



Luka Čehovin

VIZUALNO SLEDENJE NETOGIM OBJEKTOM

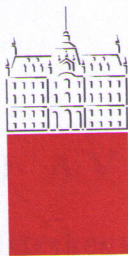


Magistrska naloga

Mentor: prof. dr. Aleš Leonardis

Ljubljana, 2010

Št.: 130-MAG-RI /2010
Datum: 1. 3. 2010



Luka Č E H O V I N, univ. dipl. inž. rač. in inf.

Ljubljana

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo magistrsko nalogo

Naslov naloge: **Vizualno sledenje netogim objektom**

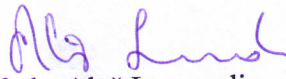
Visual tracking of non-rigid objects

Tematika naloge:

Vizualno sledenje objektom predstavlja pomembno podpodročje računalniškega vida. Algoritmi vizualnega sledenja so sposobni slediti določenemu objektu brez posebnega predznanja o tipu objekta, torej le na podlagi vizualne informacije, ki je pridobljena iz slik video posnetka. Ta ugodna lastnost omogoča širok spekter njihove uporabe, na primer pri interaktivnih vmesnikih med človekom in strojem, pri sledenju ljudi v urbanih prizorih, športu, itd. Vendar pa večji del takih algoritmov predpostavlja, da so opazovani predmeti togi ter, da se njihov izgled ne spreminja. Te predpostavke seveda zmanjšujejo zmožnosti uporabe vizualnega sledenja v splošnejših primerih, zato so novejša raziskave usmerjene v omilitev in odpravo teh predpostavk.

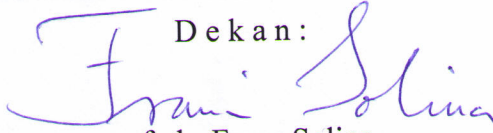
V okviru magistrske naloge proučite področje vizualnih sledilnikov, ki za doseganje robustnejšega sledenja objektom povezujejo lokalno in globalno informacijo o izgledu objekta. Pri tem se osredotočite na algoritme za vizualno sledenje netogim artikuliranim objektom, torej objektom, ki so sestavljeni iz več medsebojno povezanih togih delov. Izpostavite probleme obstoječih sledilnikov ter raziščite možnosti nadaljnjih izboljšav z novimi pristopi. Proučite možnosti vpeljave sprotnega učenja večih modalnosti pri opisu izgleda objekta ter vpliv tega na izboljšanje sledenja. V okviru magistrske naloge predlagane algoritme implementirajte ter jih z ustreznimi eksperimenti ovrednotite glede na obstoječe sledilnike na tem področju.

Mentor:


prof. dr. Aleš Leonardis



Dekan:


prof. dr. Franc Solina

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov magistrske naloge je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Zahvala

V prvi vrsti se zahvaljujem ljudjem, ki so bili v izdelavo te magistrske naloge neposredno vpleteni, mentorju prof. dr. Alešu Leonardisu in dr. Mateju Kristanu, za komentarje in napotke, ki so moje začetne korake v svet raziskav na področju računalniškega vida ter razpoznavanja vzorcev usmerjali v pravo smer. Zahvaljujem se sestri, Urši Čehovin, za lektorski pregled magistrske naloge. Zahvaljujem se tudi članom Laboratorija za umetne vizualne spoznavne sisteme za dobro družbo v preteklih dveh letih in pol ter družini za pomoč in podporo na moji učni poti, pa tudi naspluh v življenju.

Kazalo

1	Uvod	1
2	Pregled področja	5
2.1	Optimizacija ujemalne funkcije	5
2.1.1	Omejevanje ranga	5
2.1.2	Jedrne funkcije	6
2.1.3	Vzorčenje optimizacijske funkcije	10
2.1.4	Iskanje ujemanja stabilnih regij	12
2.2	Osveževanje modela izgleda	15
2.2.1	Izbira značilnic	15
2.2.2	Pravila jate	17
2.2.3	Barvna podobnost za osveževanje množice lokalnih značilnic	18
2.2.4	Stabilne regije	18
3	Dvoplastni vizualni model	21
3.1	Stanje vizualnega modela	22
3.2	Iskanje ujemanja množice lokalnih značilnic	24
3.2.1	Optimizacijska funkcija ujemanja lokalnih značilnic	24
3.2.2	Optimizacija z uporabo križne entropije	26
3.3	Osveževanje množice lokalnih značilnic	27
3.3.1	Posodabljanje uteži značilnic	28
3.3.2	Posodabljanje množice lokalnih značilnic	31
3.4	Posodabljanje globalne plasti	32
3.5	Sledilnik	36
3.5.1	Ocenjevanje dinamike objekta	36
3.5.2	Lokalne značilnice	38
3.5.3	Inicializacija sledilnika	39
4	Eksperimenti	41
4.1	Računska hitrost implementacije	41
4.2	Uporabljena slikovna zaporedja	42
4.3	Metodologija meritev	43
4.4	Lastnosti sledilnika	45
4.4.1	Parametri sledilnika	45
4.4.2	Dinamika množice lokalnih značilnic	47

4.4.3	Vpliv delov algoritma na sledenje	49
4.5	Primerjava z obstoječimi sledilniki	50
5	Diskusija	59
5.1	Globalnost in lokalnost	59
5.2	Vzporednice z Gestalt teorijo	61
5.3	Meje vizualnega sledenja	63
6	Zaključek	65
A	Metodologija	67
A.1	Metoda križne entropije	67
A.1.1	Kullback-Liebert-jeva divergenca	68
A.1.2	Uporaba v optimizaciji	68
A.1.3	Zvezna optimizacija	70
A.2	RANSAC	71
A.2.1	Določitev parametrov algoritma	72
A.3	Ocenjevanje gostote porazdelitve verjetnosti z jedri	73
A.4	Bayesov rekurzivni filter	75
A.4.1	Nelinearni filtri	75
A.4.2	Osnovni problem	77
A.4.3	Kalmanov filter	78
B	Uporabljeno programje	81
B.1	Odprtokodna orodja in knjižnice	81
B.2	Izvorna koda sledilnikov	81

Seznam slik

1.1	Ilustracija predstavitev vizualnega modela.	2
1.2	Preprost model človeškega telesa.	3
2.1	Primer sledenja z omejevanjem ranga.	7
2.2	Sledenje z jedrno funkcijo.	8
2.3	Ilustracija povezav med značilnicami.	9
2.4	Preprosti geometrijski model za sledenje obrazov.	10
2.5	Geometrijski model s povezavami na središče.	11
2.6	Povezave med blizu ležečimi objekti.	13
2.7	Izbira podmnožice ujemanj.	14
2.8	Iskanje ujemanja podgrafa.	15
2.9	Izbira značilnic za sledenje.	16
2.10	Prikaz sledenja sprotno osveževanim detektorjem na podlagi spodbujanja.	16
2.11	Ilustracija pravila ξ	18
2.12	Razbitje opisa objekta na zaplate.	19
2.13	Prikaz polja ujemanja.	20
3.1	Shematična predstavitev postopka osveževanja.	22
3.2	Prikaz določitve ϵ_t^i	25
3.3	Ilustracija izračuna variance razdalj med točkami dveh trajektorij.	29
3.4	Shematični prikaz uporabe barvnega modela.	33
3.5	Shematični prikaz uporabe modela gibanja.	35
3.6	Grafični prikaz komponent verjetnostne porazdelitve.	36
3.7	Prikaz začetne postavitve značilnic.	39
4.1	Pregled izbranih slik iz uporabljenih video zaporedij.	44
4.2	Rezultati za različne vrednosti parametra togosti λ_G	47
4.3	Prikaz sprememb velikosti množice značilnic v posameznih video zaporedjih.	48
4.4	Primerjava sledenja za zaporedje slik <i>PETS2000</i>	52
4.5	Primerjava sledenja za zaporedje slik <i>PETS2001-1</i>	53
4.6	Primerjava sledenja za zaporedje slik <i>PETS2001-2</i>	53
4.7	Primerjava sledenja za zaporedje slik <i>bicycle</i>	54
4.8	Primerjava sledenja za zaporedje slik <i>hand</i>	55
4.9	Primerjava sledenja za zaporedje slik <i>can</i>	56
4.10	Primerjava sledenja za zaporedje slik <i>diver</i>	56

SEZNAM SLIK

4.11	Primerjava sledenja za zaporedje slik <i>highjump</i>	57
4.12	Primerjava sledenja za zaporedje slik <i>torus</i>	57
5.1	Ilustracija problema združevanja modalnosti.	60
5.2	Primeri Gestalt zakonov.	62
A.1	Ilustracija Kullback-Liebert-jeve divergence.	68
A.2	Primer uporabe metode RANSAC.	73
A.3	Ilustracija ocene funkcije gostote verjetnosti za različne vrednosti h	76

Seznam tabel

4.1	Rezultati meritev hitrosti posameznih delov algoritma.	42
4.2	Pregled lastnosti uporabljenih video zaporedij.	45
4.3	Pregled statistike starosti značilnic za posamezna video zaporedja.	49
4.4	Pregled vpliva posameznih delov algoritma na sledenje.	49
4.5	Rezultati primerjave z obstoječimi sledilniki.	51

Seznam algoritmov

1	Postopek globalne optimizacije na podlagi metode križne entropije.	27
2	Postopek lokalne optimizacije na podlagi metode križne entropije.	28
3	Pregled celotne psevdokode sledilnika	37
4	Splošna metoda križne entropije.	70
5	Konsenz naključnih vzorcev.	72

Povzetek

V okviru magistrske naloge je obravnavano vizualno slednje netogih, artikuliranih objektov. Za sledenje takih objektov je potrebno vizualne modele, ki se navadno uporabljajo za opis togih objektov, razširiti tako, da se opis lahko prilagaja deformacijam objekta. V tem delu je predstavljena razširitev, ki temelji na hierarhičnem pristopu h konstrukciji vizualnega modela. Le-ta je osnovana na hierarhičnem združevanju lokalne in globalne vizualne informacije. Predstavljeni vizualni model nadgrajuje obstoječe modele, ki za opis objekta uporabljajo množico med seboj povezanih lokalnih značilnic. Le-ta predstavlja spodnjo plast opisanega vizualnega modela. Na podlagi lokalnih značilnic se med samim sledenjem gradi večmodalni globalni vizualni model izgleda, ki predstavlja zgornjo plast. Na podlagi teh informacij se določi območje v posamezni sliki zaporedja, ki pripada objektu, na podlagi tega pa se množica lokalnih značilnic tekom sledenja obnavlja z novimi značilnicami. V nalogi je najprej predstavljeno področje vizualnega sledenja netogim objektom, ki opisani vizualni model umesti v raziskovalni okvir. Na podlagi tega sta potem opisana sam vizualni model ter njegova integracija v preprost sledilnik. Delovanje sledilnika je empirično preizkušeno z več eksperimenti ter z uporabo devetih raznolikih video slikovnih zaporedij. Prednosti in slabosti sledilnika so prikazane v primerjavi rezultatov s tremi različnimi sodobnimi vizualnimi sledilniki. Magistrska naloga je zaključena z diskusijo na temo teoretičnih in praktičnih omejitev opisanega vizualnega modela ter idejami za nadaljnje delo.

Ključne besede: računalniški vid, vizualno sledenje, netogi objekti, barva, križna entropija, optimizacija, optični tok, RANSAC, ocenjevanje gostote jeder, Kalman-ov filter

Abstract

In this thesis we study the field of visual tracking of non-rigid, articulated objects. For this task a typical visual model, used mostly for the description of rigid objects has to be extended in a way that it can adapt to the deformations of such objects. In our work we present an extension that is based on a hierarchical approach towards visual model construction. It is based on a hierarchical combination of local and global visual information. The resulting visual model extends the existing visual models that use a set of local features connected with geometrical constraints. This set represents the bottom layer of the presented visual model. Using local features the visual model builds a multi-modal representation of the object's appearance that represents the top layer of the model. Based on this information an area of the object in a frame is determined, and based on this area, the local feature set is updated with new features. In the thesis our work is first placed into a research context by describing recent published work on visual tracking of non-rigid objects. Next, the proposed visual model is described in detail together with its integration in a simple tracker. The performance of the tracker is assessed in various experiments using nine different video sequences. Advantages and disadvantages of the tracker are shown in comparison of the tracker with three different state-of-the-art visual trackers. The thesis is concluded with a discussion in which some theoretical and practical limitations of the presented visual model are laid out as well as some ideas for further development.

Keywords: computer vision, visual tracking, non-rigid objects, color, cross entropy, optimization, optical flow, RANSAC, kernel density estimation, Kalman filter

Poglavje 1

Uvod

Vizualno sledenje (ang. *visual tracking*) oziroma sledenje na podlagi izgleda objekta v video zaporedju slik predstavlja pomembno raziskovalno področje v okviru računalniškega vida. Osnovna ideja vizualnega sledenja je, da v zaporedju slik z uporabo vizualnih lastnosti, kot so na primer barva, robovi v posamezni sliki, gibanje med dvema zaporednjima slikama itd. določimo položaj izbranega objekta. Za doseg tega izbrane vizualne lastnosti objekta združimo v t.i. *vizualni model*, ki povzema izgled objekta. Ker so vizualne lastnosti zelo splošne in izvirajo neposredno iz lastnosti slike, je tak opis zelo splošen ter široko uporaben. Vendar po drugi strani ta splošnost prinaša tudi določene probleme. Ker se izgled objekta skozi čas spreminja, se lahko statičen vizualni model, ki ga zgradimo ob začetku sledenja, z uporabo slikovnih informacij v prvi sliki zaporedja [1, 13, 27] kmalu izkaže za preveč togega, saj ne more zajeti teh sprememb. Če se izgled objekta torej preveč spremeni, lahko njegova podobnost z vizualnim modelom postane enaka ali celo nižja kot podobnosti vizualnega modela in ozadja (oziroma okolice). To pa posledično zmanjša robustnost nadaljnjega sledenja.

Ta problem rešimo tako, da vizualni model z vsako novo sliko v zaporedju popravimo oziroma prilagodimo glede na spremembe izgleda objekta [38, 47]. Tak pristop sicer reši problem opisne togosti vizualnega modela, vendar pa vnese v sledenje nov problem, znan kot drsenje (ang. *drifting*). Drsenje nastopi, ko je vizualni model posodobljen z (delno) napačnimi podatki. To se lahko zgodi, ker sledilnik napačno napove nov položaj objekta. S tem v podatke, ki se uporabijo za popravljanje vizualnega modela, vnese tudi podatke o okolici. Druga možnost za nastop drsenja pa je, da vizualni model vključuje samo vizualne lastnosti, ki so v dani sliki dvomne (na primer lastnosti, ki ne ločijo objekta od okolice). Drsenje se ponavadi verižno stopnjuje do točke, ko sledilnik odpove oziroma "zdrsne" z objekta. To se največkrat zgodi brez opaznega zmanjšanja ujemanja med modelom in sliko, saj se model sproti prilagaja informaciji v sliki.

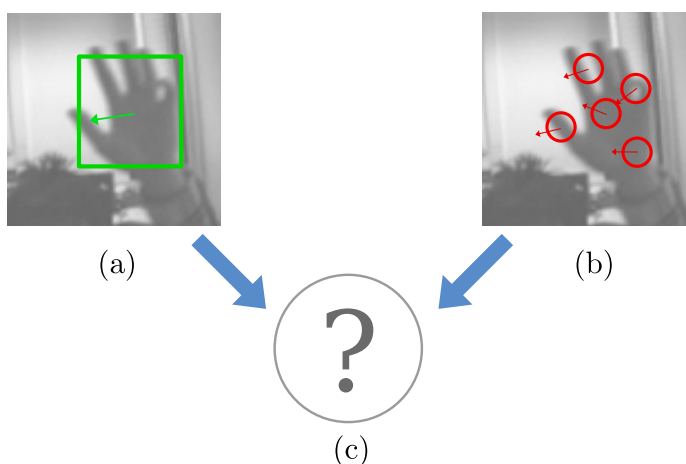
Za uspešno premostitev problema drsenja je potrebno pri načrtovanju vizualnega modela odgovoriti na dve temeljni vprašanji [53]:

- Kako predstaviti izgled objekta tako, da ga bo mogoče uspešno ločevati od ozadja in okolice ter

- Kako med slikovnim zaporedjem popravljati opis izgleda, da bo le-ta ostal konsistenten z dejanskim izgledom objekta?

Obe vprašanji se navezujeta eno na drugo, saj formalizacija opisa izgleda pogojuje način njegovega posodabljanja, posodabljanje pa seveda vpliva na dolgoročno uspešnost predstavitve.

Če pogledamo na problem še s stališča ravni podrobnosti (ang. *level of detail*), imamo na voljo dva pola, med katerima je potrebno najti kompromis (glej Sliko 1.1). V okviru prvega pola obravnavamo izgled objekta z enim *celostnim* oziroma globalnim, vizualnim modelom, ki opisuje celoten objekt. Tak način opisa je zaradi navidezne preprostosti bolj obširno obravnavan [1, 13, 27, 38, 47]. Vendar pa ima slabost, da se ob njegovem posodabljanju vanj zlahka vnesejo napake, ki izvirajo iz napačnega opisa dela izgleda. Ker je v samem modelu opis obravnavan kot celota, je take napake težko odkriti in še težje natančno odstraniti. Drugi pristop pa narekuje, da vizualni model razdelimo na več delov, vsak izmed katerih opisuje samo del izgleda objekta. V tem primeru se posodabljanje vizualnega modela preslika na posodabljanje določene podmnožice teh lokalnih vizualnih modelov, kar naredi celoten model bolj robusten glede drsenja. Vendar po drugi strani posamezni lokalni modeli z lokalnim opisom izgleda ne vidijo celote, kar lahko hitro vodi do dvoumnosti, oziroma več podobno ugodnih rešitev, izmed katerih bi jih lahko večino zavrgli, če bi imeli na voljo globalno informacijo, s katero bi lahko določili položaj dela v objektu.

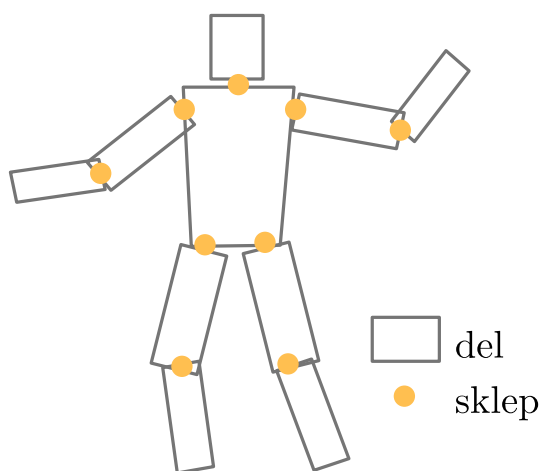


Slika 1.1: Ilustracija predstavitev vizualnega modela.

(a) - globalni vizualni model, ki opisuje izgled celotnega objekta, (b) - večje število lokalnih vizualnih modelov, od katerih vsak opisuje svoj del objekta, (c) - iščemo model, ki bi izkoriščal prednosti (a) in (b)

V okviru te magistrske naloge je predstavljen vizualni model, s katerim poskušamo odgovoriti na zgoraj zastavljeni vprašanji z novim pristopom k povezavi med globalnim in lokalnim opisom izgleda objekta. Kot je razvidno že iz naslova, je poudarek na obravnavi *netogih objektov*. Z našim pristopom sicer ciljamo predvsem na sledenje *artikuliranih objektov*, ki predstavljajo podmnožico množice netogih objektov. Artikulirani objekti so

objekti, ki so sestavljeni iz večjega števila delov, ki so med seboj povezani z gibljivimi sklepi. Dober primer artikuliranega telesa je človeško telo, katerega sledenje je predmet velikega števila aplikacij vizualnega sledenja. Preprosta ilustracija artikuliranosti človeškega telesa je prikazana na Sliki 1.2. Za razliko od splošnega koncepta netogih objektov lahko v primeru artikuliranih objektov še vedno predpostavljamo določene geometrijske omejitve med deli. Po drugi strani pa zaradi dodatnega nivoja deformacij izgleda, ki jih prinaša taka konstrukcija objekta, artikulirani objekti v vizualnem sledenju veljajo za težji problem kot togi objekti. Problem je mogoče omiliti z uporabo dodatnega znanja o strukturi objekta, vendar tak pristop po drugi strani omeji splošnost uporabe sledilnika. Primer uporabe geometrijskega modela za sledenje artikuliranega objekta najdemo v [18, 52]. Tak pristop nam nudi več informacij o samem stanju objekta, vendar je po drugi strani omejen na uporabo za sledenje točno določenih tipov artikuliranih objektov.



Slika 1.2: Preprost model človeškega telesa.

Gre za poenostavljen model, ki zaradi razlike v velikosti zanemari manjše sklepe na rokah in nogah.

Vizualni model, ki je predstavljen v tej magistrski nalogi, temelji zgolj na uporabi vizualnih informacij iz obravnavanega zaporedja slik, *brez predhodnega znanja o strukturi in obnašanju objekta*. Prednost takega pristopa je, da je zelo splošen, vendar ne izkorišča prednost, ki jih nudi poznavanje lastnosti objekta. Vse lastnosti objekta se mora naučiti med samim sledenjem.

Ideja temelji na formalizaciji vizualnega modela kot dvoplastne povezave lokalnega in globalnega vizualnega modela. Sama formalizacija neposredno sicer ne izboljša robustnosti sledenja, ponudi pa drugačen pogled na osveževanje vizualnega modela. Na podlagi le-te je bil nato razvit mehanizem vzajemnega osveževanja lokalnega ter globalnega opisa izgleda objekta. V magistrski nalogi bomo z eksperimenti dokazali, da ima tak pristop določene prednosti pred obstoječimi pristopi k sledenju, še posebej v primerih, kjer se lokalni izgled predmeta neprestano spreminja. Posodabljanje opisa je namreč zastavljeno s ciljem bolj zanesljivega določanja območja objekta v sliki, na podlagi katerega se lahko osvežuje množica lokalnih značilnic.

Uporabnost sledilnikov, ki so sposobni objektu na podlagi vizualnih informacij kljub njegovim deformacijam slediti daljše časovno obdobje, bi bilo tako mogoče uporabiti tudi v okviru implementacij interaktivnih vmesnikov človek-računalnik, sledenju ljudi v urbanih okoljih, športu itd., skratka v primerih, kjer sledenje z globalnimi vizualnimi modeli, kakršno je v tem času najbolj prisotno, otežuje spreminjanje izgleda objekta.



Zgradba magistrske naloge: Predlagani vizualni model oziroma na njem temelječi sledilnik je v poglavju 2 najprej vpet v raziskovalno področje s predstavitvijo sledilnikov, ki se sledenja objektov lotevajo na podoben način. V okviru predstavitve sledilnikov so izpostavljene tudi njihove omejitve. V poglavju 3 je predstavljena teorija vizualnega modela, skupaj s predstavitvijo uporabljenih algoritmov ter umestitvijo modela v preprost sledilnik. V poglavju 4 so opisani eksperimenti ter predstavljeni in komentirani rezultati ter primerjave z nekaterimi obstoječimi sledilniki. Poglavje 5 vsebuje razširjen komentar našega dela, tako glede teoretične plati kot tudi glede eksperimentalnih zaključkov, poglavje 6 pa vsebuje zaključni povzetek vsebine magistrske naloge ter smernice za nadaljnji razvoj. V dodatku A so predstavljene teoretične osnove metod, na katerih temelji naše delo, dodatek B pa vsebuje nekaj podrobnosti o uporabljeni programski opreml.

Poglavje 2

Pregled področja

V preteklih letih je kar nekaj raziskovalnih skupin predlagalo vizualne modele ter sledilnike, s katerimi so si prizadevali zaobjeti najboljše lastnosti globalnega in lokalnega opisa izgleda objekta. V tem poglavju bomo našli in opisali tiste predloge oziroma sledilnike, ki so najbližji našim lastnim raziskavam. Pregled je tematsko razdeljen na dva sklopa. Prvi je primarno namenjen pregledu raziskav glede na uporabljene geometrijske predpostavke v vizualnih modelih. Gre za vizualne modele, ki uporabljajo množico lokalnih značilnic za neposreden opis izgleda objekta. Opisane so metode za iskanje optimalnega ujemanja glede na te predpostavke popravkov ter pristopi za posodabljanje geometrijskega modela. Drugi sklop je namenjen pregledu sledilnikov, ki za robustno sledenje v daljšem časovnem obdobju uporabljajo izbiro značilnic. Gre za posodabljanje množice lokalnih značilnic ali za izbor znotraj določene množice globalnih komponent izgleda.

2.1 Optimizacija ujemanja funkcije

Najbolj razširjena formalizacija problema posodabljanja vizualnega modela je optimizacija ujemanja vizualnega modela in slike. Taka optimizacija v prvi vrsti sicer služi za samo iskanje ujemanja. Vendar pa se med optimizacijo lahko poleg položaja v sliki spreminjajo tudi drugi parametri sledilnika in vizualnega modela, kar lahko interpretiramo kot posodabljanje le-tega. V tem primeru gre za optimizacijo v visoko dimenzionalnem prostoru. Ker pa je taka optimizacija zelo zahtevna, je pogosta praksa, da se vanjo vpelje določene predpostavke. Te predpostavke izvirajo iz narave problema, njihova funkcija pa je znižanje dimenzije problema na tako, ki jo uporabljena optimizacijska metoda še lahko obvlada.

2.1.1 Omejevanje ranga

Torresani in sod. [54] uporabljajo fiksno množico značilnic, ki temeljijo na optičnem toku. Pri sledenju so upoštevane tudi relacije med posameznimi značilnicami, zato govorimo o $2N$ -dimenzionalnem problemu, kjer je N število značilnic. Optimizacija, na kateri temelji iskanje prilagoditve vizualnega modela, je osnovana na ugotovitvi, da podoben problem, v katerem nastopa togo telo, lahko opišemo z matriko, katere rang je v idealnih pogojih

omejen z $r \leq 4$, v primeru kamere z ortogonalno projekcijo oziroma $r \leq 8$, če je kamera perspektivna. V primeru netogega telesa pa se predpostavi, da je mogoče njegove deformacije in gibanje aproksimirati z linearno kombinacijo K osnovnih oblik [7, 26]:

$$S = \sum_{i=1}^K l_i S_i \quad S, S_i \in \mathbb{R}^{3 \cdot N}, l_i \in \mathbb{R}. \quad (2.1)$$

V tem primeru se v idealnih pogojih ter ob predpostavki šibke perspektivne projekcije¹ rang poveča na $r \leq 3K$. Avtorji predstavijo tudi iterativen optimizacijski pristop, s katerim rešijo potrebno razčlenitev matrike. Ena izmed prednosti takega pristopa je, da je zaradi enovite obravnave mogoče napovedati tudi položaje značilnic, ki jih v nekem časovnem koraku ni mogoče zanesljivo določiti s samo lokalno detekcijo. Primer je prikazan na Sliki 2.1.

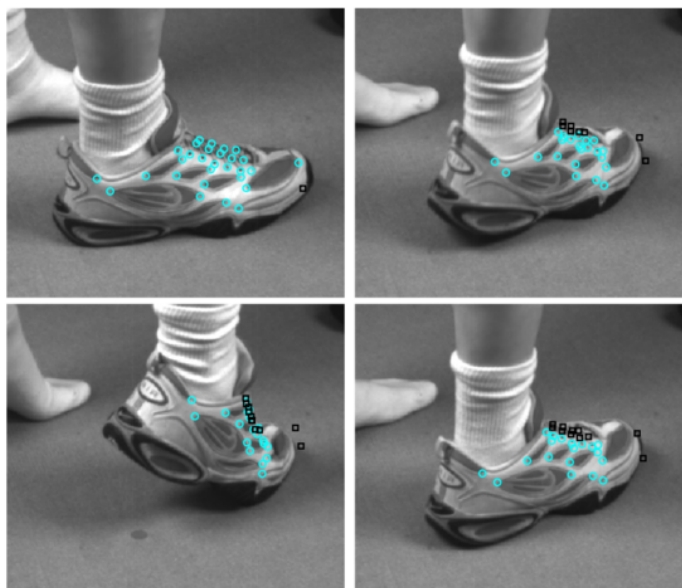
Problem takega pristopa je, da potrebuje vhodno video zaporedje slik z nizko vsebnostjo šuma, podrobno ročno inicializacijo ter da obdeluje zaporedje kot celoto (ang. *batch processing*). Zato ni primeren za sprotno sledenje, kar je pomembna zahteva v veliko praktičnih aplikacijah sledenja (vmesniki človek-stroj, sledenje za nadzor, šport, itd.), je pa, zaradi natančnosti rezultatov, primeren za operacije, kot je rekonstrukcija 3D modelov iz video posnetka s pomočjo gibanja.

Poleg tega tak način ni primeren za daljše sledenje netogih objektov, saj predpostavlja zelo preprost model gibanja, ki se v daljših časovnih obdobjih izkaže za preveč togega, njegovo omiljenje pa opazno zmanjša natančnost določanja trajektorije. Zato Buchanan in Fitzgibbon [9] k problemu pristopita s kombinacijo (časovno) lokalnega in globalnega modela gibanja. Njun algoritem naredi dva prehoda skozi video zaporedje slik. V prvem razbije problem oziroma video zaporedja na krajša pod-zaporedja (približno 10 slik), znotraj katerih je model gibanja bolj preprost in zato bolj ocenljiv z uporabo omejitve ranga. V drugem prehodu so tej lokalni modeli gibanja uporabljeni kot predznanje, s katerim se usmerja preprost Bayes-ovski sledilnik, ki deluje na podlagi lokalnih zaplat. Seveda pa ima tudi ta pristop omejitve, da mora biti celotno zaporedje slik poznano vnaprej.

2.1.2 Jedrne funkcije

Sledenje z uporabo jedrnih funkcij (ang. *kernel tracking*) [12] je popularna metoda pristopa k problemu vizualnega sledenja, saj jo odlikujeta široka uporabnost in računska učinkovitost, poleg tega pa je uspešna tudi pri nepopolnih ujemanjih vizualnega modela. Osnovna ideja sledenja z jedrno funkcijo je prikazana in opisana v okviru Slike 2.2. Prvi poskusi uporabe jedrnih funkcij za reševanje problema sledenja so uporabljali eno jedrno funkcijo v kombinaciji s preprostim globalnim vizualnim modelom [13, 58], v zadnjem času pa se je pojavilo nekaj predlogov, ki uporabo razširjajo na večje število jedrnih funkcij v kombinaciji z ločenimi vizualnimi modeli. Posamezen par jedrne funkcije in lokalnega vizualnega modela bomo od sedaj naprej v kontekstu tega razdelka imenovali značilnica. Namen take delitve je povečati robustnost sledenja, saj vsak vizualni model opiše samo del izgleda

¹Aproksimacija šibke perspektivne projekcije predpostavlja, da točke ležijo na ozkem globinskem intervalu, zato lahko popačenje, ki nastopi zaradi perspektivne projekcije, zanemarimo.



Slika 2.1: Primer sledenja z omejevanjem ranga.

Modri krogi označujejo zaznane značilnice, črni kvadrati pa značilnice, ki v danem časovnem koraku niso zaznane, je pa njihov položaj določen na podlagi ostalih značilnic. Slika povzeta po [54].

objekta, to pa pomeni, da se ob lokalni spremembi izgleda objekta zmanjša natančnost zgolj določene podmnožice le-teh.

