROC za segmentacijski algoritem H2Y.

Pri močnem šumu odstotek razpoznanih elips po pričakovanju pade, v slikah s SNR

-1 dB je le še 0,44 %, dodan šum pa pri segmentaciji inducira mnoge napačne regije, zato je tudi odstotek razpoznanih regij nizek (v slikah s SNR -1 dB je 0,0044 %).

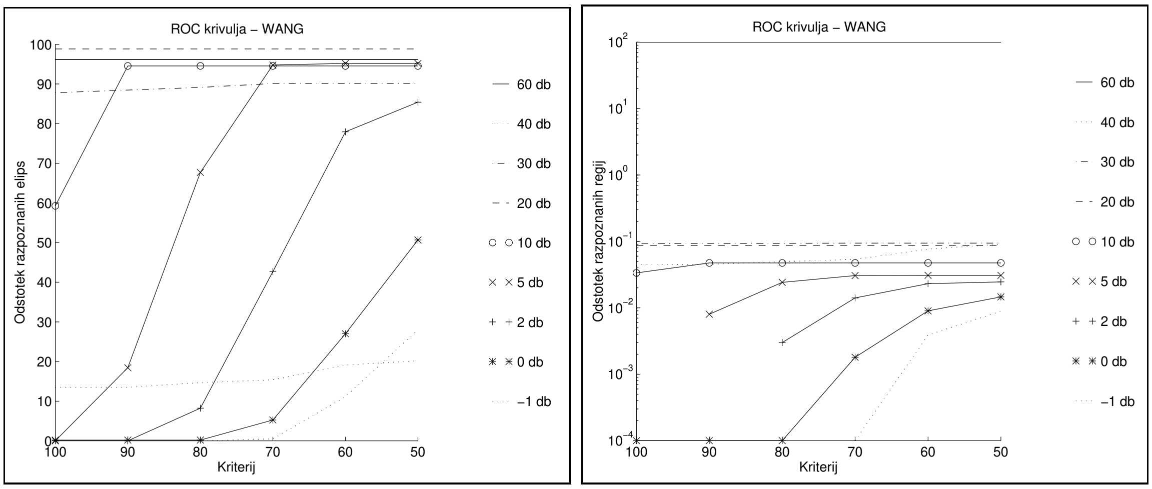
Algoritem WANG

Ker so naše testne slike sestavljene iz elips različnih sivin, smo preizkusili tudi segmentacijski algoritem, ki segmentira slike na osnovi več pragov. Izbrali smo segmentacijski algoritem avtorja Wanga (podpoglavje 3.1.3), v nadaljevanju ga bomo imenovali WANG. Ta algoritem potrebuje natančno opredeljeno sliko robov. Sliko robov smo dobili s pomočjo t. i. operatorja Laplace iz Gaussa, ki ga ponavadi označimo kot LoG [10, 13, 60, 63]. Ta operator ima obliko mehikanskega klubuka in pri določitvi robov upošteva tudi globalne lastnosti slike. V splošnem je izredno velikih dimenzij, mi smo postavili σ na 0,75 in s tem dobili filter dimenzij 7x7 pikslov.

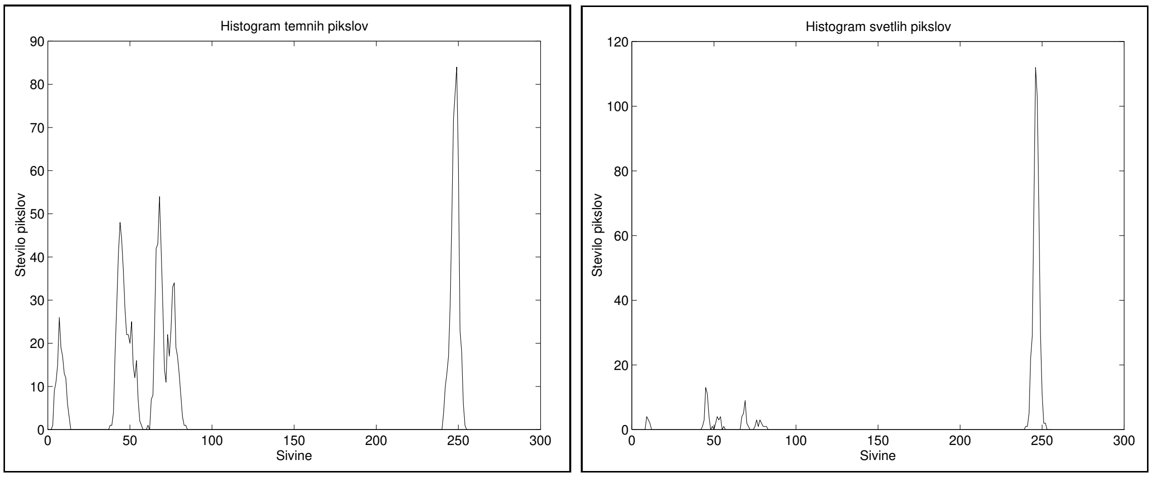
V algoritmu WANG (algoritem 1) lahko nastavljamo dva parametra: konstanto N in zaključni pogoj (točka 2 algoritma 1). Konstanto N smo nastavili na 10 (z njenim spreminjanjem nismo dosegali bistveno drugačnih rezultatov). Z zaključnim pogojem pa posredno vplivamo na število pragov, zato smo ustavili algoritem, ko je bilo v enem od obeh histogramov število temnih oz. svetlih robnih pikslov manjše od desetine vseh robnih pikslov (z znižanjem tega kriterija bi dobili več pragov, vendar pa to za naše slike ne bi bilo smiselno).

Na sliki 3.14 smo podali dobljene rezultate. Krivulja ROC za odstotek razpoznanih elips je podobna krivuljam že opisanih algoritmov (z dodajanjem šuma pada tudi odstotek razpoznavanja). Veliko odstopanje zasledimo pri testnih slikah, kjer je SNR 40 dB, saj je tukaj odstotek razpoznanih elips le 20 % oz. odstotek razpoznanih regij le 0,0908 %. Ti rezultati so posledica tega, da je algoritem WANG v teh primerih določal previsoke prage (velikokrat le enega, ki je bil v povprečju okrog sivine 240). Na sliki 3.15 sta prikazana histograma sivin za temne in svetle robne piksle za tipično testno sliko s SNR 40 dB (slika 3.2). Maksimalna vrednost v histogramu temnih pikslov je pri sivini 248, v histogramu svetlih pikslov pa pri sivini 245. Algoritem WANG določi prag pri sivini 248. To pa nastane zaradi tega, ker je vsota temnih pikslov do pragu 248 veliko večja kot vsota svetlih robnih pikslov do istega pragu (zato je verjetnost, da dobimo temne regije, velika), medtem ko sta vsoti temnih pikslov od praga 245 naprej in svetlih od istega praga približno enaki ter je zato verjetnost, da dobimo svetle regije, manjša od verjetnosti, da dobimo temne regije. Z dodajanjem šuma se maksimumi v obeh histogramih premaknejo iz zelo svetlega območja in ne prihaja več do teh problemov (zato odstotek razpoznih elips ponovno naraste).

V testnih slikah z veliko šuma (SNR od 5 dB dalje, kjer slika robov vsebuje tudi veliko robnih pikslov, nastalih zaradi šuma) algoritem WANG določi prvi prag v povprečju v okolici sivine 140 (malo nad polovico), vse ostale (nekateri izmed njih so zelo skupaj, tudi do 1 sivine) pa določi do sivine 190. Ker je prvi prag določen malo nad polovico, ostanejo vse elipse tudi po segmentaciji dobro vidne, vendar pa s tem ostane v segmentirani sliki tudi veliko šuma. To je razlog, da tudi pri večjem šumu ostane odstotek razpoznanih elips visok (v slikah s SNR 0 dB je 50 %, v slikah s SNR -1 dB pa 28 %), vendar pa je zaradi



Slika 3.14: Krivulji ROC za segmentacijski algoritem avtorja Wanga.



Slika 3.15: Histograma temnih in svetlih robnih pikslov za testno sliko (slika 3.2), kjer je SNR pri dodanem šumu 40 dB.

množice neizločenega šuma v segmentirani sliki odstotek razpoznanih regij izredno nizek (v slikah s SNR 0 dB je 0,0089 %, kar pomeni, da v povprečju dobimo po segmentaciji okrog 15000 regij).

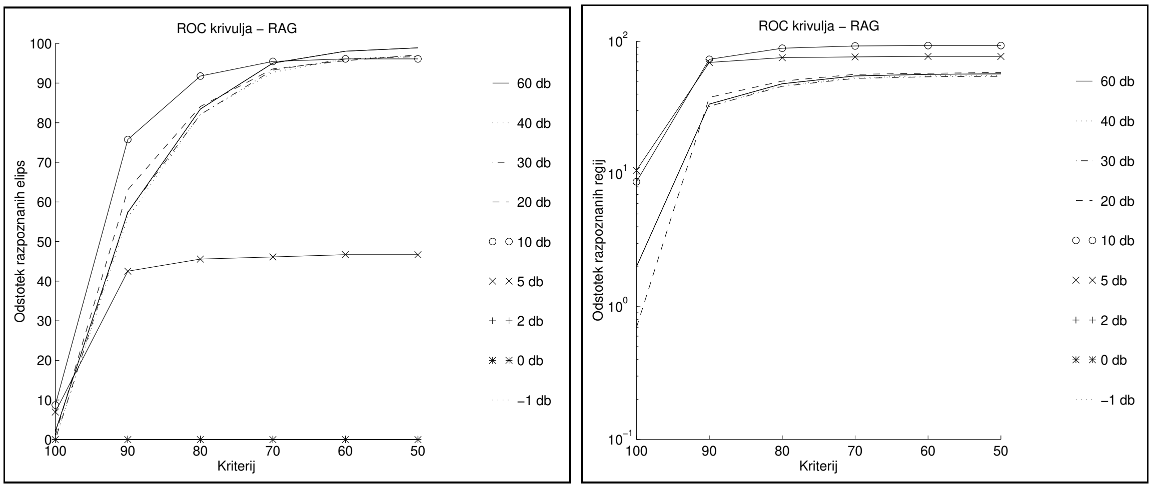
Na koncu bomo opisali še rezultate, dobljene z dvema segmentacijskima algoritmoma, ki delujeta na osnovi rasti regij. Prvi algoritem uporablja pri rasti regije podatkovno strukturo RAG (podpoglavje 3.3.1) – zato ga imenujemo kar algoritem RAG, drugi pa zliva regije s strategijo združi-išči (podpoglavje 3.3.1) – imenujemo ga algoritem UNION-FIND.

Algoritem RAG

Algoritem RAG najprej celotno sliko razdeli na majhne homogene regije, s pomočjo katerih tvori graf sosednosti (v grafu RAG so vse sosednje regije povezane). Celotno sliko smo razdelili na regije velikosti 4x4 piksle in s tem v osnovnem grafu RAG dobili 4096 regij. Zavedamo se, da regije velikosti 4x4 piksle kriterija homogenosti zmeraj ne zadovoljujejo popolnoma, vendar smo morali narediti ta kompromis zaradi izjemne časovne zahtevnosti algoritma. Še druga slabost izbire prevelikih začetnih regij so nazobčani in na neki način umetni robovi (segmentiranih) objektov.

Ko imamo tvorjen začetni graf sosednosti, združujemo sosednje regije, ki so si najbližje glede na izbrani kriterij. Izbrali smo kriterij, po katerem združimo dve regiji, katerih produkt razlike sivin med obema regijama in standardnega odklona potencialne združene regije je minimalen (če bi gledali le razliko sivin, ne bi zajeli celotne informacije). Pri zlivanju regij smo uporabili pristop Popovića [57], kjer smo v eni iteraciji zlili vse možne pare regij (in ne le regiji z minimalno vrednostjo na povezavi) – s tem smo bistveno zmanjšali časovno kompleksnost algoritma. Algoritem RAG konča zlivanje, ko ni več dveh regij, ki bi ju lahko zlili. Zlivanje končamo, ko je razlika sivin med dvema regijama (z minimalno vrednostjo povezave) večja od predpisanega praga (30). Na koncu smo dodali še korak poobdelave, kjer smo na osnovi kriterija homogenosti zlili majhne regije z najbližjimi večjimi (ta korak se ponavadi doda k algoritmom rasti regij).

Na sliki 3.16 so prikazani dobljeni rezultati. V slikah, kjer je SNR do 10 dB, je odstotek razpoznanih elips pri kriteriju 50 okrog 97 %, pri višjih kriterijih pa je odstotek nekoliko nižji, kar je posledica tega, da združujemo regije velikosti 4x4 piksle (nobene elipse ne moremo popolnoma pokriti s temi osnovnimi regijami, zato nizek odstotek pri visokih kriterijih). V slikah, kjer je SNR 5 dB, odstotek razpoznavanja pade na 47 %, medtem ko pri višjem šumu ne razpoznamo več nobene elipse – ker je dodanega toliko šuma, imajo vse regije približno enako povprečno sivino in standardni odklon, zato algoritem RAG združi vse regije, po segmentaciji tako dobimo le 1 regijo. Iz ROC-krivulje za odstotek razpoznanih regij lahko opazimo tudi visok odstotek razpoznavanja v slikah, kjer je SNR vse do 5 dB (77 %). Ta visok odstotek pa je tudi posledica dodanega koraka poobdelave (združevanje majhnih regij).



Slika 3.16: Krivulji ROC za segmentacijski algoritem RAG.

Algoritem UNION-FIND

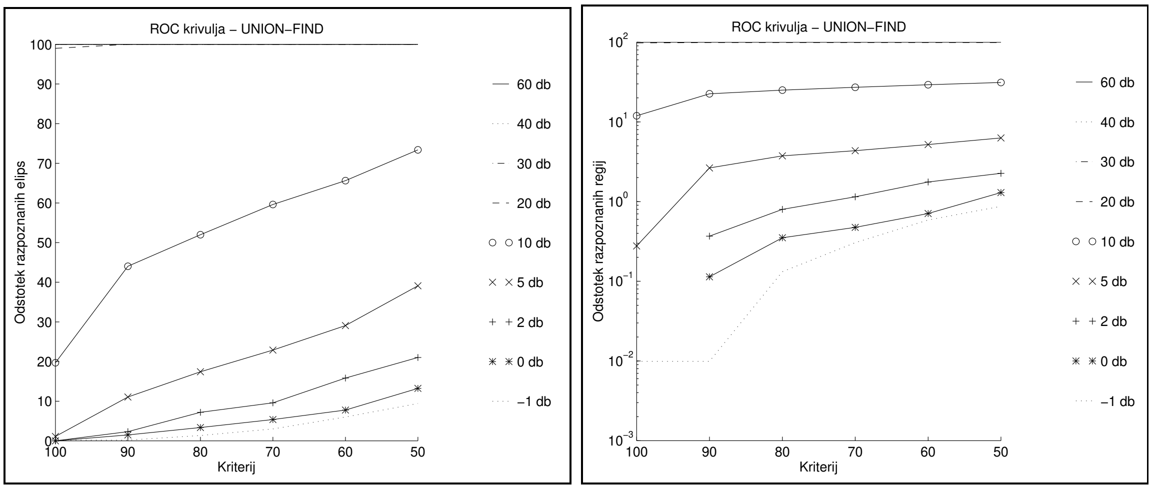
Zadnji algoritem, ki ga bomo opisali, je segmentacijski algoritem, ki zliva regije na osnovi strategije združi-išči – algoritem UNION-FIND. Ta algoritem je podrobno opisan v podpoglavju 3.3.1 (glej algoritem 8). Uporabili smo enak homogenostni kriterij kot pri algoritmu RAG in tudi dodali korak poobdelave (združevanje majhnih regij). Dobljeni rezultati so prikazani na sliki 3.17. V slikah, kjer je SNR manjše ali enako 20 dB, sta tako odstotek razpoznanih elips kakor tudi razpoznanih regij enaka 100 %, zatem pa odstotek razpoznavanja pada (glej ROC-krivulji). V slikah s SNR -1 dB je tako odstotek razpoznanih elips le še 9,4 %, odstotek razpoznanih regij pa 0,872 %.

3.5.4 Razprava

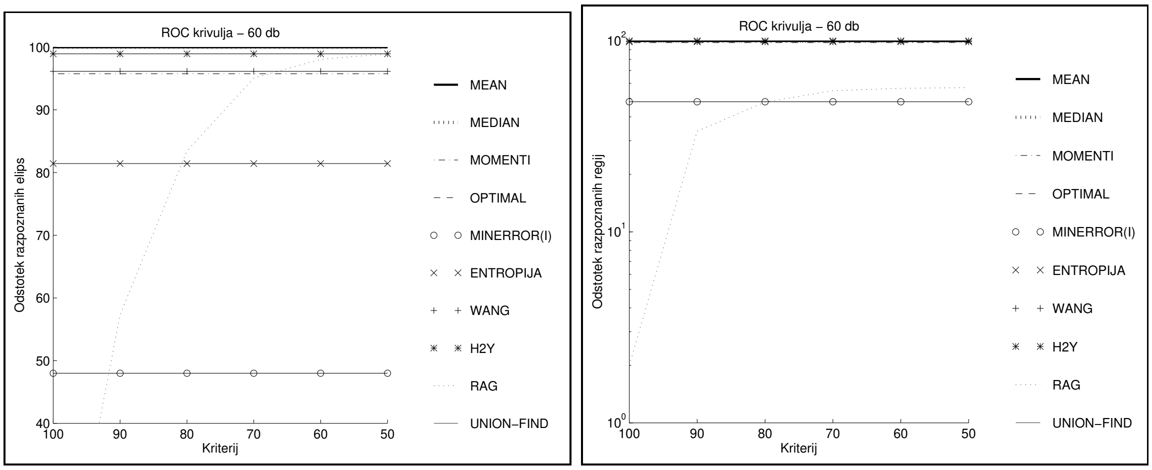
V prejšnjem podpoglavju smo natančno opisali dobljene rezultate z izbranimi segmentacijskimi algoritmi. Pojasnili smo določene slabosti in mejne primere delovanja posameznih algoritmov. V tem podpoglavju jih bomo primerjali med seboj, na koncu pa jih bomo na osnovi izbranih kriterijev razvrstili po uspešnosti.

Na slikah 3.18, 3.19, 3.20 in 3.21 je prikazana primerjava segmentacijskih algoritmov pri določenem šumu. Vse te ROC-krivulje smo že prikazali na prejšnjih slikah (slike 3.4 in 3.6 do 3.17), tukaj pa skušamo narediti nekakšen vizualen prikaz uspešnosti algoritmov v primerjavi z ostalimi.

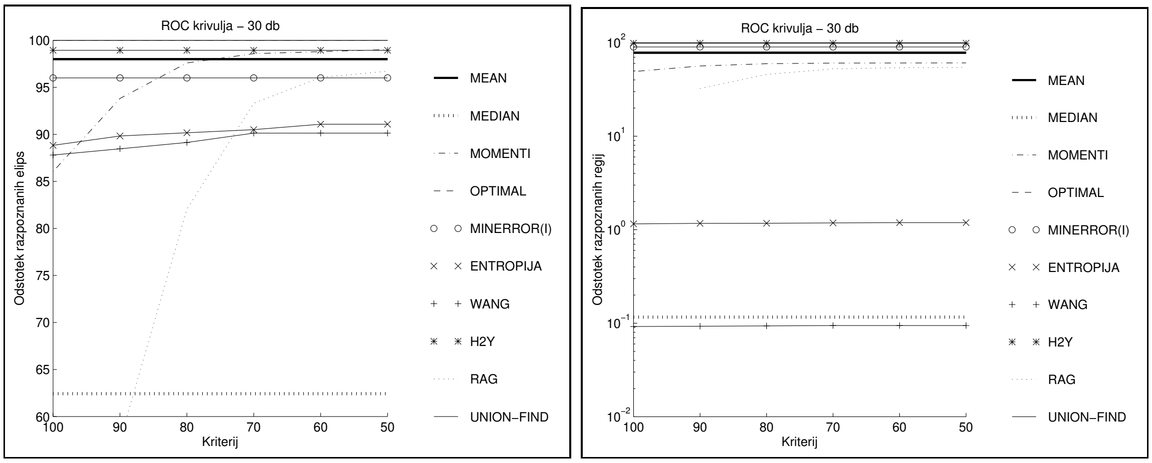
Na sliki 3.19 lahko tako npr. vidimo, da ima algoritem RAG pri kriteriju 50 višji odstotek razpoznanih elips kakor WANG. Ali lahko zato z gotovostjo rečemo, da je RAG boljši kakor WANG, še posebej, če dodamo, da se odstotek razpoznavanja pri algoritmu RAG dvigne šele pri kriteriju 70? Zaradi tega smo sklenili, da ne bomo dajali končne ocene o uspešnosti algoritma le na osnovi odstotka razpoznanih elips pri kriteriju 50.



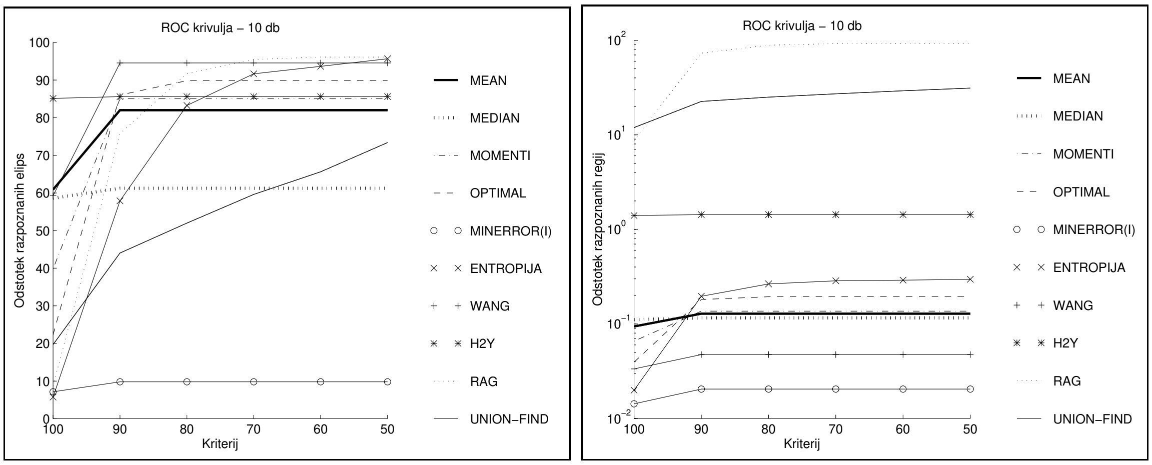
Slika 3.17: Krivulji ROC za segmentacijski algoritem UNION-FIND.



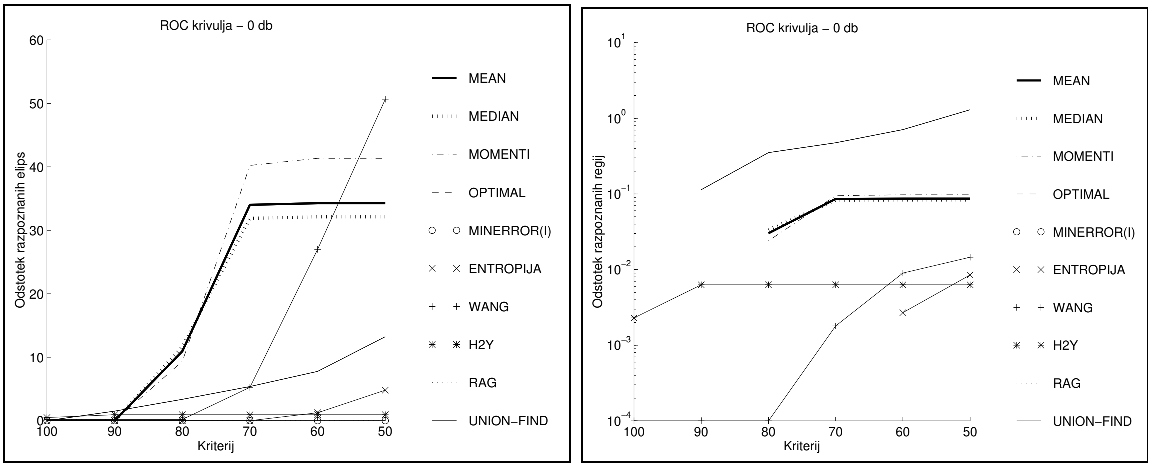
Slika 3.18: Krivulje ROC za vse opisane segmentacijske algoritme na slikah brez šuma.



Slika 3.19: Krivulje ROC za vse opisane segmentacijske algoritme na slikah, kjer je SNR 30 dB.



