Univerza v Mariboru

Fakulteta za elektrotehniko, računalništvo
in informatiko

Božidar Potočnik

Razpoznavanje objektov iz zaporedja slik s postopki predikcije

Doktorska disertacija

Maribor, september 2000

Avtor: Božidar Potočnik

Naslov: Razpoznavanje objektov iz zaporedja slik s postopki predikcije

UDK: 681.3.019:621.397:616

Ključne besede: digitalna obdelava slik, razpoznavanje objektov, zaporedje slik, gibajoči se objekt, sledenje objektom, segmentacija, rast regij, metode prediktor-korektor, Kalmanov filter, predikcija, modeliranje, ultrazvočni posnetki, pegasti šum, jajčni mešički

Število izvodov: 9

Obdelava besedila in slik: Božidar Potočnik

Razmnoževanje: Laboratorij za sistemsko programsko opremo,
Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor

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

mag. Božidar Potočnik, dipl. inž. računalništva,

dne 14. 9. 2000 uspešno zagovarjal svojo doktorsko disertacijo z naslovom

Razpoznavanje objektov iz zaporedja slik s postopki predikcije

pred komisijo v sestavi:

prof. dr. Ivan Rozman, predsednik

prof. dr. Damjan Zazula, mentor – član

prof. dr. Franc Solina, komentor – član

prof. dr. Nikola Guid, član

Razpoznavanje objektov iz zaporedja slik s postopki predikcije

UDK: 681.3.019:621.397:616

Ključne besede:

digitalna obdelava slik, razpoznavanje objektov, zaporedje slik, gibajoči se objekt, sledenje objektom, segmentacija, rast regij, metode prediktor-korektor, Kalmanov filter, predikcija, modeliranje, ultrazvočni posnetki, pegasti šum, jajčni mešički

Povzetek:

V disertaciji se ukvarjamo z avtomatskim razpoznavanjem objektov iz zaporedja 2D slik. Ob tem predpostavljamo, da o samem zaporedju ni na voljo nobenih vnaprejšnjih informacij, razen morebiti o tipu slik in iskanih objektov, in niti ni možno zagotoviti, da so spremembe od slike do slike majhne. Sledimo vsem objektom v sliki, vsak objekt pa se lahko giblje s svojim lastnim gibanjem. V začetku disertacije podajamo pregled obstoječih metod in tehnik obdelovanja zaporedja digitalnih slik, poseben poudarek je namenjen metodam razpoznavanja. Metode kritično ovrednotimo in jih razvrstimo v pet področij: a) metode za analizo gibanja v slikah, b) segmentacijske metode na osnovi gibanja, c) aktivne krivulje, d) tehnike pri 3D slikah in e) metode prediktor-korektor.

Raziskavo nadaljujemo s konstrukcijo dveh novih razpoznavalnih sistemov za avtomatsko razpoznavanje objektov iz zaporedja 2D posnetkov. Zaradi lažjega verificiranja uspešnosti razpoznavanja uporabljamo pri testiranju ob realnih ultrazvočnih posnetkih še umetne slike jajčnikov z mešički. Za tvorjenje umetnih slik razvijemo nov postopek. Simulacijski 3D model jajčnikov z mešički tvorimo na osnovi ovalnih geometrijskih konstruktov. Model kalibriramo, zatem pa vzpostavimo računalniški simulator za projiciranje prerezov skozi model jajčnika pod poljubnim kotom in zasukom.

Sledi opis osnovnega postopka za razpoznavanje temnih ovalnih objektov na eni statični 2D sliki, t. i. algoritem za statične 2D slike. Ob rasti regije, uporabljeni za segmentacijsko metodo, vpeljujemo pri razpoznavanju še obteženi gradient in množico statističnih testov. Razviti algoritem verificiramo na umetnih in realnih ultrazvočnih posnetkih jajčnikov. V umetnih slikah so vse razmere točno znane, na realnih posnetkih pa uspešnost metode merimo na osnovi odčitkov, ki so jih posredovali zdravniki specialisti. Opišemo ustrezne kvantitavne mere za vrednotenje uspešnosti algoritma. Odstotek razpoznavanja je na umetnih slikah okrog 90 %, na realnih pa okrog 78 %. Povprečna absolutna razdalja med mejo razpoznanega in pravilnega mešička je v obeh primerih okrog 1 mm.

Zatem preidemo od razpoznavanja objektov v posameznih statičnih 2D slikah k razpoznavanju, kjer upoštevamo celotno informacijo iz zaporedja slik. Po natančni matematični formulaciji problema dokažemo, da je najoptimalneje, če zgodovino razpoznavanja na prejšnjih slikah vgradimo v Kalmanov filter in jo kot tako uporabimo pri razpoznavanju na trenutni sliki. Podrobno opišemo nov algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije, t. i. predikcijski algoritem. Sestavni del algoritma je Kalmanov filter. Merilni sistem realiziramo z algoritmom za statične 2D slike. Na osnovi meritve v začetni sliki vzpostavimo model objekta. Ta model zatem s Kalmanovim filtrom na osnovi meritev iz slike v sliko modificiramo. Izračunani model pa na posamezni sliki določa nov najboljši približek iskanega objekta. Identificiramo tudi možne probleme, ki se lahko pojavijo pri razpoznavanju, ter ob osnovni varianti predikcijskega algoritma podamo še tri izboljšave: za primer velikih napak v meritvah, za upoštevanje lokalnosti krivulje in za popravljanje neustreznih meritev. Končno obliko predikcijskega algoritma podamo v obliki psevdokoda.

Predikcijski algoritem prav tako preizkusimo na zaporedju umetnih in realnih ultrazvočnih posnetkov jajčnikov z mešički. Za vrednotenje uporabimo že vpeljane kvantitativne mere. Po primerjavi rezultatov pokažemo, da so rezultati predikcijskega algoritma veliko bolj kompaktni in natančni kot rezultati algoritma za statične 2D slike. Občutno se zmanjša število napačno razpoznanih mešičkov (tudi do 80 % glede na začetne vrednosti). Izkaže se, da s takšnim obdelovanjem zaporedja slik povečamo gotovost v dobljene rezultate. Predikcijski algoritem v določenih težavnih situacijah detektira mešičke pravilneje kot pa zdravnik ekspert.

Prediction-Based Object Recognition from a Sequence of Images

UDC: 681.3.019:621.397:616

Keywords:

digital image processing, object recognition, image sequence, moving object, object tracking, segmentation, region growing, predictor-corrector methods, Kalman filter, prediction, modelling, ultrasound images, speckle noise, ovarian follicles

Abstract:

In doctoral thesis, automated object recognition from a sequence of 2D images is dealt with. We suppose that no prior knowledge about the sequence is disposable, with the exception of type of images and objects that are searched for, neither is possible to assure that variations from image to image are small. All objects in the image are tracked, each object, however, could move with its own motion. The thesis begins with an overview of existing methods and techniques for processing the sequence of digital images, emphasising the recognition methods. The methods are critically assessed and are classified into five areas: a) methods for motion analysis in images, b) segmentation methods based on motion, c) active contours, d) techniques for 3D images, and e) predictor-corrector methods.

The research is continued with construction of two new recognition systems for automated object recognition from a sequence of 2D images. To be able to verify the recognition efficiency easier, real ovarian ultrasound images with follicles and also artificial images are tested. A new procedure for creating artificial images is developed. A 3D simulation ovarian model with follicles is formed based on oval geometric constructs. The model is calibrated; afterwards, a computer simulator for projecting the cross-sections through ovarian model at an arbitrary angle and shift is set.

Basic procedure for recognition of dark oval objects in a static 2D image, i.e. an algorithm for static 2D images, follows. Beside the region growing used as a segmentation method, a weighted gradient and a set of statistical tests are introduced for recognition. The developed algorithm is verified using artificial and real ovarian ultrasound images. In the artificial images, all conditions are precisely known, while for real images the efficiency of method is measured according to the readings given by a doctor specialists. Appropriate quantitative measures for assessing the efficiency of algorithm are described. Recognition rate for the artificial images is around 90 % and for real images around 78 %. An average absolute distance between the boundary of recognized and correct follicles is in both cases about 1 mm.

Then we pass from object recognition in a single static 2D image to the recognition where entire information from an image sequence is considered. Through accurate mathematical formulation of a problem we prove that it is the most optimal if recognition history from previous images is built in a Kalman filter and is used for recognition on current image. The new algorithm for object recognition from an image sequence with prediction procedures, i.e. prediction algorithm, is described in detail. It is based on the Kalman filter. The measurement system is realized with the algorithm for static 2D images. Based on measurements in the first image, an object model is set. Afterwards, this model is modified from image to image with Kalman filter regarding new measurements. The model calculated for particular image defines the new best estimate of the object searched for. Possible problems appearing at such a recognition are identified, and beside the basic prediction algorithm three improvements are given: for the case of big errors in measurements, for considering the locality of object contours, and for correcting the unsuitable measurements. The final prediction algorithm is given in a pseudocode form.

The prediction algorithm is tested on sequences of artificial and real ovarian ultrasound images with follicles. For the assessment, already introduced quantitative measures are used. The obtained results are much more compact and accurate with the prediction algorithm than with the algorithm for static 2D images. The number of misidentified follicles is considerable lowered (also to 80 % according to starting values). It points out that with such processing of image sequences reliability of the obtained results is greatly increased. The prediction algorithm in some difficult situations detects follicles even more correctly than a doctor expert does.

Zahvala

Za strokovno pomoč v celotnem času podiplomskega študija in še posebej pri izdelavi doktorske disertacije se iskreno zahvaljujem svojemu mentorju prof. dr. Damjanu Zazuli, ki me je s svojimi idejami, razpravami in nasveti pri raziskovanju vedno hrabril, vzpodbujal in pomagal. Za nasvete in komentarje se zahvaljujem tudi komentorju prof. dr. Francu Solini in poročevalcu prof. dr. Nikoli Guidu.

Prav tako se zahvaljujem vsem sodelavcem iz Laboratorija za sistemsko programsko opremo, ki so mi z drobnimi nasveti velikokrat sneli iz ramen velika bremena. Posebej bi se rad zahvalil Borisu, ki je digitaliziral ves slikovni material uporabljen v tej disertaciji.

Velika zahvala gre tudi prof. dr. Veljku Vlaisavljeviću iz Oddelka za človeško reprodukcijo in endokrinologijo, Splošna bolnišnica Maribor, ki je svojimi nasveti ter evalviranjem dobljenih rezultatov pripomogel 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.

Hvala vsem.

Kazalo

Seznam algoritmov

Seznam važnejših uporabljenih oznak

$\mathbf{A}$	matrika $\mathbf{A}$; velike mastne črke predstavljajo matrike;
$\mathbf{b}$	vektor $\mathbf{b}$; male mastne črke predstavljajo vektorje;
d($\mathbf{a}$,B)	razdalja točke $\mathbf{a}$ do krivulje B;
d(A,B)	razdalja med krivuljo A in B;
$\mathbf{H}_k$	matrika Kalmanovega filtra, ki daje idealno povezavo med stanjem sistema $\mathbf{x}_k$ in meritvijo $\mathbf{z}_k$;
$\mathbf{I}$	zaporedje digitalnih 2D slik;
I_k	k-ta digitalna slika iz zaporedja;
I	slikovna funkcija (zvezna predstavitev);
$I(\mathbf{p})$	vrednost slikovne funkcije I pri pikslu $\mathbf{p}$;
$I(x,y,t)$	vrednost slikovne funkcije I na mestu določenim s prostorskima koordinatama x in y ter časovno koordinato t;
$\triangle I_k$	slika razlik med dvema zaporednima digitalnima slikama I_k in I_{k+1};
$\frac{\partial I}{\partial x}$, I_x	parcialni odvod slikovne funkcije I po prostorski koordinati x (podobno po y);
$\frac{\partial I}{\partial t}$, I_t	parcialni odvod slikovne funkcije I po času t;
∇I	prostorski gradient slikovne funkcije I;
I_{xy}	drugi odvod slikovne funkcije I, najprej po x, zatem pa po y (podobno

označimo odvod I po x in nato še po času t z I_{xt});

$\mathcal{I}$ enotska matrika;

k_i najboljši približek (krivulja) objekta v i-tem koraku;

$\mathbf{k}_{j,i}$ j-ta točka (piksel) s krivulje najboljšega približka objekta $\mathbf{k}_i$ v i-tem koraku;

$m()$ operator povprečja, npr. $m(R)$ – povprečna sivina regije R;

$p(R)$ ploščina regije R;

$\mathrm{p}(x)$ gostota verjetnosti;

$\mathbf{p}_i$ i-ti piksel;

p_i^J element p (ploščina) iz stanja sistema $\mathbf{x}_i^J$ za Kalmanov filter J v i-tem koraku (podobno za κ, $\triangle p$ itd.);

$\mathbf{P}_k$ kovariančna matrika napake v koraku k za Kalmanov filter;

$\mathbf{Q}_k$ matrika šuma v sistemu v koraku k za Kalmanov filter;

r_1, r_2 količnika za merjenje senzitivnosti oz. specifičnosti rezultatov;

$\mathbf{R}_k$ matrika šuma v merilnem sistemu v koraku k za Kalmanov filter;

$\mathbf{S}$ zaporedje segmentiranih slik;

t_k k-ti časovni trenutek;

$\triangle t$ razmik med dvema časovnima trenutkoma;

T, T_i prag oz. i-ti prag;

$\mathbf{u}$ optični pretok;

$\mathbf{v}$ 2D polje gibanja;

$\mathbf{x}_k$ vektor stanja sistema v Kalmanovem filtru v koraku k;

$\hat{\mathbf{x}}_k$ najboljša ocena stanja sistema; s strešico označujemo oceno;

$\hat{\mathbf{x}}_k^-$ predikcija stanja sistema; z zgornjim minusom označujemo predikcijo;

$\mathbf{x}_{j,i}$ stanje sistema za globalni Kalmanov filter za j-to točko v i-tem koraku;

$\mathbf{x}_i^J$ stanje sistema za Kalmanov filter za ocenjevanje velikih napak v meritvah v i-tem koraku; oznaka J v potenci označuje Kalmanov filter J;

$\mathbf{x}_i^{LOK}$ stanje sistema za Kalmanov filter za ocenjevanje lokalnosti krivulje v i-tem koraku; oznaka LOK v potenci označuje Kalmanov filter LOK;

$x_{j,i}$ element x (koordinata x) iz stanja sistema $\mathbf{x}_{j,i}$ za globalni Kalmanov filter za j-to točko v i-tem koraku (podobno za y, $\triangle x$ itd.);

x_i^{LOK} element x (koordinata x) iz stanja sistema $\mathbf{x}_i^{LOK}$ za lokalni Kalmanov filter v i-tem koraku (podobno za y, $\triangle x$ itd.);

z_i meritev (krivulja) objekta v i-tem koraku;

$\mathbf{z}_{j,i}$ meritev za globalni Kalmanov filter za j-to točko v i-tem koraku (podobno za lokalni KF in KF J);

$\mathbf{z}_k$ vektor meritev za Kalmanov filter v koraku k;

$z_{x_{j,i}}$ meritev za element x iz vektorja meritev $\mathbf{z}_{j,i}$ za j-to točko v i-tem koraku za globalni Kalmanov filter (podobno za y);

$z_{p_i^J}$ meritev za element p (ploščina) iz vektorja meritev $\mathbf{z}_i^J$ v i-tem koraku za Kalmanov filter J (podobno za Kalmanov filter LOK);

σ standardni odklon;

κ kompaktnost;

$\mathbf{\Phi}_k$ matrika, ki povezuje stanja sistema $\mathbf{x}_k$ in $\mathbf{x}_{k+1}$ v Kalmanovem filtru.

1.

Uvod

Razpoznavanje objektov z računalniško podprtimi postopki se že od vsega začetka uvršča v aktualno raziskovalno področje digitalne obdelave slik in razpoznavanja vzorcev. V prvih letih, ko se je to področje raziskovanja šele formiralo, so bile metode razpoznavanja sila preproste ter z veliko uporabniške interakcije, v zadnjih letih pa dobivajo vse večjo veljavo polavtomatske oz. tudi popolnoma avtomatske metode razpoznavanja objektov v digitalnih slikah. V [DA00] so avtorji pregledno opisali področje obdelovanja in analize medicinskih slik. Prikazali so, kako se je področje razvijalo skozi leta. Podoben razvoj, od preprostih k izredno sofisticiranim metodam razpoznavanja objektov v digitalnih slikah, zasledimo tudi na drugih področjih obdelovanja.

V letih pred 1984 so se pojavljale predvsem metode, ki so analizirale in razpoznavale vzorce v statičnih 2D digitalnih slikah [DA00]. Uporabljene so bile metode, ki jih danes imenujemo klasične. Tipičen pristop je bil takle: v sliki detektirati robove, zatem pa jih z določenimi operatorji povezati med seboj in s tem določiti objekt. Razvoj metod je v letih med 1985 in 1991 šel naprej. Pojavljati so se začeli razni novi pristopi na osnovi znanja (*knowledge-based approaches*). Razpoznavalni sistemi sedaj vsebujejo razne strategije razmišljanja višjih nivojev. Te sisteme so prej vodili zgolj pristopi obdelovanja slik in razpoznavanja vzorcev v smeri od spodaj navzgor (*bottom-up*). Pojavijo se tudi nove veje raziskovanja, kot npr. računalniško podprto diagnosticiranje, zelo sofisticirane segmentacijske metode (aktivne krivulje, rast regij z uporabo multiparametričnih podatkov, uporaba tekstur pri segmentaciji) in začetni pristopi k analizi gibanja v zaporedju 2D slik. Mejnik v razvoju metod predstavljajo devetdeseta leta, predvsem zaradi prihoda novih, zmogljivejših naprav za zajem slik oz. zaporedja slik (3D slik). V letih od 1992 do danes tako zasledimo, da postanejo jedro raziskav 3D slike, zasledimo pa tudi premik k bolj integrirani analizi. Pojavi se potreba po večjem sofisticiranju postopkov za obravnavo vse bolj kompleksnih problemov. Ti nastanejo zaradi hitre rasti novih tipov podatkov, podatki pa so ob tem vse bolj natančni in zanesljivi. Kljub temu pa ostaja danes še nekaj ključnih izzivov, ki bi jih morali razvijalci upoštevati pri konstruiranju razpoznavalnih sistemov [DA00]:

1. Vsak razpoznavalni postopek naj bi bil skonstruiran in jasno motiviran s problemi iz realnosti (npr. medicinsko-biološki problemi). Pri analizi je treba zato upoštevati ne le normalne strukture in funkcije (npr. normalno delovanja organov in procesov), ampak tudi abnormalne oz. patološke primere. Slednje mnogi razpoznavalni algoritmi popolnoma ignorirajo.

2. Postopki obdelave in analize naj bodo čim bolj v stiku s samo strategijo zajemanja slik. Treba je upoštevati način tvorjenja slik in ga čim bolje integrirati v postopek razpoznavanja objektov (npr. v fazo predobdelave).

3. Razvoj ustreznih validacijskih in evalvacijskih pristopov za preverjanje uspešnosti razpoznavalnih sistemov je nujno potreben. Velika je tudi potreba po kreiranju skupnih podatkovnih baz s podatki za slike, zaporedja 2D slik, 3D slike, zaporedja 3D slik, na katerih bi lahko med seboj primerjali algoritme s posameznih področij.

Konstrukcija takšnih razpoznavalnih sistemov je zahtevna naloga. Ob samem tipu slik, s katerimi imamo opravka pri iskanju objektov, je treba upoštevati še mnoge druge faktorje, kot so npr. način tvorjenja slik, šum, zahtevana natančnost rezultata, interpretacija rezultatov. Obstaja kar nekaj področij, kjer se takšni razpoznavalni postopki že koristno uporabljajo (npr. razpoznavanje pisav, gradbeništvo, nadzor industrijskih procesov, mnogo je tudi medicinskih aplikacij). Treba je še poudariti, da ne obstaja metoda, ki bi bila preverjena in dokazana ter splošnonamenska. Problem se še stopnjuje, ko preidemo od razpoznavanja objektov v dvodimenzionalnih slikah k razpoznavanju v zaporedju dvodimenzionalnih posnetkov oz. k pravim 3D slikam (zaporedju 3D slik). Tukaj ne gre le za povečanje časovne in prostorske kompleksnosti metod na račun večje količine podatkov, ampak se z novimi tipi podatkov pojavijo tudi nove naloge, ki jih rešujemo pri 3D rekonstrukciji scene, določitvi 3D hitrosti gibanja objektov ipd.

V našem delu se bomo ukvarjali z avtomatskim razpoznavanjem objektov v zaporedju dvodimenzionalnih slik. Čeprav je področje razpoznavanja objektov iz zaporedja slik relativno mlado področje raziskovanja, so metode obdelave in analize v zaporedju slik ter 3D segmentacijske tehnike v literaturi že dobro zastopane [J93, SHB94, TV98, Rus95, Cas96]. Zasledili pa smo, da imajo te metode skupne nekatere omejitve (ali vsaj katero izmed njih): a) ozadje skozi celotno zaporedje slik naj bo konstantno oz. lahko pride le do majhnih sprememb v svetlosti, b) v sceni se ponavadi gibljejo vsi objekti podobno, c) pri opazovanju scene z gibajočo se kamero so parametri in hitrosti kamere vnaprej znani, d) za medicinske slike so znani parametri, kot so npr. razmik med dvema prerezoma, koti, pod katerimi so zajete slike (koti snemanja), oblika iskanih objektov, v posameznih primerih pa še tretja, tj. prostorska koordinata. Metode ponavadi tudi niso popolnoma avtomatske. Le delno je tudi rešen problem razpoznavanja oz. zasledovanja objektov v zaporedju slik, pri čemer objekti skozi zaporedje izginjajo in se ponovno pojavijo. Podobno lahko gre tudi za pojav novega objekta.

Te probleme bomo študirali. V delu se torej ukvarjamo z avtomatskim razpoznavanjem objektov iz zaporedja dvodimenzionalnih slik. Ob tem pa predpostavljamo, da o samem zaporedju ni na voljo nobenih vnaprejšnjih informacij (o pogojih zajemanja slik, položaju objektov), razen morebiti o tipu slik in iskanih objektov, in niti ni možno zagotoviti, da so spremembe od slike do slike majhne. Sledili bomo vsem objektom v sliki, pri tem pa bomo dopustili, da se vsak objekt giblje s svojim lastnim gibanjem. Dovoljeno je tudi, da objekti skozi zaporedje izginejo in se ponovno pojavijo. Skušali bomo pokazati, da metode analize in zaznavanja gibanja ter 3D tehnike v svoji osnovni obliki niso primerne za razpoznavanje objektov v takšnih situacijah. Vse izledke bomo preverili na umetnih in na realnih ultrazvočnih posnetkih jajčnikov, kjer bomo skušali slediti jajčnim mešičkom.

V tem delu skušamo dokazati naslednjo tezo:

Razpoznavanje na osnovi modela, zgrajenega v odvisnosti od področja problema in tipa slik, ki ga pri razpoznavanju iterativno modificiramo skozi zaporedje 2D posnetkov ali prerezov istega 3D objekta, ter ustrezne kriterijske funkcije, je optimalno. Nadgradnja osnovnega postopka za razpoznavanje objektov v zaporedju slik s postopki predikcije bistveno izboljša uspešnost in natančnost razpoznavanja.

Poglejmo, kako smo zastavili raziskavo. Najprej v 2. poglavju podajamo pregled obstoječih metod in tehnik obdelovanja zaporedja digitalnih slik, poseben poudarek je namenjen metodam razpoznavanja. Metode ustrezno razvrstimo, ovrednotimo jih tudi glede na naš zadani problem, ob tem pa predlagamo možne izboljšave. V 3. poglavju razvijemo osnovni postopek za razpoznavanje objektov, t. i. algoritem za razpoznavanje objektov v statičnih 2D slikah. Na začetku poglavja najprej opišemo nov postopek za tvorjenje umetnih ultrazvočnih posnetkov jajčnikov. Računalniški, tj. simulacijski 3D model jajčnikov z mešički razvijemo na osnovi ovalnih geometrijskih konstruktov. Model kalibriramo in potrdimo glede na dosegljive klinične podatke o realnih tvorbah mešičkov v jajčnikih. Zatem vzpostavimo računalniški simulator za projiciranje prerezov skozi model jajčnika pod poljubnim kotom in zasukom ter izhajajoč iz različnih izhodiščnih točk: ta simulator posnema ultrazvočno sondo pri kliničnih snemanjih jajčnikov. Sledi podroben opis novega algoritma za razpoznavanje objektov v statičnih 2D slikah. Razviti algoritem verificiramo na umetnih in realnih ultrazvočnih posnetkih jajčnikov z mešički. Vpeljemo tudi ustrezne kvantitativne mere za vrednotenje uspešnosti algoritma. V začetku 4. poglavja najprej opišemo omejitve algoritma za statične 2D slike pri obdelovanju zaporedja 2D slik, zatem pa matematično formuliramo reševani problem. Upravičimo tudi uporabo Kalmanovega filtra kot predikcijske metode. V nadaljevanju razvijemo izboljšan detekcijski algoritem, pri čemer kot osnovo uporabljamo adaptivne predikcijske postopke (Kalmanov filter). Identificiramo možne probleme, ki se lahko pojavijo pri razpoznavanju objektov iz zaporedja slik, ter ob osnovni varianti algoritma na osnovi metode prediktor-korektor, podamo še tri izboljšave: za primer velikih napak v meritvah, za upoštevanje lokalnosti krivulje in za popravljanje neustreznih meritev. Končno obliko algoritma za razpoznavanje objektov iz zaporedja slik s postopki predikcije podamo v strnjeni obliki psevdokoda. V 5. poglavju podrobno opišemo rezultate, dobljene z algoritmom za razpoznavanje objektov s postopki predikcije. Za testne slike prav tako uporabimo zaporedja umetnih in realnih ultrazvočnih posnetkov jajčnikov z mešički. Rezultate analiziramo ter jih ovrednotimo na osnovi vpeljanih kvantitavnih mer. Zatem jih primerjamo z rezultati, dobljenimi z algoritmom za statične 2D slike. Pokažemo tudi, da so rezultati na osnovi predikcije superiorni. Delo zaključimo s šestim, sklepnim poglavjem, v katerem ovrednotimo spoznanja iz prejšnjih poglavij ter ga primerjalno umestimo med dela s sorodnih področij.

2.

Metode za analizo zaporedij slik

V tem poglavju bomo predstavili razne koncepte in metode za obdelovanje zaporedij slik. Metode bomo smiselno razvrstili glede na tip naloge, ki jo opravljajo, v grobem opisali njihove značilnosti, pri najpomembnejših pa bomo podali tudi psevdokod njihovih algoritmov. Za vsako izmed teh metod bomo poiskali tudi možne aplikacije, ki posamezno metodo, povzeto po literaturi, najbolje okarakterizirajo.

Pri opisovanju se bomo osredotočili predvsem na metode, ki obdelujejo zaporedja 2D digitalnih slik.

2.1 Zaporedje slik

Preden se lotimo pregleda posameznih metod obdelovanja, definirajmo pojem zaporedja digitalnih slik. Zaporedje digitalnih 2D slik $\mathbf{I}$ je zaporedje K digitalnih 2D slik in ga formalno definiramo kot

$$\mathbf{I} = \{I_k, k = 0, ..., K-1\}, \tag{2.1}$$

kjer je I_k k-ta digitalna slika velikosti N x M slikovnih elementov – pikslov. Takšna definicija zaporedja je seveda preveč splošna in s tem ohlapna, saj nismo postavili nikakršne omejitve glede vsebine zaporedja slik niti kakšne spremembe v vsebini so dopustne med dvema zaporednima slikama iz zaporedja. Ob takšni definiciji bi imeli lahko za zaporedje tudi množico popolnoma različnih slik, kar pa seveda ne bi bilo smiselno.

Zato ob definiciji, podani z enačbo (2.1), zahtevamo še, da vsaka slika iz zaporedja popisuje isti opazovani pojav (npr. spremljanje vožnje avtomobila po cesti, ultrazvočni pregled fetusa), vendar ob različnih časovnih trenutkih. Ob tem mora veljati: če je bila k-ta slika iz zaporedja zajeta ob času t_k, potem je bila $k+1$ slika zajeta ob času t_{k+1}, pri čemer je diskretni časovni trenutek t_{k+1} večji od t_k. V realnosti imamo največkrat opravka z zaporedji slik, kjer je časovni interval $\triangle t$ med sosednjimi slikami iz zaporedja fiksen, torej velja $t_k = t_0 + k\triangle t$. Ob takšni definiciji je zaporedje slik časovno spremenljivo in ga lahko zato tudi označujemo kot $I = I(x, y, t)$, kjer sta x in y prostorski koordinati, koordinata t pa označuje čas. To je zvezna predstavitev zaporedja slik. Tako simbol $I(x, y, t_k)$ označuje piksle iz slike, ki je bila zajeta v času t_k.

Kljub tako dopolnjeni definiciji zaporedja slik se v realnosti srečamo z zelo različnimi zaporedji, pri tem ne mislimo le na vsebinsko komponento, ampak z zaporedji, ki izkazujejo popolnoma drugačne značilnosti. Te razlike nastanejo v prvi vrsti zaradi različni tehnik zajemanja digitalnih slik. Ob tem se takoj intuitivno pojavi prva delitev zaporedij 2D digitalni slik in s tem metod obdelave ter analize, in sicer na: a) zaporedja slik vizualne scene ter b) zaporedja medicinskih slik – zaporedja slik prerezov določenih struktur.

Zaporedja slik vizualne scene dobimo tako, da sceno posnamemo s kamero. Primeri zaporedja vizualne scene so npr.: zaporedje, kjer spremljamo pešca, ki vstopa v stavbo, ali pa zaporedje slik, ki jih zajema vojaški helikopter v nizkem letu nad bojiščem, ali pa zaporedje slik, ki jih vrača kamera za nadzorovanje določenega prostora, zaporedje slik iz TV filma itd. V takšnih zaporedjih slik imamo opravka z gibanjem objekta (žoga, ki leti po zraku). Osnovni problemi, ki jih skušamo reševati, so: od najpreprostejšega zaznavanja gibanja, segmentiranja gibajočih se objektov, do določanja posameznih parametrov gibanja gibajočih se objektov. Ob osnovni varianti zajemanja s statično kamero se zgoraj navedeni problemi stopnjujejo, če med zajemanjem slik spreminjamo osvetlitvene pogoje, če so gibajoči se objekti delno/popolnoma zakriti, če so gibajoči se objekti elastična telesa (se lahko ob premikanju tudi deformirajo), če je v zaporedju več gibajočih se objektov in se vsak giblje na svoj način. Ob vsem tem pa lahko imamo tudi opravka z gibajočo se kamero.

Zaporedja slik struktur pa dobimo po snemanju z raznimi prebirniki (npr. ultrazvočne naprave, naprave CT in MRI). Takšna zaporedja, najpogosteje jih najdemo na področju medicine in biologije, so npr.: zaporedje magnetno resonančnih slik (MRI) možganov, ultrazvočno zaporedje slik krčenja in raztezanja levega srčnega prekata, zaporedje slik celic, posnetih z elektronskim mikroskopom, itd. V takšnih zaporedjih nimamo opravka z dejanskim gibanjem objektov, vsaj v večini primerov ne, saj so zaporedne slike ponavadi prerezi skozi določeno strukturo (so rezine). V teh primerih so slike res zajete ob znanih časovnih trenutkih, vendar ker gre za opazovanje statičnih struktur, se ti časovni trenutki pretvorijo v razmike med rezinami. Najdemo tudi zaporedja slik, kjer se struktura s časom spreminja (npr. pri opazovanju srca). Tudi v takšnih primerih ne gre za pravo gibanje, ampak zgolj za deformacije objektov – časovno spremenljive oblike (srce je še vedno v svojem fiziološkem položaju, gre zgolj za spremembe v obliki). Osnovni problemi, ki jih skušamo reševati, so: segmentiranje opazovane strukture, rekonstrukcija v prostoru, v omejenih primerih pa analiza gibanja. Problemi so toliko težji, ker se oblika struktur iz slike v sliko elastično spreminja, v posameznih primerih (npr. klasičen 2D ultrazvok) niso znani koti, pod katerimi so zajeti prerezi, težko je modelirati opazovane strukture. V teh slikah je ponavadi tudi prisoten močan šum (npr. dekoreliran pegasti šum v ultrazvočnih slikah) in tudi ločljivost v slikah je omejena.

Vidimo, da so problemi oz. tipi nalog, ki jih rešujemo v zaporedjih digitalnih slik, jasno opredeljeni z načinom zajemanja slik. To osnovno delitev zaporedij bomo privzeli pri opisu metod obdelovanja. Tako se bomo na začetku lotili predstavitve konceptov in posameznih metod za obdelovanje zaporedij vizualne scene. V podpoglavju 2.2 bomo študirali metode, ki določajo gibanje ter posamezne parametre gibanja, nadaljevali pa z algoritmi, ki na osnovi raznih značilk gibanja izločijo gibajoče se objekte (podpoglavje 2.3). V podpoglavju 2.4 bomo opisali modele aktivnih krivulj, ki so posebej primerne za sledenje elastičnim telesom. Aktivne krivulje so enakomerno zastopane tako v zaporedjih vizualne scene kakor v medicinskih zaporedjih in s tem predstavljajo nekakšen mejnik med metodami obdelovanja oz. analize zaporedij. Zatem bomo predstavili tehnike, ki obdelujejo 3D slike (podpoglavje 2.5). Pregled bomo zaključili z metodami prediktor-korektor (podpoglavje 2.6), ki jih uporabljamo kot nadgradnjo drugih tehnik in lahko v veliki meri izboljšajo natančnost razpoznavanja.

2.2 Analiza gibanja v slikah – optični pretok

Imejmo opravka z zaporedji slik vizualne scene. Želimo odgovor na vprašanje, ali se v zaporedju pojavlja gibanje, in ob tem določiti parametre gibanja objektom. Intuitivno povezujemo gibanje s spremembami. Zanima nas predvsem vizualna informacija, ki jo lahko izvlečemo iz prostorskih in časovnih sprememb, ki se pojavljajo v zaporedju.

Če predpostavimo, da se osvetlitveni pogoji med zajemanjem slik ne spreminjajo, potem so vse spremembe v sliki povzročene zaradi relativnih premikov med kamero in sceno, pri tem se lahko kamera giblje proti statični sceni ali se del scene giblje glede na statično kamero, v splošnem pa se lahko gibljejo tako kamera kakor objekti.

Na začetku opazujemo razlike med dvema slikama iz zaporedja. Pojavi se problem korespondence, ki zahteva opredelitev, kateri elementi iz začetne slike ustrezajo določenemu objektu iz naslednjega okvira zaporedja. Če poznamo odgovor na to vprašanje, lahko dokaj preprosto določimo parametre 2D gibanja za objekte (premik iz slike v sliko), medtem ko v splošnem 3D gibanja in strukture opazovanega sveta brez dodatnih informacij ne moremo določiti [TV98].

Ko se giblje predmet pred biološkim oz. umetnim očesom (kamero), definira 3D polje hitrosti. Zaradi naprave za zajem slik se 3D polje hitrosti v slikovni ravnini oz. digitalni sliki transformira v 2D polje gibanja $\mathbf{v} = [v_x, v_y]$ [dMUT90]. Informacija, ki jo imamo na voljo, ni 2D polje gibanja $\mathbf{v}$, ampak skalarno polje $I(x, y, t)$, ki določa svetlost v lokaciji

(x, y) slikovne ravnine ob času t.

Optični pretok (*optical flow*) označuje 2D vektorsko polje $\mathbf{u}$, ki ga dobimo iz $I(x, y, t)$ in je zelo blizu dejanskemu 2D polju gibanja $\mathbf{v}$. Ker gre za aproksimacijo, je očitno, da obstaja mnogo različnih optičnih pretokov, ki imajo svoje lastnosti in karakteristike, določene glede na algoritem, po katerem se izračunavajo, ter glede na kriterije konvergence [dMUT90]. V primeru privzetih naslednjih predpostavk: a) objekti so Lambertove površine (točka na površini objekta je enako svetla iz vseh smeri opazovanja), b) točkasti izvor svetlobe je v neskončnosti, c) ni fotometričnih motenj, je napaka aproksimacije polja gibanja z optičnim pretokom majhna v točkah z velikim prostorskim gradientom, oziroma točno 0 za translacijsko gibanje ali za togo gibanje, kjer je smer osvetlitve paralelna kotni hitrosti.

Obstajata dve strategiji za reševanje problema korespondence oz. določanja vektorjev razmika (*displacement*) iz zaporednih slik [J93, TV98]:

1. Diferencialne tehnike – osnovna ideja metode je, da gibanje povzroča spremembe v sivinah. Rezultat so vektorji razmika, ki so izračunani za vsak piksel v sliki. Pri izračunih se uporabljajo približki časovnih odvodov, zato mora biti časovni interval med zajemanjem zaporedja slik zelo majhen (visoka frekvenca vzorčenja).

2. Korelacijske tehnike – rezultat so vektorji razmika, ki so določeni le za neko podmnožico pikslov. Tukaj najprej izberemo iz slik značilke, za katere lahko pričakujemo nedvoumno določitev vektorjev razmika. Ta pristop izključi vse vprašljive slikovne regije že od začetka. V drugem koraku pa skušamo poiskati pripadajoče izbrane značilke v naslednjih slikah. Fleksibilnost in uspeh te metode je v ustrezni izbiri pravilnih značilk. Če le-te niso odvisne od osvetlitvenih pogojev zajemanja, dobimo robustno ocenitev vektorjev razmika.

Oba pristopa gledata na problem analize gibanja z različnih zornih kotov, nobena od njiju pa ne obravnava zadostno celotno kompleksnost problema [J93].

2.2.1 Diferencialne tehnike

Diferencialne tehnike za izračunavanje optičnega pretoka temeljijo na zveznosti optičnega pretoka – problem rešujejo s t. i. enačbo konstantne svetlosti v sliki. V večini primerov ostane svetlost gibajočega se objekta konstantna. Konstantnost svetlosti opazovane scene zapišemo kot stacionarno svetlost v sliki I skozi čas [TV98, J93]:

$$\frac{dI}{dt} = 0. \tag{2.2}$$

Zaporedje slik I je funkcija časa, $I = I(x, y, t)$, in ker sta tudi x in y funkciji časa t, potem lahko enačbo (2.2) razvijemo v

$$\frac{dI(x(t), y(t), t)}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{dy}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0, \tag{2.3}$$

oz. zapisano v strnjeni notaciji

$$(\nabla I)^T \cdot \mathbf{v} + I_t = 0, \tag{2.4}$$

kjer ∇I označuje prostorski gradient, $\mathbf{v}$ je polje gibanja, I_t pa označuje parcialni slikovni odvod po času.

V realnih razmerah nikoli ne operiramo z dejanskim 2D poljem gibanja, ampak zgolj z njegovimi približki. Optični pretok je 2D vektorsko polje, ki zadosti enačbi (2.4) po $\mathbf{v}$ in je "ohlapno" definirano kot navidezno gibanje vzorca svetlosti v sliki [TV98].

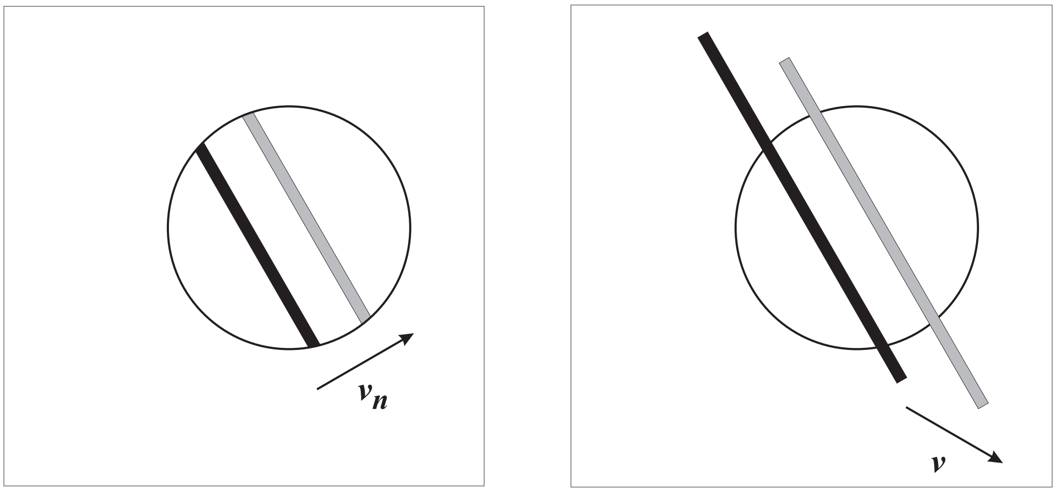
Zveznost optičnega pretoka je v veliki meri odvisna od pogojev osvetljevanja in odbojnih lastnosti opazovanih objektov. V primeru, ko imamo opravka samo z enim togim in hkrati relativnim gibanjem med kamero ter opazovano sceno in če se osvetlitveni pogoji ne spreminjajo, potem je pogoj konstantne svetlosti izpolnjen. Ta predpostavka je porušena pri zaporedju iz nogometne tekme, scene iz prometa, polne ulice, velja pa pri opazovanju zgradb, ki jih gleda gibajoči se opazovalec. Predpostavka tudi ne velja za elastična telesa, kot sta npr. obleka ali gibajoči se človek [TV98].

Iz tega lahko hitro vidimo, da ne moremo vzpostaviti splošno veljavne enačbe zveznosti svetlosti. Če npr. ima objekt lastnost Lambertove površine in če je izvor svetlosti točkast ter postavljen v izhodišče koordinatnega sistema, potem dobi osnovna enačba (2.4) še dva dodatna člena. Določitev hitrosti (pretoka $\mathbf{u}$) je najbolj zanesljiva za strme robove sivin, medtem ko je lahko bistveno popačena v regijah z majhnimi gradienti sivin. Torej, zelo je pomembno zagotoviti ustrezno osvetljenost scene [J93].

Ponavadi določamo optični pretok z lokalnimi operatorji. Takšen operator pa vidi le majhen del opazovanega objekta. Pridemo do t. i. problema reže (*aperture problem*). Natančno lahko določimo le komponento pretoka, ki je normalna na rob, medtem ko komponenta, ki je vzporedna robu, ostane neznana. Enačba (2.4) ne pogojuje komponente polja gibanja v smeri, ki je pravokotna na prostorski slikovni gradient. Na sliki 2.1 je shematsko prikazan problem reže. Natančno lahko določimo le komponento, ki je normalna na rob. Premik iz slike v sliko pa lahko enoumno določimo, če je znotraj maske operatorja vogal (kot) objekta [TV98, J93].

Tehnike za izračunavanje optičnega pretoka

Obstaja več različnih tehnik za določanje optičnega pretoka glede na enačbo (2.4) [TV98]:



Slika 2.1: Problem reže: črne in sive linije prikazujejo dva položaja iste linije v 2 zaporednih slikah. Hitrost $\mathbf{v_n}$, ki jo skozi majhno režo zaznamo na levi sliki, je zgolj vzporedna komponenta slikovnega gradienta prave hitrosti $\mathbf{v}$, prikazane na desni sliki.

a) rešitev na osnovi velikega sistema parcialnih diferencialnih enačb,

b) rešitev po izračunu drugih in višjih odvodov svetlosti v sliki,

c) ocenjevanje optičnega pretoka po metodi najmanjših kvadratov.

Prednosti metod c) glede na ostali dve vrsti so v tem, da so metode c) lokalne, niso iterativne in so manj pristrane kot iterativne metode, ki lahko imajo posamezne nezveznosti v polju gibanja. Hkrati pa tudi ne vsebujejo odvodov višjih redov ter so zato manj občutljive na šum.

Poglejmo si rešitev enačbe (2.4) po metodi c). Za eksaktno rešitev potrebujemo le dva piksla $\mathbf{p}_1$ in $\mathbf{p}_2$. Dobimo naslednji sistem

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \end{bmatrix}}_{\text{matrika } \mathbf{A}} + \begin{bmatrix} u_x \\ u_y \end{bmatrix} = -\underbrace{\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \end{bmatrix}}_{\text{vektor } \mathbf{b}}, \tag{2.5}$$

kjer je $\mathbf{u}$ optični pretok, $\mathbf{u} = [u_x, u_y]^T$, I_x in I_y pa označujeta parcialna odvoda po x oz. y. Formalno dobimo rešitev z invertiranjem matrik, ki vsebujejo parcialne odvode. Rešitev obstaja, če determinanta matrike ni enaka 0. To je tedaj, če sta gradienta v obeh točkah različna od 0 in nista orientirana v isto smer. Pri eksaktni rešitvi ne moremo delati analize napake, zato raje pri izračunu uporabljamo predeterminiran sistem enačb. To pomeni, da v izračunu upoštevamo L pikslov iz okolice (piksli $\mathbf{p}_i$). Sistem je sedaj oblike

$$\underbrace{\mathbf{A}}_{L\mathrm{x}2}\underbrace{\mathbf{u}}_{2\mathrm{x}1} = \underbrace{-\mathbf{b}}_{L\mathrm{x}1}, \tag{2.6}$$

rešitev pa dobimo po enačbi (2.7)

$$\mathbf{u} = -\mathbf{A}^{-g}\mathbf{b}, \tag{2.7}$$

kjer je $\mathbf{A}^{-g}$ psevdo inverz, ki se določi kot $\mathbf{A}^{-g} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T$. Matrika $\mathbf{A}$ ima sedaj L vrstic. Vsaka vrstica pomeni gradient za posamezen piksel $\mathbf{p}_i$. Rešitev enačbe (2.7) obstaja le, če lahko invertiramo matriko

$$(\mathbf{A}^T \cdot \mathbf{A}) = \begin{bmatrix} \sum_i I_x^2(\mathbf{p}_i) & \sum_i I_x(\mathbf{p}_i)I_y(\mathbf{p}_i) \\ \sum_i I_x(\mathbf{p}_i)I_y(\mathbf{p}_i) & \sum_i I_y^2(\mathbf{p}_i) \end{bmatrix}.$$

Invertiranje pa lahko izvedemo, če sta izpolnjena dva pogoja:

- vsi odvodi I_x in I_y ne smejo biti 0, kar pomeni, da okolica ne sme biti sestavljena iz področja s konstantno sivino;
- gradienti ne smejo kazati v isto smer.

Napaka pri računanju optičnega pretoka bo najmanjša, če bodo navzkrižni korelacijski elementi $\sum I_x I_y$ poniknili. To se zgodi, kadar so gradienti enakomerno porazdeljeni v vse smeri. Napaka bo majhna tudi če je: a) povprečen kvadriran gradient velik, b) okolica piksla velika, c) varianca v podatkih majhna [J93].

Podajmo algoritem za računanje optičnega pretoka v strnjeni obliki [TV98]:

VHOD: Časovno spremenljivo zaporedje slik $\mathbf{I}$. *Naj bo* Q *kvadratna regija iz* L *x* L *pikslov (tipično* $L = 5$*).*

IZHOD: Optični pretok $\mathbf{u}$.

1. *Filtriraj vsako sliko iz zaporedja z Gaussovim filtrom s standarnim odklonom* σ_S *vzdolž vsake prostorske dimenzije (tipično* σ_S *=1,5 piksla).*
2. *Filtriraj vsako sliko iz zaporedja vzdolž časovne dimenzije z Gaussovim filtrom s* σ_T *(tipično* σ_T *=1,5 okvira).*
3. *Za vsak piksel v vsaki sliki zaporedja:*

a. *Izračunaj matriko* $\mathbf{A}$ *in vektor* $\mathbf{b}$ – *enačbi (2.5) in (2.6). Množico pikslov* $\mathbf{p}_i$ *vzemi iz* Q*-okolice trenutnega piksla.*

b. *Določi optični pretok po enačbi (2.7).*

Algoritem 1 Algoritem za izračun optičnega pretoka po metodi najmanjših kvadratov.

Algoritem lahko izboljšamo, če vsaki točki znotraj okolice Q priredimo svojo utež w_i. Točke blizu središča imajo večje uteži.

Kot smo že uvodoma omenili, je možno optični pretok izračunati tudi na druge načine. Avtor Otte [ON95] je predstavil cenilko (*estimator*) za optični pretok, ki spada v skupino metod b). Ta v svojem izračunu uporablja tudi odvode četrte stopnje slike sivin. Vendar so ob tem definirali tudi nova jedra za Gaussov filter, ki upoštevajo efekt skrajšanja (*truncation*) in vzorčenja. To je še posebej pomembno, ker lahko slabi operatorji za ocenjevanje višjih odvodov hitro izničijo metodološke izboljšave novega pristopa.

V [NSKO94] avtorji ob osnovni enačbi konstantne svetlosti (enačba (2.2)) upoštevajo še, da so parcialni odvodi v lokalni okolici medsebojno odvisni. Prednost tega veliko bolj rigoroznega pristopa je, da lahko vpeljemo eksaktne stohastične teste pri določanju nezveznosti v polju optičnega pretoka.

Nekateri avtorji pa za izračun optičnega pretoka namesto kriterija konstantne svetlosti (enačba (2.2)), uporabljajo predpostavko, da je optični pretok lokalno konstanten [dMUT90]. To formuliramo matematično kot

$$\frac{d}{dt}\nabla I = 0. \tag{2.8}$$

Z metodo najmanjših kvadratov določimo optični pretok $\mathbf{u}$ kot

$$\mathbf{u} = -\mathbf{A}^{-1} \cdot \frac{\partial}{\partial t}\nabla I, \tag{2.9}$$

kjer je matrika $\mathbf{A}$ Hesseva matrika za I, $\mathbf{A} = \left[\frac{\partial^2 I}{\partial x_i \partial x_j}\right]$, parametra x_i in x_j pa označujeta prostorski koordinati slike.

Z analiziranjem matrike $\mathbf{A}^{-1}$ lahko ocenimo, koliko je izračunan optični pretok $\mathbf{u}$ blizu dejanskega polja gibanja $\mathbf{v}$. Vektorsko polje $\mathbf{u}$ se bo bližalo polju $\mathbf{v}$, če bodo vstopi v matriko $\mathbf{A}^{-1}$ majhni. Ta pogoj, ki hkrati zagotavlja tudi numerično stabilnost metode, je izpolnjen, če je determinanta za $\mathbf{A}$ velika in pogojno število c_A matrike $\mathbf{A}$ blizu 1. Matrika $\mathbf{A}$ je simetrična, zato je $c_A = |\lambda_1/\lambda_2|$, kjer sta λ_1 in λ_2 dve realni lastni vrednosti za $\mathbf{A}$.

Tistarelli je študiral [Tis96], kako obe omejitvi, konstantna svetlost (enačba (2.2)) in lokalna konstantnost pretoka (enačba (2.8)), če ju upoštevamo hkrati, vplivata na izračun optičnega pretoka **u**. V tem primeru se **u** izračuna z rešitvijo predeterminiranega sistema enačb, in sicer tri enačbe dve neznanki. Avtor predlaga, da namesto upoštevanja vseh treh enačb naenkrat raje obravnavamo tri pare enačb posebej in ob tem študiramo stabilnost. Celoten problem razpade na analizo treh matrik:

$$\mathbf{A}_1 = \begin{bmatrix} I_x & I_y \\ I_{xx} & I_{xy} \end{bmatrix},\ \mathbf{A}_2 = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix},\ \mathbf{A}_3 = \begin{bmatrix} I_x & I_y \\ I_{yx} & I_{yy} \end{bmatrix}. \tag{2.10}$$

Pri reševanju imamo dve alternativi:

1. Za pravilno rešitev vzamemo najbolj zanesljivo presečišče v prostoru hitrosti. Optični pretok **u** določimo kot

$$\mathbf{u} = \mathbf{A}_i^{-1} \cdot \mathbf{b}_i \Leftrightarrow det(\mathbf{A}_i) > \{det(\mathbf{A}_j), det(\mathbf{A}_k)\}, i, j, k \in \{1, 2, 3\} \wedge i \neq j \neq k$$
$$\mathbf{b}_1 = \begin{bmatrix} I_t \\ I_{xt} \end{bmatrix} \qquad \mathbf{b}_2 = \begin{bmatrix} I_{xt} \\ I_{yt} \end{bmatrix} \qquad \mathbf{b}_3 = \begin{bmatrix} I_t \\ I_{yt} \end{bmatrix}.$$

2. Optični pretok določimo po naslednjem postopku:
   - uporabi par enačb z najvišjima determinantama,
   - zatem izračunaj psevdo inverz,
   - določi povprečje vektorjev pretoka.

Postopka dajeta skoraj identične rezultate, le da je gostota izračunanega polja različna. Polje je redko na mestih, kjer optičnega pretoka ni možno določiti.

2.2.2 Korelacijske tehnike

Videli smo, da je zveznost optičnega pretoka veljavna le v zelo omejenih primerih. Nekateri avtorji [J93, SHB94] zato predlagajo, da je treba v sliki najprej poiskati regije, kjer lahko pričakujemo uspešno ocenitev vektorjev razmika. Takšen pristop sestoji iz dveh korakov: 1) poiskati primerne značilke, ki so neobčutljive na spremembe v svetlosti, 2) zatem pa poiskati pripadajoče točke v dveh sosednjih slikah ter določiti premik. Rezultat teh metod je redko polje gibanja – polje je določeno le v značilkah.

Podobnost med značilkami v zaporednih slikah določamo z mero podobnosti. Takšna mera mora biti robustna na spremembe v osvetlitvi. Za diskretne slike je definiranih več različnih mer podobnosti S [ADPM, J93]:

1. Razlika
$$S(m,n;k,l) = \sum\sum_{p,q\in Q} \Big| I_{s+1}(m+p+k, n+q+l) - I_s(m+p, n+q) \Big|$$

2. Kvadrirana razlika

3. Navzkrižni korelacijski koeficient
$$S^2(m,n;k,l) = \frac{\left(\sum\sum_{p,q\in Q} I_{s+1}(m+p+k,n+q+l) I_s(m+p,n+q)\right)^2}{\sum\sum_{p,q\in Q} I_s^2(m+p,n+q) \sum\sum_{p,q\in Q} I_{s+1}^2(m+p+k,n+q+l)}$$

4. Korelacijski koeficient
$$S^2(m,n;k,l) = \frac{\left(\sum\sum_{p,q\in Q}(I_{s+1}(m+p+k,n+q+l) - \overline{I}_{s+1})(I_s(m+p,n+q) - \overline{I}_s)\right)^2}{\sum\sum_{p,q\in Q}(I_s(m+p,n+q) - \overline{I}_s)^2 \sum\sum_{p,q\in Q}(I_{s+1}(m+p+k,n+q+l) - \overline{I}_{s+1})^2}.$$

Indeksa m in n označujeta položaj v sliki, medtem ko k in l tečeta preko iskalnega območja Q za maksimalnim koeficientom S pri položaju (m,n). Vektor (k,l) dejansko definira premik med slikama.