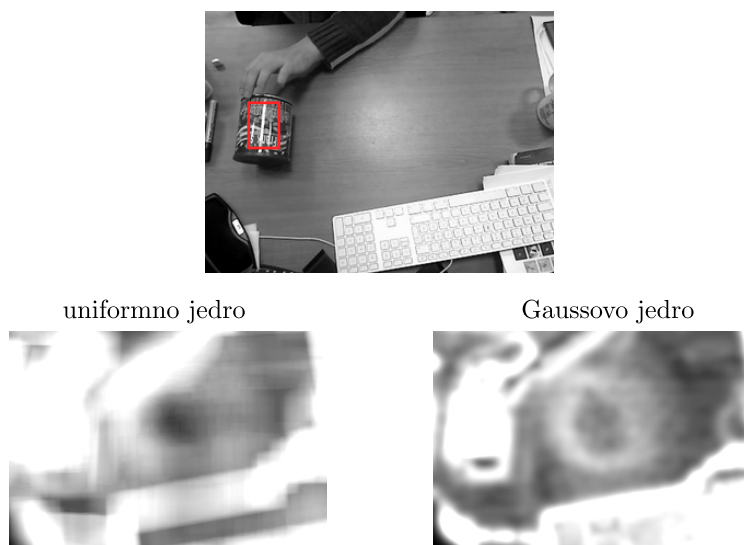
V vseh delih, obravnavanih v okviru tega razdelka, vizualni modeli temeljijo na lokalnih slikovnih zaplatah (ang. *image patches*), torej pravokotnih regijah slikovnih elementov. Za kompakten opis lastnosti zaplate je najpogosteje uporabljen histogram sivinskih vrednosti slikovnih elementov. Histogram zaplate, ki jo definira njena centralna točka \mathbf{c} , lahko, uteženega z jedrno funkcijo k , zapišemo v matrični obliki kot:

$$p(\mathbf{c}) = \frac{1}{C} \mathbf{U}^T \mathbf{K}(\mathbf{c}), \quad (2.2)$$

kjer C predstavlja normalizacijski koeficient, \mathbf{U} matriko histograma zaplate², $\mathbf{K}(\mathbf{c})$ pa matriko jedrne funkcije, centrirane na položaju \mathbf{c} [23]. Če opis referenčne zaplate oziroma njenega uteženega histograma označimo s q , potem definiramo optimizacijsko funkcijo kot vsoto kvadratov razdalj (ang. *sum of squared distances*) referenčnega histograma in trenutnega histograma:

$$\|\sqrt{q} - \sqrt{p(\mathbf{c} + \Delta\mathbf{c})}\|^2, \quad (2.3)$$

²Matrika ima dimenzije $n \times m$, kjer je n število slikovnih elementov zaplate, m pa število celic histograma. Vsak stolpec matrike vsebuje 1 v celicah, za katere vrednost pripadajočega slikovnega elementa pade v pripadajočo celico histograma, drugače pa 0.



Slika 2.2: Sledenje z jedrno funkcijo.

Primer prikazuje ujemanje histograma rdeče označenega območja z ostalimi področji enakih dimenzij na sliki. Konstrukcijo histogramov določa enačba (2.2), funkcijo razdalje pa enačba (2.3). Bela barva nakazuje večje ujemanje. Leva slika je pridobljena z uporabo uniformnega jedra, kar je ekvivalentno temu, da jedra ne uporabljamo. Desna slika prikazuje odziv z uporabo Gaussovega jedra. Že iz tega primera je razvidno, da je površina funkcije ujemanja v primeru uniformnega jedra dosti bolj groba kot v primeru Gaussovega jedra, kar onemogoča kvalitetno lokalno optimizacijo.

kjer $\mathbf{c} + \Delta\mathbf{c}$ označuje točko, katere ujemanje hočemo maksimirati, le-to pa dosežemo z minimizacijo optimizacijske funkcije. Ker ujemanjska funkcija podeduje zveznost jedrne funkcije, poleg tega pa imamo opravka z sorazmerno majhnimi premiki, lahko tak optimizacijski problem rešujemo z uporabo linearizacije optimizacijske funkcije.

Pri uporabi večih značilnic je potrebno optimizacijsko funkcijo razširiti. Najpreprostejša razširitev je, da ne upoštevamo odvisnosti med značilnicami [15]. V tem primeru lahko funkcijo zapišemo kot:

$$\sum_{i=1}^N \|\sqrt{q_i} - \sqrt{p(\mathbf{c}_i + \Delta\mathbf{c}_i)}\|^2, \quad (2.4)$$

kjer je N število značilnic. Pravi potencial pristopa uporabe večjega števila značilnic pa je prav v vpeljavi dodatnih omejitev v optimizacijsko funkcijo. Fan in sod. [15] tako predlagajo vpeljavo dodatnega člena, ki v optimizacijo vnese medsebojne prostorske omejitve značilnic. Optimizacijsko funkcijo razširimo z R omejitvami z dodanim drugim členom:

$$\sum_{i=1}^N \|\sqrt{q_i} - \sqrt{p(\mathbf{c}_i + \Delta\mathbf{c}_i)}\|^2 + \sum_{i=1}^R \gamma_i \|\Omega_i(\{\mathbf{c}_j + \Delta\mathbf{c}_j\}_{j=1:N})\|^2, \quad (2.5)$$

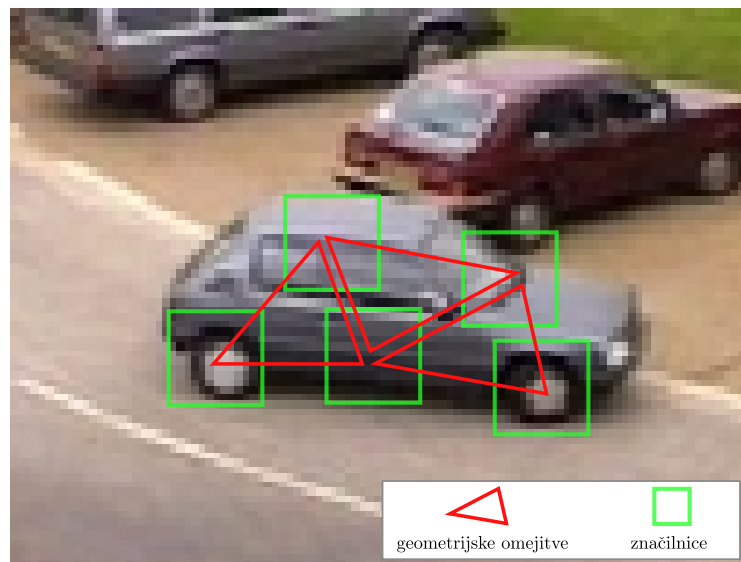
kjer $\Omega_i(\mathbf{c}_1, \dots, \mathbf{c}_N)$ predstavlja eno omejitev, γ_i pa utež posamezne omejitve napram dru-

gim omejitvam ter prvemu členu funkcije. V [15] so avtorji predlagali tudi konkretno optimizacijsko funkcijo z eno samo omejitvijo nad vsemi značilnicami. Omejitev teži h konstantni medsebojni razdalji značilnic. Taka omejitev je seveda v večini realnih scenarijev sledenja velika poenostavitev, saj predpostavlja togo konfiguracijo značilnic glede na ravnino kamere, kljub temu pa prispeva k robustnosti napram uporabi enega jedra in globalnega vizualnega modela.

Martinez in Binefa v [42] sta zato predlagala omilitev togosti z vpeljavo večjega števila omejitev, ki pa vplivajo samo na gibanje podmnožic teh značilnic. Predlagata optimizacijsko funkcijo, ki vsebuje $R = N - 2$ omejitev, pri čemer vsaka omejitev vpliva zgolj na tri sosednje značilnice (Slika 2.3) ter njihovo gibanje omejuje na podprostor, ki ga opisuje njihova skupna afina transformacija:

$$\sum_{i=1}^N \|\sqrt{q_i} - \sqrt{p(\mathbf{c}_i + \Delta\mathbf{c}_i)}\|^2 + \sum_{i=1}^{N-2} \gamma_i \|\Omega_i(\mathbf{c}_i + \Delta\mathbf{c}_i, \mathbf{c}_{i+1} + \Delta\mathbf{c}_{i+1}, \mathbf{c}_{i+2} + \Delta\mathbf{c}_{i+2})\|^2. \quad (2.6)$$

Z matrično formulacijo optimizacijske funkcije, definirane v (2.6), je mogoče celoten problem iterativno rešiti v enem kosu. Taka razširitev sledenja z uporabo jedrnih funkcij je uspešna pri deformacijah, ki v kratkih slikovnih zaporedjih nastopijo zaradi afinih transformacij izven ravnine kamere, odpove pa pri daljšem sledenju oziroma večjih stopnjah deformacije. Pri takih pogojih se izgled objekta namreč spremeni do mere, ko bi morali za uspešno nadaljevanje sledenja določeno podmnožico značilnic zavreči oziroma obnoviti. Poleg tega je uspešnost sledenja zelo odvisna od ročne postavitve vseh značilnic pred začetkom sledenja.



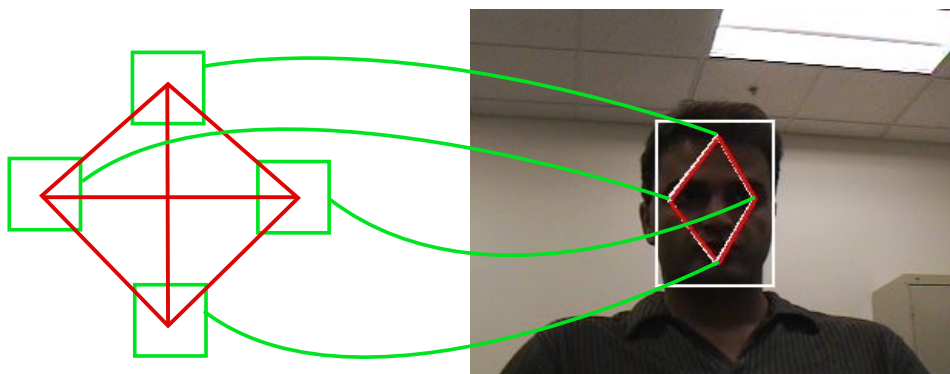
Slika 2.3: Ilustracija povezav med značilnicami.
Ilustracija povezav med značilnicami v vizualnem modelu iz [42].

2.1.3 Vzorčenje optimizacijske funkcije

V primeru, ko funkcije, ki ocenjuje ujemanje vizualnega modela s sliko, ni mogoče (učinkovito) optimizirati z uporabo analitičnih iterativnih pristopov, lahko to storimo s pomočjo statistike. Optimizacijsko funkcijo preoblikujemo v verjetnostno porazdelitev, tako formuliranega problema pa se lahko lotimo na primer z nekaterimi metodami iz družine Monte Carlo. Badrinarayanan in Perez v [2] spojita filter z delci s sledenjem z lokalnimi slikovnimi zaplatami preko preprostega geometrijskega modela, ki ga sestavljajo štiri med seboj povezane slikovne zaplate (Slika 2.4). Skupno stanje modela je tako vzorčeno na podlagi skupne aposteriorne verjetnostne porazdelitve:

$$p(\mathbf{X}|\mathbf{Y}) \propto l_t(\mathbf{Y}|\mathbf{X})l_{\mathcal{P}}(\mathbf{X}|\mathcal{P}) \prod_{(i,j) \in \Gamma} \psi(x_i, x_j), \quad (2.7)$$

kjer $l_t(Y|X)$ označuje lokalno podobnost zaplat z ustreznimi področji v sliki³, $l_{(P)}(X|\mathcal{P})$ predstavlja geometrijsko podobnost modela s fiksnim modelom (P), $\psi(x_i, x_j)$ pa predstavljajo omejitve na povezavi med parom značilnic (i, j). Izkaže se, da taka formulacija problema uspešno sledi objektom, kot je na primer glava, še vedno pa ima nekatere pomanjkljivosti. Najbolj pomembna izmed njih je, da celotna optimizacija poteka neposredno v visokodimenzionalnem prostoru, ki ga definira vektor položajev vseh značilnic. V omenjenem primeru gre za osem dimenzij, ob povečevanju števila značilnic pa število dimenzij linearno raste. Da bi pri tej rasti obdržali enako kvaliteto opisa verjetnostne porazdelitve, bi morali signifikantno povečati število delcev, s tem pa bi se povečala tudi računski zahtevnost sledilnika.



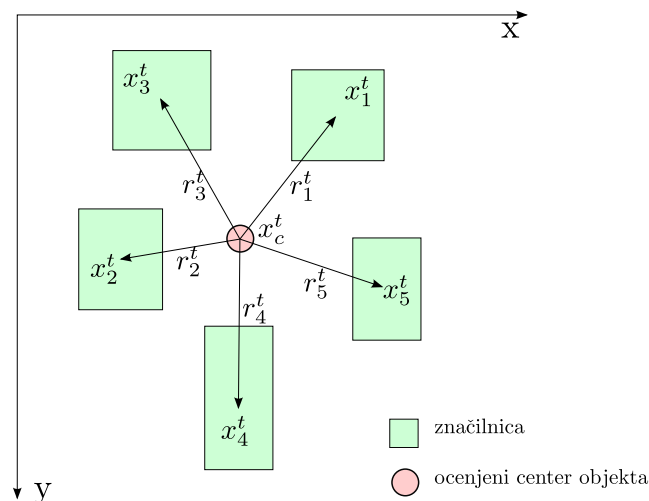
Slika 2.4: Preprosti geometrijski model za sledenje obrazov.
Sestava geometrijskega modela iz [2].

Chang in sod. [10] predlagajo metodo za sledenje delov objekta, ki temelji na Bayesovih mrežah. V bistvu je že filter z delci, ki obravnava vsako opisano značilnico, ki opisuje del objekta ločeno, poseben primer Bayesove mreže, vendar gre v tem primeru samo za verižno odvisnost v okviru časovnih stanj posamezne značilnice, ki je preprosta za obravnavo ter ni računsko zahtevna. Po drugi strani pa to ne omogoča "sodelovanja" značilnic, kar bi

³ Avtorji za primerjavo uporabljajo normalizirano križno korelacijo.

pomagalo v primerih, ko posamezne lokalne značilnice zaradi dvoumnosti odzivov v sliki nimajo jasno določenega novega stanja. Chang in sod. zato predlagajo uvedbo odvisnosti med posameznimi značilnicami, obenem pa predlagajo postopek, kako te odvisnosti primerno obravnavati in se s tem izogniti naraščajoči časovni kompleksnosti. Vsaka lokalna značilnica se vzorči najprej ločeno, nato pa se ocene delcev, ki opisujejo verjetnostno porazdelitev položaja značilnice, popravijo glede na geometrijske omejitve med značilnicami. Geometrijske omejitve so podane vnaprej in so fiksne čez celo obdobje sledenja, kar predstavlja omejenost modela v primeru prevelike deformacije objekta ter v primeru rotacije izven ravnine kamere.

Kwon in Lee [37] predlagata uporabo množice lokalnih slikovnih zaplat različnih dimenzij, povezanih s preprostim geometrijskim modelom. Model predpostavlja, da vse zaplate ohranjajo razdaljo glede na center vizualnega modela (ocenjeno središče objekta). Shema takega modela je prikazana na Sliki 2.5.



Slika 2.5: Geometrijski model s povezavami na središče.

Skica modela, uporabljenega v [37]. Simboli x_*^t opisujejo absolutne položaje posameznih komponent v prostoru slike, simboli r_*^t pa relativne položaje značilnic glede na x_c^t .

Za samo sledenje avtorja predlagata novo metodo iz družine MCMC metod. Imenujeta jo *Adaptive Basin Hopping Monte-Carlo*, temelji pa na poenostavitvi "površine" funkcije verjetnostne porazdelitve. Cilj glajenja je, da se poveča možnost, da z optimizacijo pridemo do dejanskega globalnega optimuma. Stanje modela avtorji zapišejo kot vektor, ki vsebuje ocenjeno središče objekta, položaje posameznih zaplat ter relativne položaje posameznih zaplat (slednji se med optimizacijo ne spreminjajo). Podobno kot pri splošni MCMC metodi, A-BHMC sestavljata dva koraka: predlaganje ter sprejemanje. V okviru prvega koraka se prvi približek novega stanja določi z vzorčenjem verjetnostne porazdelitve, ki jo določata stanje modela v prejšnjem koraku ter fiksna gostota. Nadaljnji predlogi pa se vzorčijo z uporabo regulacijskega člana, ki skrbi, da povprečje položajev zaplat ne zdrsi predaleč od ocenjenega središča objekta. Sprejemanje uvaja kriterij, ki temelji na lokalnem ujemanju

posamezne slikovne zaplate s sliko v lokalnem optimumu⁴ ter razdalji med relativnim položajem lokalnega optimuma glede na predlagano središče objekta ter relativnim položajem zaplate v prejšnjem koraku. Kljub temu, da Kwon in Lee s tem pristopom poenostavljata obravnavanje visokih dimenzij, pa ne smemo pozabiti, da gre za šibek geometrijski model, ki upošteva zelo malo omejitev za gibanje posameznih značilnic, po drugi strani pa se upoštevane omejitve med sledenjem ne posodablajo.

Geometrijski model je mogoče aplicirati tudi na drugačen problem sledenja - sledenje večjega števila vizualno podobnih objektov. V tem primeru geometrijske relacije služijo za identifikacijo posameznih objektov. Khan in sod. [29] si pri sledenju večjega števila mravelj pomagajo z gradnjo Markovskega naključnega polja z povezavami med mravljami, ki se nahajajo blizu ena drugi. Glavni problem takega sledenja je, da so si mravlje vizualno zelo podobne, zato se rado zgodi, da sledilnik posamezne mravlje v določenem trenutku preskoči na sosednjo mravljo. Posamezne sledilnike, ki temeljijo na filtru z delci, avtorji zato povežejo v skupni filter z delci, pri čemer visoko-dimenzionalni prostor, ki pri tem nastane, razbijejo na podprostore (glede na bližino posameznih mravelj, kot je to prikazano na sliki 2.6). V posameznem podproblemu potem vpeljejo v vzorčeno verjetnostno porazdelitev geometrijske potenciale:

$$p(\mathbf{X}|\mathbf{Y}) \propto \prod_i P(x_i|y_i) \prod_{(i,j) \in \Gamma} \psi(x_i, x_j), \quad (2.8)$$

kjer Γ predstavlja množico povezav v podprostoru, $\psi(x_i, x_j)$ pa funkcijo posameznega potenciala, ki drži posamezne sledilnike narazen.

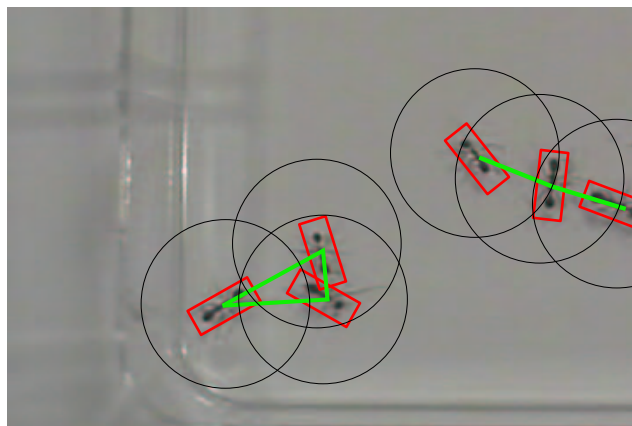
Še enkrat poudarjamo, da gre v tem primeru za drugačen problem, v okviru katerega je predpostavljeno, da imamo konstantno število "delov", relacije med njimi pa so uporabljene zgolj po potrebi.

2.1.4 Iskanje ujemanja stabilnih regij

Problem klasične optimizacije je, da prehitro zaide v smer lokalno optimalnih rešitev, ki se kasneje izkažejo za neustrezne. Zaradi visokih dimenzij prostorov, v katerih se rešujejo taki problemi, si je le-te težko interpretirati ter jih s pomočjo omejitev usmeriti proti globalni rešitvi. Alternativa temu pristopu je iskanje optimalne konfiguracije v diskretnem prostoru. Če ima dimenzija, ki opisuje vsako značilnico, na voljo le nekaj diskretnih stanj, se lahko problem poenostavi na iskanje optimalne kombinacije teh stanj pri ustreznih geometrijskih omejitvah. Druga, podobna, možnost je, da je za vsako značilnico na voljo le eno ujemanje, najti pa moramo ustrezno podmnožico teh ujemanj, ki ustreza določeni družini transformacij.

V računalniškem vidu je že dalj časa popularen koncept t.i. *stabilnih regij*. To so področja v sliki, katerih izgled je delno neodvisen od transformacij, ki nastanejo zaradi spremembe položaja kamere ali objekta v prostoru. V zvezi s stabilnimi regijami govorimo o dveh pojmi: o detektorju zanimivih področij, ki take regije v sliki najde, ter o opisniku,

⁴Zaplata se s sliko ne primerja neposredno v predlaganem položaju. Za izboljšavo ocene je uporabljena metoda za iskanje lokalnega optimuma.



Slika 2.6: Povezave med blizu ležečimi objekti.

Skica dinamičnega geometrijskega modela, uporabljenega v [29]. Povezave (zelene črte) med posameznimi sledilniki (rdeči pravokotniki) so vzpostavljene na podlagi prekrivanja okolice sledilnikov (črni krogi).

ki tako regijo na kompakten način opiše. Take opise se lahko nato robustno primerja med seboj in posledično ugotovi, kateri opisniki se dejansko ujemajo.

Yin in Collins [60] za sledenje avtomobilom s helikopterja predlagata uporabo Harrisovega detektorja robov za detekcijo zanimivih točk. Področja okrog teh točk so opisana kot sivinske zaplate in v tem kontekstu ustrezajo formulaciji lokalnih značilnic. Med sledenjem se najdene zaplate v novi sliki video zaporedja slik primerjajo z obstoječimi značilnicami v vizualnem modelu z uporabo normalizirane križne korelacije. Vsaka zaplata je pripisana natanko eni značilnici⁵. Iz tako določene množice parov avtorja izbereta podmnožico, katere gibanje najbolje opisuje afino transformacijo v dveh dimenzijah. Za to je uporabljen algoritem, ki temelji na idejah metode soglasja naključnih vzorcev (ki je na kratko predstavljena v razdelku A.2). Ideja iskanja podmnožice je prikazana na Sliki 2.7.

Yang in Wu v svojem članku [59] lokalne značilnice opredelita kot 2D histograme v normaliziranem barvnem podprostoru. Le-tej so v sliki določeni z uporabo Harris-Laplaceovega detektorja zanimivih točk. Iskanje ujemanj je potem formulirano kot iskanje optimalne konfiguracije Markovskega naključnega polja (ang. *Markov random field*). Skupna a posteriori verjetnost v tem primeru je formulirana kot:

$$p(\mathbf{X}|\mathbf{Y}) \propto \frac{1}{(\sqrt{2\pi}\sigma)^{|C|}} \prod_{(i,j) \in C} e^{-V(x_i, x_j)} \prod_i p(y_i | x_i), \quad (2.9)$$

kjer C predstavlja množico parov povezanih značilnic (po definiciji so to tiste značilnice, katerih regije se prekrivajo), $V(x_i, x_j)$ predstavlja geometrijsko omejitev med dvema značilnicama na podlagi medsebojnih kotov, $p(y_i | x_i)$ pa ujemanje i -te zaplate z predlogo. Parameter σ definira elastičnost geometrijskega modela in se spreminja dinamično.

⁵Gre za dvostranski problem linearnega dodeljevanja oziroma "problem porok". Vsaka značilnica se preslika v tisto novo zaplato, ki ji najbolj ustreza.

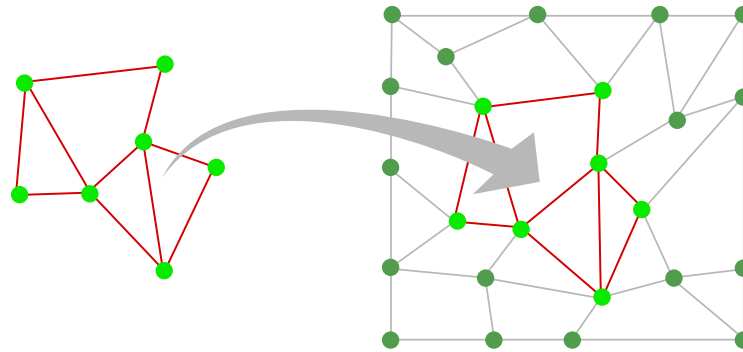


Slika 2.7: Izbira podmnožice ujemanj.

Ilustracija izbire podmnožice ujemanj, slika povzeta po [60]. Leva slika prikazuje najdena ujemanja, desna pa izbrano podmnožico ujemanj, katere premik je mogoče opisati z afino transformacijo v dveh dimenzijah.

Tang in Tao [53] sta v svojem vizualnem modelu uporabila dobro poznane SIFT opisnike. Vizualni model vsebuje dve množici SIFT značilnic: množico *stabilnih značilnic* ter množico *kandidatov*. Stabilne značilnice so tiste značilnice, ki so se pojavile v več zaporednih časovnih korakih. Take značilnice se zato uporabijo za samo sledenje. Množica kandidatov pa vsebuje značilnice, ki vsebuje potencialne stabilne značilnice, ki se, glede na njihovo ponovljivost, premaknejo v množico stabilnih značilnic, ali pa so odstranjene. Na podoben način so odstranjene tudi značilnice iz množice stabilnih značilnic, če se niso pojavile v določenem številu zaporednih slik. Samo sledenje poteka na iskanju ujemanja podgrafa, ki ga opisujejo stabilne značilnice v grafu, ki ga opisujejo zaznane značilnice v sliki trenutnega časovnega koraka (Slika 2.8). Iskanje podgrafa je v splošnem NP-težek problem, ki pa se ga da obvladati z lokalnim iskanjem ter ustrezno inicializacijo. Povezave v grafu se tvorijo samo med značilnicami, ki so si dovolj blizu ter dovolj podobne glede velikosti ter orientacije.

Pristop iskanja ujemanja v diskretnem prostoru stanj na videz reši probleme zvezne optimizacije, vendar pa tudi tak način sledenja ni brez pomankljivosti. Največja izmed njih izvira prav iz dejstva, da za iskanje ujemanja uporabljamo samo ustrezno število vnaprej izbranih ujemanj lokalnih značilnic. S tem, ko je onemogočena možnost delnih ujemanj, se namreč pojavi problem, da za določeno značilnico v novi sliki ni mogoče najti niti enega dovolj dobrega ujemanja, kar pomeni tudi, da pri iskanju ujemanja v ustrezni dimenziji ne moremo poiskati preslikave. Če je del takih značilnic prevelik, ima to neposreden vpliv na uspešnost iskanja ujemanja, v nasprotnem primeru pa je še vedno potrebno izgubljene značilnice nadomestiti z novimi. S tem pa se seveda pojavi problem iskanja novih ustreznih stabilnih regij. Poleg tega je potrebno omeniti še problem, da trenutno znani opisniki stabilnih regij delujejo samo na določenih objektih oziroma na določenem tipu izgledov le-teh. Pomembno je na primer, da ima objekt značilno ter kontrastno teksturo, kar nam nekoliko omeji možnosti aplikacije takega sledilnika, poleg tega pa so opisniki omejeni s stopnjo rotacije ter spremembo velikosti objekta.



Slika 2.8: Iskanje ujemanja podgrafa.

Levi del slike prikazuje model grafa, ki predstavlja povezave med posameznimi značilnicami v vizualnem modelu. Desni del predstavlja večji graf, ki opisuje regijo v sliki v okolici trenutnega položaja objekta. Algoritem mora poiskati optimalno ujemanje levega grafa v desnem grafu. Slika poveza po [53].

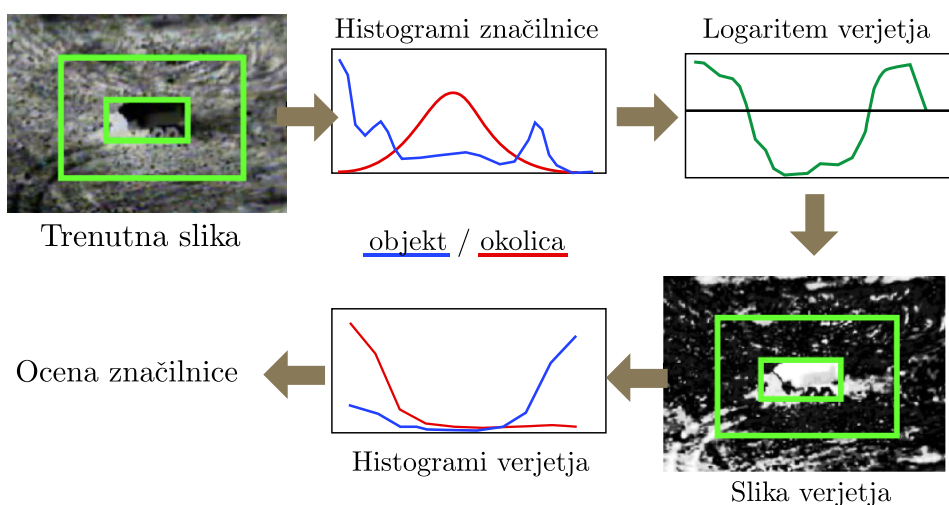
2.2 Osveževanje modela izgleda

Pomemben aspekt vizualnega modela, ki bi bil sposoben dalj časa slediti netogim objektom, je tudi ustrezno posodabljanje modela izgleda objekta. V okviru modelov, ki se zanašajo na lokalne opise modela, govorimo o odstranjevanju ter dodajanju lokalnih značilnic, na splošno pa o izbiri ustreznih značilnic, ki omogočajo robustno sledenje. Večina vizualnih modelov, omenjenih v razdelku 2.1, menjave značilnic ne podpira, tisti pa, ki jo, so ponovno omenjeni v tem razdelku skupaj z nekaterimi opazno drugačnimi pristopi.

2.2.1 Izbira značilnic

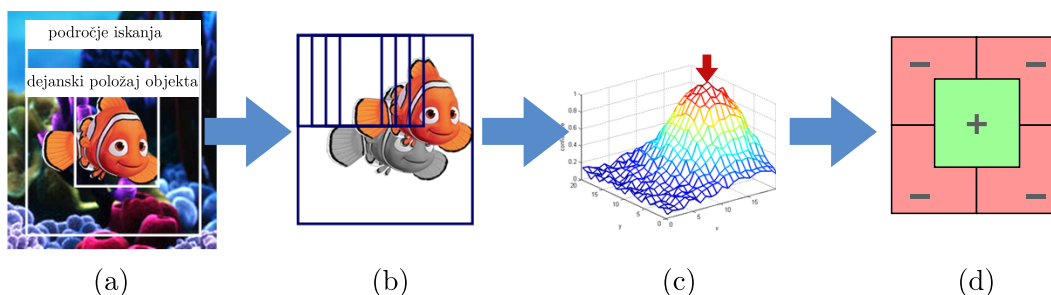
Enega izmed bolj poznanih sledilnikov, ki med sledenjem spreminjajo množico značilnic, so predstavili Collins in sod. v [11]. Avtorji opisujejo sledilnik, ki temelji samo na globalni informaciji, značilnice pa so definirane kot opisi izgleda objekta v 49 linearnih kombinacijah barvnih komponent barvnega prostora RGB. Posamezna značilnica je v vsakem časovnem koraku ocenjena na podlagi tega, kako dobro ločuje izgled objekta od okolice, ter na podlagi tega vključena v sledenje ali ne. Prikaz ocenjevanja značilnice je prikazan na Sliki 2.9. Omejitev takega pristopa je predpostavljena fiksna velikost objekta, ki mora biti določena vnaprej. Poleg tega vizualni model temelji zgolj na barvni informaciji in tako ni uporaben v primeru sivinskih video zaporedij.

Grabner in sod. v [21, 22] obravnavjao sprotno izbiro značilnic v okviru algoritmov spodbujanja (ang. *boosting*). Vizualni model sestavlja množica preprostih značilnic, ki temeljijo na razliki intenzitete dveh soležnih regij v sliki. Značilnice so izbrane iz večje množice značilnic, ki vsebuje vse možne kombinacije dveh soležnih regij v določenem pravokotnem območju. Metoda temelji na algoritmu AdaBoost [19], katerega uporabo sta ga v računalniškem vidu popularizirala že Viola in Jones [57] z njuno uporabo algoritma za detekcijo obrazov. Grabner in sod. metodo detekcije razširijo iz paketnega učenja na



Slika 2.9: Izbira značilnic za sledenje.
 Ilustracija izbire značilnic, povzeta po [11].

sprotno učenje. Na ta način lahko detektor hitro posodabljam (izbiramo novo množico značilnic), kar omogoča uporabo v sledenju, kjer se izgled objekta neprestano spreminja. Ideja pristopa je prikazana v Sliki 2.10.



Slika 2.10: Prikaz sledenja sprotno osveževanim detektorjem na podlagi spodbujanja.
 (a) - algoritem deluje na podlagi preiskovanja lokalne okolice trenutnega položaja objekta (b) - v vsakem časovnem koraku je to področje preiskavano z detektorjem (c) - na podlagi odzivov detektorja je za novi položaj objekta izbran položaj z največjim odzivom (d) - detektor je na podlagi novega položaja osvežen s pozitivnim primerom in negativnimi primeri, ki so vzeti iz neposredne okolice pozitivnega primera. Slika je povzeta po [22].