Slika 3.20: Krivulje ROC za vse opisane segmentacijske algoritme na slikah, kjer je SNR 10 dB.



Slika 3.21: Krivulje ROC za vse opisane segmentacijske algoritme na slikah, kjer je SNR 0 dB.

Odločili smo se narediti kompleksnejši kriterij, na osnovi katerega bomo razvrstili algoritme. Namesto da upoštevamo le odstotek razpoznanih elips pri določenem kriteriju, smo v našo oceno vključili odstotke razpoznanih elips pri vseh kriterijih (dejansko smo izračunali ploščino pod krivuljo ROC za odstotek razpoznanih elips) za določen šum. Ploščine pod krivuljami ROC smo računali čisto v matematičnem smislu. To pomeni, da smo izračunali ploščino lika, ki je levo in spodaj omejen z obema koordinatnima osema, zgoraj z ROC-krivuljo, na desni pa s premico, ki gre skozi kriterij 50 in je paralelna osi odstotek razpoznanih elips. Pri tem moramo opozoriti, da smo pri takšnem računanju ploščin naredili določeno napako (v smislu eksaktnosti). Naše krivulje ROC so namreč definirane le za točno določene diskretne vrednosti (pri kriterijih 100, 90, 80 itd.), ploščine pa smo računali tako, kot da bi bile krivulje zvezne.

Ker se uspešnost segmentacijskih algoritmov s šumom spreminja, smo izračunali ploščino pod krivuljo ROC za določen algoritem pri vsakem šumu (v resnici se nam je zdelo smiselno upoštevali le testne slike do SNR 2 dB, saj je potem šum že prevelik). Zatem pa smo izračunali povprečno vrednost vseh teh ploščin za določen algoritem. Te povprečne vrednosti smo vnesli v tabelo 3.1 v polje "Ploščina - elipse".

Na osnovi polja "Ploščina - elipse" lahko že razvrstimo segmentacijske algoritme. Po tem kriteriju se je najslabše odrezal MINERROR(I), sledita MEDIAN in RAG, ki sta zelo skupaj, zatem UNION, WANG ter H2Y, najbolje pa so se odrezali MEAN, MOMENTI in OPTIMAL (sta tudi zelo skupaj). S takšnim ocenjevanjem le na osnovi ploščine pod ROC-krivuljo za odstotek razpoznanih elips nismo zadovoljni, saj nismo nikjer upoštevali, kako segmentacijski algoritem odstranjuje šumne regije (če jih sploh). Ta podatek je izredno važen, saj ni pomembno le, ali algoritem uspe razpoznaniti elipse (v splošnem objekte), ampak tudi, koliko regij je moral pregledati in jih zavrniti, preden je razpoznal elipse. Zato smo na enak način izračunali tudi ploščine pod ROC-krivuljami za odstotek razpoznanih regij (glej polje "Ploščina - regije" v tabeli 3.1). Obe izračunani ploščini (pod krivuljama ROC za odstotek razpoznanih elips in regij) smo zatem pomnožili ter tako dobili končni kriterij, na osnovi katerega smo rangirali segmentacijske algoritme (polje "Produkt" v tabeli 3.1).

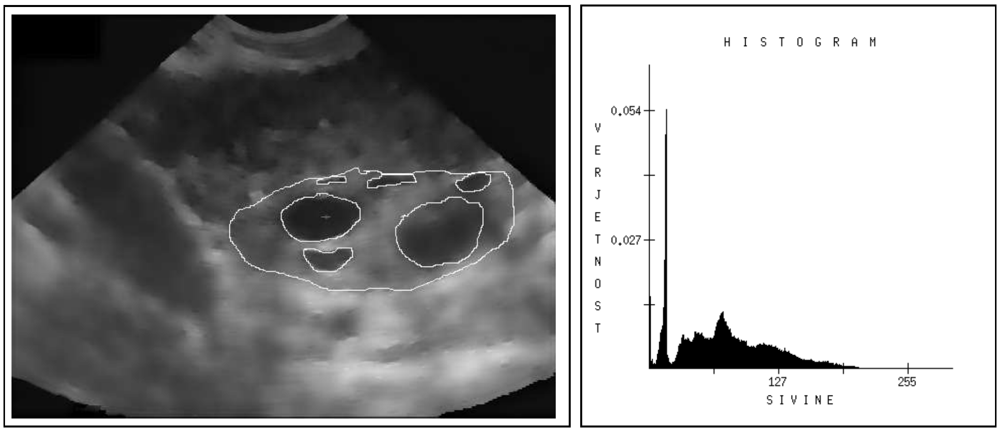
Algoritem	**Ploščina - elipse**	**Ploščina - regije**	**Produkt**
MEAN	3,9568	1,9774	7,8241
MEDIAN	3,0215	0,7188	2,1719
MOMENTI	4,0349	1,6624	6,7078
OPTIMAL	4,0453	2,7133	10,9762
MINERROR(I)	2,3824	2,0970	4,9959
ENTROPIJA	3,2260	0,7437	2,3991
WANG	3,5026	0,7166	2,5098
H2Y	3,8173	2,7060	10,3299
RAG	3,0598	2,3143	7,0814
UNION	3,4476	3,0624	10,5579

Tabela 3.1: Podatki o ploščinah pod ROC-krivuljama za posamezne segmentacijske algoritme za testne slike (kriterij 50). Polje "Ploščina - elipse" pomeni ploščino pod ROC-krivuljo za odstotek razpoznanih elips (maksimalna ploščina je lahko 5,0), polje "Ploščina - regije" pomeni ploščino pod ROC-krivuljo za odstotek razpoznanih regij, v polju "Produkt" je produkt obeh prejšnjih polj.

Na osnovi polja "Produkt", ki je tudi naš končni odločitveni kriterij, pa dobimo naslednjo razvrstitev algoritmov: najslabše so se odrezali algoritmi MEDIAN, ENTROPIJA in WANG (vrednosti produkta izredno skupaj), zatem sledijo MINERROR(I), MOMENTI in RAG (slednja imata zelo podobne vrednosti produkta). Najbolje pa so se odrezali algoritem MEAN ter algoritmi H2Y, UNION in OPTIMAL (vsi trije imajo zelo podobne vrednosti produkta). Iz vsega povedanega lahko rečemo, da je segmentacijski algoritem OPTIMAL najboljši, čeprav moramo takoj dodati, da je najboljši za dano vrsto testnih slik (za tako zastavljeno simulacijo) in med izbranimi in preizkušenimi segmentacijskimi algoritmi.

Takšne rezultate smo dobili pri kriteriju 50 – to pomeni, da je regija razpoznana kot pravilna, tudi če prekriva le 50 % elipse, kar je zagotovo zelo ohlapen kriterij. Zato smo ves ta postopek (izračun ploščin pod krivuljama ROC za odstotek razpoznanih elips in regij) ponovili pri bolj restriktivnem kriteriju, in sicer pri kriteriju 70 (70 % elipse mora biti prekrite z regijo, če naj bi bila ta regija razpoznana kot elipsa). Kljub temu, da smo kriterij dvignili, smo dobili zelo podobno razvrstitev algoritmov, saj sta mesti zamenjala le algoritma MOMENTI in RAG ter algoritma UNION in H2Y.

Uspešnost teh algoritmov smo preizkusili tudi na realnih slikah, in sicer smo z njimi iskali jajčne mešičke v ultrazvočnih posnetkih jajčnika. Eden izmed teh posnetkov je prikazan skupaj s svojim histogramom sivin na sliki 3.22. Ta naloga je podobna opravljeni simulaciji (iskanje temnih elips na svetlem ozadju), saj so jajčni mešički jajčaste temne tvorbe, ki se nahajajo na svetlejšem ozadju (ta trditev velja za lokalno okolico mešička). Ker želimo ocenjevati uspešnost algoritmov, smo podali natančne podatke o številu in položaju jajčnih mešičkov. S tem so bili izpolnjeni vsi pogoji, da smo lahko uporabili popolnoma enako shemo razpoznavanja kot pri testnih slikah (slika 3.3). Enaki so tudi postopki pri razpoznavanju regij (podpoglavje 3.5.2).

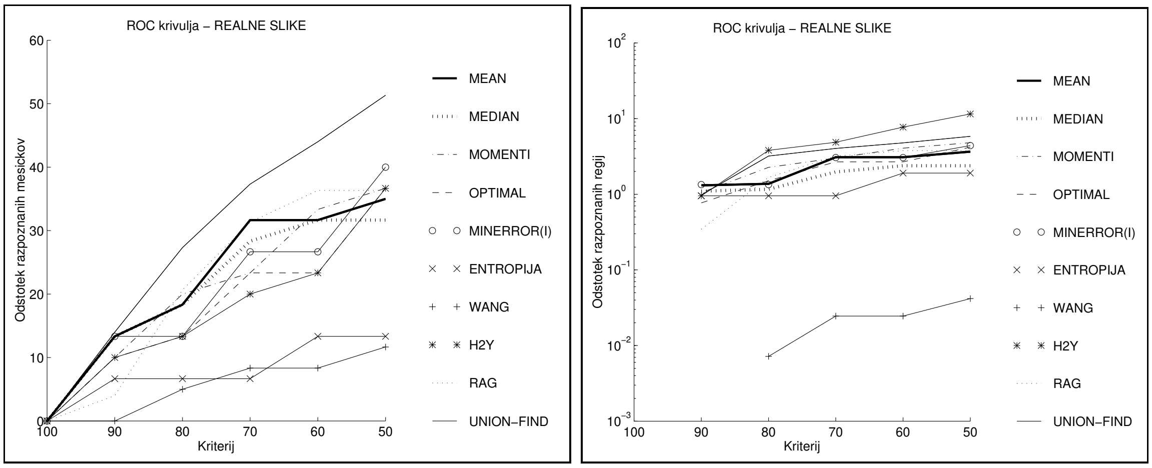


Slika 3.22: Ultrazvočni posnetek jajčnika. Jajčnik in 6 mešičkov je obkrožil zdravnik ekspert. Na desni sliki je pripadajoč histogram sivin.

Dobljene rezultate smo podali tudi v obliki krivulj ROC. Na sliki 3.23 sta prikazani ROC-krivulji za odstotek razpoznanih mešičkov in razpoznanih regij za vsak segmentacijski algoritem.

Najvišji odstotek razpoznavanja mešičkov dosežemo pri algoritmu UNION (51 %), najnižji odstotek razpoznavanja, 11 %, pa ima algoritem WANG (vse pri kriteriju 50). Ti dobljeni rezultati pa niso popolnoma merodajni, saj nismo obdelali velike količine ultrazvočnih posnetkov, ampak le 5 ultrazvočnih posnetkov jajčnika (testnih umetnih slik smo imeli 50), ki so bili sicer od različnih žensk in posneti pod različnimi pogoji.

Podobno kot za testne slike smo tudi za realne ultrazvočne slike rangirali segmentacijske algoritme po njihovi uspešnosti. Izračunali smo ploščine pod ROC-krivuljami za odstotek razpoznanih mešičkov in regij. Vse te ploščine so v tabeli 3.2. Na osnovi polja "Produkt" dobimo naslednjo razvrstitev algoritmov. Najslabše rezultate daje algoritem WANG, to pa predvsem zaradi tega, ker so jajčni mešički sestavljeni iz več različnih sivin (sivine so nižje, če se približujemo središču mešička), kar povzroči, da po večpragovni segmentaciji dobimo iz enega mešička več regij, ki jih ne moremo razpoznati kot mešičke. Sledi algoritem ENTROPIJA, zatem pa štirje algoritmi: OPTIMAL, RAG, MEDIAN in MINERROR(I), ki dajejo podobne rezultate. Algoritma MEAN in MOMENTI imata skoraj identične produkte, medtem ko algoritma H2Y in UNION izstopata. Algoritem UNION, ki smo ga opredelili kot najboljšega, ima tudi sicer najvišji odstotek razpoznanih mešičkov. Najboljše rezultate pa je dajal zaradi tega, ker večini drugih segmentacijskih



Slika 3.23: Krivulje ROC za vse opisane segmentacijske algoritme na realnih ultrazvočnih slikah.

Algoritem	**Ploščina - mešički**	**Ploščina - regije**	**Produkt**
MEAN	0,4750	0,0423	0,0201
MEDIAN	0,4583	0,0325	0,0149
MOMENTI	0,4167	0,0483	0,0201
OPTIMAL	0,3500	0,0362	0,0127
MINERROR(I)	0,4000	0,0421	0,0168
ENTROPIJA	0,1667	0,0238	0,0040
WANG	0,0917	0,0002	0,0000
H2Y	0,3333	0,0719	0,0240
RAG	0,4033	0,0361	0,0146
UNION	0,6000	0,0620	0,0372

Tabela 3.2: Podatki o ploščinah pod ROC-krivuljama za posamezne segmentacijske algoritme za realne ultrazvočne slike. Maksimalna ploščina pod krivuljo ROC je 5.

algoritmov (v svoji osnovni obliki) ni uspelo upoštevati, da so mešički temnejši glede na svojo okolico in ne na celotno sliko ter so zato mnoge mešičke priključili k ozadju.

3.5.5 Zaključek

Na osnovi obeh simulacij (razpoznavanja elips v umetnih slikah in mešičkov v ultrazvočnih posnetkih) smo ocenili uspešnost izbranih algoritmov. V umetnih slikah se je najbolje izkazala pragovna operacija OPTIMAL, pri ultrazvočnih posnetkih pa segmentacijska metoda rasti regije UNION. Ta algoritma (ali mogoče neko kombinacijo obeh) tudi priporočamo za segmentacijo podobnih slik.

Kljub tej razvrstitvi segmentacijskih algoritmov pa ne moremo (niti si ne upamo) trditi, da sta prav ta dva algoritma najboljša in da ju moramo zato vedno uporabljati. Izbira posamezne segmentacijske metode je kljub vsemu še vedno najbolj odvisna od tipa naloge, s katero se ukvarjamo. V splošnem pa tudi velja, da popolne segmentacije, kjer regije po segmentaciji že tudi direktno ustrezajo objektom, ne moremo opraviti brez uporabe dodatnega znanja o problemu, torej brez kontekstno odvisnega obdelovanja slik.

4.

Houghova transformacija in jajčni mešički

V tem poglavju se bomo seznanili s teoretičnimi osnovami Houghove transformacije, zatem pa jo bomo preizkusili pri reševanju problema razpoznavanja jajčnih mešičkov (poglavje 2). Houghova transformacija je segmentacijska metoda na osnovi robov, ki skuša v slikah direktno poiskati posamezne objekte. V podpoglavjih 4.1 in 4.2 se bomo zato podrobno seznanili s samo Houghovo transformacijo ter njenimi metodami, namenjenimi za iskanje elips v slikah. Zatem pa bo sledilo podpoglavje 4.3, kjer bodo podrobno prikazani in opisani dobljeni rezultati razpoznavanja jajčnih mešičkov s pomočjo Houghove transformacije.

4.1 Houghova transformacija

Houghovo transformacijo (HT) je prvi vpeljal Paul Hough leta 1962. HT je vpeljal kot metodo za razpoznavanje kompleksnih vzorcev v binarnih slikah. To je dosegel z določanjem specifičnih vrednosti parametrov, ki so okarakterizirali takšne vzorce. Prostorsko razvlečne vzorce je transformiral tako, da so tvorili prostorsko kompaktne značilnosti v prostoru možnih vrednosti parametrov. HT pretvori težek, globalni problem razpoznavanja v slikovnem prostoru v lažje rešljiv problem iskanja lokalnega maksimuma (vrha) v parametričnem prostoru [30, 42].

4.1.1 Definicija Houghove transformacije

Osnovno idejo Houghove metode bomo predstavili na primeru razpoznavanja množice kolinearnih točk v sliki. Množico slikovnih točk (x, y), ki ležijo na skupni premici, lahko definiramo z relacijo f takole:

$$f((\hat{m}, \hat{c}), (x, y)) = y - \hat{m}x - \hat{c} = 0, \tag{4.1}$$

kjer sta $\hat{m}$ in $\hat{c}$ dva parametra, smerni koeficient in odsek na ordinatni osi, ki določata premico. Enačba (4.1) preslika vsako kombinacijo parametrov $(\hat{m}, \hat{c})$ v množico slikovnih točk. Z oznako strešica smo označili veličine v domeni preslikave. Preslikava je ena proti mnogo iz prostora možnih vrednosti parametrov v prostor slikovnih točk. HT uporablja idejo, da enačba (4.1) dejansko predstavlja vzajemno omejitev med slikovnimi točkami in parametri, ter jo zato lahko interpretiramo tudi kot preslikavo ena proti mnogo iz slikovne točke v množico možnih vrednosti parametrov. V našem primeru to pomeni, da HT dejansko ustreza izračunanju parametrov vseh možnih premic, ki gredo skozi dano slikovno točko $(\hat{x}, \hat{y})$. To operacijo imenujemo "vzvratna projekcija" (angl. backprojection) slikovne točke in jo opišemo z enačbo (4.2) [30]:

$$g((\hat{x}, \hat{y}), (m, c)) = \hat{y} - \hat{x}m - c = 0. \tag{4.2}$$

V primeru premice se vsaka slikovna točka $(\hat{x}, \hat{y})$ projecira nazaj oz. definira premico v (m, c)-prostoru. Točke, ki so kolinearne v slikovnem prostoru, se vse stekajo v skupno točko v parametričnem prostoru, koordinati te skupne parametrične točke pa določata premico, ki povezuje kolinearne slikovne točke. HT identificira te presečiščne točke v parametričnem prostoru. Določanje presečiščnih točk v parametričnem prostoru je lokalna operacija (iskanje lokalnega maksimuma) in je izrazito lažja operacija kot odkrivanje velikih vzorcev v slikovnem prostoru [30].

Opisali smo idejo Houghove transformacije na primeru premice, ki smo jo parametrizirali s parametri (m, c). Ta parametrizacija pa ni najboljša, saj ima singularne točke za premice z velikim naklonom $(m \to \infty)$. To bi dejansko pomenilo, da bi moral biti parametričen prostor neskončno velik (zaradi parametra m), čeprav bi imeli opravka s slikovnim prostorom vnaprej predpisanih dimenzij (digitalno sliko znanih dimenzij). Avtorja Duda in Hart [14] sta za rešitev tega problema predlagala t. i. normalno parametrizacijo. Po tej parametrizaciji lahko vsako premico opišemo s kotom θ njene normale (orientacijo) in razdaljo ρ od izhodišča. Enačba premice, ki ustreza tej geometriji, je

$$\rho = x \cos(\theta) + y \sin(\theta). \tag{4.3}$$

S parametrizacijo premice po enačbi (4.3) postane parametrični prostor (θ, ρ) omejen. Če omejimo kot θ na interval $[0, \pi]$, potem vsaki premici v slikovnem prostoru (x, y) ustreza natanko ena točka v parametričnem prostoru (θ, ρ). Iz enačbe (4.3) tudi vidimo, da se vsaka slikovna točka preslika v sinusno krivuljo v dvodimenzionalnem parametričnem prostoru (θ, ρ). Hitro lahko pokažemo, da imajo kolinearne slikovne točke tudi s takšno parametrizacijo skupno presečišče v parametričnem prostoru [30, 60, 63]. Lokalni maksimum $(\theta_{max}, \rho_{max})$ v parametričnem prostoru pa kaže na verjetno prisotnost premice $y = -x\, ctg\,(\theta_{max}) + \rho_{max} / \sin(\theta_{max})$ v slikovnem prostoru.

Razširitev Houghove transformacije za razpoznavanje tudi drugih, parametrično definiranih slikovnih krivulj, različnih od premic, je popolnoma enostavna. Ideja je popolnoma enaka kot za premico. Slikovne točke na krivulji, ki jo določa n parametrov, $a_1, \ldots, a_n$, definiramo z enačbo (4.4)

$$f((\hat{a}_1, \ldots, \hat{a}_n), (x, y)) = 0, \tag{4.4}$$

z zamenjavo vloge parametrov in spremenljivk v enačbi (4.4), pa definiramo relacijo za vzvratno projekcijo slikovnih točk v parametričen prostor:

$$g((\hat{x}, \hat{y}), (a_1, \ldots, a_n)) = 0. \tag{4.5}$$

S preslikavo po enačbi (4.5) tvorimo hiperpovršino v n-dimenzionalnem parametričnem prostoru za slikovno točko (x, y). Najbolj verjetni parametri za slikovno krivuljo so na presečišču več takšnih hiperpovršin.

Metodo lahko posplošimo tudi na odkrivanje poljubnih oblik, ki se ne dajo opisati z analitičnimi enačbami. Takšno posplošeno HT imenujemo posplošena HT [30, 63].

4.1.2 Lastnosti Houghove transformacije

V nadaljevanju bomo spoznali moč Houghove transformacije, zatem pa še njene omejitve in hibe.

Prednosti Houghove transformacije

Houghova transformacija ima nekaj zelo dobrih lastnosti [30, 42, 60, 63]. Prvič, vsako slikovno točko lahko obravnavamo neodvisno in zato lahko HT implementiramo z več kot eno procesorsko enoto, kar pomeni, da je možno paralelno obdelovanje točk. S paralelnim obdelovanjem pa dejansko naredimo ta algoritem primeren tudi za realnočasovne aplikacije (s tem npr. postane algoritem primeren tudi za razpoznavanje oblik v bioloških sistemih).

Drugič, HT lahko razpoznava delno oz. rahlo deformirane oblike. Prekrivanje je velik problem za večino ostalih metod za razpoznavanje oblik, HT pa ta problem izvrstno rešuje, saj je velikost lokalnega maksimuma (vrha) v parametričnem prostoru direktno proporcionalna številu ujemajočih se točk z meje. Velikost in položaj lokalnega maksimuma v prostoru zagotavljata mero podobnosti med obliko in modelom.

Tretjič, metoda HT je zelo robustna za naključno dodane podatke (slikovne točke), ki so nastali s slabo opravljeno predobdelavo slike. K naključno dodanim podatkom lahko štejemo tudi aditiven šum (bel šum, Gaussov šum, šum sol in poper, šum zaradi slabih pogojev zajemanja slik, pegasti šum itd.). Naključne slikovne točke koherentno prispevajo le na enem mestu v akumulatorskem polju (parametrični prostor pogosto imenujemo akumulator, saj v tem prostoru dejansko le seštevamo prispevke slikovnih točk) in zato tvorijo le zelo nizko ozadje prispevkov v polju. Bolj resen problem kot naključni podatki so podatki z mej (robov) oblik, ki jih ne iščemo. To lahko povzroči strukturirano ozadje dodanih prispevkov in potrebnega je precej dela za odpravljanje oz. identificiranje takšnih situacij.

Četrtič, HT lahko tudi sočasno akumulira prispevke za več primerov določenega razreda oblik v isti sliki. Vsak primerek takšne oblike preprosto tvori različen vrh (maksimum) oz. grozd (cluster) v akumulatorskem polju.

Slabosti Houghove transformacije

Glavna slabost Houghove trasformacije so ogromne prostorske in računske zahteve pri njeni standardni implementaciji [30]. Poglejmo si standardno implementacijo HT pobliže.

Uporaba HT je v prvi vrsti namenjena za iskanje kompleksnih vzorcev v binarnih slikah. Ker pa imamo opravka z digitalno obdelavo slik, mora biti tudi slika digitalizirana oz. razdeljena na majhne slikovne elemente – piksle (dimenzija slik oz. število pikslov je naravno število). Čeprav so domene vrednosti iz slikovnega prostora naravna števila, lahko zaloga vrednosti v parametričnem prostoru zajame celotno realno ravnino $\mathcal{R}^2$ (glej enačbo (4.5)). Implementacija realnega parametričnega prostora z računalnikom ni možna, zato moramo tudi parametričen prostor diskretizirati. Diskretizacijo naredimo tako, da diskretiziramo vsak parameter posebej. S tem dosežemo, da tudi parametričen prostor preide v večdimenzionalen prostor naravnih števil. Parametričen prostor lahko tako dejansko imenujemo (akumulatorsko) polje, ki je dimenzije α^n, pri čemer n pomeni število vseh parametrov, potrebnih za opis določene oblike, vsak parameter pa je diskretiziran na α intervalov. Prišli smo do prve slabosti HT: akumulatorsko polje je lahko izredno veliko, če je število vseh parametrov n veliko (razpoznavanje krožnice npr. zahteva 3 parametre, elipse 5 parametrov itd.) in so diskretizirani na veliko število intervalov α (orientacijo θ lahko npr. diskretiziramo z natančnostjo 1^o, kar pomeni 360 elementov v θ smeri, glej enačbo (4.3)). Veliko akumulatorsko polje hkrati zahteva tudi zelo veliko pomnilnega prostora.