Obstajata dve hitri implementaciji teh tehnik [J93, SLQ90]: a) direktno iskanje maksimuma, kjer te koeficiente aproksimiramo z razvojem v Taylorjevo vrsto drugega reda, b) grobo-v-fino iskanje na piramidalni sliki (Gaussova ali Laplaceova piramida).

V delu [SLQ90] so predstavili algoritem, ki temelji na t. i. multiresolucijski predstavitvi slike – slike predstavljene z Laplaceovo piramido. Podobnost oz. lokalno korelacijo med dvema slikama lahko v splošnem formuliramo kot

$$S(m,n,k,l) = A_1[F[I_s(m,n)] \cdot F[I_{s+1}(m+k,n+l)]]. \tag{2.11}$$

Mera podobnosti oz. korelacija S se izračuna med lokalnim vzorcem v sliki I_s, centriranim v pikslu (m,n), in lokalnim vzorcem v sliki I_{s+1}, centiranem v pikslu $(m+k,n+l)$. Operator $F[]$ je filter, ki ga uporabimo v fazi predobdelave. Operator $\cdot$ je lahko poljuben operator za primerjanje vrednosti pikslov. Funkcija A_1 se uporablja za izračun povprečnih vrednosti in definira proces hierarhične diskretne korelacije. Mero podobnosti S lahko obravnavamo tudi kot stopnjo zaupanja za ujemanje med lokalnimi vzorci. V prejšnjem naštevanju smo podali primere takšnih mer podobnosti. V [SLQ90] pa vpeljujejo modificirano korelacijsko mero S kot

$$S(m,n,k,l) = \frac{A_1[L_{s,r}(m,n) \cdot L_{s+1,r}(m+k,n+l)]}{\sqrt{A_1[L_{s,r}^2(m,n)]}\sqrt{A_1[L_{s+1,r}^2(m+k,n+l)]}}, \tag{2.12}$$

kjer $L_{s,r}$ in $L_{s+1,r}$ predstavljata r-ti nivo Laplaceove piramide za sliko I_s oz. I_{s+1}. S takšno implementacijo zmanjšamo veliko časovno zahtevnost, ki je potrebna pri analizi zaporedja slik s korelacijskimi tehnikami.

Operatorji za določanje značilk

Eden izmed najpomembnejših faktorjev za uspešnost korelacijskih tehnik je določitev ustreznih značilk v sliki. V literaturi najdemo celo vrsto operatorjev in algoritmov za določanje značilk, mi si jih bomo ogledali le nekaj.

Monotoni operator [J93] je lokalni operator za izbiro značilk, posebej primernih pri analizi gibanja v zaporedjih slik. Je skoraj popolnoma neobčutljiv na spremembe v osvetljevanju scene. Monotoni operator izračunava po naslednjem postopku: najprej postavimo vse piksle znotraj maske operatorja na ena, če je njihova sivina manjša od vrednosti centralnega piksla, sicer pa na 0, zatem pa le še preštejemo enice v maski. Prvi korak formalno zapišemo kot

$$c(m,n) = \begin{cases} 1, & I(m,n) > I(m+k,n+l) \\ 0 & \text{sicer} \end{cases} \qquad \forall (k,l) \in Q,$$

kjer Q označuje masko. V drugem koraku pa monotoni operator Mon prešteje enice znotraj maske

$$Mon(m,n) = \sum \sum_{(k,l) \in Q} c(k,l).$$

Pri maski velikosti 3 x 3 piksle preslika monotoni operator strukturo sivin na interval $[0,8]$, in sicer lokalni maksimum v 8, linije v 6, vogale (kote) v 5 ter robove v 3. Ocenimo še lastnosti monotonega operatorja:

+ preprost izračun, potrebne le primerjave,
+ rezultat ni odvisen od absolutnih vrednosti sivin in kontrastov – rezultat je enak po transformaciji sivin, ki ohranjajo monotonost sivin,
+ dobljene značilke bistveno pospešijo algoritem iskanja ujemanja, saj je treba iskati le med značilkami istega razreda,
- transformacija sivin v vrednosti ni nedvoumna, saj po permutaciji pikslov znotraj maske (razen centra) dobimo enak rezultat,
- ne razlikuje med lokalnim minimumom, področji s konstantno sivino in dolinami vrednosti sivin – te strukture z monotonim operatorjem izginejo.

Zelo koristen operator za določanje značilk je t. i. predznak Laplaceovega operatorja ($sgn(Laplace)$) [J93].

V [WB95a] je predstavljen še en hiter algoritem za detektiranje značilk, in sicer vogalov. Vogali so dejansko lastnosti drugega reda površine (odvodi). Znano pa je, da so višji odvodi zelo občutljivi na šum. Algoritem temelji na opazovanju skupne ukrivljenosti slike, ki je proporcionalna odvodu drugega reda v smeri tangente na rob ter obratno proporcionalna jakosti roba (velikost normale roba). Določanje značilk poteka tako, da najprej naredimo razliko med drugim odvodom v smeri tangente in jakostjo roba, zatem pa poiščemo maksimume. Vogal ohranimo, če je maksimum večji od prednastavljenega praga.

Sledenje objektom

Zelo hitro lahko opazimo, da s korelacijskimi tehnikami za izračun optičnega pretoka dejansko sledimo značilkam skozi zaporedje. Ko določimo ujemanje za značilko med slikama, lahko točno izračunamo vektor premika. Torej, za vsako značilko točno vemo, kje se nahaja v kateri sliki – ji lahko sledimo.

Sledenje dobi smisel, če analiziramo daljše zaporedje slik, ne le par okvirov. Problem vzpostavljanja ujemanja med značilkami skozi zaporedje slik je zelo težek problem. Če predpostavimo, da imamo v zaporedju K slik, v vsaki sliki pa L značilk, potem je možnih $L!^{(K-1)}$ različnih ujemanj točk skozi zaporedje (množic trajektorij). Le ena izmed te množice trajektorij je pravilna [Meh98].

V realnosti ni možno pregledati vseh možnih kombinacij značilk, zato se pri iskanju dodajo razne omejitve. V [Meh98] je predstavljen nov pristop za določanje trajektorij. Trajektorija se lahko začne ali zaključi v kateri koli sliki iz zaporedja in ne le na koncu ali začetku. Vpeljali so t. i. omejitev uniformnosti gibanja – spremembe v gibanju gibajočih se točk skozi kratek časovni interval so majhne in časovno zvezne. Rezultat takšne omejitve je, da je trajektorija skozi čas gladka, kot je le mogoče, ter da je dolžina, ki jo "prehodi" točka, majhna, kot je le možno.

Če je gibanje opazovane scene zvezno, kar je skoraj vedno v realnosti, potem lahko predvidimo gibanje na osnovi njene predhodne trajektorije, kot smo videli v prejšnjem primeru. Primerno orodje je tudi Kalmanov filter (podpoglavje 2.6), s katerim lahko ocenimo položaj in nejasnost gibajoče se značilke v naslednji sliki. To pomeni, da lahko ocenimo, kje iskati značilke in kako veliko regijo naj preiskujemo v naslednji sliki okoli predvidenega položaja.

Poseben problem pa predstavlja sledenje elastičnim objektom s korelacijskimi tehnikami. V [HNR92, HNR93] je nazorno prikazano, kako se lotiti takšnega problema. Rešujejo problem sledenja več elastičnim telesom v kompleksni sceni. Uporabljajo t. i. modelni pristop, kjer najprej izločijo 2D modele iz slikovnih podatkov, zatem pa iščejo ujemanje v naslednjih slikah. Model objekta se zajema dinamično iz zaporedja. Glavna ideja je v tem, da se lahko 2D slika gibajočega se predmeta v 3D prostoru razcepi na dva dela, in sicer: a) na spremembo oblike objekta in b) na 2D gibanje. Ob tem predpostavijo, da se 2D oblika objekta ne spreminja veliko med dvema slikama, za gibanje pa ni omejitev. Objekt je predstavljen kot zaporedje binarnih podslik, ki ustrezajo vsakemu okvirju iz vhodnega binarnega zaporedja slik. Sledimo tako, da model M_k v sliki I_k primerjamo z naslednjo sliko I_{k+1}. Pri tem iščemo transformacijo, ki določa najboljši položaj tega modela v I_{k+1}. Nato kreiramo nov model M_{k+1} iz podmnožice točk slike I_{k+1}, ki so blizu transformiranega modela M_k. Model M_{k+1} predstavlja obliko objekta v I_{k+1}. Za primerjavo 2D oblik uporabimo Hausdorffovo razdaljo. Izmed vseh možnih vrst gibanja je dovoljena le translacija, vse ostalo gibanje (npr. rotacija) se obravnava kot sprememba v obliki.

Opisan postopek je bil uspešno uporabljen pri sledenju gibanja človeka, tudi v primerih, ko je bil človek delno oz. popolnoma zakrit v nekaj zaporednih slikah.

V [DRM, ADPM] so v 3D časovno spremenljivih slikah (zaporedju 3D slik) analizirali gibanje celic. 3D sledenje celicam je izrednega pomena za področje biologije. Razvitih je bilo že kar nekaj pristopov, npr. metoda najbližjega soseda, ki poveže trenutno celico z najbližjo celico v naslednjem trenutku. Ta postopek je manj uspešen, če so objekti zelo blizu drug drugega in če je časovni razmik med posnetki velik. Zato večina preostalih metod poskuša slediti na osnovi statistik ujemanja. Najboljše ujemanje teži k ohranitvi nekaterih kombinacij značilk objekta: velikosti, oblike, intenzivnosti in/ali pričakovanega premika.

Razvili so nov algoritem za avtomatsko 3D sledenje. Ta na začetku uporablja več pragov za identificiranje različnih celic v začetnem časovnem trenutku. Zatem pa uporabijo korelacijsko tehniko pri sledenju celic iz enega časovnega trenutka v naslednjega. Pri 3D korelacijskih tehnikah se časovna kompleksnost povzpne na $\mathcal{O}(p^3 \times k^3)$, kjer je p^3 velikost šablone, $p + k$ pa je ena stran volumna, ki ga moramo preiskati.

Zaradi izjemne časovne zahtevnosti so algoritem implementirali na porazdeljenem računalniškem sistemu.

Kot smo lahko videli, obstajajo tudi mnoge aplikacije, ki so uspešno uporabile metode sledenja na zaporedju medicinskih slik. Tako so v [KSM95] in [YLFP98] ocenjevali gibanje pegastega šuma v ultrazvočnih slikah. V [KSM95] so šum modelirali, samo gibanje pa so ocenjevali z iskanjem ujemanja med bloki. Razvili so optimalno mero podobnosti za ultrazvočne slike. Za blok v trenutni sliki iščejo ujemanje v predhodni sliki, t. i. vzvratna predikcija. V [YLFP98] pa gibanje ocenjujejo s t. i. spremenljivimi mrežami, ki jih vodijo s postopki metod končnih elementov.

2.2.3 Polje vektorja premika

V podpoglavjih 2.2.1 in 2.2.2 smo spoznali tehnike za določanje vektorja premika za eno točko v sliki. Ponavadi pa nas zanima gibanje objekta kot celote, ne le nekaj njegovih značilk. Torej, želimo določiti polje vektorja premika – DVF (*displacement vector field*). Za dosego tega cilja pa moramo zbrati razpršene informacije o hitrosti (premiku), ki smo jih dobili z lokalnimi operacijami, omenjenimi v prejšnjih podpoglavjih. Lokalne operacije ne dajejo nikakršnih informacij o gibanju v točkah, kjer vektorjev premika nismo mogli izračunati (področja konstantnih sivin itd.), niti o tem, katere točke pripadajo istemu objektu. Določitev DVF iz enega vektorja premika ni preprost interpolacijski problem [J93]:

- v splošnem lahko v eni točki določimo le eno komponento vektorja premika (problem reže – glej podpoglavje 2.2.1), saj so robovi v slikah veliko bolj pogosti kot vogali oz. lokalni ekstremi;

- v splošnem scena vsebuje več kot en gibajoči se objekt. Polje hitrosti je v takšnih primerih nezvezno ob robovih objektov. Velike razlike v sivinah se pojavljajo ob robovih teh objektov, zato dobimo dobro oceno za vektor premika, vendar na žalost ne moremo predpostaviti, da se DVF razširja zvezno na obe strani robov.

Pri reševanju interpolacijskega problema moramo na optimalen način upoštevati dve nasprotujoči si zahtevi [J93]: a) DVF se mora približati vrednostim vektorja premika v točkah, kjer ga je bilo možno določiti z zadostno verjetnostjo, b) treba je upoštevati kinematiko in dinamiko gibanja.

Če pretehtamo vse težave pri določanju DVF, potem se je tehtno vprašati, ali je smiselno računati DVF in ga potem ekstrapolirati v regije, kjer ne moremo določiti informacij o hitrosti. Morebiti bi bilo primerneje počakati z izračunom DVF v teh zgodnjih fazah analize gibanja ter ga izračunati šele potem, ko imamo na voljo več informacij, npr. razpoznane individualne objekte v sceni.

2.3 Segmentacijske metode na osnovi gibanja

Kadar analiziramo gibanje v vizualni sceni, nas zanimajo informacije o gibanju objektov kot celote in ne le podatki o hitrosti za posamezne točke. Dejansko želimo sceno razbiti na posamezne gibajoče se predmete – sceno želimo segmentirati. V tem podpoglavju bomo opisali nekaj metod, ki segmentirajo sliko na osnovi informacij o gibanju.

2.3.1 Uvod

Pri analiziranju zaporedja digitalnih slik nas zanimajo predvsem objekti, ki se skozi zaporedje gibljejo. Srečujemo se s problemom, kako v slikah poiskati regije, ki ustrezajo različnim gibajočim se objektom. V [SHB94, Rus95, Cas96, J93, Pot98] so opisane razne metode za segmentiranje statičnih 2D digitalnih slik. Te metode razbijejo posamezno sliko na vse možne regije – objekte, ki jih najdemo v sliki, in nikjer ne upoštevajo, ali se objekti gibljejo ali ne. Takšne segmentacijske metode za 2D slike zato v svoji osnovni izvedbi niso primerne za analiziranje scen z gibajočimi se objekti in jih je treba dopolniti z informacijami o gibanju. Vhod v metode za segmentiranje statičnih 2D slik je največkrat slika sivin, ki jo zatem po nekakšnem postopku razbijemo na disjunktne regije. Pri segmentiranju zaporedja 2D slik pa mora vhodna slika v segmentacijsko metodo zajemati informacije o gibanju. Sliko sivin iz zaporedja moramo zato transformirati v neko novo sliko, ki za vsak piksel hrani vrednost mere gibanja. V literaturi najdemo celo vrsto takšnih transformacij [TV98]. Šele ko imamo formirano takšno novo sliko, lahko nad njo uporabimo katero izmed znanih metod za segmentiranje slik (npr. pragovno operacijo, segmentacijske metode na osnovi robov, metode na osnovi regij). Izbira ustrezne segmentacijske metode je seveda odvisna od tipa vhodne slike [Pot98].

Poglejmo si primer, ko je zaporedje slik zajeto s fiksno kamero, v sceni pa lahko imamo opravka z več gibanji (objekti). Potem lahko na problem segmentiranja gibajočih se objektov gledamo kot na problem zaznavanja sprememb v primeri s konstantnim ozadjem. Vsak piksel iz slike zaporedja moramo dejansko klasificirati ali kot gibajoči ali kot konstanten glede na kamero. Za uspešno klasifikacijo pa potrebujemo ustrezno mero, ki bo temeljila na gibanju.

V nadaljevanju bomo predstavili nekaj takšnih mer ter opisali možne postopke za segmentacijo s pomočjo informacij o gibanju.

2.3.2 Segmentacijske metode in mere gibanja

Slike razlik

Najpreprostejša strategija za detektiranje sprememb v zaporedju je odštevanje slik – slika razlik. Sliko razlik $\triangle I_k$ med dvema zaporednima slikama I_k in I_{k+1} določimo tako, da odštejemo istoležeče piksle po naslednji formuli

$$\triangle I_k(m,n) = I_{k+1}(m,n) - I_k(m,n). \tag{2.13}$$

Ko imamo sliko razlik določeno, lahko nad njo uporabimo katerokoli segmentacijsko metodo za statične 2D digitalne slike. Ta trditev seveda velja tudi za vse preostale mere, ki jih bomo še opisali.

Sliko razlik najpreprosteje segmentiramo s pragovno operacijo. Piksel (m, n) označimo za "gibajoči se", če absolutna vrednost v sliki razlik preseže pozitiven prag T, oz. zapisano matematično $|\triangle I_k(m, n)| > T$. Prag T moramo določiti tako, da je verjetnost napačne klasifikacije minimalna [TV98]. Preprost način je, da zajamemo dve sliki statične scene, kjer ni sprememb v osvetlitvi, ter opazujemo histogram njune slike razlik. Histogram izgleda kot Gaussova funkcija s povprečno vrednostjo 0. Prag lahko ocenimo kot mnogokratnik standardnega odklona izračunane porazdelitve.

Slike razlik so seveda zelo preprosto merilo in dajejo, razen pri zaporedjih, zajetih s statično kamero, slabe rezultate. Bolj sofisticirana mera je npr. obteženo povprečje velikosti normalnega pretoka [TV98], ki se določi kot $|I_t|/\|\nabla I\|$, izračunano na lokalni okolici Q_i piksla $\mathbf{p}_i$. Uteži določimo kot kvadrat norme prostorskega gradienta slike. Končna oblika te mere je

$$M = \frac{|I_t|\|\nabla I\|}{C + \sum\sum_{Q_i} \|\nabla I\|^2},$$

kjer vsota teče po okolici Q_i, časovni in prostorski odvodi se izračunajo pri vsaki točki iz Q_i, parameter C pa je konstanta, ki preprečuje nestabilnosti metode v okolicah z uniformnimi sivinami.

Statično ozadje

Alternativa sliki razlik je t. i. koncept statičnega ozadja. Ozadje je najpogosteje najbolj stabilen del slike. V kontroliranem okolju je možno dobiti sliko ozadja brez gibajočih se objektov. V realnih primerih pa je ozadje večinoma vedno prekrito z gibajočimi se objekti, zato so potrebne tehnike za regeneriranje ozadja. Takšen sistem za generiranje ozadja poizkuša poiskati vse stacionarne dele v sliki. Obstaja več tehnik za tvorjenje statičnega ozadja [LY90]. Najpreprostejše tehnike, kot je npr. povprečenje celotnega zaporedja slik, dajejo v zahtevnejših situacijah slabe rezultate. Metoda detektiranja robov opazuje vrednost piksla skozi čas. V trenutkih, ko piksel predstavlja gibajoči se objekt, opazimo v sliki na tem mestu ostre spremembe v intenzivnosti. Ostre spremembe detektiramo z operatorji za robove (robove skozi čas). Ozadje tvorimo s povprečenjem vrednosti pikslov, ki niso blizu robov. Še bolj sofisticirana je metoda detektorja gladkosti [LY90], ki je nasprotna pravkar opisani metodi. Osnovna ideja je v tem, da ima vrednost ozadja najdaljšo in stabilno vrednost, gledano skozi čas. Ta metoda išče stabilna območja in ignorira nestabilna območja. To dosežemo z drsečim oknom (skozi čas) po sledi vrednosti piksla – opazujemo isti piksel skozi vse slike. Kadar imajo vsi piksli, ki so znotraj drsečega okna, vrednost v preddefiniranem območju (dopuščena so določena nihanja sivin ozadja), potem prištejemo povprečno vrednost teh pikslov k skupni vsoti in povečamo število elementov. Drseče okno zatem premaknemo za eno sliko in ponovimo ves postopek. Na koncu skupno vsoto delimo s številom vseh elementov ter tako določimo končno povprečje za ta piksel.

Izboljšavi opisane metode sta adaptivna metoda detektorja gladkosti, ki adaptivno spreminja velikost drsečega okna in preddefiniranega območja, ter metoda polnjenja stabilnih območij [LY90]. Ko imamo določeno ozadje, potem z odštevanjem slik enostavno izločimo gibajoče se objekte.

Slika robov gibanja

Za segmentiranje gibajočih se objektov lahko uporabimo tudi sliko robov. Ti robovi morajo prav tako kakor ostale mere gibanja vsebovati informacije o gibanju. Ko imamo določene robove gibanja, potem lahko nad takšno sliko uporabimo razne segmentacijske metode na osnovi robov [SHB94, Pot98]. V grobem bomo opisali nekaj takšnih detektorjev robov gibanja. Detektor gibajočih se robov M na trenutni sliki I_k je definiran kot [VCP95]:

$$M(I_k) = \left\|\nabla(\triangle I_{k-1})\right\| \times \left\|\nabla(\triangle I_k)\right\|,$$

kjer $\triangle I_k$ označuje sliko razlik med slikama I_{k+1} in I_k, podobno velja za $\triangle I_{k-1}$. Z $\|\nabla()\|$ pa označimo velikost odziva (absolutna vrednost) te slike razlik po obdelavi z differencialnim operatorjem ∇.

Seveda najdemo v literaturi še celo vrsto drugih operatorjev za detektiranje robov gibanja. V [CN96] je predstavljen preprost, a hkrati računsko učinkovit detektor robov. Zasnovan je na dveh prepostavkah: a) intenzivnost v sliki lahko modeliramo kot odsekoma zvezno linearno funkcijo, b) kamera se mora gibati zvezno. V prvem koraku določimo drugi odvod slike po času – pri računanju upoštevamo tri zaporedne slike – zatem pa eliminiramo napačne pozitivne kandidate. V zadnjem koraku ta rezultat le še obdelamo s pragovno funkcijo. V [DC92] najdemo še en detektor robov gibanja. Njegova dobra lastnost je, da je sposoben razlikovati med spremembami v intenzivnosti, nastalimi zaradi gibanja ter časovnih sprememb v osvetlitvi. Ta detektor je možno uporabiti tudi pri izračunu optičnega pretoka.

Optični pretok

Pri klasificiranju pikslov na gibajoče se in statične lahko uporabimo tudi sliko z izračunanim optičnim pretokom. Najpreprostejša segmentacijska metoda je gotovo pragovna operacija. Piksle, ki imajo optični pretok dovolj močan, označimo kot "gibajoče se", ostale pa kot statične. Slabost takšnega pristopa je v tem [TV98]: a) da dobimo slabo oceno za optični pretok znotraj lokalne okolice, ki vsebuje piksle več gibajočih se objektov, b) da je v nekaterih aplikacijah (npr. za nadzor) treba zgolj detektirati gibanje oz. spremembe v sceni.

Zanimiv je pristop, ki ga uporabijo avtorji v [WA94] za ocenjevanje parametrov gibanja in segmentacijo. Predpostavljajo, da je možno gibanje v vsaki točki (m, n) med dvema zaporednima slikama I_k in I_{k+1} popisati z enačbo (2.14):

$$I_k(m - u_x(m,n), n - u_y(m,n)) = I_{k+1}(m,n), \tag{2.14}$$

kjer je z $\mathbf{u} = [u_x, u_y]^T$ označen optični pretok. S to enačbo dejansko predpostavijo, da veljajo vse omejitve, ki so potrebne za izračun optičnega pretoka. Ko določijo optični pretok (glej podpoglavje 2.2.1), identificirajo regije s podobnim gibanjem (segmentacija). Predpostavimo, da so v sliki zgolj afina gibanja (npr. translacija, strig in skaliranje). Tedaj lahko optični pretok izrazimo v obliki

$$\begin{aligned} u_x(m,n) &= a_{x0} + a_{xx}m + a_{xy}n, \\ u_y(m,n) &= a_{y0} + a_{yx}m + a_{yy}n. \end{aligned} \tag{2.15}$$

Naj bo $\mathbf{a}_i^T = [a_{x0_i}, a_{xx_i}, a_{xy_i}, a_{y0_i}, a_{yx_i}, a_{yy_i}]$ i-ti hipotetični vektor v 6D prostoru parametrov afine transformacije ter $\mathbf{a}_{x_i}^T = [a_{x0_i}, a_{xx_i}, a_{xy_i}]$ in $\mathbf{a}_{y_i}^T = [a_{y0_i}, a_{yx_i}, a_{yy_i}]$ vektorja, ki ustrezata komponentam m in n. Označimo še $\phi^T = [1, m, n]$, tako da lahko preoblikujemo enačbo (2.15) v

$$\begin{aligned} u_x(m,n) &= \phi^T \cdot \mathbf{a}_{x_i}, \\ u_y(m,n) &= \phi^T \cdot \mathbf{a}_{y_i} \end{aligned} \tag{2.16}$$

in linearna ocena po metodi najmanjših kvadratov za $\mathbf{a}_i$ za dano polje optičnega pretoka pa je naslednja:

$$[\mathbf{a}_{x_i}, \mathbf{a}_{y_i}] = [\sum\sum_{Q_i} \phi\phi^T]^{-1} \sum\sum_{Q_i} (\phi \cdot [u_x(m,n), u_y(m,n)]), \tag{2.17}$$

kjer teče vsota po vseh pikslih znotraj regije Q_i. Ne glede na velikost začetne regije bo mnogo afinih hipotez napačnih. Zanesljivost hipoteze pa ocenimo s statističnim testom oz. z napako ostanka r_i:

$$r_i^2 = \frac{1}{\|Q_i\|} \sum\sum_{Q_i} \Big(\mathbf{u}(m,n) - \mathbf{u}_{a_i}(m,n)\Big)^2, \tag{2.18}$$

kjer z $\mathbf{u}_{a_i}$ označimo optični pretok, določen po enačbi (2.15), $\|Q_i\|$ pa označuje število pikslov v regiji Q_i. Hipoteze s preveliko napako zavržemo.

Modeli gibanja, ki v regiji pokrivajo isti objekt, bodo imeli podobni vektor afine transformacije $\mathbf{a}_i$. Izboljšava opisane metode je v [BBAT97].

V [MBC94] avtorji predstavljajo dinamični sistem za štetje oseb, ki vstopajo v avtobus in izstopajo iz njega. Za določanje smeri gibanja oseb prav tako uporabljajo optični pretok. Slike so zajete s kamero, ki je pritrjena nad vstopno-izstopnimi vrati avtobusa.

Detektiranje sprememb

V posameznih primerih želimo dobiti zgolj odgovor na vprašanje, ali je v zaporeju slik prišlo do sprememb ali ne – do gibanja ali ne. Za takšen odgovor seveda ne potrebujemo posebej sofisticiranih metod, saj ni potreben izračun parametrov gibanja ipd. Kljub vsemu pa so takšne metode zelo uporabne, predvsem v sistemih za nadzor, ali pa če želimo hitro dobiti odgovor na vprašanje, ali gre v zaporedju za gibanje, in šele s pritrdilnim odgovorom se lotimo obdelave zaporedja s kompleksnimi metodami.

V [IT93] avtor uporablja za detektiranje gibanja normaliziran histogram, izračunan za vsako sliko. Za vsak histogram določi porazdelitveno funkcijo za sivine (povprečna vrednost sivin in varianca). Zatem pa za dve zaporedni sliki izračuna tri statistike: Kolmogorov kriterij, Pirsonov kriterij in kompleksen kriterij – test pomembnosti. Te statistike primerja s pragom, pri čemer so pragovi določeni iz tabel za intervale zaupanja. Če katera statistika preseže prag, potem rečemo, da smo v sliki detektirali gibanje. Zaradi svoje preprostosti je to rutino možno implementirati kot realnočasovno, avtor pa jo je uporabil v sistemu za nadzor.

Druge metode

Ob vseh segmentacijskih metodah, ki smo jih predstavili, obstaja seveda še cela vrsta drugih metod, ki uporabljajo za segmentacijo ali kombinacijo omenjenih metod ali pa popolnoma svoje pristope. V nadaljevanju bomo opisali nekaj tipičnih takšnih metod.

V [NCRT98] predlagajo metodo za razločanje gibajočih se objektov od ozadja v splošnem video zaporedju. Sliko modelirajo kot vsoto nepremičnega ozadja, pokritega in odkritega ozadja (ozadja, preko katerega se objekt giblje), signala, ki ustreza gibajočemu se objektu, ter ustreznega šuma (šum kamere, napaka, ki jo prinese faza predobdelave za kompenzacijo premikanja kamere in sprememb v osvetlitvi). Objekt je lahko v naslednjih slikah premaknjen in/ali izkrivljen glede na začetni trenutek opazovanja. Ozadje in objekt pa zatem ločijo med seboj na osnovi testa pomembnosti statistik višjih redov izračunanih, na množici slik razlik.

V [IRP97] so razvili metodo za računanje lastnega gibanja kamere v statični vizualni sceni. Regije določijo z dominatnimi 2D parametričnimi transformacijami, zatem pa izračunajo 2D parametre gibanja med dvema zaporednima slikama. Uporabljajo tudi optični pretok za izračun fokusa širjenja FOE kamere (*focus of expansion*).

Ponavadi imamo opravka z zaporedji slik, ki so povzorčena z visokimi frekvencami v času in so zato prostorske razlike med zaporednimi slikami majhne [J93]. V [IRP92] pa je predstavljen algoritem za detektiranje in sledenje več gibajočim se objektom z uporabo velikih prostorskih in časovnih regij brez predpostavke o časovni konstantnosti gibanja.

Na osnovi dveh zaporednih slik se določi gibajoči se objekt in njegovo gibanje. Ko je objekt določen, ga izločijo iz slike, v preostalem delu slike pa iščejo morebitne druge objekte. Zatem pa objektom sledijo skozi zaporedje s t. i. časovno integracijo, kjer skušajo trenutno regijo na osnovi modelov gibanja (afine transformacije, projekcijske transformacije) transformirati k naslednjemu novemu okvirju. Objekt, ki mu sledijo, ostane pri sledenju oster, vsi ostali deli v sliki, tudi regije z drugačnim gibanjem, pa so zapacani.

2.4 Aktivne krivulje – kače

V tem podpoglavju bomo predstavili aktivne krivulje, znane tudi pod imenom kače. Čeprav so aktivne krivulje prav tako ena izmed segmentacijskih metod, smo jim zaradi njihove vsestranske uporabnosti pri obdelavi slik in zaporedij slik namenili lastno podpoglavje. Aktivne krivulje tudi predstavljajo nekakšno prelomnico med tehnikami za obdelovanje zaporedij slik, saj se enakovredno pojavljajo pri zaporedjih vizualne scene in pri medicinskih slikah, kjer imamo opravka z zaporedji slik prerezov.

V podpoglavju 2.4.1 bomo najprej vpeljali matematični model za aktivne krivulje, v nadaljevanju pa bomo prikazili njihovo uporabo pri obdelovanju zaporedja slik (podpoglavje 2.4.2).

2.4.1 Definicija

Aktivna krivulja (kača) je elastična oz. spreminjajoča se zvezna krivulja. Njeno obliko kontroliramo z notranjimi zveznimi silami E_{not} ter zunanjimi silami E_{zun} v sliki, in sicer na osnovi minimizacije energije. Kača dejansko spreminja svojo obliko in lego, dokler vrednost energijskega funkcionala ne doseže lokalnega minimuma. Notranje sile delujejo kot omejitve gladkosti, zunanje sile pa vodijo kače k značilkam, ki nas v sliki zanimajo (npr. linije, vogali).

Naj bo $s(t) = (x(t), y(t))$ parametrično podana kača, kjer sta $x(t)$ in $y(t)$ slikovni koordinati krivulje, t pa je proporcionalen dolžini krožnega loka, $t \in [0,1]$. Njena skupna energija je [WEHK96, PZ98]

$$E_{ka\check{c}e} = \int_{ka\check{c}e} E(s(t))dt = \int_{ka\check{c}e} \Big[E_{not}(s(t)) + E_{zun}(s(t))\Big]dt. \tag{2.19}$$

Notranja sila je definirana kot

$$E_{not}(t) = \frac{1}{2}\left(\alpha(t)\Big|\frac{\partial s(t)}{\partial t}\Big|^2 + \beta(t)\Big|\frac{\partial^2 s(t)}{\partial t^2}\Big|^2\right), \tag{2.20}$$

kjer sta $\alpha(t)$ in $\beta(t)$ poljubni funkciji, ki regulirata elastičnost (napetost) in togost krivulje, ponavadi pa sta kar konstantni: $\alpha(t) = \alpha$ in $\beta(t) = \beta$.

Zunanjo energijo E_{zun} določimo iz podatkov iz slike tako, da zavzame minimalne vrednosti pri značilkah, ki so za nas zanimive (npr. pri mejah). Tipično jo skonstruiramo tako, da se lahko kača približa strmim robovom, ter jo zapišemo:

$$E_{zun} = -|\nabla I(x,y)|^2. \tag{2.21}$$

Cilj je poiskati kačo s^*, ki minimizira enačbo (2.19) ob podani zunanji energiji E_{zun}. Če problem diskretiziramo, lahko dobimo naslednjo iterativno rešitev [WEHK96, PZ98]:

$$\begin{aligned} x_i &= (\mathbf{A} + \gamma\mathcal{I})^{-1}\Big(x_{i-1} - I_x(x_{i-1}, y_{i-1})\Big), \\ y_i &= (\mathbf{A} + \gamma\mathcal{I})^{-1}\Big(y_{i-1} - I_y(x_{i-1}, y_{i-1})\Big), \end{aligned} \tag{2.22}$$

kjer parameter γ določa velikost koraka (lepljivost), matrika $\mathbf{A}$ je matrika parametrov ter je odvisna od α in β. I_x in I_y sta končni aproksimaciji razlik za odvode, (x, y) je točka z diskretizirane kače, indeks i označuje iteracijski korak, medtem ko je $\mathcal{I}$ enotska matrika.

Največji problemi pri segmentiranju s kačami so [WEHK96]: a) ni natančnih postopkov, kako določiti vrednosti parametrov α in β, b) počasna konvergenca, c) težko je kontrolirati obnašanje kače med njeno evolucijo, d) treba je določiti zelo dober začetni položaj kače.

Obstaja mnogo različnih vrst kač, ki skušajo odpraviti ali vsaj omiliti zgoraj omenjene probleme. Tako ob klasičnih kačah najdemo še kače, ki temeljijo na Fourieru, B-zlepkih, končnih elementih, diskretnih reprezentacijah in večstopenjske optimalne kače [TS92, WEHK96]. Ob teh različnih vrstah kač pa se pojavljajo še razne variatne kač [PZ98], ki se razlikujejo med seboj po načinu minimizacije energijskega funkcionala (dinamični algoritem, požrešni algoritem), načinu zapisa zunanjih sil, načinu določitve začetnega položaja itd.

2.4.2 Aktivne krivulje pri obdelavi zaporedja digitalnih slik

Aktivne krivulje oz. kače so spremenljive krivulje, ki so bile uspešno uporabljene v mnogih aplikacijah analize slik, tudi pri zasledovanju togih oz. netogih teles v slikah. Glavni razlog za uporabo kač pri segmentaciji je nedvomno v tem, da omogočajo vključevanje geometrijskih omejitev [LOG99]. V tem podpoglavju si bomo pogledali uporabo kač pri analiziranju zaporedij digitalnih slik.

Najbolj preprosta shema uporabe aktivnih krivulj pri segmentiranju zaporedja slik je gotovo naslednja: v prvi sliki iz zaporedja določimo začetni približek za kačo, le-to pa zatem spreminjamo po enačbi (2.22), dokler ne dosežemo konvergence. Rezultat, dobljen v trenutni sliki, zatem uporabimo kot začetni približek na naslednji sliki. V [HA92] so s kačami sledili nekaterim telesnim strukturam v ultrazvočnih slikah po omenjeni shemi. Za počasi gibajoče se strukture (dejansko gre bolj za deformacijo oblike objekta kot za pravo gibanje), npr. srčno votlino, je časovno sledenje po opisani shemi dalo dobre rezultate. Za hitro gibajoče se strukture pa so uporabili kače neodvisno na vsaki sliki. Za določanje grobega začetnega približka so uporabili tehnike iz matematične morfologije, medtem ko so parametri kač bili določeni empirično.

Aktivne krivulje so učinkovite le, če je začetni približek kače blizu dejanskih robov objekta. To pomeni, da kača sama lahko sledi le zelo počasnim objektom [DBB95]. Da bi prešli to omejitev, moramo predvideti začetni položaj v naslednji sliki z uporabo informacije o gibanju. V [MKT98] so s kačami sledili zaporedju echokardiografskih slik. Začetni približek kače vnese uporabnik, končni rezultat na trenutni sliki pa se s pomočjo informacij o optičnem pretoku prenese v naslednjo sliko, kjer predstavlja začetni približek. Problem te metode je ustrezno določiti parametre kač ter oceniti optični pretok za nepravilne in velike premike. V [MS99] so prav tako sledili gibanju levega prekata v echokardiografskih slikah. Začetni položaj določijo s Houghovo transformacijo za iskanje elips. Pri sledenju s kačami dinamično vključujejo dinamiko srca.

V [DBB95] avtorji formalno podajajo modificirano osnovno shemo uporabe aktivnih krivulj za sledenje objektom v kompleksni sceni. Naj bo s_t kontura objekta O v času t, s^p_{t+1} pa predvidena krivulja v času $t+1$, ki jo dobimo z uporabo ocenjenih parametrov gibanja. Treba je izvesti le še prilagoditev krivulje in dobimo končen opis s_{t+1} v času $t+1$. Prilagoditev je nujna, saj je treba popraviti obliko predvidene krivulje z uporabo novih prostorskih informacij, ki se pojavijo v $t+1$.

Za sledenje predlagajo kače na osnovi B-zlepkov. Slabost osnovne variante kač je v tem, da moramo postaviti $\beta = 0$, če hočemo slediti vogalu, kar pa povzroči nezveznost drugega reda. Pri B-zlepkih pa takšno točko zgolj potrojimo. Osnovna izvedba B-kač ima kljub temu še vedno dve težavi pri sledenju, in sicer: a) izgubo roba objekta zaradi šibkega kontrasta in b) občutljivost na moteče robove iz ozadja. Avtorji [DBB95] so zato razvili nov model aktivnih krivulj, t. i. prilagodljive poligone. Osnovna ideja je v tem, da se vsak segment kače lahko deformira le znotraj vnaprej predpisanega okna. Uvedli so še novo zunanjo energijo, kjer upoštevajo, da lahko v realnih situacijah objekt pogosto modeliramo kot regijo s povprečno sivino, kateri je dodan šum.

V [VCP95] obravnavajo problem analize dinamične scene z več gibajočimi se objekti po kompleksnem ozadju. Večina metod predvideva, da so spremembe v velikosti in obliki gibajočih se objektov majhne ter zvezne (gladke) iz slike v sliko. Detekcija gibajočih se robov presega to zadnjo omejitev, vendar pa je manj uspešna, če se objekti gibljejo po kompleksnem ozadju. Vpeljujejo postopek za sledenje objektom z aktivnimi krivuljami, ne da bi bilo treba ročno inicializirati gibajoči se objekt v prvi sliki, kjer se pojavi. Modificirali so zunanjo energijo E_{zun} kot

$$E_{zun} = -\Big(w_1 M(I_k) + w_2 \nabla I_k\Big),$$

kjer sta w_1 in w_2 uteži, z M označimo sliko robov gibanja (glej podpoglavje 2.3.2), z ∇I_k pa statične robove v sliki.

Predpostavijo, da se objekt v sliki ne pojavi ali izgine magično med dvema zaporednima slikama iz zaporedja. Objekt lahko vstopi v sceno le pri enem od robov slike. Vstopajoči objekt ujame t. i. kača z meje. Le-ta se razdeli na dva dela, ko je objekt popolnoma znotraj slike. Prvi del kače se povrne v svoj prvotni položaj ob rob slike, drugi del pa sledi objektu skozi zaporedje. To razpoznavalno shemo so uspešno uporabili pri analiziranju prometa v mestu. Nerešen pa ostaja problem, kako ločiti objekte od njihovih senc, ter problem prekrivanja objektov (*occlusion*).

Ko kača sledi objektu, lahko posreduje detajlne kvantitativne informacije o njegovem položaju, hitrosti, pospešku in obliki. V [TS92] so izkoristili to lastnost ter modificirali kače v t. i. Kalmanove kače. Kalmanove kače uporabljajo dinamiko kač po eni strani ter po drugi strani vpeljujejo nove informacije, dobljene iz nezanesljivih izmer na optimalen način preko Kalmanovega filtra. Z njimi je možno fleksibilno načrtovati lastnosti sledenja, kar ni možno s konvencionalnimi kačami. Parametre modela, npr. koliko upoštevati nove izmere glede na stare ocene oblik, lahko izpeljemo iz statističnih modelov senzorja in se lahko skozi čas spreminjajo. V [Pet99] so uporabili izboljšan model Kalmanovih kač pri robustnem sledenju položaju in hitrosti.

V zadnjem času se pojavlja še ena nadgradnja aktivnih krivulj [MSV95, PD00]. Gre za modeliranje oblike z razširjanjem osnovne oblike – fronte (*front propagation*), t. i. pristop z množico nivojev (*level set approach*). Ta pristop ohranja prednosti znanih metod ter odpravlja njihove težave. Končni rezultat je relativno neodvisen od začetne inicializacije krivulje. Krivulja se lahko med evolucijo poljubno razbija na dva dela ali se zliva.

2.5 Tehnike pri 3D slikah

Večina metod, ki smo si jih ogledali do tega podpoglavja, pri obdelovanju zaporedja uporablja največ dve do tri zaporedne slike iz celotnega zaporedja. S tem so dejansko degradirale celotno zaporedje na nekaj slik, hkrati pa je tudi količina informacije, ki jo lahko izvlečemo iz nekaj slik, veliko manjša od informacije iz celotnega zaporedja.

V tem podpoglavju bomo predstavili nekaj pravih "3D metod". Te metode gledajo na zaporedje slik kot na 3D strukturo. Večina metod, ki jih bomo predstavili, je bila uporabljena pri zaporedju medicinskih slik. Časovna komponentna v 3D slikah nima nobenega pomena [MCD90]. 3D slika je namreč sestavljena iz zaporedja 2D prerezov. Koncept slikovnega elementa – piksla se v teh slikah razširi na pojem volumskega slikovnega elementa – voksla. Piksel kot osnovni gradnik 2D slik je ponavadi modeliran kot pravokotnik oz. kvadrat, medtem ko je voksel, ki je osnovni gradnik 3D slik, modeliran kot kvader oz. kocka.

S pregledom metod za obdelovanje 3D slik želimo pokazati na različna področja in probleme, ki jih takšne metode pokrivajo oz. rešujejo. Tako si bomo ogledali metode iz skupin: a) 3D detektorjev robov, b) 3D segmentacijskih metod, c) 3D rekonstrukcije, d) tehnike za analizo gibanja na prostorsko-časovnih slikah.

3D detektorji robov

V splošnem rešujejo tako 2D kot 3D metode za določanje robov v slikah isti problem: kako določiti nezveznosti diskretne 2D oz. 3D slikovne funkcije (intenzivnosti). Rob v 3D je ravnina, ki gre skozi center sfere. Pojavlja se tudi vprašanje, zakaj je boljše uporabljati 3D detektor robov kot pa klasičen 2D detektor. To pa predvsem zaradi tega, ker so takšne metode bolj imune na šum in z njimi lažje ocenjujemo velikost gradienta ter izračunamo lahko tudi 3D gradient [MDR91].

V literaturi [MDR91] najdemo mnogo posplošitev znanih 2D tehnik za določanje robov, in sicer 3D posplošitve Robertsovega, Sobelovega, Cannyjevega in Derichevega operatorja. Zaradi izredne časovne zahtevnosti lahko mnogi 3D detektorji robov uporabljajo le majhne konvolucijske maske, največ 5 x 5 x 5, ki pa so zelo senzitivne na šum.

3D segmentacijske metode

Segmentacija diskretnih 3D volumnov ostaja eno izmed najatraktivnejših problemov v računalniškem vidu in obdelavi slik. Volumetrične podatke dobimo iz množice slikovnih aplikacij, npr. seizmografskih in oceanografskih nalog [EM96]. Posebej pomembne so 3D biomedicinske slike, ki jih dobimo z raznimi tehnikami za zajem slik [SG91]: MRI (*Magnetic Resonance Imaging*), CSLM (*Confocal Scanning Laser Microscopy*), 3D ultrazvok, CT (*Computed Tomography*), 3D SPECT (*Single Position Emission Computed Tomography*) itd.

V praktično vseh aplikacijah se pri segmentiranju volumetričnih podatkov srečujemo [EM96]: a) z degradacijami zaradi šuma, b) s prisotnostjo drugih motečih struktur, c) s kompleksnostjo kot rezultatom topoloških karakteristik iskanega objekta, d) z nejastnostjo, povezano s prisotnostjo nenatančnih in nepopolnih vizualnih ključev, e) in z možno nadaljnjo kompleksnostjo, ki je posledica časovnih in dinamičnih efektov.

Ponavadi so 3D segmentacijske metode posplošitev znanih segmentacijskih metod na 2D slikah. Pri posploševanju sekvenčnih segmentacijskih metod lahko [SG91]: 1) uporabimo $(n-1)$-D algoritem za vsako hiper ravnino n-D slike, 2) uporabimo pravi n-D segmentacijski algoritem. Pri izbiranju med variantama je treba pretehtati, katera od obeh variant prinese naši aplikaciji večjo korist (uspešnost). Verjetno bo $(n-1)$-D pristop manj pomnilniško zahteven in bo hitrejši, n-D pristop pa lahko implicitno uporablja vse informacije o n dimenzijah.

Algoritem v [MCD90] segmentira 3D površine z metodo rasti regije. Veljajo popolnoma enake omejitve in predpostavke kot pri rasti regij v 2D [Pot98]. Osnovno površino, pri čemer je v sliki lahko več površin, razdelijo na t. i. primarno bazno množico, ki jo zatem širijo. Pri širjenju se ne opirajo na vrednosti vokslov, ker je slika namreč binarna, ampak upoštevajo razne omejitve sosednosti. Na koncu celotna površina razpade na množico disjunktnih segmentov.

V [SG91] najdemo posplošitev metode 2D deli, združi in grupiraj na 3D algoritem. Pri tej metodi je izbira homogenostnega kriterija najpomembnejši faktor, ki vpliva na segmentacijske rezultate. Če sprejmemo napačnega kandidata, napake ne moremo več popraviti v naslednjih fazah. Ta napaka mora biti zato majhna, kot je le možno. Na izbiro imamo dve vrsti homogenostnih kriterijev: 1) konsistentni kriteriji (npr. kriterij max-min omeji razliko med maksimalno in minimalno sivino v homogeni regiji), 2) inkonsistentni kriteriji (npr. kriterij variance, kriterij sigma).

Avtorji v [EM96] so vpeljali nov pristop za avtomatsko segmentiranje 3D struktur v 3D sliki. Algoritem so uporabili pri določanju 3D orientacije levega srčnega prekata. Sestoji iz več zaporednih korakov: modeliranja histogramov, morfološke obdelave, grupiranja na osnovi modela in označevanja na osnovi znanja. Te operacije se izvajajo dinamično iterativno, kar vodi k temu, da se posamezne informacije dodajajo oz. odvzemajo med samim procesom segmentacije.

Ob predstavljenih 3D segmentacijskih metodah najdemo v literaturi še celo vrsto drugih. Preprost pristop najdemo v [YJK88], kjer v 3D sliki najprej določijo robove s 3D implementacijo filtra Laplace Gaussa (LoG), zatem izboljšajo potek objekta, na koncu pa le še označijo povezane komponente. V [JM93] so se lotili 3D segmentacije MRI slik možganov. Le-to opravijo na osnovi intenzivnosti pikslov ter pričakovane oblike. Najprej opravijo na vsakem prerezu dve pragovni operaciji za izločanje belega in sivega področja v možganih. Geometrijsko segmentacijo obeh področij pa dosežejo s 3D algoritmom polnjenja – gre za 3D povezovanje vokslov s semenom. Zadnji korak algoritma je interpolacija obeh področij, tako belega kakor možganskega tkiva. Ta korak je potreben, ker je resolucija v tretji dimenziji ponavadi dva- ali štirikrat manjša od ostalih dveh prostorskih dimenzij (imamo opravka s slikami 256 x 256 x 40 vokslov).

3D rekonstrukcija

Medicinsko slikovno gradivo je postalo močno diagnostično orodje, predvsem zaradi sposobnosti, da prikaže interne strukture človeškega telesa v obliki slik. Medicinske slike lahko prikažejo informacijo o obliki in strukturi. Vse medicinske slike so dvodimenzionalne. Zelo koristno pri diagnosticiranju pa je, če lahko rekonstruiramo realen 3D objekt iz zaporedja 2D slik ter ga prikažemo s simuliranjem funkcij človeškega očesa.

Avtorji v [WYP97] se ukvarjajo z rekonstrukcijo CT slik (možne tudi rentgenske CT ali ultrazvočne CT slike). Vsi prerezi so vzporedni. Če bi hitro prevrteli posnetke vseh teh prerezov, bi dobili podobo 3D objekta. Za dejansko rekonstrukcijo 3D objekta iz 2D rezin pa potrebujemo podatek o površinski normali. V delu so uporabili informacijo o optičnem pretoku za rekonstruiranje oblike 3D objekta.

Če gledamo dva zaporedna prereza, lahko to obravnavamo, kot da gre za perturbacijo oz. deformacijo enega prereza v drugega, ob predpostavki, da je prostorski razmik med prerezoma majhen. Torej lahko gledamo na problem kot na problem gibanja in zato lahko uporabimo optični pretok.

Znano je, da odvod krivulje določa tangeto na krivuljo. Podobno velja za 3D vektor hitrosti, ki je odvod intenzivnosti površine vzdolž časovne osi (z osi), in je dobra aproksimacija površinske tangente. 2D vektor hitrosti, ki je dejansko projekcija 3D vektorja hitrosti na slikovno ravnino, lahko ocenimo z optičnim pretokom. Ker pa imamo opravka s prebirnikom CT, kjer je razmik med dvema rezinama (prerezoma) fiksen (znano število pikslov), poznamo komponento hitrosti vzdolž osi gibanja. Torej dobimo 3D vektor hitrosti s seštevanjem komponente hitrosti vzdolž osi gibanja ter 2D vektorja hitrosti. Točno vrednost 2D hitrosti pa dobimo, če izračunamo razdaljo od točke na prvi konturi (prvi prerez) vzdolž vektorja optičnega pretoka do pripadajoče točke v naslednji konturi (drugi prerez).

3D normala je pravokotna na 3D tangetno in jo lahko preprosto določimo. S tem so izpolnjeni pogoji za uspešno rekonstrukcijo.

V [ZFA97] pa ocenjujejo 3D oblike in tvorijo grafične modele za takšne objekte. Obdelujejo objekte z zrcalnimi površinami, kar še bolj oteži rekonstrukcijo. Zrcalno odbiti vzorec se namreč premika po gladki površini, če premikamo gledišče. Rekonstrukcija poteka iz zaporedja slik, ki jih dobimo po snemanju objekta, ki ga rotiramo okoli osi z in ob konstantni osvetlitvi.

V [SMH+97] so predstavili algoritem za natančno rekonstrukcijo časovno spremenljivih 3D oblik z znanim ciklom (npr. levi srčni prekat). Algoritem temelji na direktnem prileganju časovno spremenljivih površin, ki so predstavljene s funkcijami B-zlepkov, k delno zakritim krivuljam. Te krivulje so dobljene iz različnih gledišč in časovnih trenutkov. Vpeljali so tudi model za zajemanje časovno spremenljivih slik s spremenljivim gledišcem.

Analiza gibanja na prostorsko-časovnih slikah

Na analizo zaporedja slik lahko gledamo kot na analizo v večdimenzionalnem prostoru, ki je razpet z dvema do tremi prostorskimi koordinatami ter eno časovno. 3D prostorsko-časovne slike si lahko predstavljamo kot sklad zaporednih slik, položenih eno na vrh druge v t. i. sklad slik, ali pa kot slikovno kocko.

Ob takšni predstavitvi se gibanje pojavlja kot orientacija v prostorsko-časovnih slikah [J93]. Obstajajo trije glavni pristopi za analizo gibanja v Fourierovem prostoru na prostorsko-časovnih slikah [J93]: a) z množico filtrov (npr. metoda množice kvadratnih filtrov – Gaborjevi filtri), b) s tenzorsko metodo in c) s fazno metodo.

V [LJ89] so uporabili osnovno idejo, da na problem gibanja lahko gledamo kot na problem detektiranja orientacije tangentnih ravnin na robove v prostorsko-časovni predstavitvi. V okolici časovne točke so funkcijo intenzivnosti modelirali kot Mongovo krpo.

2.6 Metode prediktor-korektor – Kalmanov filter

V podpoglavjih od 2.2 do 2.5 smo se seznanili z raznimi tehnikami za obdelovanje zaporedja slik. Videli smo, da so takšne metode časovno izredno zahtevne. Še posebej se časovna zahtevnost stopnjuje pri korelacijskih tehnikah (podpoglavje 2.2.2), kjer je treba za najboljše ujemanje preiskati veliko iskalno območje. V takšnih primerih, še posebej pri sledenju objektom, je zato smiselno na neki način omogočiti vključevanje znanja o gibanju objektov na predhodnih slikah v algoritem, ki analizira gibanje na trenutni sliki.

To smo že nakazali pri uporabi aktivnih krivulj za obdelavo zaporedja slik. Tukaj smo rezultat na prejšnji sliki uporabili kot začetni približek na trenutni sliki (glej podpoglavje 2.4) in ga potem modificirali do končne oblike.

Pri izbiri, kako vključiti "zgodovino" gibanja objekta skozi zaporedje, imamo neskončno možnosti. Tako na primer lahko upoštevamo le gibanje na predhodni sliki ter celotno gibanje modeliramo kot markovski proces (trenutno stanje odvisno le od predhodnega). Trenutno trajektorijo gibanja lahko tudi aproksimiramo z zvezno in gladko funkcijo ter zatem na osnovi ekstrapolacije določimo verjeten položaj (stanje) v naslednji sliki. Znanje lahko tudi vgrajujemo skozi mnoge algoritme, ki imajo shemo prediktor-korektor. Algoritmi, ki jih uporabljamo kot nadgradnjo že predstavljenih metod, v veliki meri izboljšajo natančnost razpoznavanja. Le-ti skušajo na osnovi prejšnjih rezultatov določiti rezultat v trenutnem posnetku, ta predviden položaj (predikcija) pa se na osnovi ene od metod obdelovanja zaporedja slik korigira. Obstajajo razni filtri in metode na osnovi sheme prediktor-korektor [Hos95], npr. filter alfa-beta, Kalmanov filter, stohastična aproksimacija itd.