Problem pristopa je omejitev na pravokotno regijo, ki definira meje za množico potencialnih značilnic. Na ta način se omejimo na objekte, ki imajo določeno vizualno velikost in obliko. Če se oblika objekta zelo razlikuje od pravokotnega območja, se lahko zgodi tudi, da se vizualni model nauči lastnosti ozadja. To se zgodi v primerih, ko sam izgled objekta ne vsebuje območij z dobrim medsebojnim kontrastom, na katerih temeljijo značilnice, izgled ozadja pa jih.

Badrinarayanan in sod. [3] so predlagali sledilnik, ki temelji na dveh modalnostih, med katerima dinamično preklaplja glede na njuno kvaliteto. Avtorji združijo dva filtra z delci, prvi temelji na sivinskih zaplatah, ki se s sliko primerjajo na podlagi križne korelacije, drugi pa na podlagi barvnega modela. Oba filtra z delci delujeta neodvisno, brez izmenjave informacij. Avtorji za obe modalnosti predlagajo kriterija, ki določita, ali je posamezen filter z delci pri sledenju uspešen. Krovni sledilnik potem preklaplja med modalnostma glede na to, kateri filter z delci je uspešen. Tudi v tem primeru je glavna omejitev sledilnika predpostavljena fiksna velikost objekta.

2.2.2 Pravila jate

Eden izmed zgodnjih poizkusov združevanja lokalnega in globalnega opisa izgleda predmeta je t.i. koncept jate značilnic (ang. *flock-of-features*) [30]. Ideja jate je, da uporabimo množico lokalnih značilnic, katerih skupinska dinamika je regulirana s preprostimi pravili. V osnovnem modelu je jata definirana kot struktura s tremi komponentami

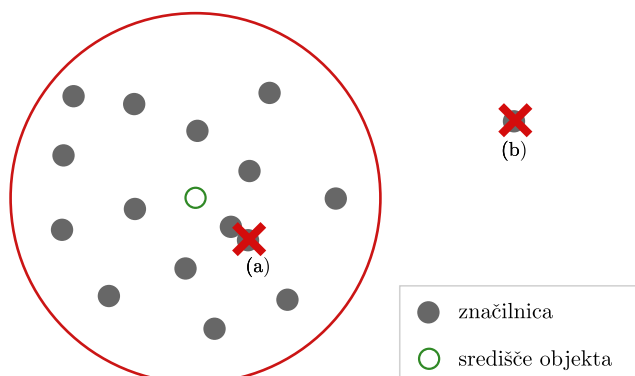
$$F = \{W, \theta, \xi\}, \quad (2.10)$$

kjer W predstavlja množico lokalnih značilnic, θ pravilo postavitve novih značilnic, ξ pa pravilo odstranitve obstoječih značilnic.

Koncept sta prva predstavila Kölsch in Turk v [30]. Poglavitni namen članka je formulacija metode za sledenje netogim objektom, primarno človekovi roki, z možnostjo uporabe takega sistema v uporabniških vmesnikih za mobilne računalniške sisteme. Avtorja uporabljata množico značilnic, ki temeljijo na optičnem toku. Značilnice sledijo posameznim lokalnim delom predmeta. Posamezne značilnice so ponastavljene na koordinatah v sliki, vzorčenih na podlagi verjetnostne porazdelitve (pravilo θ). Le-ta je sestavljena na podlagi ustrežanja posameznih slikovnih elementov statičnemu barvnemu modelu. Ker si v omenjenem članku avtorja za cilj zastavita sledenje rok, uporabljata za to barvni model kože. Posamezna značilnica nima velike opisne moči, zato se lahko hitro zgodi, da zaradi prevelike dvoumnosti zaide stran od večine (se oddalji od jate) ali pa pride preblizu posamezni drugi značilnici. Prav na tem temeljita koncepta, ki sestavljata pravilo ξ (ilustrirano tudi na Sliki 2.11). Prvi del pravila je maksimalna oddaljenost od središča jate, drugi pa prepoved trkov med posameznimi značilnicami. Odstranjeno značilnico sistem nadomesti z novo na položaju, določenim na podlagi že omenjene verjetnostne porazdelitve oziroma pravila θ .

Glavni problem koncepta jate značilnic je njegov preprost prostorski model, ki se uporablja samo za odstranjevanje potencialno problematičnih značilnic, ne pa za konkretno razreševanje dvoumnosti. Tak pristop vodi do hitrih sprememb znotraj množice značilnic, kar naredi celoten sledilnik zelo odvisen od stabilnosti metode za ponovno inicializacijo značilnic. Le-ta torej v omenjenem kontekstu predstavlja dodatno omejitev sistema, saj lahko statični barvni model ob spremembi osvetlitve hitro odpove.

Koncept jate značilnic je kasneje razširil Hoey [25], ki je pravili omejevanja (ξ) ter postavitve (θ) integriral neposredno v postopek sledenja, ki v tem primeru temelji na filtru z delci (ang. *particle filter*). Taka obravnava primera torej razširi osnovno funkcijo pravila ξ , saj lahko sedaj le-to tudi usmerja sam potek sledenja. Hoey je uporabil tudi bolj robustne značilnice, ki temeljijo na barvni porazdelitvi štirih sosednjih slikovnih elementov. Pravilo θ



Slika 2.11: Ilustracija pravila ξ .

(a) - Značilnici sta preblizu ena drugi, zato je ena odstranjena. (b) - Značilnica je predaleč od večine, zato je odstranjena.

je v tem primeru prav tako statična barvna porazdelitev, opisana z Gauss-ovo porazdelitvijo v barvnem prostoru.

2.2.3 Barvna podobnost za osveževanje množice lokalnih značilnic

Kwon in Lee [37] v svojem sledilniku, ki smo ga omenili že v razdelku 2.1.3, omogočata, da se posamezne zaplate, ki so uporabljene za opis objekta, tekom sledenja iz modela odstranjene ali osvežene, prav tako pa so lahko v model dodane nove zaplate. Odstranjevanje oziroma osveževanje je sproženo za posamezno zaplato glede na dva kriterija. Prvi kriterij je barvna podobnost zaplate z barvo ozadja, drugi pa prekrivanje zaplate z drugimi zaplatami ⁶.

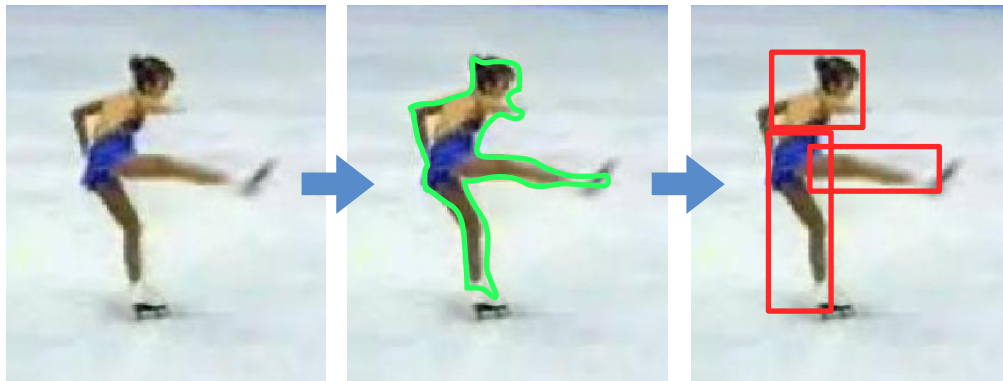
Nejhum in sod. v [43] vizualni opis objekta razbijejo na več pravokotnih področij, ki čimbolj celovito opišejo objekt na sliki. Predlagani postopek v vsakem koraku segmentira objekt glede na njegovo barvo napram barvi ozadja. Segmentirano območje je nato razbito na več pod-področij z uporabo požrešnega algoritma, ki išče kompromis med številom področij ter natančnostjo opisa. Dobljena področja so nato uporabljena v togi konstelaciji za določitev položaja objekta v naslednjem časovnem koraku (Slika 2.12).

V obeh opisanih sledilnikih pa je predpostavljen fiksni barvni model, ki po eni strani zagotavlja stabilnost sledenja v primerih, ko se barva objekta ne spreminja, obenem pa barva okolice ni podobna barvi objekta. Po drugi strani pa taka predpostavka predstavlja omejitev, ko od začetka sledenja ne poznamo celotnega barvnega opisa objekta ali pa se le-ta med sledenjem zaradi spremembe osvetlitve spremeni.

2.2.4 Stabilne regije

Za vse metode, ki so opisane v razdelku 2.1.4, velja, da morajo svoj vizualni model oziroma množico opisnikov stabilnih regij, ki pripadajo objektu, neprestano osveževati. Kljub temu,

⁶Gre torej za podobno pravilo, kot je uporabljeno tudi v jati značilnic v razdelku 2.2.2.



Slika 2.12: Razbitje opisa objekta na zaplate.

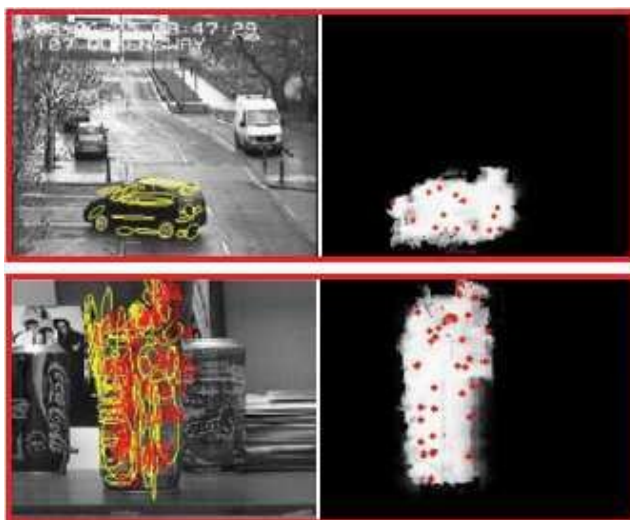
Na podlagi barve objekta ter okolice se določi segmentacija objekta (srednja slika). Na podlagi segmentacije se nato določijo posamezne zaplate z uporabo požrešnega algoritma (desna slika).

Slika je povzeta po [43].

da so stabilne regije odporne na določeno mero osnovnih deformacij, vseh sprememb, ki se pojavijo med sledenjem, enostavno ne morejo opisati. Odstranjevanje opisnikov iz vizualnega modela v večini predlaganih sledilnikov temelji na njihovem pojavljanju v video zaporedju slik. Yang in Wu [59] recimo odstraniti tiste opisnike, ki se v določenem časovnem oknu video zaporedja niso pojavili dovolj pogosto. Dodajanje novih opisnikov temelji na metodah za detekcijo stabilnih regij. Yin in Collins v [60] uporabljata Harrisov detektor robov za detekcijo stabilnih lokalnih opisov slike, njihovo vključevanje v model pa je odvisno od tega, ali se njihovo gibanje ujema z globalnim gibanjem objekta. Yang in Wu [59] uporabljata Harris-Laplaceov detektor zanimivih področij, kot pa je bilo že omenjeno v razdelku 2.1.4, zaznane stabilne regije najprej dodata v množico kandidatov, šele kasneje pa tudi v sam vizualni model.

Tran in Davis [55] predpostavljata, da lokalni elementi slike, ki so blizu objekta in se gibljejo konsistentno z objektom, z veliko verjetnostjo pripadajo objektu. V njunem primeru ima vizualni model dve komponenti. Prva je množica lokalnih SIFT opisnikov, ki se dejansko uporabljajo za sledenje. Regije, ki se opišejo z opisniki, so določene z MSER detektorjem stabilnih regij. Za razliko od vizualnih modelov, opisanih v razdelku 2.1.4, se pri samem iskanju ujemanja ne uporablja geometrijskega modela, ampak se pari opisnikov v zaporednih slikah video zaporedja določijo na podlagi najboljšega ujemanja opisnikov. Druga komponenta je *polje ujemanja*, verjetnostna porazdelitev, ki opisuje, kako dobro se posamezne regije v sliki ujemajo z gibanjem objekta (Slika 2.13). Polje se osvežuje z primerjavo spremembe slike z globalnim modelom gibanja, ki je pridobljen iz ujemanja SIFT značilnic v slikah zaporednih časovnih korakov. Krog je zaključen z osvežitvijo množice značilnic na podlagi področja, ki ga določa polje ujemanja (področja posameznih značilnic so določena z MSER detektorjem stabilnih področij).

Problemi pristopov, ki temeljijo na stabilnih regijah, so bili omenjeni že v razdelku 2.1.4. Gre za problem ponovljivosti regij ter njihovo detekcijo na objektih brez potrebnih vizualnih lastnosti.



Slika 2.13: Prikaz polja ujemanja.

Ilustracija polja ujemanja, kakor je ta koncept uporabljen v [55].

■

V tem poglavju so bili predstavljeni obstoječi pristopi h konstrukciji sledilnikov s poudarkom na njihovih vizualnih modelih. Za izbrane vizualne modele lahko rečemo, da so pomembno vplivali na naše lastno delo na tem področju. Opazimo lahko kar nekaj skupnih lastnosti, še posebej v okviru pristopov k združevanju lokalne in globalne informacije. V razdelku 3.2 je le-ta formalizirana kot iskanje optimuma cenovne funkcije, v okviru katere se združita vizualno ujemanje na lokalnem nivoju posameznih značilnic ter ujemanje z globalnim geometrijskim modelom. Geometrijski modeli se razlikujejo glede fleksibilnosti v okviru optimizacije ter sposobnosti prilagajanja se novim razmeram. Večina modelov, ki temeljijo na geometrijskih modelih, pa je omejena, kar se tiče posodabljanja le-tega v smislu dodajanja novih značilnic ali brisanja starih. Z izjemo sledilnika, predstavljenega v [37], ter sledilnikov, ki temeljijo na stabilnih regijah [59, 60], taki vizualni modeli niso sposobni posodabljati informacije o lokalnem izgledu. Tudi v primeru, ko je tako posodabljanje mogoče, pa le-to temelji na statičnih informacijah o izgledu objekta, kakršen je na primer primer jate značilnic [30].

Poglavje 3

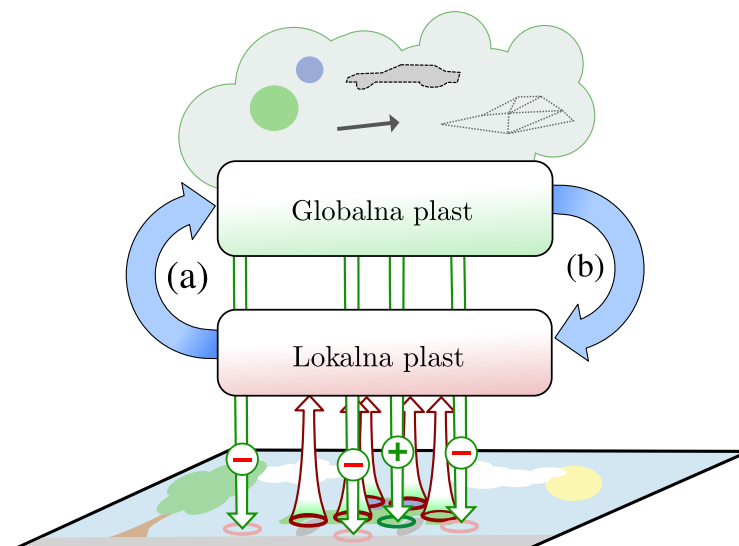
Dvoplastni vizualni model

V tem poglavju bomo predstavili vizualni model, ki je osrednja tema te magistrske naloge. Cilj v razvoju vizualnega modela je bil razširiti koncept množice lokalnih značilnic, temelječih na lokalnih zaplatah ter kombiniranih z informacijsko bogato geometrijsko strukturo, podobno tisti, ki je bila uporabljena v [42, 2, 10]. Razširitev primarno temelji na zmožnosti posodabljanja vizualnega modela z odstranjevanjem značilnic, dodajanjem novih značilnic [30, 37] ter na več-modalnem učenju globalnega izgleda objekta.

Predlagani vizualni model *hierarhično* [9, 33] razdeli informacijo o izgledu v dve plasti, v *globalno* in *lokalno* plast, kot je to prikazano v Sliki 3.1. Lokalna oziroma spodnja plast je sestavljena iz množice lokalnih značilnic, vsaka izmed katerih opisuje lokalni izgled dela objekta. Globalna oziroma zgornja plast pa vsebuje globalni opis izgleda objekta. Le-ta se v sledenju uporablja posredno in preko ujemanja različnih modalnost izgleda določa območje, ki naj bi v sliki pripadalo objektu. Ko se izgled objekta med sledenjem spreminja, se obe plasti vizualnega modela vzajemno osvežujeta. Posodabljanje je *selektivno*, kar pomeni, da v njem sodelujejo le deli opisa v posamezni plasti, ki so ocenjeni kot zanesljivi. Tak konzervativni pristop k posodabljanju zmanjšuje možnost posodabljanja delov opisa, ki posodabljanja ne potrebujejo. Posodabljanje takih delov bi namreč lahko negativno vplivalo na robustnost sledenja. Obenem pa tak pristop še vedno zagotavlja fleksibilnost modela ter sposobnost le-tega, da se prilagaja spremembam v izgledu objekta. Osnovne ideje hierarhije temeljijo na naših dognanjih, predstavljenih v članku [56].

Slika 3.1 ilustrira shematično predstavitev strukture vizualnega modela ter postopka osveževanja le-tega. Osveževanje globalne plasti poteka z obdelavo podatkov o lokalnem izgledu posameznih področij, ki jih pokrivajo lokalne značilnice. Pri tem niso obravnavane vse lokalne značilnice ampak samo tiste, za katere smo prepričani, da še dovolj zanesljivo sledijo objektu. Osveževanje lokalnega vizualnega modela se začne s posodobitvijo položajev posameznih lokalnih značilnic. Cilj posodobitve položajev je maksimizirati ujemanje opisa značilnice z lokalnim opisom slike na danem položaju, pri čemer pa se poskuša zadostiti tudi omejitvam geometrijskega modela. Nato so iz množice značilnic odstranjene vse značilnice, za katere iz njihovega obnašanja lahko sklepamo, da niso več zanesljive oziroma, da ne pokrivajo več območja, ki pripada objektu. Zadnji korak osveževanja lokalne plasti vključuje iskanje potencialnih novih lokalnih značilnic. To iskanje poteka z uporabo

globalne informacije o izgledu objekta, s katero se določijo območja v sliki, ki (na podlagi te informacije) pripadajo objektu.



Slika 3.1: Shematična predstavitev postopka osveževanja.

(a) - izbira in obdelava pridobljenih podatkov o lokalnem izgledu za osveževanje globalnega vizualnega modela. (b) - uporaba globalnega vizualnega modela služi za vodenje posameznih lokalnih značilnic ter dodajanje novih značilnic in odstranjevanje starih.

■

V nadaljevanju poglavja bo v 3.1 najprej bolj podrobno definirano stanje vizualnega modela. V razdelku 3.2 bo predlagana optimizacijska funkcija, ki se v okviru vizualnega modela uporablja za iskanje ujemanja v sliki ter metoda, s katero učinkovito rešimo optimizacijski problem. V razdelku 3.3 so opisani mehanizmi, preko katerih se z novimi informacijami osvežuje množica lokalnih značilnic, v razdelku 3.4 pa metode za osveževanje globalnega opisa objekta. Na koncu poglavja, v razdelku 3.5, je opisana še umestitev vizualnega modela v sledilnik ter implementacija lokalnih značilnic, ki so bile uporabljene v okviru eksperimentov.

3.1 Stanje vizualnega modela

Prva naloga, ki jo moramo opraviti, je formalizacija stanja vizualnega modela. Ker se stanje v vsakem časovnem koraku spreminja, je njegoa definicija odvisna od indeksa časovnega koraka t . Tako je stanje vizualnega modela v časovnem koraku t predstavljeno kot par

$$\mathcal{V}_t = \{\mathbf{X}_t, \mathcal{A}_t\}, \quad (3.1)$$

kjer \mathbf{X}_t definira stanje lokalne plasti, \mathcal{A}_t pa stanje globalne plasti vizualnega modela. Kot smo že omenili v uvodu v poglavje, lokalno plast sestavlja množica lokalnih značilnic. Tako je \mathbf{X}_t definiran kot množica parov vektorjev. Prva komponenta para je položaj značilnice, kot drugo komponento para pa uvedemo utež značilnice, ki nakazuje ocenjeno zanesljivost značilnice. Ostale lastnosti značilnice na tem mestu niso pomembne, saj gre za abstrakcijo, ki omogoča uporabo različnih vrst lokalnih opisov. Podrobnosti o uporabljenih značilnicah so predstavljene v razdelku 3.5.2. Formalno je množica \mathbf{X}_t definirana kot

$$\mathbf{X}_t = \{\langle \mathbf{x}_t^i, w_t^i \rangle \mid i \in L_t\}, \quad (3.2)$$

kjer \mathbf{x}_t^i predstavlja vektor položaja lokalne značilnice v 2D prostoru slike, $w_t^i \in [0, 1]$ pa predstavlja utež značilnice ob času t . L_t je množica indeksov vseh aktivnih lokalnih značilnic ob času t . Zaradi priročnosti definiramo tudi $\bar{\mathbf{x}}_t$ kot uteženo povprečje položajev posameznih značilnic ter $\mathbf{r}_t^i = \mathbf{x}_t^i - \bar{\mathbf{x}}_t$ kot relativni položaj značilnice glede na to povprečje. Poleg tega uporabljamo simbol $s(i)$ za določitev časovnega koraka, v katerem se je značilnica z indeksom i prvič pojavila v množici značilnic vizualnega modela. Z oznako $\mathbf{x}_i^{s(i)}$ tako na primer označimo položaj značilnice z indeksom i v prvem časovnem koraku njene pojavitve v vizualnem modelu, torej

$$s(i) = \min t : i \in L_t. \quad (3.3)$$

Globalni vizualni model, ki je v (3.1) predstavljen s simbolom \mathcal{A}_t , vključuje informacije o globalnem izgledu objekta, na podlagi katerih se določajo položaji novih lokalnih značilnic. \mathcal{A}_t je, po zgledu iz [3, 32, 11] sestavljen iz več vizualnih modalnosti

$$\mathcal{A}_t = \{\mathcal{M}_t^i \mid i \in 1 \dots\}. \quad (3.4)$$

Vsaka izmed modalnost uporablja svojo specifično obliko predstavitve informacije. V okviru magistrske naloge smo uporabili tri modalnosti: barvo, gibanje ter prostorsko enovitost. Tako \mathcal{A}_t zapišemo kot končno množico treh elementov

$$\mathcal{A}_t = \{C_t, M_t, S_t\}, \quad (3.5)$$

kjer C_t predstavlja opis modalnosti barve, M_t opis modalnosti gibanja ter S_t opis modalnosti prostorske enovitosti. Barva in gibanje sta zelo dobro poznana pojma v računalniškem vidu, njune lastnosti pa so se izkazale za zelo uporabne v veliko praktičnih aplikacijah, tudi v sledenju [34]. Prostorska enovitost je posebna modalnost, s katero se zagotavlja kompaktno prostorsko strukturo množice lokalnih značilnic v okviru dolgoročnega sledenja. Vse tri modalnosti so bolj podrobno opisane v razdelku 3.4.

Ob tem je treba še enkrat poudariti, da je naš cilj splošen in ne predpostavlja uporabe točno določenega števila vizualnih modalnosti. Hoteli smo definirati okvir, ki bi omogočal poljubno dodajanje novih modalnosti v globalni model. S tem namenom smo za formalizacijo stanja globalne plasti uporabili zapis, ki temelji na verjetnostnih porazdelitvah. Za vsako modalnost se v vsakem časovnem koraku generira verjetnostna porazdelitev v prostoru slike. Za vsak slikovni element se določi verjetnost, da na podlagi ujemanja z modelom modalnosti pripada objektu. Kot v [32] predpostavljamo, da so posamezne modalnosti

med seboj statistično neodvisne, zato lahko zapišemo združeno verjetnostno porazdelitev, da slikovni element \mathbf{x} pripada objektu, kot

$$p(\mathcal{A}_t|\mathbf{x}) = p(C_t, M_t, S_t|\mathbf{x}) = p(C_t|\mathbf{x}) p(M_t|\mathbf{x}) p(S_t|\mathbf{x}). \quad (3.6)$$

V (3.6) $p(C_t|\mathbf{x})$ predstavlja verjetnostno porazdelitev, pridobljeno z uporabo barvnega modela objekta, $p(M_t|\mathbf{x})$ predstavlja ujemanje z gibanje objekta ter $p(S_t|\mathbf{x})$ ujemanje z modelom prostorske enovitosti. Skupna verjetnostna porazdelitev $p(\mathcal{A}_t|\mathbf{x})$ torej vsebuje informacijo vseh treh modalnost. Na podlagi vzorčenja te skupne verjetnostne porazdelitve vseh modalnosti potem lahko določimo najbolj ugodne položaje novih značilnic.

3.2 Iskanje ujemanja množice lokalnih značilnic

Za uporabo vizualnega modela v sledenju je treba definirati kriterij ujemanja le-tega z informacijo na sliki. Visokodimenzionalna optimizacijska funkcija je eden izmed načinov formalne definicije iskanja ujemanja, ki smo ga omenili že v poglavju 2.

3.2.1 Optimizacijska funkcija ujemanja lokalnih značilnic

V tej magistrski nalogi smo iskanje ujemanja množice značilnic s sliko formulirali kot optimizacijski problem iskanja ujemanja posameznih lokalnih značilnic ter netodega geometrijskega modela. Trenutni položaji vseh značilnic sestavljajo $2|L_t|$ -dimenzionalni vektor stanja, ki ga označimo kot $\widehat{X}_t = (\widehat{x}_t^i)_{i \in L_t}$. Geometrijski model je definiran kot množica položajev lokalnih značilnic v prejšnjem časovnem koraku. Naša formalizacija optimizacijskega problema temelji na optimizacijski funkciji, ki združi med seboj odvisne komponente v uteženo povprečje

$$\bar{\xi}(\widehat{X}_t) = \frac{1}{\sum_{i \in L_t} w_t^i} \sum_{i \in L_t} w_t^i \xi(\widehat{\mathbf{x}}_t^i, i). \quad (3.7)$$

Komponente (3.7) so optimizacijske funkcije posameznih značilnic, ki so definirane kot

$$\xi(\widehat{\mathbf{x}}_t^i, i) = S(\widehat{\mathbf{x}}_t^i, i, t) e^{-\lambda_G \|\widehat{\mathbf{x}}_t^i - \mathbf{f}(\widehat{\mathbf{x}}_t^i)\|}, \quad (3.8)$$

kjer funkcija $S(\mathbf{x}, i, t)$ predstavlja lokalno ujemanje značilnice z indeksom i s sliko v časovnem koraku t na položaju \mathbf{x} . Vrednost $S(\mathbf{x}, i, t) = 1$ pomeni, da se lokalni vizualni model značilnice z indeksom i ter lokalna okolica točke \mathbf{x} v sliki popolnoma ujemata, proti 0 padajoče vrednosti pa nakazujejo povečujočo se razliko med vizualnim modelom ter področjem v sliki. Funkcija $\mathbf{f}(\widehat{\mathbf{x}}_t^i)$ določa položaj, ki ga za lokalno značilnico z indeksom i predlaga okolica oziroma druge točke glede na medsebojne geometrijske odnose. Konstanta λ_G uravnava vpliv geometrijskih omejitev med lokalnimi značilnicami na optimizacijo, lahko pa bi rekli, da posredno uravnava tudi togost samega geometrijskega modela. Na opisani način je geometrijski model na fleksibilen način vpleten v iskanje optimalnega položaja značilnice.

Na tem mestu se pojavi vprašanje, kako definirati funkcijo $\mathbf{f}(\widehat{\mathbf{x}}_t^i)$. Določitev le-te je seveda odvisna od strukturnih lastnosti objekta. Ker v splošnem teh lastnosti ne moremo

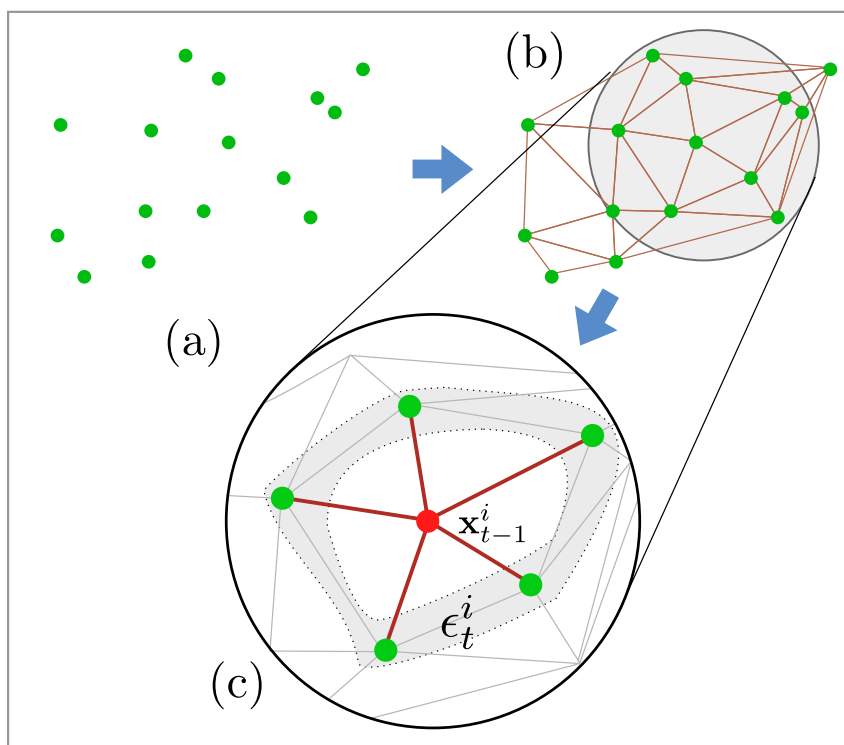
predvideti, v našem vizualnem modelu uporabljamo splošen *ad-hoc* strukturni model, ki predpostavlja, da med sosednimi značilnicami veljajo osnovne omejitve afine geometrije v ravnini. S tem namenom definiramo okolico lokalne značilnice z indeksom i kot podmnožico značilnic, ki zadoščajo naslednjemu pogoju

$$\epsilon_t^i = \{j \mid \langle \mathbf{x}_{t-1}^i, \mathbf{x}_{t-1}^j \rangle \in DT(X_{t-1})\}, \quad (3.9)$$

kjer $DT(X_{t-1})$ označuje množico povezav grafa, določenega z uporabo Delaunay-eve triangulacije nad množico lokalnih značilnic X_{t-1} . Postopek določitve okolice ϵ_t^i je ilustriran tudi na Sliki 3.2. Z uporabo značilnic iz ϵ_t^i lahko ocenimo lokalno transformacijo z določitvijo afine transformacijske matrike na podlagi položajev značilnic v prejšnjem časovnem koraku ter njihovimi trenutnimi položaji med optimizacijo. Določitev matrike zapišemo kot minimizacijski problem

$$\mathbf{A}_t^i = \operatorname{argmin}_{\mathbf{A}} \sum_{j \in \epsilon_t^i} \|\mathbf{A} \mathbf{r}_{t-1}^j + \bar{\mathbf{x}}_{t-1} - \hat{\mathbf{x}}_t^j\|^2. \quad (3.10)$$

Ta transformacijska matrika je nato uporabljena v funkciji $\mathbf{f}(\hat{\mathbf{x}}_t^i) = \mathbf{A}_t^i (\hat{\mathbf{x}}_t^i - \bar{\mathbf{x}}_{t-1}) + \bar{\mathbf{x}}_{t-1}$.



Slika 3.2: Prikaz določitve ϵ_t^i .

(a) - položaji posameznih značilnic v časovnem koraku $t - 1$ (b) - za točke je določen graf z metodo Delaunay-eve triangulacije (c) - okolica značilnice z indeksom i vsebuje značilnice, ki si v grafu z le-to delijo povezavo

3.2.2 Optimizacija z uporabo križne entropije

Problem visokodimenzionalne optimizacije je, da se postopek lahko hitro ustavi v lokalnem optimumu, ki je daleč od iskanega globalnega optimuma. Da bi učinkovito rešili ta problem, smo optimizacijo razdelili v dva koraka. Prvi korak se imenuje *globalna optimizacija*, v okviru katerega se išče optimalno ujemanje toge konstelacije delov. Taka konstelacija se lahko premika le v prostoru petih dimenzij, ki opisujejo translacijo, rotacijo ter skaliranje v dvodimenzionalnem prostoru. Drugi korak optimizacije je *lokalna optimizacija*. Le-ta obravnava množico lokalnih značilnic kot elastično strukturo, katere prostorske spremembe so s strani geometrijskih relacij med posameznimi značilnicami le delno omejene.