Standardna HT poteka takole. Vsako postavljeno slikovno točko v binarni sliki transformiramo v parametričen prostor oz. akumulator. Transformacijo izvedemo po enačbi (4.5), kjer fiksiramo slikovno točko $(\hat{x}, \hat{y})$, medtem ko postanejo parametri, $a_1, \ldots, a_n$, spremenljivke. Celoten problem transformacije je torej poiskati vse možne kombinacije parametrov, ki zadostijo enačbi (4.5) oz. v parametričnem prostoru poiskati n-dimenzionalno hiperpovršino. Za vsako kombinacijo ustrezajočih parametrov $(v_1, \ldots, v_n)$ povečamo v celici oz. točki $(v_1, \ldots, v_n)$ akumulatorskega polja vrednost za 1 (zaradi tega tudi ime akumulatorsko polje za parametrični prostor). Glavni strošek izračunavanja tega algoritma je iskanje presečišča med celicami akumulatorja in n-dimenzionalno hiperpovršino. Druga slabost HT je ogromno število izračunavanj, ki rastejo eksponentno z velikostjo problema.

Ogledali smo si slabosti standardne implementacije HT. Učinkovitost HT lahko povečamo z raznimi izboljšavami te metode, kot npr. z uporabo akumulatorjev majhnih dimenzij ali z uporabo dodatnih podatkov, s katerimi omejimo obseg parametrov [30, 42, 63].

4.1.3 Uporaba Houghove transformacije

Uvodoma smo že povedali, da je Houghova transformacija namenjena za iskanje kompleksnih vzorcev v slikah. Dolgo časa so HT uporabljali le v industriji, npr. pri opravljanju raznih meritev oz. kot pomoč pri nadzorovanju raznih industrijskih procesov. V tem zelo šumnem okolju se je izkazala kot zelo kvalitetna in je dajala zelo točne rezultate. V zadnjem času pa se veliko uporablja na širokem področju problemov iz računalniškega vida. V industriji se je razvoj uporabe HT širil predvsem v smer avtomatskega krmiljenja in vizualnega nadzorovanja industrijskih nalog (rezultat je predvsem večja produktivnost in kvaliteta produktov).

Metoda HT se danes veliko uporablja na področju medicine, predvsem kot pomoč pri postavljanju diagnoz na osnovi medicinskih slik. Tako je bila npr. HT uporabljena za detektiranje mej levega in desnega srčnega prekata v CT slikah [63], pri razpoznavanju robnih linij v SEM slikah (angl. Scanning Electron Microscope) [30], pri modeliranju 3D arterije iz ultrazvočnih posnetkov [42] itd.

Razumevanje slik je naslednja pomembna veja uporabe HT. V to področje lahko štejemo avtomatsko nadzorovanje teritorija s pomočjo podatkov, dobljenih iz satelitov ali radarskih slik, posnetih iz letal. Aplikacije, ki jih najdemo, so: s pomočjo HT so razpoznali linearne in hiperbolične značilnosti v seizmogramih, HT je bila uporabljena tudi pri razpoznavanju nejasnih linij v slikah SAR, posplošeno HT so uporabili tudi za iskanje ujemanja podgrafov, dobljenih iz geografskih kart [30], za razpoznavanje okroglih struktur v satelitskih slikah [42] itd. Avtonomna vozila in določanje parametrov gibanja iz zaporedja posnetkov sta še dve izmed mnogih aplikacij HT na področju računalniškega vida. Z metodo HT so npr. razpoznali ceste in steze, to informacijo pa so zatem uporabili pri vodenju avtorobota [30]. HT so uporabili tudi za navigacijski sistem [42] itd.

Avtomatiziranje pisarniškega dela, kot je npr. pregledovanje oz. vnašanje velikih količin informacij s papirja ipd., prinaša veliko korist vsakemu podjetju. Tako je nastalo kar nekaj aplikacij, ki so z uporabo HT obdelovali dokumente in razpoznavali črke [42]. Metodo HT so uporabili tudi pri razpoznavanje hebrejskih znakov [30].

Metodo HT je možno uporabiti še na mnogih drugih področjih. Naj za prikaz njene vsesplošne uporabnosti naštejemo še nekaj teh področij. S HT so eksperimentirali pri štetju črt v avtomatskem sistemu za analizo prstnih odtisov. Generalizirano HT so uspešno uporabili tudi pri razpoznavanju 3D oblik iz zaporednih podatkov (angl. Range data). Metodo HT so predlagali tudi za kompresijo krivuljnih oz. valčnih podatkov [30].

4.2 Razpoznavanje elips s pomočjo Houghove transformacije

Mnogo predmetov oz. kompleksnih oblik lahko ponazorimo z elipsami. Elipsa je preprosta, analitično opisna krivulja 2. reda, ki jo lahko relativno preprosto odkrijemo v slikah. V tem podpoglavju si bomo ogledali, kako s pomočjo HT poiščemo elipse v slikah.

Splošna enačba nerotirane elipse s središčem v točki (x_0, y_0) ter osema a in b je podana z enačbo (4.6)

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1. \tag{4.6}$$

Iz enačbe (4.6) ter ob upoštevanju rotacije elipse za kot ϕ ugotovimo, da lahko poljubno elipso opišemo s petimi parametri, in sicer (x_0, y_0, a, b, ϕ). Sedaj si pa oglejmo nekaj metod HT, s katerimi lahko razpoznavamo elipse v slikah.

Standardna Houghova transformacija

Standardno implementacijo HT smo že delno spoznali v prejšnjem podpoglavju, v tem podpoglavju pa jo bomo bolj detaljno opisali na primeru elips. Ugotovili smo že, da za opis poljubne elipse potrebujemo 5 parametrov, to hkrati pomeni, da je parametričen prostor oz. akumulator petdimenzionalen. Če predpostavimo, da smo vse parametre razdelili na α intervalov (v praksi ponavadi to ni res), potem je akumulator velik α^5 oz. prostorska zahtevnost je reda $\mathcal{S}(n^5)$, kar npr. pri izbiri za α=360 znese približno 6*10^{12} elementov akumulatorskega polja (to je približno 6000 giga zlogov pomnilnega prostora ob predpostavki, da je vsak element velik le en zlog). Torej je prvi problem, ki ga moramo rešiti pri standardni HT, da izberemo primerno število intervalov za vsak parameter. Treba je skleniti kompromis med natančnostjo končnega rezultata in velikostjo akumulatorja: zahtevana večja natančnost rezultatov zahteva tudi večje akumulatorsko polje in obratno.

Ko smo si izbrali primerno akumulatorsko polje za naš problem, se lahko lotimo HT. Sedaj moramo vsako postavljeno slikovno točko $(\hat{x}, \hat{y})$ preslikati v vse možne kombinacije parametrov v parametričnem prostoru oz. za vsako možno kombinacijo parametrov, povečati ustrezno celico akumulatorja za ena. Vse možne kombinacije parametrov pa lahko določimo po naslednjem postopku. Ko smo izbrali oz. fiksirali slikovno točko $(\hat{x}, \hat{y})$, potem enačba (4.6), v kateri moramo upoštevati še rotacijo, preide v enačbo oblike:

$$z(a, b, x_0, y_0, \phi) = 0, \tag{4.7}$$

oziroma

$$\phi = w(a, b, x_0, y_0). \tag{4.8}$$

Funkcijo w iz enačbe (4.8) lahko rešimo s štirimi zankami, npr. z zankami FOR (prva zanka FOR teče po vseh možnih vrednostih za a, druga po b itd.). Iz tega vidimo, da je najbolj preprosta implementacija standardne HT za elipse časovne zahtevnosti reda $\mathcal{O}(n^6)$ (štiri zanke FOR za rešitev funkcije w za eno točko, vse slikovne točke popišemo z dvema zankama FOR), kar pomeni tudi skupno časovno zahtevnost te metode.

Na koncu moramo v akumulatorskem polju opraviti še iskanje lokalnih maksimumov oblike $(x_{0max}, y_{0max}, a_{max}, b_{max}, \phi_{max})$. Lokalni maksimum oblike $(x_{0max}, y_{0max}, a_{max}, b_{max}, \phi_{max})$ v akumulatorju kaže na verjetno prisotnost elipse s središčem (x_{0max}, y_{0max}) in osema a_{max} in b_{max}, ki je rotirana za kot ϕ_{max}.

Iskanje lokalnih maksimumov v akumulatorskem polju je v najslabšem primeru časovne zahtevnosti reda $\mathcal{O}(n^5)$ (pet zank FOR). Ob natančnem pregledu te metode ugotovimo, da lahko s fiksiranjem obeh osi elipse a in b zmanjšamo prostorsko zahtevnost algoritma na $\mathcal{S}(n^3)$, vendar pa se ob tem poveča časovna zahtevnost.

Kombinatorična Houghova transformacija

Kombinatorično HT (KHT) sta prva predstavila avtorja Ben-Tzvi in Sandler [2]. Ideja standardne HT je, da za vsako slikovno točko poiščemo vse možne kombinacije parametrov v parametričnem prostoru oz. poiščemo n-dimenzionalno hiperpovršino. Ben-Tzvi in Sandler pa sta se tega problema lotila z drugega zornega kota. Če pogledamo enačbo (4.5), vidimo, da ob fiksirani točki $(\hat{x}, \hat{y})$ nastopa v tej enačbi n spremenljivk. Torej, če izberemo n slikovnih točk, ki ležijo na iskani krivulji oz. obliki, lahko iz sistema n enačb izračunamo vse parametre, $a_1, \ldots, a_n$, za iskano krivuljo. Če pa izbrane točke ne ležijo na isti krivulji, sistem ni rešljiv.

Kombinatorična HT poteka torej takole. Najprej tvorimo vse možne kombinacije n-teric postavljenih slikovnih točk. Za vsako n-terico izračunamo pripadajoč sistem enačb, ki nam da ali ustrezajočo n-terico parametrov $(v_1, \ldots, v_n)$ ali pa sistem ni rešljiv. Za vsako ustrezajočo n-terico parametrov $(v_1, \ldots, v_n)$ povečamo akumulatorsko celico $(v_1, \ldots, v_n)$ za 1. Ko pregledamo vse možne kombinacije slikovnih točk, moramo v akumulatorju poiskati le še lokalne maksimume, ki kažejo na morebitno prisotnost iskane krivulje v sliki. Avtorja Ben-Tzvi in Sadler sta algoritem izpeljala za primer n=2 (premice).

Kombinatorična HT ima kar nekaj prednosti pred standardno HT. Izkaže se, da je večina prispevkov, zbranih v okolici lokalnih maksimumov, hkrati pa so ti lokalni maksimumi veliko bolj izraziti kot pri standardni HT. KHT izkazuje tudi veliko boljšo razmerje med signalom in šumom v akumulatorskem polju, saj ena sama šumna točka v sliki nima skoraj nobenega vpliva na akumulator, medtem ko pri standardni HT vsaka šumna točka tvori sinusoido v akumulatorju (primer za premico) [2].

Avtorja Ben-Tzvi in Sandler sta v svojem članku [2] tudi navedla, da je KHT računsko učinkovitejša od standardne HT. Ovrednotimo to trditev. Če je število vseh postavljenih točk v sliki m, potem imamo opravka s

$$C_m^n = \binom{m}{n} = \frac{m!}{n!(m-n)!}$$

možnimi kombinacijami vseh n-teric slikovnih točk. To število pa je v slikah, kjer je izredno veliko postavljenih slikovnih točk (m je izredno velik), zelo veliko tudi za majhne n, npr. za n=2, kot sta predlagala avtorja (npr. za m=500 je $\binom{500}{2}$=124750 in $\binom{500}{5} \doteq 2{,}6{*}10^{11}$, pri čemer m niti ni pretirano velik). Število vseh kombinacij (število kombinacij je tudi enako številu potrebnih ciklov za izračun HT) je tako veliko, da bi celotno izračunavanje KHT preseglo potrebno izračunavanje za standardno HT. Zato sta ista avtorja predlagala segmentiranje celotne slike na majhne kvadratke oz. pravokotnike, s katerim se kompleksnost KHT zmanjša. Kompleksnost takšne, izboljšane KHT, pa je še vedno odvisna od porazdelitve postavljenih slikovnih točk v sliki. V najslabšem primeru so tudi po segmentaciji vse postavljene točke še vedno le znotraj enega segmenta in se torej kompleksnost KHT ne izboljša. Nekateri avtorji [23] pa za izboljšanje učinkovitosti KHT predlagajo, da iz celotne množice vseh postavljenih slikovnih točk velikosti m vzamemo v izračun KHT le eno podmožico, npr. velikosti $s \ll m$.

KHT lahko uporabimo na dva načina tudi za razpoznavanje elips. V prvem načinu uporabimo klasično idejo KHT, in sicer najprej tvorimo kombinacije peteric slikovnih točk, zatem pa še rešimo sistem petih enačb, iz katerega dobimo ustrezne parametre, ki jih zatem akumuliramo v akumulatorju. V drugem načinu pa rečemo, da je vsaka elipsa opisana s šopom tangent. Vsaka tangenta tvori lokalni maksimum v parametričnem prostoru ter tako vpliva na hrbet maksimuma. Ta hrbet maksimuma se zatem ponovno transformira s KHT, v tem drugem transformacijskem prostoru pa lahko razberemo parametre elipse.

Razpoznavanje elips z metodo Casasenta in Krishnapurama

Avtorja Casasent in Krishnapuram [6] sta predlagala novo metodo za iskanje krivuljastih objektov, in sicer s pomočjo HT ter njenega inverza. Glavna ideja te metode je, da lahko na vsak krivuljast objekt gledamo, kot da je sestavljen iz majhnih, tangentnih, ravnih segmentov, in ga je zato možno opisati s HT za daljice (angl. straight-line HT) – enačba (4.3). Opis objekta z daljicami je še posebno pravilen, kadar objekt leži na pravokotni mreži (to je vedno, kadar imamo opravka z digitalnimi slikami). Vsak raven segment v sliki se preslika v točko (θ, ρ) parametričnega Houghovega prostora, ki je proporcionalna dolžini ravnega segmenta. Ker je mnogo kratkih ravnih segmentov v sliki in ker se ti segmenti preslikajo v različne majhne vrhove v Houghovem prostoru, bodo ti vrhovi tvorili neki vzorec v Houghovem prostoru. Naj bo ta vzorec v Hougovem prostoru (po pragovni operaciji, ki odstrani tiste premice, ki gredo le skozi dve točki) opisan z

$$\rho = T(a_1, \ldots, a_n, \theta). \tag{4.9}$$

Če lahko predvidimo ta vzorec za objekt na vhodu, potem lahko izvedemo skaliranje vhodne slike in rotacijo v Houghovem prostoru ter nato še inverzno HT za razpoznavanje krivuljastega objekta.

Poglejmo si to metodo pobliže. Če je objekt rotiran za kot ϕ, potem se linijski segment preslika v drugo točko (θ', ρ') v Houghovem prostoru, in sicer v

$$\rho' = \rho \quad \text{in} \quad \theta' = \theta + \phi, \tag{4.10}$$

če pa je vhodni objekt premaknjen za (x_0, y_0), se linijski segment preslika v

$$\begin{aligned} \rho' = -\rho - t\cos(\theta - \alpha) \quad &\text{in} \quad \theta' = \theta + \pi, \\ &\text{če} \quad \rho + t\cos(\theta - \alpha) < 0; \\ \rho' = \rho + t\cos(\theta - \alpha) \quad &\text{in} \quad \theta' = \theta, \\ &\text{če} \quad \rho + t\cos(\theta - \alpha) \geq 0, \end{aligned} \tag{4.11}$$

kjer sta

$$t = \sqrt{x_0^2 + y_0^2} \quad \text{in} \quad \alpha = arctg\,(y_0/x_0). \tag{4.12}$$

Torej premaknjen in rotiran objekt lahko zapišemo z notacijo, uporabljeno v enačbi (4.9), v Houghovem prostoru $H(\theta', \rho')$ kot

$$\rho' = t\cos(\theta' - \alpha) + T(a_1, \ldots, a_n, \theta' - \phi)$$

oziroma

$$\rho' = t\cos(\theta' - \alpha) - T(a_1, \ldots, a_n, \theta' - \phi - \pi) \tag{4.13}$$

glede na obe varianti v enačbi (4.11). Algoritem poteka takole: najprej izračunamo preprosto HT za daljice, zatem opravimo preprosto transformacijo v Houghovem prostoru (vertikalni premik in skaliranje) po enačbi (4.14), na koncu pa izvedemo še inverzno HT. V rezultirajočem prostoru nato poiščemo vse potrebne parametre za objekt:

$$\rho = \rho' \mp T(a_1, \ldots, a_n, \theta' - \phi). \tag{4.14}$$

Za HT ni definiranega pravega inverza v literaturi [6], zato smo v tem algoritmu uporabili za izračun inverza Houghovega prostora metodo, ki vrne le približek. Vsaka točka v Houghovem prostoru $H(\theta, \rho)$ se preslika v premico v slikovnem prostoru. Pred inverzno transformacijo moramo zaradi odstranjevanja vpliva šuma uporabiti pragovno operacijo nad Houghovim prostorom. Če po tem receptu transformiramo vse točke, dobimo približek inverznega Houghovega prostora.

Metodo Casasenta in Krishnapurama, ki je namenjena za krivuljaste objekte, lahko uporabimo pri odkrivanju elips. Centrirano elipso zapišemo kot

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{4.15}$$

po transformiranju takšne elipse s HT za daljice pa lahko rezultirajoči vzorec v Houghovem prostoru opišemo z enačbo (4.16):

$$\rho = T_e(a, b, \theta) = \sqrt{a^2 \cos^2(\theta) + b^2 \sin^2(\theta)}. \tag{4.16}$$

Elipsa, rotirana za kot ϕ in premaknjena v točko (x_0, y_0), pa je v Houghovem prostoru opisana z enačbo (4.13), kjer splošen izraz za funkcijo T zamenjamo s $T_e(a, b, \theta' - \phi)$. Iskanje elipse poteka enako kot pri že opisanem splošnem algoritmu (HT za daljice, pragovna operacija, transformacija HT prostora po enačbi (4.13) - oba dela seštejemo, inverzna HT). V inverznem prostoru se po takšni transformaciji pojavi vrh, ki kaže na verjetno prisotnost elipse v sliki, iz koordinat vrha pa razberemo položaj središča elipse (x_0, y_0), parametri a, b ter ϕ pa morajo biti vnaprej znani (glej enačbo (4.16)) – to nam hkrati tudi omogoča, da funkcijo $T_e(a, b, \theta' - \phi)$ vnaprej izračunamo oz. tabeliramo. Če pa ti parametri niso znani, potem opravimo grobo iskanje po obsegu vnaprej predpisanih vrednosti za te parametre.

Opisana metoda je zelo preprosta in zaradi tega izredno učinkovita. S preprosto HT za daljice smo se izognili večdimenzionalnemu Houghovemu prostoru in generiranju hiperpovršin (glej prejšnja podpoglavja), kar pomeni bistveno manjšo prostorsko zahtevnost (sedaj reda n^2). Tudi transformacije, ki jih opravljamo v Houghovem prostoru, so zelo preproste, gre zgolj za premike in skaliranje, kljub temu pa je časovna zahtevnost reda $\mathcal{O}(n^5)$ (tri zanke FOR za parametre a, b in ϕ, za vsako takšno kombinacijo pa je treba izračunati še inverz Houghovega prostora – dve zanki FOR). Časovno zahtevnost algoritma lahko zmanjšamo, če uporabimo na posameznih mestih algoritma že tabelirane vrednosti funkcije T_e.

Avtorji Pao, Li in Jayakumar [52] so nadgradili to metodo s tem, da so opravili še dodatno dekompozicijo parametričnega prostora. Z izbiro več točk iz Houghovega prostora so eliminirali izraz $T_e(a, b, \theta' - \phi)$, za osi elipse a in b (ali pa za središče elipse (x_0, y_0)) pa prispevajo direktno. Ta metoda vsebuje le iskanje po enodimenzionalnem prostoru kota rotacije ϕ. Avtorji navajajo, da je njihov izboljšan algoritem 3 do 4-krat hitrejši od osnovnega.

Izboljšana Houghova metoda po Huangu

Avtor Huang [29] je v svojem članku zelo domiselno nadgradil osnovno HT za razpoznavanje elips. Za identificiranje eliptičnih značilnosti uporabljamo dualnost med središčem elipse in točkami na meji. Če na točke z meje elipse gledamo kot na središča nekih drugih elips, ki imajo enake parametre a, b in ϕ, potem lahko imajo te elipse skupno točko, ki je središče originalne elipse (z istimi parametri). Za točko na meji (x_i, y_i) ter za znani osi a in b lahko elipso v parametričnem prostoru tvorimo kot

$$\begin{aligned} x_0 &= x_i + a\cos(\varphi), \\ y_0 &= y_i + b\sin(\varphi), \end{aligned} \tag{4.17}$$

pri čemer kot φ teče od 0 do 2π. Vendar za izračun presečiša vseh "enakih" elips ni potrebno, da kot φ teče po celotnem območju $[0, 2\pi]$. Pokažemo lahko, da normalen gradient v mejni točki (x_i, y_i) elipse, pomnožen s faktorjem a^2/b^2, kaže na središče elipse. To lastnost lahko uporabimo pri zmanjšanju kota φ.

Huangov algoritem poteka takole: najprej oštevilčimo vse povezane mejne krivulje. Tako dobimo več povezanih komponent. Zatem za vsako takšno komponento po zgoraj opisanem postopku poskušamo identificirati elipso oz. eliptični lok. Lokalni maksimum (npr. točka večja od določenega praga) v parametričnem prostoru kaže na možno prisotnost elipse oz. eliptičnega loka v sliki. V zadnjem koraku v slikovnem prostoru le še verificiramo, ali ta lokalni maksimum dejansko predstavlja elipso v sliki.

Ta metoda je zelo učinkovita, saj je prostorska zahtevnost reda $\mathcal{S}(n^2)$, medtem ko je časovna zahtevnost reda $\mathcal{O}(n^3)$ za eno komponento (tri zanke FOR po parametrih a, b in ϕ, zanka FOR po kotu φ pa je zanemarljivo majhna). Metoda je posebej učinkovita, če jo implementiramo na paralelnih računalnikih. Odlika tega pristopa je, da deluje dobro tudi, ko je meja pokvarjena oz. nepovezana zaradi šuma ali prekrivanja.

Izpopolnjena Houghova transformacija za razpoznavanje elips na osnovi 2D polja

Avtorji Yip, Tam in Leung [69] predlagajo nov algoritem za razpoznavanje elips v slikah, s katerim želijo predvsem zmanjšati prostorsko zahtevnost (potreben pomnilniški prostor za akumulator) in izboljšati učinkovitost HT. Osnovna ideja tega algoritma je, da s pomočjo štirih simetričnih točk z meje elipse reduciramo dimenzijo akumulatorja in hkrati razpoznamo elipse v sliki. Pri tem uporabimo pet geometrijskih lastnosti simetričnih točk elipse: 1) paralelnost tangent, ki potekajo skozi te točke, 2) enako oddaljenost točk od središča, 3) določenost položaja središča, 4) kot med tangento in daljico, ki povezuje središče s točko, je za vse štiri simetrične točke enak, in 5) štiri simetrične točke tvorijo pravokotnik, ki je enako orientiran kot elipsa.

Na osnovi teh geometrijskih lastnosti lahko za določeno elipso izračunamo vse potrebne parametre (x_0, y_0, a, b, ϕ) – podrobne formule so v [69]. V tem trenutku je treba akumulirati rezultat. Klasičen pristop bi bil, da povečamo ustrezno celico v petdimenzionalnem akumulatorskem polju, Yip in ostali pa so se temu elegantno izognili s tem, da so iz vseh znanih parametrov izračunali štiri simetrične točke na elipsi in jih zatem shranili v dvodimenzionalnem akumulatorju. Na koncu je treba opraviti le še iskanje lokalnih maksimumov po tem dvodimenzionalnem akumulatorju. Štirje lokalni maksimumi lahko tvorijo eno elipso v sliki, zato je treba pregledati vse možne kombinacije teh četveric (iz ene četverice maksimumov pa lahko izračunamo vse potrebne parametre za elipso).