Čeprav obstaja mnogo algoritmov prediktor-korektor oz. tovrstnih filtrov, zavzema najpomembnejšo mesto med njimi zagotovo Kalmanov filter (KF). KF je optimalni rekurzivni algoritem za obdelavo podatkov [May94]. Mnogo načinov je, kako definirati optimalnost; odvisno je od kriterija, ki ga izberemo za vrednotenje zmogljivosti (npr. povprečje, mediana). KF je optimalen glede na katerikoli smiseln kriterij. Če bi torej KF poganjali na veliko testih in bi merili rezultate, bi v povprečju dajal boljše rezultate od katerega koli drugega linearnega filtra oz. algoritma [May94].

Kalmanov filter ima tudi posebno značilnost, ki je druge statistične metode prediktor-korektor nimajo. Ima namreč sposobnost, da sam prilagodi svoje parametre glede na statistiko meritev [Hos95]. V drugih filtrih, npr. v filtrih alfa-beta, je odstotek napake konstanten za celotno zaporedje slik oz. ga lahko spremeni le programer. Ta napaka med izmerjenimi in pričakovanimi podatki se uporablja za nastavljanje modelnih parametrov. Kalmanov filter pa dovoljuje, da se ta odstotek spreminja avtomatsko glede na trenutno zaupanje v natančnost parametrov stanja.

Optimalnost je bil glavni razlog, da si bomo v nadaljevanju detajlneje ogledali le optimalni – Kalmanov filter. Najprej bomo predstavili teoretična izhodišča KF, podali ustrezne enačbe, zatem pa še prikazali njegovo uporabo pri obdelovanju zaporedja slik.

2.6.1 Kalmanov filter

Kalmanov filter je bil v prvi vrsti namenjen za področje kontroliranja procesov oz. sistemov. KF obdela vse dosegljive izmere (meritve), ne glede na njihovo natančnost, za ocenjevanje trenutnih vrednosti ciljnih spremenljivk pa uporabi: a) znanja o dinamiki sistema in dinamiki merilne naprave, b) statistični opis sistemskega šuma, napak v meritvah in negotovosti v dinamiki modelov, c) katerekoli dosegljive informacije o začetnih pogojih ciljnih spremenljivk [May94].

Mislili bi si lahko, da sta pri determinističnem modelu sistema, ki ga krmilimo na osnovi kontrolne teorije, že zagotovljena popolna analiza in načrt nekega procesa. Vendar temu ni tako. Obstajajo trije razlogi za to [May94]: 1) noben matematični model sistema ni perfekten, 2) dinamični sistemi so vodeni ne le z njihovimi lastnimi kontrolnimi vhodi, ampak tudi z motnjami, ki jih ne moremo niti kontrolirati niti modelirati deterministično, 3) senzorji ne posredujejo perfektnih in popolnih podatkov o sistemu.

KF združuje vse dosegljive meritve, vnaprejšnje znanje o sistemu in merilni napravi, da tvori oceno želenih spremenljivk na takšen način, da je napaka statistično minimizirana [May94].

Za delovanje KF morajo biti izpolnjene tri osnovne predpostavke, in sicer linearnost sistema, beli in Gaussov šum v sistemu in meritvah [May94]. Če odstranimo Gaussovo predpostavko, potem je možno pokazati, da je KF najboljši filter, z najmanjšo varianco napake, iz razreda linearnih nepristranih filtrov. Beli šum pomeni, da vrednost šuma ni korelirana v času. Pomeni pa tudi, da ima šum enako moč pri vseh frekvencah. Predpostavka, da je Gaussov, ima opravka z amplitudo. V vsaki točki v času zavzame gostota verjetnosti amplitude Gaussovega šuma obliko normalne zvonaste krivulje.

Enačbe diskretnega Kalmanovega filtra

Predpostavimo, da lahko naključni proces, ki ga ocenjujemo, modeliramo v naslednji obliki [BH97]:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k. \tag{2.23}$$

Meritve procesa so na voljo v diskretnih časovnih trenutkih in so med seboj v linearni povezavi

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \tag{2.24}$$

Razložimo še uporabljene oznake:

$\mathbf{x}_k$ – vektor stanja procesa v času t_k, dimenzije n x 1;

$\mathbf{\Phi}_k$ – matrika, ki povezuje $\mathbf{x}_k$ in $\mathbf{x}_{k+1}$, dimenzije n x n;

$\mathbf{w}_k$ – šum v sistemu. Vektor, za katerega se predpostavlja, da je belo Gaussovo zaporedje z znano kovariančno strukturo, dimenzije n x 1;

$\mathbf{z}_k$ – vektor meritev v času t_k, dimenzije m x 1;

$\mathbf{H}_k$ – matrika, ki daje idealno - brezšumno povezavo med meritvami in vektorjem stanja v času t_k, dimenzije m x n;

$\mathbf{v}_k$ – napaka meritve. Predpostavlja se, da je belo Gaussovo zaporedje z znano kovariančno strukturo in ima križno korelacijo z $\mathbf{w}_k$ enako 0. Dimenzije m x 1.

Kovariančne matrike za šum v sistemu $\mathbf{w}_k$ in šum v meritvah $\mathbf{v}_k$ so podane kot [BH97]

$$\begin{aligned} E[\mathbf{w}_k, \mathbf{w}_i^T] &= \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases}, \\ E[\mathbf{v}_k, \mathbf{v}_i^T] &= \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases}, \\ E[\mathbf{w}_k, \mathbf{v}_i^T] &= 0 \qquad \forall k \wedge \forall i. \end{aligned} \tag{2.25}$$

Definirajmo še napako ocenitve kot

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-, \tag{2.26}$$

kjer z $\hat{\mathbf{x}}_k^-$ označujemo oceno (strešica v oznaki), ki je najboljša ocena, preden vključimo meritev v času t_k (zgornji minus v oznaki). Potem je pripadajoča kovariančna matrika napake $\mathbf{P}_k^-$ enaka

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}]. \tag{2.27}$$

Sedaj lahko zapišemo enačbe diskretnega linearnega KF [BH97, GKN+74, May94]:

1. *Izračun ocen v koraku k*

   - *Računanje Kalmanovega cilja:*

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \Big(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k\Big)^{-1} \tag{2.28}$$

   - *Ažuriranje ocene z meritvijo $\mathbf{z}_k$:*

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \Big(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-\Big) \tag{2.29}$$

   - *Izračun kovariance napake za ažurirano oceno:*

$$\mathbf{P}_k = \Big(\mathcal{I} - \mathbf{K}_k \mathbf{H}_k\Big) \mathbf{P}_k^- \Big(\mathcal{I} - \mathbf{K}_k \mathbf{H}_k\Big)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \tag{2.30}$$

2. *Predikcija za naslednji korak $k+1$*

   - *Predikcija stanja:*

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_k \tag{2.31}$$

   - *Kovariančna matrika napake za predvidene parametre:*

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{Q}_k \tag{2.32}$$

Kalmanov filter je izrazito rekurziven. Ko vstopimo v cikel njegovih enačb (enačbe od (2.28) do (2.32)), se lahko v njih neskončno vrtimo. Za zagon diskretnega KF je treba podati začetno oceno stanja sistema $\hat{\mathbf{x}}_0^-$ ter kovariančno matriko napake $\mathbf{P}_0^-$, ko poznamo matriki šuma $\mathbf{Q}$ in $\mathbf{R}$.

Enačba (2.29) pove, da je optimalna ocena v času k, tj. $\hat{\mathbf{x}}_k$, enaka najboljši predikciji njegove vrednosti, $\hat{\mathbf{x}}_k^-$, preden je izmerjen $\mathbf{z}_k$, plus korekcijski člen z optimalno obteženo vrednostjo, pomnoženo z razliko med izmero $\mathbf{z}_k$ in najboljšo predikcijo $\hat{\mathbf{x}}_k^-$. To je predikcijsko-korekcijska struktura filtra [May94].

Treba je poudariti, da obstaja več različnih oblik enačbe (2.30). Kovariančno matriko $\mathbf{P}_k$ lahko določimo še na tri druge načine, ki pa zaradi svoje numerične nestabilnosti niso primerne za vsako aplikacijo.

Izraz $\left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-\right)$ iz enačbe (2.29) imenujemo inovacija meritve oz. ostanek [WB95b]. Matrika $\mathbf{K}$, ki je cilj oz. faktor mešanja (*blending factor*), minimizira posteriorno kovariančno matriko napake $\mathbf{P}$. Če se matrika $\mathbf{R}_k$ približuje 0, potem matrika $\mathbf{K}$ močneje obtežuje ostanek, po drugi strani pa, če se matrika $\mathbf{P}$ bliža 0, potem $\mathbf{K}$ obtežuje ostanek šibkeje. Trditev lahko formuliramo tudi takole. Če se $\mathbf{R}$ bliža 0, potem bolj zaupamo meritvam kot pa modelu sistema (majhno zaupanje v predikcijo). Če pa se $\mathbf{P}$ bliža 0, potem imamo majhno zaupanje v meritve, medtem ko se močno zanesemo na predikcijo.

Na začetku, ko imamo ponavadi še bolj grobe ocene modelnih parametrov, nam matrika $\mathbf{K}$ zagotavlja, da imajo meritve močan vpliv pri ocenjevanju parametrov stanja. Ko pa zaupanje v natančnost parametrov raste z vsako iteracijo, se vrednosti v matriki $\mathbf{K}$ zmanjšajo, kar povzroči, da se zmanjša vpliv meritev pri ažuriranju parametrov in pripadajoče napake [Hos95].

Lastnosti Kalmanovega filtra

Če imamo za proces, ki ga modeliramo s Kalmanovim filtrom, točno znane in pravilne parametre $\mathbf{\Phi}$, $\mathbf{Q}$, $\mathbf{H}$ in $\mathbf{R}$, potem dobimo kot rezultat optimalne ocene za $\hat{\mathbf{x}}_k$ in $\mathbf{P}_k$. Vendar v realnosti vsi ti parametri niso nikoli točno znani. Tako imamo opravka s KF, kjer so $\mathbf{\Phi}$, $\mathbf{Q}$, $\mathbf{H}$ in $\mathbf{R}$ implementirani z napačnimi parametri. Kot rezultat takšnega filtriranja dobimo podoptimalne ocene, kjer matrika $\mathbf{P}$ nima nobenega pomena v smislu ocenjevanja napake [GKN+74].

V KF ni nikjer specificirano, kako določiti kovariančne matrike šuma (matriki $\mathbf{Q}$ oz. $\mathbf{R}$). Tako pogosto uporabimo kar konstantno matriko oz. celo konstanten skalar. Če naredimo kovariančno matriko konstantno, potem reduciramo ocene KF na standardne ocene, dobljene po metodi najmanjših kvadratov [Rao96]. Dobro pa je znano, da so ocene po metodi najmanjših kvadratov močno občutljive na velike napake – podatke, ki ležijo daleč stran od gruče opazovanih podatkov (*outlier*). V [Rao96] so zato razvili robustno obliko KF, ki rešuje problem velikih napak. Uporabili so robustne ocenitvene procedure (ocenjevanje tipa M).

V realnih situacijah uporabe KF se lahko pojavijo problemi z divergiranjem. Le-to se pojavi zaradi [BH97]: a) zaokrožitvenih napak pri računanju – možna rešitev je, da po vsakem rekurzivnem koraku simetriziramo $\mathbf{P}$ in $\mathbf{P}^-$, b) napak v modeliranju (predvideti, da je neki model oz. košček procesa vseskozi konstanten, je zelo tvegano) – možna rešitev je, da dodamo nekaj šuma v vsako spremenljivko stanja, c) problema observabilnosti (ena ali več spremenljivk stanja je skritih pogledu opazovalca – meritvi). Včasih meritve ne zagotavljajo dovolj informacij za ocenjevanje vseh spremenljivk stanja sistema. Če obstajajo nestabilne napake ocene, je to vidno v enem ali več členih po glavni diagonali $\mathbf{P}$, ki se povečuje preko vseh mej. Možna rešitev je, da dodamo ustrezne meritve in tako naredimo sistem bolje opazovan.

Kalmanov filter je stabilen [BH97], če vse lastne vrednosti izraza $(\mathbf{\Phi}_{k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{\Phi}_{k-1})$ ležijo znotraj enotskega kroga v z-ravnini.

Če je proces, ki ga ocenjujemo, in/ali če so meritve sistema nelinearne, potem predpostavke linearnega KF ne veljajo več. V teh primerih lahko uporabimo modificiran KF, t. i. razširjen Kalmanov filter (*Extended Kalman Filter – EKF*). EKF dobimo tako [BH97, WB95b], da Kalmanov filter lineariziramo okoli trenutnega povprečja in kovariance (na neki način podobno razvoju v Taylorjevo vrsto).

Ob razširjenem Kalmanovem filtru obstajajo še razne druge variante [BH97, GvBSD97], npr. lineariziran KF, adaptivni KF, zvezni KF, Schmidtov KF, gladko omejen KF itd.

2.6.2 Kalmanov filter pri obdelovanju zaporedja slik

Kalmanov filter je zaradi svojih dobrih karakteristik dosegel množično uporabo tudi na področju računalniškega vida. Posebej vidno vlogo ima KF pri sledenju objektom v zaporedju slik. V najbolj osnovni varianti sledenje s KF izvedemo na naslednji način [TV98]: ciljni objekt aproksimiramo z njegovim težiščem, zatem pa informacije o premikanju težišča iz slike v sliko vgradimo v KF. V stanju sistema hranimo koordinati x in y težišča ter odmika $\triangle x$ in $\triangle y$. Na osnovi predikcije $\mathbf{\Phi} \cdot \hat{\mathbf{x}}_k$ pa predvidimo položaj težišča v naslednji sliki.

V [JKC97] so predstavili realnočasovni sistem za sledenje objektom s kamero po modificirani, zgoraj omenjeni shemi. Sistem detektira objekt, ko vstopi v polje pogleda kamere, in mu sledi s kontroliranjem servo naprave na takšen način, da objekt vedno leži v sredini slike. Za sledenje gibajočemu se objektu uporabljajo tehniko ujemanja modelov (korelacijske tehnike). Dovoljujejo, da se model objekta spreminja dinamično med procesom sledenja, tako da lahko asimilira variacije v obliki in intenzivnosti ciljnega objekta. Zgodovino sledenja popišejo s parametri stanja KF. Ocenjeni parametri stanja se uporabljajo za reduciranje področij pri iskanju ujemanja med modelom in za upravljanje servo naprave. Stanje sistema so popisali z 8D vektorjem $\mathbf{x} = [\triangle x, \triangle y, xs, ys, \triangle\overline{x}, \triangle\overline{y}, \overline{xs}, \overline{ys}]^T$, kjer sta $\triangle x$ in $\triangle y$ premik težišča objekta (modela), xs in ys sta horizontalna in vertikalna dolžina obdajajočega pravokotnika z objektom. Vrednosti s prečno črto pa pomenijo odvode po času t. Matriko $\mathbf{\Phi}$ so definirali kot:

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & 0 & 0 & \triangle t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \triangle t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \triangle t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \triangle t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

V vsakem časovnem trenutku merijo le spremembe v položaju in velikosti objekta. Meritev $\mathbf{z}$ in matrika $\mathbf{H}$ sta naslednje oblike:

$$\mathbf{z} = \begin{bmatrix} \triangle x \\ \triangle y \\ xs \\ ys \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Matrike $\mathbf{P}$, $\mathbf{Q}$ in $\mathbf{R}$ se v splošnem določijo na osnovi določenih statistik. Avtorji [JKC97] so vzeli za njih kar identiteto. Za začetno stanje $\hat{\mathbf{x}}_0^-$ uporabijo informacije, ki jih dobijo pri tvorjenju začetnega modela ciljnega objekta.

Poročajo o dobrih rezultatih. Tudi predvidene vrednosti po KF se zelo dobro ujemajo z izmerjenimi. Razlike so le na mestih, kjer se parametri gibanja objekta spremenijo nelinearno. Predpostavili so namreč, da bo trajektorija ciljnega objekta variirala s konstantnim pospeškom ter da bo velikost objekta variirala linearno.

KF pa ne podaja vedno pravilne predikcije [ZM97]. To se večinoma zgodi, ko objekt izvede kakšen manever, ki ni predvidljiv, in tako zavede KF. V teh primerih dobimo boljše rezultate, če namesto gole predikcije uporabimo zadnje ocenjeno gibanje objekta.

Žal je nemogoče predvideti, kdaj bo objekt izvedel manever. Zato je potreben test, da se odločimo, kdaj je bolje uporabiti za predikcijo zadnjo oceno gibanja namesto Kalmanove predikcije. Avtorji predlagajo tehniko, ki temelji na oceni kompenzacije napake gibanja. Objekt, ki mu sledimo, kompenziramo z rezultatom oz. oceno v času t in času $t+1$. Kompenzacija gibanja med obema, ki daje manjšo povprečno kvadratno napako, je izbrana za boljšo predikcijo.

V [BRM97] so predstavli algoritem za določanje trajektorije gibajočega se objekta, ob tem pa so uporabili te trajektorije za predikcijo bodočega gibanja. Pri predikciji so uporabili tri razširjene KF. Podatki o gibanju in trajektoriji služijo za kalibriranje kamere. Kamera sledi objektu s pravo hitrostjo in izostreno.

V [Bur93] so razvili paralelni razširjeni KF. Uporabili so ga na šumnih zaporedjih slik za izboljšanje razmerja SNR ter za ocenjevanje hitrosti objekta, ki se giblje v zaporedju.

Kombinacijo več modulov standardnih in razširjenih KF so uporabili tudi za sledenje ljudem na podzemni postaji [RT94].

V [TK96] se ukvarjajo s problemom sledenja več objektom ob gibajoči se kameri. To je izredno zahteven problem, in sicer: a) ker se objekti gibljejo v sceni, se njihove 2D projekcije v sliki spreminjajo zelo drastično, b) dinamika objektov ni nujno, da je med procesom sledenja konstantna, c) hitro gibajočim se objektom je težko slediti, medtem ko počasi gibajoče se objekte lahko sploh prezremo.

Avtorji rešujejo problem neodvisno gibajočih se objektov in kamere. Algoritem so uporabili pri sledenju objektom s helikopterjem. Iz nekaj začetnih slik s pomočjo optičnega pretoka določijo gibajoče se regije. Te regije zatem grupirajo v objekte. Objektom zatem z metodo iz [HNR93] sledijo v naslednjih slikah. Pri sledenju uporabljajo razširjen KF, ki bistveno zmanjša velikost preiskovalnega območja za iskanje ujemanja v naslednjih slikah.

Problem sledenja vozilom vključuje kompleksne osvetlitvene pogoje in neodvisno gibajoče se objekte. Ob tem pa se še pojavljajo druge omejitve, ki jih lahko vključimo preko časovnih in geometrijskih modelov [GL96].

Za sledenje vozilom so vpeljali dva modela: a) model gravitacije – vozila se lahko gibljejo le pravokotno na smer gravitacije, b) model vozila – uporabnik ima na voljo šest prostostnih stopenj, da sam vnese začetni model. Znotraj modela vozila se določijo značilke, ki se grupirajo, če ležijo skupaj.

Gibanje vozila je lokalno ravninsko, zato je dovoljena rotacija lahko le okoli normale ravnine. Pri določanju gibanja so uporabili KF. Model šuma v sistemu $\mathbf{Q}_i$ so empirično postavili na $\mathbf{Q}_i = \frac{\mathbf{R}_i}{10}$. Parameter $\mathbf{P}$ je mera za napako v oceni. V začetku je $\mathbf{P}_i$ enak $\mathbf{R}_i$. Parameter $\mathbf{R}_i$ je model za šum v meritvenih enačbah in je določen iz parametrizacije modela gravitacije in vozila. Ker je $\mathbf{P}_i$ tudi mera napake v trenutni oceni, je bila uporabljena za omejevanja iskanja med izločanjem značilk.

3.

Algoritem za avtomatsko razpoznavanje objektov v statičnih 2D slikah

V prejšnjem poglavju smo skušali predstaviti čimbolj raznovrstne in hkrati tipične metode, ki jih danes uporabljamo pri obdelovanju zaporedja digitalnih slik. Seznanili smo se s koncepti teh metod in spoznali njihove omejitve ter možnosti njihove uporabe v realnih situacijah.

V naslednjih poglavjih bomo razvili in podrobno opisali algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Pokazali bomo, da se kvaliteta razpoznanih objektov s pomočjo informacij iz celotnega zaporedja slik bistveno izboljša, kot pa če bi razpoznavali objekte zgolj z razpoznavalnim algoritmom na eni statični 2D sliki. Oba nova algoritma, tako algoritem za razpoznavanje objektov na eni statični 2D sliki kot algoritem za razpoznavanje objektov iz zaporedja slik, sta v svoji zasnovi splošna in s tem neodvisna od tipa slik, predpostavljata le, da so objekti temnejši od svoje okolice. Njuno delovanje bomo demonstrirali na realnih ultrazvočnih posnetkih jajčnikov z jajčnimi mešički. Izbira domene vpliva zgolj na določitev posameznih parametrov, potrebnih za delovanje obeh algoritmov.

V tem poglavju bomo zato najprej podrobno predstavili nov algoritem za razpoznavanje objektov v eni statični 2D digitalni sliki. Uspešnost algoritma bomo merili tako na fantomih ultrazvočnih posnetkov jajčnikov kakor tudi na realnih ultrazvočnih posnetkih jajčnikov z jajčnimi mešički.

3.1 Ultrazvočne slike in jajčni mešički

V tem podpoglavju bomo razložili uporabo ultrazvočnih slik jajčnikov z jajčnimi mešički za primer naših testnih slik.

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. Kljub že kar nekajletni prisotnosti 3D ultrazvočnih naprav so klasični 2D ultrazvočni sistemi še vedno daleč najbolj razširjeni, in to predvsem zaradi svoje cenovne dostopnosti ter sorazmerno enostavnega upravljanja [Pot98]. Zajem ultrazvočnih slik je neinvaziven, cena zajemanja je relativno nizka v primerjavi z drugimi medicinskimi slikovnimi tehnikami [HA92].

Pri klasičnem 2D ultrazvoku moramo računati kar na nekaj objektivnih omejitev [Pot98], in sicer: a) subjektivnost konvencialnega pregleda, ki se pojavi zaradi različnih izkušenj in znanja diagnostika (sam prilagaja pregled in postavlja diagnozo); b) težko lociranje tanke 2D ultrazvočne slikovne ravnine v organih (na voljo imamo le dve prostorski koordinati, zato je težko orientiranje v prostoru); c) po že opravljenem pregledu je težko reproducirati določeno lokacijo v sliki; d) pacientova anatomija včasih omejuje kot, pod katerim lahko pošljemo ultrazvočni snop.

Ob teh se pojavljajo še drugi moteči dejavniki, kot so npr. pegasti šum in odboj, ki jih moramo upoštevati tudi pri avtomatskem razpoznavanju različnih struktur v posnetkih, dobljenih s konvencionalnim 2D ultrazvočnim pregledom. Avtomatska analiza ultrazvočnih slik združuje večino težkih problemov iz računalniškega vida, ob tem pa še nekatere specifične [HA92]: a) slike so ponavadi zajete v polarni geometriji namesto v kartezični (potrebna je transformacija med sistemoma), b) zaradi visoke stopnje šuma je degradacija slik velika, c) opazovani objekti so večinoma nestatični, elastični in ne poliederski. Iz tega lahko vidimo, da so ultrazvočne slike za obdelavo med najzahtevnejšimi tipi medicinskih slik.

Ob vseh problemih, ki smo jih nanizali, se pojavlja še problem verificiranja rezultatov. Dobljene rezultate avtomatskega razpoznavanja lahko primerjamo le s segmentacijskimi rezultati, ki so jih opravili eksperti opazovalci, saj so le takšni rezultati edino merilo, ki je na voljo, pa še ti so v veliki meri popačeni (inter in intra opazovalčeva variabilnost). Takšna opazovalčeva variabilnost pa lahko močno zamegli pravo podobo o uspešnosti ali neuspešnosti razpoznavalnega algoritma. Možna rešitev problema verificiranje avtomatskih postopkov razpoznavanja v ultrazvočnih posnetkih je, da uspešnost algoritmov preverimo najprej na fantomih (umetnih slikah), kjer so vse razmere vnaprej točno znane, šele zatem pa na realnih slikah. Fantome pa dobimo tako, da ustrezno modeliramo razmere, ki se pojavljajo pri dejanskem tvorjenju ultrazvočnih slik. Tega pristopa se bomo oprijeli tudi pri našem delu. V podpoglavju 3.2 bomo zato razvili postopek za tvorjenje umetnih ultrazvočnih posnetkov jajčnikov z jajčnimi mešički.

Zakaj razpoznavanje jajčnih mešičkov? V [GNVT95] 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 sistemom, 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. Avtomatsko razpoznavanje jajčnih mešičkov bi v veliki meri razbremenilo eksperte.

Računalniško podprti postopki za analiziranje ultrazvočnih slik jajčnih mešičkov so redki. V [Pot98, PZK97, PVZ98] so se lotili segmentacije ultrazvočnih posnetkov z jajčnimi mešički s pragovno operacijo z optimalnim pragom, ki so jo uporabili nad grobo ocenjenim jajčnikom. Poročali so, da ta popolnoma avtomatska metoda, ki uporablja robove za ocenjevanje mej jajčnika, in pragovna operacija kot segmentacijsko orodje ne dajeta optimalnih rezultatov (prenizek odstotek razpoznavanja). V [PZ98] so ta algoritem nadgradili z aktivnimi krivuljami in tako bistveno izboljšali kvaliteto razpoznanih objektov, robovi razpoznanih objektov so bili veliko bližje pravim mejam. Nerešeno pa ostaja vprašanje, kako avtomatsko določiti ustrezne parametre za aktivne krivulje. Za segmentiranje mešičkov so bili uporabljeni tudi celični avtomati [VDZ98, PVZ98] in celične nevronske mreže [PVZ98]. Rezultati so zelo obetajoči, vendar je očitna pomanjkljivost obeh metod v tem, da je izredno težko določiti ustrezne parametre za segmentiranje mešičkov. V [SLSP98] poročajo o polavtomatski metodi za določanje zunanje stene mešičkov. Za delovanje te metode je treba ročno orisati notranjo steno mešičkov, sicer zelo v grobem, pa vendar. To je vsekakor pomanjkljivost metode. Avtorji v [KS98] so določali notranjo steno mešičkov z uporabo t. i. segmentacijske metode razvodja (*watershed segmentation*). Segmentacijsko metodo razvodja so uporabili nad zglajeno sliko, kar je povzročilo, da so se posamezni majhni sosednji mešički zlili. Za razdruževanje teh regij so v naslednjem koraku uporabili metode binarne matematične morfologije. Posamezni koraki so bili po našem mnenju dodani ad hoc.

Iz vsega naštetega lahko vidimo, da ne obstaja uspešna avtomatska metoda za določanje in analiziranje jajčnih mešičkov v posamezni sliki ultrazvočnega posnetka. Še več. Pri enem ultrazvočnem pregledu jajčnikov dobimo dejansko celo množico 2D slik. V smislu digitalne obdelave slik imamo tako opravka z zaporedjem slik. Vsi dosedaj razviti algoritmi pa segmentirajo mešičke le na eni sliki iz zaporedja in ne omogočajo vključevanja informacij iz celotnega zaporedja. Vse to so bili razlogili, da smo si izbrali ultrazvočne slike jajčnikov z mešički za testiranje postopkov z zaporedji slik.

3.2 Algoritem za tvorjenje fantomov jajčnikov

Že v uvodu k poglavju smo zapisali, da se pri analizi uspešnosti segmentacijskih algoritmov pojavlja problem njihovega verificiranja. Ponavadi algoritme testiramo na realnih posnetkih, kjer nam referenčne položaje določijo eksperti. Ti odčitki pa vnašajo v vrednotenje uspešnosti veliko mero nejasnosti. Zato je pravilen pristop, da uspešnost algoritmov ovrednotimo najprej na fantomih, kjer so vse razmere popolnoma znane, ter šele nato na realnih posnetkih.

Kot smo že omenili, bodo naše testne slike zaporedje ultrazvočnih posnetkov jajčnikov z jajčnimi mešički. V tem podpoglavju bomo razvili nov algoritem za tvorjenje umetnih ultrazvočnih posnetkov jajčnikov z jajčnimi mešički. Za tvorjenje umetnih ultrazvočnih posnetkov jajčnikov z jajčnimi mešički potrebujemo ustrezna matematična modela jajčnika in ultrazvočne sonde.

3.2.1 Model jajčnika z jajčnimi mešički

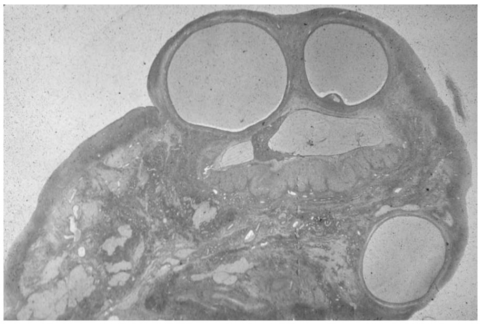
Jajčnika sta ploščati, ovalni oz. mandeljni tvorbi, belkaste neravne površine in sta vpeta v zadnjo stran podvojka ob strani maternice [Pot98, App, GNVT95]. Jajčniki merijo tipično od 2,2 do 5,2 cm v dolžino, od 1,5 do 2,0 cm v širino ter od 1,5 do 3,0 cm v višino [App]. V skorji jajčnika so številni jajčni mešički v različnih fazah zorenja. Pramešički so najbolj nezreli. Tvorijo jih jajčne celice, ki jih obdaja enoslojna ploščata vrhnjica. Najzrelejši so Graafovi mešički, ki se ciklično pojavljajo vsak mesec, praviloma po eden pri spolno zrelih ženskah [GNVT95, PZK97, Pot98]. Tudi mešički so bolj ovalne oblike. Mešički zrastejo v povprečju od 8 do 10 mm, le dominanten zrel jajčni mešiček lahko doseže tudi od 17 do 25 mm v premeru [PZK97, Pot98, App]. Na ultrazvočnih posnetkih zaznamo mešičke, ko so veliki od 2 do 3 mm, zaznamo pa jih kot temna ovalna področja. V sredini jajčnika se nahajajo tudi druge strukture, kot so krvne žile, mezgovnice in živčne niti [Pot98].

Na sliki 3.1 je prikazan prerez skozi jajčnik. Dobro lahko vidimo zrela jajčna mešička v desnem zgornjem kotu, opazimo pa lahko tudi nehomogeno notranjost jajčnika.

Na osnovi pridobljenega znanja o jajčnikih in mešičkih smo postavili naslednji model. Jajčnik modeliramo kot kvader velikosti 4 x 2 x 3 cm (dolžina x širina x višina). To je vsekakor aproksimacija pravega jajčnika, ki smo jo pa naredili zaradi poenostavljenega računanja z matrikami v naslednjih korakih. Prav tako smo zaradi enostavnejše izvedbe snemanja s sondo postavili koordinatno izhodišče v sredino kvadra.

Notranjost jajčnika je sestavljena iz mnogih različnih struktur, mi smo modelirali le jajčne mešičke. Mešičke aproksimiramo z nerotiranim elipsoidom, definiranim s formulo:

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} + \frac{(z-z_0)^2}{c^2} = 1, \tag{3.1}$$



Slika 3.1: Prerez jajčnika. V desnem zgornjem kotu sta dobro vidna dva zrela Graafova jajčna mešička, desni vsebuje jajčece.

kjer je točka (x_0, y_0, z_0) središče elipsoida, parametri a, b in c pa pomenijo njegovo dolžino, širino ter višino. Parametre a, b in c določimo naključno po porazdelitveni funkciji χ^2 s povprečno vrednostjo 10 mm. Površina mešičkov v realnosti ni gladka, zato smo na površino takšnih elipsoidov dodali množico izrastkov. Izrastek smo popisali z izredno majhno kroglo, ki smo jo postavili naključno na površino elipsoida. Naključno je izbrana tudi velikost kroglice.

Tako tvorjene mešičke smo zatem naključno postavili v jajčnik, modeliran kot kvader. Prekrivanja in dotikanja mešičkov med seboj seveda nismo dovolili.

Zadnji korak tega modeliranja je diskretizacija. Ker imamo pri našem delu opravka z avtomatskimi postopki in računalniki, je treba celoten model še diskretizirati. V splošnem lahko vzamemo poljubno natančnost diskretizacije, vendar smo omejeni s prostorsko in časovno zahtevnostjo takšnega postopka, zato moramo sprejeti ustrezen kompromis. Naš model smo diskretizirali z natančnostjo 0,1 mm in kot rezultat dobili model jajčnika z jajčnimi mešički v matriki 3D z dimenzijami 400 x 200 x 300 točk.

3.2.2 Model ultrazvočne sonde

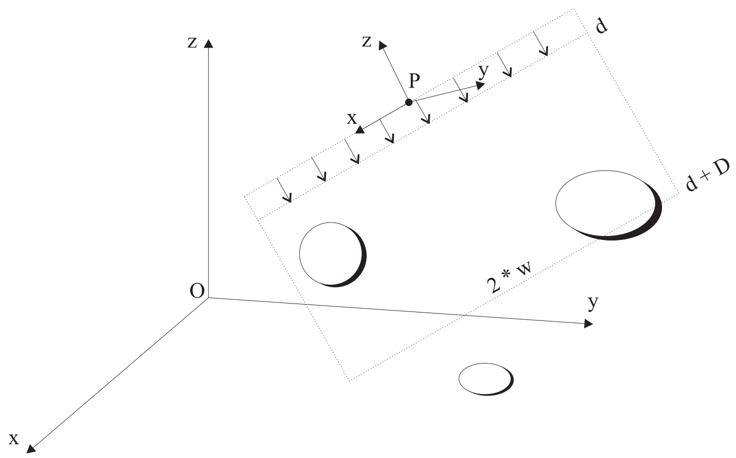
Ultrazvočno sondo, njen najpomembnejši del je ultrazvočni pretvornik - ploščica, uporabljamo tako za oddajnik kot za sprejemnik ultrazvoka. Sonda oz. pretvornik, ki jo vzbudimo s kratkotrajnim napetostnim impulzom (nekaj nanosekund), ima sposobnost oddajanja ultrazvočnih valov. S sondo snop ultrazvočnih valov usmerimo v želeno smer. Ko ultrazvočni valovi potujejo skozi telo, se na posameznih telesnih strukturah odbijajo (del se odbije, del potuje naprej). Odbite ultrazvočne valove s sondo zaznamo, sprejeta energija pa se pretvori v napetostni signal, ki ga zatem lahko prikažemo na zaslonu [Pot98]. Rezultat pregleda z ultrazvokom je dejansko prerez skozi telesno strukturo – v našem primeru skozi jajčnik.

Tvorjenje prerezov

Model ultrazvočne sonde oz. pregleda smo zasnovali iz dveh zaporednih korakov. V prvem koraku s sondo naredimo le prerez skozi jajčnik, v drugem koraku pa dobljenemu prerezu dodamo efekt ultrazvočnega šuma. Najprej si oglejmo prvi korak – tvorjenje prerezov.

Na začetku sondo umestimo v prostor ter določimo njeno orientacijo. Sondo postavimo naključno za L cm nad jajčnik (naključno v smeri x in y). Oddaljenost L smo dobili po statistični analizi množice realnih ultrazvočnih posnetkov jajčnikov (L = 3,05 cm). V središče sonde postavimo nov koordinatni sistem z enako orientacijo, kot jo ima osnovni. Zatem naključno spremenimo orientacijo koordinatnega sistema sonde (gre za rotacijo sistema). Vsako rotacijo koordinatnega sistema lahko popolnoma popišemo s tremi Eulerjevimi koti [BS78, Eul], in sicer s θ (rotacija sistema okoli osi y), ψ (rotacija okoli osi x) ter s ϕ (rotacija okoli osi z). Kota θ in ψ izbiramo naključno iz intervala $[\frac{-\pi}{12}, \frac{\pi}{12}]$, kot ϕ pa med 0 in π.

Snemanje s sondo izvajamo tako, da z ravnino XZ koordinatnega sistema sonde prerežemo vse, kar se nahaja pred sondo. V sliki 3.2 je shematično prikazano snemanje s sondo P.



Slika 3.2: Shematičen prikaz snemanja jajčnika (trije mešički) z našim modelom sonde (P). Smer snemanja označuje množica vektorjev.

Zaradi fizikalnih lastnosti ultrazvoka je aktivno področje snemanja sonde omejeno [HA92, Pot98]. Sonda lahko namreč posname le področja, ki so od nje oddaljena vsaj za d cm in maksimalno za $d + D$ cm (oddaljenost v smeri oddanih ultrazvočnih valov). Kadar snemamo, smo omejeni tudi s širino področja, ki ga preglejujemo (w). Obe omejitvi, globino in širino področja, smo upoštevali pri tvorjenju prerezov. Prerez skozi jajčnik je tako dejansko pravokotne oblike, kot je dobro vidno na sliki 3.2. Le-ta se razširja od $-d$ do $-d - D$ cm v smeri osi z ter od w do $-w$ cm v smeri osi x, merjeno glede na koordinatni sistem sonde.

Na koncu rezultirajoči prerez v obliki pravokotnika še diskretiziramo (korak 0,2 mm). Parameter d smo prav tako dobili po analizi realnih ultrazvočnih slik (d = 0,82 cm), medtem ko smo parametra D in w izbrali glede na želeno velikost rezultirajočega preseka (256 x 256 pikslov). D in $2 * w$ smo postavili na 5,12 cm.

Ko posnamemo en prerez, lahko nadaljujemo s snemanjem, in sicer tako, da premaknemo sondo za dy cm po osi y (osnovni koordinatni sistem) ter zajamemo nov prerez. Pri tem premiku dopuščamo tudi majhno rotacijo koordinatnega sistema sonde, s tem dejansko modeliramo tresljaj roke, ki se pojavi, ko ekspert opravlja ultrazvočni pregled v realnosti (maksimalen dovoljen kot pri tresljaju je empirično omejen na $\frac{\pi}{90}$).

Dodajanje efekta ultrazvočnega šuma

Najprej pojasnimo izraz efekt ultrazvoka. Pod tem izrazom razumemo vse komponente, ki so tipične za ultrazvočne slike. Najpomembnejše med njimi so: a) koncentrični odbojni valovi, ki se pojavljajo kot svetli polkrožni valovi od enega roba slike do drugega, b) pegasti šum, ki daje ultrazvočni sliki zrnasto strukturo in c) neizraziti robovi objektov.

Po simulaciji snemanja z našim modelom sonde dobimo množico prerezov skozi jajčnik – slik s temnimi ovalnimi strukturami na belem ozadju. Če želimo dobiti umetne slike, ki bodo čim bolje posnemale realne ultrazvočne slike jajčnikov, moramo prerezom dodati efekt ultrazvočnega šuma (vključno s pegastim šumom) in koncentričnih odbojnih valov.

Gotovo je najzahtevneje posnemati pegasti šum, ki se pojavlja v ultrazvočnih slikah. Pegasti šum je ime za kaotičen in neurejen vzorec, ki se nahaja v slikah [God85]. Pojavi se, kadar preglejujemo strukture, ki so manjše od valovne dolžine ultrazvoka. V literaturi obstajata dva modela za modeliranje pegastega šuma za ultrazvočne slike [KMPS94, KP92, SK97, EN96]: 1) množilen model šuma in 2) model šuma, odvisen od signala.

1. Množilen model šuma
   Ta model velja za ultrazvočne B-slike, ki jih dobimo takoj po zaznavanju signala na ultrazvočni sondi brez nadaljnjih obdelav. Pokazano je bilo [KMPS94], da takšen zajet signal x lahko obravnavamo kot Rayleighovo naključno spremenljivko s funkcijo gostote verjetnosti
   $$\mathrm{p}(x) = \frac{x}{\sigma^2}\, e^{-\left(\frac{x^2}{2\sigma^2}\right)}, \quad x > 0. \tag{3.2}$$

   Pegasti šum lahko tako modeliramo kot

   $$x = s\, n, \tag{3.3}$$

kjer sta x detektiran ter s originalen signal, člen n pa je šum, ki je statistično neodvisen od s [KP92, KMPS94]. Povprečna vrednost m naključne spremenljivke x ter njena varianca var sta določena kot [KP92]:

$$\begin{aligned} m &= \sigma\sqrt{\pi/2}\,, \\ var &= \sigma^2\Big(\frac{4-\pi}{2}\Big). \end{aligned} \tag{3.4}$$

2. Model šuma, odvisen od signala
   Opisani model pa velja za prikazane ultrazvočne B-slike. Takšne slike so šle skozi zahtevno obdelavo, preden so bile prikazane (logaritemska kompresija, filtriranje z nizkim in visokim filtrom itd.), kar močno oteži modeliranje statistike prikazanih slik. Realističen model je [KMPS94]

$$x = s + \sqrt{s}\,n. \tag{3.5}$$

   Za šum n se predpostavlja, da je Gaussova naključna spremenljivka s povprečno vrednostjo 0 in standardnim odklonom σ.

Na osnovi obeh opisanih modelov smo poskušali umetno dodajati efekt ultrazvočnega šuma. V obeh primerih smo dobili nezadovoljive rezultate, ki se niso približale realnim ultrazvočnim slikam. Tak rezultat razlagamo z dejstvom, da nismo uspeli določiti točne prostorske porazdelitve in velikosti peg. Zato smo se odločili uporabiti simulator ultrazvočnih sistemov.

Uporabili smo program Field avtorja Jensena [Jen, JM97, Jen96]. S tem programom za simuliranje ultrazvočnih sistemov lahko tvorimo poljubne geometrije pretvornikov znotraj sonde (linearna polja pretvornikov, okrogle itd.), modeliramo različne sheme fokusiranja (dinamično fokusiranje) in dušenja ultrazvoka (frekvenčno odvisno in neodvisno) itd. Z njim lahko izračunamo prostorski impulzni odziv za poljuben ultrazvočni sistem, kreiramo pa lahko tudi 2D ultrazvočne slike v odvisnosti od objektov, ki jih pregledujemo.

Program Field uporablja za določanje ultrazvočnega polja koncept prostorskega impulznega odziva in teorije linearnih sistemov [Jen91, JS92]. Prostorski impulzni odziv podaja vrednost oddanega ultrazvočnega polja v specifični točki v prostoru kot funkcijo časa, če je bil pretvornik vzbujen z Diracovo funkcijo delta. Polje za katero koli vzbujanje pa lahko potem izračunamo s konvolucijo prostorskega impulznega odziva z vzbujevalno funkcijo. Prostorski impulzni odziv se seveda spreminja kot funkcija položaja točke glede na pretvornik. Prejeto ultrazvočno polje na pretvorniku se pa določa po principu akustične recipročnosti. Prostorski impulzni odziv majhne oscilirajoče sfere v prostoru je enak prejetemu odzivu za sferični val, oddan iz te točke. Skupni prejeti odziv torej dobimo kot konvolucijo pretvornikove vzbujevalne funkcije s prostorskim impulznim odzivom oddajajoče ploskve ter s prostorskim impulznim odzivom sprejemne ploskve pretvornika. Pri tem moramo upoštevati še elektro-mehansko prenosno funkcijo za pretvornik.

Kadar s programom Field tvorimo 2D ultrazvočne slike, moramo izbrati množico neodvisnih točk, ki raztresajo ultrazvok (*scatter*), ter jim določiti ustrezno amplitudo. Kvalitetna ultrazvočna slika je rezultat pravilne izbire števila točk in njihovih relativnih amplitud [JM97, Jen96].

Program Field smo uporabili za dodajanje učinkov ultrazvočnega šuma prerezom jajčnika. Ker smo prereze jajčnikov dobili kar s sekanjem ravnine XZ sonde in jajčnika, smo pri simuliranju uporabili sondo z linearnim poljem pretvornikov (24 elementov, širina posameznega elementa je 0,5 mm, višina 5 mm, razmik med elementi pa 0,1 mm). Centralno frekvenco pretvornika smo nastavili na 3 MHz (v ginekologiji se uporablja ultrazvok od 2 do 5 MHz [Pot98]), uporabili pa smo tudi fiksen fokus tako pri oddajanju kakor tudi pri sprejemanju (fokus je 70 mm). Celotno področje pregledovanja (5,12 cm) smo razdelili na 128 linij. Te nastavitve dokaj dobro popisujejo realen ultrazvočni sistem na področju ginekologije.

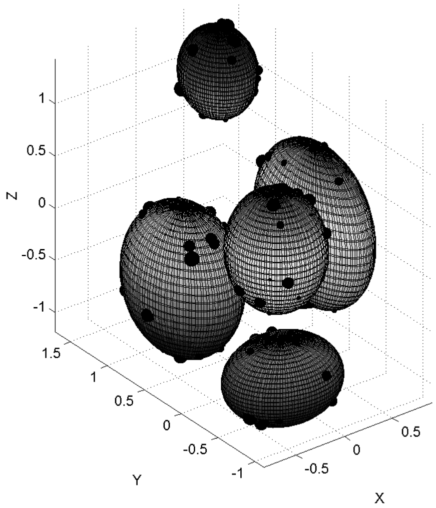
Ultrazvočne slike smo tvorili s pomočjo 4000 točk, ki raztresajo ultrazvok. Točke smo izbrali popolnoma naključno. Amplitude posameznih točk smo določili glede na prerez jajčnika, in sicer smo točkam, ki se nahajajo na področju mešičkov (temne ovalne strukture), priredili amplitudo 0, ostalim točkam pa smo določili amplitudo po Gaussovi porazdelitvi s povprečno vrednostjo 0 in standardnim odklonom 1. Točke smo razmestili enakomerno po vsej sliki.

3.2.3 Rezultati in razprava

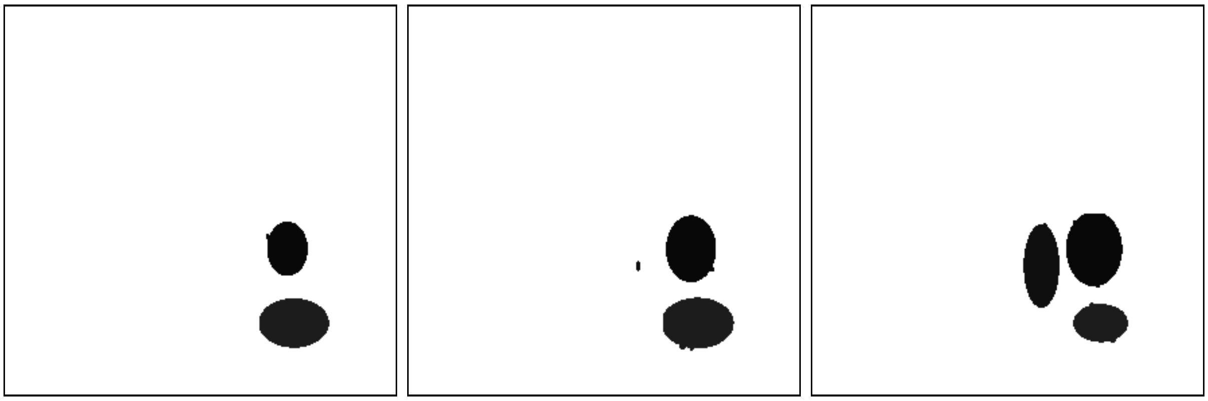
V nadaljevanju bomo predstavili rezultate, ki smo jih dobili z uporabo vpeljanih modelov jajčnika in sonde.

Na sliki 3.3 je prikazan dobljen model jajčnika s petimi jajčnimi mešički. Dobro lahko opazimo izrastke na posameznih mešičkih. Mešički se ne prekrivajo niti se ne dotikajo. Največji mešiček meri 1,52 cm v dolžino, najmanjši pa 0,84 cm.

Ta model jajčnika smo zatem v simulaciji pregledali z ultrazvočno sondo. Sonda vrača prereze skozi jajčnik. Uporabili smo nerotirano sondo, razmik med posameznimi prerezi je bil 0,1 mm, tresenja nismo dovolili. Po celotnem pregledu jajčnika smo dobili 284 prerezov (ponovitve prerezov brez mešičkov nismo upoštevali). Na sliki 3.4 so prikazani trije prerezi skozi jajčnik. Prerezi so dobljeni s premikanjem sonde od leve strani jajčnika proti desni. Razmik med prvima dvema prerezoma je 1 mm, med drugim in tretjim pa 3 mm.

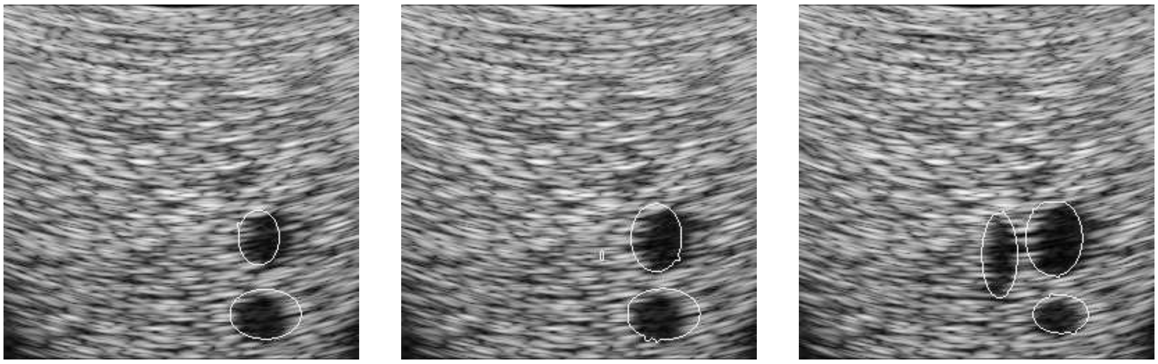


Slika 3.3: Model jajčnika z jajčnimi mešički.



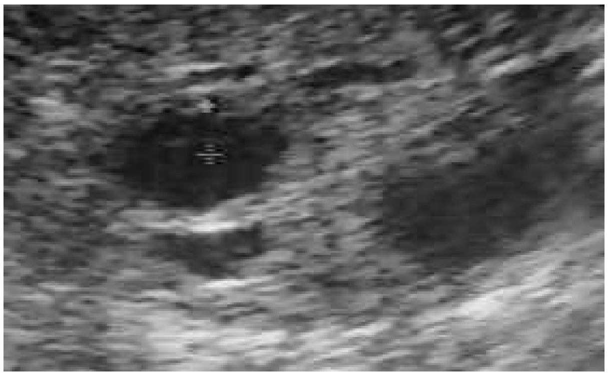
Slika 3.4: Trije prerezi skozi jajčnik. Razmik med levim in srednjim prerezom je 1 mm, med srednjim in desnim pa 3 mm. Temne strukture predstavljajo jajčne mešičke.

Tretji simulacijski korak doda k prerezom s slike 3.4 efekt ultrazvočnega šuma. Uporabili smo program Field. Rezultirajoče umetne ultrazvočne slike jajčnika z mešički so prikazane na sliki 3.5. Z belo črto so obkroženi originalni objekti pred dodajanjem ultrazvočnega efekta. Na teh slikah lahko jasno opazimo homogen vzorec peg, ki je dobro viden po celotni sliki, tudi na področju mešičkov. Tudi robovi mešičkov niso več izraziti, ampak so delno zapacani, kar je ena izmed neželenih lastnosti ultrazvoka.

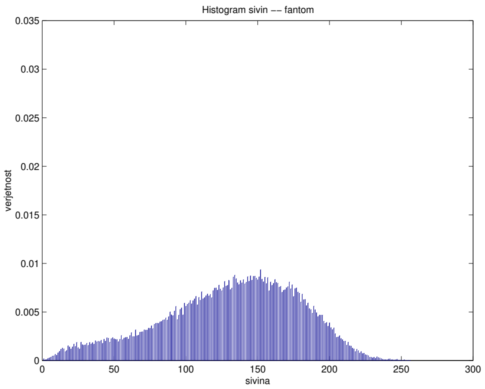


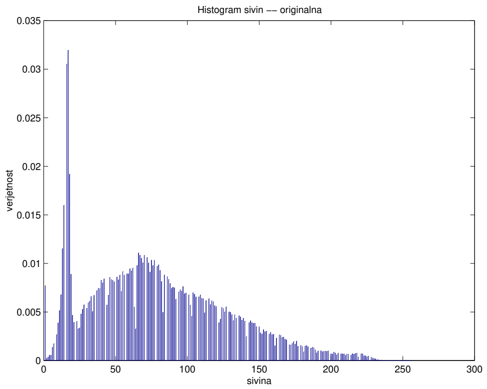
Slika 3.5: Umetno tvorjene ultrazvočne slike jajčnika z mešički. Sliki 3.4 dodan ultrazvočni efekt. Z belo črto je označen originalen mešiček pred dodajanjem efekta ultrazvoka.

Umetne ultrazvočne slike jajčnika s slike 3.5 primerjajmo z realnim ultrazvočnim posnetkom jajčnika s petimi mešički (slika 3.6). Opazimo veliko podobnost med obema posnetkoma. To trditev podkrepimo tudi z analizo histrogramov sivin obeh posnetkov, kjer prav tako vidimo, da oba histograma izkazujeta približno enako Gaussovo porazdelitev sivin (od prave Gaussove porazdelitve odstopata zaradi mešičkov), razlika je le v tem, da imata oba histograma različni povprečni vrednosti sivin. Na sliki 3.7 sta prikazana oba histograma, na levi strani je pripadajoč histogram za fantoma, na desni pa za celoten originalen ultrazvočni posnetek (ne le za izsek).



Slika 3.6: Izsek iz realnega ultrazvočnega posnetka jajčnika s pet mešički.





Slika 3.7: Histogram sivin. Na levi strani je pripadajoč histogram za fantoma, na desni pa za realen ultrazvočni posnetek.

Razlike, ki jih opazimo v histogramu sivin pri fantomu in originalnem posnetku, so zagovo nastale, ker naš model jajčnika in ultrazvočne sonde nista popolna. V originalnem ultrazvočnem posnetku so ob mešičkih še druge temne strukture, ki jih v umetnih slikah nismo modelirali – zato se pojavi tudi opazna konica pri črnem nivoju v histogramu sivin. Oba modela pa je možno zelo enostavno dopolniti. Tukaj mislimo predvsem na model jajčnika, kjer bi lahko preprosto dodali še druge strukture, kot so npr. žile – žilo bi lahko modelirali kot skupek tankih valjev.

Neoptimalnost rezultatov našega algoritma izhaja tudi iz dejstva, da smo morali pri nekaterih parametrih, potrebnih za delovanje programa Field, narediti kompromis. Tako smo celotno področje pregledovanja razdelili na 128 linij, kar pomeni, da smo morali vsak drugi stolpec v rezultirajoči sliki interpolirati (fantomi so velikosti 256 x 256 pikslov). Razlog za to je bil v izjemni časovni zahtevnosti postopka za dodajanje efekta ultrazvoka. Program Field je za celotni postopek dodajanja efekta ultrazvoka k eni sliki z nastavljenimi parametri potreboval na delovni postaji HP 715 / 100 XC s 128 MB delovnega pomnilnika približno 60 minut. S povečanjem števila linij in točk, ki raztresajo ultrazvok, se je časovna zahtevnost eksponentno povečala. Zaradi obdelovanja izredno velike količine slik– en pregled vsebuje tipično okrog 300 slik – si takšnega povečanja nismo mogli privoščiti.