Globalna optimizacija

Globalna optimizacija nam omogoča, da omejimo optimizacijski prostor, oziroma se osredotočimo le na pod-prostor izvornega problema. Ker so razmaki med časovnimi koraki majhni, lahko na tak način aproksimiramo tudi gibanje objektov, ki se v splošnem ne premikajo na podlagi tega preprostega modela. Tak pristop nam omogoča, da na učinkovit način poiščemo dober začetni položaj za visokodimenzionalno optimizacijo.

Ker je afina transformacija uporabljena tudi za izračun funkcije $f(\hat{\mathbf{x}}_t^i)$, razdalja med le-to ter trenutno pozicijo lokalne značilnice z indeksom i , uporabljena v enačbi (3.8) ostane v vsakem primeru 0 ter jo lahko odstranimo iz optimizacijske funkcije (3.7), ki jo tako spremenjeno zapišemo kot

$$\bar{\xi}_G(\mathbf{a}) = \frac{1}{W_t} \sum_{i=1}^N w_i S_t^i(\mathbf{A}_a \cdot \mathbf{r}_t^i + \bar{\mathbf{x}}_t), \quad (3.11)$$

kjer $\mathbf{a} = [t_x, t_y, r, s_x, s_y]$ označuje vektor s petimi optimizacijskimi parametri, ki določajo matriko afine transformacije \mathbf{A}_a kot

$$\mathbf{A}_a = \begin{bmatrix} \cos(r)s_x & -\sin(r) & t_x \\ \sin(r) & \cos(r)s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.12)$$

Za optimizacijo funkcije $\bar{\xi}_G(\mathbf{a})$ uporabimo metodo križne entropije. Teoretične osnove metode so opisane v dodatku A.1. Iterativni postopek optimizacije je prikazan v Algoritmu 1. Za razliko od klasične metode križne entropije je v naši metodi dodan ustavitveni pogoj, ki optimizacijo prekine, če se le-ta ne zaključi dovolj hitro oziroma v n_{max} korakih. V praksi se sicer izkaže, da je zadostna konvergenca v veliki večini primerov dosežena že v manj kot petih korakih. Stopnjo konvergence ocenjujemo z determinanto kovariančne matrike Σ . V 1 z $elite_i$ označimo izbiro N_G^{elite} indeksov najboljših vzorcev glede na optimizacijsko funkcijo $\bar{\xi}_G$, kot je to opisano v dodatku A.1. Ko je postopek optimizacije končan, je afina transformacija, ki jo določajo optimalni parametri, aplicirana na množico lokalnih značilnic.

Lokalna optimizacija

Algoritem 1 Postopek globalne optimizacije na podlagi metode križne entropije.

$\Sigma \leftarrow \Sigma_G, \mu \leftarrow \mathbf{0}, n \leftarrow 0, S^* = 0, \gamma = 0$

Dokler $n < n_{max}$ ponavljaj:

1. $n \leftarrow n + 1, N \leftarrow N_G$
2. $\mathbf{a}_1, \mathbf{a}_2 \dots \mathbf{a}_N \sim \mathcal{G}(\mu, \Sigma)$
3. $E = \{e_1, \dots, e_{N_G^{elite}}\} \leftarrow \text{elite}_i(\overline{\xi}_G(\mathbf{a}_i))$
4. če $S^* > \overline{\xi}_G(\mathbf{a}_{e_1}) \wedge \gamma > \overline{\xi}_G(\mathbf{a}_{e_{N_G^{elite}}})$, potem povečaj N in pojdi na korak 2.
5. $W \leftarrow \sum_{e \in E} \overline{\xi}_G(\mathbf{a}_e)$
 $\mu \leftarrow \frac{1}{W} \sum_{e \in E} \overline{\xi}_G(\mathbf{a}_e) \mathbf{a}_e$
 $\Sigma \leftarrow \frac{1}{W} \sum_{e \in E} \overline{\xi}_G(\mathbf{a}_e) (\mathbf{a}_e - \mu)(\mathbf{a}_e - \mu)^T$
6. $S^* \leftarrow \overline{\xi}_G(\mathbf{a}_{e_1}), \gamma \leftarrow \overline{\xi}_G(\mathbf{a}_{e_{N_G^{elite}}})$
7. če $\det(\Sigma) < t_G$, potem končaj.

$\hat{\mathbf{x}}_t^i \leftarrow A_\mu \mathbf{r}_t^i + \bar{\mathbf{x}}_t$ za $i \in L_t$

Fazi globalne optimizacije sledi faza lokalne optimizacije, kjer v optimizaciji v večji meri upoštevamo lokalno ujemanje ter dovolimo, da geometrijska struktura tudi deformira. Ideja v tem primeru je, da položaj vsake lokalne značilnice optimiziramo posebej z uporabo optimizacijske funkcije $\xi_i(\hat{\mathbf{x}}_t^i)$. Tudi tu uporabimo metodo križne entropije, vendar namesto zaporedne optimizacije podproblemov le-te prepletamo na nivoju iteracij optimizacije. S tem zagotovimo nizko dimenzijo problema (namesto enega $2n$ -dimenzionalnega problema sedaj obravnavamo n 2-dimenzionalnih), ob tem pa še vedno obravnavamo omejitve, ki jih predstavljajo povezave med posameznimi lokalnimi značilnicami.

Algoritem 2 prikazuje bolj podrobno skico tega postopka. Ustavitveni pogoji so podobni kot pri Algoritemu 1, le da v tem primeru položaje posameznih značilnic nehamo optimizirati posamično, celoten postopek pa se zaključi, ko konvergira vseh $|L_t|$ problemov. Z elite_i v Algoritemu 4 označimo izbiro N indeksov najboljših vzorcev glede na optimizacijsko funkcijo $\xi(y_i, j)$, kot je to opisano v dodatku A.1.

3.3 Osveževanje množice lokalnih značilnic

Določena množica lokalnih značilnic se lahko spreminjanju izgleda objekta s spreminjanjem lokacij posameznih značilnic prilagaja le določen čas. Nato se izgled objekta enostavno spremeni do te mere, da nekatere značilnice v sliki nimajo več dovolj dobrega ujemanja. V takem primeru pride do problema drsenja, ko ima značilnica na voljo veliko število podobno dobrih ujemanj. Take značilnice imenujemo *problematične značilnice*, saj obstaja nevarnost, da se njihova nezanesljivost prenese na celoten vizualni model. Zato

Algoritem 2 Postopek lokalne optimizacije na podlagi metode križne entropije.

$\Sigma_i \leftarrow \Sigma_L, n \leftarrow 1, L \leftarrow L_t$

Dokler $L \neq \emptyset \wedge n < n_{max}$ ponavljaj:

1. Za $j \in L$:

- (a) $\mathbf{A}_t^j = \min_{\mathbf{A}} \sum_{k \in e_t^j} \|\mathbf{A} \mathbf{r}_{t-1}^k + \bar{\mathbf{x}}_{t-1} - \hat{\mathbf{x}}_t^k\|^2$
- (b) $\mathbf{y}_1, \mathbf{y}_2 \dots \mathbf{y}_N \leftarrow \mathcal{G}(\mathbf{x}_t^j, \Sigma_j)$
- (c) $E \leftarrow \text{elite}_i(\xi(\mathbf{y}_i, j))$
- (d) $W \leftarrow \sum_{e \in E} \xi(\mathbf{y}_e, j)$
 $\hat{\mathbf{x}}_t^j \leftarrow \frac{1}{W} \sum_{e \in E} \xi(\mathbf{y}_e, j) \mathbf{y}_e$
 $\Sigma_j \leftarrow \frac{1}{W} \sum_{e \in E} \xi(\mathbf{y}_e, j) (\mathbf{y}_e - \hat{\mathbf{x}}_t^j)(\mathbf{y}_e - \hat{\mathbf{x}}_t^j)^T$
- (e) če $\det(\Sigma_j) < t_L$, potem $L \leftarrow L/\{j\}$.

2. $n \leftarrow n + 1$

jih je treba iz množice aktivnih značilnic čim prej odstraniti ter nadomestiti z novimi, bolj zanesljivimi značilnicami.

3.3.1 Posodabljanje uteži značilnic

V opisanem vizualnem modelu za detekcijo takih značilnic uporabljamo uteži značilnic, ki smo jih omenili že v razdelku 3.1. Glavni pomen uteži značilnic je določanje njihove pomembnosti v okviru vizualnega modela. Če je utež visoka, to kaže na veliko pomembnost značilnice oziroma strinjanje z večino značilnic, če pa je utež prenizka, to smatramo za znak, da je značilnica problematična in izstopa. Na podlagi tega sklepamo, da jo lahko odstranimo.

Da bi bil ta vpliv posodabljanja množice značilnic čimbolj tekoč, vizualni model postopno spreminja pomembnost posamezne značilnice z indeksom i z regulacijo njene uteži w_t^i . Uteži spreminjajo tri pravila, ki obravnavajo različne aspekte preteklega stanja značilnice glede na celotno množico. Prvo pravilo je *moč lokalnega ujemanja*, drugo *samosvoje gibanje*, tretje pa *drsenje stran od večine*. Skupni mehanizem posodabljanja je predstavljen za opisom posameznih pravil.

Moč lokalnega ujemanja

Če značilnica v danem časovnem koraku nima dobrega ujemanja v sliki, se njena utež zmanjša. To pomeni, da značilnice, ki v več zaporednih korakih ne najdejo več dela objekta, ki ga opisujejo (zaradi zakritja ali deformacije), najprej izgubijo vpliv na druge značilnice v množici, nato pa izpadejo iz nje. Po drugi strani se značilnicam z izrazito dobrim ujemanjem utež poveča. Kaj je slabo in kaj dobro ujemanje, je določeno z pragovnima konstantama M_B ter M_G na naslednji način:

$$\begin{aligned} r_B^S(i) &= S_t^i(\mathbf{x}_t^i) < M_B \quad \text{in} \\ r_G^S(i) &= S_t^i(\mathbf{x}_t^i) > M_G. \end{aligned} \quad (3.13)$$

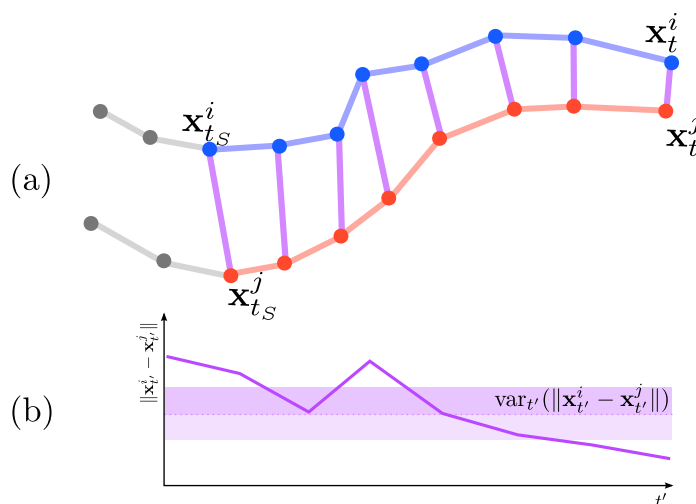
Samosvoje gibanje

Drugi mehanizem prilagajanja uteži temelji na iskanju odstopajočih značilnic glede na nedavno zgodovino gibanja le-teh. S tem namenom opazujemo gibanje posameznih značilnic v časovnih korakih $t_s \dots t$, kjer t_s označuje začetni časovni korak obravnavane zgodovine. Radi bi ločili večji del značilnic, ki imajo v tem časovnem okviru podobno gibanje, od tistih, katerih gibanje opazno izstopa.

Za obravnavanje razdalje med dvema trajektorijama gibanja lahko uporabimo različne metrike. V okviru te magistrske naloge je bila uporabljena metrika, ki temelji na variaciji razlik posameznih točk [8]

$$\delta(\{\mathbf{x}_{t'}^i\}, \{\mathbf{x}_{t'}^j\}) = 1 - \frac{1}{1 + \text{var}_{t'}(\|\mathbf{x}_{t'}^i - \mathbf{x}_{t'}^j\|)}, \quad (3.14)$$

kjer $\{\mathbf{x}_{t'}^i\}$ in $\{\mathbf{x}_{t'}^j\}$ predstavljata obe zaporedji točk na intervalu $t' = t_s \dots t$ ter $\text{var}_{t'}(\|\mathbf{x}_{t'}^i - \mathbf{x}_{t'}^j\|)$ predstavlja variacijo razdalj med pari točk obeh zaporedij glede na t . Ilustracija izračuna variance je prikazana na Sliki 3.3.



Slika 3.3: Ilustracija izračuna variance razdalj med točkami dveh trajektorij.

(a) - za danih zadnjih $t - t_s$ časovnih korakov se izračunajo razdalje med soležnimi točkami trajektorij (b) - na podlagi le-teh je izračunana varianca razdalj, ki je potem uporabljena v enačbi (3.14)

Določanja značilnic s samosvojim gibanjem se lotimo z algoritmom, ki temelji na metodi soglasja naključnih vzorcev (RANSAC) [16], opisani v dodatku A.2. V koraku k za naključno izbrano podmnožico značilnic $S_k \subset L_t$ zgradimo model njihovega skupnega gibanja kot

$$\mathbf{x}_{t'}^{S_k} = \frac{1}{W_k} \sum_{i \in S_k} w_{t'}^i \mathbf{x}_{t'}^i \quad t' = t_S \dots t, \quad (3.15)$$

kjer W_k označuje vsoto uteži značilnic iz množice S_k . Kot je to formulirano v enačbi (3.15), je model skupnega gibanja prav tako zaporedje točk, ki predstavlja uteženo oceno gibanja objekta. Na podlagi tako dobljenega modela lahko določimo značilnice, ki niso bile uporabljene za grajenje modela, vendar pa so mu dovolj podobne. Če je razdalja med zaporedjem točk modela ter zaporedjem točk značilnice $i \in L_t/S_k$ manjša kot največja razdalja med zaporedjem točk modela in zaporedjem točk poljubne značilnice iz S_k , značilnico z indeksom i dodamo v posplošeni model S_k^* .

Zadnja stvar, ki jo moramo definirati za metodo soglasja naključnih vzorcev, je metrika napake posplošenega modela. Le-to merimo kot največjo razdaljo

$$E(S_k^*) = \max_{i \in L_t} (\delta(\{\mathbf{x}_{t'}^{S_k^*}, \{\mathbf{x}_{t'}^i\})), \quad (3.16)$$

kjer trajektorijo $\{\mathbf{x}_{t'}^{S_k^*}\}$ izračunamo po enačbi (3.15). Funkcija δ označuje metriko razdalje med trajektorijama $\{\mathbf{x}_{t'}^{S_k^*}\}$ in $\{\mathbf{x}_{t'}^i\}$.

Pri tem velja poudariti, da je naš cilj drugačen od tipičnega cilja metode soglasja naključnih vzorcev. Naš namen namreč ni sama določitev čim boljšega modela gibanja, temveč iskanje značilnic, ki od tega modela odstopajo, kar je v tipičnem scenariju uporabe metode soglasja naključnih vzorcev "stranski produkt". Rezultat pravila tako temelji na prisotnosti značilnice v končnem modelu

$$\begin{aligned} r_B^R(i) &= i \notin S^* & \text{in} \\ r_G^R(i) &= i \in S^*, \end{aligned} \quad (3.17)$$

kjer S^* označuje končni model.

Drsenje stran od večine

Zadnji uporabljeni mehanizem zmanjšuje uteži značilnicam, ki zdrsijo predaleč od večine ter tako deluje kot varovalka, ki ohranja kompaktnost množice. Za vsako značilnico določimo mediano razdalje le-te do vseh drugih značilnic v množici. Uteži vseh značilnic, katerih izračunana mediana razdalje presega prag T_S , so zmanjšane. To formalno zapišemo kot

$$r_B^D(i) = \text{median}_{j \in L_t} (\|\mathbf{x}_t^j - \mathbf{x}_t^i\|) > T_S. \quad (3.18)$$

Skupna formula posodabljanja uteži za posamezno značilnico združi vsa tri pravila v enotno obliko

$$w_{t+1}^i = \begin{cases} w_t^i e^{-R_B/2}, & \text{če } r_B^S \vee r_B^R \vee r_B^D \\ \min(w_t^i + 0.1, 1), & \text{če } r_G^S \wedge r_G^R \\ w_t^i, & \text{drugače} \end{cases}, \quad (3.19)$$

kjer R_B predstavlja vsoto logičnih vrednosti simbolov r_B^S , r_B^R in r_B^D . Na ta način značilnice, ki kršijo več pravil, hitreje zgubijo pomembnost v okviru vizualnega modela. Formula je bila določena na podlagi eksperimentov.

3.3.2 Posodabljanje množice lokalnih značilnic

Posodabljanje množice značilnic je sestavljeno iz treh korakov: odstranjevanja problematičnih lokalnih značilnic, dodajanja novih lokalnih značilnic ter zlivanje lokalnih značilnic.

Odstranjevanje problematičnih lokalnih značilnic

Če utež značilnice zaradi uporabe enačbe (3.19) pade pod določen prag, ki ga označimo z T_R , se lokalna značilnica smatra za nepomembno oziroma škodljivo za nadaljnjo uporabo in je odstranjena iz množice značilnic.

Dodajanje novih lokalnih značilnic

Pri dodajanju lokalnih značilnic v vizualni model moramo nasloviti dve pomembni vprašanji: (a) koliko značilnic lahko dodamo ter (b) kam naj dodamo te značilnice. Število značilnic je določeno na podlagi dinamično spremenljivega praga, ki preprečuje, da bi se število značilnic spremenilo prehitro, obenem pa še vedno omogoča, da se vizualni model lahko prilagaja spremembam velikost objekta na sliki. Želeno število značilnic se ob koncu vsakega časovnega koraka prilagodi trenutnemu stanju po enačbi

$$N_{t+1}^T = (1 - \lambda_N)N_t^T + \lambda_N|L_t|, \quad (3.20)$$

kjer N_t^T predstavlja prag v trenutnem časovnem koraku, $\lambda_N \in [0, 1]$ konstanto, ki določa hitrost prilagajanja, $|L_t|$ pa predstavlja število značilnic v trenutni množici značilnic (po opravljenih korakih odstranjevanja problematičnih značilnic ter dodajanja novih značilnic). Ta prag služi kot osnova za izračun želenega števila značilnic. Ob tem je treba poudariti, da to število ni vedno doseženo, daj se lahko izkaže, da ustreznih novih položajev v sliki ni dovolj. Ponavadi se to zgodi, ko se objekt vizualno zmanjša. V tem primeru se prag začne postopoma nižati ter s tem prepreči, da bi se število značilnic v bližnji prihodnosti skokovito povečalo zaradi napak v vizualnem modelu.

Vizualni model, predstavljen v tej magistrski nalogi, položaje novih značilnic določi na podlagi informacij o globalnem izgledu objekta oziroma na podlagi verjetnostnih porazdelitev, ki so določene na podlagi ujemanja posameznih slikovnih elementov slike v danem časovnem koraku z informacijo o izgledu. Bolj podrobno so same modalnosti izgleda objekta, ki jih vizualni model spremlja, opisane v razdelku 3.4. Uteži na novo dodanih značilnic so na začetku postavljene nizko, blizu praga za odstranitev, saj obstaja možnost, da so bile postavljene na napačno mesto. Kot take so torej obravnavane kot manj zanesljive, vendar lahko preko mehanizmov spreminjanja uteži, opisanih v razdelku 3.3.1, s časom pridobijo na vplivu. Na ta način lahko vizualni model počasi premika pozornost iz starih značilnic na novejši, pri čemer se minimizira verjetnost, da bi slabo postavljene nove značilnice pokvarile celoten vizualni model.

Zlivanje lokalnih značilnic

V primeru, ko posamezne značilnice med sledenjem, zaradi vizualnega manjšanja objekta ali deformacije, zdrsnejo preblizu ena drugi, jih združimo v eno samo značilnico. Glavni razlog za to je, da želimo ohraniti dovolj prostora med sosednjimi značilnicami v geometrijskem modelu, saj je od tega odvisna deformacijska sposobnost modela. Poleg tega je od zadostne razdalje med značilnicami odvisna tudi numerična stabilnost nekaterih uporabljanih algoritmov. Formalno gledano zlivanje sestavlja odstranjevanje vseh vpletenih značilnic ter dodajanje nove značilnice. Položaj nove lokalne značilnice je izračunan kot uteženo povprečje položajev odstranjenih značilnic, njena utež pa kot povprečna utež le-teh.

3.4 Posodabljanje globalne plasti

Kot smo omenili že v razdelku 3.1, globalno plast sestavljajo tri modalnosti. Vsaka modalnost vsebuje svoj specifičen zapis izgleda ter drugačen mehanizem posodabljanja le-tega.

Barva

Barvni model je predstavljen v obliki dveh ločenih dvo-dimenzionalnih histogramov, ki opisujeta odtenek (ang. *hue*) ter nasičenost (ang. *saturation*) barve.

$$C_t = \{\mathcal{H}_t^F, \mathcal{H}_t^B\}. \quad (3.21)$$

Histogram \mathcal{H}_t^F modelira *barvo objekta*, histogram \mathcal{H}_t^B pa *barvo okolice*. Z uporabo povratne projekcije (ang. *backprojection*) obeh histogramov na sliko dobimo verjetnostni porazdelitvi $p(\mathbf{x}|\mathcal{H}_t^F)$ in $p(\mathbf{x}|\mathcal{H}_t^B)$. Ob upoštevanju apriorne verjetnosti prisotnosti objekta v sliki p_A lahko za slikovni element \mathbf{x} ocenimo verjetnost, da pripada objektu, z uporabo Bayesovega pravila

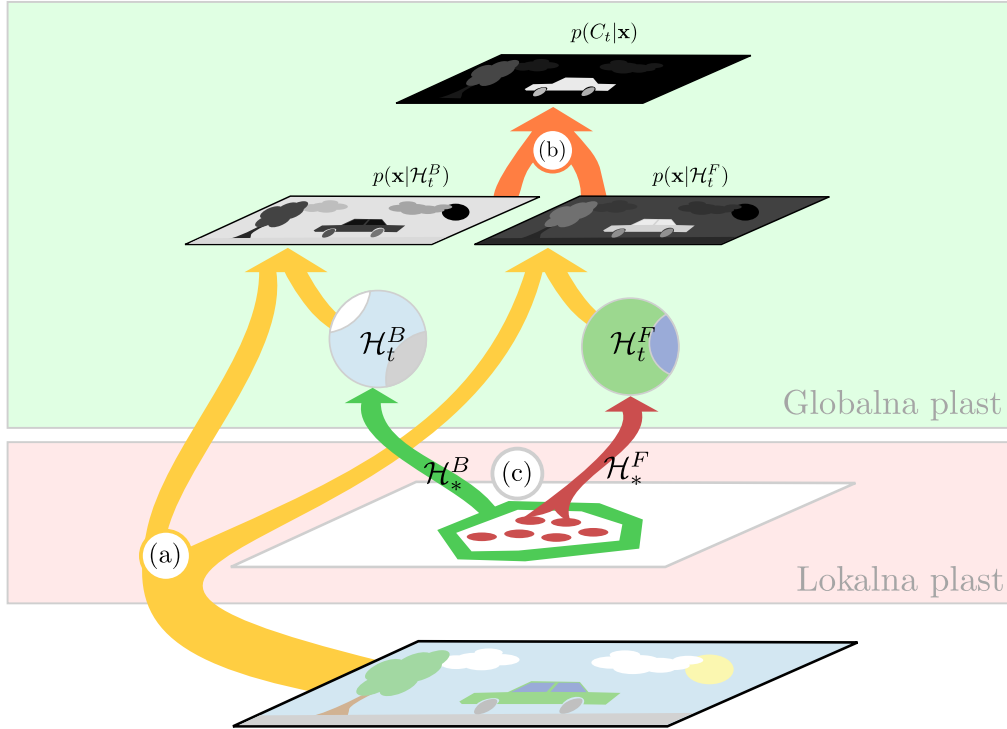
$$p(C_t|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{H}_t^F)}{p_A p(\mathbf{x}|\mathcal{H}_t^F) + (1 - p_A) p(\mathbf{x}|\mathcal{H}_t^B)}. \quad (3.22)$$

Oba histograma se tekom sledenja spreminjata tako, da starim podatkom uteženo prištejemo nove, pridobljene iz slike trenutnega časovnega koraka. Za histogram barve objekta pridobimo nove podatke iz slikovnih elementov, ki ležijo v okolici značilnic z dovolj visokimi utežmi. Za histogram barve okolice pa uporabimo slikovne elemente, ki ležijo v območju, ki ga definira konveksna ovojnica nad vsemi lokalnimi značilnicami v množici. Osveževanje barvnega modela s takimi podatki nam omogoča, da se osredotočimo na razlike v barvi v neposredni okolici objekta in ne v sliki na splošno. Oba histograma se v vsakem koraku osvežita po naslednjem postopku

$$\begin{aligned} \mathcal{H}_{t+1}^F &\leftarrow k_F \mathcal{H}_t^F + (1 - k_F) \mathcal{H}_*^F & \text{in} \\ \mathcal{H}_{t+1}^B &\leftarrow k_B \mathcal{H}_t^B + (1 - k_B) \mathcal{H}_*^B, \end{aligned} \quad (3.23)$$

kjer konstanti k_F in k_B označujeta del prisotnosti starih histogramov (\mathcal{H}_t^F ter \mathcal{H}_t^B) napram novim podatkom (histograma \mathcal{H}_*^F ter \mathcal{H}_*^B). V splošnem pa se izkaže, da je k_B bolj postaviti

nižje kot k_F , saj se barva okolice objekta ponavadi spreminja hitreje kot barva samega objekta. Celoten postopek uporabe barve v vizualnem modelu je povzet v Sliki 3.4.



Slika 3.4: Shematični prikaz uporabe barvnega modela.

Koraki si sledijo v časovnem zaporedju. (a) - na podlagi barvnih modelov \mathcal{H}_t^F in \mathcal{H}_t^B z uporabo povratne projekcije dobimo $p(\mathbf{x}|\mathcal{H}_t^F)$ in $p(\mathbf{x}|\mathcal{H}_t^B)$. (b) - na podlagi (3.22) izračunamo $p(C_t|\mathbf{x})$, ki se nato uporabi v (3.6) ter posledično za posodabljanje množice lokalnih značilnic. (c) - na podlagi lokalnih značilnic se določita histograma \mathcal{H}_*^F ter \mathcal{H}_*^B , ki se uporabita za posodobitev barvnega modela, kot je to opisano v (3.23).

Gibanje

Ujemanje druge uporabljene modalnosti, gibanja, je definirano kot konsistentnost lokalno določenega gibanja v sliki z modelom gibanja objekta. Model gibanja objekta je določen kot uteženo povprečje gibanja vseh lokalnih značilnic. Na žalost zaradi problema zaslonke (ang. *aperture problem*) lokalnega gibanja v sliki ne moremo natančno oceniti za poljubno točko. Zato smo v okviru magistrske naloge uporabili Harrisov detektor robov [24], z uporabo katerega določimo stabilne točke. Okolica teh točk je namreč bolj ugodna za ocenjevanje lokalnega gibanja z metodami optičnega toka. V stabilnih točkah nato izračunamo optični tok z metodo Lucas-Kanade [41]. Za točko \mathbf{x} , v kateri smo določili vektor gibanja, izračunamo ujemanje z modelom gibanja M_t na podlagi naslednje formule

$$v(\mathbf{x}) = e^{-\|f(\mathbf{x}) - M_t\|}. \quad (3.24)$$

Na tak način dobimo ujemanje samo v določenem številu točk na sliki, ne pa v vseh točkah. Posplošitev ocene na vse točke je dobljena z ocenjevanjem gostote porazdelitve verjetnosti z jedri (ang. *Kernel density estimate - KDE*), z uporabo katere nad množico točk dobimo naslednji približek verjetnostne porazdelitve

$$p(\mathbf{x} | M_t D_t) = \frac{1}{K} \sum_{\mathbf{x}_i} v(\mathbf{x}_i) \Phi_{\mathbf{P}}(\mathbf{x} - \mathbf{x}_i), \quad (3.25)$$

kjer K predstavlja normalizacijski koeficient, $\Phi_{\mathbf{P}}$ pa Gaussovo jedro

$$\Phi_{\mathbf{P}}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^2 |\mathbf{P}|}} \exp\left(-\frac{1}{2} \mathbf{x}^T \mathbf{P}^{-1} \mathbf{x}\right). \quad (3.26)$$

V funkciji (3.26) \mathbf{P} predstavlja kovariančno matriko, ki jo izračunamo z uporabo Parzen-ove okenske metode nad stabilnimi točkami. Osnove metode so opisane v dodatku A.3. V tem primeru je uporabljena Parzen-ova metoda, prirejena za delo z uteženimi vzorci, opisana v [31]. Oznaka D_t označuje model gostote točk, v katerih je bilo merjeno ujemanje. Z ocenjevanjem gostote porazdelitve točk je v model vključena tudi ocena pristranskosti, ki se pojavi zaradi različne gostote meritev v različnih delih slikovnega prostora. V nadaljevanju definiramo $p(M | \mathbf{x} D_t)$ kot približek zelene porazdelitve $p(M_t | \mathbf{x})$. Ta približek lahko zapišemo kot

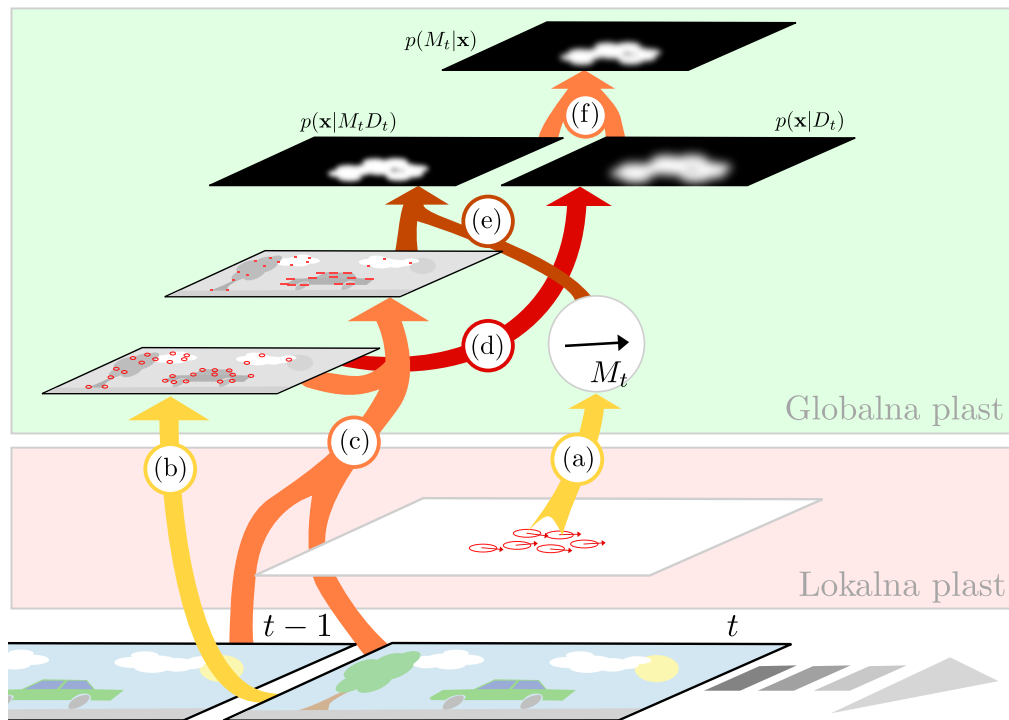
$$p(M_t | \mathbf{x} D_t) \propto \frac{p(\mathbf{x} | M_t D_t)}{p(\mathbf{x} | D_t)}, \quad (3.27)$$

kjer je $p(\mathbf{x} | M_t D_t)$ definirana kot v enačbi (3.25). Verjetnostna porazdelitev $p(\mathbf{x} | D_t)$ je ocenjena na podoben način kot (3.25), vendar brez uporabe uteži za konstrukcijo ocene gostote. Celoten postopek je povzet na Sliki 3.5.