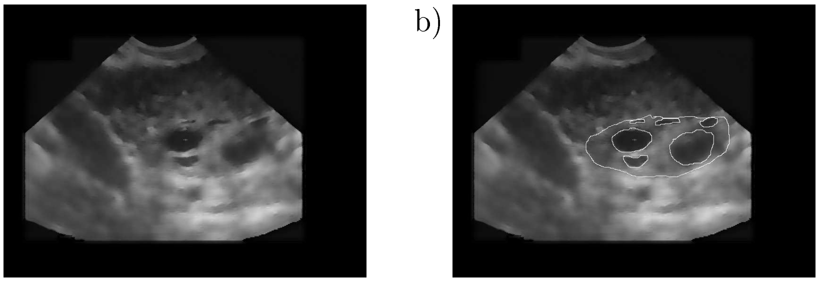
Ta metoda razpoznavanja elips v slikah je izredno učinkovita, saj je njena prostorska zahtevnost reda $\mathcal{S}(n^2)$, časovna zahtevnost pa je reda $\mathcal{O}(n^3)$ (tri zanke FOR po slikovnih točkah). Algoritem vrača zadovoljive rezultate tudi v zapletenih situacijah (prisotnost šuma itd.). Metodo lahko v določenih primerih (pri rotacijski in refleksijski simetriji objekta) uporabimo tudi za neanalitično opisne objekte.

4.3 Houghova transformacija za razpoznavanje jajčnih mešičkov

V prejšnjih podpoglavjih smo predstavili teoretične osnove HT, opisali njene dobre strani in slabosti, prikazali smo tudi njeno uporabnost, poseben del pa smo namenili razpoznavanju elips s HT. V tem podpoglavju pa bomo preizkusili HT tudi v praksi, in sicer na ultrazvočnih posnetkih jajčnikov, kjer bomo skušali razpoznati jajčne mešičke.

Na sliki 4.1a vidimo tipično ultrazvočno sliko jajčnika. Na levem delu te sektorske ultrazvočne slike vidimo podolgovato temno liso, ki je krvna žila, zgoraj je maternična sluznica, ki se kot temen širok pas razteza do vrha slike, spodaj desno so svetle lise, ki so nastale zaradi šuma ultrazvočne naprave (sonde), v sredini slike pa je jajčnik, ki je ovalne oblike, znotraj njega pa se nahaja 6 jajčnih mešičkov (temne pike). Na sliki 4.1b je ta ista slika, kjer je ekspert (ginekologinja) obkrožil z belo črto jajčnik, znotraj njega pa še jajčne mešičke.

a) b)



Slika 4.1: Tipična ultrazvočna slika jajčnika. Slika je velikosti 576x768 pikslov z 256 sivimi nivoji. V sliki je že odstranjen pegasti šum. a) Originalna slika. b) Ista slika, v kateri je ekspert obkrožil jajčnik in 6 jajčnih mešičkov.

Iz obeh slik lahko opazimo, da imajo jajčni mešički nekakšno elipsasto obliko. Elipsasta oblika nas je takoj napeljala na idejo, da bi uporabili HT za iskanje mešičkov v slikah. Naš osnovni problem iskanja jajčnih mešičkov je tako postal problem iskanja elips v ultrazvočnih slikah.

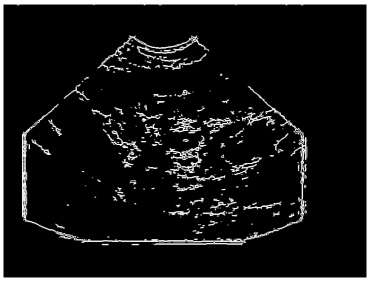
Drugi razlog, s katerim lahko opravičimo uporabo HT za naš problem, je prisotnost šuma v ultrazvočnih slikah. Ultrazvočne slike so že ponavadi zelo šumne (podpoglavje 2.1.4), zato je smiselno, da uporabimo HT, ki se zelo dobro obnaša tudi v šumnem okolju (podpoglavje 4.1.2).

4.3.1 Iskanje jajčnih mešičkov v ultrazvočnih slikah z metodo Casasenta in Krishnapurama

V uvodu smo že nakazali, da lahko problem razpoznavanja jajčnih mešičkov prevedemo na iskanje elips z metodo HT. Pri našem delu smo se odločili za metodo Casasenta in Krishnapurama, in sicer zaradi njene majhne prostorske zahtevnosti in relativno preprostega iskanja po inverznem Houghovem prostoru (podpoglavje 4.2).

Houghov prostor smo diskretizirali v smeri ρ od 0 do dolžine diagonale slike po koraku 1, v smeri θ pa od 0 do 359^o v koraku po 1^o. Iščemo le nerotirane elipse. Parametri a in b elips niso vnaprej znani, zato smo uporabili grobo iskanje po obsegu parametrov a in b (a in b tečeta od 5 do 100 po koraku 5 – te vrednosti smo dobili iz dodatnega znanja o velikostih jajčnih mešičkov – so velikosti od 2 mm do 10 mm). V inverznem Houghovem prostoru smo poiskali le maksimum (položaj maksimuma ustreza tudi položaju morebitne elipse – glej podpoglavje 4.2) in ga ovrednotili, ali je lahko elipsa ali ne. Vse ustrezajoče maksimume smo shranili ter jih vrisali na sliko.

Na sliki 4.2 je tipična binarna slika robov, nad katero smo izvajali metodo Casasenta in Krishnapurama (metode HT pričakujejo binarno sliko na vhodu – podpoglavje 4.1 in 4.2). Binarno sliko robov smo dobili po postopku, ki je enak kot pri našem razpoznavalnem sistemu "xultra", opisanem v poglavju 5. Originalno ultrazvočno sliko smo najprej obdelali s filtrom za odstranjevanje peg HRGMF (podpoglavje 2.3). S tem smo odstranili velik del šuma (v nadaljevanju bomo tudi pokazali, kaj se zgodi, če ga ne odstranimo) – takšno sliko vidimo na sliki 4.1a. Zatem smo na tej sliki uporabili Kirschev operator (različica Sobelovega operatorja) [60, 63]. To sliko robov smo zatem binarizirali (prag 40) in ji še stanjšali robove na velikost 1 piksla.

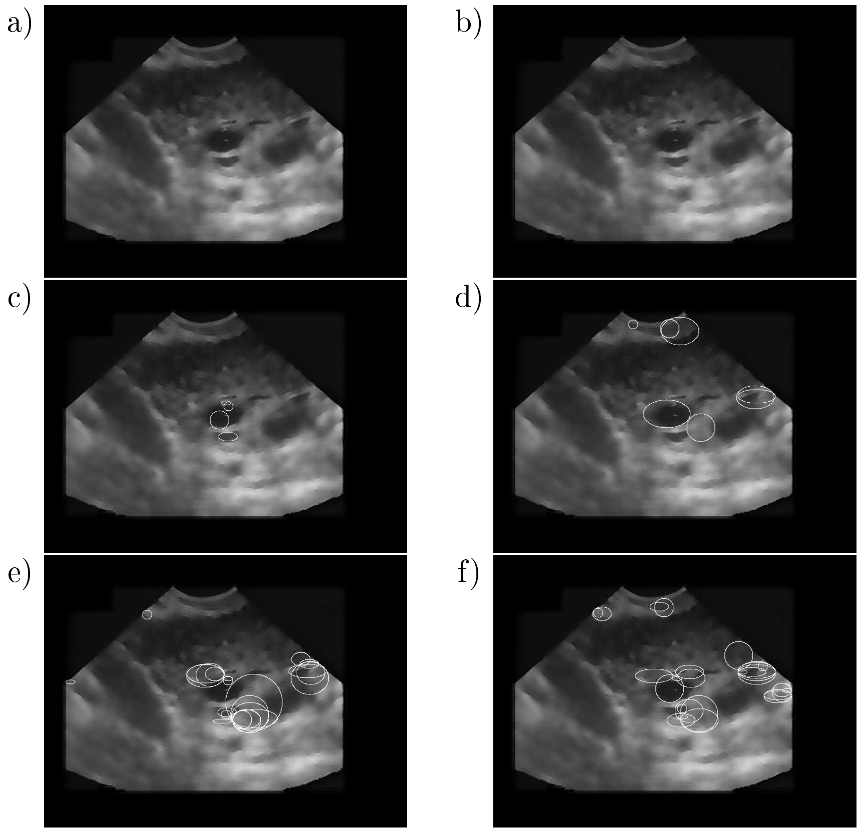


Slika 4.2: Binarna slika robov za sliko 4.1a.

4.3.2 Rezultati

V tem podpoglavju bomo predstavili rezultate, ki smo jih dobili z metodo Casasenta in Krishnapurama ter zgoraj opisanimi predpostavkami. Metodo smo implementirali v programskem jeziku C za okolje Unix. Teste smo poganjali na stroju HP 715/100XC s 128 MB RAM-a.

Najprej bomo predstavili rezultate za sliko 4.1a oz. za njeno pripadajočo sliko robov (slika 4.2). Na sliki 4.3 vidimo rezultat iskanja elips. Slike 4.3a, b in c prikazujejo rezultat Casasentove metode za različne prage T_i (prag T potrebujemo za pragovno operacijo, ki jo moramo izvesti v tej metodi, preden tvorimo Houghov inverzni prostor – podpoglavje 4.2). Prage T smo izbrali tako, da velja $T_2 > T_3 > T_5$, pri čemer T_2 pomeni, da smo za prag vzeli polovico maksimalne vrednosti v Houghovem prostoru (podobno za T_3 in T_5). Iz slik



Slika 4.3: Rezultat iskanja elips v sliki 4.1a: a) prag T_2, b) prag T_3, c) prag T_5. V slikah d, e in f smo originalno sliko razdelili na 6 enakih podslik, vsako podsliko smo obdelali s to metodo: d) prag T_2, e) prag T_3, f) prag T_5.

4.3a, b in c vidimo, da rezultati niso dobri. Na slikah 4.3a in b metoda ni našla celo nobene elipse, medtem ko je na sliki c našla nekaj elips, ki ležijo na območju jajčnih mešičkov. Iz tega lahko sklepamo, da ima višina praga T pomembno vlogo pri tej metodi. Praga T_2 in

T_3 sta razmeroma visoka, saj sta porezala veliko točk v Houghovem prostoru. To hkrati tudi pomeni, da je bilo razmeroma malo pravih maksimumov v inverznem Houghovem prostoru (večino smo po ovrednotenju zavrgli). Prag T_5 je pa dovolj nizek, da smo dobili neničeln rezultat.

Kljub temu, da je metoda v sliki 4.3c našla elipse na območju meščkov, pa so te elipse pokrile le dva jajčna mešička izmed šestih (glej sliko 4.1b). Zato smo preučili metodo Casasenta in Krishnapurama še bolj natančno. Ta metoda temelji na opisu elipse s kratkimi tangentnimi odseki (kratke ravne črte), zato pri preslikavi iz slikovnega v parametričen prostor uporablja t. i. HT za daljice. Ta vrsta preslikave preslika vsako točko iz slikovnega prostora na enak način (enačba (4.3)). To hkrati tudi pomeni, da vse točke iz iste premice v slikovnem prostoru vplivajo na isto točko v parametričnem prostoru. Ker so naše slike razmeroma velikih dimenzij (576x768 pikslov), pomeni, da lahko tudi zelo oddaljene točke (npr. točke z robov slike, ki jih je v naših slikah zelo veliko – glej sliko 4.2) vplivajo na trenutno preslikovano točko. Casasentova metoda temelji na lokalnosti, medtem ko HT za daljice tega ne zagotavlja (vsaj v naših slikah ne).

To je bila glavna motivacija, da smo poskušali sam postopek iskanja elips nekoliko izboljšati. Ker je glavni problem premajhna lokalnost metode, smo se odločili, da bomo sliko razdelili na podslike, vsako podsliko pa bomo obdelovali na enak način kakor osnovno. Osnovno sliko smo razdelili na 6 kvadratnih podslik (število podslik je bilo izbrano poljubno). Rezultat tega pristopa je prikazan na slikah 4.3d, e in f. Vidimo lahko, da je metoda v tem primeru našla veliko več elips na območjih jajčnih mešičkov, nekatere elipse pa kar dejansko ustrezajo mešičkom.

Vse rezultate smo zbrali tudi v tabeli (glej tabelo 4.1). Tabela ima več polj. Polje "Slika" vsebuje ime slike, za katero so navedeni podatki, polje "Čas obdelave" vsebuje podatke o porabljenem času za iskanje elips (v sekundah), polje "Σ_{vseh}" pove število vseh najdenih elips v sliki, "Σ_{pravih}" je število elips, ki ležijo na območjih jajčnih mešičkov, v polju "% pravih" je izračunano razmerje med poljema "Σ_{pravih}" in "Σ_{vseh}", polje "Σ_{fol}" vsebuje število jajčnik mešičkov, ki smo jih odkrili z elipsami, polje "% fol" pa je razmerje med poljema "Σ_{fol}" in številom vseh jajčnih mešičkov v sliki (ta podatek nam poda ekspert).

Iz tabele 4.1 vidimo, da čas obdelovanja narašča z izbiro nižjega praga. To smo lahko pričakovali, saj nižji prag pomeni tudi večje število elementov Houghovega prostora, ki jih je treba preslikati z inverzno HT. Hkrati pa nižji prag daje tudi boljše rezultate (več elips na območjih mešičkov) – tudi to je očitno, saj lahko po inverzni HT izbiramo med več pravimi maksimumi. Tudi delitev slike na podslike se je izkazala za primerno, saj je bistveno izboljšala rezultate, vendar pa se je zato povečal čas obdelovanja. Kljub sorazmerno dobrim rezultatom pa nas še vedno motijo piksli z robov slike (rob slike pomeni mejo med črnim ozadjem in vidnim delom slike – glej sliko 4.1a). Teh nepomembnih pikslov je zaradi zelo izrazite meje v naših slikah izredno veliko, hkrati pa s svojo številnostjo močno vplivajo na rezultat metode. Ker smo želeli natančneje videti vpliv teh pikslov na rezultate, smo naredili še dodaten poizkus, kjer smo iskali elipse v slikah, iz katerih smo prej odstranili robove slike (dejansko smo izrezali viden del slike – brez robov). Rezultat tega poskusa je v sliki 4.4 in tabeli 4.2. Vidimo lahko, da se sedaj skoraj vse elipse naha-

	Slika	Prag	Čas obdelave (s)	Σ_{vseh}	Σ_{pravih}	% pravih	Σ_{fol}	% fol
	4.3a	T_2	77	0	0	0	0	0
	4.3b	T_3	77	0	0	0	0	0
	4.3c	T_5	164	4	3	75	2	33.3
podslike	4.3d	T_2	158	7	3	42.9	2	33.3
	4.3e	T_3	412	23	11	47.8	5	83.3
	4.3f	T_5	1051	28	14	50	5	83.3

Tabela 4.1: Rezultati iskanja elips po metodi Casasenta in Krishnapurama za sliko 4.1a. Število vseh jajčnih mešičkov je 6.

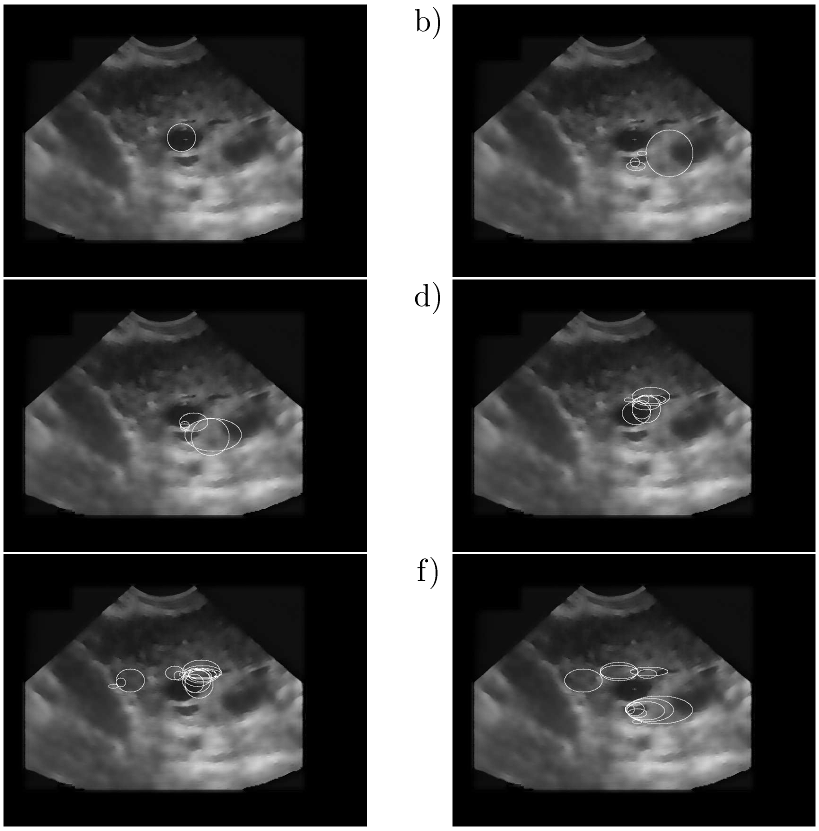
jajo znotraj jajčnika. Opazimo lahko tudi, da se je čas obdelovanja za osnovno varianto metode glede na prvi poizkus povečal. Razlog za to je, da sedaj ni več izrazitih maksimumov v Houghovem prostoru, zato tudi po pragovni operaciji v tem prostoru ostane veliko elementov. V varianti "podslik" pa se je čas zmanjšal, kar je očitno, saj se je tudi dimenzija podslik zmanjšala, hkrati pa je ostala razporeditev pikslov v posameznih podslikah enaka.

	Slika	Prag	Čas obdelave (s)	Σ_{vseh}	Σ_{pravih}	% pravih	Σ_{fol}	% fol
	4.4a	T_2	110	1	1	100	1	16.7
	4.4b	T_3	462	4	3	75	2	33.3
	4.4c	T_5	1630	5	5	100	3	50
podslike	4.4d	T_2	100	8	7	87.5	3	50
	4.4e	T_3	239	15	9	60	3	50
	4.4f	T_5	573	12	7	58.3	4	66.7

Tabela 4.2: Rezultati iskanja elips po metodi Casasenta in Krishnapurama za sliko 4.1a. Število vseh jajčnih mešičkov je 6. Pri obdelavi smo upoštevali območje brez robov slike.

Podrobno smo si ogledali rezultate na eni sliki. Preizkusimo to metodo še na nekaj slikah. Na sliki 4.5a imamo še eno testno sliko. V sliki 4.5b vidimo pripadajočo sliko robov, na sliki 4.5c pa je ekspert vrisal jajčnik in mešičke. Rezultati iskanja elips so na sliki 4.6 in v tabeli 4.3. Vidimo, da so rezultati zelo slabi. To je lahko posledica dvojega: robovi okoli jajčnih mešičkov so zelo neizraziti oz. jih ni veliko in, ker je robov v sredini slike zelo malo, prevzamejo glavno utež piksli z robov slike, zato se tudi večina elips nahaja na robu slike. Preverimo to trditev še z enim poizkusom.

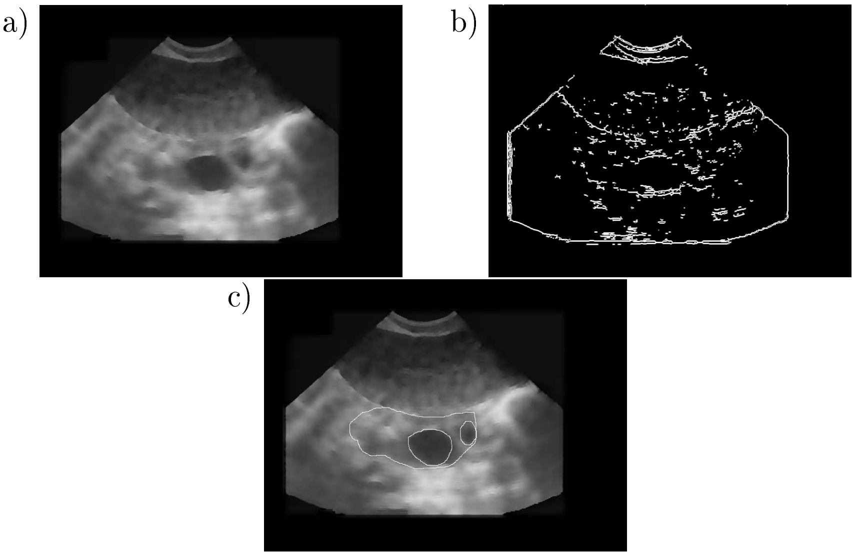
Za naslednji poizkus smo si izbrali ultrazvočno sliko jajčnika iste ženske, vendar v tretjem časovnem obdobju – glej sliko 4.7. Iz slike robov 4.7b opazimo, da so tudi v tej sliki robovi jajčnih mešičkov neizraziti, torej lahko pričakujemo, da se bo večina elips pojavila ob robu slike. Rezultati so na sliki 4.8 in v tabeli 4.4. Vidimo, da je rezultat



Slika 4.4: Rezultat iskanja elips v sliki 4.1a. Pri obdelavi smo upoštevali območje brez robov slike: a) prag T_2, b) prag T_3, c) prag T_5. V slikah d, e in f smo originalno sliko razdelili na 6 enakih podslik, vsako podsliko smo obdelali s to metodo: d) prag T_2, e) prag T_3, f) prag T_5.

	Slika	Prag	Čas obdelave (s)	Σ_{vseh}	Σ_{pravih}	% pravih	Σ_{fol}	% fol
	4.6a	T_2	72	0	0	0	0	0
	4.6b	T_3	73	0	0	0	0	0
	4.6c	T_5	100	0	0	0	0	0
podslike	4.6d	T_2	138	6	0	0	0	0
	4.6e	T_3	220	11	0	0	0	0
	4.6f	T_5	621	19	1	5.3	1	50

Tabela 4.3: Rezultati iskanja elips po metodi Casasenta in Krishnapurama za sliko 4.5a. Število vseh jajčnih mešičkov je 2.



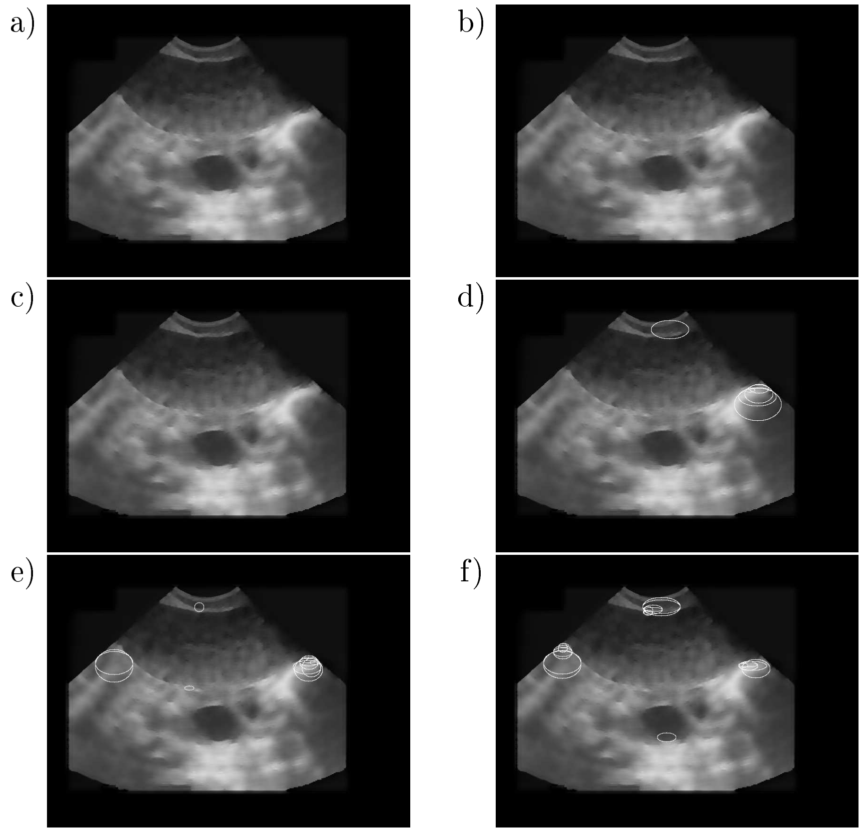
Slika 4.5: Ultrazvočna slika jajčnika za isto žensko v drugem časovnem obdobju: a) originalna slika, obdelana s filtrom HRGMF; b) binarna slika robov; c) ista slika, v kateri je ekspert obkrožil jajčnik in 2 jajčna mešička.

še slabši kot v prejšnjem primeru, saj nismo z nobeno elipso prišli na območje mešičkov. Kljub tako poraznemu rezultatu pa je bilo naše pričakovanje, da bo večina elips ob robu slike, izpolnjeno.