Kljub vsemu pa vidimo, da dobljeni fantomi jajčnikov z mešički izkazujejo veliko mero podobnosti z realnimi ultrazvočnimi posnetki. To hkrati pomeni, da jih lahko uporabimo tudi za verificiranje algoritmov razpoznavanja v ultrazvočnih posnetkih jajčnikov. Verificiranje algoritmov razpoznavanja na kvalitetnih fantomih je zaželeno, saj s tem, ko so znane vse razmere, ki so nastopale pri zajemanju slike, lažje ovrednotimo učinkovitost posameznih algoritmov, hkrati pa lahko primerjamo več podobnih algoritmov med seboj.

3.3 Razpoznavanje jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov

V tem podpoglavju bomo podrobno opisali nov algoritem za popolnoma avtomatsko razpoznavanje jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov. Dejansko gre za segmentacijski algoritem, ki poišče mešičke le na posamezni 2D sliki.

Čeprav je algoritem res zasnovan za segmentiranje mešičkov, je v svoji zasnovi popolnoma splošen. Z njim je možno poiskati temne jajčaste objekte v kakršnikoli sliki. Pričakujemo lahko celo, da bodo rezultati segmentiranja na drugih področjih še veliko boljši kot pri ultrazvočnih posnetkih, ki so izredno šumni.

3.3.1 Algoritem

Jajčni mešički so sferične strukture, napolnjene s tekočino. Interna regija, ki je napolnjena s tekočino, je obdana s t. i. notranjim zidom [KS98]. Ultrazvočne slike jajčnikov podajajo ravninske prereze skozi mešičke. Takšne slike so okarakterizirane z refleksijskimi odboji in informacijami o robovih, ki so šibki in nezvezni. Zato so tradicionalne tehnike za detektiranje robov (npr. Sobel, Prewitt), kadar jih uporabimo v ultrazvočnih slikah, občutljive na te lažne odzive [Bov88]. Velika mera dodanega pegastega šuma je zato glavni razlog, da segmentiranje le na osnovi informacij o robovih ni posebej uspešno.

Mešički so s tekočino napolnjene strukture, kar povzroči, da se na ultrazvočnih posnetkih pojavljajo kot temne ovalne strukture. Vsi mešički izkazujejo podobno odmevno teksturo tekočine, ki je bolj ali manj izrazito temnejša kot okolica. Mešičke lahko zato obravnavamo kot temne homogene regije.

Iz tega sledi glavna ideja našega algoritma: poiskati homogene regije v slikah, ki jih zatem z enim od segmentacijskih algoritmov privedemo do pravega roba objekta. V fazi poobdelave pa le še odstranimo neustrezajoče objekte. V nadaljevanju bomo podrobno opisali posamezne korake algoritma.

3.3.2 Določanje homogenih regij

V tem koraku bomo določili začetni približek za mešičke. Vsak mešiček ima znotraj sebe vsaj eno homogeno regijo. Ponavadi se te homogene regije nahajajo v sredini mešička. Pod izrazom homogen razumemo, da imajo piksli, ki tvorijo homogeno regijo, podobno vrednost sivin (podobno statistiko sivin).

Za identificiranje homogenih regij smo uporabili dve neodvisni proceduri. Rezultate obeh procedur smo na koncu zlili. Prva procedura izkorišča dejstvo, da so mešički temni. Zato v tem koraku preučujemo vrednosti sivin v sliki. Vrednosti sivin opazujemo v filtrirani sliki. Uporabili smo korak predobdelave kot v [KS98]. Filtriranje poteka po naslednji shemi: najprej originalno sliko zgladimo z adaptivnim medianim filtrom. Za piksle, ki so temnejši od praga T_1, uporabimo jedro medianega filtra večje velikosti (11 x 11 pikslov), manjše jedro pa za piksle, svetlejše od T_1 (5 x 5 pikslov). Glajenje opravimo zaporedoma dva-krat. To zagotavlja, da so mešički zglajeni bolj temeljito kot regije z visoko intenzivnostjo. V nasprotju s [KS98], kjer je treba prag T_1 določiti zelo natančno, za našo proceduro zadostuje že približno določen prag. Prag T_1 postavimo na vrednost povprečne sivine v originalni digitalni sliki I_k.

Iz tako filtrirane slike dobimo s pragovno operacijo grobe približke za mešičke. Ker pričajujemo, da bodo približki res grobi, lahko tudi prag T_2 določimo zelo približno. Prag T_2 postavimo na vrednost povprečne sivine, zmanjšane za en standardni odklon sivin v zglajeni sliki. Filtriranje in pragovna operacija lahko povzročita, da pride do medsebojnega zlitja bližnjih mešičkov ali do zlitja mešička z drugimi okoliškimi strukturami (npr. s krvno žilo). Zato smo kot zadnji korak uporabili binarno segmentacijsko metodo razvodja na osnovi Evklidovega zemljevida razdalj (*watershed using Euclidean distance map*) [Rus95]. S tem korakom dosežemo grobo ločitev staknjenih objektov.

Druga procedura uporablja za merjenje podobnosti standardni odklon sivin pikslov. Najprej smo za vsak piksel v njegovi okolici izračunali standardni odklon sivin. Velikost okolice, 11 x 11 pikslov, smo določili iz premera najmanjšega možnega mešička. V ultrazvočnih slikah tipično 50 do 60 pikslov ustreza 1 cm, minimalni premer mešička pa je med 2 in 3 mm, zato velikost okolice 11 x 11. Piksel označimo kot homogen, če njegov standardni odklon ne preseže praga T_3, hkrati pa je njegova sivina manjša od povprečne sivine v sliki. Prag T_3 je postavljen na polovico povrečnega standarnega odklona v sliki (ponavadi se giblje okrog 10). To je vsekakor zelo ohlapen kriterij. Če predpostavimo, da so sivine porazdeljene po Gaussovi porazdelitveni funkciji, kar velja za prikazane ultrazvočne slike [KMPS94], potem je piksel $\mathbf{p}$ homogen, če je sivina vseh njegovih sosedov znotraj intervala $[I_k(\mathbf{p}) - 3T_3, I_k(\mathbf{p}) + 3T_3]$, kjer z $I_k(\mathbf{p})$ označujemo vrednost sivine piksla.

Izkaže se, da s takšnima kriterijema zelo dobro določimo večino mešičkov. Problemi se pojavijo pri "svetlih" mešičkih. Ti mešički imajo sivino blizu povprečne sivine v sliki in ponavadi ležijo pri dnu slike, kjer je echogeničnost že izredno velika. Njihova notranjost ni izrazito homogena, pa tudi standarni odklon sivin je že velik. Zato smo za te izjemne primere dodali še en korak. Nad dvojno zglajeno sliko (prva procedura) opravimo še eno pragovno operacijo. Prag nastavimo na povprečno sivino v zglajeni sliki. Ob tako visokem pragu se skoraj vse regije zlijejo med seboj, izolirani ostanejo le "svetli mešički". Torej odstranimo vse regije, ki so dotikajo roba slike, saj se zlijejo, preostale regije pa dodamo k že razpoznanim mešičkom.

Kadar aproksimiramo mešičke z opisanima procedurama, dobimo tudi še mnogo drugih, neželenih regij (npr. regije znotraj maternične sluznice). Čim več teh neželenih regij želimo odstraniti že v koraku predobdelave. Najprej odstranimo vse regije, ki imajo ploščino manjšo od praga T_4 ($T_4 = 50$, kar je približno četrtina minimalne velikosti mešička). Te majhne regije razumemo kot regije, nastale zaradi pegastega šuma. Zatem odstranimo še vse regije, ki se dotikajo roba slike. S tema preprostima praviloma odstranimo večino neželenih regij.

Preostale regije zatem le še identificiramo (označimo) ter zapolnimo morebitne luknje v njih.

3.3.3 Rast regij

Identificirane homogene regije predstavljajo grobi začetni približek za mešičke. Za nadaljnje obdelovanje teh regij se nam ponujata dve segmentacijski metodi: aktivne krivulje in rast regij. Dobro je znano [PZ98], da aktivne krivulje potrebujejo že zelo natančno določen približek objekta, kar pa naša procedura za določanje homogenih regij ne zagotavlja v vseh primerih. Problem, na katerega naletimo pri kačah, je tudi določitev ustreznih parametrov za njihovo delovanje, kar bi v našem primeru predstavljalo velik problem (za vsak mešiček poiskati ustrezne). Najverjetneje bi s tako modificirano segmentacijsko metodo izgubili na njeni splošnosti.

Zato smo za segmentacijsko metodo izbrali metodo rasti regij (*region growing*) [Pot98, SHB94, Rus95]. Ideja, ki jo želimo realizirati z metodo rasti regij, je naslednja: širi homogeno regijo, dokler ne trčiš ob rob objekta. Na rast lahko gledamo tudi kot na polnjenje globeli, ki jo polnimo tako dolgo, dokler ne prispemo na vrh hriba. Ta hrib določa zaključni pogoj rasti regije. Poglejmo si sedaj ta korak algoritma pobliže.

V sliki imamo ponavadi več začetnih homogenih regij. V tej fazi je pomemben vrsti red regij – katero regijo bomo razširjali prej. Večjo prioriteto damo večjim začetnim regijam. Regije z večjo ploščino bodo obdelane prej. Ta korak je pomemben le v primeru, če ostane še nekaj neželenih regij (ponavadi majhnih) blizu mešičkov. Vsako začetno regijo zatem razširjamo (pustimo rasti) po naslednjem postopku.

Označimo začetno homogeno regijo z R_0, njeno povprečno sivino z $m(R_0)$ in standardni odklon sivin s $\sigma(R_0)$. Pri rasti iterativno dodajamo objektu piksle z zunanje meje O [Gam93]. Zunanja meja O regije je množica pikslov, ki so sosedi regije in hkrati niso njeni člani. V n-ti iteraticiji označimo z O_n zunanjo mejo regije R_n, ki raste (začetna regija je R_0). Rast izvajamo z analiziranjem pikslov iz O_n. Če piksel $\mathbf{p} \in O_n$ izpolnjuje oba kriterija za zlitje, ki ju bomo v nadaljevanju natančno opisali, ga zaznamujejo kot potencialnega kandidata za zlitje. Prvi kriterij (enačba (3.6))

$$\left| I_k(\mathbf{p}) - m(R_0) \right| \leq \alpha\, \sigma(R_0) \tag{3.6}$$

zahteva, da je sivina potencialnega kandidata za zlivanje statistično blizu začetni regiji R_0, kjer je α prednastavljena vrednost.

Ponavadi gradient zavzame visoke vrednosti po velikem delu meje regije. Zato lahko zaključek rasti nadziramo tudi na osnovi povprečnega gradienta v regiji. Rast regije lahko zato zaključimo tudi, ko zasledimo očitno povečanje v velikosti gradienta. Žal pa v ultrazvočnih slikah takšno zaznavanje ni preprosto – robovi namreč niso izraziti.

V [LOG99] so študirali lastnosti meje objektov v ultrazvočnih slikah. Ob gradientu intenzivnosti, ki ima ponavadi omejeno uporabnost, meja okoli objekta namreč ni povsod okarakterizirana z visokim gradientom, zato predlagajo še uporabo neke vrste teksturne segmentacije. Vendar pa lastnosti šuma nakazujejo, da je smiselno opazovati le statistike do drugega reda. Avtorji so zato predlagali linearno kombinacijo 1D gradienta vzdolž iskalnih linij B-kače ter parametra, določenega iz statistike prvega in drugega reda. Uteži pri linearni kombinaciji se adaptivno določajo med procesom segmentacije zaporedja slik. Rezultati so pokazali, da segmentiranje s pomočjo B-kač in z obema kriterijema, z gradientom in s teksturami, daje boljše rezultate kot le segmentacija z gradienti.

Zato smo kombinirali informacijo o gradientu z informacijo o teksturi. Preprosto teksturno statistiko tex za piksel $\mathbf{p}$ definiramo kot razmerje med povprečno sivino in standarnim odklonom v njegovi okolici (11 x 11 pikslov). Statistika tex ima visoke vrednosti v homogenih regijah in nizke vrednosti blizu robov oz. v šumnih regijah. Statistiko tex lahko prav zaradi tega obravnavamo kot mero za razmerje med signalom in šumom (SNR). Znano je, da je povprečno razmerje SNR v regijah s popolnoma razvito pego enako 1,91 [EN96]. To dejansko pomeni, da lahko štejemo piksle, ki imajo SNR blizu te vrednosti, kot potencialne piksle z robov. S to informacijo pa lahko dopolnimo informacijo o gradientu. Dobimo t. i. obtežen gradient, ki ga izračunamo kot produkt med velikostjo gradienta in inverzno vrednostjo statistike tex. Fukcija $f(z) = 1/z$ ima probleme s stabilnostjo, ko gre z proti 0. Zato smo se raje odločili za eksponentno funkcijo kot pa za čisto inverzno funkcijo. Končna oblika obteženega gradienta za piksel $\mathbf{p}$ je:

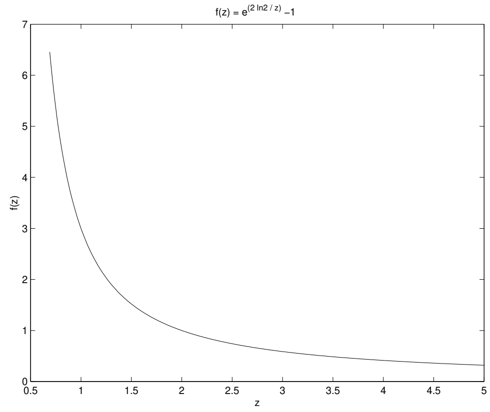
$$grad(\mathbf{p}) = \left\|\nabla I_k(\mathbf{p})\right\|\left(e^{\frac{G}{tex(\mathbf{p})}} - 1\right). \tag{3.7}$$

Parameter G določimo iz dveh robnih pogojev: 1) za vrednost $tex \to \infty$ naj eksponentna funkcija vrača vrednosti okrog 0, 2) pri $tex = \lceil 1,91 \rceil$ pa naj bo vrnjena vrednost enaka 1. Parameter G je potemtakem $2\,ln(2)$. Na sliki 3.8 je prikazan graf funkcije $f(z) = e^{\frac{2\,ln2}{z}} - 1$, s katero pomnožimo vrednost gradienta ∇I.

Ob kriteriju iz enačbe (3.6) mora piksel $\mathbf{p} \in O_n$ zadostiti tudi drugemu kriteriju:

$$\left|grad(\mathbf{p}) - m(grad(R_0))\right| \leq \alpha\, \sigma(grad(R_0)), \tag{3.8}$$

kjer sta $m(grad(R_0))$ in $\sigma(grad(R_0))$ povprečna vrednost in standardni odklon obteženega gradienta v začetni regiji R_0. Parameter α postavimo na 2. Če predpostavimo, da so



Slika 3.8: Graf funkcije $f(z) = e^{\frac{2\,ln2}{z}} - 1$

.

tudi obteženi gradienti porazdeljeni po Gaussovi porazdelitveni funkciji, kot so sivine, potem izbira takšnega α pomeni, da so praktično vsi statistično podobni piksli vključeni (natančno 95,46 %).

Če piksel zadosti obema kriterijema (enačbi (3.6) in (3.8)), potem ga zaznamujemo kot potencialnega kandidata za zlitje z regijo R_n. Potencialni kandidat $\mathbf{p} \in O_n$ pa je dejansko združen z regijo R_n, če ima vsaj 4 sosede iz R_n ali pa zaznamovane kot potencialne kandidate iz O_n (gledamo v 3 x 3-okolici). S tem korakom kontroliramo kompaktnost regije. S tem preprečimo, da bi regija po tankem črnem pasu, širokem nekaj pikslov, zbežala iz svojih pravih mej.

Pri rasti se lahko tudi zgodi, da potencialen kandidat za zlitje $\mathbf{p} \in O_n$ že pripada drugi regiji P_m. V tem primeru skušamo zliti regiji R_n in P_m (začetna regija je P_0). Hipotezo, da sta si obe regiji statistično podobni, preverimo z dvema kritičnima členoma CT_1 in CT_2, kjer

$$CT_1 = \frac{|m(R_0) - m(P_0)|}{\sigma(R_0 \cup P_0)} \tag{3.9}$$

in

$$CT_2 = \frac{|m(grad(R_0)) - m(grad(P_0))|}{\sigma(grad(R_0 \cup P_0))}. \tag{3.10}$$

Hitro je možno pokazati, da sta CT_1 in CT_2 porazdeljena po Studentovi t porazdelitvi [RZ89]. Prag za podobnost T_5 enolično določimo z izbiro intervala zaupanja. Interval zaupanja smo nastavili na 0,05, izračunan prag za podobnost pa je 1,66. Če sta oba kritična člena manjša od praga T_5, potem regiji R_n in P_m zlijemo.

Rast regije R končamo, če je po n-ti iteraciji R_n enaka R_{n-1}.

3.3.4 Izločanje mešičkov

V tem koraku poobdelave sprejmemo končno odločitev o tem, ali je zrastla regija mešiček ali ne. Odločitev o regiji sprejmemo na osnovi odločitvenih pravil. Le-ta smo določili na osnovi znanja o problemu. Odločitvena pravila so do določene mere podobna pravilom v [PZK97, Pot98].

V začetku odstranimo vse regije, ki so se razrastle do mej slike. V tem primeru predpostavimo, da so te regije ali iz področja žil ali sluznice oz. drugih okoliških tkiv, ter jih zato odstranimo. Ta korak je v splošnem vedno veljaven, saj mešički niso nikoli čisto ob meji slike. Drugih predpostavk o samem položaju jajčnika, da se npr. nahaja v sredini slike ipd., ne smemo sprejeti. Zato odstranimo še vse regije, ki imajo ploščino manjšo od 220. Prag za še minimalno dovoljeno ploščino izračunamo iz minimalnega premera mešička (2 do 3 mm).

Mešički so strukture ovalne oblike. Zato smo kot naslednje odločitveno pravilo izbrali neke vrste mero kompaktnosti regije. Definirali smo parameter b, in sicer kot razmerje med dejansko ploščino mešička ter ploščino njegove obdajajoče škatle (*bounding box*). Če je razmerje b večje od praga 0,5, potem regijo označimo kot potencialni mešiček, sicer pa jo odstranimo.

Vsi mešički ležijo znotraj jajčnika. To hkrati pomeni, da mešički ne morejo biti poljubno razstreseni po celotni sliki, ampak se lahko nahajajo le na omejenem območju slike. Kar z drugimi besedami pomeni, da ležijo skupaj. To je zadnje pravilo, ki smo ga implementirali. Za določanje položaja opazujemo težišče regije. Najprej vse preostale regije, potencialne mešičke, sortiramo na osnovi treh različnih kriterijev: po ploščini (sortiramo od največjega proti najmanjšemu), kompaktnosti (od najmanjše proti največji) ter po parametru b (od največjega proti najmanjšemu). Regijo, ki se je po teh sortiranjih najbolje odrezala (dosegla najvišja mesta po vseh kriterijih), pa sedaj dejansko označimo kot mešiček. Izračunamo globalno težišče vseh pravih mešičkov, kar pomeni dejansko težišče jajčnika. Ker imamo trenutno izbran le en mešiček, zato globalno težišče ustreza kar težišču te regije. V naslednjem koraku izberemo izmed preostalih potencialnih mešičkov mešiček, ki je po razdalji najbližji globalnemu težišču. Ta potencialni mešiček označimo kot pravi mešiček, ažuriramo globalno težišče – to je povprečje težišč vseh pravih mešičkov, ter nadaljujemo postopek. To iterativno dodajanje končamo, ko minimalna razdalja do globalnega težišča preseže prednastavljeno vrednost. Na osnovi znanih povprečnih velikosti jajčnika smo ta prag nastavili na 0,25 velikosti slike. Vse potencialne mešičke, ki jih ne dodamo, zavržemo.

Na koncu še vse razpoznane mešičke označimo in zapolnimo morebitne luknje v njih.

Podajmo opisani algoritem še v kompaktni obliki psevdokoda.

VHOD: Ultrazvočna slika jajčnika z mešički I_k.

IZHOD: Segmentirana slika S_k z razpoznanimi mešički.

I. Določi homogene regije:

1. *Dva-krat zaporedoma filtriraj I_k z adaptivnim medianim filtrom. Za adaptivni filter uporabi prag T_1, večje jedro ima velikosti 11 x 11 pikslov, manjše pa 5 x 5 pikslov. Filtrirano sliko označi s $Filt$.*
2. $Pomozna(\mathbf{p}) = \begin{cases} 1, & Filt(\mathbf{p}) \leq T_2 \\ 0, & \text{sicer} \end{cases}$.
3. *Segmentiraj $Pomozna$ z binarno metodo razvodja. Rezultat označi s Hom_1.*
4. *Za vsak piksel v I_k izračunaj standardni odklon v 11 x 11 okolici. Rezultirajočo matriko standardnih odklonov za vse piksle označi s $Pomozna$.*
5. $Hom_2(\mathbf{p}) = \begin{cases} 1, & Pomozna(\mathbf{p}) \leq T_3 \quad \wedge \quad I_k(\mathbf{p}) \leq T_1 \\ 0, & \text{sicer} \end{cases}$.
6. $Pomozna(\mathbf{p}) = \begin{cases} 1, & Filt(\mathbf{p}) \leq T_1 \\ 0, & \text{sicer} \end{cases}$.
7. *Iz $Pomozna$ odstrani vse regije, ki se dotikajo roba slike. Rezultat označi s Hom_3.*
8. $Hom = Hom_1 \vee Hom_2 \vee Hom_3$
9. *Iz Hom odstrani vse regije, ki imajo ploščino manjšo od T_4 ali, ki se dotikajo roba slike.*
10. *Vse preostale homogenih regij v Hom označi. Množica homogenih regij $H = \{H_0, H_1, .., H_k\}$.*

II. Razrasti regije:

1. *Uredi homogene regije v H po velikosti - od največje proti najmanjši.*
2. *Za vsako regijo $R_0 \in H$:*
   - a. *i = i + 1*
   - b. *Določi zunanjo mejo O regije R_{i-1}.*
   - c. *Za vsak piksel $\mathbf{p} \in O$ izračunaj kriterija v enačbah (3.6) in (3.8). Če piksel $\mathbf{p}$ izpolnjuje oba kriterija in če je izpolnjen kriterij kompaktnosti, dodaj $\mathbf{p}$ k R_{i-1}.*
   - d. *Označi ažurirano regijo z R_i.*
   - e. *Če se R_i prekriva z drugo regijo P, in če sta izpolnjena kriterija v (3.9) in (3.10), združi R_i in P v R_i.*

f. Če $R_{i-1} = R_i$, končaj rast za regijo R in dodaj R_i v sliko segmentiranih regij Seg.

III. Izloči mešičke:

1. *Odstrani vse regije iz Seg, ki se dotikajo roba slike ali imajo ploščino manjšo od 220 ali imajo indeks b manjši od 0,5. Označi rezultat s $Pomozna$.*
2. *Izmed preostalih potencialnih mešičkov v $Pomozna$ izberi najboljši mešiček glede na ploščino, kompaktnost in parameter b. Odstrani izbrani mešiček iz $Pomozna$ in ga dodaj v S_k.*
3. *Določi težišče jajčnika $\mathcal{T}_{jaj}$ kot težišče izbranega mešička.*
4. *Dokler $Pomozna \neq \{\}$:*
   - *a. V $Pomozna$ poišči regijo, ki je najbližja težišču $\mathcal{T}_{jaj}$. Če je razdalja do $\mathcal{T}_{jaj}$ večja od 0,25 velikosti slike, končaj postopek dodajanja regij.*
   - *b. Izbrano regijo odstrani iz $Pomozna$ ter jo daj v S_k.*
   - *c. Izračunaj globalno težišče $\mathcal{T}_{jaj}$ vseh regij v S_k.*
5. *Označi razpoznane mešičke v S_k in zapolni morebitne luknje v njih.*

Algoritem 2 Algoritem za avtomatsko razpoznavanje jajčnih mešičkov v ultrazvočnih posnetkih jajčnikov.

3.4 Rezultati in razprava

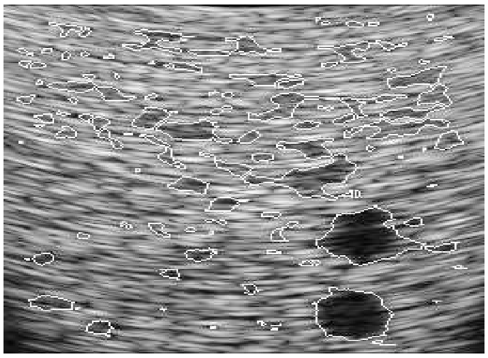
V nadaljevanju bomo prikazali in opisali dobljene rezultate segmentiranja z našim novim algoritmom na umetnih in realnih ultrazvočnih slikah.

Umetni ultrazvočni posnetki jajčnikov z jajčnimi mešički

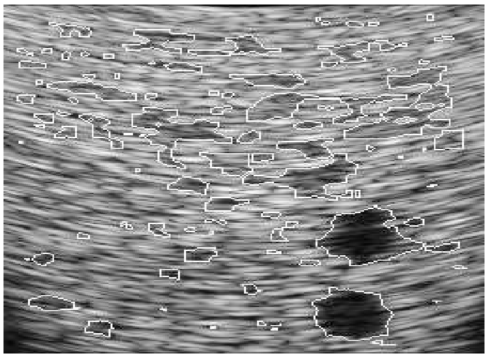
Najprej predstavimo rezultate segmentiranja na umetnih ultrazvočnih posnetkih jajčnikov.

Učinkovitost algoritma smo merili na 120 umetnih slikah jajčnika z mešički. Te slike smo dobili z algoritmom za tvorjenje umetnih ultrazvočnh slik jajčnikov (podpoglavje 3.2), zato smo imeli tudi natančno določene referenčne položaje mešičkov.

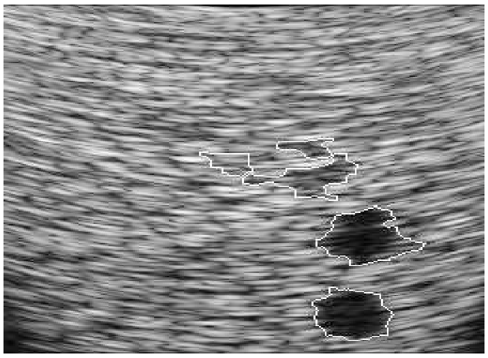
Prikažimo delovanje našega algoritma po korakih. Na sliki 3.9 so prikazane homogene regije, ki jih dobimo z našo proceduro za določanje homogenih regij. Te regije so položene na umetno ultrazvočno sliko jajčnika. Zatem uporabimo rast regij na teh slikah. Rezultirajoče zrastle regije so prikazane v sliki 3.10. Na sliki 3.11 pa so po uporabi procedure za izločanje mešičkov nad zrastlimi regijami prikazani razpoznani mešički. Mešički, ki so orisani z belo, so položeni na umetno sliko. Dobljene rezultate lahko primerjamo z referenčno sliko, kjer je z belo označen pravi položaj mešičkov (slika 3.12).



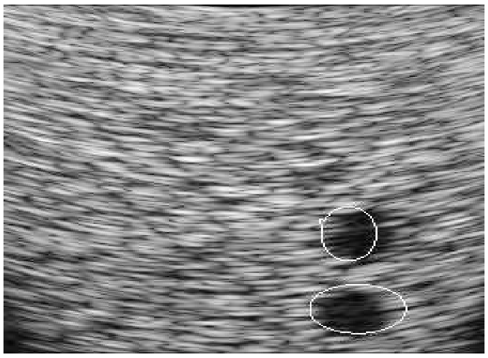
Slika 3.9: Homogene regije (v belem), položene na umetno ultrazvočno sliko.



Slika 3.10: Regije po koraku rasti.



Slika 3.11: Razpoznani mešički, položeni na umetno ultrazvočno sliko.



Slika 3.12: Pravi položaj mešičkov.

Poglejmo si še kvantitavne rezultate učinkovitosti našega algoritma. Zato najprej definirajmo metriko za določanje pravilnosti razpoznavanja. Uporabili smo enake kriterije kot v [Pot98]. O tem, ali je regija pravilno razpoznala dejanski mešiček, se odločamo na osnovi izračunanega preseka med regijo in mešičkom. V [Pot98] so vpeljali dva količnika, in sicer r_1 in r_2. Količnik r_1 je razmerje med ploščinama preseka med razpoznano regijo in dejanskim mešičkom in dejanskega mešička, količnik r_2 pa je razmerje med ploščinama preseka in razpoznane regije. Za regijo rečemo, da je razpoznala dejanski mešiček, če sta tako r_1 kot tudi r_2 večja od 0,5.

Odstotek razpoznavanih mešičkov z našim algoritmom je po tem kriteriju okrog 90 %. To mero smo dobili tako, da smo izračunali povprečje odstotka razpoznanih mešičkov skozi vse testne slike. Standardni odklon je 19 %. Defirajmo še odstotek napačno razpoznanih. Ta odstotek definiramo kot razmerje med napačno razpoznanimi regijami in vsemi razpoznanimi regijami. Povprečen odstotek napačno razpoznanih je okrog 45 %, standardni odklon pa okrog 27 %.

Poglejmo, kako se spreminja učinkovitost po posameznih korakih algoritma. Po koraku določanja homogenih regij je približno 95 % mešičkov identificiranih z vsaj eno homogeno regijo. Po drugi strani pa te pravilne regije predstavljajo le 7 % med vsemi identificiranimi regijami. Po rasti regij je odstotek razpoznanih še vedno okrog 95 %, medtem ko se je delež pravilnih regij povečal na 8 %. Po končanju vseh korakov algoritma je odstotek razpoznavanja okrog 90 %, pri čemer pravilno razpoznane regije predstavljajo okrog 55 % od vseh razpoznanih regij.

Poglejmo še, kakšna je kvaliteta razpoznanih mešičkov. Grobo oceno o kvaliteti pravilno razpoznanih mešičkov dobimo s količnikoma r_1 ter r_2. Količnik r_1 je v povprečju 0,78 (standardni odklon je 0,07), količnik r_2 pa je v povprečju prav tako 0,78 (0,07).

Zadnji kriterij, na osnovi katerega ocenjujemo kvaliteto rezultatov, je razdalja med mejama razpoznane regije in dejanskega mešička. V literaturi [CK97, MKT98] obstaja cela vrsta različnih postopkov oz. razdalj, ki so namenjene za merjenje razdalje med dvema krivuljama (npr. Hausdorffova razdalja, radialna rms razdalja med krivuljama). Izbrali smo modificirano Hausdorffovo razdaljo oz. povprečno absolutno razdaljo med dvema krivuljama [MKT98]. Razdalja med krivuljama A in B je definirana z enačbo (3.11):

$$d(A,B) = \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n} d(\mathbf{a}_i, B) + \frac{1}{m}\sum_{i=1}^{m} d(\mathbf{b}_i, A)\right), \tag{3.11}$$

kjer je $A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ in $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots \mathbf{b}_m\}$. Točke $\mathbf{a}_i$ in $\mathbf{b}_i$ predstavljajo koordinate točk, ki definirajo krivuljo. Z $d(\mathbf{a}_i, B)$ je označena razdalja od točke $\mathbf{a}_i$ do najbližje točke na krivulji B in je definirana kot

$$d(\mathbf{a}_i, B) = \min_j \|\mathbf{b}_j - \mathbf{a}_i\|.$$

Povprečna napaka oz. absolutna razdalja med mejo razpoznane regije in dejanskega mešička je 4,10 piksla (standardni odklon je 1,31). Če upoštevamo, da v ultrazvočni sliki približno 50 do 60 pikslov tvori 1 cm, potem ta napaka znaša približno 0,8 mm – okrog 8 % glede na povprečno razdaljo od središča do meje mešička.

Realni ultrazvočni posnetki jajčnikov z jajčnimi mešički

Poglejmo si še rezultate segmentiranja pri realnih ultrazvočnih posnetkih.

Realne ultrazvočne posnetke jajčnikov smo dobili na Oddelku za človeško reprodukcijo in endokrinologijo Splošne bolnišnice 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 MIRO VIDEO DC-30 PLUS digitalizirali ter tako dobili posamezne slike v grafičnem formatu PNG (*portable network graphics*). Hitrost vzorčenja smo nastavili na najvišjo možno vrednost, kjer med digitaliziranjem še ni bilo izgubljenih slik. Hitrost vzorčenja je tako bila 15 slik na sekundo. Originalne ultrazvočne slike so velikosti 640 x 480 pikslov v 256 sivinah. Ultrazvočne slike smo dobili od 12 pacientk, ki so bile različnih starosti in brez znanih ginekoloških problemov. Za testiranje smo tako imeli na voljo 5800 ultrazvočnik slik. Za testiranje algoritma smo iz te množice slik izbrali popolnoma naključno 50 slik. Te testne slike je pregledal in ocenil ekspert – ginekolog. S tem smo dobili tudi referenčne položaje za mešičke.

Poglejmo si vmesne rezultate našega algoritma na realni ultrazvočni sliki. Na sliki 3.13 so prikazane homogene regije, ki jih dobimo z našo proceduro za določanje homogenih regij. Te regije so položene na originalno ultrazvočno sliko jajčnika. Zatem uporabimo rast regij na teh slikah. Rezultirajoče zrastle regije so prikazane v sliki 3.14. Na sliki 3.15 pa so po uporabi procedure za izločanje mešičkov nad zrastlimi regijami prikazani razpoznani mešički. Mešički, ki so orisani z belo, so položeni na originalno sliko. Dobljene rezultate lahko primerjamo s sliko 3.16, kjer je mešičke ročno označil ekspert – doktor.

Slika 3.13: Homogene regije (v belem), položene na originalno ultrazvočno sliko.

Slika 3.14: Regije po koraku rasti.

Na sliki 3.17 je prikazanih še nekaj tipičnih rezultatov. Leva slika iz posameznega para slik prikazuje razpoznane mešičke z našim algoritmom, na desni sliki pa je prikazan referenčni položaj mešičkov, kot jih je določil ekspert.

Podajmo še kvantitativne mere za učinkovitost segmentiranja na realnih slikah. Uporabili smo enake mere kot pri sintetičnih slikah. Odstotek razpoznavanih mešičkov z našim algoritmom je okrog 78 %. Standardni odklon je 21 %. Povprečen odstotek napačno razpoznanih je okrog 29 %, standardni odklon pa okrog 25 %.

Slika 3.15: Razpoznani mešički, položeni na originalno ultrazvočno sliko.

Slika 3.16: Mešički, kot jih je ročno označil ekspert.

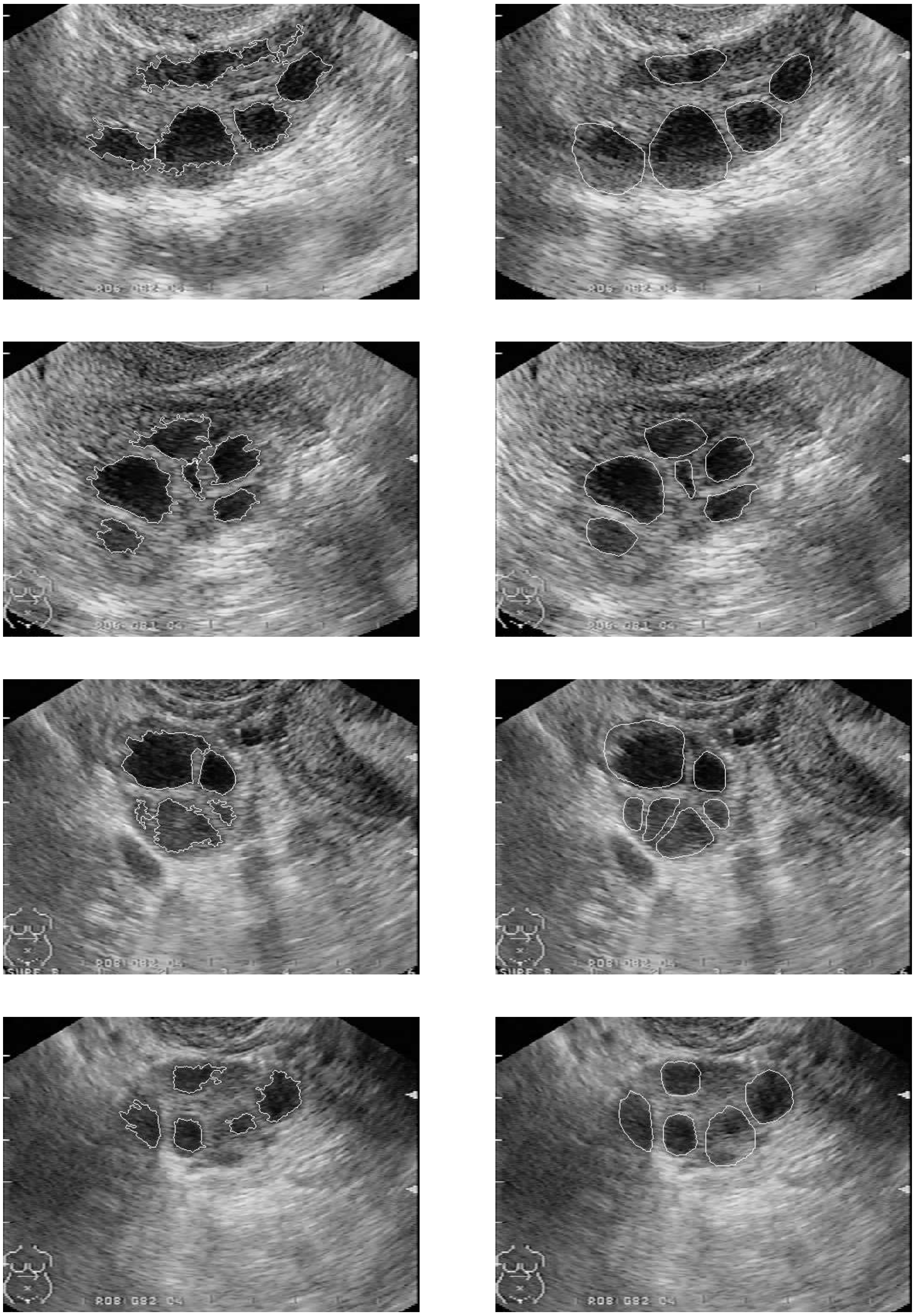
Poglejmo še, kako se spreminja učinkovitost po posameznih korakih algoritma. Po koraku določanja homogenih regij je približno 87 % mešičkov identificiranih z vsaj eno homogeno regijo. Po drugi strani pa te pravilne regije predstavljajo le 18 % med vsemi identificiranimi regijami. Po rasti regije je odstotek razpoznanih še vedno okrog 87 %, medtem ko se je delež pravilnih regij povečal na 28 %. Po končanju vseh korakov algoritma je odstotek razpoznavanja okrog 78 %, pri čemer pravilno razpoznane regije predstavljajo okrog 71 % od vseh razpoznanih regij.

Poglejmo še, kakšna je kvaliteta razpoznanih mešičkov. Količnik r_1 je v povprečju 0,75 (standardni odklon je 0,09), količnik r_2 pa je v povprečju 0,85 (z odklonom 0,1). Povprečna absolutna razdalja med mejo razpoznanega in pravilnega mešička je 5,30 piksla (standardni odklon je 1,80). Če upoštevamo, da v ultrazvočni sliki približno 50 do 60 pikslov tvori 1 cm, potem ta napaka znaša približno 1,1 mm – okrog 10 % glede na povprečno razdaljo od središča do meje mešička.

Razprava

Predstavili smo nov algoritem za detektiranje jajčnih mešičkov v ultrazvočnih slikah jajčnikov. Ta popolnoma avtomatski algoritem za detektiranje mešičkov opravi segmentacijo z rastjo regij. Rast začetnih homogenih regij kontroliramo z dvema preprostima praviloma zlitja – povprečno sivino in obteženim gradientom. Na koncu mešičke le še izločimo s preprostimi pravili, ki so vključena v korak poobdelave. Dobljeni rezultati so zelo obetajoči.

Na umetnih slikah je bil odstotek razpoznanih mešičkov okrog 90 %, na realnih ultrazvočnih pa je bil ta odstotek nekoliko nižji, in sicer okrog 78 %. Odstotek napačno razpoznanih regij pa se je gibal okrog 45 % pri fantomih ter okrog 29 % pri realnih ultra-



Slika 3.17: Nekaj tipičnih rezultatov. Leva slika vsakega para prikazuje mešičke, kot jih je razpoznal naš algoritem, na desni sliki pa je ekspert označil pravilen položaj mešičkov.

zvočnih posnetkih. Odstopanja v rezultatih med obema vrstama slik so gotovo posledica nepopolnega modeliranja jajčnika v umetnih slikah. V jajčniku smo namreč modelirali le mešičke, v realnih slikah pa se pojavljajo tudi druge strukture, prav tako temne, ki motijo delovanje algoritma. Ker so bili v umetnih slikah prisotni le mešički in šum, ob tem pa so bili mešički veliko temnejši kot okolica, ni imela procedura za določanje homogenih začetnih regij nikakršnih problemov pri identificiranju mešičkov. Začetne homogene regije so bile tudi že zelo dober približek mešičkov, tako da procedura rasti regij ni dodala večjih popravkov. Še največji problem predstavljajo ozka in črna področja, ki se dotikajo mešičkov (nekakšni pasovi). Stičišča med takšno regijo in mešičkom ni možno zaznati le na osnovi opazovanja sivine in gradienta. Zato smo v proceduro rasti regije vgradili kriterij kompaktnosti (šteli smo sosede – glej podpoglavje 3.3.3). Ta kriterij je prekinil rast pri ozkih pasovih, medtem ko širših regij nismo mogli zaznati. Zato se je mešiček na posameznih področjih preveč razrastel. V umetnih slikah je izredno velik odstotek napačno razpoznanih, kar 45 %. Te napačne regije so sestavljene iz nekaj temnejših, skupaj ležečih peg. Do razlike med odstotkoma napačno razpoznanih v umetnih in realnih slikah prihaja iz dejstva, da je v umetni sliki približno 90 % območja, kjer se pojavljajo pege, v realnih slikah pa je ta odstotek veliko nižji, približno 50 % (drugi del predstavljajo mešički, žile, maternična sluznica). Pričakujemo pa lahko, da bi se na slikah, dobljenih po pregledovanju modela jajčnika s samimi velikimi mešički, odstotek napačno razpoznanih bistveno zmanjšal.

Odstotek razpoznavanja je na realnih posnetkih nekoliko nižji kot na umetnih. Treba je poudariti, da so bili pri računanju uspešnosti razpoznavanja upoštevani vsi mešički, tako majhni (premera 2 mm) kakor veliki (premera 10 mm in več). Algoritem ima največje težave pri razpoznavanju zelo majhnih mešičkov, ki so ponavadi še svetlejši, težko so vidni tudi s prostim očesom. Pri teh mešičkih zataji že procedura za določanje začetnih homogenih regij. Drug problem algoritma pa predstavlja več skupaj ležečih mešičkov na zelo temnem območju (glej sliko 3.17), pri čemer ni izrazite meje med njimi. V takšnem primeru že identifikacija homogenih regij na tem področju določi le eno samo regijo. Takšnega problema ne moremo rešiti brez vgrajevanja dodatnega znanja o obliki iskanih objektov v sam segmentacijski algoritem. Tega smo se izogibali, saj želimo čim bolj splošen algoritem. Možna rešitev problema bi bilo dodajanje koraka poobdelave, kjer bi se na osnovni meje razpoznanega objekta odločali, ali gre za staknjene objekte ali ne. Eksperti ugotavljajo, da regija z mejo v obliki osmice skoraj zagotovo vsebuje dva mešička.

Korak izločanja mešičkov smo zasnovali na osnovi znanja o problemu (glej podpoglavje 3.3.4). Zadnji kriterij, ki smo ga pri tem uporabili, je bila sosednost mešičkov – vsi mešički ležijo skupaj. Pri tem smo predvideli, da je regija, ki je ena izmed največjih ob tem pa še ena izmed najkompaktnejših, zagotovo mešiček. Okoli tega "mešička" pa zatem nadaljujemo iterativno dodajanje preostalih mešičkov. Takšno določanje prvega pravega mešička je seveda hevristično in ne daje zmeraj pravilnih rezultatov. Če bi npr. imeli v sliki zelo veliko, temno in ovalno žilo, ki se ne bi dotikala roba, bi jo po tem kriteriju napačno izbrali za prvi pravi mešiček in bi jajčnik zatem gradili na napačnem mestu. Seveda takšni primeri niso pogosti, vsekakor so pa možni. Če bi v tem koraku dovolili uporabnikovo interakcijo, recimo, da bi z miško pokazal na največji mešiček v slik, potem bi se seveda občutno izboljšala učinkovitost algoritma, bi pa ob tem izgubili popolno avtomatičnost postopka.

Poglejmo še kvaliteto dobljenih mešičkov. Količnik r_1 je bil tako pri umetnih kakor tudi pri realnih slikah okoli 0,75, medtem ko je bil količnik r_2 pri umetnih slikah okoli 0,78, pri realnih pa okoli 0,85. To z drugimi besedami pomeni, da so naši razpoznani mešički v povprečju pokrivali 75 % območja pravega mešička, ob tem pa je bilo okrog 22 % območja razpoznega mešička zunaj pravega območja pri umetnih in 15 % pri realnih slikah. Razdalja med mejo pravega in razpoznanega mešička se je pri obeh vrstah slik gibala okoli 5 pikslov. Iz teh rezultatov lahko ugotovimo naslednje. Očitno je, da se je na posameznih delih mešička rast regije končala prehitro. Če pogledamo strukturo mešička pobliže, opazimo, da je v grobem sestavljen iz treh regij, ki izkazujejo različne lastnosti. V sredini mešička je najhomogenejši, tj. najtemnejši del. Zatem sledi druga regija, pas okoli prve regije, ki je nekoliko svetlejša in manj homogena od prve. Zadnji pas pa je najsvetlejši, najbolj nehomogen in z največjo echogeničnostjo. Ta pas se razlikuje od mešička do mešička in sega do samega notranjega roba mešička. Pri posameznih mešičkih ima ta pas tako različno statistiko sivin in obteženega gradienta od statistik regije, ki jo razširjamo, da po obeh kriterijih rast končamo, preden pridemo do prave meje mešička (enačbi (3.6) in (3.8)). Razlog, da je tudi količnik r_2 manjši od 1, pa so tanka črna področja (pasovi), ki se dotikajo mešička. Po teh malo širših pasovih nam regija preraste pravo mejo mešička. Teh pasov je v umetnih slikah veliko več kakor pri realnih posnetkih, zato je v umetnih slikah tudi r_2 manjši.

Moramo še poudariti, da rast regij izvajamo na originalnih, nefiltriranih ultrazvočnih slikah. S tem minimiziramo možnost, da se sosednji mešički zlijejo. Žal pa ima uporaba nefiltrirane slike pri obdelovanju tudi stranski učinek, in sicer imajo razpoznani mešički zelo nazobčane robove.

Bistven faktor pri ocenjevanju učinkovitosti algoritma na realnih posnetkih je ekspert-opazovalec [CK97, Pot98], saj predstavljajo njegovi odčitki edino pravo referenco. Kot smo že uvodoma omenili, je vse testne ultrazvočne slike ocenil le en ekspert. Ker je slike ocenjeval le en opazovalec, nismo imenili nikakršne možnosti, da ocenimo vpliv opazovalčeve variabilnosti. Ekspert je podal svoje mnenje le na osnovi ene same 2D slike iz zaporedja, kar pomeni, da ni videl slik, ki so v zaporedju pred ali po njej. Ocenil je sliko pod enakimi pogoji, kot jih ima algoritem za razpoznavanje na statičnih 2D slikah. Prav zaradi tega lahko upravičeno pričakujemo, da je tudi ekspert podal za posamezne mešičke napačno oceno (v splošnem torej, ali je označena regija res mešiček, in tudi oceno o samem obrisu mešička). V fazi testiranja algoritma smo za vsako testno sliko imeli podani dve neodvisni mnenji dveh ocenjevalcev, res pa je, da ocenjevalca nista bila ginekologa in s tem merodajna. Kljub temu, smo izračunali opazovalčevo variabilnost, ki je znašala kar 8 % glede na dobljene rezultate. Tukaj mislimo predvsem na količnika r_1 in r_2 ter na razdaljo razpoznanega do roba dejanskega mešička, pri številu mešičkov sta namreč oba opazovalca dosegla soglasje.

Naš algoritem uporablja mnogo pragov. Te pragove pa je možno zelo preprosto določiti iz osvojenega znanja o mešičkih in ultrazvočnih slikah. Vrednosti pragov določimo enkrat samkrat in so enake za vse tipe ultrazvočnih slik jajčnikov. Uporabljeni pragovi v naši raziskavi so bili enaki tako za umetne kakor tudi za realne ultrazvočne slike

Če na koncu primerjamo še učinkovitost naše metode z drugimi algoritmi za detektiranje mešičkov, vidimo, da očitno prekaša algoritme, opisane v [PZK97, PVZ98, VDZ98], in ima podobno učinkovitost kot algoritem v [KS98], vendar slednji izkazuje višjo kvaliteto razpoznanih mešičkov. Treba pa je poudariti, da je algoritem v [KS98] polavtomatski, medtem ko je naš algoritem popolnoma avtomatski. Ob tem velja še pripomniti, da vsi ti algoritmi niso bili testirani na isti testni množici in je zato primerjava lahko nezanesljiva.

3.5 Zaključek

V tem poglavju smo razvili postopek za segmentiranje statičnih 2D slik. Razviti algoritem, podrobno je opisan v podpoglavju 3.3, je namenjen za izločanje temnih ovalnih območij s slik. Preverili smo ga na umetnih in realnih ultrazvočnih slikah jajčnikov.

Pri testiranju algoritma smo uporabili tudi umetne ultrazvočne slike jajčnika z jajčnimi mešički. Simulator tvorjenja umetnih posnetkov smo podrobno opisali v podpoglavju 3.2. Razlog za uporabo umetnih slik je v tem, da lahko le na umetnih slikah dobimo eksaktno oceno o učinkovitosti algoritma. Realne slike lahko namreč ocenjujemo le na osnovi ekspertovih ocen, ki pa zaradi velikih opazovalčevih variabilnosti ne morejo biti zlati standard.

Algoritme za segmentiranje statičnih 2D slik, seveda tudi naš algoritem, lahko uporabimo na zaporedju 2D slik. Celotno zaporedje slik razbijemo na posamezne slike in jih potem eno za drugo s takšnim algoritmom obdelamo. Seveda pa ob takšni obdelavi zaporedja slik nikjer ne prenašamo informacij o dobljenih rezultatih na prejšnjih slikah. Zato lahko upravičeno pričakujemo, da bodo rezultati obdelovanja zaporedja slik na takšen način slabši, kakor pa bi bili, če bi si ob tem pomagali še z rezultati, dobljenimi na nekaj prejšnjih slikah.

4.

Algoritem za avtomatsko razpoznavanje objektov iz zaporedja slik s postopki predikcije

V prejšnjem poglavju smo predstavili algoritem za razpoznavanje temnih ovalnih objektov na posamezni statični 2D digitalni sliki. Razviti algoritem uporablja pri segmentiranju metodo rasti regij in množico statističnih testov, ki ga ohranjajo splošno namenskega. Preverili smo ga na množici umetnih in realnih ultrazvočnih posnetkov jajčnika z mešički. Na osnovi dobljenih rezultatov in po primerjavi z drugimi podobnimi algoritmi smo videli, da je opisani algoritem eden izmed najboljših algoritmov za razpoznavanje mešičkov v ultrazvočnih slikah jajčnikov.

Ta algoritem lahko uporabimo tudi za segmentiranje zaporedja 2D slik, in sicer tako, da obdelamo vsako sliko iz zaporedja posebej. Kljub temu, da se algoritem bliža optimalnosti pri razpoznavanju na eni 2D sliki, opazimo pri takšnem obdelovanju zaporedja slik določene pomanjkljivosti. Po obdelovanju zaporedja umetnih ultrazvočnih slik jajčnikov smo izmerili relativno visok odstotek razpoznavanja mešičkov, vendar pa je bil ob tem tudi odstotek napačno razpoznanih izredno velik (podpoglavje 3.4). Odstotek napačno razpoznanih je velik, navkljub vsemu znanju o ciljnih objektih (mešičkih), ki smo ga vgradili v algoritem. Po razpoznavanju s tem algoritmom dobimo v sliki množico regij, za katere trdimo, da so zaradi svojih lastnosti (sivina, velikost, oblika), najverjetneje mešički. Na osnovi le ene statične slike pa je nemogoče sprejeti še kakršnokoli drugo odločitev. Podoben dvom se je pojavil tudi v ekspertove odčitke, ki jih je določil le na osnovi ene 2D slike (podpoglavje 3.4).

Veliko lažje se odločamo po analizi rezultatov segmentiranja na množici sosednjih slik. Če se posamezna regija pojavlja na več zaporednih slikah, smo lahko bolj gotovi, da ne gre za šumno regijo. Podobno, če se oblika in velikost razpoznane regije zelo malo spreminjata od slike do slike, potem se lahko zanesemo na razpoznavanje. Na ti in še na mnoge druge nejasnosti lahko dobimo odgovor iz zaporedja.