Prostorska enovitost

V mnogo primerih tudi kombinacija barve in gibanja ne zagotavlja dovolj natančno in robustno definirane področja objekta v sliki. Zato je globalnemu vizualnemu modelu dodana še tretja modalnost, ki zagotavlja kompaktno strukturo množice lokalnih značilnic, obenem pa zmanjšuje možnost, da bi se nove značilnice prekrivale z že obstoječimi značilnicami. To modalnost imenujemo *prostorska enovitost*, sestavljata pa jo dve komponenti. Prvo komponento imenujemo *bližina*, katere pomen je omejevanje področja vzorčenja na osnovi področja, ki ga opisujejo trenutno aktivne značilnice. To je doseženo z uporabo maske, ki temelji na konveksni ovojnici množice značilnic. Problem take maske je, da ne omogoča širjenja območja objekta v sliki, kar predstavlja problem, ko objekt spremeni svojo vizualno velikost ali pa, ko je vizualno območje objekta ocenjeno manjše, kot je v resnici. Ta problem lahko rešimo z razširitvijo območja. Masko, ki jo določa konveksna ovojnica, povečamo z razširitvenim filtrom (ang. *dilate filter*). S tem omogočimo, da se množica značilnic lahko nadzorovano širi.

Druga komponenta je *razdalja*. Njen namen je omejevanje vzorčenja novih značilnic na področjih, ki jih opisujejo obstoječe značilnice. Tako masko ustvarimo tako, da invertiramo konvolucijo filtra, ki temelji na Gaussovem jedru z matriko, ki opisuje pozicije trenutnih



Slika 3.5: Shematični prikaz uporabe modela gibanja.

Koraki si sledijo v časovnem zaporedju. (a) - Na podlagi premika lokalnih značilnic je izračunan model globalnega gibanja. (b) - V sliki so z uporabo Harrisovega detektorja robov določene stabilne točke. (c) - V stabilnih točkah so za zaporedni sliki določeni vektorji optičnega toka. (d) - Na podlagi ocene gostote porazdelitve točk je sestavljena verjetnostna porazdelitev $p(\mathbf{x} | D_t)$. (e) - Na podlagi ocene gostote porazdelitve točk, uteženih glede na ujemanje z modelom gibanja, kot je to opisano v (3.24), je sestavljena verjetnostna porazdelitev $p(\mathbf{x} | M_t D_t)$. (f) - $p(M_t | \mathbf{x})$, kije določena na podlagi (3.27).

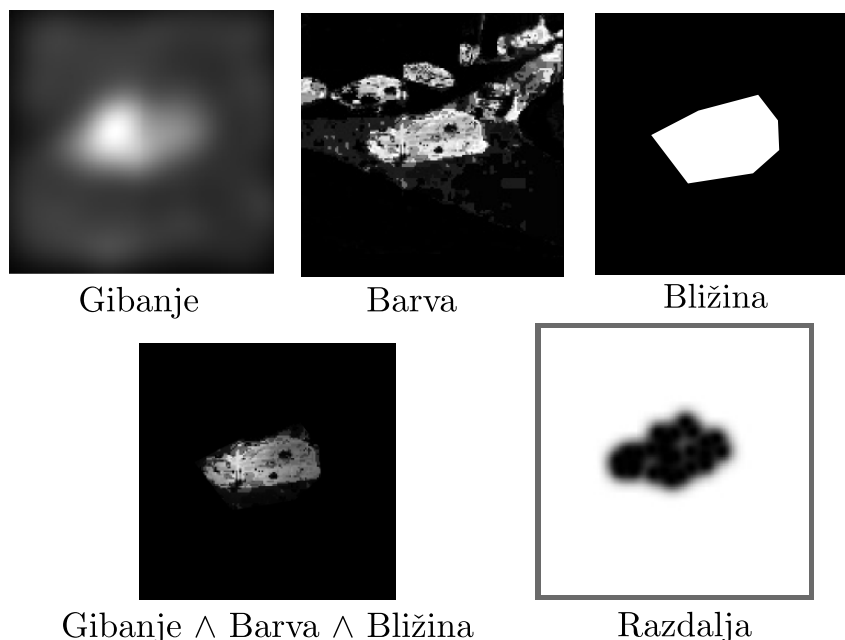
značilnic v sliki¹. Na Sliki 3.6 so ilustrirane verjetnostne porazdelitve posameznih modalnosti na podatkih iz sekvence *PETS2001-1* (glej poglavje 4).

■

Za vsak slučaj še enkrat omenimo, da obstaja možnost, da se pri konstrukciji skupne verjetnostne porazdelitve izkaže, da je le-ta "prazna", saj se posamezne verjetnostne komponente med seboj izključujejo. V idealnih razmerah se to zgodi, ko je objekt v celoti že pokrit z značilnicami. Na ta način se postopek vzorčenja novih značilnic lahko zaključi še preden dosežemo želeno število značilnic. Ta mehanizem implicitno daje vizualnemu

¹Matrika velikosti slike vsebuje vrednost 1 v celicah, ki ustrezajo položajem posameznih značilnic. Druge celice imajo vrednost 0.

modelu možnost regulacije števila značilnic, preko katere je mogoče tudi prilagajanje opisa objekta glede na njegove spremembe velikosti v sliki.



Slika 3.6: Grafični prikaz komponent verjetnostne porazdelitve.

Grafični prikaz uporabljenih verjetnostnih porazdelitev, ki jih generirajo posamezne modalnosti. Komponenti bližina in razdalja modalnosti prostorska enovitost sta zaradi večje jasnosti prikazani ločeni. Kot je razvidno iz prikaza družene verjetnostne porazdelitve barve, gibanja in bližine, te tri komponente opišejo področje objekta v sliki. Razdalja pa služi izključevanju delov tega področja, ki so že opisani z lokalnimi značilnicami.

3.5 Sledilnik

V tem razdelku je predstavljeno ogrodje sledilnika, ki integrira opisani vizualni model. Opisane so tudi lokalne značilnice, ki smo jih uporabili v okviru magistrske naloge.

3.5.1 Ocenjevanje dinamike objekta

V sledilniku smo želeli poleg samega iskanja ujemanja v sliki, ki ga zagotavlja vizualni model obravnavati še dinamiko objekta. Zato, kot nekateri obstoječi sledilniki [32], tudi naš sledilnik za napovedovanje položaja objekta v naslednjem časovnem koraku uporablja model gibanja, ki temelji na predpostavkah ter empirični lokalni zgodovini gibanja objekta. Že z dokaj preprostim modelom gibanja se za krajše časovne sekvence lahko napove naslednji položaj objekta, s čimer se zoži prostor iskanja ter s tem pohitri postopek sledenja. Poleg tega nam tak model pomaga razločevati tudi med nekaterimi vrstami dvoumnosti, ki

Algoritem 3 Pregled celotne psevdokode sledilnika**Inicializacija:**

- i **Vnos:** koordinate začetnega pravokotnega področja objekta
- ii Izberi začetne položaje značilnic in nastavi utež vsake značilnice na 1.

Sledenje: Za časovni korak $t = 1, 2, 3 \dots$

1. $\hat{\mathbf{x}}_t^i \leftarrow \mathbf{x}_{t-1}^i + [\dot{x}_t, \dot{y}_t]'$ for $i \in L_{t-1}, L_t = L_{t-1}$
2. Iskanje ujemanja (glej razdelek 3.2)
3. Posodabljanje uteži (glej razdelek 3.3.1)
4. Odstrani problematične značilnice:
Če velja $w_t^i < T_R$, potem $L_t \leftarrow L_t / \{i\}$
5. Posodobi informacije o globalnem izgled objekta ter sestavi $p(C_t, M_t, S_t | \mathbf{x})$ (glej razdelek 3.4)
6. Osveži množico značilnic:
Določi število novih značilnic n_{add} z uporabo (3.20).
Za $a = 1 \dots n_{add}$ naredi:
 - (a) $\mathbf{x}_t^* \leftarrow$ vzorči iz $p(C_t, M_t, S_t | \mathbf{x})$
 - (b) $w_t^* \leftarrow W_R$, kjer $T_R < W_R \leq 1$ (nastavi začetno utež)
 - (c) Posodobi verjetnostno porazdelitev $p(S_t | \mathbf{y})$, da bo odražala tudi postavitev nove značilnice.
 - (d) $L_t \leftarrow L_t \cup \{*\}$
7. $\bar{\mathbf{x}}_t \leftarrow \frac{\sum w_t^i \mathbf{x}_t^i}{W}$
8. Posodobi stanje Kalmanovega filtra z $\bar{\mathbf{x}}_t$

se pojavijo v sledenju. V našem sledilniku z uporabo Kalmanovega filtra, katerega teorija je opisana v dodatku A.4, ocenjujemo model, ki predpostavlja konstantno gibanje objekta (ang. *nearly constant velocity motion model - NCV*) [39] za globalno gibanje objekta. Vektor stanja in vektor meritve sta torej definirana kot

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix}, \mathbf{y}_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix}, \quad (3.28)$$

kjer x_t in y_t predstavljata koordinati položaja objekta v prostoru slike ter \dot{x}_t in \dot{y}_t kompo-

menti vektorja njegove hitrosti. Dinamični model in model opazovanja sta definirana kot

$$\mathbf{x}_t = \mathbf{F} \cdot \mathbf{x}_{t-1} + \mathbf{w}_t, \quad (3.29)$$

$$\mathbf{y}_t = \mathbf{H} \cdot \mathbf{x}_t + \mathbf{v}_t, \quad (3.30)$$

kjer \mathbf{x}_t predstavlja trenutno stanje, \mathbf{x}_{t-1} predhodno stanje, \mathbf{w}_t šum procesa ter \mathbf{v}_t šum meritve. Matriki \mathbf{F} in \mathbf{H} narekujeta prehode med stanji. Ob predpostavki enakomernega gibanja objekta sta matriki definirani kot

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ in } \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (3.31)$$

Algoritem 3 prikazuje, kako je vizualni model vgrajen v skelet sledilnika. V okviru časovnega koraka t je najprej uporabljen model gibanja objekta, ki napove naslednji položaj objekta. Začetna ocena položajev posameznih značilnic, označena z \hat{X}_t , temelji na tem premiku (Algoritem 3, step 1). Nato sledi iskanje ujemanja v obliki optimizacije, posodabljanje uteži posameznih značilnic in osvežitev množice značilnic (Algoritem 3, koraki od 2 do 7, opisani v razdelku 3.2). Nov položaj objekta je določen kot \bar{x}_t , ta ocena pa je nato uporabljena pri posodabljanju modela gibanja (Algoritem 3, koraka 8 in 9).

3.5.2 Lokalne značilnice

V okviru magistrske naloge je bila za opis lokalnih značilnosti objekta uporabljena diskretna verjetnostna porazdelitev intezitete, pridobljena v okolici točke \mathbf{x} v sivinski sliki. Ta preprosti opis izgleda smo izbrali, ker ima dobro neodvisnost glede na rotacijo v primerjavi z bolj specifičnimi opisi, kot so na primer zaplate. Poleg tega tak način opisa sam po sebi ni preveč specifičen, kar je ugodno za sledenje spreminjajočih se objektov, kjer je potrebno obravnavati tudi nepopolna ujemanja. Predpostavka, na kateri temelji ta odločitev, je, da specifičnost vizualnega modela namesto iz lokalnega opisa izvira iz globalne kombinacije večjega števila šibkih lokalnih značilnic. V okviru opisa lokalnega izgleda smo se namerno izognili uporabi barvne informacije. Glavni razlog za to je, da smo želeli ohraniti obravnavo barvne informacije kot neobvezno modalnost v okviru globalnega opisa objekta. Drugi razlog za uporabo sivinske informacije je študija, kako je opisno moč vizualnega modela mogoče povečati z ustrezno kombinacijo večjega števila takih značilnic.

Verjetnostno porazdelitev intezitete opišemo z enodimenzionalnim normaliziranim histogramom sivinske slike, ki je definiran kot

$$\mathcal{H}(\mathbf{x}, t) = \left\langle \frac{1}{|R(\mathbf{x})|} \sum_{\mathbf{y} \in R(\mathbf{x})} \delta_j(b(\mathcal{I}_t(\mathbf{y}))) \right\rangle_{j=1:N_h}, \quad (3.32)$$

kjer $R(\mathbf{x})$ predstavlja okolico točke \mathbf{x} , δ je Kronecker-jeva funkcija ter N_h označuje število celic v histogramu. Funkcija $b(\mathcal{I}_t(\mathbf{y}))$ vrne celico v histogramu, ki mu pripada intenziteta slikovnega elementa \mathbf{y} v sliki

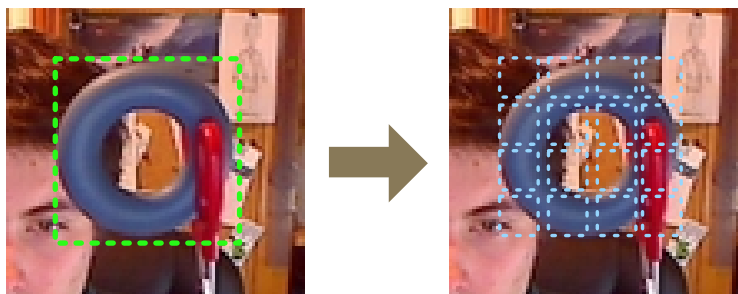
Vizualni model lokalne značilnice z indeksom i torej zapišemo kot normaliziran histogram okolice točke $\mathbf{x}_i^{s(i)}$ v sliki časovnega koraka $s(i)$ torej $\mathcal{H}(\mathbf{x}_i^{s(i)}, s(i)) = \langle h^{i,j} \rangle_{j=1:N_h}$. V nadaljevanju definiramo še histogram na poljubni poziciji \mathbf{x} v sliki časovnega koraka t kot $\mathcal{H}(\mathbf{x}, t) = \langle h_t^{i,j}(\mathbf{x}) \rangle_{i=1:N_h}$. Funkcijo ujemanja vizualnega modela lokalne značilnice z indeksom i definiramo z uporabo Bhattacharyya-evega koeficienta [4] za dva histograma

$$S(\mathbf{x}, i, t) = \exp(-D_B(\mathcal{H}(\mathbf{x}_i^{s(i)}, s(i)), \mathcal{H}(\mathbf{x}, t))) = \sum_{j=1}^{N_h} \sqrt{h^{i,j} h_t^{i,j}(\mathbf{x})}, \quad (3.33)$$

kjer D_B označuje Bhattacharyya-ev koeficient.

3.5.3 Inicializacija sledilnika

Kot je to razvidno iz Algoritma 3, je potrebno na začetku sledenja vizualni model inicializirati. V trenutnem sledilniku se to naredi tako, da uporabnik na prvi sliki označi pravokotno področje, ki vsebuje objekt, kateremu hoče slediti. Začetno število značilnic je določeno na podlagi velikosti površine področja. Ker ni podanega nobene apriorne informacije o strukturi objekta, so značilnice razporejene po področju v preprostem mrežnem vzorcu, kot je to prikazano na Sliki 3.7. Uteži vseh lokalnih značilnic so postavljene na 1, saj podrobnosti o strukturi objekta še niso znane, zato so vse značilnice obravnavane kot enako pomembne.



Slika 3.7: Prikaz začetne postavitve značilnic.

Leva slika prikazuje področje objekta, ki je ročno določeno pred začetkom sledenja. Na podlagi velikosti območja ter velikosti lokalnih zaplat, s katerimi operirajo značilnice, se potem določi optimalno število značilnic za pokritje tega področja. Značilnice so razvrščene v mrežnem vzorcu, kot je to prikazano na desni sliki. Kot je to razvidno iz primera (gre za začetno sliko iz sekvence *torus*, ki je predstavljena v razdelku 4.2), se pri tem lahko zgodi, da nekatere značilnice niso postavljene neposredno na sam objekt, saj je pravokotni opis začetnega področja samo grob približek njegove oblike. Take značilnice zaradi zgradbe vizualnega modela hitro izpadejo iz množice lokalnih značilnic.

Poglavje 4

Eksperimenti

V tem poglavju so predstavljeni eksperimenti, ki so bili opravljeni v okviru magistrske naloge. Njihov namen je empirično ovrednotenje vizualnega modela oziroma sledilnika, opisanega v poglavju 3. V razdelku 4.1 so opisane performančne značilnosti naše trenutne implementacije sledilnika, opisanega v poglavju 3. V razdelku 4.2 so opisane lastnosti posameznih slikovnih zaporedij, uporabljenih v okviru magistrske naloge. Razdelek 4.3 opisuje metodologijo eksperimentov. V razdelku 4.4 so opisani rezultati eksperimentov, ki obravnavajo samo delovanje trenutne implementacije sledilnika. V razdelku 4.5 so predstavljeni rezultati eksperimentov, katerih namen je ovrednotiti kvaliteto sledenja predstavljenega sledilnika napram izbranim obstoječim sledilnikom.

4.1 Računska hitrost implementacije

Večji del implementacije sledilnika, ki je bila uporabljena v okviru eksperimentov, je spisane v programskem jeziku Matlab, del pa v programskem jeziku C.

Opravila v okviru posameznega časovnega koraka na računalniku s procesorjem Intel Core 2 Quad 6600 potrebujejo od 0.2 do 0.6 sekunde (za nastavitve parametrov, ki smo jih najbolj pogosto uporabljali v okviru eksperimentov). Velik del porabe časa lahko pripišemo implementaciji v skriptnem jeziku, v okviru katere je bila pozornost primarno usmerjena na preglednost kode ter njeno razširljivost. Tako ostaja velik del hitrostnega potenciala v trenutni obliki še neizkoriščenega. Po naših ocenah bi lahko čista C implementacija sledilnika na podobni strojni opremi dosegla tudi hitrosti, ki bi bile primerne za tekoče procesiranje video tokov. Tabela 4.1 prikazuje rezultate meritev hitrosti za posamezne bolj opazne segmente algoritma vizualnega modela. Pri tem je pomembno poudariti, da je največkrat obiskani del implementacije, to je računanje histogramov in njihovega ujemanja, spisan v jeziku C. Drugi del, ki je tako pospešen, je računanje optičnega toka, kjer je uporabljena optimizirana implementacija metode Lucas-Kanade iz prosto dostopne knjižnice OpenCV.

Iz Tabele 4.1 je razvidno, da trenutno največ časa porabita RANSAC detekcija značilnic s samosvojim gibanjem ter posodabljanje modalnosti. V okviru slednjega največ časa porabi Parzen-ova okenska metoda, ki pa bi se jo v prihodnje dalo nadomestiti s kakšno manj natančno (a hitrejšo) metodo ocenjevanja gostote verjetnostne porazdelitve.

Segment	<i>povprečna hitrost</i>	<i>varianca</i>
globalna optimizacija	10	0.1
lokalna optimizacija	10	0.1
RANSAC	200	6
iskanje prekrivanja	2	0.1
osvežitev modalnosti	120	2
dodajanje novih značilnic	15	0.1
celotni korak	360	10

Tabela 4.1: Rezultati meritev hitrosti posameznih delov algoritma.

Vsi rezultati so v milisekundah. Rezultati posameznih segmentov se ne seštejejo v rezultat celotnega koraka procesiranja, saj nekaterih, v splošnem časovno zanemarljivih delov, nismo prikazali.

4.2 Uporabljená slikovna zaporedja

V magistrski nalogi so bila za eksperimente uporabljena v nadaljevanju opisana video slikovna zaporedja, ki obravnavajo različne scenarije sledenja. Pregled lastnosti zaporedij je povzet v Tabeli 4.2 ter na Sliki 4.1.

PETS slikovna zaporedja (PETS2000, PETS2001-1, PETS2001-2)

PETS je okrajšava za mednarodno delavnico na področju ocenjevanja sistemov za sledenje ter nadzor (*Performance Evaluation of Tracking and Surveillance*). Delavnica poteka vsako leto, prav tako pa je vsako leto objavljen tudi paket video posnetkov, na katerih lahko udeleženci primerjajo svoje sisteme. Tematika paketov se vsako leto nekoliko spremeni, zato smo za naše eksperimente uporabili izseke iz paketov, uporabljenih na delavnicah leta 2000 in 2001. Ta zaporedja so primarno namenjena sledenju avtomobilov med gibanjem na parkirišču, glavni razlog za njihovo uporabo pa je primerjava našega algoritma z algoritmom PAKT, predstavljenim v [42]. Omenjeni algoritem je bil v članku testiran na video posnetkih s podobnimi lastnostmi. Poleg tega so video zaporedja zanimiva tudi zaradi sprememb izgleda objekta (avtomobila). Kljub temu, da gre za tog objekt, se le-ta med samim gibanjem obrača, zato so spremembe v lokalnem opisu izgleda nujne za uspešno sledenje.

Kolesarka (bicycle)

Četrto video zaporedje slik prikazuje žensko na kolesu, ki se vozi mimo snemalca. To zaporedje vključuje nekaj izzivov za sodobne vizualne sledilnike. Velikost kolesarke na sliki se namreč tekom posnetka spreminja. Prav tako se zaradi tresljajev kamere hitro spreminja njen položaj v sliki. V drugem delu zaporedja nastopi tudi popolno zakritje kolesarke z drogom semaforja.

Sledenje roke (hand)

V okviru petega zaporedja slik je cilj sledilnika, da sledi človeški roki, ki se premika po pro-

storu. Gre za scenarij, katerega obvladovanje bi omogočilo zanimive aplikacije na področju interakcije med človekom in računalnikom oziroma robotom. Zahtevnost tega problema leži v hitrem spreminjanju vizualne strukture ter posledičnem pomanjkanju stabilnih značilnic, na katere bi se lahko sledilnik zanesel v daljšem obdobju. Zaradi teh problemov obstoječi sledilniki k sledenju rok pristopajo z uporabo globalnega barvnega modela [12, 36, 61]. Tak pristop na prvi pogled sicer izgleda zanimiv, vendar predpostavlja, da je barva roke v danem okolju unikatna, velikokrat pa tudi, da se velikost roke ne spreminja. To v splošnem seveda ne drži. Glavni razlog uporabe tega zaporedja je primerjava predstavljenega modela s sledilnikom, ki temelji na jati značilnic [30]. Avtorja omenjenega sledilnika se v svojem članku osredotočita ravno na sledenje roke z namenom interakcije z računalniškim vmesnikom.

Pločevinka, ki se kotali (can)

Šesto zaporedje slik je zahtevno za sledilnike, ki hočejo natančno opisati lokalni izgled objekta. Le-ta se namreč v tem primeru neprestano spreminja na način, ki ga je težko opisati z deformacijo lokalnih delov. Pri kotaljenju se namreč neprestano pojavljajo novi deli, stari pa izginjajo. V našem primeru je objekt dejansko bolj smiselno slediti samo na podlagi globalne barve, saj pazljivo sestavljen barvni model uspešno ločuje področje slike, ki pripada objektu od okolice. Vendar pa je treba poudariti, da gre za močno predpostavko, ki v veliki večini sledilnih scenarijev ne velja.

Šport (diver, highjump)

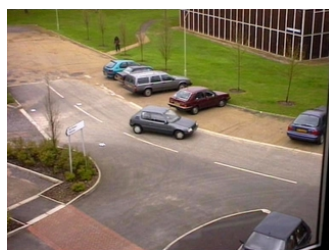
Naslednji dve video zaporedji sta povzeti po članku [37] in izvirata iz sveta športa. Prvo zaporedje prikazuje skok v vodo. Med samim skokom skakalka opravi nekaj akrobacij. Drugo zaporedje prikazuje skakalca v višino. V obeh primerih gre torej za ne-togo artikularno telo, ki se tekom video zaporedja hitro premika ter spreminja obliko.

Objekt z luknjo v geometrijskem središču (torus)

Zadnje uporabljeno video zaporedje vsebuje torus modre barve. Za veliko sledilnikov predstavlja objekt take vrste velik problem, saj ima v večini pogledov v svojem središču mase luknjo. Središče objekta pa je področje, na katerega se sledilniki pogosto najbolj zanašajo, saj velikokrat zagotavlja najbolj stabilen izgled, posledično pa torej stabilne detekcije. Če je središče torej votlo, ga nadomesti izgled ozadja, kar pa sledilniku otežuje sledenje. Poleg tega se torus tekom zaporedja tudi obrača izven ravnine kamere, tako, da, se njegova vizualna velikost hitro spreminja.

4.3 Metodologija meritev

Za ocenjevanje sledilnikov obstajaj več metrik, vsaka izmed njih pa ima dobre in slabe lastnosti. Nekatere temeljijo na številu potrebnih ročnih posredovanj ter razdalji med napovedjo sledilnika ter ročno anotirano informacijo (ang. *ground truth*) o položaju objekta. V naših eksperimentih uporabljamo dve metriki. Prva uporabljena metrika temelji na funkciji



PETS2000



PETS2001-1



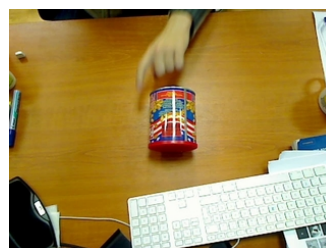
PETS2001-2



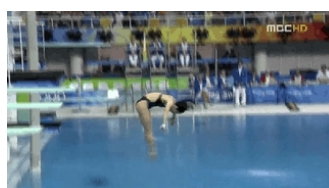
bicycle



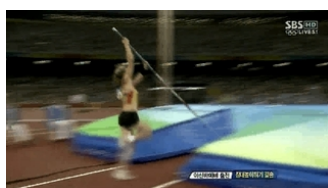
hand



can



diver



highjump



torus

Slika 4.1: Pregled izbranih slik iz uporabljenih video zaporedij.

Na sliki so prikazane izbrane slike iz uporabljenih slikovnih zaporedij, ki nakazujejo lastnosti posameznih zaporedij.

razlike pravokotnih področij. Za ročno določeno področje objekta R_a ter območje, ki ga določi sledilnik R , definiramo metriko razdalje kot

$$\delta_R(R_a, R) = \frac{R_a \cap R}{R_a \cup R}. \quad (4.1)$$

Funkcija (4.1) ima vrednost 1, ko velja $R_a = R$, ter 0, ko se področji R_a in R ne prekrivata. Na podlagi funkcije razlike (4.1) definiramo napako sledilnika, ko prekrivanje regij pade pod 0.1. V tem primeru je sledilnik ponastavljen na ročno določeno regijo objekta v danem časovnem koraku, sistem pa zabeleži dogodek kot napako sledilnika. Po tej metodologiji je torej v okviru posameznega poskusa sledenja boljši tisti sledilnik, ki ima manjše število napak.

Drugi način ocenjevanja temelji na varianci razdalje med ocenjenim središčem objekta, ki ga določi sledilnik, in ročno določenim središčem. Cilj takega ocenjevanja ni iskanje

Video	št. slik	Komentarji
PETS2000	184	avto, več pogledov, konstantno gibanje
PETS2001-1	185	avto, več pogledov, sprememba velikosti
PETS2001-2	250	avto, več pogledov, sprememba velikosti, nelinearno gibanje s postanki
bicycle	270	polno zakritje objekta, treslaji kamere, sprememba velikosti
hand	242	zelo deformabilen objekt, sunkoviti gibi
can	210	objekt, ki se kotali, hitro spreminjanje izgleda, sunkoviti gibi
diver	214	ne-togo artikulirano telo, veliko deformacij, rotacija
highjump	70	ne-togo artikulirano telo, veliko deformacij
torus	262	središče objekta ni polno

Tabela 4.2: Pregled lastnosti uporabljenih video zaporedij.

neposrednega ujemanja, pač pa, podobno kot v enačbi (3.14) v razdelku 3.3.1, določanje podobnosti gibanja oziroma trajektorije. Funkcija metrike je definirana kot

$$\delta_M(\{\mathbf{x}_t\}, \{\mathbf{x}_t^a\}) = \text{var}_{t=1\dots t_E} (\|\mathbf{x}_t - \mathbf{y}_t\|), \quad (4.2)$$

kjer $\{\mathbf{x}_t\}$ predstavlja trajektorijo gibanja sledilnika, $\{\mathbf{x}_t^a\}$ pa ročno določeno trajektorijo gibanja objekta. Konstanta t_E označuje število slik v zaporedju. Ob tem je treba poudariti, da je treba to oceno jemati z zadržkom, saj vključuje tudi premike, ki so nastali zaradi ponastavitve sledilnika ob njegovi odpovedi. Taka obravnava je potrebna, saj bi po drugi strani neupoštevanje teh nenadnih premikov delovalo v prid sledilnikom, ki pogosto odpovejo.

Ker imamo v našem primeru opravka s stohastičnimi metodami, vsak eksperiment za iste parametre sledilnika ponovimo 30-krat, razen v primerih, v katerih je število ponovitev eksplicitno navedeno. Zbrane rezultate nato ustrezno statistično obdelamo.

4.4 Lastnosti sledilnika

4.4.1 Parametri sledilnika

V poglavju 3 je bilo omenjenih kar nekaj konstant, ki, ene bolj, druge manj, vplivajo na delovanje vizualnega modela. V tem razdelku bomo opisali vrednosti konstant, ki smo jih uporabili v okviru eksperimentov.

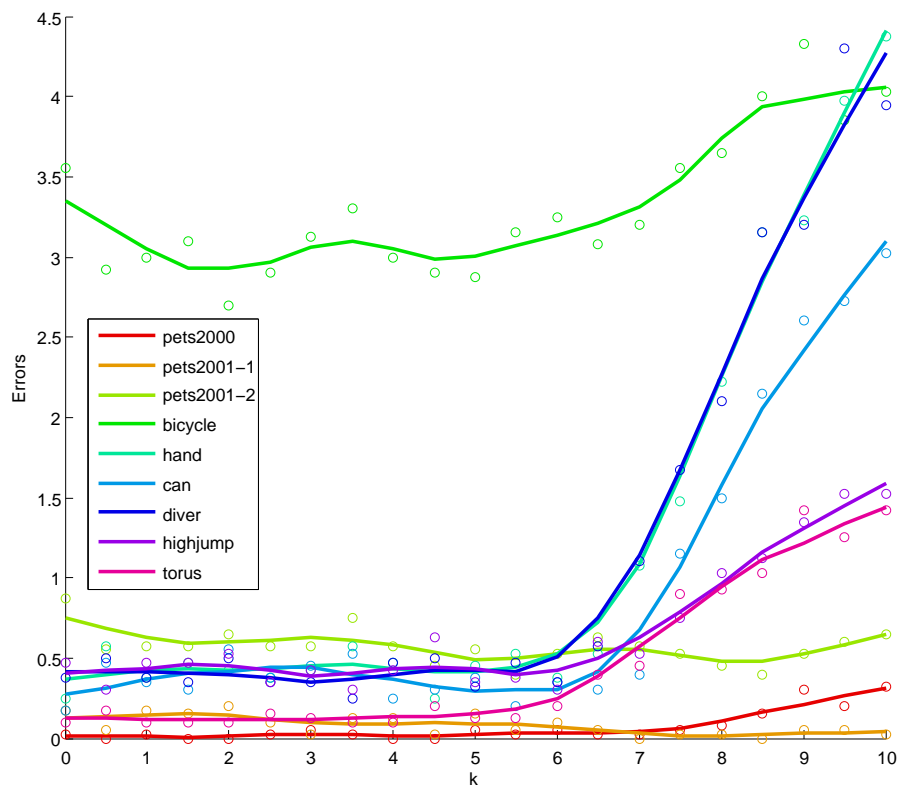
Za opis lokalnih značilnic smo uporabil sivinske histograme s šestnajstimi celicami, torej velja $N_h = 16$. V okviru koraka globalne optimizacije algoritem v vsaki iteraciji vzorči $N_G = 50$ primerov, od katerih jih izbere $N_G^{elite} = 10$ najboljših. Izhodiščna kovariančna matrika Σ_G je diagonalna matrika z vrednostmi diagonale [20 20 0.08 0.005 0.005]. Prve dve komponenti vektorja določata področje translacije, tretja rotacije, četrta in peta pa skaliranje. Naši eksperimentalni rezultati so pokazali, da je za sledenje najpomembnejše široko

začetno območje translacije, ki more v primeru napačne napovedi modela gibanja le-to popraviti. Ostale tri komponente dopuščajo zgolj lokalne spremembe. Če v okviru iteracije algoritma križne entropije vzorčeni primeri ne zadostujejo za izboljšanje ocene rezultata, se število primerov poveča za 10. Povečevanje števila vzorcev se konča pri 300 vzorcih. Ustavitveni pogoj pravi, da je rešitev dovolj dobra, ko determinanta matrike Σ_G pade pod $t_G = 0.1$. V koraku lokalne optimizacije se za vsako značilnico vzorci $N_L = 40$ primerov pozicije, od katerih se izbere $N_L^{elite} = 5$ najboljših. Izhodiščna kovariančna matrika Σ_L je dvodimenzionalna diagonalna matrika z vrednostima diagonale 5. Optimizacija posamezne značilnice j se zaključi, če determinanta kovariančne matrike Σ_j pade pod $t_L = 0.001$. Vpliv parametra λ_G , definirane v (3.8), smo preverili bolj podrobno z eksperimentom. Naša teorija namreč pravi, da bi moral imeti prav ta parameter velik vpliv na uspešnost sledenja za različno toge predmete. Da bi čimbolj enakomerno pokrili prostor parametra, smo meritve izvajali za različne vrednosti k , pri čemer je $\lambda_G = \frac{e^k}{10000}$. Rezultati so prikazani na grafu v Sliki 4.2.