	Slika	Prag	Čas obdelave (s)	Σ_{vseh}	Σ_{pravih}	% pravih	Σ_{fol}	% fol
	4.8a	T_2	73	0	0	0	0	0
	4.8b	T_3	75	1	0	0	0	0
	4.8c	T_5	135	3	0	0	0	0
	4.8d	T_2	134	5	0	0	0	0
podslike	4.8e	T_3	221	19	0	0	0	0
	4.8f	T_5	640	16	0	0	0	0

Tabela 4.4: Rezultati iskanja elips po metodi Casasenta in Krishnapurama za sliko 4.7a. Število vseh jajčnih mešičkov je 3.

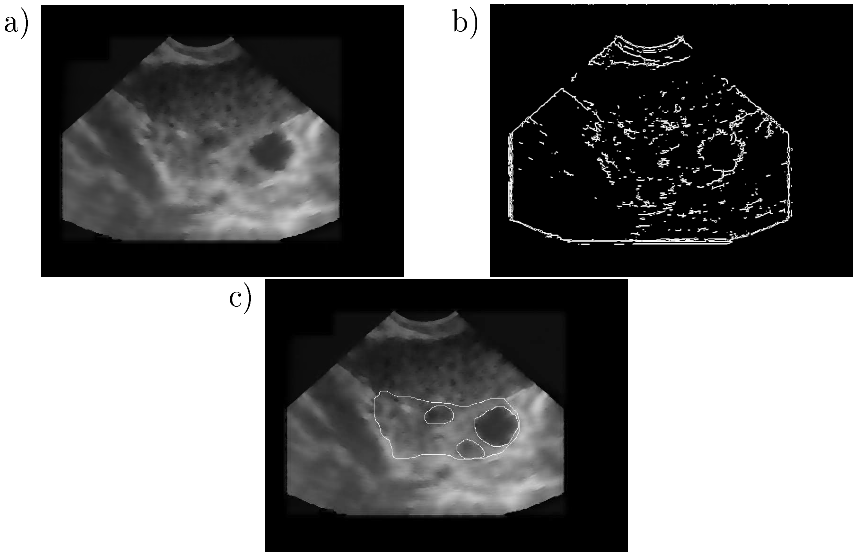
Našo trditev, da po odstranitvi robov slike dobimo boljše rezultate, smo potrdili s poizkusom, kjer smo v sliki 4.7a odstranili robove slike. Z robom slike mislimo na mejo med vidnim delom slike ter črnim ozadjem (na sliki 4.7 je vidni del slike trikoten). Rezultati so na sliki 4.9 in v tabeli 4.5. Rezultati so dobljeni le za prag T_3, že iz teh pa lahko vidimo, da so neprimerno boljši od prejšnjih. V sliki 4.9a je ta metoda popolnoma pravilno obkrožila jajčni mešiček.



Slika 4.6: Rezultat iskanja elips v sliki 4.5a: a) prag T_2, b) prag T_3, c) prag T_5. V slikah d, e in f smo originalno sliko razdelili na 6 enakih podslik, vsako podsliko smo obdelali s to metodo: d) prag T_2, e) prag T_3, f) prag T_5.

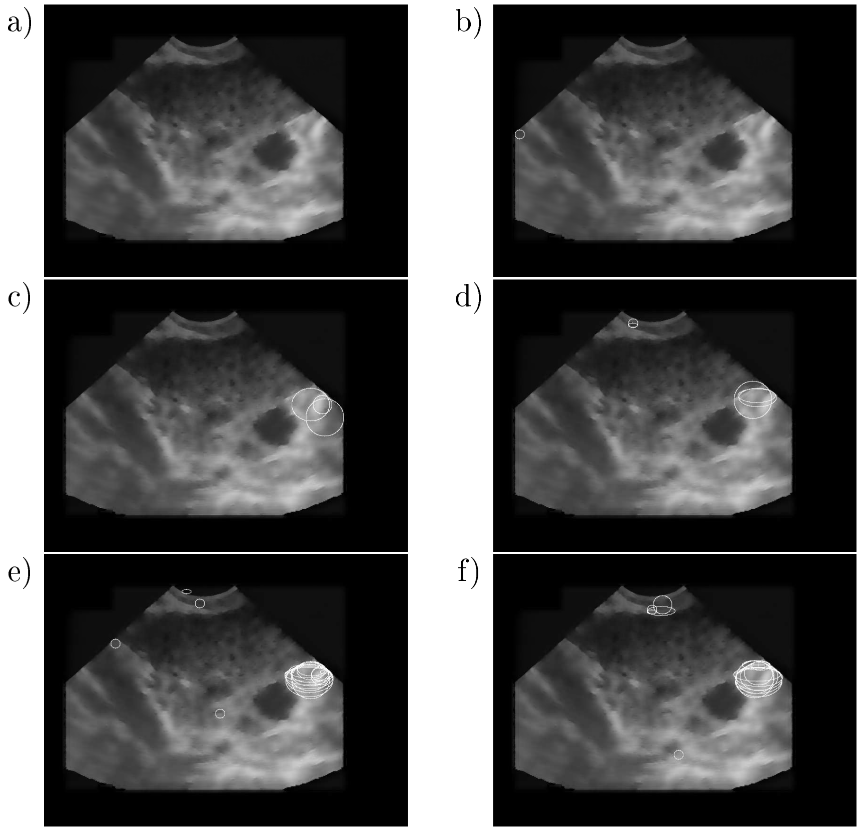
	Slika	Prag	Čas obdelave (s)	Σ_{vseh}	Σ_{pravih}	% pravih	Σ_{fol}	% fol
	4.9a	T_3	134	1	1	100	1	33.3
podslike	4.9b	T_3	132	8	0	0	0	0

Tabela 4.5: Rezultati iskanja elips po metodi Casasenta in Krishnapurama za sliko 4.7a. Število vseh jajčnih mešičkov je 3. Pri obdelavi smo upoštevali območje brez robov slike.

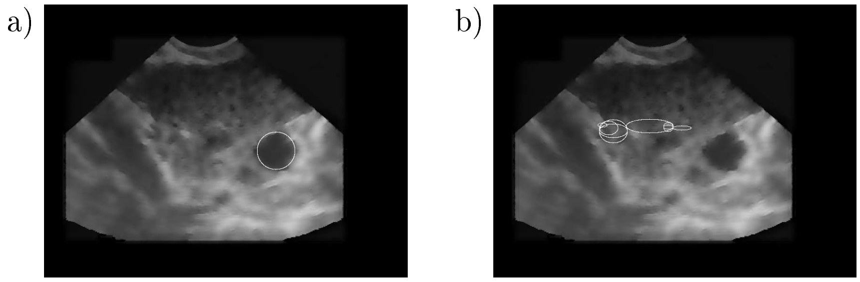


Slika 4.7: Ultrazvočna slika jajčnika za isto žensko v tretjem časovnem obdobju: a) originalna slika, obdelana s filtrom HRGMF; b) binarna slika robov; c) ista slika, v kateri je ekspert obkrožil jajčnik in 3 jajčne mešičke.

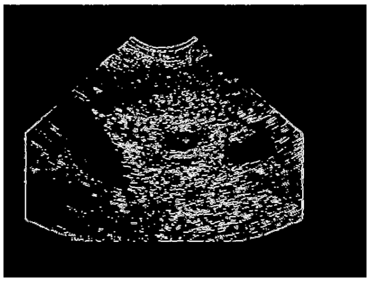
V prejšnjem podpoglavju smo navedli, da bomo namesto z originalnimi slikami, ki jih dobimo direktno iz ultrazvočne naprave ali po digitaliziranju video traku, imeli raje opravka s slikami, kjer smo s filtrom odstranili pegasti šum. Takšne slike smo imenovali originalne. Sedaj pa si oglejmo, kako se obnaša metoda, če tega šuma ne odstranimo. Sliko robov tudi v tem primeru tvorimo na enak način (podpoglavje 4.3.1), le da se sedaj spremeni prag pri binarizaciji, ki je 150. Na sliki 4.10 vidimo takšno sliko robov. Ta slika je vhod v našo metodo. Naredili smo poizkus z originalno metodo pri pragu T_5 in z metodo, kjer smo sliko razdelili na podslike (pri istem pragu). Rezultati so na sliki 4.11 in v tabeli 4.6. Iz teh podatkov vidimo, da se je čas obdelovanja povečal pri osnovni varianti kar za 26-krat, pri metodi z delitvijo na podslike pa za 2-krat. Povečanje časa obdelovanja lahko pripišemo večjemu številu "belih" pikslov v novi sliki robov (glej sliko 4.10) – več belih pikslov pa je zato, ker operatorji za detektiranje robov zaznajo tudi majhne spremembe v sivinah kot robove (te majhne spremembe v sivinah smo v prejšnjih poizkusih odstranili s filtrom HRGMF). Število belih pikslov lahko zmanjšamo tudi z dvigovanjem praga, vendar se v tem primeru zniža tudi število pikslov, ki dejansko predstavljajo robove jajčnih mešičkov. Če pogledamo kvalitativne rezultate, vidimo, da smo z elipsami pokrili vse jajčne mešičke, vendar pa je hkrati izredno veliko elips tudi zunaj našega interesnega območja (jajčnika) – to je posledica velikega števila pikslov, ki so nastali zaradi šuma. Ob primerjavi tabel 4.6 in 4.1 (oz. 4.2) lahko hitro vidimo, da se glede na odnos stroški-rezultati veliko bolj izplača iskati elipse v slikah z že odstranjenimi pegami, saj ustrezni filter porabi veliko manj časa, kot je dodatni čas, ki nastane zaradi iskanja v neobdelani sliki, hkrati pa so tudi rezultati v prvem primeru v povprečju boljši.



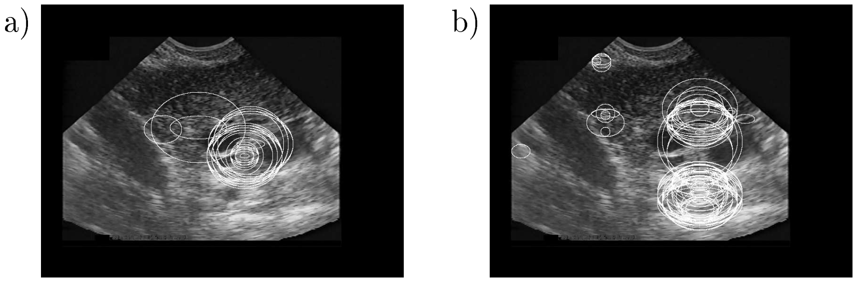
Slika 4.8: Rezultat iskanja elips v sliki 4.7a: a) prag T_2, b) prag T_3, c) prag T_5. V slikah d, e in f smo originalno sliko razdelili na 6 enakih podslik, vsako podsliko smo obdelali s to metodo: d) prag T_2, e) prag T_3, f) prag T_5.



Slika 4.9: Rezultat iskanja elips v sliki 4.7a. Pri obdelavi smo upoštevali območje brez robov slike: a) prag T_3. V sliki b smo originalno sliko razdelili na 6 enakih podslik, vsako podsliko smo obdelali s to metodo: b) prag T_3.



Slika 4.10: Binarna slika robov za sliko 4.1a (pege niso odstranjene).



Slika 4.11: Rezultat iskanja elips v sliki 4.1a, pri čemer originalna slika ni bila obdelana s filtrom za odstranjevanje peg: a) prag T_5. V sliki b smo originalno sliko razdelili na 6 enakih podslik, vsako podsliko smo obdelali s to metodo: b) prag T_5.

4.3.3 Razprava

V podpoglavju 4.3.2 smo si ogledali rezultate iskanja elips v ultrazvočnih slikah jajčnikov z metodo Casasenta in Krishnapurama. Čas iskanja s to metodo je proporcionalen številu belih pikslov v sliki robov, zato je dobro, če lahko že v fazi predobdelave slik odstranimo veliko šuma. To zahtevo smo izpolnili s pomočjo filtra za odstranjevanje peg.

Čeprav v splošnem za HT velja, da se zelo dobro obnaša tudi v šumnem okolju, smo pri našem delu videli, da imajo šumni oz. nepomembni piksli z robov zelo velik vpliv na končni rezultat. Zato predlagamo, da že v fazi predobdelave odstranimo nepomembne piksle, če je to mogoče. V našem primeru je, in sicer z odstranjevanjem peg ter izrezanjem vidnega dela slike.

Izredno pomemben faktor pri tej metodi je tudi izbira primernega praga za pragovno operacijo, ki jo moramo opraviti v Houghovem prostoru pred inverzno operacijo. Naredili smo poizkuse za tri različne prage (izbrane poljubno): za prage, ki so polovica, tretjina

	Slika	Prag	Čas obdelave (s)	Σ_{vseh}	Σ_{pravih}	% pravih	Σ_{fol}	% fol
	4.11a	T_5	4318	26	14	53.9	6	100
podslike	4.11b	T_5	2213	85	18	21.2	5	83.3

Tabela 4.6: Rezultati iskanja elips po metodi Casasenta in Krishnapurama za sliko 4.1a. Število vseh jajčnih mešičkov je 6. Originalna slika ni bila obdelana s filtrom za odstranjevanje peg.

in petina maksimalne vrednosti v Houghovem prostoru. Rezultati se z nižanjem praga izboljšujejo, hkrati pa se povečuje porabljen čas. Treba je poiskati najboljši prag glede na kriterij stroški-rezultati. Prag je smiselno določati avtomatsko (podpoglavje 3.1.1).

Zadnja pomembna odločitev, ki jo moramo sprejeti pri tej metodi, je, na kakšen način bomo iskali maksimume v inverznem Houghovem prostoru. Odločili smo se poiskati le globalni maksimum (za znana parametra a in b) ter potem verificirali, ali je elipsa ali ne. Verifikacijo maksimuma smo naredili tako, da smo potegnili navidezno elipso s središčem v tem maksimumu ter osema a in b, zatem pa smo prešteli, koliko belih pikslov dejansko leži na tej elipsi. Če je bilo to število večje od vnaprej postavljenega praga (npr. četrtina elipse), smo elipso sprejeli, sicer pa jo zavrnili. Seveda pristop, kjer upoštevamo le globalni maksimum, ni najboljši, saj pri takšnem pristopu spregledamo lokalne maksimume, ki bi morebiti ustrezali elipsam in s tem jajčnim mešičkom. Iskanje (in verificiranje) vseh lokalnih maksimumov v inverznem Houghovem prostoru pa bi bistveno povečalo časovno zahtevnost metode.

Na koncu te razprave si oglejmo še kvaliteto naših rezultatov. Iz vseh slik z rezultati (slika 4.3, 4.4 itd.) lahko vidimo, da smo z metodo Casasenta in Krishnapurama našli veliko elips v okolici jajčnih mešičkov oz. elips, ki so se delno pokrivale z mešički, le nekaj elips pa je popolnoma obkrožilo jajčne mešičke (s tem mislimo, da so te elipse popolnoma identificirale jajčne mešičke). To pomeni, da s pomočjo večine naših razpoznanih elips ne moremo ničesar reči o velikosti, obliki in ostalih parametrih jajčnega mešička. Kljub temu pa smo prepričani, da naše delo ni bilo zaman. Ker večina najdenih elips leži v okolici jajčnih mešičkov ali ima skupen del z jajčnim mešičkom, lahko vsako elipso obdelamo še z dodatnim algoritmom, ki bo iz tega začetnega položaja konvergirala v pravo mejo jajčnega mešička. Predlagamo naslednji postopek. Vsaka najdena elipsa naj predstavlja začetni približek za jajčni mešiček. Ta začetni približek naj bo zatem vhod v eno izmed metod aktivnih kontur [63]. Algoritem aktivnih kontur lahko konvergira v eno končno obliko. Za to končno obliko zatem izračunamo razne parametre (npr. velikost, ploščino, obseg, kompaktnost itd.), na osnovi teh parametrov pa se dokončno odločimo, ali je dana oblika (seveda, če je v fazi aktivnih kontur konvergirala) jajčni mešiček.

5.

Avtomatsko razpoznavanje jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov

V 3. poglavju smo se podrobno seznanili z različnimi segmentacijskimi metodami, ki jih uporabljamo pri računalniški obdelavi digitalnih slik (tudi medicinskih). V podpoglavju 3.5 smo nekaj teh metod tudi primerjali med seboj, in sicer na osnovi umetno generiranih slik. Na osnovi postavljenih kriterijev smo izbrali najperspektivnejše. Že v 2. poglavju smo opisali določene lastnosti ultrazvočnih slik oz. ultrazvoka na splošno, v podpoglavju 2.2.1 pa smo videli, da je ultrazvok lahko zelo koristen pri opazovanju sprememb, ki se dogajajo v menstrualnem ciklusu. Te spremembe lahko zaznamo tudi na ultrazvočnih posnetkih. Vse to, in še potreba po računalniški obdelavi ultrazvočnih posnetkov jajčnikov [21], nas je spodbudilo, da smo se spoprijeli s to nalogo. Tako smo v prejšnjem poglavju (poglavje 4) s transformacijo HT skušali direktno razpoznati jajčne mešičke v ultrazvočnih posnetkih jajčnikov. Jajčne mešičke smo aproksimirali z elipsami, vendar se je takšna poenostavitev problema izkazala za neprimerno, saj je bil odstotek razpoznanih elips – mešičkov, zelo nizek (le okoli 30 %). V tem poglavju pa bomo naredili korak naprej. Razvili bomo namreč algoritem "xultra" (razpoznavalni sistem), s katerim bomo avtomatsko razpoznavali jajčne mešičke. Pri njegovem konstruiranju se bomo v fazi segmentacije izogibali vsakršnim predpostavkam o oblikah objektov (mešičkov). S tem bomo dosegli nekakšno univerzalnost algoritma, saj če zamenjamo medicinsko domeno (namesto razpoznavanja jajčnih mešičkov npr. razpoznavamo možganske prekate), moramo le v fazi klasifikacije spremeniti nekaj odločitvenih pravil, ne pa celotnega segmentacijskega algoritma.

Avtomatsko razpoznavanje jajčnih mešičkov pa ni samo sebi namen, ampak ima veliko uporabno vrednost. V [21] smo videli, da eksperti pri opazovanju dinamike rasti jajčnih mešičkov opravljajo celotno analizo ročno. Vse ultrazvočne preglede jajčnikov snemajo na video trak in jih zatem ročno analizirajo okvir za okvirom. Za posamezen mešiček so opazovali oz. merili velikost – merili so jo z mrežnim merilnim sistem, ki so ga polagali na video zaslon, obliko, ehogeničnost, kvaliteto antralnega roba in položaj jajčnika. Za kvalitetno analizo pa morajo skozi več dni menstrualnega ciklusa primerjati mešičke med seboj ter študirati njihov medsebojni vpliv (dominantne mešičke morajo primerjati z neuspešnimi dominantnimi in poddominantnimi). To rutinsko, ročno delo je seveda velika dodatna obremenitev za eksperte. Prav v tej razbremenitvi ekspertov vidimo uporabnost našega razpoznavalnega sistema.

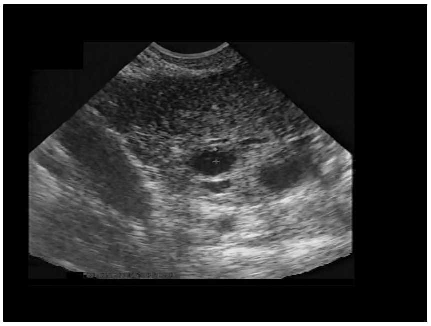
V tem poglavju bomo najprej podrobno opisali naš razpoznavalni sistem (podpoglavje 5.1), zatem bomo v podpoglavju 5.2 prikazali dobljene rezultate ter ocenili uspešnost razpoznavalnega sistema, nadaljevali s kratko razpravo (podpoglavje 5.3), poglavje pa zaključili s kratkim zaključkom.

5.1 Razpoznavalni sistem za avtomatsko razpoznavanje jajčnih mešičkov

Pogosta naloga, ki jo moramo opraviti, kadar imamo opravka z digitalnimi slikami, je razpoznavanje oz. iskanje določenih objektov v slikah. Tipična razpoznavalna shema, ki jo uporabljamo v teh primerih, ima dobro znano zgradbo [7, 31, 60, 63]. Najprej moramo sceno, ki jo želimo analizirati, zajeti. Slike lahko zajemamo na razne načine, npr. s prebiranjem fotografij, s slikovnim digitalizatorjem (angl. frame grabber), sceno lahko tvorimo umetno itd. V naslednjem koraku, v fazi predobdelave, lahko pridobljeno digitalno sliko obdelamo z raznimi filtri oz. operatorji, s katerimi odstranimo šum in izboljšamo kvaliteto slike (npr. Gaussov filter, nizko in visoko sito, popravljanje histograma slik). Sledi faza segmentacije, kjer skušamo razdeliti sliko na posamezna območja (poglavje 3). Vsako dobljeno regijo oz. območje zatem popišemo z značilkami (npr. z obsegom, težiščem, ploščino, polarnim histogramom itd.). V fazi klasifikacije pa zatem tako popisano regijo primerjamo z vzorčnim objektom – pralikom (objektom, ki ga iščemo) in jo sprejmemo kot iskani objekt ali pa jo zavrnemo.

Tudi naš razpoznavalni sistem smo zasnovali po tej tipični shemi in ga bomo po takšnem vrstnem redu opisali. Pri opisu algoritma (oz. razpoznavalnega sistema) si bomo pomagali s tipično ultrazvočno sliko jajčnika z jajčnimi mešički (slika 5.1). Podroben opis te slike najdemo v podpoglavju 4.3.

Originalna sektorska ultrazvočna slika je velikosti 576x768 pikslov in je v 256 sivinah (na sliki 5.1 je prikazana slika brez vseh oznak, sicer na sliki zasledimo še podatke o pacientki in o nekaterih parametrih ultrazvočne naprave). Naša naloga je, da iz takšnih slik razpoznamo jajčne mešičke. Opazimo lahko še, da so mešički precej temnejši od svoje okolice. To dejansko pomeni, da imamo podobne razmere, kot smo jih imeli pri simulaciji segmentacijskih metod v umetno tvorjenih testnih slikah (podpoglavje 3.5), ter da bodo za te ultrazvočne posnetke veljale vse, že zapisane lastnosti teh metod. V nadaljevanju bomo predstavili naš algoritem.



Slika 5.1: Tipičen ultrazvočni posnetek jajčnika.

5.1.1 Predobdelava

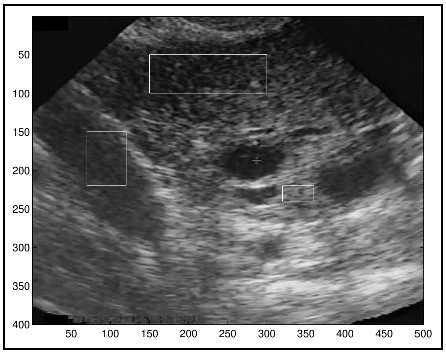
Glavni cilj predobdelave je, da ustrezno pripravi sliko za nadaljnjo analizo. Z našim sistemom smo v fazi predobdelave iz ultrazvočnih posnetkov odstranili pege, zatem pa še ocenili položaj jajčnika v takšni sliki.

Odstranjevanje šuma

Pri našem delu imamo opravka z ultrazvočnimi posnetki. Za to vrsto slik pa vemo, da se v njih pojavljajo neobstoječe majhne strukture, in sicer zaradi pegastega šuma ter reverberacije (podpoglavje 2.1.4). Ta splošno znan pojav ne moti ekspertov pri postavljanju diagnoz iz takšnih posnetkov, vendar pa je izredno moteč pri avtomatskem obdelovanju z računalnikom. Posebej velik problem predstavlja pegasti šum. V podpoglavju 2.3 smo zapisali nekaj napotkov, kako se lotiti predobdelave ultrazvočnih slik ter kaj je njen namen, glavni poudarek pa smo dali odstranjevanju peg (podpoglavje 2.3.1).

V sliki 5.1 lahko opazimo oba omenjena pojava, in sicer pege, ki se kažejo kot zelo majhne regije, kakor tudi pojav reverberacije, ki je vidna v obliki svetlih črt, ki se v polkrogu raztezajo od levega roba slike do desnega. Po svoji presoji smo se odločili, da je pegasti šum lahko veliko bolj moteč ter ga bomo zato v slikah odstranili. V podpoglavju 2.3.1 smo nanizali celo vrsto različnih operatorjev in algoritmov, s katerimi lahko odstranjujemo pege. Izmed teh smo jih nekaj izbrali, njihovo učinkovitost pa preizkusili na sliki 5.1.