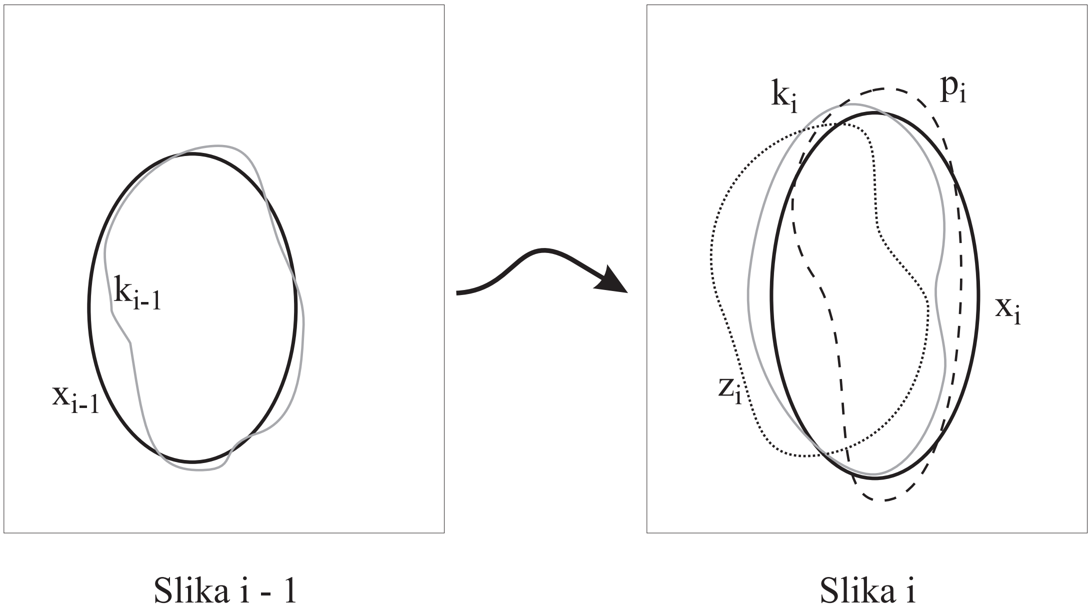
V tem poglavju bomo študirali, kako čim optimalneje uporabiti zgodovino razpoznavanja na predhodnih slikah za razpoznavanje objektov v trenutni sliki. Razvili bomo nov algoritem, ki bo razpoznaval objekte iz zaporedja slik s postopki predikcije. Algoritem bomo podrobno opisali, pri tem bomo poseben poudarek namenili razlagi vpeljanih korakov. Probleme, na katere lahko naletimo, bomo detajlno opisali na mestu, kjer se prvič pojavijo, in ob tem skicirali možne rešitve. Delovanje algoritma bomo predstavili na umetnih in realnih ultrazvočnih posnetkih jajčnika z mešički. Ne glede na to pa je nov algoritem neodvisen od domene testiranja in je s tem splošen.

4.1 Definicija problema

V tem podpoglavju bomo eksaktno definirali problem, ki ga rešujemo. Po obdelovanju zaporedja slik z optimalnim algoritmom za 2D slike za dano domeno dobimo za vsako sliko množico razpoznanih regij. Te razpoznane regije imenujmo meritve. To pomeni, da algoritem vrača meritev (izmero) posameznega ciljnega objekta. Že uvodoma smo nakazali, da pa te meritve, četudi imamo opravka z optimalnim algoritmom 2D, niso eksaktne. Razlogov za to je več: a) tip slik (šum), b) pogoji pri zajemanju zaporedja slik (osvetljevanje scene, tresenje roke pri ultrazvočnem pregledu, neustrezno vzorčenje ipd.), c) algoritmi za razpoznavanje na 2D slikah niso v realnih razmerah nikoli 100-odstotno učinkoviti oz. natančni. Zaradi teh faktorjev variabilnosti se ne smemo nikoli popolnoma zanesti na dobljene meritve.

Iz tega sledi, da moramo meritev z_i, ki jo dobimo v i-ti sliki, na osnovi določenih informacij popraviti. Te informacije pa so lahko le informacije o razpoznavanju na predhodnih slikah iz zaporedja. To pomeni, da lahko na osnovi razpoznavanja v preteklih slikah iz zaporedja določimo predviden položaj objekta v trenutni sliki. Imenujmo predviden položaj objekta predikcija in ga označimo s p_i. Torej imamo v i-ti sliki opravka z dvema položajema objekta, in sicer predvidenim položajem p_i ter izmerjenim položajem z_i. Končni položaj objekta k_i v i-ti sliki pa dobimo, če na optimalen način upoštevamo p_i in z_i. Torej je k_i neka funkcija $\mathcal{W}$ spremenljivk p_i in z_i, $k_i = \mathcal{W}(p_i, z_i)$.

Na sliki 4.1 je shematsko prikazan problem, s katerim imamo opravka. Na $(i-1)$-ti sliki je označen pravilni položaj objekta z x_{i-1}. Objekt je popolnoma popisan z njegovo notranjo mejo [SHB94], kar tudi upoštevamo v nadaljevanju. Torej je x_{i-1} krivulja, ki določa mejo objekta. Krivulja k_{i-1} pa predstavlja najboljši približek objekta v $(i-1)$-ti sliki, ki jo dobimo na osnovi razpoznavanj na predhodnih slikah. Iz tega približka lahko določimo predikcijo p_i za i-to sliko. Predikcija p_i je torej rezultat neke funkcije $\mathcal{P}$ nad k_{i-1}, $p_i = \mathcal{P}(k_{i-1})$. V i-ti sliki dobimo tudi meritev z_i, ki jo vrne algoritem za razpoznavanje



Slika 4.1: Osnovna shema razpoznavanja s postopki predikcije. Z x je označen pravilni položaj objekta v dveh zaporednih slikah (črna neprekinjena črta). Najboljši približek objekta v sliki označimo s k (siva neprekinjena črta), predikcijo s p (črtkano) in meritev z z (pikčasto).

na statičnih 2D slikah. Končni približek objekta, k_i, pa dobimo kot rezultat funkcije $\mathcal{W}$ nad meritvijo in predikcijo. Z x_i je označen pravilni položaj objekta.

Naj še enkrat poudarimo, da so x, z, p in k krivulje. Vsaka krivulja pa je sestavljena iz n točk, torej so dimenzije n x 2. Na sliki 4.1 je dobro vidno, da se krivulje p, z in k razlikujejo od pravega položaja objekta x. To pomeni, da je objekt razpoznan z določeno napako e. Tako lahko zapišemo:

$$x_i = z_i + e(z_i, x_i), \tag{4.1}$$
$$x_i = p_i + e(p_i, x_i), \tag{4.2}$$
$$x_i = k_i + e(k_i, x_i), \tag{4.3}$$

kar pomeni, da dobimo pravilen položaj objekta x_i v i-ti sliki, če prištejemo meritvi z_i napako $e(z_i, x_i)$, izvirajočo iz z_i in x_i. Podobno velja za predikcijo p_i in najboljši približek k_i. Na napako e lahko gledamo tudi kot na razdaljo med krivuljama. Če izenačimo desni strani enačb (4.1) in (4.2), dobimo:

$$z_i - p_i = e(p_i, x_i) - e(z_i, x_i), \tag{4.4}$$

oz. če vpeljemo oznako $e(p_i, z_i) = z_i - p_i$, dobimo

$$e(p_i, z_i) = e(p_i, x_i) - e(z_i, x_i). \tag{4.5}$$

Podobno dobimo po izenačitvi enačb (4.1) in (4.3):

$$e(k_i, z_i) = e(k_i, x_i) - e(z_i, x_i). \tag{4.6}$$

Enačbi (4.5) in (4.6) lahko preoblikujemo v naslednjo končno obliko:

$$\begin{aligned} e(p_i, x_i) &= e(p_i, z_i) + e(z_i, x_i), \\ e(k_i, x_i) &= e(k_i, z_i) + e(z_i, x_i). \end{aligned} \tag{4.7}$$

Iz enačbe (4.7) vidimo naslednje. Če bi lahko določili napako meritve do pravega položaja, $e(z_i, x_i)$, potem bi s tem imeli tudi določeni napaki predikcije do pravega položaja, $e(p_i, x_i)$, in najboljšega približka do pravega položaja, $e(k_i, x_i)$. V realnih razmerah ne poznamo točnega položaja objekta x_i, na voljo imamo zgolj z_i, p_i in k_i. Zato lahko operiramo zgolj z ocenami za napake glede na pravi položaj objekta.

Razmišljanje najlažje nadaljujemo, če se osredotočimo le na eno točko s krivulje in na eno dimenzijo. Posplošitev je trivialna. Predpostavimo, da je napaka $e(z_i, x_i)$ naključna spremenljivka, porazdeljena po Gaussovi porazdelitvi: $e(z_i, x_i) \sim N(a_1, \sigma_1)$. Naj bo tudi napaka $e(p_i, z_i)$ naključna spremenljivka, porazdeljena po normalni porazdelitvi: $e(p_i, z_i) \sim N(a_2, \sigma_2)$. Potem je tudi napaka $e(p_i, x_i)$, ki je vsota dveh naključnih spremenljivk, porazdeljenih po normalni porazdelitvi, porazdeljena po normalni porazdelitvi [BS78], in sicer $e(p_i, x_i) \sim N(a_1 + a_2, \sqrt{\sigma_1^2 + \sigma_2^2})$. Označimo z Z napako $e(x_i, z_i)$, $Z \sim N(-a_1, \sigma_1)$, ter s P napako $e(x_i, p_i)$, $Z \sim N(-a_1 - a_2, \sqrt{\sigma_1^2 + \sigma_2^2})$.

Ob teh predpostavkah nas zanima, kako določiti najboljši približek k_i v trenutni sliki, da bo napaka $K = e(x_i, k_i)$ najmanjša. K je neka funkcija $\mathcal{W}$ parametrov P in Z, $K = \mathcal{W}(P, Z)$. Omejimo funkcijo $\mathcal{W}$ na množico linearnih funkcij; potem lahko zapišemo:

$$K = l \cdot Z + (1 - l) \cdot P + s, \qquad l \in [0, 1], \tag{4.8}$$

kjer je parameter l utež, s pa je konstanta. Tudi K je porazdeljena po normalni porazdelitvi

$$K \sim N\Big(-a_1 - (1 - l)a_2 + s, \sqrt{l^2\sigma_1^2 + (1 - l)^2(\sigma_1^2 + \sigma_2^2)}\Big).$$

Napaka K bo najmanjša, ko bo izraz

$$-a_1 - (1 - l)a_2 + s \quad \rightarrow \quad 0 \tag{4.9}$$

in izraz

$$f = \sqrt{l^2\sigma_1^2 + (1 - l)^2(\sigma_1^2 + \sigma_2^2)} \quad \rightarrow \quad \text{min}. \tag{4.10}$$

Iz pogoja v (4.9) dobimo, da

$$s = a_1 + (1 - l)a_2. \tag{4.11}$$

Drugi pogoj v enačbi (4.10) pa upoštevamo tako, da najprej določimo $\frac{\partial f}{\partial l}$, izračunani odvod enačimo z 0 in dobimo

$$l = \frac{\sigma_1^2 + \sigma_2^2}{2\sigma_1^2 + \sigma_2^2}. \tag{4.12}$$

S tako izračunanimi utežmi določimo najboljši približek objekta kot linearno kombinacijo predikcije in meritve. Iz izračunanih uteži s in l lahko tudi vidimo, da so te vrednosti popolnoma enake, kot če bi jih določili s Kalmanovim filtrom, kjer bi za prvo meritev vzeli P, za drugo meritev pa Z (glej 1. poglavje v [May94]).

Sklenimo uvodno razmišljanje z naslednjo ugotovitvijo. Če so meritve normalno porazdeljene okoli pravega položaja objekta in če so predikcije normalno porazdeljene okoli meritev, potem smo pokazali, da lahko določimo najboljši približek objekta v trenutni sliki kot linearno kombinacijo predikcije in meritve. Pokazali smo, da se uteži za to linearno kombinacijo določijo s Kalmanovim filtrom. To pomeni, da lahko ob teh predpostavkah pri razpoznavanju objektov v zaporedju slik uporabimo Kalmanov filter.

4.2 Algoritem za razpoznavanje s postopki predikcije

V uvodnem podpoglavju smo podrobno definirali problem, ki ga rešujemo. Pokazali smo tudi, da je pri razpoznavanju s postopki predikcije optimalno, če vgradimo rezultate predhodnih razpoznavanj v Kalmanov filter.

V tem podpoglavju pa bomo detajlno opisali novi algoritem za avtomatsko razpoznavanje objektov iz zaporedja slik s postopki predikcije.

4.2.1 Osnovna oblika algoritma

Preden začnemo z opisom novega algoritma, moramo razložiti nekaj pojmov, ki jih bomo uporabljali. Algoritem za razpoznavanje objektov na eni statični 2D sliki razpozna v posamezni sliki več regij – na eni sliki imamo torej več meritev. Naš novi algoritem pa bo obdeloval le eno regijo naenkrat. Torej, ko bomo uporabljali izraz meritev, bomo pri tem mislili le na eno regijo. Vse regije na sliki pa obdelamo tako, da za vsako regijo poženemo naš algoritem.

Treba je še opredeliti, kako se odločamo, katera meritev v trenutni sliki ustreza meritvi na prejšnji sliki. Odločitev sprejemamo na osnovi preseka. Meritvi j v $(i-1)$-ti sliki, z_{i-1}^j, kar je seveda isto kot regija, ustreza meritev k_{max} iz i-te slike, $z_i^{k_{max}}$, ki ima z njo največji, od nič različni presek. Ta presek je največji presek, ki ga ima meritev z_{i-1}^j s katerokoli drugo meritvijo iz i-te slike, hkrati pa je ta presek tudi največji presek, ki ga ima meritev

$z_i^{k_{max}}$ z meritvami iz $(i-1)$-te slike. To lahko formalno zapišemo kot:

$$k_{max} = \max_k \|z_{i-1}^j \cap z_i^k\| \quad \wedge \quad j = \max_l \|z_{i-1}^l \cap z_i^{k_{max}}\|. \tag{4.13}$$

Če na i-ti sliki ne obstaja meritev, ki bi ustrezala pogoju ujemanja (4.13), potem rečemo, da meritev z_{i-1}^j nima ujemanja, hkrati pa to tudi za nas pomeni, da se je objekt, ki ga določa meritev, zadnjič pojavil na $(i-1)$-ti sliki iz zaporedja. Če se ta objekt čez nekaj slik ponovno pojavi na istem mestu v sliki, obravnavamo to regijo – meritev kot neko novo regijo. Iz enačbe (4.13) tudi vidimo, da lahko eni meritvi v prejšnji sliki ustreza le ena meritev na trenutni sliki (in obratno). S tako zasnovanim ujemanjem za vsak objekt točno vemo, kdaj se prvič pojavi v zaporedju in kdaj ponikne.

Algoritem je v osnovi neodvisen od tipa slik, saj je vhod v algoritem zaporedje že segmentiranih slik. Zaradi lažje razlage pa bomo pri razvijanju algoritma posamezne korake demonstrirali na razpoznavanju mešičkov v zaporedju ultrazvočnih slik jajčnikov.

Shema algoritma

Potem ko smo opredelili osnovne pojme, lahko začnemo z opisom algoritma. Uporabimo idejo osnovne sheme sledenja objektom s Kalmanovim filtrom [TV98]. V tej osnovni shemi popišemo objekt s težiščem. V stanju sistema hranimo koordinati x in y težišča ter premik po x in y. Na novi sliki pa lahko izmerimo le nov položaj težišča.

To shemo smo uporabili tudi v našem algoritmu, vendar nekoliko modificirano. Ker nismo postavili nikakršnih omejitev glede oblike objektov, ki jih razpoznavamo – objekti so lahko celo elastični – je aproksimacija objekta zgolj z njegovim težiščem za nas neustrezna. Zato smo shemo sledenja namesto na težišču razvili na vsaki točki iz meje objekta. To pomeni, da sledimo posamezni točki iz meje skozi zaporedje. Stanje sistema $\mathbf{x}_{j,i}$ za točko j v i-ti sliki definiramo kot

$$\mathbf{x}_{j,i} = \begin{bmatrix} x_{j,i} \\ y_{j,i} \\ \triangle x_{j,i} \\ \triangle y_{j,i} \end{bmatrix}. \tag{4.14}$$

Enačba sistema je oblike

$$\mathbf{x}_{j,i} = \mathbf{\Phi} \cdot \mathbf{x}_{j,i-1} + \mathbf{w}_{i-1}, \tag{4.15}$$

kjer je matrika $\mathbf{\Phi}$ oblike

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.16}$$

šum v sistemu, $\mathbf{w}_{i-1}$, pa je porazdeljen po normalni porazdelitvi: $\mathbf{w}_{i-1} \sim N(0, \mathbf{Q}_{i-1})$. Matrika $\mathbf{\Phi}$ torej določa, da novi predvideni x in y-koordinati točke dobimo tako, da prištejemo k stari vrednosti koordinat ustrezen premik po x oz. y. Vrednosti premikov se zgolj preneseta iz prejšnjega koraka.

K meritvi, ki je na voljo, spadata le novi koordinati točke na trenutni sliki. Torej je meritev $\mathbf{z}_{j,i-1}$ oblike

$$\mathbf{z}_{j,i-1} = \begin{bmatrix} z_{x_{j,i-1}} \\ z_{y_{j,i-1}} \end{bmatrix}, \tag{4.17}$$

kjer $z_{x_{j,i-1}}$ in $z_{y_{j,i-1}}$ označujeta koordinati x oz. y izmerjene točke. Merilna enačba je oblike

$$\mathbf{z}_{j,i-1} = \mathbf{H} \cdot \mathbf{x}_{j,i-1} + \mathbf{v}_{i-1}. \tag{4.18}$$

Matrika $\mathbf{H}$ je definirana kot

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{4.19}$$

šum v meritvah, $\mathbf{v}_{i-1}$, pa je porazdeljen po normalni porazdelitvi: $\mathbf{v}_{i-1} \sim N(0, \mathbf{R}_{i-1})$. Vrednosti v stanju sistema in meritvi so podane v enakih enotah, zato ni potrebna nobena transformacija med stanji sistema in meritvijo – v matriki $\mathbf{H}$ so zato same enice.

Začetno stanje sistema

Podajmo algoritem. V svoji osnovni obliki razpoznava objekte iz zaporedja slik po naslednji shemi. Najprej vzamemo prvo sliko S_0 iz zaporedja $\mathbf{S}$, kjer se prvič pojavi iskani objekt O. Zaporedje slik $\mathbf{S}$ je zaporedje z že razpoznanimi objekti v slikah. Ti objekti so bili razpoznani z enim izmed algoritmov za razpoznavanje v 2D slikah. Objektu O zatem določimo notranjo mejo. To krivuljo vzamemo za najboljši približek objekta k_0 na sliki S_0. Krivulja k_0 je sestavljena iz množice točk oz. pikslov $\mathbf{k}_{j,0}$:

$$k_0 = \{\mathbf{k}_{j,0}, \quad j = 0, \ldots, n_0 - 1\}, \tag{4.20}$$

kjer je n_0 število točk na krivulji. Piksel $\mathbf{k}_{j,0}$ ima dve koordinati, in sicer koordinato x, ki jo v nadaljevanju označujemo kot $k_{x_{j,0}}$, ter koordinato y, ki jo označujemo s $k_{y_{j,0}}$.

Vsaki točki $\mathbf{k}_{j,0}$ v koraku 0 priredimo vektor stanja sistema $\hat{\mathbf{x}}_{j,0}$, in sicer

$$\hat{\mathbf{x}}_{j,0} = \begin{bmatrix} k_{x_{j,0}} \\ k_{y_{j,0}} \\ 0 \\ 0 \end{bmatrix}, \tag{4.21}$$

kjer določata $k_{x_{j,0}}$ in $k_{y_{j,0}}$ koordinati x in y piksla $\mathbf{k}_{j,0}$.

Meritev stanja sistema

S tem smo definirali začetno oceno stanja sistema, ki je potrebna za delovanje Kalmanovega filtra. Če poznamo matrike $\mathbf{P}$, $\mathbf{Q}$ in $\mathbf{R}$, lahko vstopimo v cikel enačb diskretnega Kalmanovega filtra ((2.28) do (2.32)). Torej, iz slike S_0 gremo v sliko S_1 – v korak 1. Na osnovi pogoja ujemanja v (4.13) določimo objekt O na sliki S_1. Objektu zatem določimo notranjo mejo. Od te slike oz. koraka naprej pa ta krivulja predstavlja meritev objekta. Meritev z_1 je prav tako sestavljena iz množice točk:

$$z_1 = \{\mathbf{z}_{j,1}, \quad j = 0, \ldots, m_1 - 1\}, \tag{4.22}$$

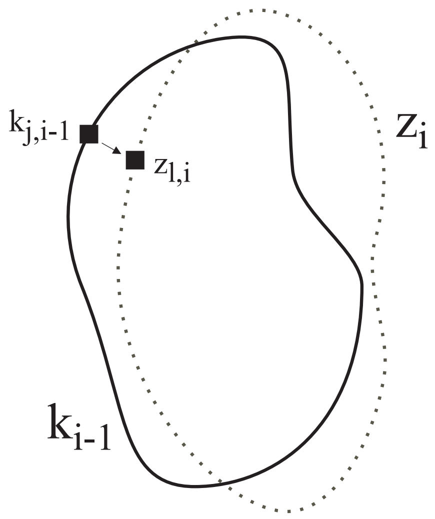
kjer je m_1 število točk na krivulji.

Iz vsega povedanega lahko opazimo, da začnemo shemo "prediktor-korektor" Kalmanovega filtra opravljati šele na drugi sliki iz zaporedja (S_1). Ker imamo v vsaki točki krivulje k_0 definiran enak sistem $\hat{\mathbf{x}}_{j,0}$, moramo torej za vsako točko $\mathbf{k}_{j,0}$, ki ustreza temu sistemu, določiti še pripadajočo meritev $\mathbf{z}_{l,1}$ iz krivulje z_1. Povedano bolj preprosto: določiti moramo najboljše ujemanje med krivuljo k_0 in z_1.

Določanje ujemanja med krivuljo k in z

Na sliki 4.2 je shematsko prikazan problem ujemanja krivulj. Za vsako točko $\mathbf{k}_{j,i-1}$ s krivulje k_{i-1} je treba poiskati pripadajočo točko $\mathbf{z}_{l,i}$ iz meritve z_i. Gre za enak problem kot pri računanju razdalje med dvema krivuljama, kjer je treba najprej določiti pripadajoče pare točk z obeh krivulj, zatem pa še izračunati razdaljo med njima [CK97, Pot98]. V [CK97] najdemo več možnosti za določanje ujemanja točk iz dveh krivulj. Nobena od teh metod ni optimalna niti popolnoma zanesljiva. Ujemanje med krivuljama lahko npr. določamo tako, da krivulji popišemo s polarnim histrogramom, ta histograma pa zatem poravnamo tako, da je razdalja med njima najmanjša. Pripadajoči točki imata isti indeks v polju. Ta metoda je dobra za lepe in konveksne oblike ter za krivulje, ki imajo isto število točk. V realnih razmerah to seveda ni izpolnjeno, predvsem pogoj o istem številu točk. V tem primeru je treba krivuljo z manjšim številom točk interpolirati, tako da dobimo isto število točk. S tem se pojavi problem, na katerih mestih interpolirati, ali enakomerno ali na posameznih delih bolj gosto itd. Ujemanje med krivuljama na dveh zaporednih slikah lahko določimo tudi na osnovi optičnega pretoka [WYP97]. Za vsako točko prve krivulje izračunamo optični pretok, pripadajočo točko v drugi krivulji pa poiščemo v smeri optičnega pretoka iz točke na prvi krivulji. Pri tej metodi potrebujemo ob segmentiranih slikah še slike sivin. Znano je tudi, da je ocenjevanje optičnega pretoka izredno občutljivo na šumne podatke (podpoglavje 2.2). Problem za to metodo predstavljajo tudi velike deformacije krivulje iz slike v sliko.

Obstaja še en preprost način določanja ujemanja krivulj, in sicer na osnovi opazo-



Slika 4.2: Problem ujemanja krivulj k_{i-1} in z_i.

vanja najmanjše razdalje točke do krivulje [CK97]. Krivulji najprej ustrezno premaknemo (translacija), da se njuni težišči ujameta. Zatem pa določimo ujemanje krivulj. Točka iz prve krivulje se ujema z najbližjo točko iz druge. To metodo ujemanja smo uporabili v našem algoritmu. Formalno zapišimo določanje ujemanja med krivuljama. Točka $\mathbf{k}_{j,i-1}$ s krivulje k_{i-1} se ujema s točko $\mathbf{z}_{l,i}$ s krivulje z_i, če je izpolnjeno:

$$\|\mathbf{k}_{j,i-1} - \mathbf{z}_{l,i} - \mathcal{T}\| < \|\mathbf{k}_{j,i-1} - \mathbf{z}_{p,i} - \mathcal{T}\|, \ \forall p \in [0, m_i - 1] \ \wedge \ l \neq p, \tag{4.23}$$

kjer je $\mathcal{T}$ vektor, ki določa velikost translacije med obema krivuljama.

Določitev matrik P, Q in R

Na koncu moramo definirati le še matrike $\mathbf{P}$, $\mathbf{Q}$ in $\mathbf{R}$, ki so potrebne za delovanje Kalmanovega filtra. V splošnem ne vemo ničesar o natančnosti meritev niti o natančnosti sistema. Zato ne moremo podati eksaktnih ocen za te matrike. Predpostavimo, da sta oba sistema, tako merilni kot sistem stanj, enako šumna ter naj zato Kalmanov filter enakomerno upošteva obe oceni (meritev in predikcijo). Tako smo določili matriki šuma v sistemu, $\mathbf{Q}$, in v merilnem sistemu, $\mathbf{R}$, kot:

$$\mathbf{Q}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{in} \qquad \mathbf{R}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.24}$$

V prvih nekaj začetnih korakih (slikah) obdelovanja s KF lahko pričakujemo bolj slabe ocene modelnih parametrov [Hos95], zato je smiselno na začetku dati večji vpliv meritvam kakor pa predikciji. To idejo upoštevamo. Zato postavimo elemente v kovariančni matriki napake $\mathbf{P}_0^-$ na visoke vrednosti. Matriko $\mathbf{P}_0^-$ definiramo kot:

$$\mathbf{P}_0^- = \begin{bmatrix} 1000 & 1000 & 1000 & 1000 \\ 1000 & 1000 & 1000 & 1000 \\ 1000 & 1000 & 1000 & 1000 \\ 1000 & 1000 & 1000 & 1000 \end{bmatrix}. \tag{4.25}$$

Moramo še poudariti, da ima izbira drugačne $\mathbf{P}_0^-$ ob predpostavki, da imajo elementi velike vrednosti (> 100), zanemarljivo majhen vpliv na končni rezultat.

Delovanje algoritma

Tako smo opredelili še zadnje gradnike, potrebne za delovanje našega algoritma.

Za točko $\mathbf{k}_{j,0}$, ki definira stanje sistema $\hat{\mathbf{x}}_{j,0}$, sedaj uporabimo diskretni linearni Kalmanov filter. Pri tem upoštevamo njeno pripadajočo meritev $\mathbf{z}_{l,1}$. Tako dobimo za to točko njeno novo stanje $\hat{\mathbf{x}}_{j,1}$. Ob tem seveda dobimo tudi ažurirano kovariančno matriko napake $\mathbf{P}_{j,2}^-$. Ko tako obdelamo vse točke s krivulje k_0, imamo s tem določen najboljši približek objekta v sliki 1, tj. k_1. Stanje sistema $\hat{\mathbf{x}}_{j,1}$ namreč hrani x in y-koordinati točk ter ustrezen premik po x in y (glej enačbo (4.14)). Krivuljo k_1 torej dobimo tako, da medsebojno povežemo vse točke, ki jih definirata prvi in drugi element v posameznem stanju sistema $\hat{\mathbf{x}}_{j,1}$. Točke Υ_j, $\Upsilon_j(\hat{x}_{j,1}, \hat{y}_{j,1})$, ki jih dobimo na takšen način, pa so lahko tudi po več pikslov narazen. S tem je omogočeno širjenje krivulje k iz slike v sliko. To pa tudi pomeni, da ima v tem primeru krivulja k_1 več elementov od k_0 in s tem od elementov v polju stanja sistema. Za nove elemente, ki jih ni v polju stanja sistema, je treba opraviti interpolacijo za vrednosti premika po x in y. Elementu $\mathbf{k}_{l,1}$ iz k_1 zato priredimo vrednosti premikov iz novega stanja $\hat{\mathbf{x}}_{j,1}$ in matriko $\mathbf{P}_{j,2}^-$, če je razdalja med $\mathbf{k}_{l,1}$ in točko Υ_j najmanjša, oz. formalno zapisano:

$$\begin{aligned} (\mathbf{x}'_{l,1}, \mathbf{P}'^{-}_{l,2}) \;\; = \;\; & \Big\{([k_{x_{l,1}}, k_{y_{l,1}}, \triangle\hat{x}_{j,1}, \triangle\hat{y}_{j,1}]^T, \mathbf{P}_{j,2}^-), \\ & \|\mathbf{k}_{l,1} - \Upsilon_j\| < \|\mathbf{k}_{l,1} - \Upsilon_p\| \quad \forall p \in [0, m_0 - 1] \,\wedge\, j \neq p\Big\}, \end{aligned} \tag{4.26}$$

kjer sta z $(\mathbf{x}'_{l,1}, \mathbf{P}'^{-}_{l,2})$ označena stanje sistema in kovariančna matrika za točko $\mathbf{k}_{l,1}$, $\triangle\hat{x}_{j,1}$ ter $\triangle\hat{y}_{j,1}$ pa označujeta vrednosti premika po x in y za stanje sistema $\mathbf{x}_{j,1}$ (glej enačbo (4.14)). Enačba (4.26) dejansko pove, da vsaki novi točki, ki je nastala pri povezovanju točk Υ_{j-1} in Υ_j, priredimo vrednosti premika in matriko $\mathbf{P}$, kot ju ima njena najbližja soseda (ali Υ_j ali Υ_{j-1}). Uporabili smo torej interpolacijo na osnovi najbližjega soseda. Možno je seveda uporabiti tudi druge vrste interpolacije (linearno, kvadratno, kubično itd.).

Podobno ravnamo v primeru, ko se krivulja k krči. Takrat ima krivulja k_1 manj elementov od k_0. Ohranimo le tista stanja sistema $\hat{\mathbf{x}}_{j,1}$, ki izpolnjujejo pogoj v (4.26).

Ko opravimo ažuriranje stanj za vse točke krivulje k_1, potem pomožno stanje sistema $\mathbf{x}'_{l,1}$ in kovariančno matriko $\mathbf{P}'^{-}_{l,2}$ priredimo pravemu stanju sistema $\hat{\mathbf{x}}_{l,1}$ in kovariančni matriki $\mathbf{P}^{-}_{l,2}$.

Zatem se premaknemo v sliko S_2 in jo obdelamo po istem postopku. Najprej izločimo objekt O in določimo zanj meritev z_2. Nato izračunamo ujemanje med k_1 in z_2. Vsako točko $\mathbf{k}_{j,1}$ s k_1 obdelamo s Kalmanovim filtrom. Iz novih stanj določimo najboljši približek objekta v sliki S_2, torej krivuljo k_2. Na koncu ažuriramo stanja sistema in kovariančnih matrik za vse točke $\mathbf{k}_{j,2}$ ter postopek nadaljujemo na naslednji sliki. Postopek končamo, ko v naslednji sliki ne najdemo več objekta O.

Zapišimo še algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije v strnjeni obliki.

VHOD: Zaporedje segmentiranih in označenih slik $\mathbf{S}$*. Oznaka objekta, ki ga razpoznavamo, je* O*.*

IZHOD: Zaporedje najboljših približkov k_i *objekta* O *na posamezni sliki* S_i*.*

1. *i=0*
2. *Poišči prvo sliko iz zaporedja* $\mathbf{S}$*, kjer se pojavi objekt z oznako* O*. Označi to sliko s* S_0*.*
3. *Določi notranjo mejo objekta* O*. To je najboljši približek objekta* O *v koraku 0, popisan s krivuljo* k_0*.*
4. *Vsaki točki* $\mathbf{k}_{j,0}$ *s krivulje* k_0*:*
   a. *po enačbi (4.21) določi začetno stanje sistema* $\hat{\mathbf{x}}_{j,0}$*,*
   b. *po enačbi (4.25) določi matriko* $\mathbf{P}^{-}_{j,1}$*.*
5. *ZANKA: i=i+1*
6. *Na osnovi pogoja ujemanja v (4.13) poišči na sliki* S_i *objekt* O*.*
7. *Če objekta* O *nismo našli, potem KONEC, sicer nadaljuj.*
8. *Določi notranjo mejo objekta* O*. Označi dano krivuljo kot meritev* z_i*.*
9. *Za vsako točko* $\mathbf{k}_{j,i-1}$ *s krivulje* k_{i-1}*:*
   a. *po enačbi (4.23) določi pripadajočo točko* $\mathbf{z}_{l,i}$ *s krivulje meritve,*
   b. *po enačbah (4.24) določi matriki* $\mathbf{Q}_i$ *in* $\mathbf{R}_i$ *ter po enačbah (4.16) in (4.19) matriki* $\mathbf{\Phi}$ *in* $\mathbf{H}$*,*
   c. $[\hat{\mathbf{x}}_{j,i}, \mathbf{P}^{-}_{j,i+1}]$ = *KalmanovFilter(*$\hat{\mathbf{x}}_{j,i-1}$, $\mathbf{z}_{l,i}$, $\mathbf{P}^{-}_{j,i}$, $\mathbf{Q}_i$, $\mathbf{R}_i$, $\mathbf{\Phi}$, $\mathbf{H}$*).*

10. *Iz stanj sistema* $\hat{\mathbf{x}}_{j,i}$ *tvori najboljši približek objekta* k_i.
11. *Za vsako točko* $\mathbf{k}_{j,i}$ *s krivulje* k_i*:*

    a. *ažuriraj stanje sistema* $\mathbf{x}'_{j,i}$ *in* $\mathbf{P}'^{-}_{j,i+1}$ *po enačbi (4.26).*

12. *Priredi* $\hat{\mathbf{x}}_{:,i} = \mathbf{x}'_{:,i}$ *in* $\mathbf{P}^{-}_{:,i+1} = \mathbf{P}'^{-}_{:,i+1}$. *Vrni se na vrstico z oznako ZANKA.*

Algoritem 3 Osnovna oblika algoritma za razpoznavanje objektov iz zaporedja slik s postopki predikcije.

4.2.2 Problem velikih napak v meritvah

V prejšnjem podpoglavju smo opisali osnovno obliko novega algoritma za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Psevdokod je v algoritmu 3. Videli smo, da iz prve slike, kjer se pojavi objekt, določimo začetno stanje sistema, obdelovanje prediktor-korektor s KF pa začnemo šele v drugi sliki. Matriki $\mathbf{Q}$ in $\mathbf{R}$ sta bili skozi celotno zaporedje konstantni. Dobro je znano [Rao96], da so v primeru, ko naredimo kovariančno matriko konstantno, vse ocene, dobljene s KF, zreducirane na ocene, dobljene po metodi najmanjših kvadratov. Vemo pa tudi, da so ocene po metodi najmanjših kvadratov zelo občutljive na velike napake v podatkih – torej podatkih, ki ležijo daleč stran od gruče opazovanih podatkov.

Kovariančne matrike $\mathbf{P}$ sicer nismo eksplicitno postavili na konstantno vrednost, vendar, ker sta tako $\mathbf{Q}$ in $\mathbf{R}$ konstantni, postane po določenem številu korakov tudi kovariančna matrika konstantna (enačbe (2.28), (2.30) in (2.32)). To pomeni, da naš algoritem po določenem številu korakov začne vračati rezultate, kot da bi bili dobljeni po metodi najmanjših kvadratov.

Na sliki 4.3 je prikazan problem velikih napak v meritvah na primeru razpoznavanja jajčnih mešičkov. V koraku $i+1$ opazimo veliko napako v meritvi (levi stolpec), kar povzroči tudi veliko napako v najboljšem približku objekta v koraku $i+1$ (desni stolpec).

Na osnovi tega primera lahko sklepamo, da algoritem ne sme vsako meritev upoštevati enako. Meritev, ki pade daleč stran od predvidenih vrednosti, je najverjetneje napaka in jo moramo zato upoštevati z zelo majhno utežjo – velike vrednosti v matriki $\mathbf{R}$. Če se takšne meritve pojavljajo skozi daljši čas, pa se mora Kalmanov filter teh modelnih parametrov ponovno priučiti. Torej, gre za problem določitve kriterijske funkcije J, ki se bo na osnovi predvidenih vrednosti in meritve odločila, koliko upoštevati oz. zavrniti meritev.

Splošne kriterijske funkcije J ni mogoče skonstruirati, saj je prav tako kakor vsaka segmentacijska metoda odvisna od tipa objekta, ki ga razpoznavamo. V splošnem lahko

Korak i

Korak $i+1$

Slika 4.3: Problem velikih napak v meritvah. Par slik za posamezni korak prikazuje meritev na levi in najboljši približek objekta, določen z našim algoritmom, na desni sliki. Obe krivulji sta položeni na originalno ultrazvočno sliko. V koraku $i+1$ je velika napaka v meritvi.

opazujemo od slike do slike spremembe v ploščini objekta, obsegu, težišču, kompaktnosti, ekscentričnosti, povprečni sivini itd.

V nadaljevanju bomo prikazali primer takšne kriterijske funkcije, in sicer za naše testne objekte – jajčne mešičke. O primernosti posamezne meritve se bomo odločali le na osnovi ploščine in kompaknosti meritve. Če se ploščina in kompaktnost preveč razlikujeta od predhodnih meritev, potem meritev kaznujemo s tem, da ji povečamo vrednosti v matriki $\mathbf{R}$, sicer pa ne.

Primerjava zgolj vrednosti predhodne in trenutne meritve pa vnaša v vsak sistem še dodatno nejasnost. Ponavadi zato upoštevamo povprečja predhodnih meritev, iz optimalne teorije sistemov pa vemo, da je najbolje, če predhodne vrednosti vgradimo v Kalmanov filter. Ta koncept smo uporabili tudi mi.

Tako smo ob glavnem Kalmanovem filtru (glej algoritem 3) v algoritem uvedli še en Kalmanov filter, imenujmo ga KF J, na osnovi katerega ocenjujemo "ustreznost" meritev za glavni Kalmanov filter. Stanje sistema za ta KF J, $\mathbf{x}_i^J$, popišemo s ploščino p, spremembo ploščine $\triangle p$ in kompaktnostjo κ meritve (regije), kar označimo:

$$\mathbf{x}_i^J = \begin{bmatrix} p_i^J \\ \triangle p_i^J \\ \kappa_i^J \end{bmatrix}. \tag{4.27}$$

Iz te enačbe opazimo, da smo kompaktnost modelirali kot konstanto vrednost z dodanim majhnim šumom (glej še enačbo (4.15)). Takšno modeliranje je posledica načina izvajanja ultrazvočnih pregledov. Ko pri ultrazvočnem pregledu preglejujemo mešiček z enega konca proti drugemu, dobimo prereze skozi mešiček (segmentirana slika). Ti prerezi so na začetku majhni in ovalni, toda čim bolj se približujemo sredini mešička, tem večji postajajo. Ko se bližamo drugemu koncu mešička, pa seveda spet postajajo manjši. Ob tej rasti/krčenju se oblika mešička ne spreminja bistveno. Kompaktnost pa je ena izmed značilk, s katero lahko popišemo obliko. Kompaktnost mešičkov lahko zato modeliramo kot konstanto.

Matrika $\mathbf{\Phi}^J$ je tako naslednje oblike:

$$\mathbf{\Phi}^J = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.28}$$

Matrika $\mathbf{\Phi}^J$ torej določa (enačba (4.28)), da novo predvideno vrednost za ploščino dobimo iz stare vrednosti ploščine, kateri prištejemo spremembo ploščine, vrednosti spremembe ploščine in kompaktnosti pa se le preneseta iz prejšnjega koraka.

Začetno stanje sistema $\hat{\mathbf{x}}_0^J$ določimo iz lastnosti objekta na prvi sliki, in sicer tako, da mu določimo ploščino in kompaktnost:

$$\hat{\mathbf{x}}_0^J = \begin{bmatrix} p(z_0) \\ 0 \\ \kappa(z_0) \end{bmatrix}. \tag{4.29}$$

Z merilnim sistemom KF J lahko izmerimo le novo vrednost ploščine $p(z_i)$ in kompaktnosti $\kappa(z_i)$ za meritev z_i. Torej je meritev $\mathbf{z}_i^J$ oblike:

$$\mathbf{z}_i^J = \begin{bmatrix} z_{p_i^J} \\ z_{\kappa_i^J} \end{bmatrix} = \begin{bmatrix} p(z_i) \\ \kappa(z_i) \end{bmatrix}. \tag{4.30}$$

Iz tega sledi (glej še enačbo (4.18)), da je matrika $\mathbf{H}^J$, ki povezuje meritve s sistemom, oblike:

$$\mathbf{H}^J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.31}$$

Določiti moramo le še matriki šuma $\mathbf{Q}^J$ in $\mathbf{R}^J$ ter začetno kovariančno matriko $\mathbf{P}_0^{J-}$. Iz istega razloga kot pri globalnem Kalmanovem filtru smo matriki za šum v koraku 0 postavili na:

$$\mathbf{Q}_0^J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{in} \qquad \mathbf{R}_0^J = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.32}$$

Začetno kovariančno matriko $\mathbf{P}_0^{J-}$ pa postavimo na velike vrednosti pri ploščini, ter na majhne vrednosti pri kompaknosti, kjer smo veliko bolj prepričani o pravilnosti modelnih parametrov. Tako je kovariančna matrika oblike

$$\mathbf{P}_0^{J-} = \begin{bmatrix} 1000 & 1000 & 0 \\ 1000 & 1000 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.33}$$

S tem imamo podane vse količine tudi za Kalmanov filter J. Ta filter vzpostavi začetno stanje v prvi sliki S_0, obdelovanje prediktor-korektor pa se začne v drugi sliki. Delovanje filtra poteka po že znanem scenariju. Včitamo objekt s slike, zanj določimo ploščino in kompaktnost. Najboljšo oceno pa zatem dobimo, ko meritev vstavimo v KF.

Na osnovi rezultatov Kalmanovega filtra J pa se odločamo, koliko lahko zaupamo meritvi z_i. Naj še enkrat poudarimo, da je to modeliranje prirejeno za jajčne mešičke, saj kot vemo, splošne kriterijske funkcije J ni možno vzpostaviti. V našem primeru opazujemo razliko med predvidenimi vrednostmi $\hat{\mathbf{x}}_i^{J-}$ za korak i, ki jih vrača KF J, in vrednostmi meritve $\mathbf{z}_i^J$. Predvidene vrednosti dobimo po formuli

$$\hat{\mathbf{x}}_i^{J-} = \begin{bmatrix} \hat{p}_i^{J-} \\ \triangle\hat{p}_i^{J-} \\ \hat{\kappa}_i^{J-} \end{bmatrix} = \mathbf{\Phi}^J \cdot \hat{\mathbf{x}}_{i-1}^J. \tag{4.34}$$

Razliko med obema vrednostima opazujemo tako za ploščino kot za kompaktnost. Najprej si poglejmo vrednotenje pri ploščini. Tukaj opazujemo razmerje sf med predvideno ploščino meritve $\hat{p}_i^{J-}$ (glej enačbi (4.34) in (4.30) za uporabljene oznake) in dejansko izmerjeno ploščino $z_{p_i^J}$ $(= p(z_i))$. Sorazmernostni faktor sf definiramo kot razmerje med večjo in manjšo ploščino od obeh ploščin:

$$sf = \frac{\max\{\hat{p}_i^{J-}, z_{p_i^J}\}}{\min\{\hat{p}_i^{J-}, z_{p_i^J}\}}. \tag{4.35}$$

Razmerje sf je zmeraj večje ali enako 1 in beleži, kakšno je razmerje med predvideno in dejansko ploščino meritve. Če je to razmerje zelo veliko, potem je zelo verjetno, da je nova meritev z_i daleč stran od prejšnjih podatkov in jo zato manj upoštevamo v globalnem Kalmanovem filtru. Slabo meritev pa kaznujemo tako, da povečamo vrednosti elementov v matriki R. Kazen f_{plos} za meritev z_i določimo na osnovi razmerja sf po neki kazenski funkciji. Izbrali smo eksponetno kazensko funkcijo in ob tem postavili dva robna pogoja: a) za $sf \to 1$ naj vrača vrednosti f_{plos} okrog 1 (ni kazni), b) za $sf = 1,5$ naj vrne $f_{plos} = 10$. Kazen f_{plos} torej določimo kot:

$$f_{plos} = 100^{(sf-1)}. \tag{4.36}$$

Prag v pogoju b) dejansko pove, da mora biti ena od obeh ploščin za polovico večja od druge, šele potem jo resneje kaznujemo. To je zelo ohlapno.

Podobno ravnamo pri opazovanju kompaktnosti. Najprej izračunamo razliko med izmerjeno kompaktnostjo meritve $z_{\kappa_i^J}$ in predvideno kompaktnostjo $\hat{\kappa}_i^{J-}$:

$$\triangle\kappa = z_{\kappa_i^J} - \hat{\kappa}_i^{J-}.$$

Na osnovi te razlike pa se odločimo, kako bomo kaznovali meritev. Če je ta razlika večja od 10, šele potem meritev resneje kaznujemo. Tudi kazen f_{komp} določimo po eksponentni funkciji in ob upoštevanju robnega pogoja dobimo

$$f_{komp} = 10^{(\frac{\triangle\kappa}{10})}. \tag{4.37}$$

Če pa je ta razlika manjša od 0, potem meritve ne kaznuje. Negativna razlika namreč pomeni, da je nova meritev bolj kompaktna od prejšnjih, kar pa nam seveda ustreza, in zato meritve ne kaznujemo ($f_{komp} = 1$).

Obe izračunani kazni, f_{plos} in f_{komp}, nam določata, s kakšno težo bomo v globalnem Kalmanu upoštevali meritev z_i (algoritem 3). Upoštevamo ju tako, da za globalni Kalmanov filter ustrezno spremenimo matriko šuma $\mathbf{R}$. To naredimo tako, da vse elemente osnovne matrike šuma (enačba (4.24)) pomnožimo s produktom obeh kazni. Tako se sedaj matrika $\mathbf{R}$ spreminja od slike do slike. Njena končna oblika pa je naslednja:

$$\mathbf{R}_i = (f_{plos} \cdot f_{komp}) \cdot \mathbf{R}_0. \tag{4.38}$$

Matrika $\mathbf{R}_i$ bo imela visoke vrednosti elementov, če bo meritev močno odstopala od predhodnih rezultatov, sicer pa nizke. V prvem primeru takšne meritve ne bomo resneje upoštevali (vrednosti so lahko tudi reda 10^8), medtem ko se v drugem primeru na meritev močno zanesemo. To razmišljanje upoštevamo tudi pri ažuriranju Kalmanovega filtra J. Če bi namreč pustili, da matriki $\mathbf{Q}^J$ in $\mathbf{R}^J$ vedno enakomerno upoštevata meritev, bi s tem povzročili, da bi problem velikih napak v podatkih oslabil delovanje KF J. Tako s KF J ne bi prav ničesar pridobili. Zato tudi v tem KF na osnovi obeh kazni ažuriramo matriko šuma $\mathbf{R}^J$. Kazen f_{plos} vpliva na ažuriranje ploščine, medtem ko kazen f_{komp} na ažuriranje kompaktnosti. Končna oblika matrike šuma v meritvah je naslednja

$$\mathbf{R}_i^J = \begin{bmatrix} f_{plos} & 0 \\ 0 & f_{komp} \end{bmatrix}. \tag{4.39}$$

Strnimo celoten postopek, opisan v tem podpoglavju, in zapišimo psevdokod algoritma za razpoznavanje objektov iz zaporedja slik s postopki predikcije, kjer je odpravljen problem velikih napak v meritvah. Dodani koraki so uokvirjeni.

VHOD: Zaporedje segmentiranih in označenih slik $\mathbf{S}$. *Oznaka objekta, ki ga razpoznavamo, je* O.

IZHOD: Zaporedje najboljših približkov k_i *objekta* O *na posamezni sliki* S_i.

1. *i=0*
2. *Poišči prvo sliko iz zaporedja* $\mathbf{S}$, *kjer se pojavi objekt z oznako* O. *Označi to sliko s* S_0.
3. *Določi notranjo mejo objekta* O. *To je najboljši približek objekta* O *v koraku 0, popisan s krivuljo* k_0.
4. *Vsaki točki* $\mathbf{k}_{j,0}$ *s krivulje* k_0:
   - a. *po enačbi (4.21) določi začetno stanje sistema* $\hat{\mathbf{x}}_{j,0}$,
   - b. *po enačbi (4.25) določi matriko* $\mathbf{P}^{-}_{j,1}$.
5. $\boxed{5.}$ *Določi začetno stanje sistema* $\hat{\mathbf{x}}^J_0$ *Kalmanovega filtra J po enačbi (4.29) ter matriko* $\mathbf{P}^{J-}_1$ *po enačbi (4.33).*
6. *ZANKA: i=i+1*
7. *Na osnovi pogoja ujemanja v (4.13) poišči na sliki* S_i *objekt* O.
8. *Če objekta O nismo našli, potem KONEC, sicer nadaljuj.*
9. *Določi notranjo mejo objekta* O. *Označi dano krivuljo kot meritev* z_i.
10. $\boxed{10.}$ *Meritev* $\mathbf{z}^J_i$ *za krivuljo* z_i *določi po enačbi (4.30).*
11. $\boxed{11.}$ *Ažuriraj matriko šuma* $\mathbf{R}_i$ *za globalni Kalmanov filter po enačbi (4.38).*
12. $\boxed{12.}$ *Določi* $\mathbf{\Phi}^J$, $\mathbf{H}^J$ *in* $\mathbf{Q}^J_i$ *po enačbah (4.28), (4.31) in (4.32).*
13. $\boxed{13.}$ *Ažuriraj matriko šuma* $\mathbf{R}^J_i$ *za Kalmanov filter J po enačbi (4.39).*
14. $\boxed{14.}$ $[\hat{\mathbf{x}}^J_i, \mathbf{P}^{J-}_{i+1}] =$ *KalmanovFilter(*$\hat{\mathbf{x}}^J_{i-1}$, $\mathbf{z}^J_i$, $\mathbf{P}^{J-}_i$, $\mathbf{Q}^J_i$, $\mathbf{R}^J_i$, $\mathbf{\Phi}^J, \mathbf{H}^J$*)*
15. *Za vsako točko* $\mathbf{k}_{j,i-1}$ *s krivulje* k_{i-1}:
    - a. *po enačbi (4.23) določi pripadajočo točko* $\mathbf{z}_{l,i}$ *s krivulje meritve,*
    - b. *po enačbi (4.24) določi matriko* $\mathbf{Q}_i$ *ter po enačbah (4.16) in (4.19) matriki* $\mathbf{\Phi}$ *in* $\mathbf{H}$,
    - c. $[\hat{\mathbf{x}}_{j,i}, \mathbf{P}^{-}_{j,i+1}] =$ *KalmanovFilter(*$\hat{\mathbf{x}}_{j,i-1}$, $\mathbf{z}_{l,i}$, $\mathbf{P}^{-}_{j,i}$, $\mathbf{Q}_i$, $\mathbf{R}_i, \mathbf{\Phi}, \mathbf{H}$*).*
16. *Iz stanj sistema* $\hat{\mathbf{x}}_{j,i}$ *tvori najboljši približek objekta* k_i.
17. *Za vsako točko* $\mathbf{k}_{j,i}$ *s krivulje* k_i:
    - a. *ažuriraj stanje sistema* $\mathbf{x}'_{j,i}$ *in* $\mathbf{P}'^{-}_{j,i+1}$ *po enačbi (4.26).*

18. Priredi $\hat{\mathbf{x}}_{:,i} = \mathbf{x}'_{:,i}$ *in* $\mathbf{P}^-_{:,i+1} = \mathbf{P}'^-_{:,i+1}$. *Vrni se na vrstico z oznako ZANKA.*

Algoritem 4 Algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije – odpravljen problem velikih napak v meritvah. Dodani koraki so uokvirjeni.

4.2.3 Upoštevanje lokalnosti krivulje

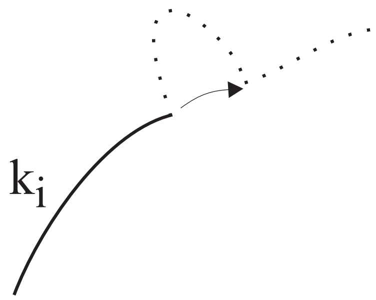
Algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije, dopolnjena verzija je v algoritmu 4, smo zasnovali na ideji, da sledimo vsaki točki krivulje iz slike v sliko. Sledenje smo realizirali za vsako točko s krivulje neodvisno, kar pomeni, da nismo nikjer upoštevali sosednih točk. Do določene mere to predstavlja slabost našega pristopa, saj bi z uvedbo mere, ki bi beležila lokalnost krivulje, dobili večji nadzor nad obliko krivulje. Da ni upoštevana lokalnost krivulje, se opazi tudi v matrikah šuma v sistemu in pri meritvah, saj se obe matriki določita v vsakem koraku in sta za vse točke s krivulje enaki (glej točko 15.c algoritma 4).

V realnih razmerah pa vse točke s krivulje zagotovo nimajo istih lastnosti (ene točke so namreč bližje pravemu položaju objekta kot druge) in je zato predpostavka o enakih matrikah $\mathbf{Q}$ in $\mathbf{R}$ neveljavna. V tem podpoglavju bomo zato razvili postopek za ocenjevanje lokalnosti krivulje in pokazali, kako lahko te ocene uporabimo v globalnem Kalmanovem filtru. Pri tem bomo privzeli, da so objekti, ki jih razpoznavamo, kompaktni.

Krivulja k_i naj kakor do sedaj označuje najboljši približek objekta v sliki i. Predstavljajmo si, da hodimo po krivulji npr. v smeri urinega kazalca. Na osnovi že prehojene poti lahko predvidimo, kje bi morala potekati krivulja v naslednji točki. Iz razlike med predvidenim in dejanskim položajem točke pa se odločimo, kako upoštevamo to točko. Na sliki 4.4 je ideja prikazana shematsko. S polno črto je označena že prehojena pot po krivulji. Na njeni osnovi lahko predvidimo naslednji položaj – označeno s puščico. S pikicami je označen dejanski potek krivulje. Jasno je nekaj: čim bolj je krivulja nekompaktna, razgibana v določenem delu, tem večja je razlika med predvidenim in dejanskim položajem, in seveda obratno. S tem postopkom navidezno izvajamo glajenje krivulje. Rezultate upoštevamo v globalnem Kalmanovem filtru. V delih, kjer je krivulja že gladka, se zanesemo na modelne parametre, v drugih delih pa raje na meritev.