Iz rezultatov je razvidno, da so objekti na nekaterih video zaporedjih bolj občutljivi na vrednost parametra λ_G kot drugi. Na eni strani imamo bolj toge objekte, ki na spremembe niso zelo občutljivi. V to skupino sodijo vsa tri PETS zaporedja, kjer je obravnavani objekt avtomobil. Vmesna skupina z opazno večjim vplivom parametra λ_G vsebuje zaporedja *bicycle*, *highjump* ter *torus*. V prvih dveh omenjenih zaporedjih gre za človeško telo, ki pa mu je v dani resoluciji slik težko pripisati jasno razdelitev delov. Tudi rezultati za zaporedje *torus* se z omejevanjem deformacij predvsem slabšajo. To bi lahko pripisali načinu vizualnih deformacij objekta v tem zaporedju, saj gre za hitre spremembe velikosti. V zadnjo skupino z največjim vplivom sodijo zaporedja *hand*, *can* in *diver*. Za prvo in zadnjo zaporedje te skupine je taka umestitev pričakovana, saj gre v teh dveh primerih za jasno netoga artikulirana predmeta. Srednje zaporedje sicer vsebuje tog predmet (pločevinko), vendar gre za način gibanja, ki ga naš vizualni model ne upošteva neposredno. Z ozirom na rezultate v Sliki 4.2 smo za nadaljnje eksperimente izbrali vrednost $k = 5$, kar posledično pomeni, da je $\lambda_G \simeq 0.015$.

Kot dobro ujema očne značilnice so obravnavane lokalne značilnice, katerih ujemanje v sliki presega prag $M_G = 0.8$, ujemanje slabih značilnic pa je nižje od $M_B = 0.4$. V okviru metode RANSAC se primerjajo trajektorije zadnjih 10 časovnih korakov, torej $t_s = t - 10$. Sam algoritem RANSAC ima 100 iteracij, pri čemer model sprejmemo, če pojasnjuje gibanje vsaj 80% vseh značilnic. Kriterij pripadanja posamezne značilnice modelu smo opisali v razdelku 3.3.1. Za drsenje značilnic stran od večine se uporablja eksperimentalno določen prag $T_S = 40$. Poudariti je treba, da taka nastavitev delno pogojuje tudi samo velikost objekta, še posebej, če imamo opravka z zaporedji z različnim številom slikovnih elementov. V okviru opravljenih eksperimentov smo uporabili zaporedja s podobnim številom slikovnih elementov, zato težav z različnimi velikostmi ni bilo. Problematične lokalne značilnice odstranimo iz množice lokalnih značilnic, če njihova utež pade pod $T_R = 0.1$. Dinamiko velikosti množice lokalnih značilnic določa konstanta λ_N , omenjena v enačbi (3.20). Za vrednost le-te smo v okviru eksperimentov izbrali $\lambda_N = 0.8$.

Globalno modalnost barve opisujeta histograma v HSV barvnem prostoru s $16 \times 16 \times 4$ celicami. Kot je to navedeno v enačbi (3.23), se histogram barve objekta posodablja s $k_F = 0.95$, histogram barve okolice pa s $k_B = 0.5$. Apriorna verjetnost objekta, uporabljena v



Slika 4.2: Rezultati za različne vrednosti parametra togosti λ_G .

Nižja vrednost k pomeni manjše upoštevanje geometrijskih omejitev. Rezultati so podani kot povprečno število napak. Dejanski rezultati za posamezne vrednosti so prikazani kot krogi, krivulje pa predstavljajo glajeno vizualizacijo trenda.

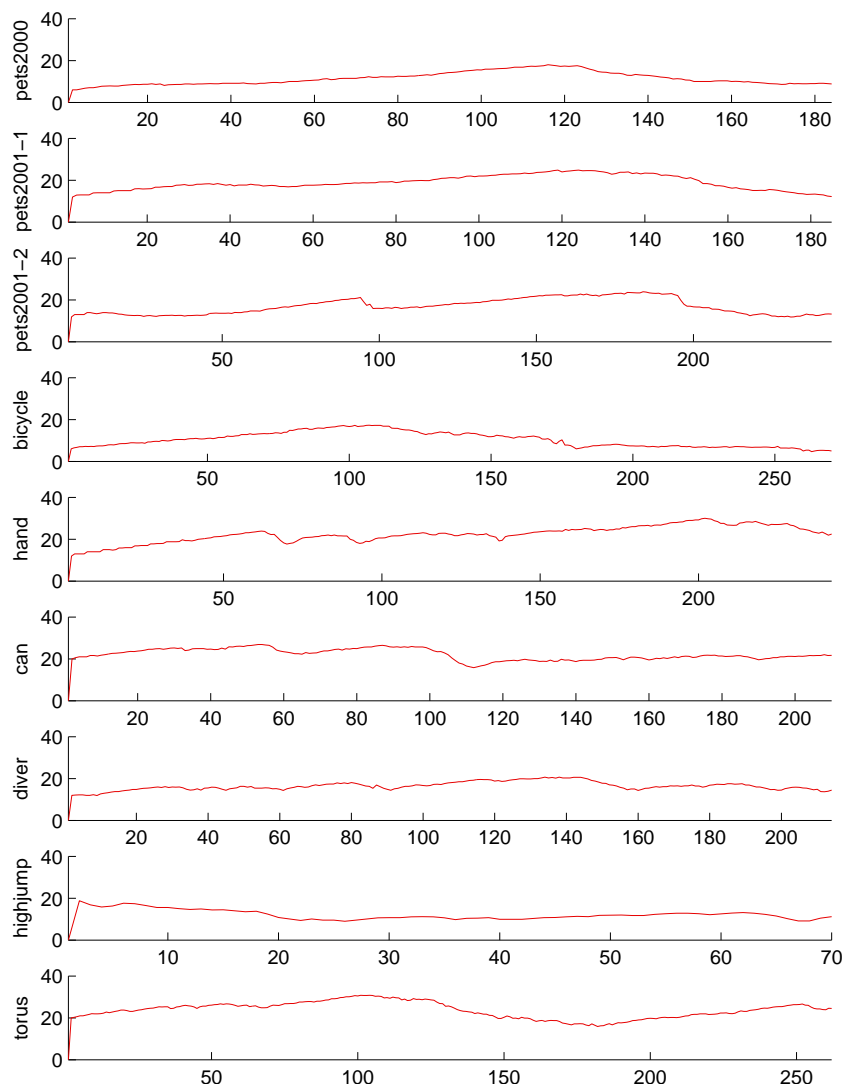
enačbi (3.22), je $p_A = 0.1$. Globalna modalnost gibanja je definirana na podlagi Harrisovih kotnih točk. Prag detekcije točk v Harrisovem algoritmu je empirično nastavljen na 20. Ta vrednost nam zagotavlja dovolj veliko število stabilnih točk. V teh točkah je ocenjen optični tok z uporabo metode Lucas-Kanade. Okolica točke, v kateri ocenjujemo optični tok, je velika 8 slikovnih elementov.

V okviru Kalman-ovega filtra, ki je uporabljen za modeliranje gibanja objekta, smo kovariančni matriki za šum procesa ter šum meritve nastavili na $Q = I$ ter $R = I$, kjer je I enotska matrika.

4.4.2 Dinamika množice lokalnih značilnic

Pomembna lastnost predstavljenega vizualnega modela je sposobnost, da se prilagaja velikosti objekta na sliki. Prilagajanje poteka s spreminjanjem števila lokalnih značilnic v

spodnji plasti vizualnega modela. Na Sliki 4.3 so za posamezna zaporedja prikazani grafi spreminjanja velikosti množice značilnic čez čas. Povezanost velikosti množice z velikostjo objekta na sliki je najlepše razvidna za zaporedja *pets2000*, *pets2001-1* ter *bicycle*. V teh primerih gre za objekt, ki vizualno najprej raste (se približuje kameri ali obrača), potem pa se zmanjšuje.



Slika 4.3: Prikaz sprememb velikosti množice značilnic v posameznih video zaporedjih.

Poleg same velikosti množice lokalnih značilnic si je zanimivo pogledati tudi starost samih značilnic. V Tabeli 4.3 sta prikazana povprečje ter mediana starosti značilnic za posamezno video zaporedje. Iz tabele lahko razberemo, da posamezne značilnice v zaporedjih z bolj enostavnim in počasnim gibanjem "živijo" dlje kot v tistih zaporedjih, v katerih neprestano prihaja do velikih sprememb v izgledu objekta. Primer zaporedji s počasnim

obnavljanjem množice značilnic so vsa tri PETS zaporedja ter zaporedje *bicycle*, po drugi strani pa imajo zaporedja *hand*, *diver*, *can* in *highjump* tudi za 50% krajšo povprečno starost značilnic.

Video	povprečje	mediana
PETS2000	18.8	17.0
PETS2001-1	15.4	12.4
PETS2001-2	16.7	13.9
bicycle	18.5	15.3
hand	11.9	9.3
can	10.8	8.3
diver	9.1	6.7
highjump	7.3	5.7
torus	13.8	10.2

Tabela 4.3: Pregled statistike starosti značilnic za posamezna video zaporedja.

4.4.3 Vpliv delov algoritma na sledenje

Opisani algoritem sledilnika, predvsem pa uporabljeni vizualni model, sestavlja kar nekaj med seboj povezanih delov. Zato se pojavlja vprašanje, kakšen je sploh vpliv posameznega dela na končno uspešnost sledenja. Da bi določili vpliv posameznih delov, smo opravili dodane meritve na spremenjenem sledilniku, ki posameznih izbranih delov algoritma ne vključuje. Za ta eksperiment smo izbrali naslednje štiri dele algoritma: globalni korak optimizacije, globalno modalnost barve, globalno modalnost gibanja ter ocenjevanje naslednjega položaja objekta s Kalman-ovim filtrom.

Video	g. opt.	barva	gibanje	Kalman	original
PETS2000	0.07 / 1.45	0.80 / 20.9	0.10 / 0.95	0.03 / 3.86	0.05 / 3.47
PETS2001-1	2.00 / 11.5	1.47 / 13.5	0.75 / 24.7	0.17 / 12.23	0.15 / 8.81
PETS2001-2	1.27 / 12.55	2.59 / 10.97	1.90 / 40.84	0.62 / 8.04	0.55 / 10.64
bicycle	5.12 / 8.47	2.22 / 24.12	2.95 / 3.77	3.20 / 3.35	2.87 / 3.56
hand	8.47 / 49.01	4.02 / 27.30	0.25 / 11.89	1.47 / 9.51	0.45 / 4.31
can	0.95 / 22.71	1.30 / 45.78	0.27 / 29.94	0.32 / 25.37	0.37 / 26.08
diver	1.37 / 6.57	3.52 / 28.76	0.72 / 8.55	0.32 / 9.73	0.32 / 10.40
highjump	1.72 / 20.51	2.00 / 12.66	0.67 / 14.07	0.57 / 15.45	0.35 / 11.58
torus	1.12 / 13.01	1.27 / 23.12	0.05 / 2.34	0.12 / 7.59	0.12 / 7.98

Tabela 4.4: Pregled vpliva posameznih delov algoritma na sledenje.

Posamezen stolpec prikazuje rezultate za spremenjeni sledilni algoritem, v katerem je onemogočen določen del. Za vsak eksperiment sta navedena povprečno število odpovedi na poskus ter povprečna varianca ujemanja trajektorij (kot je to opisano v razdelku 4.3).

Rezultati za posamezna video zaporedja so prikazani v Tabeli 4.4. Glede na rezultate v

tabeli lahko zaključimo, da je globalna optimizacija pomemben korak v vseh obravnavanih scenarijih sledenja. Njena odstranitev iz algoritma namreč opazno poslabša robustnost sledenja še posebej v primerih, kjer imamo opravka s hitrimi premiki objekta med časovnimi koraki ter gibanjem, ki nasprotuje napovedim modela gibanja. Primer takega zaporedja je *hand*, kjer roka med hitrim premikom spremeni smer gibanja za približno 90 stopinj. V takih primerih je globalna optimizacija nujna za pravilno postavitev celotnega modela, ki ga lokalna optimizacija potem bolj natančno prilagodi trenutnemu izgledu objekta.

Odstranitev globalne modalnosti barve ali gibanja po pričakovanjih prinese mešane rezultate. Odstranitev modalnosti barve sicer v večih primerih prinese poslabšanje, predvsem pa bolj nestabilno sledenje (večja varianca razlik trajektorij). Kar se tiče samih napak v sledenju, je izjema zaporedje *bicycle*, kjer odsotnost barve sledenje izboljša. Odstranitev modalnosti gibanja po drugi strani sledenje izboljša v primeru zaporedij *hand*, *can* in *torus*. Značilnost obravnavanih objektov v teh zaporedjih je, da na njih ni mogoče določiti dovolj stabilnih točk, ki so potrebne za ocenitev optičnega toka. V primeru roke in torusa gre za objekta brez teksture, v primeru pločevinke pa je razlog zameglitev zaradi vrtenja. Glede na te ugotovitve lahko sklepamo, da bi pristop, ki bi lahko dinamično izbiral posamezne modalnosti v danem časovnem koraku, lahko robustnost predstavljenega vizualnega modela še izboljšal.

Ocenjevanje naslednjega položaja sledilnika s Kalmanovim filtrom prinese opazno izboljšanje zgolj v zaporedjih, kjer imamo opravka s hitrim gibanjem objekta. Taka zaporedja so *bicycle*, *hand* in *highjump*.

4.5 Primerjava z obstoječimi sledilniki

Empirična primerjava z obstoječimi sledilniki je eden izmed bolj pogosto uporabljenih pokazateljev zmogljivosti na področju sledenja v računalniškem vidu. V okviru te magistrske naloge so bili za primerjavo uporabljeni trije različni sledilniki, ki temeljijo na dognanjih, objavljenih v zadnjih letih. Kriterij za izbor sledilnikov je bil raznovrstnost značilnosti sledilnikov ter dostopnost njihovih implementacij.

PAKT sledilnik (*PAKT*)

Sledilnik PAKT, predstavljen v [42], temelji na sledenju z več jedrnimi funkcijami, povezanimi z fleksibilnim geometrijskim modelom. Sledenje z uporabo večih jedrnih funkcij je bilo bolj podrobno opisano v razdelku 2.1.2. Specifičnost implementacije, ki v članku sicer ni omenjena, je, da je vsaka značilnica v bistvu sestavljena iz treh togo povezanih jeder, kar naj bi zagotovilo večjo stabilnost. Po tri take značilnice so povezane skupaj, njihovo gibanje pa omejeno z afino transformacijo. Ob tem velja omeniti tudi, da je sledilnik PAKT potrebno na začetku vsakega sledenja previdno ponastaviti z vnosom položaja vsakega jedra posebej. Zaradi tega zamudnega procesa smo eksperimente omejili na 10 poskusov za vsako izbrano zaporedje slik.

Sledenje z jato značilnic (*FOF*)

Koncept jate ptic, predstavljen v [30], temelji na množici lokalnih značilnic, katere dinamika je uravnavana s preprostimi pravili jate, ki so bolj podrobno opisana v razdelku 2.2.2. Položaji novih značilnic so vzorčeni na podlagi preprostega statičnega barvnega modela. V naših eksperimentih smo sledilnik implementirali v jeziku Matlab, pri tem pa smo se opirali na originalno kodo, razvito v okviru projekta HandVU¹.

Filter z delci (PF)

Eden izmed bolj uveljavljenih pristopov k vizualnemu sledenju temelji na globalnem vizualnem modelu ter filtru z delci [14]. Taka kombinacija je bila in je še vedno predmet intenzivnih raziskav [46, 6, 38]. Za našo primerjavo smo uporabili filter z delci v kombinaciji z barvnim modelom v obliki histograma v HSV barvnem prostoru. Sledilnik temelji na prosto dostopni implementaciji (glej dodatek B), ki smo jo spremenili do te mere, da se barvni model med sledenjem prilagaja izgledu objekta.

Sledilnik	PAKT	FOF	PF	naš sledilnik
<i>PETS2000</i>	0.60 / 23.38	1.10 / 2.36	0.50 / 3.54	0.05 / 3.47
<i>PETS2001-1</i>	2.90 / 31.21	1.43 / 28.93	0.63 / 6.55	0.15 / 8.81
<i>PETS2001-2</i>	7.20 / 22.64	1.50 / 16.11	6.60 / 29.11	0.55 / 8.81
<i>bicycle</i>	8.10 / 7.41	2.73 / 6.10	6.15 / 15.41	2.87 / 3.56
<i>hand</i>	*	10.17 / 77.81	3.86 / 32.01	0.45 / 4.39
<i>can</i>	8.90 / 43.92	0.10 / 28.38	0.00 / 6.90	0.37 / 26.08
<i>diver</i>	3.40 / 28.87	2.20 / 41.38	4.60 / 20.94	0.32 / 10.40
<i>highjump</i>	4.30 / 12.86	0.06 / 14.36	1.36 / 24.32	0.35 / 11.58
<i>torus</i>	10.20 / 31.44	5.06 / 35.69	1.83 / 30.05	0.12 / 7.98

Tabela 4.5: Rezultati primerjave z obstoječimi sledilniki.

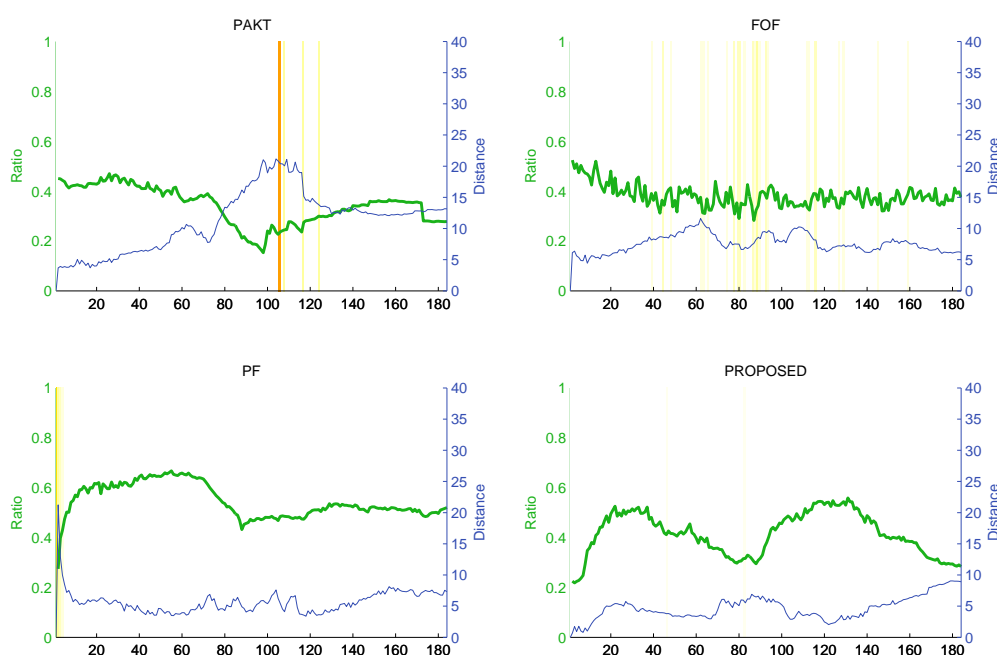
Za vsak eksperiment so rezultati zapisani v obliki x / y , kjer x predstavlja povprečno število odповіdi na poskus, y pa povprečno varianco ujemanja trajektorij (kot je to opisano v razdelku 4.3). Eksperimenta, označenega z *, ni bilo mogoče izvesti, ker je sledilnik prepogosto odповovedal.

■

Rezultati eksperimenta so povzeti v tabeli 4.5 ter bolj podrobno prikazani v obliki grafov na Slikah 4.4 do 4.12. Vsak graf prikazuje rezultate za en sledilnik. Zelena krivulja prikazuje vrednost $\delta_R(R_a, R)$, ki je definirana v (4.1), v odvisnosti od časa, modra krivulja pa razdaljo med ročno določenim ter ocenjenim središčem objekta. Iz definicije obeh metrik sledi, da ima dober sledilnik zeleno krivuljo čim višjo (dobra pokritost objekta), modro pa čim nižjo (majhna razdalja do središča). Vertikalne črte prikazujejo povprečno število odповіdi v posameznem časovnem koraku. Rdeča črta pomeni zelo pogosto odповoved na točno določenem časovnem koraku, široki rumeni pasovi pa pogosto odповoved na določenem intervalu.

¹Originalna koda sledilnika se je namreč izkazala za preveč integrirano z ostalimi komponentami projekta HandVU, da bi jo bilo mogoče uporabiti kot samostojno celoto v našem ogrodju za izvajanje eksperimentov.

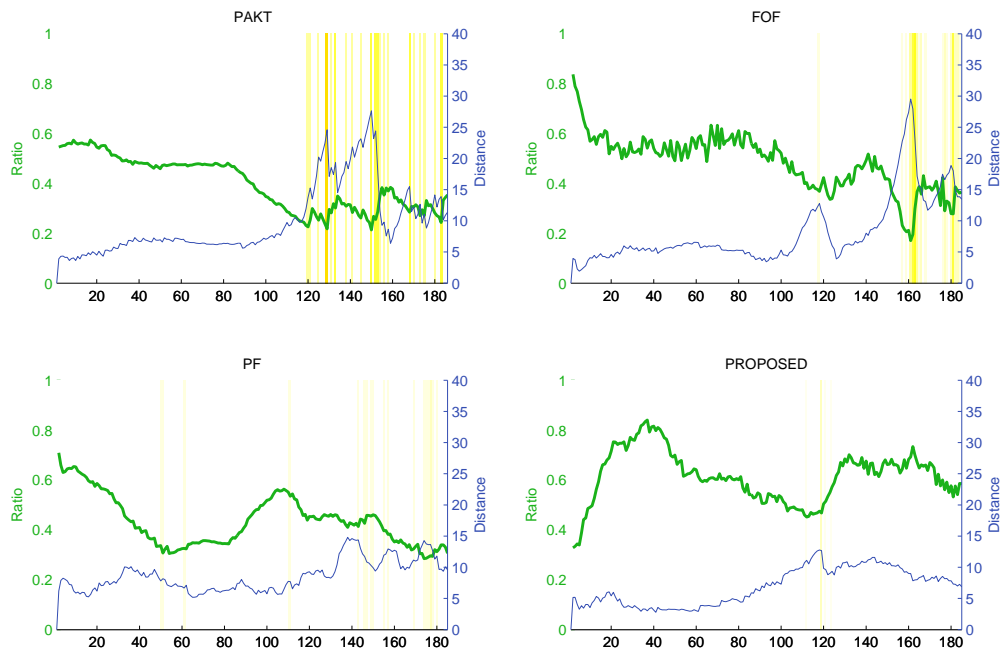
Iz prvih treh zaporedjih slik (PETS2000, PETS2001-1, PETS2001-2) je razvidno, da je naš sledilnik v takem scenariju bolj robusten kot ostali sledilniki. Ima manj odpovedi na posamezen poskus, varianca razdalje pa je boljša ali primerljiva z najboljšim rezultatom. Kljub temu ostaja še kar nekaj prostora za izboljšave. Predvsem so presenetljive odpovedi sledilnika v zaporedju PETS2000, ki so sicer redke. Po podrobnem pregledu rezultatov smo ugotovili, da je razlog za te napake togo postavljen kriterij odpovedi sledilnika. Zaradi odstranitve velikega dela značilnic se lahko namreč področje, ki ga opisuje vizualni model, v določenih razmerah v enem koraku naglo zmanjša, tako da metrika (4.1) pade prenizko. Razlogi za nastop teh razmer v okviru magistrske naloge niso bili raziskani.



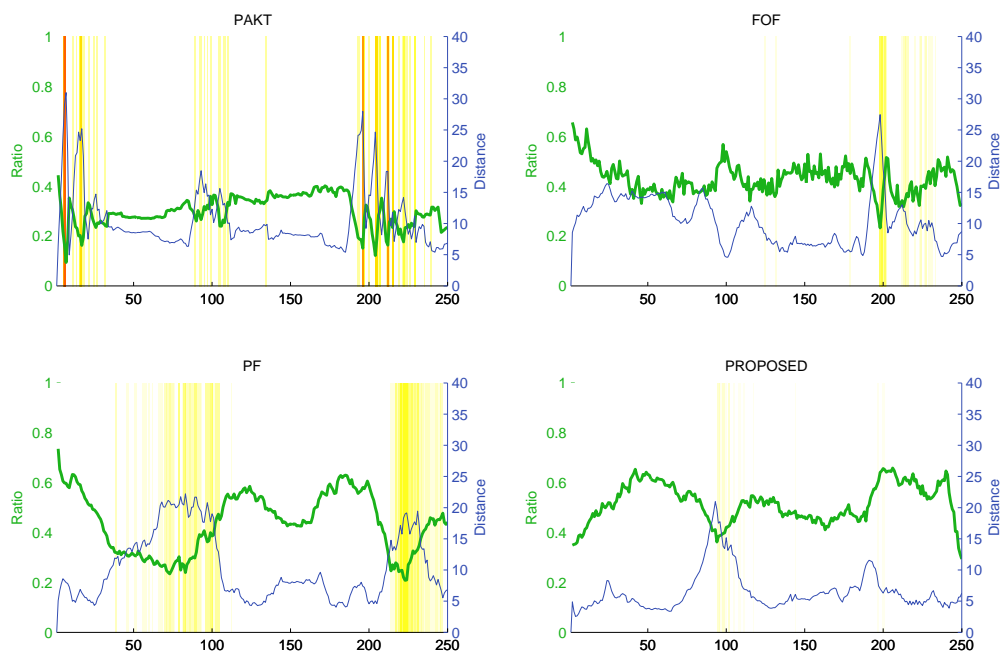
Slika 4.4: Primerjava sledenja za zaporedje slik *PETS2000*.

Glavni cilj uporabe PETS zaporedij je bil primerjava s PAKT sledilnikom. Le-ta se ni odrezal dobro, še posebej v okviru zaporedij PETS2001-1 in PETS2001-2, katerih rezultati so prikazani na Slikah 4.5 in 4.6. To lahko pripišemo hitrim premikom avtomobila, ki nastopijo v delih teh dveh zaporedij. PETS sledilnik namreč temelji zgolj na lokalni optimizaciji, kar pomeni, da se pri večjih spremembah položaja ne prilagaja dovolj hitro. Poleg tega obračanje avtomobila spremeni lokalni izgled objekta, ki zahteva, da se posodobijo oziroma zamenjajo tudi posamezne lokalne značilnice ter geometrijski model, ki jih povezuje.

Slikovno zaporedje *bicycle* sej je med vsemi uporabljenimi zaporedij slik izkazalo za najtežje. Največji vir težav je zakrivanje kolesarke, ki nastopi med časovnima korakoma 170 in 180. Kot lahko vidimo na grafih na Sliki 4.7, imata v tem delu največ težav sledilnik FOF ter naš sledilnik. Drugi problem je sunkovito gibanje objekta zaradi tresenja ročno vodene kamere. V takih primerih se *NCV* model gibanja (razdelek 3.5) izkaže za oviro, saj



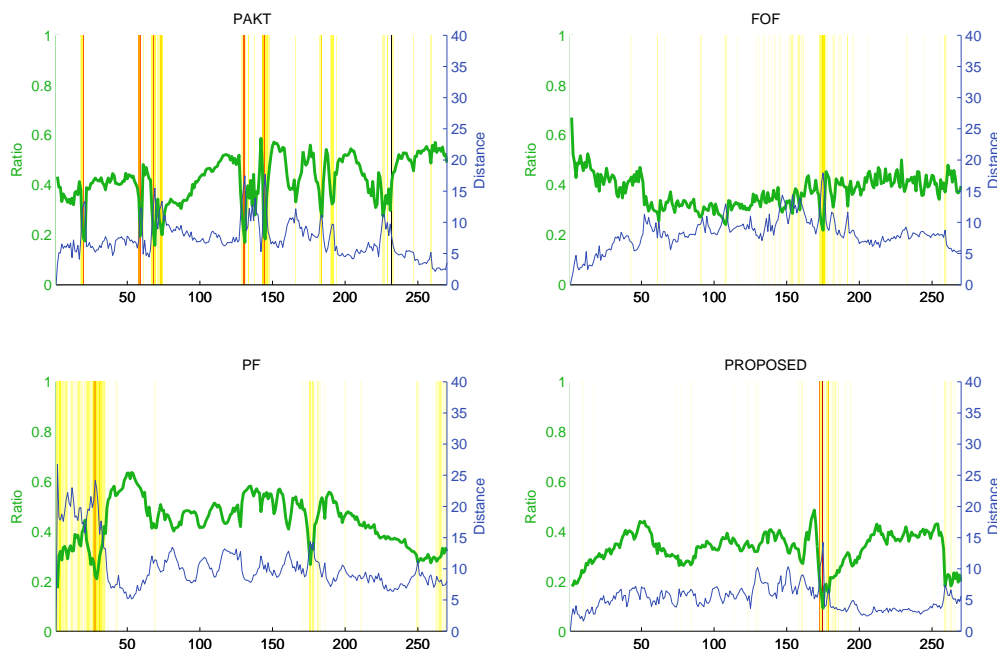
Slika 4.5: Primerjava sledenja za zaporedje slik *PETS2001-1*.



Slika 4.6: Primerjava sledenja za zaporedje slik *PETS2001-2*.

se v večini primerov moti. Sledilniku PETS je veliko problemov povzročala tudi sprememba velikosti kolesarke v kombinaciji s postopnim spreminjanjem lokalnega izgleda, ki nastopi

zaradi približevanja in oddaljevanja od kamere.

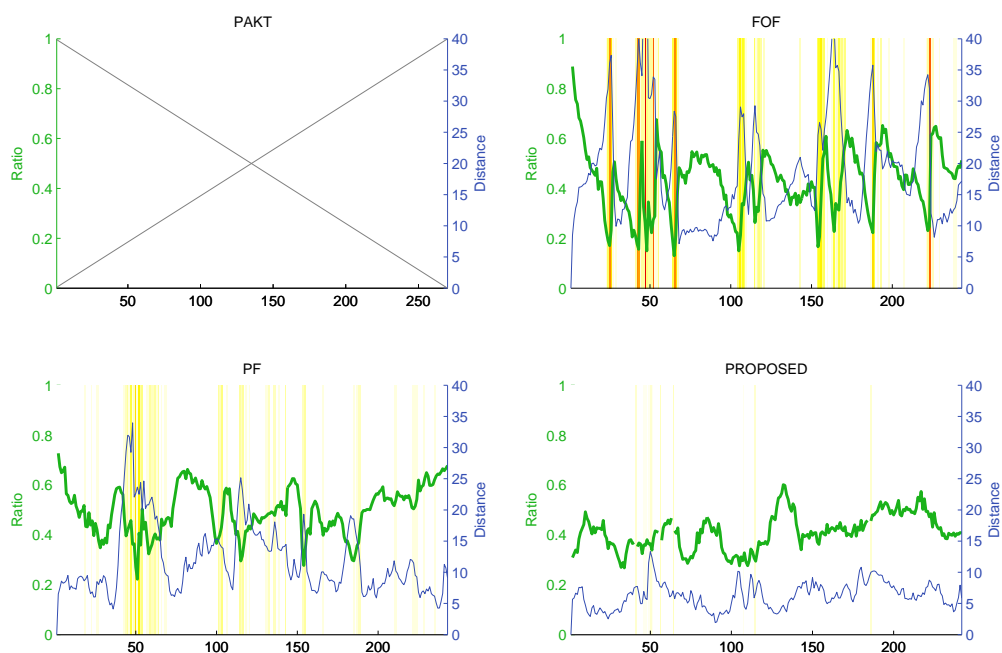


Slika 4.7: Primerjava sledenja za zaporedje slik *bicycle*.