V literaturi [12, 24, 71] smo zasledili mero S/M oz. indeks peg, s katero lahko merimo velikost pegastega šuma v homogenih regijah. Mero izračunamo tako, da delimo standardni odklon homogene regije s povprečjem znotraj te regije. V nefiltrirani sliki je ta mera velika, medtem ko se mora pri dobrem operatorju za odstranjevanje pegastega šuma ta vrednost zmanjšati. Za izračun mere S/M je treba izbrati homogene regije. Te regije izberemo iz originalne (šumne) slike, zato v resnici te regije niso homogene; dejansko izbiramo regije, ki bi bile homogene, če ne bi bilo dodanega pegastega šuma. Na sliki 5.2 so prikazane tri izbrane regije (trije beli pravokotniki), na katerih bomo preverjali uspešnost operatorjev, in sicer je po ena regija iz dela žile (levo), iz dela materniˇcne sluznice (zgoraj) in iz jajčnika (sredina).



Slika 5.2: Originalna ultrazvočna slika (slika 5.1), na kateri so z belo vrisane tri regije, ki jih bomo upoštevali pri izračunu mere S/M.

Iz slike 5.1 smo odstranjevali pegasti šum z različnimi operatorji, in sicer smo preizkusili nizko sito z jedrom velikosti 3x3 piksle ter Gaussov filter enakih dimenzij. Zraven obeh preprostih filtrov, smo odstranjevali pege tudi z adaptivnim filtrom HRGMF (podpoglavje 2.3.1). Za ta filter je treba določiti še velikost začetne regije in pragovno vrednost H_0, ki je parameter ultrazvočne naprave, zato ga je treba za vsako ultrazvočno napravo eksperimentalno izmeriti. Velikost začetne regije lahko določimo računsko iz parametrov ultrazvočne naprave (valovne dolžine valov, razdalje pretvornika do fokalne cone itd.). V [38] je po takšnem izračunu bila velikost začetne regije 7x7 pikslov, medtem ko je bila izmerjena homogenost H_0 enaka 20,0. Enake nastavitve smo prevzeli za naše filtriranje, saj smo ocenili, da se parametri naše naprave (vsaj tiste, ki jo poznamo), ne razlikujejo bistveno od parametrov v [38].

Kot zadnjo metodo pa smo preizkusili še transformacijo DWT (podpoglavje 2.3.1). Uporabili smo Daubechiesov valček stopnje 4 na treh različnih nivojih, in sicer na nivoju 2, 5 in 10.

Po odstranjevanju pegastega šuma s slike 5.1 smo za vse te operatorje izračunali tudi mero S/M. S tem smo lahko ocenili učinkovitost posameznih operatorjev. Dobljeni rezultati so prikazani v tabeli 5.1.

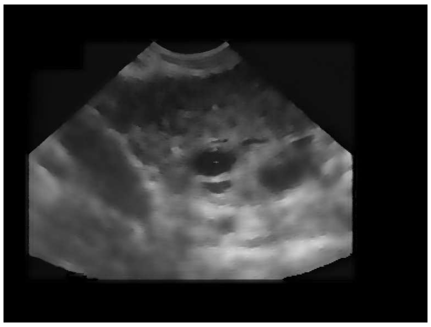
Po primerjavi med originalno in filtriranimi slikami (tabela 5.1) lahko sklepamo, da je filter HRGMF najbolje odstranil prisoten šum, saj je izračunana mera S/M zanj povsod najmanjša. Kot druga najboljša se je izkazala transformacija DWT (Daubechiesov valček stopnje 4 in nivo 10 oz. 5), pri čemer je treba poudariti, da smo takšen rezultat dobili direktno po uporabi rutin iz paketa Matlab, kar pomeni, da nismo ničesar optimirali. Sledita še nizko sito in Gaussov operator, ki sta vrnila podoben rezultat. Opazimo tudi lahko, da je filtriranje z DWT, kjer je globina dekompozicije majhna, primerljivo z nizkim sitom (glej tabelo 5.1, vrstico “Daubechies (nivo 2)”).

	Žila	Sluznica	Jajčnik
Originalna slika	0,2338	0,4608	0,1808
Nizko sito 3x3	0,2176	0,4061	0,1441
Gaussov filter 3x3	0,2201	0,4164	0,1513
HRGMF	0,1702	0,2563	0,0508
Daubechies (nivo 2)	0,2212	0,4204	0,1581
Daubechies (nivo 5)	0,2062	0,3678	0,1452
Daubechies (nivo 10)	0,2032	0,3682	0,1446

Tabela 5.1: Izračunana mera S/M za izbrane operatorje za odstranjevanje pegastega šuma v slikah. Pri izračunu smo upoštevali homogene regije s slike 5.2.

Filter HRGMF vrača res najboljše rezultate, vendar je pa časovno izredno kompleksen, saj za filtriranje slike dimenzij 576x768 pikslov potrebuje okrog 4 minute na delovni postaji HP 715/100XC, medtem ko odstranjevanje šuma z DWT, ki je druga najuspešnejša metoda, traja 10-krat manj na istem sistemu in v okolju Matlab 4.2c. Če primerjamo "stroške-učinkovitost" obeh metod, se vendarle izkaže operator HRGMF kot superioren.

Iz vsega povedanega lahko na koncu strnemo: za odstranjevanje šuma iz ultrazvočnih posnetkov jajčnikov smo izbrali filter HRGMF. Na sliki 5.3 je prikazan rezultat po filtriranju slike 5.1 s tem filtrom.



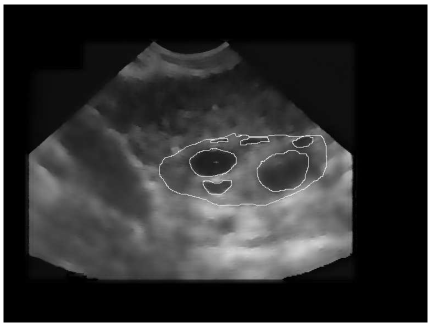
Slika 5.3: S filtrom HRGMF filtrirana slika 5.1. Začetna regija je velikosti 7x7 pikslov, homogenostni prag je 20,0.

Določanje približnega položaja jajčnika

Po izločitvi peg iz ultrazvočnega posnetka (slika 5.3) je slika že pripravljena za segmentacijo. Na takšni sliki bi lahko že uporabili eno izmed segmentacijskih metod za določanje položaja mešičkov, a ker so posnetki velikosti 576x768 pikslov, bi bilo to časovno prezahtevno. V simulaciji segmentacijskih metod, ki smo jo opisali v podpoglavju 3.5, smo segmentirali umetno tvorjene slike velikosti 256x256 pikslov in pokazalo se je, da posamezne metode segmentirajo zelo šumne slike tudi po nekaj minut (npr. algoritem MOMENTI tudi 10 in več minut).

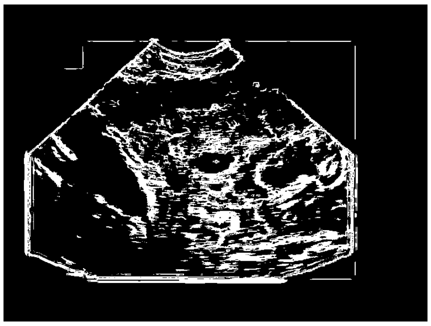
To je bil glavni razlog, da smo želeli zreducirati problem iskanja jajčnih mešičkov v celotni sliki (576x768 pikslov), na problem iskanja mešičkov znotraj določene podslike, ki je manjših dimenzij kot originalna slika. To našo željo podpira tudi dejstvo, da se vsi jajčni mešički vedno nahajajo znotraj jajčnika. Torej, s tem ko bi določili položaj jajčnika, bi takoj imeli določeno tudi podsliko, znotraj katere bi morali iskati mešičke. Vidimo lahko, da se je problem iskanja mešičkov transformiral v problem določanja položaja jajčnika. Vendar pa to ni več tako zahtevna naloga (v smislu natančnosti), ker je dovolj, če le približno ocenimo položaj jajčnika. Res je, da čim natančneje bomo določili jajčnik, tem manj bo na koncu napačno razpoznanih mešičkov (v slikah je tudi zunaj jajčnika veliko struktur, ki so zelo podobne mešičkom), vendar je pa to že bolj problem klasifikacije.

Položaj jajčnika lahko gotovo ocenimo na mnogo različnih načinov. Po posvetovanju z ekspertom (ginekologinja), ki je trdil, da lahko opazi oz. da obstaja neka meja (rob) med jajčnikom in ostalim delom slike, smo se odločili, da bomo poskušali ta položaj določiti s pomočjo slike robov. V sliki 5.1 vidimo rob jajčnika zelo nejasno, medtem ko lahko na sliki 5.4 opazujemo točen položaj jajčnika, kakor ga je določil ekspert.



Slika 5.4: Ekspert je obkrožil jajčnik in 6 jajčnih mešičkov v sliki 5.1.

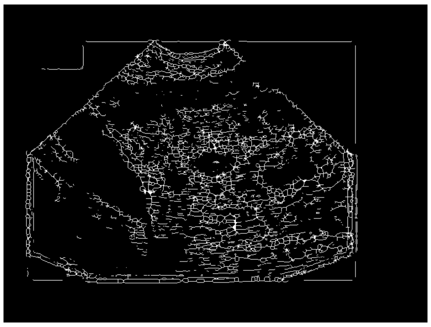
Iz literature [7, 10, 13, 31, 60, 63] poznamo celo vrsto operatorjev za odkrivanje robov: Robertsov, Laplaceov, Prewittov, Sobelov, Kirschev, Cannyjev operator, operator Laplace iz Gaussa itd. Preizkusili smo nekaj izmed teh filtrov. Odločili smo se za Kirschev operator, ki je nekakšna različica Sobelovega operatorja. Kot osnovno masko vzame enako masko kot Sobelov operator, ki jo zatem uporabi še v osmih rotiranih položajih [60]. Kirschev operator za odkrivanje robov smo izbrali predvsem na osnovi njegove majhne časovne zahtevnosti, saj so bili dobljeni robovi pri vseh operatorjih zelo podobni. Po uporabi kakšnega operatorja za detektiranje robov, tudi Kirschevega, imamo še vedno sliko sivin in če želimo dobiti sliko robov, moramo ta rezultat še binarizirati [63]. Za binarizacijo smo uporabili pragovno operacijo OPTIMAL (podpoglavje 3.1.1). Na sliki 5.5 je prikazana tipična slika robov, ki jo iz ultrazvočnih posnetkov jajčnika dobimo po tem koraku.



Slika 5.5: Slika robov. Sliko 5.3 smo filtrirali s Kirschevim operatorjem, zatem pa jo binarizirali (prag 20).

Iz slike 5.5 lahko tudi opazimo, da so robovi zelo debeli oz. da so široki več kot piksel. To je lahko zelo moteče pri sledenju notranje meje regije, saj se večina takšnih algoritmov v tem primeru zazanka (algoritem 4). To predstavlja problem tudi za nas, saj dejansko želimo iz slike robov določiti položaj jajčnika. Naša ideja za določitev položaja jajčnika je namreč, da na neki način odkrijemo točko iz meje jajčnika ter zatem iz te točke sledimo meji.

Debele robove v binarnih slikah lahko stanjšamo kar na nekaj načinov, najpreprosteje pa to nalogo opravimo z morfološkim tanjšanjem [17, 63]. To je iterativni operator, ki ima vnaprej predpisano masko velikosti 3x3 piksle, uporabljamo pa ga na binarni sliki, dokler ne dosežemo konvergence (rezultata dveh zaporednih faz tanjšanja sta enaka). Na sliki 5.6 je takšna slika robov, kjer so robovi stanjšani z morfološkim tanjšanjem. Iz te in tudi iz drugih slik, na katerih smo preverjali naš razpoznavalni sistem, lahko vidimo, da so robovi sicer na pravih mestih, vendar so na določenih delih prekinjeni oz. v robovih so velike luknje. Te moteče luknje želimo čim bolje zapolniti.



Slika 5.6: Slika robov (slika 5.5) po tanjšanju robov.

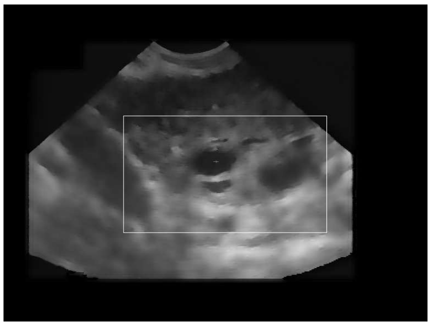
Polnjenje lukenj v robovih je zelo kompleksna naloga, saj moramo na določenih manjkajočih delih umetno ustvariti rob, ki kar najbolje ustreza dejanski meji objekta. Že iz te zahteve lahko sklepamo, da bo kvalitetna rutina za polnjenje robov morala upoštevati trenutni potek roba, položaj ostalih robov v njegovi okolici, hitrost in smer sprememb sivin v področju, kjer potekajo robovi itd. Kljub upoštevanju vseh teh kriterijev bo po polnjenju vedno tudi veliko napačno zapolnjenih robov, predvsem v slikah z veliko objekti. Ker je v našem razpoznavalnem sistemu polnjenje robov le eden izmed vmesnih korakov obdelave, smo razvili čisto preprost hevristični algoritem za to nalogo. Polnjenje poteka po naslednjem postopku. Vsakemu robu poiščemo začetno in končno točko. Če je rob sklenjen, ga ni treba polniti, sicer pa ga polnimo na obeh koncih - pri začetni in končni točki. Za obe ti krajni točki določimo smer polnjenja. Za smer lahko izberemo smer tangente na rob v tej točki (v našem primeru) ali pa smer daljice med krajno točko in točko iz istega roba, ki je od krajne točke oddaljena za določen radij. Ko imamo smer določeno, potem v tej smeri iščemo prvo točko z drugega konca roba. Ko jo najdemo, ju povežemo z daljico in rob je dopolnjen. Izkazalo pa se je, da v tej določeni smeri le redko naletimo na točko z drugega konca roba v smiselni oddaljenosti. Zato smo ta del rutine nekoliko dopolnili. Namesto da bi iskali robno točko le v tej izračunani smeri (α), smo iskanje točke razširili na iskanje v določenemu pasu okoli te smeri ($\alpha \pm \triangle\alpha$). Iskanje smo omejili tudi na določeno okolico krajne točke oz. na določen radij (r). Na sliki 5.7 je prikazana slika 5.6, kateri smo z našo hevristično rutino zapolnili robove, pri tem smo parameter $\triangle\alpha$ postavili na 5^{o}, radij r pa na 15 pikslov.

Pri tej rutini vidimo še nekaj izboljšav, ki bi jih lahko (vendar jih nismo) vključili ter tako dobili še bolj smiselno zapolnjene robove. Tako npr. bi lahko, ko naletimo na drugo robno točko, preverili, ali se smer iskanja in smer tangente v tej točki približno ujemata, ter šele po njunem ujemanju povezali obe točki.

Slika 5.7: Slika stanjšanih robov (slika 5.6) po polnjenju robov.

Ko smo zapolnili robove, smo imeli sliko pripravljeno za določanje položaja jajčnika. Najprej smo testirali, kako se obnese algoritem za sledenje notranji meji (algoritem 4). Velja namreč, da če imamo določeno mejo regije, imamo z njo popolnoma določen tudi sam objekt – jajčnik (podpoglavje 3.2). Zato smo sami vnašali začetne točke, ki so bile na meji (robu) jajčnika, iz katerih smo zatem sledili notranji meji po omenjenem algoritmu. Rezultati, ki smo jih dobili s takšnim sledenjem, so bili večinoma napačni. Tako se je v posameznih primerih jajčnik prehitro zaključil, ali pa se razvlekel skoraj skozi vso sliko. Vzrok temu pripisujemo, da je v sliki robov (glej sliko 5.7 kot primer) preveč majhnih nepomembnih robov, ki povzročajo, da algoritem pri sledenju zaide. Iz celotnega testiranja smo zaključili, da tudi spreminjanje oz. dopolnjevanje algoritma 4, ne more prinesti želenega rezultata. Zato smo morali poiskati alternativno rešitev, kako iz slike robov določiti oz. vsaj oceniti položaj jajčnika.

Pri opazovanju slik robov naših ultrazvočnih posnetkov smo opazili, da je večina belih točk v področju jajčnika ali vsaj v njegovi bližnji okolici (primerjaj sliki 5.7 in 5.4), in na osnovi tega smo prišli do ideje, da bi položaj jajčnika lahko ocenili iz gostote robnih pikslov. Vidimo lahko, da je znotraj jajčnika gostota robnih pikslov največja in to bo tudi naš kriterij. Kot sredstvo, s katerim lahko opravimo to merjenje, se nam sama ponuja uporaba histograma, ki je primeren za štetje določenih elementov (podpoglavje 3.1.1). Robne piksle smo prešteli v dveh različnih smereh, in sicer po vrsticah in po stolpcih. Tako smo dobili histogram robnih pikslov v smeri x (po vrsticah), kjer je v vsakem elementu tega histograma shranjeno število robnih pikslov za določeno vrstico, ter histogram robnih pikslov v smeri y (po stolpcih). V obeh histogramih smo zatem poiskali maksimum. Oba maksimuma, x_{max} v smeri x in y_{max} v smeri y, nam določata točko (x_{max}, y_{max}), ki leži znotraj jajčnika. Ta trditev je veljala za naše posnetke. Iz te točke moramo zatem v štirih smereh iskati robe jajčnika (v smeri x – levo in desno, ter v smeri y – gor in dol). S tem dejansko dobimo ocenjen pravokotnik, znotraj katerega se nahaja jajčnik. To iskanje smo izvedli po obeh izračunanih histogramih, in sicer smo začeli pri maksimumu, od koder smo zatem iskali lokalne minimume. Iskanje v posamezni smeri smo zaključili, ko je bila razlika med dvema zaporednima lokalnima minimumoma večja od predpisanega praga. Ko smo poiskali vse štiri mejne točke x_{zg}, x_{sp}, y_l in y_d, sta nam točki (x_{zg}, y_l) in (x_{sp}, y_d) popolnoma določili pravokotnik, prva točka zgornje levo oglišče, druga pa spodnje desno oglišče pravokotnika. Na sliki 5.8 je za sliko 5.1 vrisan položaj jajčnika, kot je bil določen z zgoraj opisano proceduro.



Slika 5.8: Slika 5.3 z belo vrisano ocenjeno lego jajčnika.

5.1.2 Segmentacija

V fazi predobdelave smo z našim razpoznavalnim sistemom pripravili ultrazvočne posnetke za nadaljnjo analizo. Iz posnetkov smo odstranili pegasti šum ter približno določili položaj jajčnika. V tem koraku pa moramo določiti jajčne mešičke.

Če pogledamo sliki 5.3 in 5.4 kot primer, vidimo, da so jajčni mešički temne, v posameznih primerih skoraj črne, jajčaste strukture na svetlejšem ozadju – jajčniku. V našem primeru gre torej za iskanje temnih objektov na svetlem ozadju, to pa je dejansko enak problem, kot smo ga že imeli v testnih slikah pri simulaciji segmentacijskih metod (podpoglavje 3.5.1). Iz tega sklepamo, da lahko pri našem razpoznavalnem sistemu uporabimo vse iz simulacije pridobljeno znanje o segmentacijskih metodah.

V simulaciji se je kot najboljša segmentacijska metoda izkazala pragovna segmentacijska metoda OPTIMAL, ki avtomatsko določa prag (podpoglavje 3.5.3). Ker veljajo enaki pogoji tudi pri naših ultrazvočnih posnetkih, smo kot segmentacijsko metodo tudi v tem primeru izbrali metodo OPTIMAL.

Tudi sam potek segmentacije smo v našem razpoznavalnem sistemu zasnovali na enak način kot pri simulaciji (podpoglavje 3.5.2). Torej, začetno sliko, v našem primeru le podsliko z jajčnikom, najprej obdelamo s pragovno operacijo OPTIMAL ter tako dobimo binarno sliko. V tej binarni sliki pa zatem vse črne regije identificiramo (označimo) z algoritmom za iskanje temnih pik. Vsako tako dobljeno regijo nato še popišemo s parametri oz. značilkami. Posamezni regiji smo izračunali prve tri momente ter centralne momente stopnje 2 [31, 60, 63], iz njih pa lahko enostavno določimo težišče in ploščino regije. Poleg teh osnovnih parametrov smo določili še mejo in obseg regije. Iz osnovnih parametrov smo izračunali izpeljane značilke posamezne regije, in sicer kompaktnost, ki se izračuna kot razmerje med kvadriranim obsegom in ploščino regije, ekscentričnost in razmerje med ploščino ter obdajajočim pravokotnikom.

V literaturi [31, 60, 63] lahko najdemo še mnogo drugih značilk, s katerimi opisujemo regije, njihov izračun pa lahko zelo preprosto vključimo v naš razpoznavalni sistem.

5.1.3 Klasifikacija

V fazi klasifikacije se moramo o vsaki, z značilkami popisani regiji odločiti, ali je jajčni mešiček ali ne. Šele s tem korakom dejansko postane jajčni mešiček razpoznan.

Klasifikacija oz. splošneje razpoznavanje vzorcev je izredno zahtevna naloga [54]. Za dan problem moramo najprej analizirati vse značilke posameznih objektov ter iz te množice po nekem pravilu določiti optimalne. Zatem moramo preveriti, ali se vzorci porazdeljujejo v rojih v parametričnem prostoru. Šele ko lahko pozitivno odgovorimo na to vprašanje, lahko razpoznavamo vzorce, npr. z razvrščanjem vzorcev, z odločitvenimi funkcijami in pravili, nevronskimi mrežami itd.

Tudi v naš razpoznavalni sistem smo vključili razvrščevalnik, vendar pa zanj nismo opravili kompletne, zgoraj opisane analize problema (porazdeljevanje v roje, izbira značilk ipd.). Po posvetovanju z ekspertom smo izbrali ustrezne značilke oz. parametre regij. Ker so mešički omejenih dimenzij, najmanjši mešiček, ki ga še lahko razpoznamo, meri v premeru 2 mm, smo kot prvi parameter izbrali ploščino regije. Mešički so jajčaste oblike, kar smo upoštevali kot drugi kriterij pri razvrščanju. Zato smo izbrali še parameter kompaktnost in parameter razmerje med ploščino regije ter obdajajočim pravokotnikom.

Na osnovi teh izbranih parametrov smo nato klasificirali posamezne regije. Regije smo razpoznavali z odločanjem oz. z odločitvenimi pravili. Tridimenzionalni parametričen prostor smo omejili z delnimi odločitvenimi pravili. To pomeni, da smo vsako od teh treh osi razdelili na dve področji: področje, kjer vrednosti ustrezajo možnim vrednostim mešičkov, in na področje, kjer te vrednosti ne ustrezajo. Če parametri posamezne regije padejo v presek teh ustreznih področij, potem lahko rečemo, da je ta regija najverjetneje jajčni mešiček, sicer pa jo zavržemo. Potrebne prage smo določili iz pridobljenega znanja o problemu. Parameter ploščina mora biti večja ali enaka 220 pikslov, kompaktnost manjša ali enaka 40 in razmerje med ploščino ter obdajajočim pravokotnikom večje ali enako 0,5.

Na sliki 5.9 so prikazani razpoznani jajčni mešički, ki smo jih dobili z našim razpoznavalnim sistemom.

Slika 5.9: Razpoznani jajčni mešički za sliko 5.1.

Na koncu še enkrat vse strnimo in podajmo naš algoritem oz. razpoznavalni sistem v kompaktni obliki psevdokoda:

1. *Odstrani pegasti šum s filtrom HRGMF.*
2. *Odkrij robove s Kirschevim operatorjem (velikost 3x3 piksle).*
3. *Binariziraj sliko z operacijo OPTIMAL.*
4. *Tanjšaj robove z morfološkim tanjšanjem (velikost 3x3 piksle).*
5. *Polni luknje v robovih.*
6. *Oceni položaj jajčnika:*
   - *Tvori histogram robnih pikslov po vrsticah (histogram x) in po stolpcih (histogram y).*

- *Poišči maksimuma za oba histograma.*
- *V histogramu x poišči ustrezna lokalna minimuma: levo (x_{zg}) in desno od maksimuma (x_{sp}).*
- *V histogramu y poišči ustrezna lokalna minimuma: levo (y_l) in desno od maksimuma (y_d).*
- *Točki (x_{zg}, y_l), zgornje levo oglišče, in (x_{sp}, y_d), spodnje desno oglišče, določata pravokotnik, ki vsebuje jajčnik.*

7. *Segmentiraj jajčnik z operacijo OPTIMAL.*
8. *Vsako temno regijo identificiraj z algoritmom iskanja temnih pik.*
9. *Vsako regijo popiši z značilkami (momenti, ploščina, obseg, kompaktnost itd.).*
10. *Razpoznaj regije z delnim odločanjem na osnovi ploščine, kompaktnosti in razmerja med ploščino regije ter obdajajočega pravokotnika.*

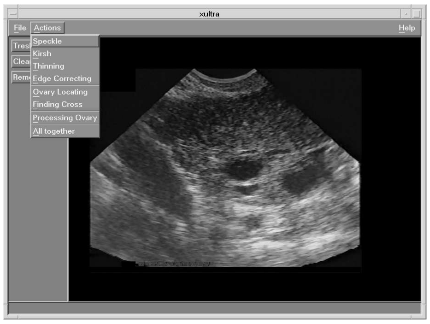
Algoritem 9: Algoritem za avtomatsko razpoznavanje jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov

5.2 Rezultati

V podpoglavju 5.1 smo podrobno opisali razpoznavalni sistem za avtomatsko določanje jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov. Podali smo tudi psevdokod tega sistema. V tem podpoglavju pa bomo ocenili njegovo učinkovitost.