Zgoraj opisano idejo smo izvedli s pomočjo Kalmanovega filtra. Označimo ta Kalmanov filter z LOK. Dejansko izvajamo sledenje točki po krivulji. Uporabimo lahko popolnoma enake matrike kot pri globalnem KF. Potrebne matrike imajo torej obliko:

$$\mathbf{\Phi}^{LOK} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{H}^{LOK} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$



Slika 4.4: Ocenjevanje lokalnosti krivulje.

$$\mathbf{Q}^{LOK} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{R}^{LOK} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

$$\mathbf{P}_0^{LOK-} = \begin{bmatrix} 1000 & 1000 & 1000 & 1000 \\ 1000 & 1000 & 1000 & 1000 \\ 1000 & 1000 & 1000 & 1000 \\ 1000 & 1000 & 1000 & 1000 \end{bmatrix}. \tag{4.40}$$

Začetno stanje sistema, $\hat{\mathbf{x}}_0^{LOK}$, vzpostavimo iz koordinat prvega elementa krivulje k_i, in sicer:

$$\hat{\mathbf{x}}_0^{LOK} = \begin{bmatrix} k_{x_{0,i}} \\ k_{y_{0,i}} \\ 0 \\ 0 \end{bmatrix}. \tag{4.41}$$

Meritev $\mathbf{z}_j^{LOK}$ pa vsebuje zgolj koordinati x in y j-te točke s krivulje k_i. Shema sledenja je znana in smo jo že večkrat opisali. Sledenje po krivulji začnemo v drugi točki. Iz stanja sistema v koraku 0, $\hat{\mathbf{x}}_0^{LOK}$, ter meritve $\mathbf{z}_1^{LOK}$ določimo po Kalmanovem filtru novo oceno stanja sistema v koraku 1, $\hat{\mathbf{x}}_1^{LOK}$. Premaknemo se v naslednjo točko, določimo novo meritev ter izračunamo nov najboljši položaj. Postopek ponavljamo, dokler ne obhodimo vseh točk krivulje.

Sledenje krivulji integriramo v naš algoritem. Na osnovi rezultatov Kalmanovega filtra LOK se odločimo, kako bomo upoštevali modelne parametre in meritev v globalnem Kalmanovem filtru. Opazujemo razliko oz. razdaljo med predvideno vrednostjo točke Υ, ki jo dobimo iz prvih dveh elementov stanja sistema $\hat{\mathbf{x}}_j^{LOK-}$, $\Upsilon(\hat{x}_j^{LOK-}, \hat{y}_j^{LOK-})$, in dejansko točko $\mathbf{k}_{j,i}$ s krivulje k_i:

$$\omega = \|\Upsilon - \mathbf{k}_{j,i}\|. \tag{4.42}$$

Če je krivulja v tem delu že gladka, torej je $\omega \leq 1$, potem se bolj zanesemo na model kakor na meritev, sicer pa na meritev. Formalno to zapišemo kot:

$$\mathbf{Q}_{j,i} = \begin{cases} \mathbf{Q}_i, & \text{če} \quad \omega \leq 1 \\ \omega \cdot \mathbf{Q}_i, & \text{sicer} \end{cases} \tag{4.43}$$

in

$$\mathbf{R}_{j,i} = \begin{cases} \mathbf{R}_i/\omega, & \text{če} \quad \omega \leq 1 \\ \mathbf{R}_i, & \text{sicer} \end{cases} . \tag{4.44}$$

Maksimalna razdalja ω je lahko 2. Prag za razdaljo smo postavili kar na polovico maksimalne možne vrednosti. Kazensko funkcijo, s katero ažuriramo matriki $\mathbf{Q}$ in $\mathbf{R}$, pa smo vzeli kar najbolj preprosto možno, in sicer vrednost razdalje (oz. njeno obratno vrednost). Seveda je možno uporabiti tudi druge smiselne prage in kazenske funkcije. Detajlno tega problema nismo študirali. Prepričani pa smo, da izbira drugačnih parametrov ne spremeni bistveno končnega rezultata.

Podajmo tako modificirano obliko algoritma za razpoznavanje objektov s postopki predikcije v strnjeni obliki. Novi, dodani koraki so spet uokvirjeni.

VHOD: Zaporedje segmentiranih in označenih slik $\mathbf{S}$. *Oznaka objekta, ki ga razpoznavamo, je* O.

IZHOD: Zaporedje najboljših približkov k_i *objekta* O *na posamezni sliki* S_i.

1. *i=0*
2. *Poišči prvo sliko iz zaporedja* $\mathbf{S}$, *kjer se pojavi objekt z oznako* O. *Označi to sliko s* S_0.
3. *Določi notranjo mejo objekta* O. *To je najboljši približek objekta* O *v koraku 0, popisan s krivuljo* k_0.
4. *Vsaki točki* $\mathbf{k}_{j,0}$ *s krivulje* k_0:
   a. *po enačbi (4.21) določi začetno stanje sistema* $\hat{\mathbf{x}}_{j,0}$,
   b. *po enačbi (4.25) določi matriko* $\mathbf{P}^{-}_{j,1}$.
5. *Določi začetno stanje sistema* $\hat{\mathbf{x}}^J_0$ *Kalmanovega filtra* J *po enačbi (4.29) ter matriko* $\mathbf{P}^{J-}_1$ *po enačbi (4.33).*
6. *ZANKA: i=i+1*
7. *Na osnovi pogoja ujemanja v (4.13) poišči v sliki* S_i *objekt* O.
8. *Če objekta* O *nismo našli, potem KONEC, sicer nadaljuj.*
9. *Določi notranjo mejo objekta* O. *Označi dano krivuljo kot meritev* z_i.
10. *Meritev* $\mathbf{z}^J_i$ *za krivuljo* z_i *določi po enačbi (4.30).*

11. *Ažuriraj matriko šuma* $\mathbf{R}_i$ *za globalni Kalmanov filter po enačbi (4.38).*
12. *Določi* $\mathbf{\Phi}^J$, $\mathbf{H}^J$ *in* $\mathbf{Q}_i^J$ *po enačbah (4.28), (4.31) in (4.32).*
13. *Ažuriraj matriko šuma* $\mathbf{R}_i^J$ *za Kalmanov filter J po enačbi (4.39).*
14. $[\hat{\mathbf{x}}_i^J, \mathbf{P}_{i+1}^{J-}] = \mathit{KalmanovFilter}(\hat{\mathbf{x}}_{i-1}^J, \mathbf{z}_i^J, \mathbf{P}_i^{J-}, \mathbf{Q}_i^J, \mathbf{R}_i^J, \mathbf{\Phi}^J, \mathbf{H}^J)$
15. *Določi začetno stanje* $\hat{\mathbf{x}}_0^{LOK}$ *Kalmanovega filtra LOK po enačbi (4.41) in kovariančno matriko* $\mathbf{P}_0^{LOK-}$ *po enačbi (4.40).*
16. *Za vsako točko* $\mathbf{k}_{j,i-1}$ *s krivulje* k_{i-1}*:*
    a. *po enačbi (4.23) določi pripadajočo točko* $\mathbf{z}_{l,i}$ *s krivulje meritve,*
    b. *določi meritev* $\mathbf{z}_j^{LOK} = [k_{x_{j,i-1}}, k_{y_{j,i-1}}]^T$,
    c. $[\hat{\mathbf{x}}_j^{LOK}, \mathbf{P}_{j+1}^{LOK-}] = \mathit{KalmanovFilter}(\hat{\mathbf{x}}_{j-1}^{LOK}, \mathbf{z}_j^{LOK}, \mathbf{P}_j^{LOK-}, \mathbf{Q}_i^{LOK}, \mathbf{R}_i^{LOK}, \mathbf{\Phi}^{LOK}, \mathbf{H}^{LOK})$,
    d. *po enačbi (4.24) določi matriko* $\mathbf{Q}_i$ *ter po enačbah (4.16) in (4.19) matriki* $\mathbf{\Phi}$ *in* $\mathbf{H}$,
    e. *po enačbah (4.43) ter (4.44) ažuriraj matriki* $\mathbf{Q}_{j,i}$ *in* $\mathbf{R}_{j,i}$,
    f. $[\hat{\mathbf{x}}_{j,i}, \mathbf{P}_{j,i+1}^-] = \mathit{KalmanovFilter}(\hat{\mathbf{x}}_{j,i-1}, \mathbf{z}_{l,i}, \mathbf{P}_{j,i}^-, \mathbf{Q}_{j,i}, \mathbf{R}_{j,i}, \mathbf{\Phi}, \mathbf{H})$.
17. *Iz stanj sistema* $\hat{\mathbf{x}}_{j,i}$ *tvori najboljši približek objekta* k_i.
18. *Za vsako točko* $\mathbf{k}_{j,i}$ *s krivulje* k_i*:*
    a. *ažuriraj stanje sistema* $\mathbf{x}'_{j,i}$ *in* $\mathbf{P}'^-_{j,i+1}$ *po enačbi (4.26).*
19. *Priredi* $\hat{\mathbf{x}}_{:,i} = \mathbf{x}'_{:,i}$ *in* $\mathbf{P}^-_{:,i+1} = \mathbf{P}'^-_{:,i+1}$. *Vrni se na vrstico z oznako ZANKA.*

Algoritem 5 Algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije – upoštevana je lokalnost krivulje.

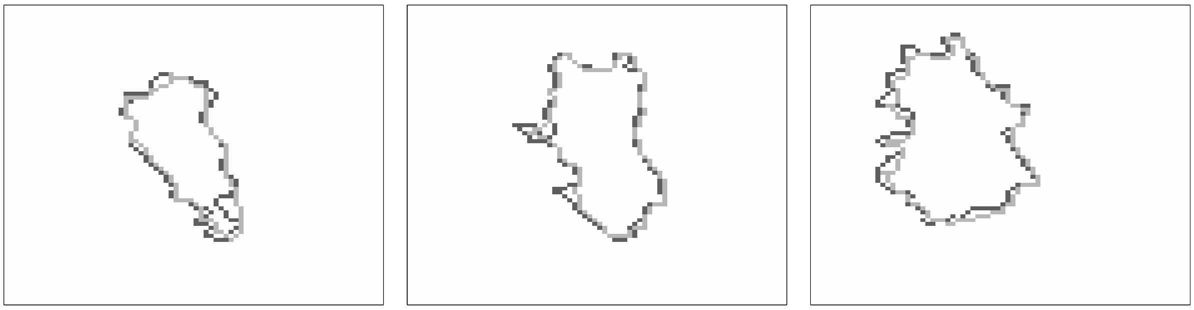
4.2.4 Neustrezna porazdelitev meritev

V algoritmu 5 smo prikazali izboljšano verzijo postopka za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Poglejmo si sedaj nekaj predhodnih rezultatov razpoznavanja na testnih slikah.

Predhodni rezultati in razprava

Razpoznavanje smo preizkusili na zaporedju slik z razpoznanimi jajčnimi mešički. Te segmentirane slike smo dobili po obdelovanju originalnega zaporedja umetnih oz. realnih ultrazvočnih slik jajčnika z mešički z našim novim algoritmom za 2D slike (podpoglavje 3.3).

Na sliki 4.5 so prikazani tipični rezultati, dobljeni pri razpoznavanju jajčnega mešička iz realnih posnetkov. Podobne rezultate dobimo tudi na umetnih ultrazvočnih slikah. S temno je označena meritev – rezultat razpoznavanja na statični 2D sliki, s sivo barvo pa rezultat po obdelavi z algoritmom 5. Prikazane so tri slike iz zaporedja slik, kjer se je pojavil ta mešiček, in sicer 5., 16. in 27. slika (indeks 0 priredimo sliki, kjer se mešiček pojavi prvič). Zaradi boljše preglednosti dobljene krivulje nismo položili na originalne slike sivin. Naj ob tem še poudarimo, da so rezultati blizu pravim položajem mešička.



Slika 4.5: Preliminarni rezultati razpoznavanja jajčnega mešička. Temna krivulja na vsaki sliki predstavlja meritev, svetlejša pa končni rezultat, dobljen z algoritmom 5. Prikazane so 5., 16. in 27. slika iz zaporedja.

Ob natančnem opazovanju rezultatov s slike 4.5 ugotovimo, da leži najboljši približek objekta, dobljen kot rezultat algoritma 5, čisto ob meritvi. Ponavadi leži celo nekaj pikslov bolj proti središču objekta in je zato manjši od meritve. Res je sicer, da je ta rezultat bolj kompakten od meritve, vendar pa po velikosti meritve ne more bistveno preseči. Razlog za to je v tem, da so izmerjeni mešički z algoritmom za statične 2D slike praviloma manjši od dejanskega mešička. Ker meritev tudi ne oscilira okrog pravega položaja mešička, najboljši približek objekta pa je vedno določen med predikcijo in meritvijo, so zato tudi rezultati algoritma 5 manjši od dejanskih mešičkov.

Ta problem lahko shematsko razložimo na hipotetičnem primeru razpoznavanja iz zaporedja slik, kjer so meritve na vsaki sliki vedno točne. Preglejujmo mešiček od levega konca proti desnemu. V takem zaporedju so prerezi skozi mešiček najprej majhni, zatem rastejo proti sredini, proti drugemu koncu mešička pa spet upadajo. V predikcijsko-korekcijskem algoritmu ne upoštevamo, da so meritve vedno točne (ne vemo ničesar o meritvi). Kot vemo, je Kalmanov filter zelo konzervativen. Povedano drugače, izredno počasi se prilagaja spremembam. Najboljši približek objekta je tako vedno določen med predikcijo in meritvijo, koliko se posamezen člen upošteva, pa je določeno z matrikami **P**, **Q** in **R**. Torej, v našem primeru bi Kalmanov filter, ki je sestavni del našega algoritma, prav zaradi svoje konzervativnosti, v prvem delu vračal predikcije, ki bi bile manjše od pravega položaja objekta (približki objekta podcenjeni), tj. meritve. To hkrati pomeni, da bi bil tudi končni najboljši razpoznani objekt, ki ga dobimo z usklajevanjem predikcije in meritve, manjši od pravega. Obratno velja za drugi del mešička (približki objekta precenjeni). V Kalmanovem filtru pa ne obstaja noben mehanizem, ki bi omogočil, da odpravimo takšne težave (da bi napihnili najboljši približek pri premajhnih meritvah in obratno).

Vse to kaže, da v našem hipotetičnem primeru zaporedne meritve prerezov skozi jajčne mešičke niso porazdeljene po Gaussovi porazdelitvi okrog pravega položaja (prava vrednost + šum, porazdeljen po Gaussu). Meritev je porazdeljena po Gaussovi porazdelitvi, če je skozi zaporedje oscilirala okoli pravega položaja – na nekaj slikah je meritev večja od pravega položaja objekta, zatem manjša itd. Raztresenost meritve okoli pravega položaja namreč zagotavlja optimalne rezultate Kalmanovega filtra.

Tudi pri razpoznavanju na dejanskih slikah smo opazili enak efekt, kot je opisan v zgornjem primeru. Ker meritve ne oscilirajo okoli pravega položaja objekta (so podcenjene), zato bo tudi najboljši končni položaj, dobljen z algoritmom 5, sicer bolj kompakten, pa vendar vedno manjši od dejanskega objekta. Obratno bi veljalo za precenjene meritve.

Modifikacija meritev jajčnih mešičkov

V prikazanih rezultatih in razpravi smo videli, da so testne meritve (razpoznani jajčni mešički) podcenjene. Zaradi tega izgubimo tudi na učinkovitosti Kalmanovega filtra, ki je bistven del algoritma 5. Možna rešitev je, da modificiramo meritve na takšen način, da so zatem normalno porazdeljene okoli pravega položaja objekta.

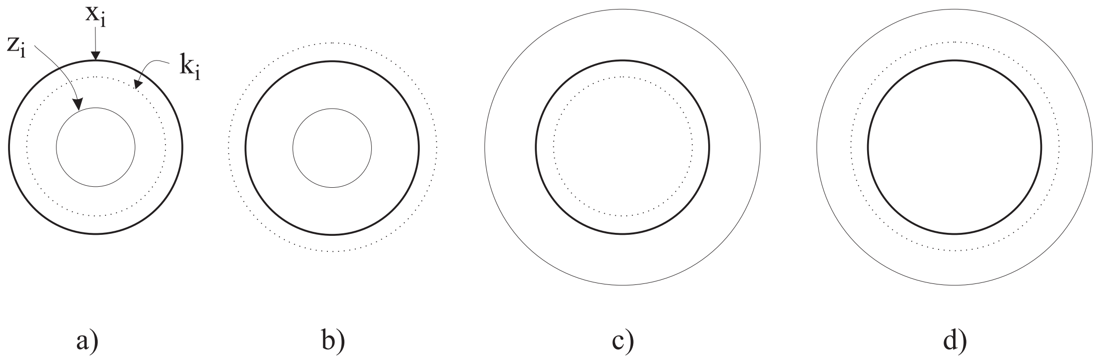
Na tem mestu bi se seveda lahko vprašali: kakšen smisel ima sploh popravljanje meritev, ali ne bi bilo bolje napisati boljši algoritem za statične 2D slike. Odgovor na to je seveda odvisen od primera do primera. V primerih, ko dobimo zgolj meritve, ob tem pa ni znano ničesar o samem algoritmu, ki je vrnil dane meritve, potem lahko zgolj popravljamo meritve. Meritve so lahko tudi rezultat neoptimalnega algoritma. V tem primeru je seveda smiselno najprej modificirati sam postopek za statične 2D slike. Rezultati algoritma za statične 2D slike se lahko tudi približujejo zgornjim mejam učinkovitosti, ki jih zaradi narave razpoznavanja in tipa slik ni mogoče preseči. Zadnji razlog pa je gotovo ta, da meritve, čeprav so optimalne, ne ustrezajo nujno pogoju Kalmanovega filtra. Če torej želimo z algoritmom 5 doseči optimalne rezultate, moramo ustrezno modificirati meritve.

Kalmanov filter predvideva, da so meritve dobljene z nekim merilnim sistemom. Šum v tem merilnem sistemu pa je modeliran kot Gaussov šum (enačba (4.18)). V našem primeru je merilni sistem algoritem za statične 2D slike. Če zaznamo v merilnem sistemu sistemsko napako (npr. meritve konstantno podcenjene), moramo za odpravo te napake ustrezno modificirati meritve (v nekaterih primerih tudi šum).

Vrnimo se sedaj k primeru naših testnih slik. Meritve smo dobili z algoritmom 2. Jedro tega algoritma je rast regij. Rast regij nadzorujemo z dvema kriterijema: sivino in obteženim gradientom. Če piksel izpolnjuje oba kriterija, potem ga dodamo objektu, sicer pa ne. Iz dobljenih rezultatov v podpoglavju 3.3 smo videli, da se lahko kljub tako modelirani rasti, kjer so kriteriji statistično nadzorovani, rast na posameznih delih mešička prehitro zaključi, v nekaterih delih pa celo preraste prave meje. Pri tem se je pokazalo, da je prva varianta veliko bolj verjetna kakor druga. To pomeni, da so meritve ponavadi manjše od dejanskega mešička. Iz razprave v uvodu k temu podpoglavju pa smo videli, da Kalmanov filter, in s tem tudi algoritem 5, nima nobenega mehanizma, ki bi lahko to upošteval, ter so zato tudi dobljeni rezultati s predikcijo in korekcijo podcenjeni. Pričakujemo lahko, da bi z ustrezno spremembo pragov pri rasti regij rešili ta problem. Če bi povečali pragove pri obeh kriterijih rasti, bi s tem le povzročili, da bi regija na večini delov prerastla meje mešičke. To bi samo pomenilo, da bi s problema premajhnih meritev prešli na problem prevelikih meritev, osnovnega problema o Gaussovi porazdelitvi meritev okoli pravega položaja pa ne bi rešili. Iz tega sklepamo, da z globalnimi pragovi lahko določimo le grobi potek razpoznanega mešička (meritve), šele z lokalnimi pragovi pa fini potek. To idejo bomo uporabili v nadaljevanju.

Na sliki 4.6 so shematsko prikazani možni tipi preliminarnih rezultatov. S polno krepko črto je prikazan pravi položaj mešička, s črtkano črto je prikazan najboljši približek objekta, ki ga dobimo po algoritmu 5, z navadno polno črto pa je označena 2D meritev. V vseh štirih primerih je najboljši približek mešička bližje pravemu položaju kakor pa meritev (obstajajo še štiri druge variante, kjer je pa meritev manj oddaljena). V primerih a) in b) je meritev manjša, v drugih dveh pa večja od pravega mešička. Kot smo že omenili, so variante, kjer je meritev manjša od pravega mešička, veliko bolj verjetne.

Če je meritev v nekaj zaporednih slikah konstantno manjša od pravega objekta, potem po določenem času postane tudi najboljši približek objekta, določen z algoritmom 5, manjši od pravega objekta. Še več, postane tudi manjši od meritve (glej razpravo v tem podpoglavju). Tega neprestanega krčenja izračunanega najboljšega položaja mešička pa Kalmanov filter brez zunanjega posega ne more preprečiti. Zato moramo v algoritem 5 vgraditi mehanizem, ki bo zaznal, da je meritev premajhna in da jo mora zato ustrezno modificirati. Seveda pa ne moremo točno vedeti, koliko je posamezna meritev napačna – ne moremo točno določiti napake $e(z_i, x_i)$. To napako lahko zgolj ocenimo na osnovi ploščine, povprečne sivine in gradienta v regiji, standardnih odklonov sivin, značilk oblike regije ipd. Napako lahko ocenjujemo tudi na osnovi razdalje med najboljšim približkom



Slika 4.6: Shematski prikaz rezultatov. S krepko črto je prikazan pravi položaj, s črtkano najboljši približek objekta, dobljen po algoritmu 5, z navadno polno črto pa meritev objekta.

objekta, k_i, in meritvijo z_i. Poglejmo si primer a) s slike 4.6. V tem primeru je možno preprosto določiti napako oz. razdaljo med obema krivuljama $e(z_i, k_i)$. V naslednji sliki bomo verjetno dobili podobne rezultate, saj so ponavadi razlike med dvema zaporednima slikama majhne (v naših testnih slikah približno 40 % pikslov ostane nespremenjenih). To pomeni, da bo najboljši približek mešička le malo spremenil obliko oz. da bo napaka ostala približno enako velika. Opazimo še nekaj. Razdaljo $e(z_i, k_i)$ s prejšnje slike lahko uporabimo za popravek meritve z_{i+1} v trenutni sliki. Meritev bi lahko popravili tako, da bi trenutni meritvi z_{i+1} prišteli razdaljo $e(z_i, k_i)$. Ponovno si pomagajmo s sliko 4.6 – primer a). Če bi meritev modificirali tako, bi nova modificirana meritev bila veliko bližje pravemu položaju mešička, s tem pa bi imeli tudi veliko bolje določen najboljši približek mešička. Podobno velja tudi za druge primere, kjer je treba v posameznih primerih zgolj uporabiti operacijo odštevanja namesto seštevanja razdalje $e(z_i, k_i)$.

Povzemimo idejo, ki jo uporabljamo pri modificiranju meritve. Na osnovi globalnih lastnosti meritve in najboljšega približka mešička se bomo najprej odločili, ali je treba meritev modificirati. Modifikacijo pa izvajamo na osnovi izračunane razdalje med najboljšim približkom mešička in meritvijo s prejšnje slike. V nadaljevanju bomo detajlno opisali vse potrebne korake za izvedbo te modifikacije in pokazali, kako jih vključiti v algoritem za razpoznavanje objektov s postopki predikcije.

Izvedba modifikacije meritve jajčnih mešičkov

V tem podpoglavju bomo opisali postopek za modificiranje meritev jajčnih mešičkov. Treba je poudariti, da meritve ne modificiramo v vsakem koraku, ampak le v tistih, kjer je izpolnjen kriterij o širjenju. Prvi korak tega postopka nam torej da oceno o tem, ali naj meritev sploh modificiramo ali ne. Kriterij smo postavili na osnovi primerjave med najboljšim približkom objekta in meritve v prejšnji, i-ti sliki. Primerjavo opravimo z dvema preprostima značilkama: s ploščino in povprečno sivino v posamezni regiji (meritvi in najboljšem približku objekta). Kriterij je preprost: če je ploščina meritve večja od ploščine najboljšega približka in je hkrati povprečna sivina v meritvi večja od povprečne sivine znotraj najboljšega približka, potem je kriterij izpolnjen – modificiranje je možno. Kriterij lahko zapišemo tudi čisto formalno:

$$\eta = \begin{cases} 1, & \left(\frac{p(z_i)}{p(k_i)} > (1-\epsilon_p)\right) \quad \wedge \quad \left(m(z_i) > (m(k_i) - \epsilon_m)\right) \\ 0, & \text{sicer} \end{cases}, \tag{4.45}$$

kjer je s p označena ploščina za posamezno regijo, m pa označuje povprečno sivino v regiji. Parametra ϵ_p in ϵ_m sta majhni pozitivni števili (lahko tudi 0), s katerima modeliramo še dovoljena odstopanja v vrednostih. Iz enačbe (4.45) vidimo, da pri tem odločanju potrebujemo tudi zaporedje slik sivin.

Ideja, ki smo jo uporabili v opisanem kriteriju, je izredno preprosta. Lahko jo primerjamo z balonom. Balon najprej napihnemo, zatem pa ga pustimo, da se izprazni. Ko je spet prazen, ga ponovno napihnemo. Ta cikel nenehno ponavljamo. Če je najboljši približek objekta manjši od meritve, potem meritev napihnemo. Kalmanov filter bo zato začel vračati predikcije, ki bodo večje od originalne meritve – večji od meritve bo tudi najboljši približek. V naslednjih korakih kriterij za modificiranje meritve ne bo več izpolnjen, kljub temu pa bo predikcija najverjetneje še nekaj korakov večja od meritve. Ko najboljši približek objekta spet zleze v notranjost meritve, meritev ponovno modificiramo itd. Hitrost tega cikla lahko nadzorujemo s parametroma ϵ_p in ϵ_m.

Kriterij v enačbi (4.45) je določen izkustveno. Ob ploščini in povprečni sivini znotraj regije meritve in najboljšega približka objekta smo opazovali še druge značilke, in sicer: standardni odklon sivine znotraj regije, povprečno sivino in standardni odklon na meji regije, povprečni gradient in standardni odklon gradienta v regiji, povprečni gradient in standardni odklon gradienta na meji. Pri študiju teh vrednosti za nekaj naključno izbranih mešičkov smo opazili, da so za meritev, ki je večja od najboljšega približka mešička (določenega z algoritmom 5), večja tudi vsa izračunana povprečja, medtem ko za standardne odklone ta trditev ne velja vedno. Velja pa tudi obratno, in sicer v primeru, ko je najboljši približek večji od meritve. Vse izračunane vrednosti se seveda od mešička do mešička izredno razlikujejo. Tudi razlike med vrednostmi, izračunanimi za meritev in najboljši približek objekta, niso nikoli konstante. Iz tega sklepamo, da pragovna operacija v takšnih primerih ne more biti uspešna. Tudi ko je meritev manjša od pravega mešička, ne moremo iz samih povprečnih vrednosti (sivin in gradienta) točno določiti, za koliko je treba meritev povečati. Iz tega lahko vidimo, da je celoten korak popravljanja meritve bolj ali manj hevrističen – v posameznih delih ni eksakten.

Z indeksom η, določenim po enačbi (4.45), dobimo globalno oceno o tem, ali je treba meritev popraviti ali ne. Kot smo že zapisali, modificiramo meritev z_{i+1} v trenutni sliki na osnovi izračunane razdalje $e(z_i, k_i)$ iz prejšnje slike. Trivialen način modificiranja je, da vsako točko meritve premaknemo navzven za izračunano povprečno razdaljo $\|e(z_i, k_i)\|$.

To bi pomenilo, da vsako točko modificiramo na enak način. Takšen pristop predvideva, da je vsaka točka meritve enako oddaljena od pravega položaja mešička, kar pa za naš primer ne drži. Pokazali smo že, da je na posameznih delih mešiček prav dobro določen, spet na drugih delih pa slabo. To pomeni, da lahko da povprečna razdalja $\|e(z_i, k_i)\|$ med krivuljama le globalno oceno o velikosti širjenja, točno vrednost premika pa je treba določiti iz lokalnih lastnosti točke.

Naj bo $\mathbf{z}_{l,i+1}$ točka iz meritve z_{i+1}, ki jo modificiramo. Za koliko bomo posamezno točko premaknili, pa je odvisno od tega, kje se točka nahaja. Če je točka že blizu pravemu položaju mešička, jo premaknemo čisto malo, če pa je daleč stran, potem jo premaknemo močno. Odločitev o tem sprejmemo po primerjavi sivine in gradienta točke $\mathbf{z}_{l,i+1}$ s povprečno sivino in povprečnim gradientom v z_{i+1}. Možne so štiri variante:

1. Če je sivina točke $\mathbf{z}_{l,i+1}$ približno enaka povprečni sivini $m(z_{i+1})$ v regiji z_{i+1} in je hkrati gradient $\nabla(\mathbf{z}_{l,i+1})$ v točki $\mathbf{z}_{l,i+1}$ približno enak povprečnemu gradientu $m(\nabla(z_{i+1}))$ v z_{i+1}, potem lahko z veliko gotovostjo trdimo, da je točka $\mathbf{z}_{l,i+1}$ močno znotraj meje pravega položaja objekta. Širimo močno.

2. Če je sivina točke $\mathbf{z}_{l,i+1}$ veliko večja od povprečne sivine $m(z_{i+1})$ in je hkrati gradient $\nabla(\mathbf{z}_{l,i+1})$ veliko večji od povprečnega gradienta $m(\nabla(z_{i+1}))$, potem lahko z veliko gotovostjo trdimo, da je točka $\mathbf{z}_{l,i+1}$ že na meji pravega položaja objekta. Ne širimo.

3. Če je sivina točke $\mathbf{z}_{l,i+1}$ približno enaka povprečni sivini $m(z_{i+1})$ in je hkrati gradient $\nabla(\mathbf{z}_{l,i+1})$ veliko večji od povprečnega gradienta $m(\nabla(z_{i+1}))$, potem lahko z veliko gotovostjo trdimo, da je točka $\mathbf{z}_{l,i+1}$ že ob meji pravega položaja objekta. Širimo malo.

4. Če je sivina točke $\mathbf{z}_{l,i+1}$ veliko večja od povprečne sivine $m(z_{i+1})$ in je hkrati gradient $\nabla(\mathbf{z}_{l,i+1})$ približno enak povprečnemu gradientu $m(\nabla(z_{i+1}))$, potem lahko z veliko gotovostjo trdimo, da je točka $\mathbf{z}_{l,i+1}$ zbežala preko meje pravega položaja objekta. Ne širimo.

Naj še enkrat opomnimo, da Kalmanov filter vrača rezultate med meritvijo in predikcijo. Če torej želimo, da se tudi izračunani najboljši približek k približa pravemu položaju objekta, moramo meritev v variantah 1-3 še dodatno povečati.

Točke od 1 do 4 so zgolj opisna pravila, ki določajo, kako naj postopamo v posameznih primerih. Ta pravila je treba pretvoriti v čisto formalen zapis, ki ga je potem možno vključiti v algoritem. Tukaj si pomagamo z znanjem o algoritmu za statične 2D slike, s katerim so bile dobljene meritve. Meritve mešičkov so bile dobljene z algoritmom, opisanim v podpoglavju 3.3. Segmentacijska metoda, uporabljena v tem algoritmu, je rast regij. Rast regij nadzira na osnovi kriterija sivine (enačba (3.6)) in kriterija gradienta (enačba (3.8)). Oba kriterija bomo privzeli tudi pri modificiranju meritve. Kriterij sivine ct_1 tvorimo kot

$$ct_1 = \frac{|I_{i+1}(\mathbf{z}_{l,i+1}) - m(z_{i+1})|}{\sigma(z_{i+1})}, \tag{4.46}$$

kjer I_{i+1} označuje $(i+1)$-to sliko sivin. Kriterij gradienta ct_2 pa ima naslednjo obliko:

$$ct_2 = \frac{|\nabla(\mathbf{z}_{l,i+1}) - m(\nabla(z_{i+1}))|}{\sigma(\nabla(z_{i+1}))}. \tag{4.47}$$

Če sta kriterija večja od 2, potem rečemo, da je vrednost točke veliko večja od povprečnih vrednosti v regiji. Če sta pa vrednosti kriterijev blizu 0, potem rečemo, da so si vrednosti podobne. Prag 2 smo privzeli kar iz algoritma za statične 2D slike – predpostavka o Gaussovi porazdelitvi sivin pikslov in izbira intervala zaupanja 95 %.

Iz obeh kriterijev tako tvorimo skupno oceno o velikosti širjenja meritve ct, in sicer:

$$ct = \begin{cases} 0, & ct_1 \geq 2 \quad \vee \quad ct_2 \geq 2 \\ (2 - ct_1)(2 - ct_2), & \text{sicer.} \end{cases} \tag{4.48}$$

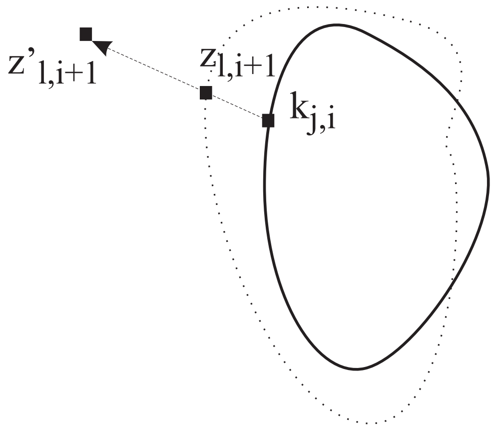
Na sliki 4.7 je prikazan postopek modificiranja meritve z_{i+1}. Naj bo $\mathbf{k}_{j,i}$ točka s krivulje k_i najboljšega približka objekta iz koraka i. Njena pripadajoča meritev v trenutni sliki pa je $\mathbf{z}_{l,i+1}$. To meritev popravljamo. Modificirano vrednost meritve $\mathbf{z}'_{l,i+1}$ določimo tako, da k vrednosti meritve $\mathbf{z}_{l,i+1}$ prištejemo ustrezen večkratnik normiranega vektorja, ki kaže v smeri od točke $\mathbf{k}_{j,i}$ do $\mathbf{z}_{l,i+1}$. Predznak tega vektorja vedno določimo tako, da se meritev po modificiranju razširi (za primer s slike 4.7 bi bil predznak tega vektorja pozitiven). Formalno to modificiranje zapišemo kot:

$$\mathbf{z}'_{l,i+1} = \mathbf{z}_{l,i+1} + predznak \;\cdot\; \nu \;\cdot\; \frac{\overrightarrow{\mathbf{k}_{j,i}\mathbf{z}_{l,i+1}}}{||\overrightarrow{\mathbf{k}_{j,i}\mathbf{z}_{l,i+1}}||}. \tag{4.49}$$

Parameter ν določa velikost premika. Določimo ga kot produkt parametra ct in razdalje $||e(z_i, k_i)||$:

$$\nu = ct \cdot ||e(z_i, k_i)||. \tag{4.50}$$

Takšen način določanja velikosti premika je seveda popolnoma hevrističen. S tem dejansko zagotovimo, da pri posameznih točkah meritev pade preko prave meje objekta, spet pri drugih pa meje ne doseže. Rezultat tega je, da meritev sedaj enakomerno poteka malo zunaj prave meje objekta, malo pa znotraj. To seveda velja le v primerih, ko meritev že v osnovi ni bila popolnoma napačna. Določanje parametra ν na takšen način ima tudi svoje omejitve. To se predvsem kaže v meritvah, ki so močno napačne (podpoglavje 4.2.2). V teh primerih postane razdalja $||e(z_i, k_i)||$ izredno velika, s tem pa tudi sam premik meritve. Problem zajezimo tako, da to razdaljo vgradimo v Kalmanov filter.



Slika 4.7: Modificiranje meritve. S puščico je označena smer širjenja.

Problem velikih napak v meritvah smo obravnavali s Kalmanovim filtrom J (podpoglavje 4.2.2). Prav zaradi velikih napak v meritvah tudi vgrajujemo povprečno razdaljo med krivuljama v Kalmanov filter. Zato bomo uporabili kar Kalmanov filter J, ki ga bomo ustrezno dogradili. Povprečno razdaljo med krivuljama modeliramo kot razdaljo d v prejšnjem koraku, ki ji prištejemo spremembo razdalje $\triangle d$. Stanje sistema $\mathbf{x}_i^J$ za KF J lahko tako iz enačbe (4.27) dogradimo v:

$$\mathbf{x}_i^J = \begin{bmatrix} p_i^J \\ \triangle p_i^J \\ \kappa_i^J \\ \dots \\ d_i^J \\ \triangle d_i^J \end{bmatrix}. \tag{4.51}$$

Začetno stanje sistema $\hat{\mathbf{x}}_0^J$ pa iz (4.29) dopolnimo v

$$\hat{\mathbf{x}}_0^J = \begin{bmatrix} p(z_0) \\ 0 \\ \kappa(z_0) \\ \dots \\ 0 \\ 0 \end{bmatrix}. \tag{4.52}$$

V vsakem koraku lahko izmerimo razdaljo med krivuljama k_{i-1} in z_{i-1}. Zato se spremeni tudi meritev $\mathbf{z}_i^J$ iz (4.30) v

$$\mathbf{z}_i^J = \begin{bmatrix} z_{p_i^J} \\ z_{\kappa_i^J} \\ \dots \\ z_{d_i^J} \end{bmatrix} = \begin{bmatrix} p(z_i) \\ \kappa(z_i) \\ \dots \\ d(k_{i-1}, z_{i-1}) \end{bmatrix}. \tag{4.53}$$

Zapišemo lahko tudi novi matriki $\mathbf{\Phi}^J$ in $\mathbf{H}^J$, in sicer:

$$\mathbf{\Phi}^J = \begin{bmatrix} 1 & 1 & 0 & \vdots & 0 & 0 \\ 0 & 1 & 0 & \vdots & 0 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 0 \\ \dots & \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & 0 & \vdots & 1 & 1 \\ 0 & 0 & 0 & \vdots & 0 & 1 \end{bmatrix}, \mathbf{H}^J = \begin{bmatrix} 1 & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 0 \\ \dots & \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & 0 & \vdots & 1 & 0 \end{bmatrix}. \quad (4.54)$$

Začetno kovariančno matriko $\mathbf{P}_0^{J-}$ in šum v sistemu Q_0^J tvorimo na enak način kot pri osnovni varianti Kalmanovega filtra J. Novi matriki sta torej oblike:

$$\mathbf{P}_0^{J-} = \begin{bmatrix} 1000 & 1000 & 0 & \vdots & 0 & 0 \\ 1000 & 1000 & 0 & \vdots & 0 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 0 \\ \dots & \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & 0 & \vdots & 1000 & 1000 \\ 0 & 0 & 0 & \vdots & 1000 & 1000 \end{bmatrix}, \mathbf{Q}_0^J = \begin{bmatrix} 1 & 0 & 0 & \vdots & 0 & 0 \\ 0 & 1 & 0 & \vdots & 0 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 0 \\ \dots & \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & 0 & \vdots & 0 & 1 \end{bmatrix}. \quad (4.55)$$

Podobno določimo tudi šum v merilnem sistemu $\mathbf{R}_0^J$ v koraku 0:

$$\mathbf{R}_0^J = \begin{bmatrix} 1 & 0 & \vdots & 0 \\ 0 & 1 & \vdots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \vdots & 1 \end{bmatrix}. \quad (4.56)$$

V osnovni varianti KF J se je ta matrika šuma v merilnem sistemu zatem na osnovi kvalitete meritev spreminjala od slike do slike (enačba (4.39)). Podobno ravnamo tudi v primeru razdalje. Nova dopolnjena matrika šuma v merilnem sistemu ima tako obliko:

$$\mathbf{R}_i^J = \begin{bmatrix} f_{plos} & 0 & \vdots & 0 \\ 0 & f_{komp} & \vdots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \vdots & f_{plos_{i-1}} \cdot f_{komp_{i-1}} \end{bmatrix}. \quad (4.57)$$

Pri modificiranju meritve dejansko upoštevamo razdaljo med meritvijo in najboljšim približkom objekta s prejšnje slike, tj. $(i-1)$-te slike. Zato moramo tudi pri modeliranju šuma v izmeri razdalje med krivuljama uporabiti kazni za ploščino in kompaktnost, izračunani na prejšnji sliki. S tem dejansko izničimo vpliv velikih napak v meritvah.

Parameter η določa, ali bomo meritev v posameznem koraku res modificirali (enačba (4.45)). To pomeni, da ima vrednost razdalje smisel le v tistih korakih, kjer je $\eta = 1$, v vseh ostalih pa ne. Pri teh korakih je tudi smiselno ažurirati stanja KF J pri razdalji. V vseh ostalih korakih pa sami postavimo stanje sistema KF J pri razdalji d in spremembi razdalje $\triangle d$ na vrednost 0.

Pri modificiranju meritve z_{i+1} tako sedaj ne upoštevamo le razdalje med krivuljama z_i in k_i, ampak uporabimo najboljšo oceno za razdaljo iz KF J. Tako lahko zapišemo končno formulo za določanje premika ν:

$$\nu = ct \cdot \hat{d}^J_{i+1}. \tag{4.58}$$

Opisali smo postopek, kako dobimo modificirano meritev. Nova meritev je dejansko stara meritev, ki ji prištejemo izračunan odmik (enačba (4.49)). To lahko formalno zapišemo:

$$\mathbf{z}'_{l,i+1} = \mathbf{z}_{l,i+1} + \triangle \mathbf{z}_{l,i+1}. \tag{4.59}$$

Takšno modificirano meritev lahko uporabimo za pravo meritev v algoritmu 5. Veliko elegantneje pa je seveda, če vgradimo ta model za meritev v sam globalni Kalmanov filter. Da bi dosegli to, moramo razširiti stanje sistema globalnega Kalmanovega filtra, in sicer ga razširimo za oba odmika pri meritvi, $\triangle z_{x_{j,i}}$ in $\triangle z_{y_{j,i}}$. Stanje sistema $\mathbf{x}_{j,i}$ za globalni Kalmanov filter je tako sedaj:

$$\mathbf{x}_{j,i} = \begin{bmatrix} x_{j,i} \\ y_{j,i} \\ \triangle x_{j,i} \\ \triangle y_{j,i} \\ \cdots \\ \triangle z_{x_{j,i}} \\ \triangle z_{y_{j,i}} \end{bmatrix}. \tag{4.60}$$

S tem se spremeni tudi matrika $\mathbf{\Phi}$, in sicer iz enačbe (4.16) v:

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & 1 & 0 & \vdots & 0 & 0 \\ 0 & 1 & 0 & 1 & \vdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \vdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 & 1 \end{bmatrix}. \tag{4.61}$$

Vidimo, da smo oba odmika pri meritvi modelirali na enak način kot odmika pri posamezni koordinati. Oba odmika pri meritvi lahko ocenimo po enačbi (4.49). Zato rečemo, da lahko ti dve vrednosti izmerimo. Ti novi meritvi lahko dodamo h meritvi globalnega

Kalmanovega filtra. Nova meritev $\mathbf{z}_{j,i}$ ima tako obliko:

$$\mathbf{z}_{j,i} = \begin{bmatrix} z_{x_{j,i}} \\ z_{y_{j,i}} \\ \dots \\ z_{\triangle z_{x_{j,i}}} \\ z_{\triangle z_{y_{j,i}}} \end{bmatrix}. \tag{4.62}$$

Iz enačbe (4.59) vidimo, v kakšni povezavi sta stara meritev in odmik meritve. Na osnovi enačb (4.18) in (4.59) lahko ustrezno preuredimo matriko $\mathbf{H}$, ki povezuje meritve s sistemom. Matrika $\mathbf{H}$ je tako:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & \vdots & -1 & 0 \\ 0 & 1 & 0 & 0 & \vdots & 0 & -1 \\ \dots & \dots & \dots & \dots & .\vdots. & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 & 1 \end{bmatrix}. \tag{4.63}$$

V začetno stanje sistema dodamo vrednosti 0 na mestih odmika meritve. Stanje $\hat{\mathbf{x}}_{j,0}$ je tako:

$$\hat{\mathbf{x}}_{j,0} = \begin{bmatrix} k_{x_{j,0}} \\ k_{y_{j,0}} \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{bmatrix}. \tag{4.64}$$

Šum v sistemu in meritvah je modeliran na popolnoma enak način kot pri globalnem Kalmanovem filtru z manj stanji. Ustrezno se spremi le dimenzija obeh matrik. Matriki $\mathbf{Q}$ in $\mathbf{R}$ sta oblike:

$$\mathbf{Q}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & \vdots & 0 & 0 \\ 0 & 1 & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \vdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \vdots & 0 & 0 \\ \dots & \dots & \dots & \dots & .\vdots. & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 & 1 \end{bmatrix} \text{ in } \mathbf{R}_0 = \begin{bmatrix} 1 & 0 & \vdots & 0 & 0 \\ 0 & 1 & \vdots & 0 & 0 \\ \dots & \dots & .\vdots. & \dots & \dots \\ 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & \vdots & 0 & 1 \end{bmatrix}. \tag{4.65}$$

Na tem mestu moramo še poudariti, da se matriki za šum seveda tudi v tem primeru med samim delovanjem algoritma ustrezno modificirata glede na kvaliteto meritev (enačba (4.38)) in lokalnost krivulje (enačbi (4.43) in (4.44)).

Enako smo postopali tudi pri kovariančni matriki napake $\mathbf{P}_0^-$, kjer smo povečali dimenzijo matrike ter ustrezno zapolnili nova polja:

$$\mathbf{P}_0^- = \begin{bmatrix} 1000 & 1000 & 1000 & 1000 & \vdots & 0 & 0 \\ 1000 & 1000 & 1000 & 1000 & \vdots & 0 & 0 \\ 1000 & 1000 & 1000 & 1000 & \vdots & 0 & 0 \\ 1000 & 1000 & 1000 & 1000 & \vdots & 0 & 0 \\ \dots & \dots & \dots & \dots & \cdot\vdots\cdot & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & 1000 & 1000 \\ 0 & 0 & 0 & 0 & \vdots & 1000 & 1000 \end{bmatrix}. \tag{4.66}$$

Sedaj lahko zapišemo končno obliko algoritma za razpoznavanje objektov iz zaporedja slik s postopki predikcije, kjer je vključen mehanizem za modificiranje meritev. Dodani koraki so uokvirjeni.

$\boxed{\text{VHOD:}}$ Zaporedje segmentiranih in označenih slik $\mathbf{S}$. *Zaporedje slik sivin* $\mathbf{I}$. *Oznaka objekta, ki ga razpoznavam, je O.*

IZHOD: Zaporedje najboljših približkov k_i objekta O na posamezni sliki S_i.

1. *i=0*
2. *Poišči prvo sliko iz zaporedja* $\mathbf{S}$, *kjer se pojavi objekt z oznako O. Označi to sliko s S_0.*
3. *Določi notranjo mejo objekta O. To je najboljši približek objekta O v koraku 0, popisan s krivuljo k_0.*
4. $\boxed{4.}$ *Vsaki točki $\mathbf{k}_{j,0}$ s krivulje k_0:*
   a. *po enačbi (4.64) določi začetno stanje sistema $\hat{\mathbf{x}}_{j,0}$,*
   b. *po enačbi (4.66) določi matriko $\mathbf{P}_{j,1}^-$.*
5. $\boxed{5.}$ *Določi začetno stanje sistema $\hat{\mathbf{x}}_0^J$ Kalmanovega filtra J po enačbi (4.52) ter matriko $\mathbf{P}_1^{J-}$ po enačbi (4.55).*
6. *ZANKA: i=i+1*
7. *Na osnovi pogoja ujemanja v (4.13) poišči v sliki S_i objekt O.*
8. *Če objekta O nismo našli, potem KONEC, sicer nadaljuj.*
9. *Določi notranjo mejo objekta O. Označi dano krivuljo kot meritev z_i.*
10. $\boxed{10.}$ *Meritev $\mathbf{z}_i^J$ za krivuljo z_i določi po enačbi (4.53). Če pogoj za širjenje meritve (enačba (4.45))* **ni** *izpolnjen, potem postavi stanje $\hat{\mathbf{x}}_i^J$ in meritev $\mathbf{z}_i^J$ pri razdalji d ter odmiku $\triangle d$ na 0.*

11. *Ažuriraj matriko šuma* $\mathbf{R}_i$ *za globalni Kalmanov filter po enačbi (4.38).*
12. *Določi* $\mathbf{\Phi}^J$, $\mathbf{H}^J$ *in* $\mathbf{Q}_i^J$ *po enačbah (4.54) in (4.55).*
13. *Ažuriraj matriko šuma* $\mathbf{R}_i^J$ *za Kalmanov filter* J *po enačbi (4.57).*
14. $[\hat{\mathbf{x}}_i^J, \mathbf{P}_{i+1}^{J-}] =$ *KalmanovFilter*$(\hat{\mathbf{x}}_{i-1}^J,\ \mathbf{z}_i^J,\ \mathbf{P}_i^{J-},\ \mathbf{Q}_i^J,\ \mathbf{R}_i^J, \mathbf{\Phi}^J, \mathbf{H}^J)$
15. *Določi začetno stanje* $\hat{\mathbf{x}}_0^{LOK}$ *Kalmanovega filtra* LOK *po enačbi (4.41) in kovariančno matriko* $\mathbf{P}_0^{LOK-}$ *po enačbi (4.40).*
16. *Za vsako točko* $\mathbf{k}_{j,i-1}$ *s krivulje* k_{i-1}*:*

    a. *po enačbi (4.23) določi pripadajočo točko* $\mathbf{z}_{l,i}$ *s krivulje meritve,*

    b. *določi meritev* $\mathbf{z}_j^{LOK} = [k_{x_{j,i-1}}, k_{y_{j,i-1}}]^T$,

    c. $[\hat{\mathbf{x}}_j^{LOK}, \mathbf{P}_{j+1}^{LOK-}] =$ *KalmanovFilter*$(\hat{\mathbf{x}}_{j-1}^{LOK},\ \mathbf{z}_j^{LOK},\ \mathbf{P}_j^{LOK-},\ \mathbf{Q}_i^{LOK},\ \mathbf{R}_i^{LOK}, \mathbf{\Phi}^{LOK}, \mathbf{H}^{LOK})$,

    d. *po enačbah (4.65) določi matriko* $\mathbf{Q}_i$ *ter po enačbah (4.61) in (4.63) matriki* $\mathbf{\Phi}$ *in* $\mathbf{H}$,

    e. *po enačbah (4.43) ter (4.44) ažuriraj matriki* $\mathbf{Q}_{j,i}$ *in* $\mathbf{R}_{j,i}$,

    f. *dopolni meritev* $\mathbf{z}_{l,i}$ *na osnovi enačbe (4.62). V nova dodana polja vpiši vrednost 0. Če je izpolnjen pogoj za širjenje meritve (enačba (4.45)), potem določi ustrezen odmik meritve po enačbah (4.49) in (4.58),*

    g. $[\hat{\mathbf{x}}_{j,i}, \mathbf{P}_{j,i+1}^-] =$ *KalmanovFilter*$(\hat{\mathbf{x}}_{j,i-1},\ \mathbf{z}_{l,i},\ \mathbf{P}_{j,i}^-,\ \mathbf{Q}_{j,i},\ \mathbf{R}_{j,i}, \mathbf{\Phi}, \mathbf{H})$.

17. *Iz stanj sistema* $\hat{\mathbf{x}}_{j,i}$ *tvori najboljši približek objekta* k_i.
18. *Za vsako točko* $\mathbf{k}_{j,i}$ *s krivulje* k_i*:*

    a. *ažuriraj stanje sistema* $\mathbf{x}'_{j,i}$ *in* $\mathbf{P}'^{-}_{j,i+1}$ *po enačbi (4.26).*

19. *Priredi* $\hat{\mathbf{x}}_{:,i} = \mathbf{x}'_{:,i}$ *in* $\mathbf{P}^-_{:,i+1} = \mathbf{P}'^{-}_{:,i+1}$. *Vrni se na vrstico z oznako ZANKA.*

Algoritem 6 Algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Dodan korak popravljanja meritev (uokvirjeno).

4.2.5 Končne izboljšave algoritma

V algoritmu 6 smo strnili končno verzijo algoritma za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Jedro tega algoritma je globalni Kalmanov filter, ki sledi vsaki točki krivulje skozi zaporedje. Ker nimamo nikakršnega vnaprejšnjega znanja o kvaliteti meritev in o natančnosti ocene za začetno stanje Kalmanovega filtra, smo določili matrike $\mathbf{P}$, $\mathbf{Q}$ in $\mathbf{R}$ na takšen način, da smo omogočili filtru, da se v določenem času sam nauči iz meritev ustreznih modelnih parametrov. Učenje pa seveda zahteva svoj čas, torej določeno število korakov. To pa dejansko pomeni, da se na začetku zaporedja pojavi t. i. prehodni pojav, kjer modelni parametri niso točni in s tem tudi rezultati ne nakazujejo najboljšega položaja objekta. Prehodni pojav lahko omilimo tako, da npr. v nekaj začetnih korakih kaznujemo meritve prizanesljiveje kot v kasnejših (ustrezno modificiramo enačbo (4.38) za zgodnje korake). Še bolje pa je, če zaporedje slik večkrat zaporedoma obdelamo, najprej z ene smeri, zatem pa še z druge. V veliki večini že odpravimo prehodni pojav, če zaporedje slik obdelamo v dveh prehodih. Zaporedje slik obdelamo z algoritmom 6 najprej v smeri naraščajočih indeksov slik. Ko pridemo do konca zaporedja, ohranimo vse izračunane parametre – matrike $\mathbf{P}$ in stanja sistema $\hat{\mathbf{x}}_{j,i}$ (ustrezno spremenimo le predznak pri odmikih) ter ponovno obdelamo zaporedje slik, sedaj pa v smeri padajočih indeksov slik. Za meritve vzamemo sedaj rezultate – najboljše približke objektov – dobljene po prvem prehodu. Obdelovanje v obe smeri lahko seveda ponavljamo v neskončnost. Moramo pa se zavedati, da po vsakem prehodu postajajo rezultati bolj in bolj kompaktni.

Algoritem 6 je zasnovan za razpoznavanje enega samega objekta v zaporedju slik. Če želimo razpoznavati več objektov hkrati, v začetni sliki imamo npr. 5 različnih objektov, ki jih želimo razpoznavati skozi zaporedje, potem moramo pognati algoritem 6 za vsak objekt posebej. Celoten postopek lahko modificiramo na naslednji način. Naj vhod v naš postopek predstavlja segmentirana slika S_k z množico objektov, ki jih želimo razpoznati. Potem moramo iz slike S_k najprej izločiti vse možne objekte O. Vsak objekt pa zatem obdelamo z algoritmom 6 v dveh prehodih.

Vhod v modificiran algoritem je torej segmentirana slika S_k z množico objektov. To sliko dobimo z enim od algoritmov za statične 2D slike. Možno je tudi, da je kateri izmed objektov iz celotne množice objektov napačen. To pomeni, da je prištet k objektom, čeprav je morebiti zgolj regija, ki je nastala po segmentiranju slike zaradi šuma ipd. Šumna regija se bo v zaporedju pojavila le na nekaj zaporednih slikah, potem pa bo izginila. Takšne regije lahko preprosto izločimo. Predpostavimo, da je regija veljavna le, če se pojavi vsaj v ϑ zaporednih slikah. V tem primeru jo obdržimo, sicer pa ta objekt izločimo.