Zaporedje *hand* se je izkazalo za zelo težavno za sledilnika FOF ter PAKT. Slednji je odpovedoval celo tako pogosto, da zaključevanje eksperimentov ni bilo smiselno². Predlagani sledilnik se je obnesel dosti bolje, še posebej kar se tiče razdalje od središča objekta ter njene variance. Najtežji del zaporedja je hiter zamah in sprememba smeri gibanja roke med slikama 40 in 50 (Slika 4.8). V takih primerih se model gibanja znova izkaže za problematičnega, zato je nujno, da je vizualni model dovolj specifičen, da premaga "vztrajnost" modela gibanja. Tudi sledilnik PF deluje zelo nezanesljivo. Razlog za to je, da njegov vizualni model temelji le na barvi. V obravnavanem zaporedju pa se pojavlja več področij s podobno barvo, kot jo ima roka (npr. glava in lesena polica), zato jih sledilnik ne more dobro ločiti.

Zaporedje *can* je eno izmed dveh slikovnih zaporedij, kjer predlagani sledilnik deluje opazno slabše kot nekateri ostali sledilniki, s katerimi smo ga primerjali. Scenarij, v katerem se objekt valjaste (ali kroglaste) oblike kotali po ravni površini, na prvi pogled sicer izgleda dokaj preprost, še posebej ko gre za objekt, ki je barvno dobro ločljiv od ozadja. To je tudi razlog, da imata sledilnika FOF in PF boljše rezultate. V primeru FOF gre za statičen barvni model, kar je idealno za omenjeni scenarij. Sledilnik PF sicer svoj barvni model spreminja, vendar so barve objekta in okolice tako različne, da ne pride do drsenja. Po drugi strani v našem vizualnem modelu barva igra zgolj sekundarno vlogo. Primarno sledenje poteka v sivinski sliki, kjer je izgled objekta dosti bolj podoben izgledu okolice. Ker gre za

²Kot je bilo omenjeno že v opisu sledilnikov, sledilnik PAKT predvideva ročno postavitev posameznih značilnic, zato je njegovo testiranje časovno zelo potratno.



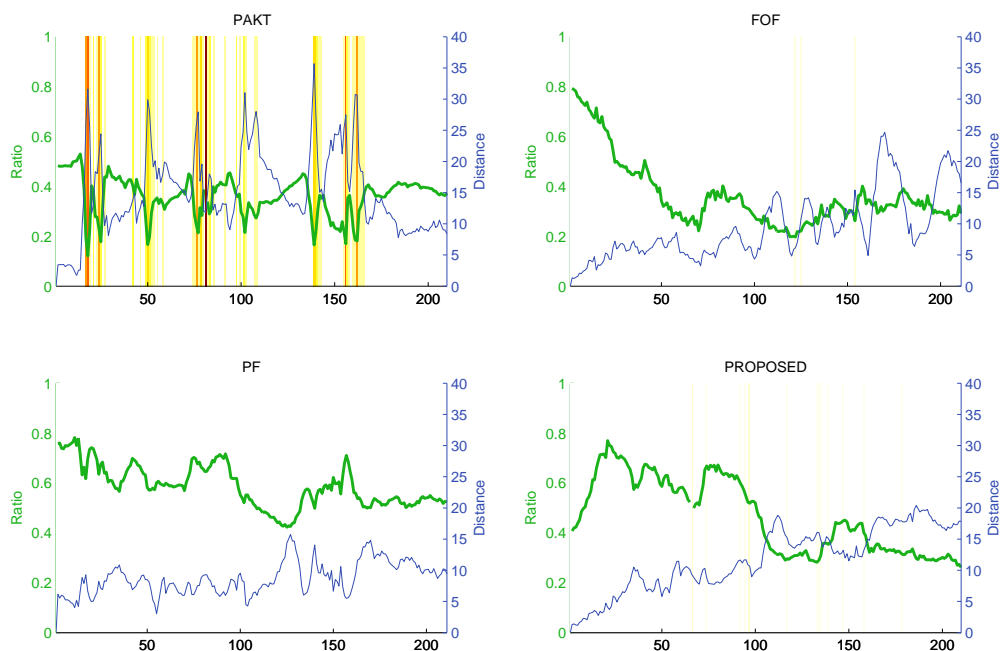
Slika 4.8: Primerjava sledenja za zaporedje slik *hand*.

Kot je to omenjeno že v opisu Tabele 4.5, eksperimenta s PAKT sledilnikom na tem zaporedju ni bilo mogoče izvesti, ker je sledilnik prepogosto odpovedoval.

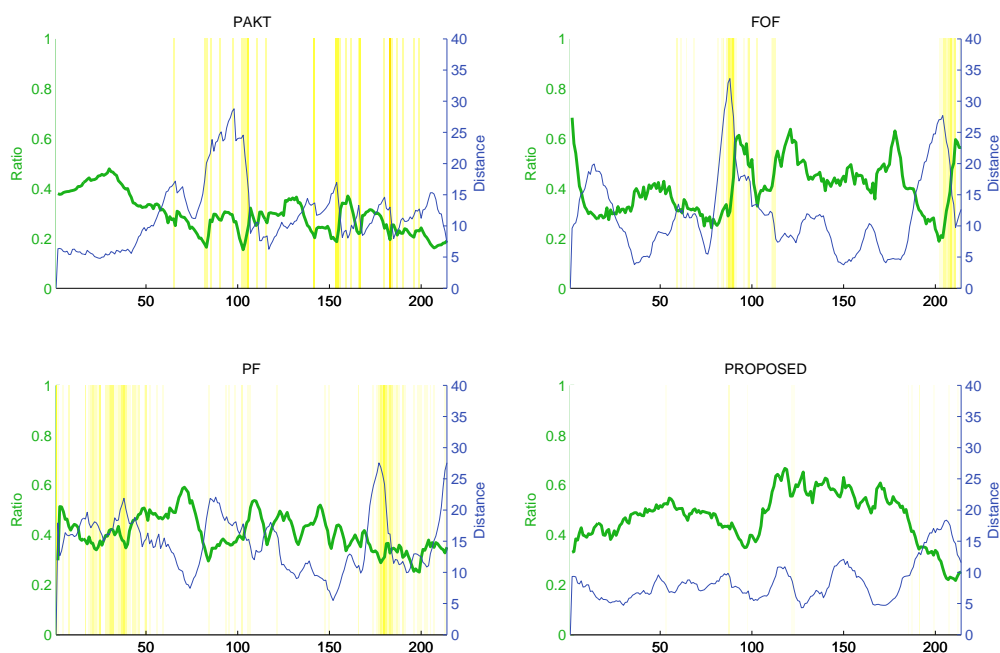
kotaleči se objekt, to pomeni, da se njegov izgled neprestano spreminja tako, da je to težko pojasniti z uporabljenim geometrijskim modelom. Za izboljšanje sledenja bi v prvi vrsti lahko uporabili lokalne značilnice, katerih vizualni model bi temeljil na lokalni informaciji o barvi, čemur smo se v okviru te magistrske naloge zavestno odpovedali. Podrobni rezultati za zaporedje so prikazani na Sliki 4.9.

Najbolj očiten primer sledenja artikuliranega telesa med uporabljenimi zaporedji je zaporedje *diver*. Skakalka v vodo med skokom namreč opazno spreminja obliko, večina transformacij pa se dogaja glede na ravnino kamere. To je ugodno za geometrijske modele, kakršne uporabljamo v opisanem vizualnem modelu. Kot smo omenili že v razdelku 3.2, opisani geometrijski model temelji na predpostavki, da se večina sprememb v geometriji objekta dogaja v dvodimenzionalnem prostoru. Ostali sledilniki so imeli z deformacijami skakalke več težav, še posebej v prvem in zadnjem delu zaporedja, kjer se skakalka zvije skupaj oziroma raztegne. Podrobni rezultati za zaporedje so prikazani na Sliki 4.10.

Podobno kot pri zaporedju *can*, tudi pri zaporedju *highjump* opisani sledilnik ni dosegel relativno najboljših rezultatov. Bolje od njega se je odrezal sledilnik FOF. Za razliko od zaporedja *diver*, gre tu za primer nagle deformacije izven ravnine kamere ter prisotnost zameglitve zaradi hitrega gibanja (ang. *motion blur*). Kot je razvidno iz Slike 4.11, je za opisani sledilnik usoden srednji del zaporedja, to je del, kjer je velikost skakalca v sliki najmanjša. Gre tudi za zelo kratko zaporedje slik (samo 70 slik) s hitrim spreminjanjem izgleda od samega začetka, kar pomeni, da naš sledilnik nima priložnosti, da bi se izgleda



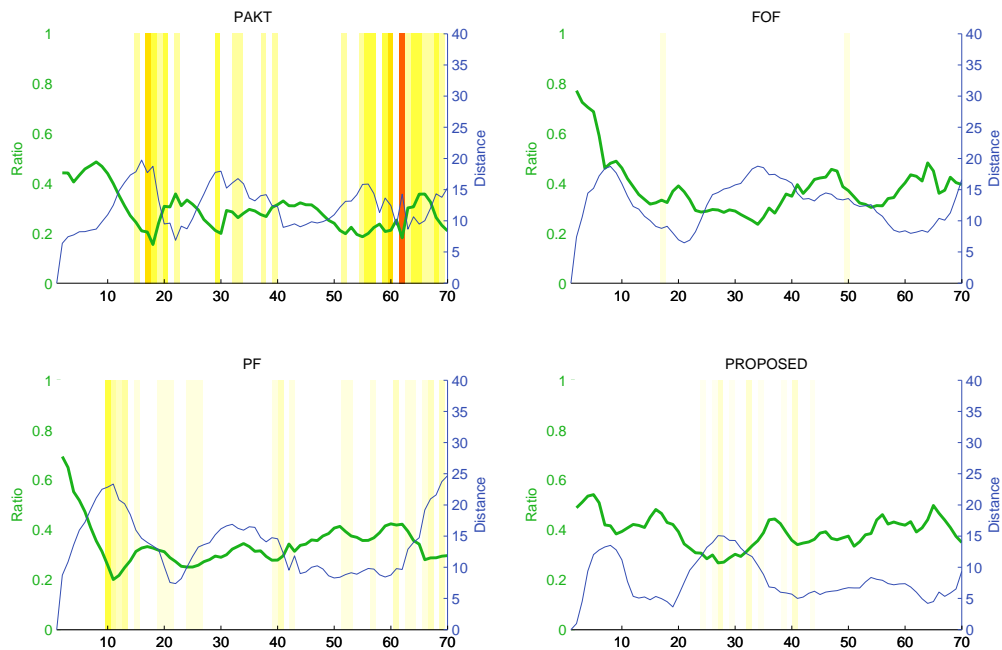
Slika 4.9: Primerjava sledenja za zaporedje slik *can*.



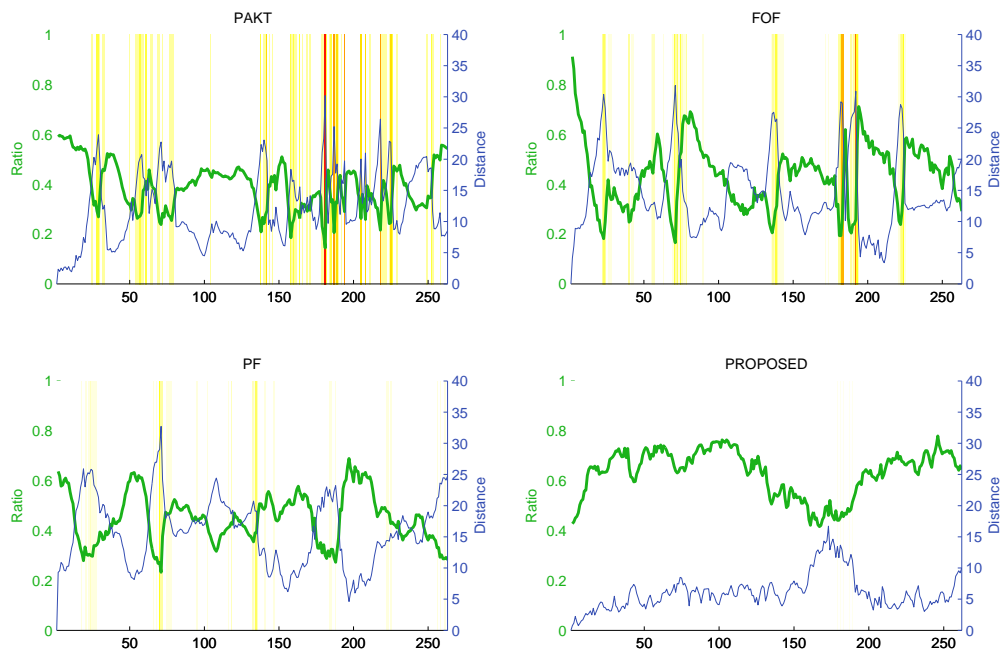
Slika 4.10: Primerjava sledenja za zaporedje slik *diver*.

objekta naučil tudi na globalnem nivoju.

Rezultati zadnjega uporabljenega slikovnega zaporedja *torus* so prikazani na Sliki 4.12.



Slika 4.11: Primerjava sledenja za zaporedje slik *highjump*.



Slika 4.12: Primerjava sledenja za zaporedje slik *torus*.

Predlagani sledilnik se je v okviru tega zaporedja odrezal bolje kot sledilniki, s katerimi smo ga primerjali. Razlogi za to so vizualne lastnosti objekta. Gre za gladek objekt, kar

predstavlja težavo za sledilnika PAKT in FOF. Oba sledilnika se namreč zanašata na lokalno stabilnost in nedvoumnost izgleda objekta v lokalnih področjih. To je potrebno tako za dobro ujemanje histogramov v PAKT sledilniku kot tudi za ocenjevanje optičnega toka, ki se uporablja v FOF sledilniku. Po drugi strani pa luknja v središču objekta predstavlja problem za globalni vizualni model, ki ga uporablja sledilnik PF. Predlagani vizualni model te probleme rešuje s kombinacijo lokalne in globalne informacije. Kot je opisano v razdelku 3.2, že sama optimizacija geometrijskega modela temelji na dveh korakih. Zaradi koraka globalne optimizacije je sledenje v obravnavanem primeru v večjem delu zaporedja stabilno, tudi če posamezne lokalne značilnice nimajo enoličnih detekcij v sliki. Le-ta se namreč določi v kombinaciji s togim geometrijskim modelom. Majhne spremembe v geometrijskem modelu med posameznimi časovnimi koraki (npr. zaradi rotacije izven ravnine kamere) se v geometrijo vnesejo na podlagi lokalnega koraka optimizacije. Poleg tega vizualni model avtomatično izloči značilnice, ki so na začetku sledenja postavljene na ozadje (primer za obravnavano slikovno zaporedje je prikazan na Sliki 3.7). Edini del zaporedja, kjer ima opisani vizualni model nekaj težav, je med slikama 150 ter 200. V tem delu se namreč objekt obrne pravokotno na ravnino kamere, s tem pa se njegova vizualna površina hitro zmanjša. Vizualni model zato zmanjša množico lokalnih značilnic. Ko pa se objekt obrne nazaj v osnovni položaj, se število značilnic ne poveča dovolj hitro zaradi omejitev, opisanih v razdelku 3.3.2, kar povzroči nestabilnost sledenja.



Kot je razvidno iz rezultatov, predlagani sledilnik v večini opisanih scenarijev deluje bolje kot sledilniki, s katerimi smo opravili primerjavo. Tudi v primerih, kjer drugi sledilniki delujejo bolje, razlika med njimi ni velika in jo lahko pripišemo večji osredotočenosti uporabljenih vizualnih modelov na naravo konkretnega problema. Kljub temu pa njegovo delovanje ni popolno. Znane vzroke napak bomo obravnavali v poglavju 5 skupaj z idejami za nadaljnje razširitve našega dela.

Poglavje 5

Diskusija

To poglavje vsebuje razpravo na temo lastnosti opisanega vizualnega modela, njegovih idejnih osnov ter predpostavljenih in empirično ugotovljenih omejitev. Med razpravo so omenjene tudi možnosti nadaljnjega razvoja vizualnega modela.

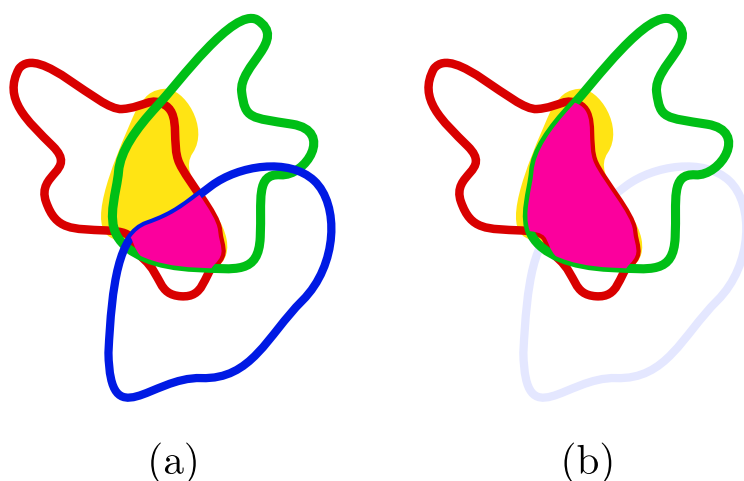
5.1 Globalnost in lokalnost

Že v uvodu smo omenili, da lahko ustrezna kombinacija globalnega in lokalnega vizualnega modela izkoristi prednosti obeh, pri tem pa minimizira vpliv slabih lastnosti. V poglavju 2 je bilo predstavljenih nekaj vizualnih modelov, ki z uporabo geometrijskega modela povežejo raznovrstne lokalne značilnice. V tem primeru je globalni model torej geometrijski model omejitev, lokalni vizualni model pa sestavlja množica lokalnih značilnic.

V opisanem vizualnem modelu je globalni vizualni model sestavljen iz vizualnih modalnosti, ki jih uporablja za določanje položajev novih lokalnih značilnic. Bolj podrobno bi lahko rekli, da te globalne vizualne modalnosti določajo območje, ki v sliki pripada objektu. Le-to je predstavljeno v obliki verjetnostne porazdelitve, da posamezen slikovni element pripada objektu. Vizualne modalnosti, ki v našem kontekstu predstavljajo globalni opis izgleda objekta, torej odločajo o dolgoročni uspešnosti sledenja, kratkoročno sledenje pa je odvisno od lokalnih značilnic ter geometrijskega modela. Oba aspekta vizualnega modela se medsebojno osvežujeta. V primeru osveževanja lokalnega modela je vloga globalne plasti očitna: na podlagi njega so določeni položaji novih lokalnih značilnic. V obratnem primeru je odvisnost nekoliko manj izrazita ter v trenutni implementaciji tudi manj izkoriščena. Kot je bilo že omenjeno v uvodu v poglavje 3 ter kasneje v razdelku 3.4, se dobre lokalne značilnice uporabijo za osveževanje posameznih modalnosti. V primeru modalnosti barve gre za sprotno posodabljanje barvnih modelov objekta ter okolice na podlagi barve v neposredni okolici posameznih lokalnih značilnic. V primeru modalnosti gibanja pa je postopek dosti bolj preprost, saj model temelji zgolj na lokalnem gibanju v trenutnem časovnem koraku. V okviru modela gibanja torej niso izkoriščene možnosti zgodovine gibanja.

Pomembna omejitev načina gradnje verjetnostne porazdelitve je naivno obravnavanje posameznih modalnosti, ki, podobno kot naivni Bayesov klasifikator, predpostavlja neodvisnost posameznih modalnosti. Pogosto se namreč zgodi, da določena modalnost v določeni

situaciji zaradi odstopanja od naučenega modela ne vsebuje dela objekta, kar ima za rezultat verjetnostno porazdelitev, ki izključuje področje, ki v sliki dejansko pripada objektu. Zaradi načina združevanja verjetnostnih porazdelitev posameznih modalnosti, opisanega v enačbi (3.6), se ta napaka propagira v končno skupno verjetnostno porazdelitev. Način združevanja namreč podpira samo razreševanje napak tipa 1 (ang. *false positive*), ki jih v kontekstu razrešujemo z dodajanjem večjega števila modalnosti, za katere pa predpostavljamo, da v celoti zajamejo območje objekta¹. Poenostavljeno lahko rečemo, da trenutni pristop išče presek množic. Preprostega odgovora na ta problem ni, saj je "konjuktivno" združevanje modalnosti nujno za uspešno razločevanje področja objekta. Ena izmed heurističnih rešitev problema, ki jo nameravamo v okviru nadaljnjega dela preizkusiti, je, da se posamezne komponente v končno verjetnostno porazdelitev vključijo na podlagi trenutne ocene ustreznosti, kot je to ilustriровано na Sliki 5.1. Podobno so ustreznost ocenjevali Collins in dr. [11], le da je bil tam cilj uporabe ocen drugačen. Pri ocenjevanju ustreznosti je problem, da se sledilnik ne more opreti na neke točne podatke, na podlagi katerih bi lahko izračunal ustreznost posamezne modalnosti. Eden bolj preprostih načinov bi bil, da bi se ustreznost določala glede na to, kako dobro modalnost potrjuje obstoj trenutnih značilnic. Predvidevamo, da bo na takšni ideji temeljilo naše nadaljnje raziskovanje tega področja.



Slika 5.1: Ilustracija problema združevanja modalnosti.

Slika poenostavlja združevanje treh modalnosti na problem preseka množic. V resnici operiramo z verjetnostnimi porazdelitvami, vendar gre v tem kontekstu za podobne koncepte. Rumeni lik predstavlja dejansko obliko objekta (ang. *ground truth*), roza lik pa ocenjeno obliko objekta. V primeru (a) so upoštevane vse tri modalnosti, pri čemer modra modalnost pokvari končno oceno, ker ne vključuje velikega dela točk, ki pripadajo objektu. Če bi jo spustili iz končne ocene, kot je to prikazano v primeru (b), bi bila končna ocena dosti boljša.

Vrnimo se nazaj na geometrijski model, katerega naloga je usmerjanje posameznih zna-

¹Kot je to razvidno iz razdelka 3.4, je strogo gledano modalnost prostorske enovitosti tu neke vrste izjema, saj njena komponenta *razdalja* namerno zakrije področja na objektu, ki so že opisana z značilnicami. To je potrebno zgolj zato, da se izognemo redundantnemu opisu delov objekta.

čilnic v primeru dvoumnih rešitev. Nastop dvoumnosti je posledica *problema zaslonke* oziroma dejstva, da posamezna lokalna značilnica nima dovolj specifičnega vizualnega modela, da bi na podlagi le-tega njen položaj v sliki lahko določili nedvoumno. Geometrijski model tu služi kot vezni člen ter način organizacije podatkov v vizualnem modelu, ki le-temu omogoča doseči visoko stopnjo granularnosti in elastičnosti. Prav ti dve kvaliteti vizualnih modelov opisujeta Yang in Wu v [59], ki definirata granularnost (ang. *granularity*) kot stopnjo opisa komponent, ki sestavljajo objekt, elastičnost (ang. *elasticity*) pa kot stopnjo fleksibilnosti vizualnega modela, da se prilagaja spremembam izgleda objekta. Ob tem je treba poudariti, da geometrijski modeli, ki smo jih obravnavali v poglavju 2, ter tisti v poglavju 3 pravzaprav ne odražajo prave strukture objekta, ampak zgolj dopuščajo prostorsko deformacijo opisa izgleda objekta, ki je sestavljena *ad-hoc* s strani samega sledilnika ali operaterja. Poleg tega tudi formulacija prilagajanja opisa z optimizacijo skupne ujemalne funkcije največkrat ne odraža realnega stanja v problemu. V okviru spreminjanja izgleda objekta se ne spreminjajo zgolj prostorske relacije med značilnicami, temveč tudi izgled posameznih delov objekta. Naivno združevanje lokalnega ujemanja in geometrijskega modela v eno funkcijo ujemanja namreč neposredno predpostavlja enakovrednost obeh meril ujemanja, kar pa v splošnem ne drži, saj gre za različni modalnosti. Če se recimo lokalni izgled objekta za eno izmed lokalnih značilnic dovolj spremeni, je to nemogoče nadoknaditi neposredno z boljšim ujemanjem geometrijskega modela, velja pa tudi obratno. Ob teh opazkah se postavlja vprašanje, kakšen je sploh pomen globalnega optimuma ujemalne funkcije ter ali je geometrijski model v praksi sploh potreben ali je celo škodljiv. V naših eksperimentih (poglavje 4) smo opazili, da v primeru bolj togih tipov objektov (na primer avtomobilov) geometrijski model v primeru zmernega vpliva pripomore k boljšemu sledenju, v primeru manj togih tipov objektov pa je razlika zanemarljiva. Geometrijski model poleg tega posredno vpliva tudi na detekcijo problematičnih lokalnih značilnic. Velikokrat se namreč zgodi, da značilnica, ki ne leži na objektu, nima dovolj dobrega ujemanja ali pa leži na vizualno homogenem področju. V takem primeru geometrijski model značilnico "vleče" za objektom, ko se le-ta premika, to pa posledično zakasni detekcijo problematičnosti značilnice. Za ta problem zaenkrat še nimamo rešitve.

5.2 Vzporednice z Gestalt teorijo

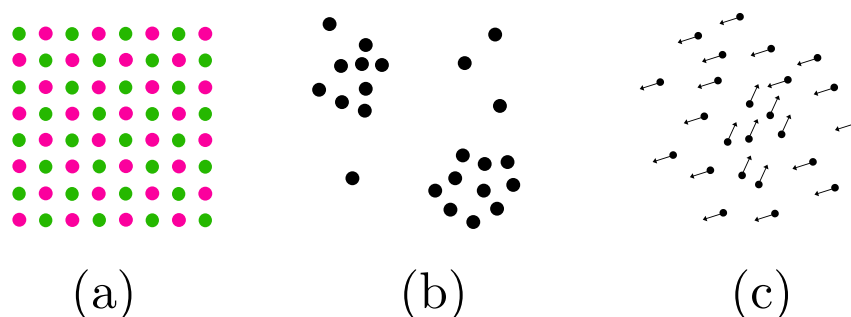
Nekatere ideje v okviru našega vizualnega modela temeljijo na pojavih, ki izhajajo iz statistike. V okviru posamezne modalnosti iščemo lokalna področja v sliki, ki se ujemajo s statistično določenim vizualnim modelom objekta. Statistika igra pomembno vlogo tudi v nekaterih psiholoških teorijah, kot je na primer Gestalt teorija. Gestalt teorija ponuja možno razlago s strani psihologov, kako bi zaznavanje objektov lahko potekalo pri človeku. Za vsako izmed uporabljenih modalnosti v našem vizualnem modelu lahko najdemo določene vzporednice s to teorijo. V nadaljevanju si bomo v tem razdelku na kratko ogledali osnove teorije ter vzporednice z opisanim vizualnim modelom.

Gestalt² psihologija se je pojavila v Nemčiji v začetku dvajsetega stoletja [20]. Njeni

²Beseda *Gestalt* je nemškega izvora in se v slovenščino nekoliko nerodno prevede kot "bistvo oziroma oblika entitete".

utemeljitelji so bili Max Wertheimer, Kurt Koffka ter Wolfgang Köhler. Osnovna ideja Gestalt psihologije je povzeta v stavku: "Celota je več kot samo vsota njenih delov". Posamezna entiteta torej ni samo skupek delov, ki jo sestavljajo, temveč je pomemben tudi način skupne interpretacije delov. Na podlagi ideje o pomenu celote je bila oblikovana množica t.i. "*Gestalt zakonov*", na podlagi katerih naj bi si človek interpretiral zaznave in jih sestavljaj v smiselne celote. Največkrat omenjeni zakoni so zakon podobnosti, zakon bližine, zakon skupne usode, zakon simetrije, zakon kontinuitete ter zakon zaprtja. Za sledenje so najbolj zanimivi prvi trije omenjeni zakoni, zato si jih bomo pogledali malo bolj podrobno.

Zakon podobnosti trdi, da človeški um v celoto povezuje dele, ki so si podobni. Podobnost lahko temelji na lastnostih izgleda, kot so barva, velikost, oblika ali svetlost. V opisanem vizualnem modelu se ta zakon odraža v modalnosti barve, opisani v razdelku 3.4. Na podoben način bi bilo mogoče uporabiti še kakšno lastnost, vendar od naštetih lastnosti prav barva nudi največjo robustnost pri nizki računski zahtevnosti. *Zakon bližine* trdi, da um povezuje dele, ki se pojavljajo eden blizu drugega v prostoru ali času. V opisanem vizualnem modelu bi temu zakonu ustrezala modalnost prostorske enovitosti (razdelek 3.4) oziroma bolj natanko njena komponenta bližine, ki omejuje vzorčenje delov na področje okolice dobrih lokalnih značilnic. *Zakon skupne usode* trdi, da um povezuje dele, ki se gibljejo v isto smer. Konceptualno lahko ta zakon povežemo z modalnostjo gibanja (razdelek 3.4) v opisanem vizualnem modelu. Trije omenjeni zakoni so s primeri prikazani tudi na Sliki 5.2.



Slika 5.2: Primeri Gestalt zakonov.

(a) - zakon podobnosti: grupiranje elementov glede na barvo, (b) - zakon bližine: grupiranje elementov glede na bližino, (c) - zakon skupne usode: puščice predstavljajo gibanje posameznih elementov, grupiranje glede na enako smer gibanja

Primerjava z Gestalt teorijo je uporabna, saj ponuja drugačen pogled na izvor posameznih uporabljenih hevristik ter ponuja nekatere dodatne ideje, katerih uporabnost v sledenju nameravamo v prihodnosti bolj podrobno raziskati. Ne smemo pa neposredno enačiti Gestalt zakonov in modalnosti, saj gre v primeru prvih za emergenten proces, v primeru drugih pa za generiranje ujemanja med opazovano sliko in danim modelom v okviru posamezne vizualne modalnosti.

5.3 Meje vizualnega sledenja

Zavedati se moramo, da je sledenje, ki temelji zgolj na tistih vizualnih lastnostih objekta, ki se jih naučimo v danem slikovnem zaporedju, omejeno. Brez znanja oziroma predpostavk o strukturnih lastnostih objekta vsak sledilnik, ki temelji izključno na teh informacijah, naleti na situacijo, kjer odpove. Pri vnašanju znanja v model pa se moramo zavedati, da tak doprinos vnaprej podanih informacij v večini primerov zmanjša splošnost sledilnika. Pri uporabi v konkretnem primeru z znanimi omejitvami je taka operacija zaželena, saj poveča robustnost sledilnika. Kljub temu pa so tudi raziskave na temo splošnega sledenja zelo uporabne, saj nudijo boljši vpogled v to, kaj je še možno doseči brez upoštevanja dodatnega znanja ter kako to znanje bolj splošno formulirati in vključiti v vizualni model na pravih mestih.

V magistrski nalogi smo se osredotočili primarno na splošno sledenje, brez prilagajanja vizualnega modela specifičnim lastnostim posameznih objektov ter scenarijev³. Seveda pa bi lahko v konkretni aplikaciji brez težav v obstoječi algoritem dodali nove omejitve. Primeri preprostih omejitev so na primer omejitve vizualne velikosti in oblike objekta, togost delov, način gibanja itd. Velikost objekta je pogojena že z omejitvijo števila lokalnih značilnic, za upoštevanje vnaprej določene oblike pa bi lahko prilagodili globalno modalnost prostorske enovitosti. Seveda bi to pomagalo le v primerih, kjer se objekt ne deformira preveč ter ne spreminja svoje velikosti v sliki. Togost geometrijskega modela smo delno že obravnavali v razdelku 4.5, kjer smo ugotovili, da različne stopnje togosti različno vplivajo na posamezne tipe objektov. Parametri vizualnega modela, ki vplivajo na togost geometrijskega modela, niso primerni za neposreden vnos ocen togosti, ki jih poda človek. Zato se njihove optimalne vrednosti najlažje določijo na podlagi eksperimentov, to pa ponuja možnost za njihovo avtomatsko učenje. Avtomatsko učenje je lahko ločena faza, kjer obnašanje sistema opazujemo na množici poznanih testnih video zaporedij, lahko pa gre za sprotno učenje oziroma prilagajanje. Slednja možnost je seveda dosti bolj zahtevna, a obenem tudi bolj uporabna. V okviru opisanega vizualnega modela bi lahko parametre togosti prilagajali na podlagi meritev stanja vizualnega modela (hitrost konvergence optimizacije, gibanje posameznih lokalnih značilnic, itd.), vendar na tem področju še nismo izvedli eksperimentov.