Opisani razpoznavalni sistem smo implementirali in preizkusili na množici realnih ultrazvočnih posnetkov. Sistem smo implementirali v programskem jeziku C z uporabo knjižnic Motif 1.2 in je namenjen za okolje X-Windows. Program smo poimenovali "xultra" in je tipično menujsko oblikovana aplikacija, kjer lahko opazujemo tudi vmesne rezultate, dobljene po posameznih korakih obdelave (npr. sliko po tanjšanju robov). Na sliki 5.10 je prikazano glavno okno aplikacije "xultra".

Razpoznavalni sistem smo preizkusili na 30 različnih ultrazvočnih posnetkih jajčnikov. Slike smo dobili na Oddelku za človeško reprodukcijo in endokrinologijo, Splošne bolnice Maribor. Vsak pregled pacientke, ki je bil opravljen s 5 MHz intravaginalno sondo, so posneli na VHS video trak. Ta video trak smo zatem s programskim digitalizatorjem digitalizirali na grafični postaji Silicon INDIGO 2 ter tako dobili posamezne slike v grafičnem formatu Iris RGB. Takšno barvno sliko smo zatem le še pretvorili v grafični format BMP in dobili originalno ultrazvočno sliko dimenzije 576x768 pikslov v 256 sivinah (slika 5.1). Testne slike smo dobili od 6 pacientk, ki so bile različnih starosti in brez znanih ginekoloških problemov, pri eni od teh pacientk smo imeli tudi slike iz različnih dnevov njenega menstrualnega ciklusa. Del testnih slik je bil posnet na eni ultrazvočni



Slika 5.10: Glavno okno aplikacije "xultra".

napravi, del slik pa na drugi, pri čemer sta bili napravi različni, s tem pa sta imeli tudi različna formata zajetih slik ter odtenke sivin slik (primerjaj posnetke na sliki 5.11).

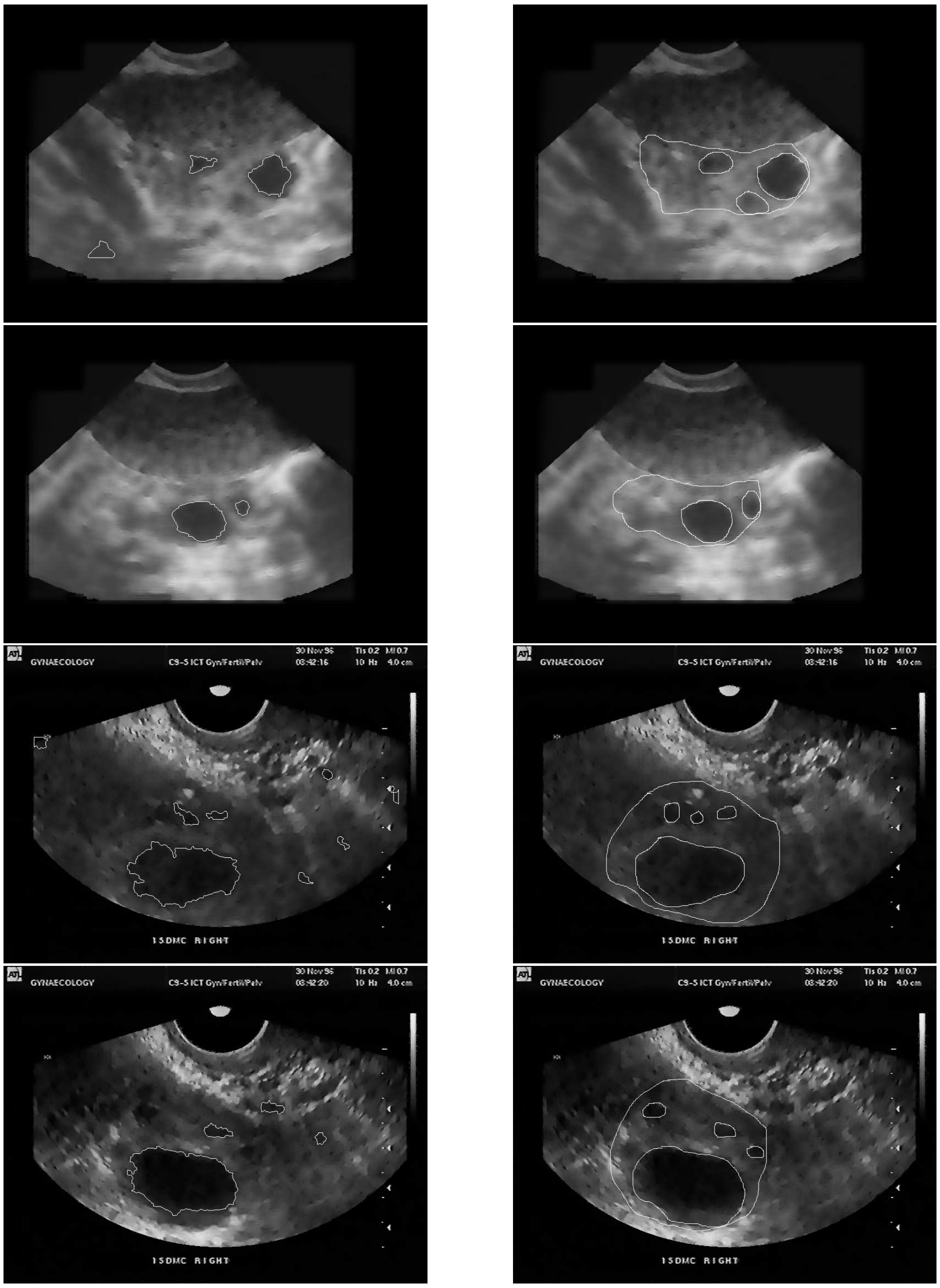
Program "xultra" smo poganjali na delovni postaji HP 715/100XC s 128 MB delovnega pomnilnika. Časovna kompleksnost razvitega algoritma je $\mathcal{O}(n^2)$, kjer je n dimenzija slike. Celotno razpoznavanje posameznega ultrazvočnega posnetka (dimenzij 576x768 pikslov) je na omenjenem sistemu trajalo približno 6 minut. V tabeli 5.2 so zbrani povprečni porabljeni časi obdelovanja za posamezno fazo razpoznavalnega sistema. Vidimo lahko, da filter za odstranjevanje peg prinese 73 % k skupnemu času obdelovanja, določanje položaja jajčnika nadaljnih 25 % (v to fazo spadajo rutine od Kirschevega operatorja naprej), medtem ko segmentacija in klasifikacija porabita skupaj le 2 % vsega časa.

Faza	Povprečen porabljen čas (s)
Odstranjevanje peg	266
Določanje položaja jajčnika	90
Segmentacija	4
Klasifikacija	6
Skupaj	366

Tabela 5.2: Povprečen čas razpoznavanja jajčnih mešičkov za eno sliko po posameznih fazah.

Na sliki 5.11 je prikazanih še nekaj tipičnih rezultatov. Leva slika iz posameznega para slik prikazuje razpoznane jajčne mešičke z našim algoritmom, na desni sliki pa je prikazan natančen položaj jajčnika in mešičkov, kot jih je določil ekspert.

Za novi razpoznavalni sistem smo želeli preveriti tudi njegovo učinkovitost oz. pravilnost razpoznavanja. Če pa želimo podati izjavo o njegovi uspešnosti, potrebujemo ustrezno mero za merjenje učinkovitosti. Že pri simulaciji segmentacijski metod (podpoglavje 3.5.2) smo uvedli ustrezni meri, in sicer odstotek razpoznanih objektov (elips) ter odstotek



Slika 5.11: Nekaj tipičnih rezultatov, dobljenih s programom "xultra". Leva slika vsakega para prikazuje mešičke, kot jih je razpoznal naš sistem, na desni sliki pa je ekspert označil pravilni položaj jajčnika in mešičkov.

razpoznanih regij. Prva mera, odstotek razpoznanih objektov, je pomenila razmerje med številom razpoznanih objektov (elips) ter številom vseh objektov v sliki, ki je vnaprej znano. Druga mera, odstotek razpoznanih regij, pa je pomenila razmerje med številom pravilno razpoznanih regij ter številom vseh regij, postavljenih v sliko s segmentacijskim postopkom. Smiselno je, da tudi pri ocenjevanju našega razpoznavalnega sistema uporabimo enaki dve meri s to razliko, da jima ustrezno spremenimo imeni (namesto razpoznanih objektov bo sedaj razpoznanih mešičkov).

V prvi vrsti nas zanima, kako uspešna je izbrana segmentacijska metoda pri tem realnem problemu (v našem primeru je to metoda OPTIMAL). Pri izračunu smo upoštevali dobljene regije direktno po segmentaciji, regije pa so bile razpoznane kot mešički na osnovi kriterijev o presekih, torej na osnovi enakih kriterijev kot pri simulaciji (podpoglavje 3.5.2). Izračunan odstotek razpoznanih mešičkov za naše ultrazvočne posnetke je bil pri kriteriju 50 enak 70 % (regija mora pokrivati vsaj 50 % mešička), odstotek razpoznanih regij pa je bil pri istem kriteriju 12,2 %. Po primerjavi dobljenih rezultatov z rezultati iz simulacije, tudi v simulaciji smo se lotili razpoznavanja mešičkov, vendar na preprost način (podpoglavje 3.5.4), vidimo, da se je odstotek razpoznanih mešičkov, po dodanem koraku predobdelave izboljšal za 18,7 % glede na rezultat, dobljen pri segmentacijski metodi z najvišjim odstotkom razpoznavanja v simulaciji, medtem ko se je odstotek razpoznanih regij povečal za 6 %.

Pri realnih problemih pa točni podatki o objektih niso vnaprej znani, ampak se moramo o njihovem razvrščanju odločiti na osnovi nekih parametrov. Tudi v naš razpoznavalni sistem smo vključili fazo klasifikacije, kjer se o posamezni regiji odločamo, ali je mešiček ali ne, pri tem pa ne upoštevamo natančnih položajev mešičkov, saj predvidevamo, da ti podatki niso vnaprej znani. Po fazi klasifikacije tako dobimo množico razpoznanih regij, ki so verjetno jajčni mešički. Te rezultate smo lahko zatem primerjali s pravilnimi rezultati, ki nam jih je posredoval ekspert, ter tako izračunali učinkovitost razpoznavalnega sistema. Za naš sistem je odstotek razpoznanih mešičkov enak 61 %, odstotek razpoznanih regij pa 44 %. Ta statistika zajema v izračun tako dominantne jajčne mešičke, ki so ponavadi veliki, kakor tudi poddominantne mešičke, ki so majhni. Če pa pri izračunu upoštevamo le dominantne mešičke, je odstotek razpoznanih mešičkov višji in je enak 86,8 %.

5.3 Razprava

V prejšnjem podpoglavju smo podrobno opisali dobljene rezultate za naš razpoznavalni sistem. V nadaljevanju pa bomo naš algoritem kritično analizirali, izpostavili določene probleme ter ponudili možne rešitve.

Najprej poglejmo časovno kompleksnost oz. porabljen čas za obdelovanje enega ultrazvočnega posnetka. Iz tabele 5.2 vidimo, da potrebujemo okrog 6 minut za eno sliko. To gotovo pomeni, da sistem takšen, kot je, ni primeren za reševanje realnočasovnih problemov. Časovno najbolj požrešen del sistema je filter za odstranjevanje peg HRGMF, ki porabi približno 4 minute. Ena izmed alternativ, ki se nam ponuja, je, da ga zamenjamo s kakšnim časovno manj zahtevnim filtrom, vendar pa moramo hkrati s tem pričakovati, da bo kakovost filtriranja nižja oz. bo več neodstranjenih peg. Veliko bolj smiselno je, da samo obdelovanje porazdelimo na več procesorskih enot, ki bi delovale istočasno. Faza predobdelave, predvsem filter HRGMF in Kirschev operator, je še posebej primerna za paralelno obdelovanje, saj se vsak piksel filtrira na popolnoma enak način, hkrati pa vsi potrebujejo iste podatke. Če bi npr. imeli 10 procesorskih enot in če predpostavimo linearno pospešitev (čas obdelovanja se linearno zmanjša s številom procesorskih enot) ter če bi paralelizirali le oba omenjena filtra, bi čas obdelovanja zmanjšali na 118 s oz. za 68 %.

Odstotek razpoznanih mešičkov se je povečal glede na rezultate v simulaciji (podpoglavje 3.5.4), kljub temu pa smo pričakovali boljše rezultate. Glavni razlog za nižje rezultate od pričakovanih vidimo v pregrobo ocenjenem jajčniku. Jajčnik smo po ekspertovem priporočilu ocenjevali na osnovi robov, a ker meje v ultrazvočnih posnetkih niso izrazite, je hkrati s tem tudi manj robnih pikslov na meji jajčnika. Zato postanejo lokalni minimumi v histogramih x in y neizraziti, kar vpliva na oceno jajčnika. Poudariti moramo, da so z našo rutino ocenjeni jajčniki za vseh 30 slik vedno vsebovali tudi pravi jajčnik. Pregrobo ocenjeni jajčniki imajo velik vpliv tudi na statistiko za odstotek razpoznanih regij, kar lahko opazimo v izračunanih nizkih odstotkih pri rezultatih, dobljenih takoj po segmentaciji, kakor pri rezultatih po klasifikaciji (glej podpoglavje 5.2). Odstotek razpoznanih regij je neposredno povezan z odstotkom napačno razpoznanih mešičkov, ki ga dobimo tako, da od 100 % odštejemo odstotek razpoznanih regij. V primeru našega razpoznavalnega sistema je ta odstotek 56 %.

Naslednji problem, ki smo ga zasledili pri razpoznavanju, so majhni jajčni mešički. Ti mešički so res temnejši od lokalne okolice, a ker so tako majhni, ne vplivajo bistveno na segmentacijsko metodo (v našem primeru metodo OPTIMAL). To pomeni, da se lahko po segmentaciji več majhnih mešičkov zlije z ozadjem ali pa med seboj. Ta naša trditev se kaže v odstotku razpoznanih mešičkov, ki je bistveno višja pri razpoznavanju le velikih oz. dominantnih jajčnih mešičkov. V [11], kjer je bila opravljena popolna analiza verjetnostnih porazdelitev v ultrazvočnih posnetkih jajčnikov, so kot sklep zapisali, da bi se z zamenjavo segmentacijske metode, predlagana je bila metoda rasti regije, hkrati izboljšal tudi odstotek razpoznavanja. S to trditvijo se ne strinjamo popolnoma, saj smo prepričani, da s tem problema majhnih mešičkov nismo rešili. Problem majhnih mešičkov oz. njihovega zlivanja lahko rešimo, če že v samo segmentacijsko metodo vgradimo mehanizem, ki skrbi za nadzor, da ne pride do zlivanja med objekti, torej neke vrste algoritem razvodja (angl. watershed) [7], v [39] pa smo tudi našli algoritem za razgradnjo staknjenih dvodimenzionalnih objektov, ki bi ga tudi lahko uporabili v takšnih primerih.

Razpoznavalni sistem "xultra" se trenutno še ne uporablja v praksi kot računalniško orodje, ki bi aktivno pomagalo ekspertom pri njihovem delu. Če želimo, da naš oz. kakršen koli podoben sistem zaživi, moramo paziti še na nekaj stvari. Najprej moramo podrobno preučiti, kako naj opravimo klasifikacijo. V našem sistemu odstotek razpoznanih mešičkov, izračunan po klasifikaciji, še ni enako visok kot isti odstotek, izračunan direktno po segmentaciji (podpoglavje 5.2), kar pomeni, da z izboljšano klasifikacijo lahko dosežemo razpoznavanje mešičkov v višini 70 %. Problem, ki ga moramo še rešiti pri klasifikaciji, so vnaprej postavljeni pragi v odločitvenih pravilih (v našem primeru je to npr. mejna vrednost za ploščino mešička). Zelo natančno je treba v teh primerih doreči, kako se naj obravnavajo regije, ki imajo vrednosti značilk zelo blizu mejnim vrednostim. Ali naj takšne regije sprejmemo ali jih zavržemo? Vsekakor je treba končno odločitev prepustiti ekspertom. Naš sistem za razpoznavanje deluje popolnoma avtomatsko z izjemo določanja nekaj pragov, ki pa jih lahko kalibriramo s prvim poganjanjem na ultrazvočnih posnetkih, dobljenih z neznane ultrazvočne naprave. Avtomatska obdelava je v posameznih primerih zelo zaželena, zahteva npr. izredno majhno interakcijo eksperta in s tem njegovega časa, lahko pa je tudi nezaželena. Če sistem pri obdelovanju naredi določeno napako, se ta napaka prenaša in verjetno še povečuje skozi vse naslednje faze sistema. Posebno zahtevna je avtomatska obdelava pri zelo kompleksnih slikah, ki so ponavadi še močno šumne. Ta problem smo zapazili v [34], kjer so se ravno iz prej omenjenega razloga raje lotili polavtomatske razgradnje mišičnih vlaken in ne avtomatske. V [48] so se lotili segmentiranja ultrazvočnih posnetkov srca s segmentacijsko metodo, v katero je bilo vključeno dodatno znanje ekspertov. Poročali so o 90 % uspehu pri segmentiranju levega srčnega prekata. Tudi v [40] smo zasledili, da se je s tem, ko so pri segmentaciji možganskih prekatov v ultrazvočnih slikah lobanje dopustili interakcijo ekspertov, odstotek razpoznavanja glede na popolnoma avtomatski algoritem bistveno izboljšal.

Tudi pri našem sistemu vidimo, da bi bilo zelo smiselno, če bi rezultate po sklopu posameznih faz verificiral ekspert, ki bi se tudi odločil, ali naj razpoznavalni sistem nadaljuje z razpoznavanjem ali ne. Ta princip bi lahko uvedli pri določanju položaja jajčnika, kjer bi ekspert lahko izračunani položaj popravil oz. nas na kakšen podoben način informiral o njegovem pravilnem položaju (ekspert lahko npr. na ultrazvočni posnetek preprosto doda posebne oznake za jajčnik), ter pri staknjenih mešičkih.

5.4 Zaključek

V tem poglavju smo se podrobno seznanili z algoritmom za avtomatsko razpoznavanje jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov. Videli smo, da je razvijanje takšnega medicinskega razpoznavalnega sistema zelo kompleksna naloga, kjer je ves čas potrebna usklajenost med strokovnjaki iz medicine ter strokovnjaki s področja obdelave slik. Vsi razpoznavalni sistemi imajo podobno zgradbo, in sicer je na začetku predobdelava, zatem sledi segmentacija ter nato še klasifikacija. V naš sistem smo po kompleksni predobdelavi vključili segmentacijsko pragovno operacijo OPTIMAL. Kvalitetna segmentacijska metoda je v takšnem sistemu nujna, saj so njeni dobri rezultati pogoj za učinkovitost samega razpoznavalnega sistema.

Z našim razpoznavalnim sistemom smo razpoznali 61 % vseh mešičkov, sama segmentacijska metoda pa nam omogoča, da razpoznamo do 70 % mešičkov, nastala razlika je zaradi slabše opravljene klasifikacije (podpoglavje 5.2). V podpoglavju 5.3 smo nanizali tudi nekaj problemov, ki so se pojavili pri razpoznavanju mešičkov ter podali možne rešitve. Videli smo pa tudi, da se pri večini medicinskih problemov avtomatsko obdelovanje ne obnese, zato smo tudi predlagali, da se naš algoritem “xultra” dopolni tako, da bi ekspert lahko vplival na sam potek obdelovanja (predvsem pri določevanju položaja jajčnika).

Na koncu zapišimo še sklep. Obdelovanje medicinskih slik je tako kompleksna naloga, da jo s kontekstno neodvisnimi segmentacijskimi metodami (poglavje 3) težko rešimo. Zato je treba pri teh nalogah segmentacijske metode dopolniti z vgrajevanjem znanja o problemu, ki nam ga posredujejo eksperti, ali pa slike v skladu s tem znanjem ustrezno pripraviti (predobdelati) za analizo, kot npr. v našem sistemu, ko smo najprej izločili iz slike jajčnik in jo šele na to segmentirali.

6.

Sklep

V magistrskem delu smo se seznanili s pomenom segmentacije pri analizi medicinskih slik. Konstrukcija algoritmov za segmentiranje medicinskih slik je zelo zapleten postopek, ki zahteva kompleksno znanje o tipu medicinskih slik, o vrsti medicinskega problema ter o kontekstno neodvisnih segmentacijskih metodah, na kateri bo temeljil algoritem. Šele ko dobimo popolne odgovore na vsa zgornja vprašanja, so izpolnjeni vsi pogoji za dobro zasnovo segmentacijskega algoritma. V tem delu smo pridobljeno znanje o segmentaciji in segmentacijskih metodah uporabili pri reševanju problema določanja položaja jajčnih mešičkov iz ultrazvočnih posnetkov jajčnikov. Segmentacijski algoritem "xultra" smo razvijali po zgoraj zapisanih priporočilih, in sicer smo najprej preučili tip slik, medicinski problem, zatem pa še segmentacijske metode.

V 2. poglavju smo se tako seznanili z ultrazvokom kot eno izmed tipičnih medicinskih domen. Opisali smo lastnosti ultrazvoka, kako lahko s piezoelektričnim efektom kristalov povzročimo ultrazvočno valovanje, zatem pa še njegovo uporabo v medicinski diagnostiki. Videli smo, da se v medicini uporablja ultrazvok, ki ima frekvenco valovanja od 1 MHz do 10 MHz. V praksi je zelo pomembno, kakšnih frekvenc ultrazvok uporabljamo. Pri ultrazvoku je namreč dušenje tem večje, čim višja je frekvenca valovanja. Zaradi tega razloga tudi priporočajo, da se za medicinske preglede organov ali tkiv, ki so v večjih globinah, uporablja ultrazvok nižjih frekvenc (npr. 3,5 MHz), saj s tem dobimo slike veliko boljših kvalitet.

V medicinski ultrazvočni diagnostiki dobimo slike po pregledu pacientov. Če takšne slike gledamo z vidika digitalne obdelave slik, lahko opazimo, da so izredno šumne. To pomeni, da imamo zaradi močnega šuma velike probleme pri razpoznavanju (segmentiranju) objektov iz takšnih slik. Glavne faktorje, ki povzročajo motnje v ultrazvočnih slikah, smo že opisali (podpoglavje 2.1.4). To so zelo majhne telesne strukture, ki povzročajo pege, večkratni odmev ultrazvoka (od sonde do tkiva), sonda ni dovolj navlažena z želejem itd. Takšne motnje je, ne da bi pri tem pokvarili ostale vsebine slike, zelo težko odstraniti. Zato je najbolje, če je seveda mogoče, da apeliramo na zdravnike, da bi optimirali pogoje zajemanja slik, s tem pa bi dobili tudi veliko bolj kvalitetne slike. Treba se je zavedati, da takoj ko začnemo popravljati določeno lastnost slike, npr. izboljšanje kontrasta z izenačevanjem histograma, že opravimo določeno transformacijo slike, hkrati pa preidemo iz originalnega v transformiran prostor pikslov, kjer je treba relacije med sosednjimi piksli šele vzpostaviti (ta naloga je odvisna od tipa slik in je pogosto hevristična). Zato je smiselno opravljati le nujno potrebne transformacije. V primeru ultrazvočnih slik je odstranjevanje peg nujno. V podpoglavju 2.3 smo podrobno opisali metode za odstranjevanje peg. V splošnem lahko izberemo katerokoli metodo, paziti moramo le, da ta metoda ne zabriše robov.