Podajmo končno obliko algoritma za razpoznavanje vseh objektov iz zaporedja slik s postopki predikcije.

VHOD: Segmentirana slika S_k z množico objektov. Zaporedje označenih slik $\mathbf{S}$. Zaporedje slik sivin $\mathbf{I}$. Parameter ϑ, ki določa minimalno število slik, na katerih se mora pojaviti posamezen objekt.

IZHOD: Zaporedje najboljših približkov k_i za vsak objekt O_p na posamezni sliki S_i.

1. *Določi množico vseh objektov $O = \{O_1, O_2, \ldots, O_p\}$ iz slike S_k.*
2. *Za vsak objekt $O_p \in O$:*
   a. *Določi začetno in končno sliko iz zaporedja* $\mathbf{S}$, *kjer se pojavi O_p.*
   b. *Če se objekt O_p pojavi na več kot ϑ slikah, potem:*
      - *prvi prehod: razpoznavaj O_p z algoritmom 6 od začetne proti končni sliki;*
      - *shrani najboljše končne rezultate k_i' in vse izračunane* $\mathbf{P}$ *ter* $\hat{\mathbf{x}}$*. Ustrezno jih preuredi za drugi prehod;*
      - *drugi prehod: razpoznavaj O_p z algoritmom 6 od končne proti začetni sliki. Uporabi shranjene nastavitve. Za meritve vzemi končne položaje k_i' iz prvega prehoda;*
      - *za objekt O_p shrani najboljše približke k_i.*

Algoritem 7 Končna oblika algoritma za razpoznavanje objektov iz zaporedja slik s postopki predikcije.

4.3 Zaključek

V tem poglavju smo opisali nov algoritem za razpoznavanje objektov s postopki predikcije. Bistven del algoritma je Kalmanov filter. Ob osnovni varianti algoritma (algoritem 3), smo vpeljali še tri izboljšave: postopek, ki eliminira velike napake v meritvah (algoritem 4), postopek, ki upošteva lokalnost (algoritem 5), in postopek, ki modificira meritve, tako da ustrezajo pogoju Gaussove porazdelitve (algoritem 6). Končna oblika algoritma za razpoznavanje več objektov z vsemi vpeljanimi izboljšavami pa je podana v algoritmu 7.

Čeprav so posamezni deli algoritma prirejeni za razpoznavanje jajčnih mešičkov, je ta algoritem popolnoma splošen. Nanj je treba gledati kot na ogrodje, ki ga lahko dopolnimo z ustreznimi rutinami, ki so odvisne od tipa reševane naloge. Tu mislimo predvsem na korake, opisane v podpoglavjih 4.2.2 in 4.2.4.

5.

Pregled rezultatov in razprava

V tem poglavju bomo predstavili rezultate, dobljene z algoritmom za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Nov algoritem, opisan v 4. poglavju, smo preizkusili tako na umetnih ultrazvočnih slikah jajčnikov z mešički kakor tudi na realnih posnetkih. V obeh primerih pa predstavljajo vhod v algoritem odčitki, ki smo jih dobili po uporabi algoritma za razpoznavanje v statičnih 2D slikah (poglavje 3).

Dobljene rezultate bomo opisali v dveh podpoglavjih, in sicer v podpoglavju 5.1, kjer bomo prikazali in statistično ovrednotili rezultate na umetnih slikah, ter v podpoglavju 5.2, kjer bomo prikazali tipične primere razpoznavanja na realnih slikah. Pri tem bomo vseskozi primerjali rezultate, dobljene z algoritmom za statične 2D slike, ter rezultate algoritma na osnovi postopkov predikcije. Dobljene rezultate bomo tudi kritično ovrednotili ter poudarili določene omejitve posameznih pristopov.

5.1 Rezultati na umetnih slikah

V 3. poglavju smo zapisali, da je eden izmed najtežjih problemov pri obdelovanju medicinskih slik verifikacija in validacija postopkov oz. algoritmov za obdelovanje takšnih slik. Dobro je tudi znano, da v večini primerov predstavljajo odčitki ekspertov referenčne rezultate, na osnovi katerih vrednotimo uspešnost računalniških postopkov. Odčitki ekspertov pa seveda niso in ne morejo biti zlati standard [CK97]. Zato je smiselno algoritme najprej preveriti na umetnih slikah, kjer so vse razmere vnaprej točno znane, šele nato pa na realnih posnetkih.

Tak pristop smo vseskozi uporabljali v našem delu. Za testne slike smo uporabili umetne ultrazvočne slike jajčnikov z mešički (podpoglavje 3.2). Algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije (algoritem 7) pričakuje na vhodu ob zaporedju slik sivin še zaporedje že segmentiranih slik. Segmentirane slike pa smo dobili kot rezultat obdelovanja z algoritmom za statične 2D slike (algoritem 2).

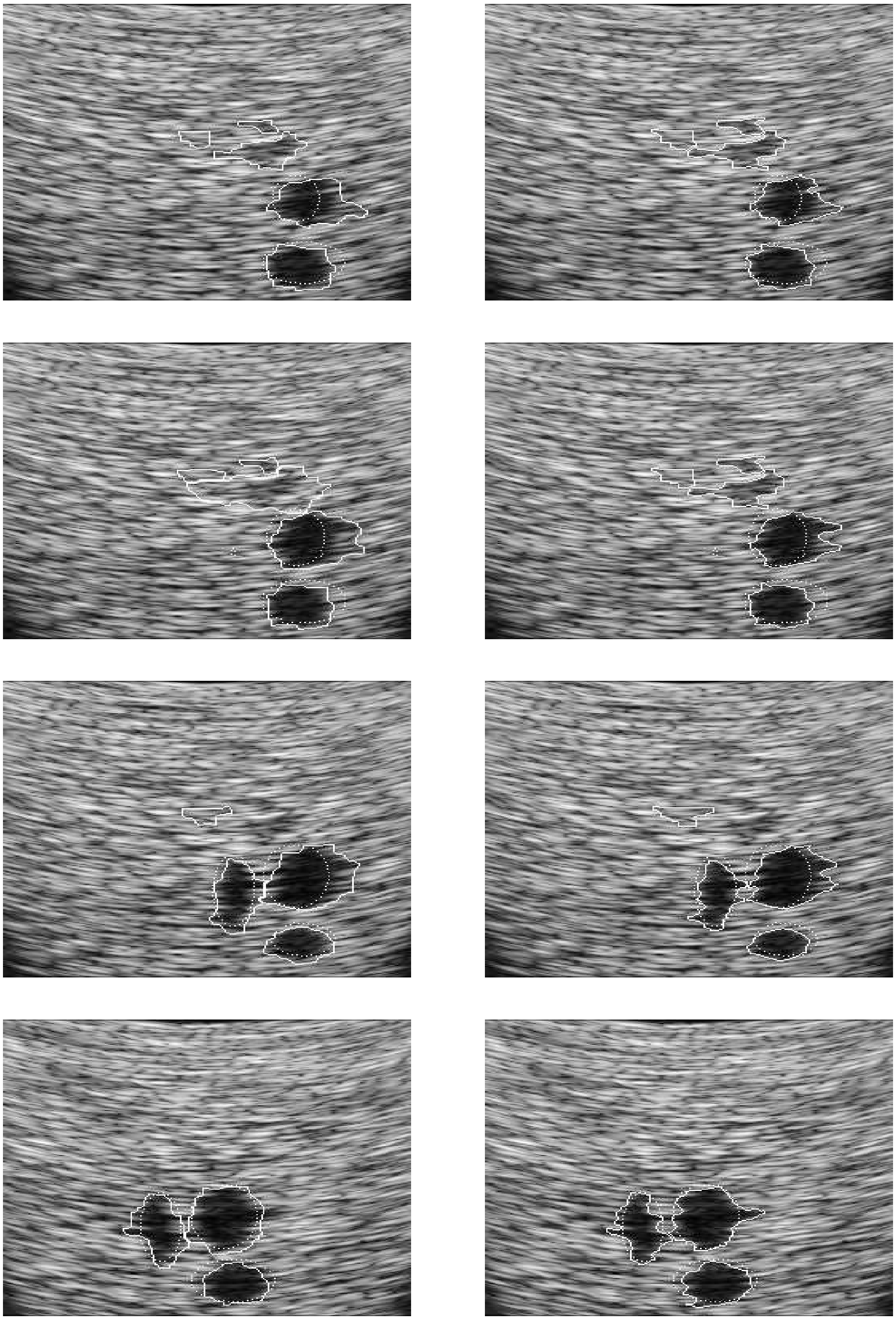
Algoritem 7 smo preverili na dveh različnih zaporedjih, pri čemer je bilo v vsakem zaporedju 60 umetnih ultrazvočnih slik jajčnikov. Prvo zaporedje $\mathcal{A}$ je izsek iz celotnega zaporedja slik, ki smo jih dobili po pregledovanju modela jajčnika z nerotirano sondo. Drugo zaporedje $\mathcal{B}$ je prav tako izsek 60 slik iz celotnega zaporedja slik, ki smo jih dobili po pregledovanju istega modela jajčnika, vendar v tem primeru z rotirano sondo, ob tem pa je bilo dovoljeno še tresenje roke (podpoglavje 3.2). Pokažimo najprej nekaj dobljenih rezultatov.

Na sliki 5.1 je prikazanih nekaj tipičnih rezultatov. Leva slika vsakega para prikazuje mešičke, kot jih je razpoznal algoritem 7 s postopki predikcije, na desni sliki pa so z belo označeni razpoznani mešički z algoritmom za statične 2D slike (algoritem 2). Elipse, ki označujejo pravilni položaj mešičkov, so na obeh slikah označene pikčasto. Prvi trije rezultati so iz zaporedja $\mathcal{A}$, in sicer za 1., 11. in 41. sliko, četrti rezultat pa je za 21. sliko iz zaporedja $\mathcal{B}$. Treba je še poudariti, da smo zaradi lažje vizualne primerjave prikazali kar vse regije. Postavili torej nismo nobene omejitve glede minimalnega števila zaporednih slik, na katerih se mora pojaviti regija, da jo štejemo kot veljavno (glej vhod v algoritem 7).

Če opravimo zgolj vizualno primerjavo med rezultati algoritma na statičnih 2D slikah (algoritem 2) – na sliki 5.1 na desni – in rezultati algoritma za razpoznavanje iz zaporedja slik s postopki predikcije (algoritem 7) – na sliki 5.1 na levi – opazimo naslednje. Vidimo, da so rezultati na osnovi postopkov predikcije veliko bolj kompaktni kakor rezultati, dobljeni le na eni statični 2D sliki. Prav tako lahko opazimo, da se razpoznani mešički z algoritmom 7 dejansko raztezajo prav do robov temnega območja, na katerem se nahajajo. Z drugimi besedami to pomeni, da algoritem 7 razpoznava temne regije v slikah optimalno. Ob primerjanju teh rezultatov s pravilnimi položaji pa zapazimo velik razkorak. Pravilni položaji se namreč tudi v veliki meri raztezajo tudi po svetlih regijah oz. so prekriti s svetlejšim ali temnejšim šumom prav do robov temnih območij. Končajo se npr. v sredini temne regije, kjer ni možno zaznati nikakršnih robov. To je vsekakor značilnost umetnih slik oz. karakteristika simulatorja za tvorjenje umetnih ultrazvočnih slik jajčnikov, ki pa ima velik vpliv na kvaliteto razpoznanih mešičkov.

Opravimo še primerjavo med rezultati na osnovi vpeljanih kvantitativnih mer za merjenje učinkovitosti algoritmov. Uporabili bomo enake metrike in mere kot pri vrednotenju učinkovitosti algoritma za statične 2D slike (podpoglavje 3.4), in sicer odstotek pravilno in napačno razpoznanih, količnika r_1 in r_2 ter razdaljo med mejama razpoznanega in dejanskega mešička.

Podajmo najprej rezultate, kjer upoštevamo vse regije v sliki – ni omejitev glede minimalnega števila zaporednih slik, na katerih se mora regija pojaviti (parameter ϑ ni upoštevan). Ker so bili mešički na umetnih slikah precej dobro razpoznani že z algoritmom



Slika 5.1: Nekaj tipičnih rezultatov na umetnih slikah. Leva slika vsakega para prikazuje mešičke, kot jih je razpoznal algoritem 7, na desni sliki pa so z belo označeni razpoznani mešički z algoritmom 2. Elipse, ki označujejo pravilni položaj mešičkov, so na obeh slikah označene pikčasto.

2, sta odstotka pravilno in napačno razpoznanih mešičkov ostala v tem primeru enaka kot pri algoritmu za statične 2D slike, in sicer je odstotek pravilno razpoznanih okrog 90 % (standardni odklon je okrog 19 %), odstotek napačno razpoznanih pa je okrog 45 % (standardni odklon je okrog 27 %). Razlike pa se pojavijo pri količnikih r_1 in r_2. Količnik r_1 se je z 0,78 (standardni odklon 0,07) pri razpoznavanju v statičnih 2D slikah povečal na 0,83 pri razpoznavanju s postopki predikcije (standardni odklon je 0,09). Pri količniku r_2 pa smo izračunali majhno zmanjšanje, in sicer od 0,78 (standardni odklon 0,07) pri algoritmu za statične 2D slike na 0,70 pri algoritmu s postopki predikcije (standardni odklon 0,08). Hkrati s tem se je malo povečala tudi povprečna napaka oz. absolutna razdalja med mejo razpoznane regije in dejanskega mešička, in sicer od 4,10 piksla (standardni odklon 1,31) pri algoritmu 2, na 4,80 piksla (standardni odklon 1,10) pri algoritmu 7.

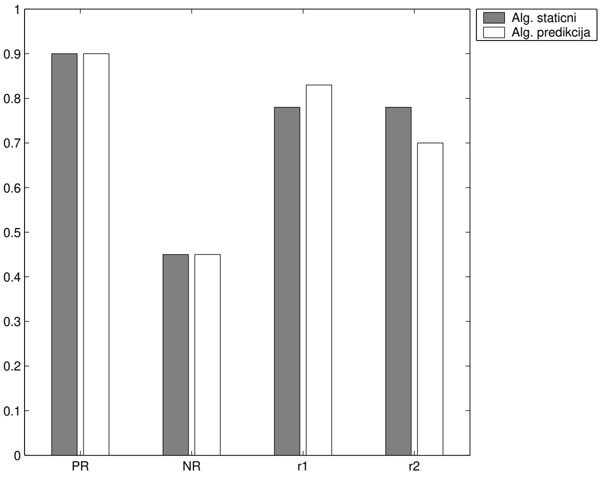
Merili smo tudi kompaktnost razpoznanih regij. Kompaktnost regije definiramo kot razmerje med kvadriranim obsegom regije in njeno ploščino [SHB94]. Povprečna kompaktnost razpoznanih regij, preden smo uporabili postopke predikcije, je bila okrog 19,00 (standardni odklon 3,70), po uporabi postopkov predikcije pa je bila okrog 15,90 (standardni odklon 2,00). Manjša vrednost pomeni, da je regija bolj kompaktna. Torej vidimo, da so po uporabi postopkov predikcije (algoritem 7) rezultati veliko bolj kompaktni.

V tabeli 5.1 so v strnjeni obliki zbrani vsi opisani rezultati za algoritem za razpoznavanje na statičnih 2D slikah (algoritem 2) in za algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije (algoritem 7).

	PR (v %)	NR (v %)	r_1	r_2	d (v pikslih)	κ
Alg. 2	90 ($\sigma = 19$)	45 ($\sigma = 27$)	0,78 ($\sigma = 0{,}07$)	0,78 ($\sigma = 0{,}07$)	4,10 ($\sigma = 1{,}31$)	19,00 ($\sigma = 3{,}70$)
Alg. 7	90 ($\sigma = 19$)	45 ($\sigma = 27$)	0,83 ($\sigma = 0{,}09$)	0,70 ($\sigma = 0{,}08$)	4,80 ($\sigma = 1{,}10$)	15,90 ($\sigma = 2{,}00$)

Tabela 5.1: Primerjava med rezultati, dobljenimi z algoritmom 2 in rezultati algoritma 7 – parameter ϑ ni upoštevan. V stolpcih PR in NR sta prikazana odstotka pravilno oz. napačno razpoznanih mešičkov, stolpca r_1 in r_2 prikazujeta vrednosti količnikov, stolpec d absolutno razdaljo med mejo razpoznanega mešička in dejanskega mešička izraženo v pikslih, stolpec κ pa prikazuje kompaktnost. S σ je označena vrednost standardnega odklona pri posamezni količini.

Na sliki 5.2 je grafično prikazana primerjava med rezultati, dobljenimi z algoritmom za statične 2D slike, in rezultati algoritma za razpoznavanje s postopki predikcije. V stolpcih PR in NR sta prikazana deleža pravilno oz. napačno razpoznanih mešičkov za posamezen algoritem (odstotek dobimo tako, da vrednost pomnožimo s 100), stolpca r_1 in r_2 pa prikazujeta vrednosti količnikov. Stolpci s sivo barvo prikazujejo rezultate algoritma 2, stolpci z belo pa prikazujejo rezultate algoritma 7.



Slika 5.2: Primerjava med rezultati algoritma za statične 2D slike in rezultati algoritma za razpoznavanje iz zaporedja slik s postopki predikcije – parameter ϑ ni upoštevan. Stolpec PR prikazuje delež pravilno razpoznanih mešičkov, stolpec NR delež napačno razpoznanih, stolpca r_1 in r_2 pa ustrezna količnika za posamezen algoritem.

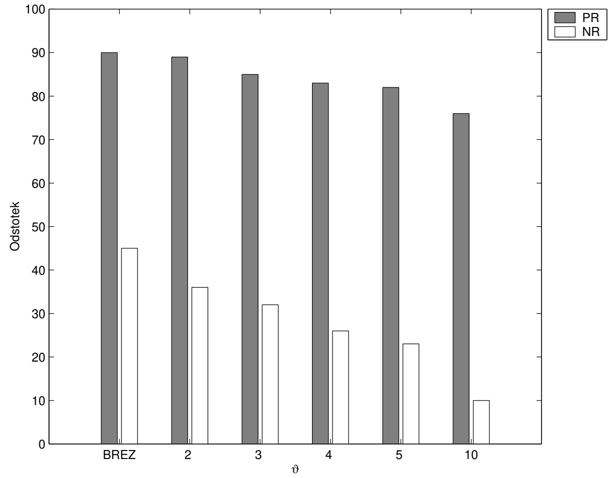
V rezultatih, ki smo jih predstavili, smo upoštevali kar vse regije v slikah, ne glede na koliko zaporednih slikah se pojavljajo. Dejansko smo ignorirali vhod ϑ algoritma 7. V nadaljevanju pa si bomo ogledali, kako se spreminjajo rezultati, če upoštevamo parameter ϑ. Parameter ϑ določa minimalno število zaporednih slik, na katerih se mora regija pojaviti, da jo še štejemo kot veljavno. Vidimo, da gre za neke vrste pragovno operacijo. Regije, ki ne dosežejo tega praga, zavržemo, preostale pa obdelamo. Nastavitev vrednosti praga ϑ je seveda odvisna od aplikacije – tipa reševane naloge. Za naše testne slike ni bilo na voljo takšnega podatka, zato praga ϑ nismo smeli postaviti na eno samo fiksno vrednost. Raje smo študirali, kako se spreminjajo kvantitavne mere uspešnosti algoritma v odvisnosti od parametra ϑ. Tako smo ovrednotili rezultate algoritma 7 za 6 različnih vrednosti parametra ϑ. Najprej smo izračunali rezultate za regije, kjer nismo postavili nobenih omejitev (BREZ) – rezultati že opisani zgoraj, zatem smo pri izračunu upoštevali vse regije, ki se pojavljajo na vsaj dveh zaporednih slikah (2), zatem na vsaj treh zaporednih slikah (3), štirih (4), petih (5) ter na vsaj desetih zaporednih slikah (10). Rezultati so pokazali, da se je ob povečanju vrednosti parametra ϑ odstotek napačno razpoznanih mešičkov manjšal, in sicer od 45 % (standardni odklon 27 %) v varianti brez omejitev (BREZ) na 36 % z odklonom 28 % pri upoštevanju, da se mora regija pojaviti na vsaj dveh zaporednih slikah, 32 % z odklonom 28 % pri $\vartheta = 3$, 26 % z odklonom 27 % pri $\vartheta = 4$, 23 % z odklonom 27 % pri $\vartheta = 5$, do 10 % z odklonom 20 % pri upoštevanju, da se mora regija pojaviti na vsaj desetih zaporednih slikah. Ob povečanju parametra ϑ se je tudi odstotek pravilno razpoznanih mešičkov manjšal, in sicer od 90 % s standardnim odklonom 19 % (BREZ), na 89 % z odklonom 20 % (2), 85 % z odklonom 26 % (3), 83 % z odklonom 26 % (4), 82 % z odklonom 26 % (5), do 76 % s standardnim odklonom 27

% (10). Rezultati so pregledno zbrani v tabeli 5.2.

	ϑ = BREZ	$\vartheta = 2$	$\vartheta = 3$	$\vartheta = 4$	$\vartheta = 5$	$\vartheta = 10$
PR (v %)	90 ($\sigma = 19$)	89 ($\sigma = 20$)	85 ($\sigma = 26$)	83 ($\sigma = 26$)	82 ($\sigma = 26$)	76 ($\sigma = 27$)
NR (v %)	45 ($\sigma = 27$)	36 ($\sigma = 28$)	32 ($\sigma = 28$)	26 ($\sigma = 27$)	23 ($\sigma = 27$)	10 ($\sigma = 20$)

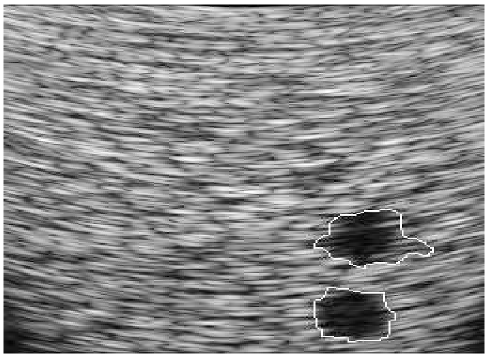
Tabela 5.2: Odstotek pravilno (vrstica PR) in napačno (vrstica NR) razpoznanih mešičkov v odvisnosti od parametra ϑ. S σ je označena vrednost standardnega odklona pri posamezni količini.

Ob spreminjanju parametra ϑ se vrednosti količnikov r_1 in r_2, povprečne absolutne razdalje in kompaktnosti niso bistveno spreminjale (do 2 %), zato jih tudi nismo posebej podajali. Prikažimo opisane rezultate še v strnjeni obliki krivulje ROC (*receiver operating characteristic*) na sliki 5.3. S sivo barvo je prikazan odstotek pravilno razpoznanih mešičkov, z belo pa odstotek napačno razpoznanih regij v odvisnosti od vrednosti parametra ϑ. Vidimo, da že z nastavitvijo parametra ϑ na vrednost 2, bistveno zmanjšamo odstotek napačno razpoznanih (za 20 % glede na začetno vrednost) ob praktično enakem odstotku pravilno razpoznanih mešičkov. S tem očitno izboljšamo razpoznavanje glede na razpoznavanje z algoritmom za statične 2D slike.



Slika 5.3: Krivulja ROC – odstotek pravilno in napačno razpoznanih mešičkov v odvisnosti od parametra ϑ. S sivo barvo je označen odstotek pravilno razpoznanih, z belo pa odstotek napačno razpoznanih.

Na sliki 5.4 je prikazan tipičen rezultat algoritma 7, kjer je upoštevan parameter ϑ ($\vartheta = 3$). Rezultat lahko primerjamo z levo sliko prvega para s slike 5.1, kjer parametra ϑ nismo upoštevali. Vidimo, da smo v tem primeru uspeli odstraniti vse napačne regije.



Slika 5.4: Tipičen rezultat algoritma 7 ob uporabi parametra ϑ. Rezultat lahko primerjamo s sliko 5.1.

5.2 Rezultati na realnih slikah

V prejšnjem podpoglavju smo prikazali in ovrednotili rezultate na umetnih slikah. V nadaljevanju pa bomo opisali dobljene rezultate na realnih slikah. Kvaliteto rezultatov bomo lahko ocenjevali le vizualno in ne z eksaktnimi kvantitativnimi merami. Seveda bomo tudi nenehno delali primerjavo med rezultati na osnovi postopka predikcije in rezultati klasičnega 2D algoritma.

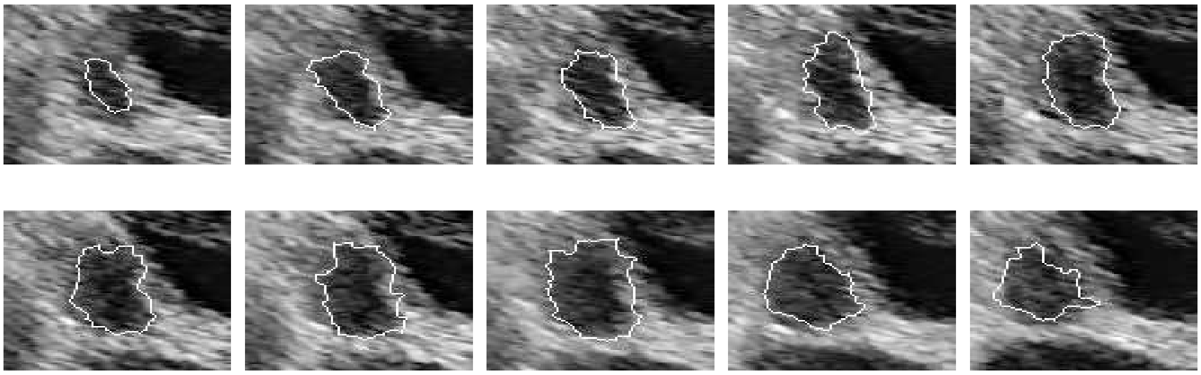
Razlog, da bomo lahko ocenjevali rezultate zgolj vizualno, leži v tem, da je ekspert ocenil le določeno majhno podmnožico slik (50 slik) iz celotnega nabora vseh slik (okrog 6000). Pri ocenjevanju je imel ekspert vpogled le v trenutno sliko iz zaporedja. Torej ni imel nikakršnih informacij o predhodnih in naslednjih slikah trenutne slike. Dejansko je imel enake razmere, kot jih ima algoritem za razpoznavanje v klasičnih 2D slikah (podpoglavje 3.3). Slike je ocenjeval le en ekspert. Iz vsega tega lahko sklepamo, da je opazovalčeva variabilnost izredno velika – v poglavju 3 smo to variabilnost ocenili na 8 %. Lahko bi se vprašali, zakaj pa ne damo slik ponovno oceniti ekspertu, ki bi imel vpogled v predhodne in naslednje slike iz zaporedja. To bi vsekakor zmanjšalo opazovalčevo variabilnost, vendar izračunane kvantitavne mere še vedno ne bi bile eksaktne. Zakaj? Naše testne slike so dejansko prerezi skozi 3D strukturo jajčnika. Ker pa dejanska 3D zgradba jajčnika za vsako pacientko ni poznana, bi bil zato vsak ekspertov odčitek na 2D sliki zgolj približek. Ekspertov približek zagotovo tudi ne bi mogel vsebovati korekcij, ki jih prinašajo detajlne informacije o objektu iz njegovega pojavljanja v zaporedju slik, saj človek ni sposoben pomniti množice drobnih detajlčkov, recimo majhnih sprememb več točk v konturi objekta, in jih nato strniti v optimalno oceno. To zmore le stroj-računalnik. Prave referenčne položaje lahko dobimo le na umetnih slikah z ustreznimi programskimi ali strojnimi simulatorji.

V tem podpoglavju bomo torej prikazali nekaj tipičnih rezultatov, ki najbolje okarakterizirajo delovanje algoritma 7. Pokazali bomo tudi nekaj primerov, kjer je algoritem odkril nekonsistentnost v ekspertovih odčitkih.

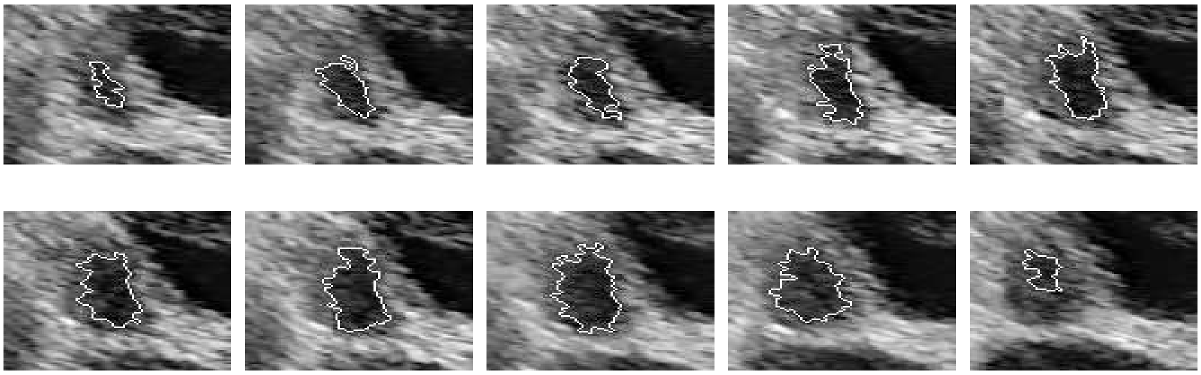
Najprej pokažimo, kako algoritem 7 obdela posamezno regijo s postopki predikcije na vseh slikah, kjer se le-ta pojavi. Na sliki 5.5 je prikazan eden izmed tipičnih rezultatov takšnega obdelovanja. Zaporedje pod a) prikazuje, kako se regija spreminja skozi zaporedne prereze. Prva slika, kjer se je regija prvič pojavila, je prikazana v levem zgornjem kotu, naslednje slike iz zaporedja pa so prikazane od leve proti desni in od vrha proti dnu. Tako je zadnja slika, kjer smo še zaznali to regijo, prikazana v desnem spodnjem kotu. Za prikaz smo izbrali vsako tretjo oz. četrto sliko iz dejanskega zaporedja, in sicer je regija prikazana na 1. sliki, 4., 8., 11., 15., 18., 22., 25., 29. in zadnji – 32. sliki zaporedja. Te rezultate lahko primerjamo z rezultati, dobljenimi z algoritmom na statičnih 2D slikah – algoritem 2, ki so na sliki 5.5 prikazani na enak način pod točko b). Naj še enkrat poudarimo, da ti rezultati hkrati predstavljajo tudi vhod v algoritem 7. Ob primerjavi obeh rezultatov opazimo, ne le da so rezultati algoritma 7 veliko bolj kompaktni, ampak so ob tem še tudi veliko bolj natančni. Dobro lahko namreč vidimo, da regija sedaj skoraj v celoti pokriva temno območje, na katerem se nahaja, kar za algoritem 2 ni držalo. Ob opazovanju rezultatov se lahko zastavi še en dvom, in sicer: na prvi in zadnji sliki iz zaporedja (glej slike pod točko a) na sliki 5.5) je regija še dokaj velika, torej se najverjetneje pojavlja tudi na nekaj predhodnih in naslednjih slikah. To drži. Ker pa algoritem 2 te regije na predhodni oz. naslednji sliki ni zaznal, smo regijo izgubili. Algoritem 7 je namreč zasnovan tako, da na vsaki sliki pričakuje veljavno meritev in zato konča obdelovanje s tisto sliko, kjer zadnjič dobi vhodne podatke za neko regijo (glej poglavje 4). Na koncu še pokažimo, da je tudi ekspert na tem področju razpoznal mešiček. Referenčni položaj mešička za 29. sliko iz zaporedja a) na sliki 5.5, kot ga je določil ekspert, je označen na zgornjem desnem posnetku na sliki 5.6 (najmanjši mešiček na sliki). Vidimo skoraj popolno ujemanje med našimi rezultati in ekspertovim odčitkom.

Prikazali smo le enega izmed rezultatov. Podobne rezultate opazimo tudi pri drugih regijah. Na sliki 5.6 jih je zbranih še. Prva slika vsake trojice prikazuje rezultate algoritma 7, druga slika rezultate algoritma 2, tretja slika pa referenčne položaje mešičkov, kot jih je določil ekspert. Zaradi lažje primerjave smo prikazali kar vse regije v slikah – parametra ϑ pri tem nismo upoštevali. Vidimo, da z dobljenimi rezultati podpiramo trditve o kvaliteti razpoznavanja z algoritmom na osnovi postopkov predikcije.

Na teh nekaj primerih smo lahko dobro videli veliko moč algoritma 7. Ob tem, da so rezultati sedaj veliko bolj kompaktni in tudi natančni, pa dosežemo z akumuliranjem znanja o razpoznavanju regij skozi zaporedje v algoritem 7 veliko večjo gotovost v pravilnost dobljenih rezultatov. Pri tem nam je v veliko pomoč parameter ϑ, s katerim nadzorujemo "veljavnost" regij. Če se regija npr. pojavlja na mnogo zaporednih slikah, potem tudi razpoznavanje v nekaj naslednjih slikah, kjer je ta regija že močno zabrisana ali kako drugače popačena, za algoritem 7 ne predstavlja posebnega problema. Prav nasprotno



(a)

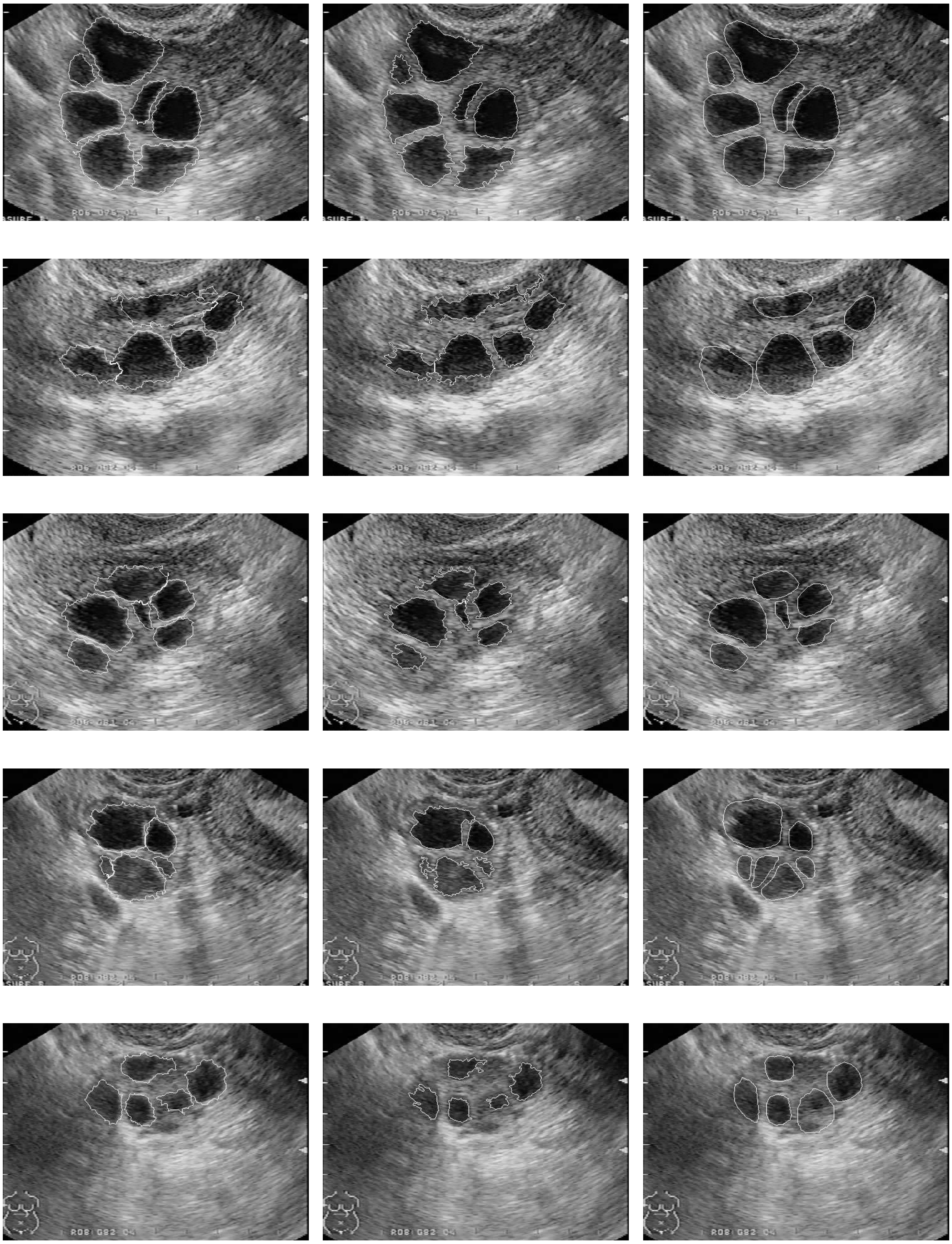


(b)

Slika 5.5: Tipični rezultat obdelovanja regije skozi zaporedje slik. Pod točko a) so prikazani rezultati algoritma 7, pod točko b) pa rezultati algoritma 2, ki so hkrati vhod v algoritem 7. Prva slika, kjer se pojavi regija, je prikazana v levem zgornjem kotu, naslednje slike ji sledijo od leve proti desni in od vrha proti dnu. Prikazane so 1., 4., 8., 11., 15., 18., 22., 25., 29. in 32. slika dejanskega računalniškega zaporedja.

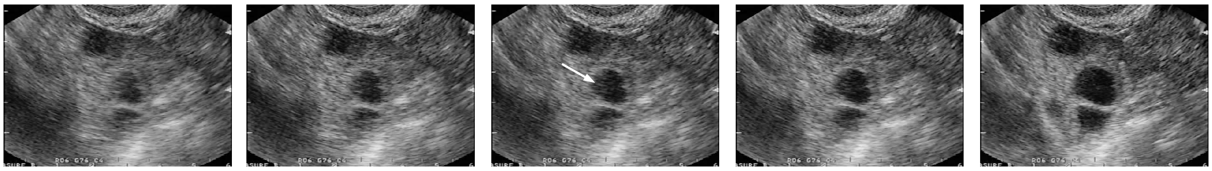
pa velja za algoritem za statične 2D slike oz. za eksperta, kjer oba ocenjujeta le eno samo 2D sliko. Pri takšnih nejasnih slikah zato ne moremo biti popolnoma prepričani o njunih odčitkih – tudi ekspert sam v takšnih primerih pokaže določeno nezaupanje v svoje odčitke (tipičen komentar eksperta: na tem mestu bi lahko bil mešiček ipd.). V nadaljevanju bomo pokazali rezultate razpoznavanja za nekaj takšnih težkih primerov.

Na sliki 5.7 je prikazan prvi takšen primer. V zaporedju a) so prikazani originalni ultrazvočni posnetki jajčnika z mešički. Številke pod slikami pomenijo, koliko slik pred (minus pred številom) oz. za (plus v številu) trenutno sliko – označena z 0 – se posamezna slika nahaja. S številom 0 označujemo trenutno sliko, ki jo je ocenjeval tudi ekspert. Na sliki 5.7 c) so prikazani vsi mešički, kot jih je ročno označil ekspert, medtem ko na sliki 5.7



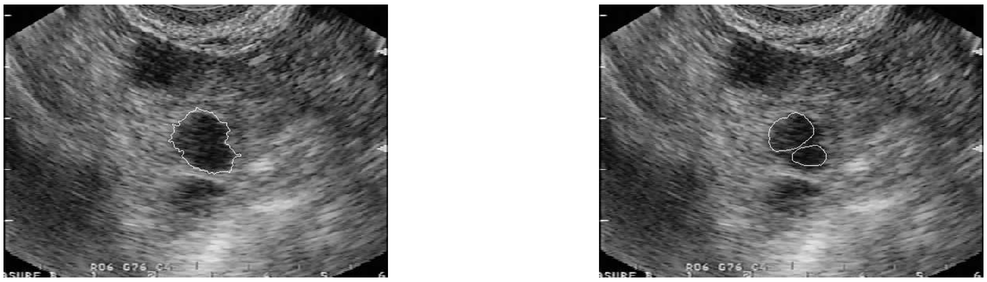
Slika 5.6: Nekaj tipičnih rezultatov na realnih slikah. Leva slika vsake trojice prikazuje mešičke, kot jih je razpoznal algoritem 7, srednja slika razpoznane mešičke z algoritmom 2, na desni sliki pa je položaj mešičkov označil ekspert.

b) vidimo največji mešiček, razpoznan z algoritmom 7. Opazimo neskladje med obema rezultatoma za največjo temno regijo v sliki – v sliki 5.7 a) označena s puščico. Ekspert je ocenil, da sta na tem področju dva mešička, medtem ko je algoritem 7, ki se je opiral na rezultate razpoznavanja skozi celotno zaporedje slik, določil na tem mestu le enega velikega. Le na osnovi trenutne slike težko presodimo, kateri rezultat je pravilen. Če pogledamo nekaj predhodnih oz. nekaj naslednjih slik, pa lahko damo verjetnejšo oceno. Na predhodnih slikah (glej sliko 5.7 a)) imamo res občutek, da gre morebiti za dva mešička, predvsem zato, ker mešiček nima popolnoma ovalne oblike. V naslednjih slikah pa postane očitno, da gre za en sam mešiček. Ta mešiček se v zaporedju slik pojavlja na približno 100 slikah. V vseh teh slikah ga je algoritem 7 tudi pravilno razpoznal.



-3 -1 0 +1 +5

(a)



(b) (c)

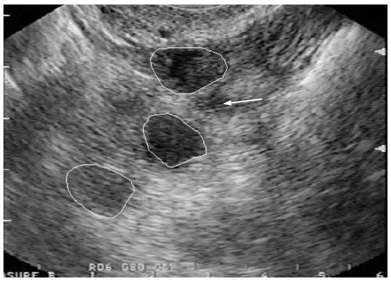
Slika 5.7: Primer 1 – ekspert označil dva mešička, algoritem 7 pravilno razpoznal na tem mestu le enega samega. Pod a) so prikazane originalne slike sivin – trenutna slika je označena z 0. Slike, ki se nahajajo v zaporedju pred njo, so označene z negativnim številom (cifra pomeni število slik) oz. za njo s pozitivnim. V b) je prikazan največji mešiček, razpoznan z algoritmom 7, v c) pa mešički, kot jih je ročno označil ekspert. S puščico je označena regija, kjer je prišlo do neskladja med rezultati eksperta in algoritma 7.

Ogledali smo si prvi tip napak, ki smo jih zasledili pri testiranju algoritma. Ekspert je dejansko v posameznih regijah določil več mešičkov, kot jih je tam res bilo. Algoritem je v večini teh primerov pravilno razpoznaval. Zasledili smo tudi drugo vrsto napak. V tem primeru pa je algoritem pravilno razpoznal mešiček, medtem ko takšnega mešička ekspert sploh ni označil. Takšen primer pa vidimo na sliki 5.8. Ekspert mešička ni označil (označeno s puščico na sliki 5.8 c)), medtem ko na sliki 5.8 b) vidimo razpoznan mešiček z algoritmom 7 (zaradi večje preglednosti je prikazan le ta mešiček). Da je regija res mešiček, lahko vidimo iz zaporedja slik (slika 5.8 a)), kjer postane to očitno. Opravka imamo z mešičkom, ki počasi izginja. Na trenutni sliki, označeni z 0, je ta mešiček že tako nejasen in zapacan, da ga je samo na osnovi te slike nemogoče določiti. Od tu tudi izhaja napaka eksperta. Algoritem na osnovi postopkov predikcije nima v takšnih zapacanih slikah posebnih težav. To pa zaradi tega, ker se skozi Kalmanov filter po vsaki sliki vgrajuje v algoritem novo znanje o obliki razpoznavanega objekta. Če je meritev v sliki še tako slaba, bo prav zaradi tega vgrajenega znanja, ki se kaže v obliki predikcije, končni rezultat še vedno dober. Rezultati bodo seveda tem boljši, na čim več slikah se mešiček pojavlja.

-9 -4 -1 0 +1

(a)

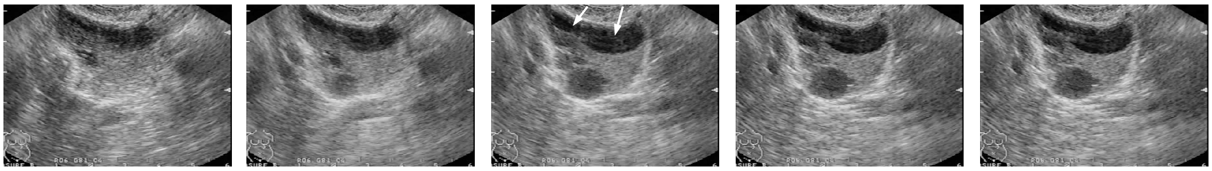
(b)



(c)

Slika 5.8: Primer 2 – Algoritem 7 je pravilno razpoznal mešiček, ekspert tega mešička ni označil. Pod a) so prikazane originalne slike sivin. V b) je prikazan "sporni" mešiček, razpoznan z algoritmom 7, v c) pa vsi mešički, kot jih je ročno označil ekspert. S puščico je označena regija, kjer je prišlo do neskladja med rezultati eksperta in algoritma 7.

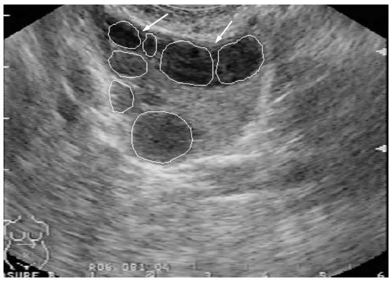
Zadnjo vrsto napak pa smo opazili pri razpoznavanju regij, ki se nahajajo na močno šumnih oz. izredno nejasnih delih slike. Dejansko gre za primere, kjer je tudi iz zaporedja slik skoraj nemogoče oceniti, ali je posamezna regija res mešiček ali ne. Na sliki 5.9 vidimo takšen primer. Ekspert je na trenutni sliki označil 7 mešičkov. S puščico smo označili po dva para mešičkov, za katere po našem mnenju, tudi iz zaporedja slik sivin, ni mogoče določiti, ali so mešički ali ne. Rezultat razpoznavanja algoritma 7 je prikazan na sliki 5.9 b). Naš algoritem teh regij ni razpoznal kot mešičke. Naj kot zanimivost omenimo še to, da je ekspert moral oceniti tudi sliko, ki se nahaja 6 slik pred trenutno sliko (na sliki 5.9 a) označena z -6. Na tej sliki pa ekspert ni označil nobenega mešička. Seveda pa lahko tudi dobro vidimo, da se obe sliki v tem predelu slike skoraj ne razlikujeta med seboj.



-6 -2 0 +2 +10

(a)

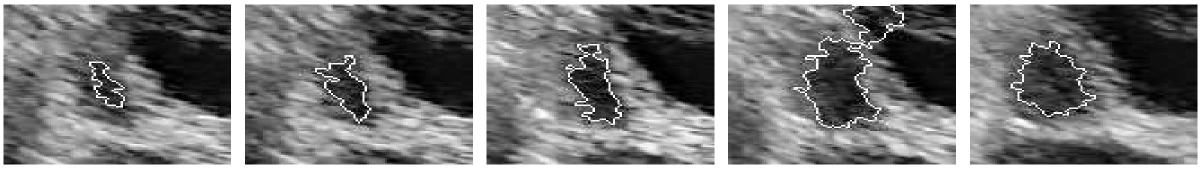
(b)



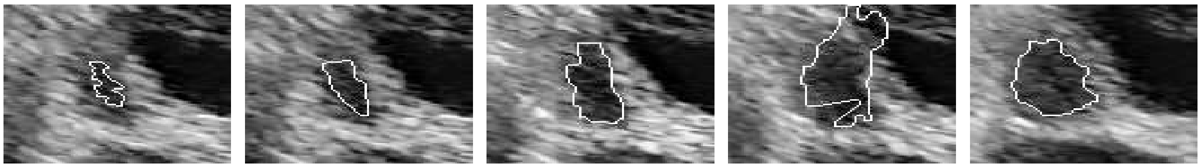
(c)

Slika 5.9: Primer 3 – Razpoznavanje regij na močno šumnih oz. nejasnih delih slike. Pod a) so prikazane originalne slike sivin. V b) so prikazani mešički, razpoznani z algoritmom 7 – na tem nejasnem delu slike algoritem ni razpoznal mešičkov, v c) pa mešički, kot jih je ročno označil ekspert. S puščico so označene regije, kjer je tudi iz zaporedja slik skoraj nemogoče oceniti, ali gre za mešičke ali ne.

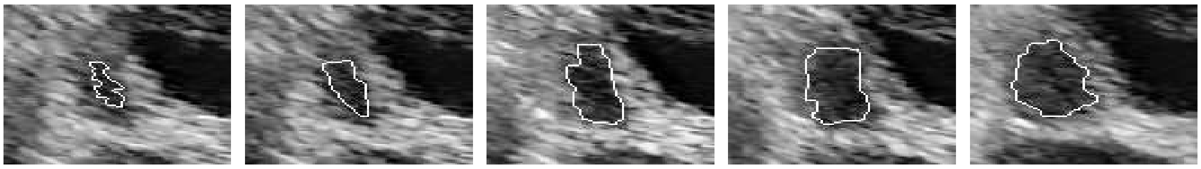
V nadaljevanju bomo primerjalno ocenili še vse popravke osnovnega predikcijskega algoritma, in sicer tako, da bomo ovrednotili delovanje brez dodatkov (algoritem 3) in z njimi (vse različice algoritma do končne oblike – algoritem 7). Zaradi boljše demonstracije smo uporabili realne slike, saj postane tukaj vpliv posameznih dodatkov veliko bolj opazen kot na umetnih slikah. Na sliki 5.10 so zbrani vsi rezultati. V zaporedju a) so prikazani razpoznani mešički, dobljeni z algoritmom za statične 2D slike. Ti rezultati predstavljajo vhod v algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Gre za isto zaporedje ultrazvočnih posnetkov kot pri primeru s slike 5.5, vendar pa smo kot algoritem za statične 2D slike uporabili eno izmed prvih različic algoritma 2 (homogene regije določali le na osnovi standardnega odklona). To različico algoritma smo uporabili zaradi boljše demonstracije, saj smo v zaporedju s tem dobili tudi veliko napako pri razpoznavanju mešičkov (velika napaka v meritvi je pri 20. in 21. sliki zaporedja), ki je z algoritmom 2 na tem zaporedju sicer ne dobimo. V zaporedju b) so prikazani rezultati, dobljeni z osnovno obliko algoritma za razpoznavanje objektov s postopki predikcije (algoritem 3). Razpoznan mešiček je prikazan za 1., 3., 11., 21. in 29. sliko zaporedja – vseh slik v zaporedju je 32. Algoritem na osnovi predikcije začne prediktor-korektor obdelovanje šele na drugi sliki iz zaporedja, zato ostane odčitek s prve slike kar enak (glej poglavje 4). V naslednjih zaporednih slikah postaja mešiček bolj kompakten (3. in 11. slika). V 21. sliki zasledimo veliko napako v meritvi, ki ima tudi izredno močan negativen vpliv na razpoznavanje. Razpoznan mešiček je močno popačen in neuporaben za kakršnokoli medicinsko analizo. Odčitki (meritve) na naslednjih zaporednih slikah so spet blizu dejanskemu poteku mešička, zato se modelni parametri Kalmanovega filtra, ki so se po veliki napaki v meritvi močno pokvarili, lahko ponovno priučijo pravilnih vrednosti, s tem pa se izboljša tudi samo razpoznavanje – glej 29. sliko iz zaporedja b). Z algoritmom 4 smo vpeljali v osnovno obliko algoritma mehanizem, ki odpravlja velike napake v meritvah. Rezultati algoritma 4 so prikazani v zaporedju c). Vidimo, da dokler se ne pojavi velika napaka v meritvah, so rezultati med obema različicama algoritma identitični. Opazimo pa, da algoritem 4, v nasprotju od osnovne oblike algoritma, brez težav obvlada nekaj zaporednih napačnih meritev (v tem primeru dve). Naslednji dodatek k osnovni obliki algoritma je bilo upoštevanje lokalnosti krivulje (algoritem 5). Rezultati algoritma 5 so prikazani v zaporedju d). Vidimo, da so dobljeni rezultati zelo podobni rezultatom algoritma 4. Če natančno opazujemo mešičke pa vidimo, da so razpoznani mešički sedaj na posameznih delih bolj gladki (glej 3. sliko). To je še bolj očitno, če imamo vpogled v celotno zaporedje slik. S tretjim dodatkom k osnovni obliki algoritma, ki je integriran v algoritem 6, smo reševali problem sistemske napake v merilnem sistemu (meritev konstantno podcenjena). Rezultati razpoznavanja za algoritem 6 so prikazani v zaporedju e). Vidimo, da so sedaj razpoznani mešički večji (bolj napihnjeni) kot pri razpoznavanju s predhodnimi različicami. Ta efekt je najbolje viden ob obeh robovih zaporedja. Rezultati, dobljeni s končno obliko algoritma za razpoznavanje objektov iz zaporedja slik s postopki predikcije – algoritem 7, so prikazani v zaporedju f). Algoritem 7 obdeluje zaporedje slik v dveh prehodih. V prvem prehodu obdela slike od začetka proti koncu, v drugem prehodu pa uporabi izračunane modelne parametre ter zaporedje obdela še od zadaj naprej. Iz rezultatov opazimo, da je v veliki meri odpravljen prehodni pojav, ki se pojavlja na začetku zaporedja. Obris mešička na 1. in 3. sliki je sedaj skoraj popolnoma pravilen. Tudi na ostalih slikah je mešiček ustrezno napihnjen in je kvaliteten približek dejanskega mešička.



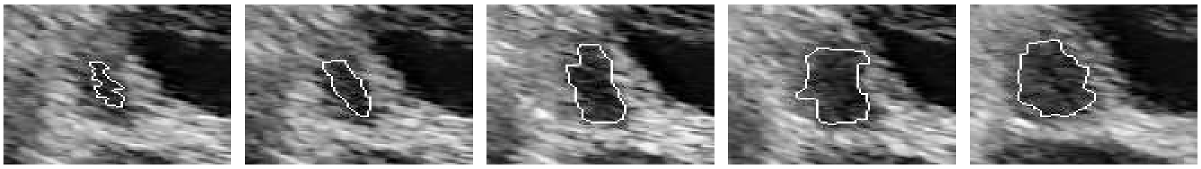
(a)



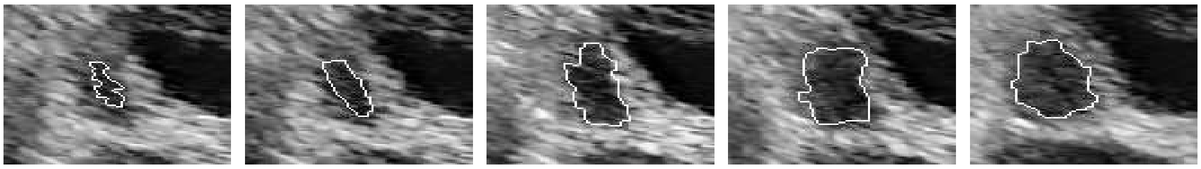
(b)



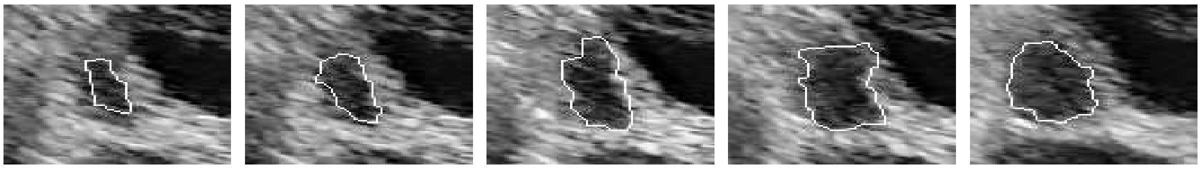
(c)



(d)



(e)



(f)

Slika 5.10: Delovanje predikcijskega algoritma brez in z vsemi dodatki. Pod točko a) so prikazani vhodi. V točkah od b) do f) so prikazani rezultati predikcijskega algoritma: b) osnovna oblika (alg. 3), c) dodan korak za odpravljanje velikih napak v meritvah (alg. 4), d) upoštevana lokalnost krivulje (alg. 5), e) dodan korak popravljanja meritve (alg. 6) in f) končna oblika algoritma (alg. 7). Prikazane so 1., 3., 11., in 29. slika dejanskega računalniškega zaporedja.

5.3 Razprava

V prejšnjih podpoglavjih smo prikazali rezultate razpoznavanja, dobljene z novim algoritmom za razpoznavanje objektov iz zaporedja slik s postopki predikcije (poglavje 4). Vpeljani algoritem je v svoji osnovi popolnoma splošen in neodvisen od tipa objektov, ki jih razpoznavamo. Lahko ga obravnavamo kot vrsto ogrodja, ki ga šele z izbiro tipa naloge dokončno oblikujemo. Naš algoritem smo preverili na primeru razpoznavanja jajčnih mešičkov v ultrazvočnih slikah. Temu ustrezno so bili prirejeni tudi posamezni koraki algoritma (podpoglavja 4.2.2 in 4.2.4).

Algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije (algoritem 7) smo torej preverili na množici umetnih in realnih ultrazvočnih posnetkov jajčnikov z mešički. Kot rečeno, je vhod v algoritem 7 ob zaporedju slik sivin še zaporedje že segmentiranih slik s ciljnimi objekti. Za segmentirane slike smo vzeli kar izhod algoritma za razpoznavanje na statični 2D sliki (algoritem 2), za katerega smo pokazali (podpoglavje 3.3), da se njegova učinkovitost za primer razpoznavanja mešičkov bliža optimalni.

Dobljene rezultate razpoznavanja smo prikazali v podpoglavjih 5.1 in 5.2. Za rezultate na umetnih slikah smo izračunali tudi kvantitavne kazalce uspešnosti algoritma, rezultate na realnih slikah pa smo zaradi nepoznavanja dejanskih referenčnih položajev mešičkov v slikah (dejanski 3D model jajčnika neznan, velika opazovalčeva variabilnost) ocenjevali zgolj vizualno.

Ker so bili mešički na umetnih slikah precej dobro razpoznani že z algoritmom 2 (glej podpoglavje 3.4)), tudi nismo dosegli povečanja odstotka razpoznanih mešičkov po obdelovanju z algoritmom 7. Odstotek pravilno razpoznanih je tako ostal okrog 90 % (s standardnim odklonom 19 %). Če smo pri obdelovanju upoštevali kar vse regije – parametra ϑ nismo upoštevali – potem je ostal enak tudi odstotek napačno razpoznanih, in sicer okrog 45 % (z odklonom 27 %).

Poglejmo še kvaliteto dobljenih rezultatov. Zasledili smo majhno povečanje količnika r_1 glede na rezultate algoritma 2, in sicer od 0,78 na 0,83, ter majhno zmanjšanje količnika r_2 od 0,78 na 0,70. S tem se je tudi povečala povprečna absolutna razdalja meje razpoznanega mešička do meje dejanskega mešička, in sicer od 4,10 piksla na 4,80 piksla. Iz teh rezultatov bi lahko sklepali, da z algoritmom 7 dejansko nismo dosegli nobenega izboljšanja razpoznavanja (tudi izboljšanja kvalitete razpoznanih mešičkov ne). Vendar temu ni tako, kar bomo pokazali v naslednjih odstavkih. Razlog za takšne majhne spremembe v količnikih r_1 in r_2, ki v grobem določata kvaliteto razpoznanih mešičkov, leži v strukturi oz. tvorjenju umetnih slik (podpoglavje 3.2). Simulator za tvorjenje umetnih ultrazvočnih posnetkov jajčnikov ne posnema popolnoma dejanskega stanja, zato tudi izračunane kvantitavne mere, predvsem količnika r_1 in r_2 ter absolutna razdalja od meje razpoznanega mešička do meje dejanskega mešička, na umetnih slikah niso popolnoma merodajne. Na sliki 5.1 v drugem stolpcu lahko vidimo dejanske položaje mešičkov (elips), na osnovi katerih smo tvorili umetne ultrazvočne slike. Opazimo lahko, da prav nobena nastala temna regija ne ustreza v celoti dejanskemu mešičku. Še več. V večini primerov je prišlo tudi do majhnega premika med nastalo regijo in dejanskim mešičkom. Tako vidimo na umetnih ultrazvočnih slikah (slika 5.1), kamor smo vrisali dejanske mešičke, da mešički na nekaterih delih potekajo tudi popolnoma zunaj temnega območja. Prav zaradi tega premika se tudi zgodi, da se dejanski mešiček zaključi, še preden doseže rob temnega območja. Treba je še poudariti, da v teh umetnih slikah roba dejanskega mešička ni mogoče zaznati niti s prostim očesom niti kako drugače. Naš algoritem, in tudi algoritem za statične 2D slike, pa sta zasnovana na predpostavki, da so objekti, ki jih razpoznavamo, temnejši kot njihova okolica. Oba algoritma torej skušata kar se da najbolje razpoznati temno regijo. Ob takšnem vrednotenju pa vidimo očitno superiornost rezultatov algoritma 7 nad rezultati algoritma 2. Algoritem 7 dejansko očrta temno regijo v sliki optimalno. Naredili smo naslednji poizkus. Vzeli smo umetne slike ter na njih sami označili “dejanski položaj” mešičkov, seveda nismo imeli vpogleda v pravi položaj – vizualno smo ocenili vsako sliko, ob tem pa smo imeli vpogled v celotno zaporedje slik. Izračunali smo skoraj popolno ujemanje med rezultati algoritma 7 in našimi odčitki (količnika r_1 in r_2 blizu 1). Rezultati predikcijskega algoritma so bili v večini primerov celo bolj kompaktni od naših odčitkov. Izkazalo se je, da so bili rezultati algoritma 7 boljši od rezultatov algoritma 2. Algoritem 7 dejansko razpoznava temne regije tako, kot bi jih sami s prostim očesom, seveda z večjo natančnostjo. Na sliki razpoznava tisto, kar je dejansko vidno. Takšna izboljšava rezultatov je posebej očitna na realnih slikah.

Opazovali smo tudi kompaktnost razpoznanih mešičkov. Zasledili smo večjo kompaktnost rezultatov pri algoritmu 7 kakor pri algoritmu 2. Vrednost kompaktnosti se je z 19,00 za algoritem 2 zmanjšala na 15,90 pri algoritmu 7 – manjša vrednost pa pomeni dejansko večjo kompaktnost razpoznane regije. Pričakujemo lahko, da bodo rezultati po obdelavi z algoritmom 7 v povprečju za okrog 15 % bolj kompaktni glede na rezultate po obdelovanju z algoritmom za statične 2D slike. Takšno izboljšanje kompaktnosti je vsekakor v veliki meri posledica upoštevanja lokalnosti krivulje pri globalnem Kalmanovem filtru (glej varianto končnega algoritma – algoritem 5).

Eden izmed največjih problemov algoritmov, ki obdelujejo le eno 2D sliko naenkrat, je množica napačno razpoznanih regij. Kadar namreč razpoznavamo objekte le iz ene 2D slike, ponavadi uporabimo v zadnjem koraku takšnega razpoznavalnika še razna klasifikacijska pravila, na osnovi katerih se odločimo, ali je regija iskani objekt ali ne. In naj bodo klasifikacijska pravila še tako sofisticirana, vedno ostane še kakšna napačno razpoznana regija. Videli smo, da je bil odstotek napačno razpoznanih za umetne slike pri algoritmu 2 kar 45 %. Algoritem 7 pa rešuje ta problem z upoštevanjem zgodovine razpoznavanja. Kar z drugimi besedami pomeni tole: če je algoritem skozi več zaporednih slik razpoznal določeno regijo vedno na istem mestu, potem je najverjetneje ta regija iskani objekt. To logiko smo zajeli v parametru ϑ, ki določa minimalno število zaporednih slik, na katerih se mora pojaviti regija, da jo še upoštevamo. Pokazali smo že, da ostane odstotek napačno razpoznanih enak kot pri algoritmu 2, če parametra ϑ ne upoštevamo, in sicer 45 % pri umetni slikah. V podpoglavju 5.1 pa smo demonstrirali, da se z upoštevanjem tega parametra odstotek napačno razpoznanih bistveno zmanjša. Če nastavimo parameter ϑ na 10, kar pomeni, da se mora regija pojaviti na vsaj 10 zaporednih slikah, potem se odstotek napačno razpoznanih zmanjša na 10 %. V tem primeru dosežemo zmanjšanje odstotka napačno razpoznanih za kar izjemnih 78 % glede na začetno vrednost, ob tem pa se odstotek pravilno razpoznanih zmanjša le za okrog 15 % (z 90 % na 76 %) – glej sliko 5.3. Vrednost parametra ϑ pa je v praksi težko določiti. Odvisna je namreč od vrste reševane naloge. Vsekakor pa velja: čim večjo vrednost nastavimo, tem bolj smo lahko gotovi, da so razpoznane regije resnično iskani objekti. Pri naših testnih slikah si lahko pomagamo tudi s krivuljo ROC s slike 5.3, kjer je prikazano, kako se spreminjata odstotek pravilno in napačno razpoznanih v odvisnosti od parametra ϑ. Tako poiščemo na krivulji ROC takšno vrednost za ϑ, kjer je odstotek pravilno razpoznanih še vedno dovolj visok, odstotek napačno razpoznanih pa ustrezno manjši – iščemo sprejemljiv kompromis med še dopustnim zmanjšanjem odstotka pravilno razpoznanih in velikim zmanjšanjem napačno razpoznanih. Za realne ultrazvočne slike, kjer takšne statistike nismo mogli izračunati, pa lahko določimo predvidene vrednosti na osnovi raziskav pri umetnih slikah. Če bi torej v realnih slikah kot veljavne upoštevali le regije, ki se pojavljajo na vsaj petih zaporednih slikah, bi se odstotek pravilno razpoznanih z 78 % zmanjšal na okrog 72 %, ob tem pa bi se odstotek napačno razpoznanih zmanjšal z 29 % na okrog 15 %.

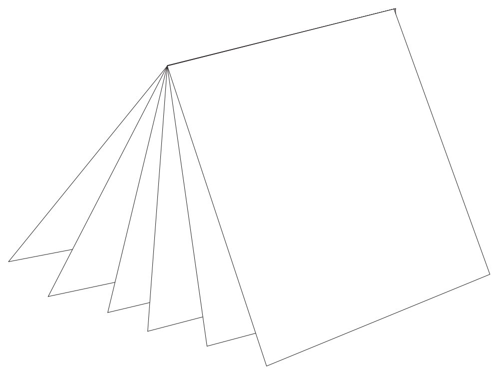
Na realnih slikah, kjer je to še bolj očitno, smo tudi pokazali, da z upoštevanjem zgodovine razpoznavanja dosežemo večjo gotovost v dobljene rezultate. To je posebej pomembno pri obdelovanju močno šumnih ali kako drugače popačenih slik. Na treh različnih vrstah problemov smo pokazali, da je algoritem 7 povsod pravilno razpoznaval mešičke in je v teh primerih celo prekašal eksperta z danega področja (podpoglavje 5.2). V takšnih mejnih situacijah je praktična vrednost našega algoritma še posebej velika. Vidimo ga kot orodje, ki opozarja eksperta na možne napake.

Algoritem 7 lahko dejansko obravnavamo tudi kot nekakšen postopek za poobdelavo rezultatov, dobljenih z algoritmi za statične 2D slike (seveda tudi rezultatov, dobljenih s katerim koli drugim algoritmom za obdelavo zaporedja slik). Najpomembnejši vhod v algoritem je namreč zaporedje že segmentiranih slik. Uspešnost razpoznavanja, predvsem mislimo na odstotek pravilno razpoznanih, se zato giblje v okvirih uspešnosti algoritma za statične 2D slike. Seveda se ob tem izboljša kvaliteta razpoznanih objektov, kljub temu pa se še vedno gibljemo v okolici rezultatov algoritma za 2D slike. Končni rezultati algoritma 7 so torej v veliki meri odvisni od vhoda – zaporedja segmentiranih slik. Če so namreč že sami vhodi konstantno slabi, morebiti celo napačni, potem bodo tudi rezultati algoritma 7 sicer res nekoliko boljši, pa še vedno slabi in s tem neuporabni. Bistvena predpostavka našega algoritma je, da so meritve (vhodi segmentirani objekti) normalno porazdeljeni okoli dejanskih položajev objektov. V podpoglavju 4.2.4 smo pokazali, kako lahko odpravimo sistemsko napako, če je meritev konstantno podcenjena (obratno bi postopali pri precenjeni meritvi). Meritve, ki so pa le na nekaj zaporednih slikah močno napačne, ne predstavljajo za naš algoritem nobenega problema. V algoritem smo namreč vpeljali postopek, ki odpravlja problem velikih napak v meritvah (podpoglavje 4.2.2).

Algoritem 7 sledi vsakemu objektu iz začetnega okvira skozi vse zaporedne slike, na katerih se objekt zatem še pojavi. Ujemanje objekta iz slike v sliko preverjamo s pogojem v enačbi (4.13). Če za objekt v trenutni sliki ne najdemo ujemanja, potem rečemo, da je poniknil. Za nas pomeni, da objekt ponikne, takoj ko ga na sliki več ne razpoznamo, tudi npr. le na eni sliki, zatem pa se lahko ponovno pojavi. Dejansko gre za problem izgube objekta med sledenjem – objekt bi se moral logično, zaradi še ustrezne svoje velikosti, pojaviti na še vsaj nekaj zaporednih slikah. Vzrok za "izgubljanje" objektov je prisotnost močnega šuma v slikah, ki moti delovanje algoritma za statične 2D slike. Algoritem 2 namreč določa, na kateri sliki objekt zaznamo prvič ter na kateri sliki zadnjič. V algoritmu 7 pa zgolj povzamemo rezultate algoritma za 2D slike, z reševanjem problema izgube objekta med sledenjem pa se nismo ukvarjali. Če objekt na nekaj zaporednih slikah izgine, pa se ponovno pojavi, potem v algoritmu 7 obravnavamo obe pojavljanji objekta, kot da sta različna objekta. Problem sledenja objektom, ki izginjajo in se zatem ponovno pojavijo, je težek problem. Če objekta v sliki ne najdemo, to za algoritem 7 pomeni, da meritev za objekt ni na voljo. Če slednje objektu prehitro zaključimo – objekt se dejansko pojavlja v zaporedju še na nekaj slikah – ali če objektu začnemo slediti prepozno – objekt je dejansko že prisoten na nekaj predhodnih slikah, kjer ga pa še nismo zaznali – oba ta problema lahko z majhno modifikacijo algoritma 7 zelo preprosto rešimo. Ko objekt v sliki izgubimo, s tem izgubimo tudi meritev. V tem primeru izredno močno obtežimo matriko $\mathbf{R}$ Kalmanovega filtra, tako da se v izračunu upošteva le predikcija. Tako bi algoritem 7 upošteval predikcijo, dokler objekt ne bi dejansko poniknil. Če bi se med takšnim obdelovanjem ponovno pojavila meritev, bi jo seveda upoštevali v izračunu. V primeru, ko sledenje zaključimo prehitro, bi takšno obdelovanje uporabili pri prvem prehodu algoritma 7, v drugem primeru, ko objektu začnemo slediti prepozno, pa pri drugem prehodu – obdelovanje od zadaj naprej. Veliko težji problem kot izguba objekta na obeh robovih (pred prvo oz. zadnjo sliko iz zaporedja, kjer zaznamo objekt) je, če objekt izgubimo nekje na sredi. Na robovih zagotovo vemo, da objekt ali raste ali pa se krči, za vmesne slike lahko spremembe zgolj predvidevamo. Tudi v tem primeru lahko uporabimo opisani postopek obdelovanja s predikcijo, vendar je gotovost v pravilnost rezultatov manjša. Treba pa je poudariti, da je izguba objekta ob obeh robnih območjih zaporedja veliko bolj pogosta, kot pa v sredini zaporedja, kjer so objekti-mešički zelo izraziti in jih lahko zato preprosto razpoznamo.

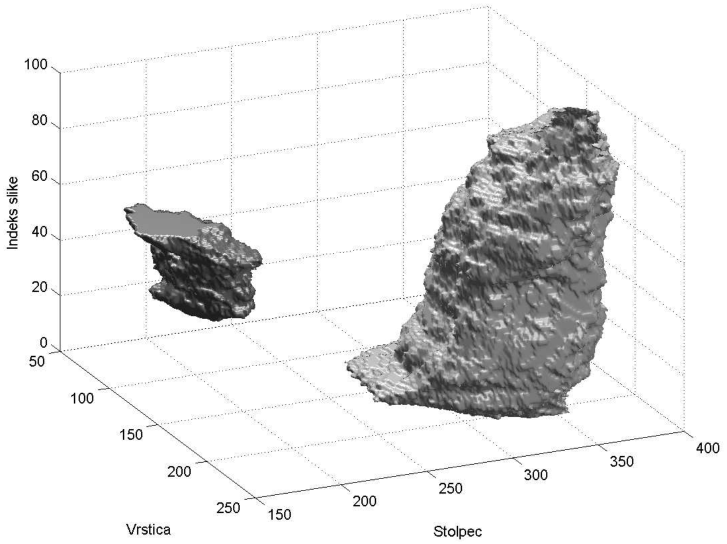
Algoritem 7 je prav zaradi uporabe Kalmanovega filtra kot svojega sestavnega dela zelo konzervativen, kar pomeni, da se počasi prilagaja na spremembe. Tu ne gre toliko za problem velikih hitrosti gibajočih se objektov (pri prerezih objektov za razmik med njimi), ki definira velikost premika objekta iz slike v sliko. To je dobro rešeno v pogoju ujemanja objekta iz slike v sliko (enačba (4.13)). Bolj mislimo na hitre spremembe v obliki objektov iz slike v sliko. Če smo prepričani, da se oblika objektov iz slike v sliko močno spreminja (močno elastični objekti), moramo dovoliti, da se Kalmanov filter bolj opira na meritve kot na modelne parametre – nastaviti majhne vrednosti v matriki $\mathbf{R}$.

Na koncu poglejmo še, kako lahko interpretiramo dobljene rezultate na naših testnih slikah. Opravka imamo z ultrazvočnimi posnetki jajčnikov z mešički. Vsaka slika predstavlja prerez skozi strukturo jajčnika in mešička (podpoglavje 3.2). Po razpoznavanju z algoritmom 7 imamo tako na vsaki sliki za vsak mešiček določeno njegovo notranjo mejo. To pa tudi pomeni, da bi lahko opravili 3D rekonstrukcijo tega mešička, če bi za posamezen mešiček zaporedoma zložili razpoznane meje eno ob drugi. Podobno velja za celoten jajčnik, kjer bi sestavili vse 3D rekonstukcije posameznih mešičkov. Ob tako izvedeni rekonstrukciji predpostavljamo, da so vsi prerezi med seboj vzporedni. To pa v realnosti ne drži popolnoma. Na sliki 5.11 je prikazano dejansko razmerje med zaporednimi prerezi, ki jih dobimo ob pregledu jajčnika z intravaginalno ultrazvočno sondo. Vidimo, da imajo vsi prerezi skupno izhodišče (sondo) ter da je kotni razmik med dvema zaporednima prerezoma konstanten. V realnosti prerezi torej niso nikoli vzporedni med seboj. Če pa je kotni razmik med dvema prerezoma majhen, ob tem pa so še opazovani objekti daleč od sonde, potem je napaka, če upoštevamo prereze, kot da so vzporedni, majhna.



Slika 5.11: Način tvorbe prerezov jajčnika ob uporabi intravaginalne ultrazvočne sonde. Kotni razmik med prerezi je konstanten.

Na sliki 5.12 je prikazan primer 3D rekonstrukcije jajčnih mešičkov. Rekonstruirana sta dva mešička iz jajčnika, in sicer mešiček s slike 5.5 – manjši temnejši mešiček na sliki 5.12, ter velik mešiček na desni (glej prvo vrstico na sliki 5.6 – velik najbolj ovalen mešiček na desni). Zaradi boljše preglednosti smo za rekonstrukcijo izbrali le dva mešička (vseh mešičkov v jajčniku je sedem). Pri rekonstrukciji smo uporabili kar standardne funkcije programskega orodja Matlab (funkcije: `isosurface, patch, view, camlight, lighting phong`), pri tem smo predpostavili, da so prerezi vzporedni. Posebne pozornosti tudi nismo posvečali določanju normal na površino, ampak smo uporabili privzete vrednosti orodja Matlab. To je tudi eden izmed razlogov, da rekonstruirana mešička nista gladka. Vidna je tudi napaka (stopnice), ki je nastala, ker smo predpostavili vzporednost med prerezi. Opazimo pa lahko tudi rekonstrukcijsko napako, ki je lepo vidna ob vrhovih mešičkov, kjer manjši prerezi niso bili razpoznani, zato je mešiček prisekan. Dejstvo bi se dalo izkoristiti za boljše razpoznavanje ob robovih. Dejanski prerez skozi jajčnik z obema rekonstruiranima in vsemi ostalimi mešički lahko vidimo v prvi vrstici na sliki 5.6.



Slika 5.12: 3D rekonstrukcija mešičkov. Uporabljena je predpostavka, da so prerezi vzporedni. Dejanski prerez skozi jajčnik z obema rekonstruiranima in vsemi ostalimi mešički lahko vidimo v prvi vrstici na sliki 5.6.

Rekonstrukcija, ki jo opravimo ob takšnih predpostavkah, je seveda približek dejanskega stanja v jajčniku. Daje nam zgolj grob pogled v jajčnik. Za verodostojno 3D rekonstrukcijo pa moramo upoštevati tudi način, po katerem se tvorijo prerezi skozi jajčnik. To prinaša v samo rekonstrukcijo nove probleme. Tudi ob sto-odstotni uspešnosti in kvaliteti razpoznavanja mešičkov v prerezih ostane problem rekonstrukcije še vedno izjemno velik. Kvalitetno razpoznavanje samo po sebi še ne omogoča 3D rekonstrukcije. Treba je namreč tudi upoštevati, da se prerez mešička med dvema zaporednima slikama vedno malo premakne in deformira. Ob tem pa seveda ostaja problem, da pri klasičnih 2D ultrazvočnih napravah kotni razmik med prerezi ni znan. Pri takšni 3D rekonstrukciji torej ves čas operiramo s približki za posamezne količine, zato lahko rezultatom le delno zaupamo.

6.

Sklep

V disertaciji smo se ukvarjali z avtomatskim razpoznavanjem objektov iz zaporedja dvodimenzionalnih posnetkov. V 2. poglavju je nastal podroben pregled obstoječih metod razpoznavanja iz zaporedja slik. Smiselno smo jih razdelili v pet področij, in sicer v: a) metode za analizo gibanja v slikah, b) segmentacijske metode na osnovi gibanja, c) aktivne krivulje, d) tehnike pri 3D slikah in e) metode prediktor-korektor. Posamezne metode smo kritično ovrednotili in predlagali izboljšave za problem razpoznavanja iz zaporedja slik, kjer ni na voljo nikakršnih vnaprejšnjih informacij.

Raziskavo smo nadaljevali s konstrukcijo dveh novih razpoznavalnih sistemov za avtomatsko razpoznavanje objektov iz zaporedja dvodimenzionalnih posnetkov. Prvi sistem – algoritem za razpoznavanje objektov v statičnih 2D slikah – smo podrobno opisali v 3. poglavju, drugega – algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije – pa v 4. poglavju. V aplikativnem delu smo oba razvita postopka preizkusili na zaporedju umetnih in realnih ultrazvočnih posnetkov jajčnikov. V podpoglavju 3.2 smo zato razvili simulator za tvorjenje umetnih ultrazvočnih posnetkov jajčnikov na osnovi vpeljanega 3D modela jajčnika z mešički. V literaturi nismo zasledili podobnega simulatorja, tudi ne 3D modela jajčnika, in lahko zato domnevamo, da je naš simulator prvi s tega področja.

Kot že rečeno, je algoritem 2, opisan v podpoglavju 3.3, namenjen za razpoznavanje temnih ovalnih objektov na eni statični 2D sliki. Ob rasti regije, uporabljeni za segmentacijsko metodo, vpeljuje pri razpoznavanju še množico statističnih testov. V končni obliki je algoritem namenjen za razpoznavanje jajčnih mešičkov, zato je pri posameznih korakih vanj vključeno znanje o ultrazvočnih slikah, jajčnikih in mešičkih. Algoritem smo verificirali na umetnih in realnih ultrazvočnih posnetkih jajčnikov. V umetnih slikah so bile vse razmere točne znane, na realnih posnetkih pa smo uspešnost metode merili na osnovi odčitkov, ki so jih posredovali zdravniki specialisti. Dobljeni rezultati so boljši od doslej znanih in primerljivih (podpoglavje 3.4). Odstotek razpoznavanja je na umetnih slikah okrog 90 %, na realnih pa okrog 78 %. V izračunih smo upoštevali vse mešičke enakovredno, tako majhne kot velike. V literaturi najdemo že kar nekaj algoritmov za razpoznavanje jajčnih mešičkov na eni 2D ultrazvočni sliki [PZK97, PVZ98, VDZ98, KS98, SLSP98, CZ00]. Na osnovi dobljenih rezultatov lahko naš algoritem uvrstimo tako po odstotku razpoznavanja kakor po kvaliteti rezultatov ob bok najboljšemu algoritmu do sedaj [KS98]. Treba pa je še poudariti, da je slednji polavtomatski, naš algoritem pa je popolnoma avtomatski.

Če primerjamo algoritem 2 po zgradbi z drugimi podobnimi algoritmi iz literature, ne najdemo popolnega ujemanja. Korak predobdelave je le do določene mere podoben korakom algoritma iz [KS98], prav tako pa tudi korak poobdelave z algoritmom iz [PZK97]. Oba omenjena koraka našega algoritma pa imata dodanih toliko novih funkcij, da ju lahko upravičeno štejemo kot popolnoma nova. Osnovno idejo izvedbe rasti regije smo povzeli po [Gam93], a smo jo modificirali tako, da smo rast nadzorovali z množico statističnih testov, vpeljali pa smo tudi obteženi gradient na osnovi tekstur. Takšne izvedbe rasti regij prav tako nismo zasledili nikjer v literaturi.

V 4. poglavju smo naredili korak naprej, in sicer smo prešli od razpoznavanja objektov v posameznih statičnih 2D slikah, k razpoznavanju, kjer upoštevamo celotno informacijo iz zaporedja slik. Na začetku poglavja smo najprej pokazali na pomanjkljivosti, ki jih imajo algoritmi za statične 2D slike, preneseni v obdelavo zaporedja slik. Po natančni matematični formulaciji problema smo dokazali, da je najoptimalneje, če zgodovino razpoznavanja na prejšnjih slikah vgradimo v Kalmanov filter in jo kot tako uporabimo pri razpoznavanju na trenutni sliki. V nadaljevanju smo podrobno opisali novi algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Sestavni del algoritma je Kalmanov filter. Merilni sistem smo realizirali z algoritmom za statične 2D slike (algoritem 2), ki vrača meritve, potrebne za delovanje Kalmanovega filtra – v našem primeru je meritev isto kot razpoznani objekt. Na osnovi meritve v začetni sliki vzpostavimo model objekta, tako da vsaki točki meje razpoznanega objekta priredimo stanje sistema Kalmanovega filtra. Ta model zatem s Kalmanovim filtrom na osnovi meritev iz slike v sliko modificiramo. Izračunani model na posamezni sliki pa določa nov najboljši približek iskanega objekta. Ob konstrukciji algoritma smo zasledili nekaj specifičnih situacij, ki negativno vplivajo na uspešnost razpoznavanja. Te primere smo detajlno študirali, podali grobe napotke za odpravljanje teh problemov, za primer razpoznavanja jajčnih mešičkov pa smo uvedli konkretne rešitve. Tako smo ob osnovni varianti algoritma dogradili še tri izboljšave. S prvo smo odpravili negativen vpliv, ki ga prinašajo velike napake v meritvah. Z drugo izboljšavo smo v algoritmu upoštevali lokalnost obravnavanega procesa (vsaka točka ima ob globalnih še lokalne lastnosti), v zadnji, tretji izboljšavi, pa smo podali rešitev za primer, ko v merilnem sistemu odkrijemo sistemsko napako (meritev konstantno podcenjena). Na koncu poglavja smo se ukvarjali še z dvema problemoma, in sicer: a) kako odpraviti prehodni pojav, ki se pojavi na začetku zaporedja, ker modelni parametri Kalmanovega filtra še niso dovolj natančno naučeni, b) kako znižati veliko število napačno razpoznanih objektov, ki se pojavijo skozi zaporedje. Za omenjena problema smo našli rešitev ter jo zatem integrirali v končno obliko algoritma (algoritem 7).

Algoritem na osnovi postopkov predikcije smo prav tako preizkusili na zaporedju umetnih in realnih ultrazvočnih posnetkov jajčnikov z mešički. Dobljene rezultate smo prikazali v 5. poglavju. Podali smo tudi primerjavo dobljenih rezultatov z rezultati algoritma za statične 2D slike. Za vrednotenje rezultatov smo uporabili že vpeljane kvantitativne mere. Po primerjavi rezultatov se je izkazalo, da so rezultati algoritma 7 veliko bolj kompaktni in natančni kot rezultati algoritma 2. Posebej je to bilo očitno pri razpoznavanju v realnih razmerah, torej pri realnih ultrazvočnih slikah. Največje izboljšanje, ki smo ga dosegli po razpoznavanju objektov iz zaporedja slik s postopkom predikcije, pa je vsekakor občutno zmanjšanje števila napačno razpoznanih mešičkov. Ob upoštevanju zgodovine razpoznavanja na predhodnih slikah se je odstotek napačno razpoznanih zmanjšal tudi do 80 % glede na začetne vrednosti. Izkazalo pa se je tudi, da s takšnim obdelovanjem zaporedja slik povečamo gotovost v dobljene rezultate. Algoritem 7 je v nekaterih težavnih situacijah (izredno šumne slike) detektiral mešičke pravilneje kot pa zdravnik ekspert.

Če želimo naš algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije umestiti med dela s sorodnega področja, naletimo na določene težave. V literaturi namreč nismo zasledili nobenega algoritma, ki bi razpoznaval jajčne mešičke iz zaporedja 2D ultrazvočnih posnetkov. Torej lahko sklepamo, da je naš algoritem prvi s tega področja. Tudi po zgradbi ga težko primerjamo z drugimi algoritmi. Prav tako v literaturi nismo nikjer zasledili, da bi bil Kalmanov filter uporabljen pri razpoznavanju objektov na takšen način. Ponavadi Kalmanov filter pri obdelavi slik uporabljamo za določanje preiskovalnega območja pri iskanju ujemanja med dvema zaporednima slikama. V teh primerih gre zgolj za sledenje težišču objekta iz slike v sliko. Na osnovi predikcije predvidimo težišče v naslednji sliki, s tem pa določimo obseg preiskovalnega območja. V našem algoritmu sledimo množici povezanih točk iz slike v sliko (sledimo krivulji). Naš algoritem je kombinacija več Kalmanovih filtrov. Sestavljen je iz treh Kalmanovih filtrov, in sicer iz globalnega Kalmanovega filtra, katerega vrednosti elementov matrike šuma $\mathbf{Q}$ v sistemu in matrike šuma $\mathbf{R}$ v merilnem sistemu določata preostala dva Kalmanova filtra. Vhod v algoritem je zaporedje odčitkov objekta na vsaki sliki (ali za več objektov), ki jih dobimo z nekim algoritmom za razpoznavanje na statičnih 2D slikah. Ob tem ugotavljamo, da meritev zgolj na eni 2D sliki ne more biti nikoli popolnoma točna.

V tem delu nam je torej do določene mere uspelo povezati področji digitalne obdelave slik in razpoznavanja vzorcev s področjem teorije sistemov oz. kontrolne teorije.

Po načinu delovanja algoritma 7 najdemo še največjo podobnost z delovanjem aktivnih krivulj oz. kač in Kalmanovih kač pri obdelovanju zaporedja slik (podpoglavje 2.4). V vseh treh postopkih modificiramo krivuljo iz slike v sliko, in sicer vsako točko s krivulje posebej. Aktivne krivulje in Kalmanove kače sta izrazito parametrična postopka. Za svoje delovanje potrebujeta natančno določene parametre α, β in γ, ki pa jih je v praksi avtomatsko izredno težko določiti. Naš algoritem pa za svoje delovanje ne potrebuje posebnih parametrov – pomeni neparametričen postopek. Na posamezni sliki je edini vhod v naš algoritem krivulja (meritev), ki približno določa iskani objekt. Začetni približek iskanega objekta, ki mora biti ob tem še zelo natančno določen, zahtevata tudi postopka aktivnih krivulj in Kalmanovih kač. Vsi parametri, ki jih naš algoritem potrebuje, so vgrajeni v model sistema in se od slike do slike s Kalmanovim filtrom samodejno ažurirajo. Tudi ni potrebna nobena ročna inicializacija postopka. Gre za samodejno avtomatsko inicializacijo vseh parametrov na osnovi približka objekta iz prve slike. Aktivne krivulje oz. Kalmanove kače sta neke vrste optimizacijska postopka, ki iščeta minimum energijskega funkcionala. Ta funkcional je treba specificirati od primera do primera. V našem algoritmu pa je optimizacijski postopek transparenten, skrit znotraj Kalmanovega filtra, zato ni treba specificirati posebnih energijskih členov. Iz vsega povedanega vidimo, da gre kljub vsemu za popolnoma različne postopke.

V delu smo torej vpeljali dva avtomatska postopka za razpoznavanje objektov iz zaporedja slik, in sicer prvega za statične 2D slike (algoritem 2), drugega pa na osnovi postopkov predikcije (algoritem 7). Oba algoritma sta v posameznih korakih prirejena za razpoznavanje jajčnih mešičkov v ultrazvočnih slikah. Sta pa zasnovana popolnoma modularno, tako da ju je možno preprosto prirediti za razpoznavanje na kakršnemkoli drugem področju – drugem tipu zaporedij slik.

Za raziskovanje v prihodnosti ostaja še delno odprto vprašanje sledenja izginjajočim objektom. V našem delu smo sicer identificirali takšen objekt, vendar pa smo ga obravnavali kot dva različna, in sicer kot prvega, preden je poniknil, in kot drugega, ko se je ponovno pojavil. V takih primerih je treba določiti le ujemanje. Drugi odprt problem pa je 3D rekonstrukcija iz zaporedja prerezov z odčitki objektov, pri čemer ni na voljo podatkov o dejanskem položaju prerezov v 3D prostoru. Tudi to je možna aplikacija zaporedja 2D ultrazvočnih posnetkov jajčnika z mešički.

V uvodnem poglavju smo postavili tezo, ki smo jo skozi disertacijo skušali dokazati. V 4. poglavju smo razvili algoritem za razpoznavanje objektov iz zaporedja slik s postopki predikcije. Algoritem razpoznava objekte na osnovi modela, ki ga tvorimo v odvisnosti od področja problema (iskanega objekta) in tipa slik. Model iterativno modificiramo skozi zaporedje 2D slik. Rezultati razpoznavanja so pokazali (poglavje 5), da se je uspešnost in natančnost razpoznavanja bistveno izboljšala glede na razpoznavanje z algoritmom za statične 2D slike. Pokazali smo, da se s takšnim obdelovanjem izboljša zaupanje v dobljene rezultate (dosegli smo občutno zmanjšanje števila napačno razpoznanih objektov) ter poveča kvaliteta razpoznanih objektov (večja kompaktnost objektov). Na osnovi teh rezultatov trdimo, da je teza dokazana.

Literatura

[ADPM] V. Awasthi, K. W. Doolittle, G. Parulkar, J. G. McNally, "Cell tracking using a distributed algorithm for 3D image segmentation", http://tyrone.wustl.edu/keith/papers/tracking.html.

[App] M. Applebaum, "Ultrasound in infertility WEB book", http://www.drapplebaum.com/art.htm.

[BBAT97] G. D. Borshukov, G. Bozdagi, Y. Altunbasak, A. M. Tekalp "Motion segmentation by multistage affine classification", *IEEE Transaction on image processing*, vol. 6, št. 11, str. 1591–1594, 1997.

[BH97] R. G. Brown, P. Y. Hwang, *Introduction to random signals and applied Kalman filtering.* New York: John Willey and Sons, 1997.

[Bov88] A. C. Bovik, "On detecting edges in speckle imagery", *IEEE Transactions on acoustics, speech and signal processing*, vol. 36, št. 10, str. 1618–1627, 1988.

[BRM97] K. J. Bradshaw, I. D. Reid, D. W. Murray, "The active recovery of 3D motion trajectories and their use in prediction", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 19, št. 3, str. 219–234, 1997.

[BS78] J. N. Bronštejn, K. A. Semendjajev, *Matematični priročnik.* Ljubljana: Tehniška založba Slovenije, 1978.

[Bur93] J. B. Burl, "A reduced order extended Kalman filter for sequential images containing a moving object", *IEEE Transaction on image processing*, vol. 2, št. 3, str. 285–295, 1993.

[Cas96] K. R. Castleman, *Digital image processing.* Englewood Cliffs: Prentice Hall, 1996.

[CK97] V. Chalana, Y. Kim, "Evaluation of boundary detection algorithms on medical images", *IEEE Transaction on medical imaging*, vol. 16, št. 5, str. 642–652, 1997.

[CN96] W. G. Chen, N. Nandhakumar, “A simple scheme for motion boundary detection”, *Pattern recognition*, vol. 29, št. 10, str. 1689–1701, 1996.

[CZ00] B. Cigale, D. Zazula, “Segmentation of ovarian ultrasound images using cellular neural networks”, v: Ž. Čučej, P. Planinšič, B. Gergič, ur., *Proceedings of* 7^{th} *International workshop on systems, signals and image processing*, Maribor, Slovenija, junij, str. 33–36, 2000.

[DA00] J. S. Duncan, N. Ayache, “Medical image analysis: Progress over two decades and the challenges ahead”, *IEEE Transaction on pattern analysis and machine intelligence*, vol. 22, št. 1, str. 85–106, 2000.

[DBB95] P. Delagnes, J. Benois, D. Barba, “Active contours approach to object tracking in image sequences with complex background”, *Pattern recognition letters*, vol. 16, št. 2, str. 171–178, 1995.

[DC92] J. H. Duncan, T.-C. Chou, “On the detection of motion and the computation of optical flow”, *IEEE Transaction on pattern analysis and machine intelligence*, vol. 14, št. 3, str. 346–352, 1992.

[dMUT90] E. de Michelli, S. Uras, V. Torre, “The analysis of time varying image sequences”, v: O. Faugeras, ur., *Proceedings of* 1^{st} *European conference on computer vision*, Springer-Verlag, str. 595–597, 1990.

[DRM] K. W. Doolittle, I. Reddy, J. G. McNally. “3D analysis of cell movement during normal and myosin-II-null cell morphogenesis in dictyostelium”, http://tyrone.wustl.edu/keith/papers/myosin.html.

[EM96] N. Ezquerra, R. Mullick, “Knowledge-guided segmentation of 3D imagery”, *Graphical models and image processing*, vol. 58, št. 6, str. 510–523, 1996.

[EN96] A. N. Evans, M. S. Nixon, “Biased motion-adaptive temporal filtering for speckle reduction in echocardigraphy”, *IEEE Transaction on medical imaging*, vol. 15, št. 1, str. 39–50, 1996.

[Eul] —–, “Definicije Eulerjevih kotov”, http://www.astro.virginia.edu/∼eww6n/math/EulerAngles.html.

[Gam93] JP. Gambotto, “A new approach to combining region growing and edge detection”, *Pattern recognition letters*, vol. 14, str. 869–875, 1993.

[GKN+74] A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, A. A. Sutherland, *Applied optimal estimation.* London: MIT press, 1974.

[GL96] W. F. Gardner, D. T. Lawton, "Interactive model-based vehicle tracking", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 18, št. 11, str. 1115–1121, 1996.

[GNVT95] M. A. Gore, P. L. Nayudu, V. Vlaisavljevic, N. Thomas. "Prediction of ovarian cycle outcome by follicular characteristics, stage 1", *Human reproduction*, vol. 10, str. 2313–2319, 1995.

[God85] J. W. Godman, *Statistical optics.* New York: John Willey and Sons, 1985.

[GvBSD97] J. D. Geeter, H. van Brussel, J. D. Schutter, M. Decreton, "A smoothly constrained Kalman filter", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 19, št. 10, str. 1171–1177, 1997.

[HA92] I. L. Herlin, A. Ayache. "Features extraction and analysis methods for sequences of ultrasound images", *Image and vision computing*, vol. 10, št. 10, str. 673–682, 1992.

[HNR92] D. P. Huttenlocher, J. J. Noh, W. J. Rucklidge, *Tracking non-rigid objects in complex scenes.* Tehniško poročilo št. TR-92-1320, Computer science department, Cornell University, 1992.

[HNR93] D. P. Huttenlocher, J. J. Noh, W. J. Rucklidge, "Tracking non-rigid objects in complex scenes", *Proceedings of Fourth International Conference on Computer Vision*, Berlin, Nemčija, str. 93–101, 1993.

[Hos95] R. Hosie, *Predicting ball trajectories in uncontrolled environments.* Diplomsko delo, Curtin university of technology, 1995.

[IRP92] M. Irani, B. Rousso, S. Peleg, "Detecting and tracking multiple moving objects using temporal integration", v: G. Sandini, ur., *Proceedings of Computer Vision-ECCV '92*, Santa Margherita Ligure, Italija, str. 282–287, 1992.

[IRP97] M. Irani, B. Rousso, S. Peleg. "Recovery of ego-motion using region alignment", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 19, št. 3, str. 268–272, 1997.

[IT93] P. Iliev, L. Tsekov, "Motion detection using image histogram sequence analysis", *Signal processing*, vol. 30, št. 3, str. 373–384, 1993.

[J93] B. Jähne, *Digital image processing.* Berlin: Springer-Verlag, 1993.

[Jen] J. A. Jensen, "Programsko orodje field", http://www.it.dtu.dk/~jaj/field.

[Jen91] J. A. Jensen, "A model for propagation and scattering of ultrasound in tissue", *Journal of acoustics society America*, vol. 19, str. 182–191, 1991.

[Jen96] J. A. Jensen, "Semiautomated segmentation of ovarian follicular ultrasound images using a knowledge-based algorithm", *Medical* & *biological engineering* & *computing*, vol. 34, št. 1, str. 351–353, 1996.

[JKC97] D. S. Jang, G. Y. Kim, H. I. Choi, "Model-based tracking of moving object", *Pattern recognition*, vol. 30, št. 6, str. 999–1008, 1997.

[JM93] M. Joliet, B. M. Mazoyer, "Three-dimensional segmentation and interpolation of magnetic resonance brain images", *IEEE Transactions on Medical Imaging Graphics*, vol. 12, št. 2, str. 269, 1993.

[JM97] J. A. Jensen, P. Munk, "Computer phantoms for simulating ultrasound B-mode and CFM images", *Proceedings of* 23^{rd} *Acoustical imaging symposium*, Boston, ZDA, 1997.

[JS92] J. A. Jensen, N. B. Svendsen, "Calculation of pressure field from arbitrary shaped, apodized and excited ultrasound transducers", *IEEE Transaction on ultrasonics, ferroelectrics and frequency control*, vol. 39, str. 262–267, 1992.

[KMPS94] C. Kotropoulos, X. Magnisalis, I. Pitas, M. G. Strintzis, "Nonlinear ultrasonic image processing based on signal-adaptive filters and self-organizing neural networks", *IEEE Transaction on image processing*, vol. 3, št. 1, str. 65–77, 1994.

[KP92] C. Kotropoulos, I. Pitas. "Optimum nonlinear signal detection and estimation in the presence of ultrasonic speckle", *Ultrasonic imaging*, vol. 14, str. 249–275, 1992.

[KS98] A. Krivanek, M. Sonka. "Ovarian ultrasound image analysis: Follicle segmentation", *IEEE Transaction on medical imaging*, vol. 17, št. 6, str. 935–944, 1998.

[KSM95] I. Kokkinidis, M. G. Strintzis, N. Maglaveras, "Motion estimation in ultrasound b-mode images using a maximum likelihood matching criterion", *Proceedings of Canadian medical and biological engineering conference*, Montreal, Canada, str. 20–24, 1995.

[LJ89] S. P. Liou, R. C. Jain, "Motion detection in spatio-temporal space", *Computer vision, graphics, and image processing*, vol. 45, št. 2, str. 227–250, 1989.

[LOG99] B. Levienaise-Obadia, A. Gee, "Adaptive segmentation of ultrasound images", *Image and vision computing*, vol. 17, str. 583–588, 1999.

[LY90] W. Long, Y.-H. Yang, "Stationary background generation: an alternative to the difference of two images", *Pattern recognition*, vol. 23, št. 12, str. 1351–1359, 1990.

[May94] P. S. Maybeck, *Stochastic models, estimation and control: Volume 1.* Arlington: Navtech Book & Software Store, 1994. Original izdan pri Academic Press, 1979.

[MBC94] A. Mecocci, F. Bartolini, V. Cappellini, "Image sequence analysis for counting in real time people getting in and out of a bus", *Signal processing*, vol. 35, št. 2, str. 105–116, 1994.

[MCD90] J. Mukherjee, B. N. Chatterji, P. P. Das, "Segmentation of three-dimensional surfaces", *Pattern recognition letters*, vol. 11, št. 3, str. 215–223, 1990.

[MDR91] O. Monga, R. Deriche, J.-M. Rocchisani, "3D edge detection using recursive filtering: application to scanner images", *Computer vision, graphics, and image processing: Image understanding*, vol. 53, št. 1, str. 76–87, 1991.

[Meh98] R. Mehrotra, "Establishing motion-based feature point correspondence", *Pattern recognition*, vol. 31, št. 1, str. 23–30, 1998.

[MKT98] I. Mikic, S. Krucinski, J. D. Thomas, "Segmentation and tracking in echocardiographic sequences–active contours guided by optical flow estimates", *IEEE Transaction on medical imaging*, vol. 17, št. 2, str. 274–284, 1998.

[MS99] S. Malassiotis, M. G. Strintzis, "Tracking the left ventricle in echocardiographic images by learning heart dynamics", *IEEE Transaction on medical imaging*, vol. 18, št. 3, str. 282–290, 1999.

[MSV95] R. Malladi, A. Sethian, B. C. Vemuri, "Shape modelling with front propagation: A level set approach", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 17, št. 2, str. 158–175, 1995.

[NCRT98] A. Neri, S. Colonnese, G. Russo, P. Talone, "Automatic moving object and background separation", *Signal processing*, vol. 66, št. 2, str. 219–232, 1998.

[NSKO94] H.-H. Nagel, G. Socher, H. Kollnig, M. Otte, "Motion boundary detection in image sequences by local stochastic tests", v: J. O. Eklundh, ur., *Proceedings ECCV '94*, Stockholm, Švedska, str. 305–315, 1994.

[ON95] M. Otte, H.-H. Nagel, "Estimation of optical flow based on higher-order spatiotemporal derivates in interlaced and non-interlaced image sequences", *Artificial intelligence*, vol. 78, str. 5–43, 1995.

[PD00] N. Paragios, R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 20, št. 3, str. 266–281, 2000.

[Pet99] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 21, št. 6, str. 564–569, 1999.

[Pot98] B. Potočnik, *Uporaba segmentacije pri analizi medicinskih slik.* Magistrsko delo, Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza Maribor, 1998.

[PVZ98] B. Potočnik, B. Viher, D. Zazula, "Computer-assisted detection of ovarian follicles based on ultrasound images", *A számítástechnika orvosi és biológiai alkalmazásai*, Veszprem, Madžarska, str. 24–34, 1998.

[PZ98] B. Potočnik, D. Zazula, "Aktivne krivulje – nadgradnja razpoznavalnih sistemov", *Zbornik sedme Elektrotehniške in računalniške konference ERK 98*, Portorož, Slovenija, zvezek B, str. 249–252, 1998.

[PZK97] B. Potočnik, D. Zazula, D. Korže. "Automated computer-assisted detection of follicles in ultrasound images of ovary", *Journal of medical systems*, vol. 21, št. 6, str. 445–457, 1997.

[Rao96] R. P. N. Rao, *Robust Kalman filters for prediction, recognition and learning.* Tehniško poročilo št. TR-645, Computer science department, University of Rochester, 1996.

[RT94] C. S. Regazzoni, A. Tesei, "Density evaluation and tracking of multiple objects from image sequences", *Proceedings of ICIP '94*, Austin, ZDA, str. 545–549, 1994.

[Rus95] J. C. Russ, *The image processing handbook.* London: CRC Press, 1995.

[RZ89] A. Repše, D. Zazula, *Teorija in uporaba matematične statistike.* Velenje: Izobraževalni center Gorenje, 1989.

[SG91] K. C. Strasters, J. J. Gerbrands, "Three-dimensional image segmentation using a split, merge and group approach", *Pattern recognition letters*, vol. 12, št. 5, str. 307–325, 1991.

[SHB94] M. Sonka, V. Hlavac, R. Boyle, *Image processing, analysis and machine vision.* London: Chapman and Hall, 1994.

[SK97] M.G. Strintzis, I. Kokkinidis, "Maximum-likelihood motion estimation in ultrasound image sequences", *IEEE Signal processing letters*, vol. 4, št. 6, str. 156–157, 1997.

[SLQ90] S. Song, M. Liao, J. Quin, "Multiresolution image motion detection and displacement estimation", *Machine vision and applications*, vol. 3, št. 1, str. 17–20, 1990.

[SLSP98] G. E. Sarty, W. Liang, M. Sonka, R. A. Pierson, "Semiautomated segmentation of ovarian follicular ultrasound images using a knowledge-based algorithm", *Ultrasound in medicine and biology*, vol. 24, št. 1, str. 27–42, 1998.

[SMH+97] Y. Sato, M. Moriyama, M. Hanayama, H. Saito, S. Tamura, "Acquiring 3D models of non-rigid moving objects from time and viewpoint varying image sequences: A step toward left ventrincle recovery", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 19, št. 3, str. 253–258, 1997.

[Tis96] M. Tistarelli, "Multiple constraints to compute optical flow", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 18, št. 12, str. 1243–1250, 1996.

[TK96] Y. L. Tang, R. Kasturi, "Tracking moving objects during low altitude flight", *Machine vision and applications*, vol. 9, št. 1, str. 20–31, 1996.

[TS92] D. Terzopoulos, R. Szelinski, "Tracking with Kalman snakes", v: A. Blake, A. Yuille, ur., *Active vision*, MIT Press, str. 3–20, 1992.

[TV98] E. Trucco, A. Verri, *Introductory techniques for 3-D computer vision.* London: Prentice Hall, 1998.

[VCP95] C. Vieren, F. Cabestaing, J.-G. Postaire, "Catching moving objects with snakes for motion tracking", *Pattern recognition letters*, vol. 16, št. 7, str. 679–685, 1995.

[VDZ98] B. Viher, A. Dobnikar, D. Zazula, "Cellular automata and follicle recognition problem and possibilities of using cellular automata for image recognition purposes", *International journal of medical informatics*, vol. 49, št. 2, str. 231–241, 1998.

[WA94] J. Y. A. Wang, E. H. Adelson, "Representing moving images with layers", *IEEE Transaction on image processing*, vol. 3, št. 5, str. 625–638, 1994.

[WB95a] H. Wang, M. Brady, "Real-time corner detection algorithm for motion estimation", *Image and vision computing*, vol. 13, št. 9, str. 695–703, 1995.

[WB95b] G. Welch, G. Bishop, *An introduction to the Kalman filter.* Tehniško poročilo št. TR-95-041, Chapel Hill: Computer science department, University of North Carolina, 1995.

[WEHK96] M. Wang, J. Evans, L. Hassebrook, C. Knapp, "A multistage, optimal active contour model", *IEEE Transaction on image processing*, vol. 5, št. 11, str. 1586–1591, 1996.

[WYP97] N. Weng, Y. H. Yang, R. Pierson, "Three-dimensional surface reconstruction using optical flow for medical imaging", *IEEE Transaction on medical imaging*, vol. 16, št. 5, str. 630–641, 1997.

[YJK88] J. Ylä-Jääski, O. Kübler, "Segmentation and analysis of 3D volume images", *Proceedings of* 9^{th} *IAPR international conference on pattern recognition*, str. 951–953, 1988.

[YLFP98] F. Yeung, S. F. Levinson, D. Fu, K. J. Parker, "Feature-adaptive motion tracking of ultrasound image sequence using a deformable mesh", *IEEE Transaction on medical imaging*, vol. 17, št. 6, str. 945–956, 1998.

[ZFA97] J. Y. Zheng, Y. Fukaqawa, N. Abe, "3D surface estimation and model construction from specular motion in image sequences", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 19, št. 5, str. 513–520, 1997.

[ZM97] F. Ziliani, F. Moscheni, "Kalman filtering motion prediction for recursive spatio-temporal segmentation and object tracking", *Proceedings of Workshop on Image analysis for multimedia interactive services*, Louvain-la-Neuve, Belgija, str. 24–25, 1997.

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,
	1995–1998	Magistrski študij na Fakulteti za elektrotehniko, računalništvo in informatiko, Maribor,
	1998–2000	Doktorski študij na Fakulteti za elektrotehniko, računalništvo in informatiko, Maribor.
Zaposlitev:	1995–danes	Fakulteta za elektrotehniko, računalništvo in informatiko Maribor.