V 3. poglavju smo nato opravili pregled obstoječih segmentacijskih metod. Preučili smo kontekstno neodvisne metode in jih razvrstili v 4 skupine, in sicer v pragovne operacije, metode na osnovi robov, na osnovi regij ter v metode ujemanja vzorcev. V drugem delu tega poglavja pa smo deset segmentacijskih algoritmov preizkusili na umetnih slikah, sestavljenih iz temnih elips, položenih na belo ozadje. Iz teh slik smo nato za posamezen algoritem merili njegovo segmentacijsko učinkovitost. Treba je poudariti, da smo slikam dodajali tudi šum, in sicer je bilo maksimalno razmerje SNR -1 dB. Na osnovi kriterijev, poimenovanih odstotek razpoznanih elips in odstotek razpoznanih regij, smo na koncu izbrane metode razvrstili. Za najboljšo se je izkazala pragovna segmentacijska metoda OPTIMAL, takoj za njo pa se je uvrstila segmentacijska metoda UNION-FIND, ki segmentira na osnovi rasti regije. Za nekatere je to morda presenetljiv rezultat. Nekateri avtorji celo trdijo, da pragovne operacije z globalnim pragom niso primerne in uspešne [27]. Vendar to ne drži popolnoma, saj je pragovna operacija v primerih, ko so sivine objektov in ozadja dobro ločene, celo odlična segmentacijska metoda. V naših umetnih testnih slikah je bila ločitev objektov (elips) in ozadja izrazita, zato je algoritem OPTIMAL, ki določa prag na polovici med povprečno sivino objekta in ozadja, uspešno segmentiral tudi zelo šumne testne slike. Metode rasti regije (tudi UNION-FIND) so veliko bolj kompleksne kot pragovne, poleg tega pa pri svojem delu potrebujejo še natančno določene začetne homogene regije in natančno postavljene kriterije homogenosti. Določitev začetnih regij se ponavadi prevede na še en problem segmentacije [63], kar pomeni, da moramo v tem primeru opraviti kar dvojno segmentacijo. Pri naši simulaciji smo začetne regije za metode rasti regij izbirali naključno.

Kljub temu, da sta se metodi OPTIMAL in UNION-FIND izkazali kot najboljši v simulaciji, ne trdimo, da ju moramo uporabljati pri vsaki segmentacijski nalogi. Izbira ustrezne metode je v prvi vrsti odvisna od tipa slik. Zato priporočamo, da opravimo analizo porazdelitev sivin v slikah, preden izberemo neko segmentacijsko metodo. To najbolje naredimo s histogramom sivin. Iz histogramov zelo preprosto vidimo, v kakšnem območju se gibljejo vrednosti sivin objektov in ozadja. Če so te vrednosti dobro ločene, potem je pragovna oz. večpragovna operacija prava segmentacijska metoda. V primerih, ko zavzemajo objekti le določen pas sivin in te sivine lahko najdemo tudi v ozadju, je najprimernejša metoda rasti regije. Ko so objekti zelo neenotnih sivin (npr. objekti z raznimi vzorci), vendar z zelo izrazito mejo, tedaj je najpreprosteje opraviti segmentacijo na osnovi robov. Metode ujemanja pa uporabimo takrat, ko v slikah iščemo že znane objekte (so lahko rotirani ali pomanjšani in povečani). Mnoge slike pa so tako kompleksne (npr. več različnih prekrivajočih se objektov na ozadju, ki se pa spreminja od enega roba do drugega), da lahko opravimo segmentacijo teh slik le z združitvijo več osnovnih metod (najprej npr. izvedemo pragovno operacijo, ki nam definira začetne regije za rast regij, zatem pa še te osnovne regije povečujemo). V nekaterih kompleksnih slikah niti kombinacija osnovnih segmentacijskih metod ne prinese želenega rezultata. Praviloma v teh slikah ne moremo najti takih lastnosti, ki bi veljale za celotno sliko. V takšnih primerih pa osnovno sliko razbijemo na množico manjših podslik, ki jih lahko preprosteje segmentiramo z osnovnimi metodami. Velja pa naslednja trditev: čim kompleksnejša je slika, tem več znanja o problemu (vrsti slik) moramo vključiti v segmentacijsko metodo, če želimo dobre rezultate.

V 4. poglavju smo reševali problem razpoznavanja jajčnih mešičkov s Houghovo transformacijo. Metoda HT je atraktivna in v zadnjem času vse bolj uporabljana segmentacijska metoda, ki segmentira na osnovi robov. Namenjena je predvsem za iskanje kompleksnih oblik v slikah, tako analitičnih (npr. premic in krožnic), kakor tudi neanalitičnih oblik (posplošena HT). To iskanje se iz prostora robnih pikslov (slikovnega prostora) prevede v parametričen prostor, kjer pa z raznimi transformacijami enostavneje odkrijemo iskane oblike. Po teoretičnem opisu transformacije HT smo jo preizkusili v praksi na problemu razpoznavanja jajčnih mešičkov. Jajčne mešičke smo aproksimirali z elipsami, pri njihovem razpoznavanju pa uporabili transformacijo HT za iskanje elips. Že po segmentiranju le nekaj ultrazvočnih posnetkov jajčnikov smo videli, da je odstotek razpoznavanja bistveno nižji kot pri segmentacijskem algoritmu "xultra" (poglavje 5), in to za približno 40 %. Vzrok za tako nizek odstotek razpoznavanja vidimo v aproksimaciji mešičkov z elipsami. Mešički so res bolj ovalne oblike, vendar pa nimajo popolnoma pravilne oz. gladke meje (nekateri deli so zelo nazobčani), in jih zato v splošnem ne moremo analitično popisati z elipsami, zato pa tudi ne povzročajo izrazitih maksimumov v Houghovem parametričnem prostoru, ki so bistveni pri razpoznavanju. Iz tega smo videli, da imamo v medicinskih slikah le redko opravka z razpoznavanjem pravilnih oblik. Zato so tudi segmentacijske metode, ki so zasnovane za iskanje pravilnih oblik, za medicinske slike manj uporabne. Njihovo uporabnost vidimo predvsem v začetnih fazah konstrukcije segmentacijskih algoritmov, ko se šele seznanjamo s problemom, saj lahko s temi metodami potrdimo določene domneve o oblikah iskanih objektov.

V nadaljevanju smo združili pridobljeno znanje o ultrazvoku, jajčnih mešičkih in segmentacijskih metodah tako, da smo skonstruirali algoritem za avtomatsko določanje položaja jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov. Zgradbo in ideje, ki so nas vodile pri njegovem konstruiranju, smo podrobno opisali v 5. poglavju. Pri tvorjenju tega algoritma smo se izogibali vsem segmentacijskim postopkom, ki bi bili kakorkoli vezani na samo obliko iskanih objektov. Znanje o obliki jajčnih mešičkov smo vpeljali v algoritem šele pri klasifikaciji (podpoglavje 5.1). Razviti algoritem "xultra", imenovali smo ga tudi razpoznavalni sistem, je kontekstno odvisen, saj smo pri njegovi tvorbi upoštevali posamezne lastnosti, specifične za ultrazvočne slike in za problem jajčnih mešičkov. Ko smo pri problemu razpoznavanja jajčnih mešičkov opravili primerjavo med kontekstno neodvisnim segmentacijskim algoritmom in algoritmom "xultra" (podpoglavje 3.5), smo videli, da se je v primeru kontekstno odvisnega segmentacijskega algoritma odstotek razpoznavanja dvignil skoraj za 20 %. To dejansko pomeni, da ob uporabi neke vsesplošne segmentacijske metode na realnih (medicinskih) slikah dobimo le povprečne rezultate, za boljše rezultate pa moramo te metode dopolniti z vgrajevanjem znanja o problemu. Z razvojem algoritma "xultra" smo se tudi naučili, da se moramo pri segmentiranju medicinskih slik izogibati postavljanju kriterijev, ki temeljijo na povprečnih vrednostih sivin oz. standardnih odklonih. V praksi moramo uporabljati normalizirane vrednosti. Ta problem smo v našem algoritmu rešili tako, da smo postavili kriterije, ki temeljijo na obliki jajčnih mešičkov (kompaktnost itd.), zato pa tudi nismo imeli nikakršnih problemov pri segmentiranju slik iz različnih ultrazvočnih naprav. Kadar imamo v segmentacijskem algoritmu veliko nastavljivih parametrov (pragov), moramo algoritem zasnovati tako, da se lahko ti parametri kalibrirajo (pragovi se lahko nastavijo po začetni obdelavi nekaj slik iz druge naprave).

Že v uvodu smo zapisali, da ni zlatih pravil za ocenjevanje učinkovitosti segmentacijskih algoritmov za medicinske slike. V praksi se dobljeni rezultati največkrat primerjajo z rezultati, ki jih posredujejo eksperti. Pri tem pa moramo vedeti, da je primerjava tem bolj verodostojna, čim več rezultatov od različnih ekspertov imamo. Najbolje je, da vsi eksperti segmentirajo isto medicinsko sliko, iz teh rezultatov pa zatem izračunamo povprečje, ki ga vzamemo za referenčno točko pri primerjavi rezultatov. S takšnim pristopom minimiziramo variabilnost, ki jo vnašajo eksperti.

Naš algoritem "xultra" je zasnovan tako, da po začetni kalibraciji pragov opravlja celotno segmentacijo oz. razpoznavanje popolnoma avtomatsko, brez interakcije uporabnika. Popolna avtomatizacija je v smislu razbremenitve rutinskega dela uporabnika-zdravnika želena, vendar pa hkrati tudi pomeni, da s tem popolnoma izgubimo nadzor nad potekom segmentacije. Tako se napaka, ki jo naredimo pri segmentaciji v določenem koraku (težava pri "xultra" je pregroba ocena jajčnika), ne more več odstraniti, ampak se skozi vse nadaljnje faze razpoznavanja le še povečuje. Zaradi tega tudi predlagamo, da medicinske segmentacijske algoritme raje zasnujemo polavtomatske. S to zahtevo vnesemo veliko večji nadzor in povečamo pravilnost segmentacije, saj dopustimo, da ekspert v kritičnih trenutkih poseže v segmentacijo in dobljene rezultate zavrne ali pa jih popravi. Ta interakcija je še posebej pomembna, kadar imamo po segmentaciji zlite objekte. Problem reševanja zlitih objektov je izredno zahteven. Če pa omogočimo poseg ekspertov, lahko le-ti razveljavijo segmentacijo tega dela slike ter zahtevajo njeno ponovitev, ali pa na tem delu slike opravijo segmentacijo ročno.

Med raziskavami smo spoznali, da se segmentacijske metode za segmentiranje medicinskih slik ne razlikujejo bistveno od metod, namenjenih za ostale slike. Tudi pri samem opisu segmentacije smo velikokrat izpustili besedo medicinski. Kljub tej podobnosti med metodami pa obstajajo med njimi tudi bistvene razlike. Segmentacijske metode za medicinske slike morajo imeti zelo nizek odstotek napačno razpoznanih objektov (bolan ali deformiran organ ne sme biti razpoznan kot zdrav), saj je lahko vsaka takšna napaka usodna. Takšna segmentacijska metoda pa mora biti v vseh fazah konstrukcije plod dela strokovnjakov s področja digitalne obdelave slik in s področja medicine.

Literatura

[1] J.C. Bamber, C. Daft, "Adaptive filtering for reduction of speckle in ultrasonic pulse-echo images", *Ultrasonics*, januar, str. 41-44, 1986.

[2] D. Ben-Tzvi, M.B. Sandler, "A combinatorical Hough Transform", *Pattern Recognition Letters*, vol. 11, str. 167-174, 1990.

[3] A.D. Brink, "Comments on grey-level thresholding of images using a correlation criterion", *Pattern recognition letters*, vol. 12, str. 91-92, 1991.

[4] P. J. Burt, T. Hong, A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchial computation", *IEEE Transactions on systems, man, and cybernetics*, vol. 11, št. 12, str. 802-809, 1981.

[5] P. J. Burt, "The pyramid as a structure for efficient computation", v: A. Rosenfeld, ur., *Multiresolution image processing and analysis* (Leesburg, VA, July 19-21, 1982), Berlin: Springer-Verlag, 1984, str. 6-35.

[6] D. Casasent, R. Krishnapuram, "Curved object location by Hough transformations and inversions", *Pattern Recognition*, vol. 20, št. 2, str. 181-188, 1987.

[7] K.R. Castleman, *Digital image processing.* Englewood Cliffs: Prentice Hall, 1996.

[8] V. Chalana, Y. Kim, "A methodology for evaluation of boundary detection algorithms on medical images", *IEEE Transactions on medical imaging*, vol. 16, št. 5, str. 642-652, 1997.

[9] Y. Chang, X. Li, "Adaptive image region-growing", *IEEE Transactions on image processing*, vol. 3, št. 6, str. 868-872, 1994.

[10] J.S. Chen, A. Huertas, G. Medioni, "Fast convolution with Laplacian-of-Gaussian masks", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 9, št. 4, str. 584-590, 1987.

[11] B.Cigale, B. Potočnik, D. Zazula, "Analiza verjetnostnih porazdelitev v ultrazvočnih posnetkih jajčnikov", v: I. Kononenko, T. Urbančič, ur., *Računalniška analiza medicinskih podatkov*, Zbornik CADAM-97, Bled, Slovenija, 12. november, str. 56-67, 1997.

[12] T.R. Crimmins, “Geometric filter for reducing speckle”, *Optical Engineering*, vol. 25, št. 5, str. 651-654, 1986.

[13] J.L Crowley, R.M. Stern, “Fast computation for the difference of low-pass transform”, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 6, št. 2, str. 212-222, 1984.

[14] R.O. Duda, P.E. Hart, “Use of the Hough transformation to detect lines and curves in pictures”, *Communications of the ACM*, vol. 15, št. 1, str. 11-15, 1972.

[15] A. Fenster, D.B. Downey, “3-D ultrasound imaging”, *IEEE Engineering in medicine and biology magazine*, vol. 15, št. 6, str. 41-51, 1996.

[16] C. Fiorio, J. Gustedt, *Two linear time union-find strategies for image processing.* Tehniško poročilo št. 375, Berlin: Tehniška Univerza Berlin, 1994.

[17] R.B. Fisher, S. Perkins, A. Walker, E. Wolfart, “Thinning”, *http://www.dai.ed.ac.uk/daidb/staff/personal_pages/rbf/HIPR/hiprdemo/html/thin.html.*

[18] C. Fuchs, W. Förstner, “Polymorphic grouping for image segmentation”, *Proceedings ICCV'95*, Boston, ZDA, str. 175-182, 1995.

[19] J.P Gambotto, “A new approach to combining region growing and edge detection”, *Pattern recognition letters*, vol. 14, str. 869-875, 1993.

[20] C.A. Glasbey, “An analysis of histogram-based thresholding algorithms”, *CVGIP: Graphical models and image processing*, vol. 55, št. 6, str. 532-537, 1993.

[21] M.A. Gore, P.L. Nayudu, V. Vlaisavljevic, N. Thomas, “Prediction of ovarian cycle outcome by follicular characteristics, stage 1”, *Human Reproduction*, vol. 10, št. 9, str. 2313-2319, 1995.

[22] L. O'Gorman, “Binarization and multithresholding of document images using connectivity”, *CVGIP: Graphical models and image processing*, vol. 56, št. 6, str. 494-506, 1994.

[23] J.Y. Goulermas, P. Liatsis, “The two-point combinatorial probabilistic Hough transform for circle detection”, *Proceedings IWISP'96*, Manchester, UK, 4.-7. november 1996, str. 169-172, 1996.

[24] H. Guo, J.E. Odegard, M. Lang, R.A. Gopinath, I.W. Selesnick, C.S. Burrus, *Wavelet based speckle reduction with application to SAR based ATD/R.* Tehniško poročilo št. CML TR94-02, Houston: Univerza Rice, 1994.

[25] I. Hannah, D. Patel, R. Davies, “The use of variance and entropic thresholding methods for image segmentation”, *Pattern recognition*, vol. 28, št. 8, str. 1135-1143, 1995.

[26] R. M. Haralick, L. Shapiro, "Image segmentation techniques", *Computer vision, graphics, and image processing*, vol. 29, str. 100-132, 1985.

[27] F. Heijden, *Image based measurement systems.* London: Wiley and Sons, 1994.

[28] I.L. Herlin, N. Ayache "Feature extraction and analysis methods for sequences of ultrasound images", *Image and vision computing*, vol. 10, št. 10, str. 673-682, 1992.

[29] C.L. Huang, "Elliptical feature extraction via improved Hough transform", *Pattern Recognition Letters*, vol. 10, str. 93-100, 1989.

[30] J. Illingworth, J.Kittler, "A survey of the Hough transform", *Computer vision, graphics and image processing*, vol. 44, str. 87-116, 1988.

[31] B. Jähne, *Digital image processing.* Berlin: Springer-Verlag, 1993.

[32] J.N Kapur, P.K. Sahoo, A.K.C Wong, "An new method for gray-level picture thresholding using the entropy of the histogram", *Computer vision, graphics, and image processing*, vol. 29, str. 273-285, 1985.

[33] M. Karaman, M. Alper Kutay, G. Bozdagi, "An adaptive speckle suppression filter for medical ultrasonic imaging", *IEEE Transactions on medical imaging*, vol. 14, št. 2, str. 283-292, 1995.

[34] A. Klemenčič, F. Pernuš, S. Kovačič "Primerjava rezultatov ročne in polavtomatske razgradnje slik mišičnih vlaken", *Zbornik ERK'97*, Portorož, Slovenija, 25.-27. september 1997, zvezek B, str. 261-264, 1997.

[35] M. Kocher, R. Leonardi, "Adaptive region growing technique using polynomial functions for image aproximation", *Signal processing*, vol. 11, str. 47-60, 1986.

[36] M. Kolenc, *Porodništvo.* Ljubljana: Zavod Republike Slovenije za šolstvo in šport, 1993.

[37] C. Kontropoulos, X. Magnisalis, I. Pitas, M.G. Strintzis, "Nonlinear ultrasonic image processing based on signal-adaptive filters and self-organizing neural networks", *IEEE Transactions on image processing*, vol. 3, št. 1, str. 65-77, 1994.

[38] J.I. Koo, S.B. Park, "Speckle reduction with edge preservation in medical ultrasonic images using a homogeneous region growing mean filter (HRGMF)", *Ultrasonic imaging*, vol. 13, str. 211-237, 1991.

[39] A. Kotnik, F. Pernuš, "Razgradnja staknjenih dvodimenzionalnih objektov", *Zbornik ERK'96*, Portorož, Slovenija, 19.-21. september 1996, zvezek B, str. 281-284, 1996.

[40] S.M. Kubica, *Automatic segmentation of cranial ultrasound images used for calculating ventricular volumes.* Magistrsko delo, New Mexico: Fakulteta za elektrotehniko in računalništvo, 1994.

[41] A. Kurjak, B. Funduk-Kurjak, *Ultrazvuk i neplodnost.* Zagreb: Stvarnost, 1984.

[42] V.F. Leavers, "Which Hough transform?", *Computer vision, graphics and image processing*, vol. 58, št. 2, str. 250-264, 1993.

[43] C.H. Li, C.K Lee, "Minimum cross entropy thresholding", *Pattern recognition*, vol. 26, št. 4, str. 617-625, 1993.

[44] W. Lie, "Automatic target segmentation by locally adaptive image thresholding", *IEEE Transaction on image processing*, vol. 4, št. 7, str. 1036-1041, 1995.

[45] G.R. Lockwood, D.H. Turnbull, D.A. Christopher, F.S. Foster, "Beyond 30 MHz", *IEEE Engineering in medicine and biology magazine*, vol. 15, št. 6, str. 60-71, 1996.

[46] F. Marino, G. Mastronardi, "Hy2: Hybrid segmentation method", *Proceedings IWISP'96*, Manchester, UK, 4.-7. november 1996, str. 311-314, 1996.

[47] H. Meden-Vrtovec in sodelavci, *Neplodnost.* Ljubljana: Cankarjeva založba, 1989.

[48] S. Molander, H. Broman, "Knowledge-based segmentation and state-based control in image analysis: Two examples from biomedical domain", *Signal processing*, vol. 32, str. 201-215, 1993.

[49] R. Momenan, R.F. Wagner, B.S. Garra, M.H. Loew, M.F. Insana, "Image staining and differential diagnosis of ultrasound scans based on the mahalanobis distance", *IEEE Transactions on medical imaging*, vol. 13, št. 1, str. 37-47, 1994.

[50] O. Monga, "An optimal region growing algorithm for image segmentation", *International journal of Pattern recognition and artificial intelligence*, vol. 1, št. 3&4, str. 351-375, 1987.

[51] F. Morii, "An image thresholding method using a minimum weighted squared-distortion criterion", *Pattern recognition*, vol. 28, št. 7, str. 1063-1071, 1995.

[52] D. Pao, H.F. Li, R. Jayakumar, "A decomposable parameter space for the detection of ellipses", *Pattern Recognition Letters*, vol. 14, str. 951-958, 1993.

[53] J.R. Parker, "Gray level thresholding in badly illuminated images", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 13, št. 8, str. 813-819, 1991.

[54] N. Pavešić, *Razpoznavanje vzorcev.* Ljubljana: Fakulteta za elektrotehniko in računalništvo, 1992.

[55] A. Pérez, R.C Gonzalez, "An iterative tresholding algorithm for image segmentation", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 9, št. 6, str. 742-751, 1987.

[56] Ž. Pilić, A. Stanković, *Ultrazvuk u ginekologiji.* Beograd: Medicinska knjiga, 1991.

[57] M. Popovic, F. Chantemarque, R. Canals, P. Bonton, "Several approaches to implement the merging step of the split and merge region segmentation", in: F.H. Post, W. Barth, ed., *EUROGRAPHICS '91*, Elsevier science publishers B.V., 1991, str. 399-412.

[58] A.A. Rodriguez, O.R. Mitchell, "Image segmentation by successive background extraction", *Pattern recognition*, vol. 24, št. 5, str. 409-420, 1991.

[59] D. Rueckert, P. Burger, S.M. Forbat, R.D. Mohiaddin, G.Z. Yang, "Automatic tracking of the aorta in cardiovascular MR images using deformable models", *IEEE Transactions on medical imaging*, vol. 16, št. 5, str. 581-590, 1997.

[60] J.C. Russ, *The image processing handbook.* London: CRC Press, 1995.

[61] K.K. Shung, M. Zipparo, "Ultrasonic transducers and arrays", *IEEE Engineering in medicine and biology magazine*, vol. 15, št. 6, str. 20-30, 1996.

[62] S. Song, M. Liao, J. Qin, "Multiresolution image dynamic thresholding", *Machine Vision and applications*, vol. 3, str. 13-16, 1990.

[63] M. Sonka, V. Hlavac, R. Boyle, *Image processing, analysis and machine vision.* London: Chapman and Hall, 1994.

[64] P.F. Stetson, F.G. Sommer, A. Macovski, "Lesion contrast enhancement in medical ultrasound imaging", *IEEE Transactions on medical imaging*, vol. 16, št. 4, str. 416-425, 1997.

[65] A. Šantić, *Biomedicinska elektronika.* Zagreb: Školska knjiga, 1995.

[66] D. Tseng, W. Shieh, "Plume extraction using entropic thresholding and region growing", *Pattern recognition*, vol. 26, št. 5, str. 805-817, 1993.

[67] S.E. Umbaugh, Y. Ewi, M. Zuke, "Feature extraction in image analysis", *IEEE Engineering in medicine and biology magazine*, vol. 16, št. 4, str. 62-73, 1997.

[68] Y. Yan, X. Bao, Y. Bai, L. Luo, H. Shu, Y. Fu, "3-D imaging and stereotactic radiosurgery", *IEEE Engineering in medicine and biology magazine*, vol. 16, št. 4, str. 47-52, 1997.

[69] R.K.K Yip, P.K.S Tam, D.N.K. Leung, "Modification of Hough transform for circles and ellipses detection using a 2-dimensional array", *Pattern Recognition*, vol. 25, št. 9, str. 1007-1022, 1992.

[70] S. Wang, R.M. Haralick, “Automatic multithreshold selection”, *Computer vision, graphics, and image processing*, vol. 25, str. 46-67, 1984.

[71] D. Wei, H. Guo, J.E. Odegard, M. Lang, C.S. Burrus, *Simultaneous speckle reduction and data compression using best wavelet packet bases with application to SAR based ATD/R*. Tehniško poročilo št. CML TR95-02, Houston: Univerza Rice, 1995.

Življenjepis

Ime in priimek:	Božidar Potočnik.	
Rojen:	2. 9. 1972 v Mariboru.	
Oče:	Adolf Potočnik.	
Mati:	Veronika Potočnik.	
Šolanje:	1979-1987	Osnovna šola Borci za severno mejo Maribor,
	1987-1991	Druga gimnazija Maribor,
	1991-1995	Fakulteta za elektrotehniko, računalništvo in informatiko Maribor.
Zaposlitev:	1995-danes	Fakulteta za elektrotehniko, računalništvo in informatiko Maribor.

Bibliografija