Poleg množice lokalnih značilnic ter togosti geometrijskega modela bi določene omejitve lahko vnesli tudi v opise modalnosti globalnega izgleda. Primera take omejitve sta že omenjeno prilagajanje modalnosti prostorske enovitosti ter omejevanje spreminjanja barvnega modela. Slednje bi bilo recimo uporabno v primerih, kjer so barvne lastnosti objekta delno poznane, delno pa se jih vizualni model nauči med sledenjem. Primer je sledenje roke ali obraza, kjer se barvni model omeji na barvo kože. Primer načina takega omejevanja je prilagajanje izbranega matematičnega modela zbranim podatkom, kjer določene parametre modela podamo vnaprej. Splošno uporabljen matematični model so mešanice Gaussovih porazdelitev, katerih ujemanje določimo z algoritmom EM (ang. *Expectation maximization*).

Bolj kompleksni mehanizmi dodajanja znanja vključujejo uporabo 3D modela objekta

³Res je, da smo eksperimentalno iskali vrednosti parametrov vizualnega modela, vendar smo v končnih eksperimentih za vse objekte uporabili isti nabor vrednosti.

ter učenje izgleda objekta ob predpostavkah, ki to učenje usmerjajo ter stabilizirajo. Uporaba 3D modela je prisotna na področju sledenja roke [40, 52], kjer je cilj ocenitev poze roke. Postopek sledenja vključuje optimizacijo glede na parametre modela (položaje posameznih sklepov). Tak pristop potrebuje za uspešno sledenje visoko ločljivost slike ter majhne spremembe objekta. V okviru naših nadaljnjih raziskav bomo obravnavali tudi možnost povezave takega 3D modela z našim bolj splošnim vizualnim modelom. Izhodiščna ideja je, da bi se modela med seboj dopolnjevala. 3D model bi ponujal bolj zanesljivo in prostorsko dobro definirano detekcijo objekta, ki bi omogočala bolj natančno postavitev značilnic, naš vizualni model pa bi v vsakem koraku nudil dobre začetne pogoje, ki bi zagotavljali hitro konvergenco parametrov 3D modela.

Učenje izgleda objekta obravnavata Yin in Collins v [60], kjer ob ob predpostavkah same geometrije objekta⁴ izgled objekta modelirata kot množico zaplat, pripisanih različnim delom geometrijskega modela. Zaplate so v množico dodane med samim sledenjem na podlagi ujemanja njihovega gibanja z ostalimi zaplatami. Tak pristop torej delno uporablja predhodno znanje o objektu, delno pa se izgleda uči sproti, kar omogoča boljše natančnost modela ter njegovo ponovno detekcijo v primeru zakrivanj. Podoben pristop bi lahko uporabili tudi v okviru opisanega vizualnega modela, vendar bi bilo potrebno v model vpeti geometrijski model, na podlagi katerega bi lahko določali relacije med posameznimi lokalnimi značilnicami.

⁴Avtorja v [60] obravnavata sledenje avtomobilov iz spreminjajočih se zornih kotov. Geometrijo avtomobilov posplošita na podolgovato sfero, kar se v okviru sledenja na avtocesti izkaže za omejitev, ki v okviru dane ločljivosti dovolj natančno opisuje obliko avtomobila.

Poglavje 6

Zaključek

V okviru te magistrske naloge smo predstavili nov pristop h konstrukciji vizualnega modela za področje vizualnega sledenja netogih objektov. Vizualni model za svoje delovanje ne potrebuje specifičnega predznanja o strukturi objekta. Najprej smo predstavili sam problem ter pregled raziskav, ki iz različnih pogledov naslavlajo podoben problem. V okviru predstavitve obstoječih raziskav smo izpostavili tudi njihove pomanjkljivosti ter nakazali, katere izmed njih smo obravnavali v okviru magistrskega dela.

Glavna tema magistrske naloge je bila predstavitev vizualnega modela. Le-ta uporablja dvoplastno kombinacijo informacij o lokalnem in o globalen izgledu objekta z možnostjo medsebojnega osveževanja obeh plasti. Spodnjo (lokalno) plast sestavlja množica lokalnih značilnic, ki za opis lokalnega izgleda objekta uporabljajo preproste sivinske histograme. Zgornjo (globalno) plast pa sestavljajo geometrijski model objekta ter opisi globalnega izgleda objekta z uporabo treh vizualnih modalnosti: barve, gibanja ter prostorske enovitosti. Sledenje poteka kot iskanje ujemanja lokalnih značilnic v sliki, pri čemer se geometrijski model uporablja kot vodilo, ki razrešuje dvoumne situacije, ki nastopijo pri iskanju ujemanja posamezne značilnice. Ko posamezne lokalne značilnice dalj časa kažejo znake nezanesljivosti (npr. zaradi spremembe lokalnega izgleda objekta), se jih odstrani iz množice ter po potrebi nadomesti z novimi. Nove značilnice se postavi na mesta v sliki, ki ustrezajo globalnemu vizualnemu modelu. Krog je sklenjen, ko se modalnosti, ki sestavljajo globalni vizualni model, osvežijo na podlagi slikovnih elementov v neposredni bližini kvalitetnih lokalnih značilnic.

Vizualni model je bil nato umeščen v preprost sledilnik, ki smo ga uporabili v okviru eksperimentov. Sami eksperimenti so bili sestavljeni iz meritev robustnosti sledilnika na devetih raznolikih video zaporedjih slik, pridobljenih iz različnih virov. Namen eksperimentov je bil dvojen: preučevanje lastnosti predlaganega vizualnega modela ter primerjava sledilnika z izborom obstoječih sledilnikov. V okviru eksperimentov smo pokazali, da ima geometrijski model večji pomen za bolj toge objekte, kar nakazuje, da je trenutna formalizacija geometrijskega modela preveč omejujoča za hitre spremembe v izgledu objekta. V okviru primerjav z drugimi sledilniki smo pokazali, da se predstavljeni sledilnik na večini testnih slikovnih zaporedij obnese bolje kot uporabljeni obstoječi algoritmi. Kljub temu obstajajo situacije, ko bolj preprosti sledilniki delujejo opazno bolje. Razlogi za to so v

različnih stopnjah omejitev v vizualnih modelih, ki omogočajo zelo dobro delovanje v določenih specifičnih scenarijih sledenja, v splošnem pa se izkažejo za preveč omejujoče.

Nadaljnji razvoj teorije in sama implementacija vizualnega modela bosta primarno temeljila na povečevanju robustnosti modela, predvsem v bolj zahtevnih video zaporedjih slik, ki na primer vključujejo tudi zakrivanje objekta. Problema se nameravamo lotiti celostno, brez hitrih rešitev, ki odpravljajo očitne simptome v specifičnih scenarijih sledenja. Obravnavane bodo tudi možnosti nadgraditve modela z dolgoročnim in bolj stabilnim učenjem ter integracijo predhodnega znanja o objektu v model. Obenem pa bo pozornost posvečena tudi hitrosti implementacije. To področje je bilo namreč do sedaj potisnjeno ob stran, vendar je za praktično uporabo sledilnika prav tako pomembno.

Dodatek A

Metodologija

V tem dodatku so predstavljene metode, na katerih temelji naše delo na vizualnem modelu. Najprej je v razdelku A.1 predstavljena metoda križne entropije, ki se v okviru vizualnega modela uporablja za skupno optimizacijo množice lokalnih značilnic ter njihovega medsebojnega geometrijskega modela, kot je to opisano v razdelku 3.2.2. Metoda RANSAC, predstavljena v razdelku A.2, se uporablja za določitev značilnic, ki na podlagi samosvojega gibanja izražajo slabo kvaliteto. V razdelku A.3 je predstavljena metoda za ocenjevanje gostote porazdelitve verjetnosti z jedri, na kateri temelji ocenjevanje verjetnostne porazdelitve na podlagi ustreznosti lokalnega gibanja. V razdelku A.4 pa je predstavljen Kalmanov filter, ki je uporabljen za ocenjevanje globalnega modela gibanja sledilnika, opisanega v razdelku 3.5.

A.1 Metoda križne entropije

Metoda križne entropije, ki jo je prvi opisal Rubinstein v [50], je splošen Monte Carlo pristop h kombinatorni in zvezni optimizaciji problemov, ki vsebujejo več lokalnih optimumov [51]. Metoda sicer izvira iz področja simulacije redkih dogodkov v zapletenih stohastičnih mrežah, kjer so nujne natančne ocene majhnih verjetnosti (minimizacija variance). Kmalu po predstavitvi je bil odkrit tudi potencial, ki ga ima metoda (s preprosto spremembo dela postopka) na področju optimizacije. Metoda je dobila ime po križni entropiji, ki jo imenujemo tudi Kullback-Liebertova divergenca in je eden izmed temeljnih konceptov moderne teorije informacije.

V grobem osnovna metoda križne entropije ter njene izpeljanke temeljijo na iterativnem postopku, ki je sestavljen iz dveh ponavljajočih se korakov:

- z uporabljenim mehanizmom (ki se razlikuje od posamezne aplikacije) generiraj naključno množico podatkovnih vzorcev, vektor, trajektorijo, itd.
- z dobljenim podatkom posodobi parametre mehanizma naključnega izbora tako, da bo le-ta generiral "boljše" vzorce.

Moč in široka uporabljivost metode temeljita prav na dejstvu, da so specifična pravila, ki se v posameznih problemih uporabljajo za posodabljanje mehanizma generiranja naključ-

nega izbora, pogosto zelo preprosta, eksplicitna ter v dobro definiranem matematičnem kontekstu optimalna. V nadaljevanju bodo opisane zgolj osnove metode križne entropije, prirejene za reševanje optimizacijskih problemov.

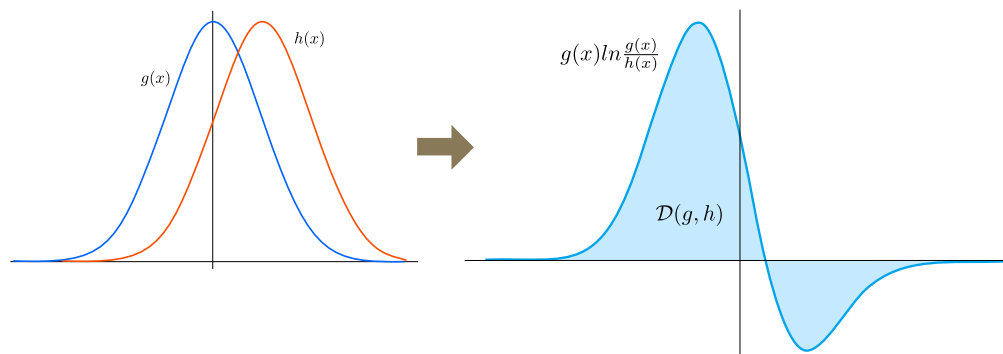
A.1.1 Kullback-Lieber-jeva divergenca

Za funkciji gostote verjetnostne porazdelitve g in h definiramo Kullback-Lieber-jevo divergenco [35] kot

$$\mathcal{D}(g, h) = \int g(x) \ln \frac{g(x)}{h(x)} dx. \quad (\text{A.1})$$

Grafični primer izračuna (A.1) je prikazan na Sliki A.1. Kot smo že omenili, imenujemo $\mathcal{D}(g, h)$ tudi križna entropija oziroma relativna entropija. V praksi obravnavamo $\mathcal{D}(g, h)$ kot metriko razdalje, saj $\mathcal{D}(g, h) \geq 0$ in $\mathcal{D}(g, h) = 0 \Leftrightarrow g(x) = h(x)$. Kljub temu je treba poudariti, da $\mathcal{D}(g, h)$ ne zadostuje vsem kriterijem metrike, saj $\mathcal{D}(g, h) \neq \mathcal{D}(h, g)$. V teoriji informacije, od koder Kullback-Lieber-jeva divergenca izvira, se le-ta interpretira kot število dodatnih bitov, ki so potrebni za opis vzorcev iz g , če uporabljamo zapis, ki temelji na h .

Metoda križne entropije temelji na iterativni minimizaciji Kullback-Lieber-jeve divergenca med funkcijama gostote verjetnostne porazdelitve g in h . Funkcija g ni znana, funkcija h pa pripada parametričnemu razredu funkcij $\{f(\cdot)\}$. V vsaki iteraciji glede na izbrano množico vzorcev iz razreda izberemo funkcijo, ki se v dani situaciji najbolj ujema z g . Ob pravilno izbranih začetnih pogojih metoda konvergira k taki funkciji.



Slika A.1: Ilustracija Kullback-Lieber-jeve divergenca.

Za dve Gaussovi funkciji gostote verjetnostne porazdelitve, prikazani na levem grafu, ustreza vrednost Kullback-Lieber-jeve divergenca površini pod grafom desne funkcije.

A.1.2 Uporaba v optimizaciji

Glavna ideja metode križne entropije, prirejene za reševanje optimizacijskih problemov, je naslednja: Predpostavimo, da želimo najti maksimum funkcije $\xi(\mathbf{x})$ v prostoru \mathcal{X} . Z γ^* označimo globalni maksimum funkcije $\xi(\mathbf{x})$, torej velja

$$\gamma^* = \max_{x \in \mathcal{X}} \xi(\mathbf{x}). \quad (\text{A.2})$$

Za generator naključnih vrednosti $\mathbf{x} \in \mathcal{X}$ definiramo družino funkcij verjetnostne porazdelitve $\{f(\cdot; \mathbf{v}), \mathbf{v} \in \mathcal{V}\}$, kjer je \mathcal{V} prostor parametrov, ki določajo posamezno funkcijo iz družine. Nato z (A.2) povežemo problem ocenjevanja

$$\ell(\gamma) = p[\xi(\mathbf{x}) \geq \gamma] = \mathbb{E}_{\mathbf{v}} I_{\{\xi(\mathbf{x}) \geq \gamma\}}, \quad (\text{A.3})$$

kjer je $\ell(\gamma)$ razmerje verjetja (ang. *likelihood ratio*) za parameter γ , $\mathbf{v} \in \mathcal{V}$ pa naključni vektor parametrov družine funkcij $\{f(\cdot; \mathbf{v})\}$. $I_{\{\xi(\mathbf{x}) \geq \gamma\}}$ predstavlja karakteristično funkcijo množice $\{\mathbf{x} | \mathbf{x} \in \mathcal{X} \wedge \xi(\mathbf{x}) \geq \gamma\}$, $\mathbb{E}_{\mathbf{v}}$ pa pričakovano vrednost te funkcije, za vrednosti \mathbf{x} vzorčene z $f(\mathbf{x}; \mathbf{v})$. Problem (A.3) imenujemo tudi povezani stohastični problem, saj po eni strani iščemo vrednost γ , po drugi pa skušamo oceniti funkcijo ℓ .

Če ocenjujemo funkcijo ℓ za nek γ , ki je blizu γ^* , lahko to obravnavamo kot ne-trivialen problem ocenjevanja verjetnosti redkega dogodka. Za ocenjevanje lahko uporabimo zaporedje funkcij $f(\cdot; \mathbf{v}_0), f(\cdot; \mathbf{v}_1), \dots$, kjer parametre \mathbf{v}_t popravljamo tako, da jih "usmerjamo" proti optimalni gostoti

$$\mathbf{v}_t = \operatorname{argmax}_{\mathbf{v} \in \mathcal{V}} \mathbb{E}_{\mathbf{v}_{t-1}} I_{\{\xi(\mathbf{x}) \geq \gamma\}} \ln f(\mathbf{x}; \mathbf{v}_t). \quad (\text{A.4})$$

Ker želimo, da obenem konvergira tudi γ proti γ^* , v iteracijo vpeljemo še iterativno popravljanje te vrednosti. Preprosta ocena nove vrednosti γ_t je definirana kot zadnji $(1 - \sigma)$ kvantil zaporedja vzorcev $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$, generiranega z $f(\mathbf{x}; \mathbf{v}_{t-1})$

$$\gamma_t = Q_{(1-\sigma)}^{\text{end}}[\xi(\mathbf{x}_q)], \quad (\text{A.5})$$

kjer konstanto $\sigma \in (0, 1)$ imenujemo parameter redkosti (ang. *rarity parameter*), s katerim nadzorujemo hitrost konvergence. Čim bližje je vrednost parametra $\sigma = 1$, tem večja je ocenjena redkost pravilne rešitve, zato je potrebna počasnejša konvergenca. Opis postopka, ki vsebuje oba koraka in generira zaporedje parov $\{(\gamma_t, \mathbf{v}_t)\}$, ki konvergira proti (γ^*, \mathbf{v}^*) , je opisan v Algoritmu 4. Pri tem z \mathbf{v}^* označimo parametre tiste funkcije gostote verjetnostne porazdelitve v družini $\{f(\cdot; \mathbf{v}), \mathbf{v} \in \mathcal{V}\}$, ki ima v okviru družine najmanjšo Kullback-Lieber-jevo divergenco z atomsko porazdelitvijo

$$\delta_{\mathbf{x}^*}(\mathbf{x}) = \begin{cases} 1, & \text{če } \mathbf{x} = \mathbf{x}^*, \\ 0, & \text{drugače} \end{cases} \quad (\text{A.6})$$

V enačbi (A.6) nastopa simbol \mathbf{x}^* , ki predstavlja točko, v kateri $\xi(\mathbf{x})$ doseže globalni maksimum

$$\mathbf{x}^* = \operatorname{argmax}_{x \in \mathcal{X}} \xi(\mathbf{x}). \quad (\text{A.7})$$

Načeloma ima funkcija $\xi(\mathbf{x})$ lahko več enakovrednih globalnih maksimumov, vendar lahko v tem primeru z izbiro urejenosti umetno določimo ustrezen \mathbf{x}^* . Oznako \mathbf{v}^* lahko sedaj formalno definiramo kot

$$\mathbf{v}^* = \operatorname{argmin}_{\mathbf{v} \in \mathcal{V}} \mathcal{D}(f(\cdot; \mathbf{v}^*), \delta_{\mathbf{x}^*}). \quad (\text{A.8})$$

Kot je bilo že omenjeno, je par (γ^*, \mathbf{v}^*) cilj konvergence, ki se ji iterativno približujemo. Ker pa velikokrat zadostuje že dovolj dober približek maksimuma, je algoritmu potrebno definirati ustavitveni pogoj. Primer splošnega pogoja je ustavitev, če za nek $d \leq t$ (na primer $d = 5$) velja

$$\gamma_t = \gamma_{t-1} = \dots = \gamma_{t-d}. \quad (\text{A.9})$$

Algoritem 4 Splošna metoda križne entropije.

Izberi začetni $\hat{\mathbf{v}}_0$. Nastavi $t = 1$ (števec korakov).

1. Z uporabo $f(\cdot; \mathbf{v})$ generiraj zaporedje vzorcev $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$,
 2. $\gamma_t = Q_{(1-\sigma)}^{end}[\xi(\mathbf{x}_q)]$,
 3. $\tilde{\mathbf{v}}_t = \operatorname{argmax}_{\mathbf{v} \in \mathcal{V}} \frac{1}{N} \sum_{i=1}^N I_{\{\xi(\mathbf{x}_i) \geq \gamma_t\}} \ln f(\mathbf{x}_i; \mathbf{v}_t)$,
 4. $\mathbf{v}_t = \alpha \tilde{\mathbf{v}}_t + (1 - \alpha) \mathbf{v}_{t-1}$,
 5. če smo izpolnili pogoj konvergence, zaključimo, drugače ponovimo od koraka 1.
-

V koraku 3 Algoritma 4 nastopa stohastična verzija enačbe (A.4). V splošnem rešitev tega koraka ni preprost problem, vendar je mogoče za določene tipe družin funkcij rešitev najti analitično. Primera takih družin sta družina eksponentnih funkcij (kamor spada tudi normalna Gaussova funkcija) ter diskretne funkcije, definirane na končni množici. Korak 4 vsebuje koeficient $\alpha \in (0, 1]$, ki uravnava glajenje spreminjanja \mathbf{v}_t . Omenjeni koeficient je pomemben pri problemih, pri katerih je \mathbf{v}_t vektor ali matrika, ki vsebuje verjetnosti. V takem primeru bi se namreč lahko zgodilo, da algoritem ob določenih začetnih pogojih konvergira k napačni rešitvi.

A.1.3 Zvezna optimizacija

Algoritem, opisan v prejšnjem razdelku, opisuje splošno metodo križne entropije za reševanje optimizacijskih problemov. Z njim lahko rešujemo tako kombinatorične kot tudi zvezne optimizacijske probleme z več ekstremin. Pri obravnavi slednjih pa lahko v Algoritem 4 vpeljemo še nekatere izboljšave.

V primeru zveznega problema iskanja maksimuma lahko formulo posodabljanja \mathbf{v}_t spremenimo tako, da vanjo vstavimo naraščajočo funkcijo $\psi(\xi(\mathbf{x}))$. Tako se korak 4 v Algoritmu 4 preoblikuje v

$$\tilde{\mathbf{v}}_t = \operatorname{argmax}_{\mathbf{v} \in \mathcal{V}} \frac{1}{N} \sum_{i=1}^N I_{\{\xi(\mathbf{x}_i) \geq \gamma_t\}} \psi(\xi(\mathbf{x}_i)) \ln f(\mathbf{x}_i; \mathbf{v}_t).$$

Funkcija $\psi(s)$ deluje kot utež, ki poveča vpliv bolj ustreznih vzorcev. Eksperimentalni rezultati [51] so pokazali, da linearna funkcija $\psi(s) = s$ v določenih primerih vodi do povečanja hitrosti konvergence algoritma. Po drugi strani pa se izkaže, da polinomi višjih stopenj niso priporočljivi. Njihova uporaba namreč poveča možnost konvergence k lokalnemu optimumu.

Druga izboljšava, predstavljena v [51], v algoritem vnese dodatno plast prilagodljivosti. Ideja izboljšave je, da se velikost množice vzorcev N z vsako iteracijo spreminja glede na "težavnost" problema v danem koraku. Če predpostavimo, da gre za problem iskanja maksimuma, lahko elemente vzorca uredimo tako, da velja

$$\xi(\mathbf{x}_{t,1}) \leq \xi(\mathbf{x}_{t,2}) \leq \dots \leq \xi(\mathbf{x}_{t,N_t}) = S_t^*.$$

Glavna predpostavka omenjene izboljšave je, da v posamezni iteraciji parametre posodobimo na podlagi fiksnega števila "elitnih" elementov. Množica ϵ_t vsebuje N^{elite} najboljših vzorcev množice. V spremenjenem algoritmu vhodna parametra N in σ nadomestimo s parametrom N^{elite} . Spremenjena koraka 2 in 3 v Algoritmu 4 sta:

$$\gamma_t = \xi(\mathbf{x}_{t,N_t - N^{elite} + 1})$$

in

$$\tilde{\mathbf{v}}_t = \operatorname{argmax}_{\mathbf{v} \in \mathcal{V}} \frac{1}{N} \sum_{\mathbf{x} \in \epsilon_t} \ln f(\mathbf{x}_i; \mathbf{v}_t).$$

Vsaka iteracija se začne z najnižjo velikostjo množice vzorcev $N_t = N^{min}$. Če velja pogoj $S_t^* > S_{t-1}^* \vee \gamma_t > \gamma_{t-1}$, posodobimo parametre ter nadaljujemo z naslednjo iteracijo. Če pogoj ne velja, povečamo N_t ter ponovimo vzorčenje. Povečevanje velikosti množice vzorcev ponavljamo, dokler ne dosežemo N^{max} . V tem primeru je dobljena rešitev označena kot nezanesljiva.

A.2 RANSAC

Konsenz naključnih vzorcev (ang. *Random Sample Consensus*), okrajšano RANSAC, je iterativna metoda, s katero ocenimo parametre matematičnega modela iz množice podatkov, ki vsebuje tudi odstopajoče primerke (ang. *outliers*). Ti odstopajoči podatki izvirajo recimo iz napak pri merjenju in lahko pokvarijo končno oceno parametrov modela. Gre za stohastično metodo, ki zagotavlja dovolj dober rezultat z določeno verjetnostjo, pri čemer se verjetnost povečuje s številom iteracij. Metoda, ki sta jo prvič objavila Fischler in Bolles v [16], je zelo razširjena v računalniškem vidu. Uporablja se pri iskanju ujemanj v slikah [44], kalibraciji sistemov kamer [48], do neke mere pa tudi v sledenju [60].

Algoritem temelji na preprosti ideji: Za majhno podmnožico naključno izbranih vzorcev iz množice podatkov zgradimo matematični model ter preverimo, koliko ostalih vzorcev mu še pripada. Na podlagi tega lahko ocenimo ustreznost modela. Postopek ponavljamo dokler z dovolj visoko verjetnostjo ne določimo dovolj kvalitetnega modela [17]. Ideja je bolj

podrobno povzeta v Algoritmu 5. Na Sliki A.2 pa je prikazan preprost primer uporabe RANSAC metode za določitev premice v množici točk.

Algoritem 5 Konsenz naključnih vzorcev.

Vhodni podatki:

1. n : najmanjše število vzorcev, ki jih potrebujemo za gradnjo matematičnega modela
2. k : število iteracij
3. t : prag, s katerim so določeni vzorci, ki še pripadajo modelu
4. d : potrebno število vzorcev, ki še pripadajo modelu, na podlagi katerega lahko sklepamo, da model dobro opisuje podatke
5. S : množica podatkov

Ponovi k -krat:

1. Naključno vzorči n vzorcev iz množice podatkov $S \rightarrow S_k$
2. Na podlagi S_k določi matematični model M_k
3. $S_k^* = \{s \mid s \in S \setminus S_k \wedge \delta_k(s) \leq t\}$
4. Če velja $|S_k^*| \geq d$, je M_k dober model. V množico \mathcal{M} ustreznih modelov dodamo M_k^* , ki ga določimo na podlagi $S_k \cup S_k^*$.

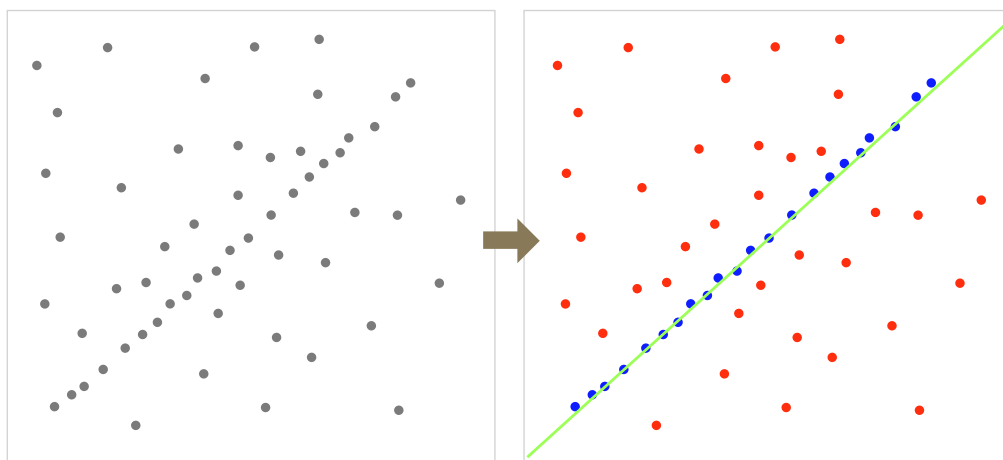
Rezultat algoritma je najboljši model iz \mathcal{M} (z najmanjšo napako nad podatki, ki so bili uporabljeni pri njegovi konstrukciji).

A.2.1 Določitev parametrov algoritma

Uspešnost metode RANSAC je v veliki meri odvisna od izbranih parametrov. Parametra t in d sta odvisna od problema ter od lastnosti sestave množice podatkov. Njuni vrednosti tako temeljita na eksperimentalnih rezultatih. Določitev parametra k pa ima dobro osnovano formulacijo, ki nam za določeno vrednost točno pove, kakšna je verjetnost, da najdemo optimalne parametre modela. Če predpostavimo, da za gradnjo modela potrebujemo n elementov ter, da je w odstotek kvalitetnih elementov v množici, lahko zapišemo pričakovano vrednost parametra k kot

$$\mathbb{E}[k] = w^n + 2(1 - w^n)w^n + 3(1 - w^n)^2w^n + \dots = w^{-n}. \quad (\text{A.10})$$

Kljub temu, da w v večini primerov vnaprej ni natančno poznan, je dovolj, da uporabimo grobo oceno. Če želimo večjo gotovost, lahko za oceno uporabimo še oceno standardnega odklona



Slika A.2: Primer uporabe metode RANSAC.

Ilustracija preprostega primera, ki ponazarja delovanje metode RANSAC. V množici točk (leva slika) hočemo potegniti premico (matematični model), ki bo ustrezala čim več točkam. Ob ustrezno izbranih vhodnih parametrih algoritem določi ustrezno premico (desna slika), katere parametri so določeni na podlagi modrih točk. Rdeče točke so odstopajoči primerki oziroma šum.

$$\sigma(k) = \frac{\sqrt{1 - w^n}}{w^n}. \quad (\text{A.11})$$

Drugi pogled na problem določitve parametra k je, da določimo število naključno izbranih elementov, ki še zagotavlja nizko verjetnost, da bomo izbrali samo slabe elemente

$$z = (1 - w^n)^k,$$

kar pomeni, da velja

$$k = \frac{\ln z}{\ln(1 - w^n)}.$$

A.3 Ocenjevanje gostote porazdelitve verjetnosti z jedri

Ocenjevanje verjetnostne porazdelitve je pomembno v primerih, ko imamo na razpolago končno množico meritev, dobljenih iz neznane verjetnostne porazdelitve, ki pa jo želimo čim boljše oceniti. Recimo, da naključno vzorčimo nepoznano verjetnostno porazdelitev $p(\mathbf{x})$ v D dimenzionalnem prostoru¹, na podlagi vzorcev pa želimo oceniti samo verjetnostno porazdelitev [5]. Če je \mathcal{R} lokalno območje, ki vsebuje \mathbf{x} , je verjetnostna "masa" povezana z območjem

$$P = \int_{\mathcal{R}} p(\mathbf{x}) d\mathbf{x}. \quad (\text{A.12})$$

¹Za prostor bomo privzeli, da zanj veljajo lastnosti Evklidskega prostora.

Množica meritev obsega N elementov. Ker ima vsak element verjetnost P , da se nahaja v območju \mathcal{R} , je skupno število elementov, ki se nahajajo v območju \mathcal{R} , porazdeljeno na podlagi binomske porazdelitve

$$\text{Bin}(K | N, P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K}. \quad (\text{A.13})$$

Za velik N bo imela binomska porazdelitev (A.13) zelo oster ekstrem okoli pričakovane vrednosti P , kar pomeni, da

$$K \simeq NP. \quad (\text{A.14})$$

Če pa predpostavimo, da je področje \mathcal{R} dovolj majhno, da je verjetnostna porazdelitev znotraj njega skoraj konstantna, lahko zapišemo oceno gostote kot

$$P \simeq p(\mathbf{x})V, \quad (\text{A.15})$$

kjer V predstavlja volumen področja \mathcal{R} . Če združimo (A.14) in (A.15), tako dobimo oceno gostote $p(\mathbf{x})$

$$p(\mathbf{x}) = \frac{K}{NV}. \quad (\text{A.16})$$

Ocena (A.16) temelji na dveh nasprotujočih si predpostavkah. Prva predpostavka je, da je področje \mathcal{R} dovolj majhno, druga pa, da je obenem dovolj veliko, da je število elementov, ki pripadajo področju, dovolj visoko, da ima binomska porazdelitev (A.13) oster ekstrem. Od tu lahko idejo razvijamo v dve smeri. Prva je, da fiksiramo K in spreminjamo V , druga pa, da fiksiramo V in spreminjamo K . V limiti $N \rightarrow \infty$ obe poti pripeljeta do enakovrednega rezultata, v nadaljevanju pa bo opisana slednja možnost, ki jo imenujemo tudi *ocenjevanje gostote porazdelitve verjetnosti z jedri*.

Za začetek predpostavimo, da je področje \mathcal{R} hiper-kocka s središčem v točki \mathbf{x} , v kateri hočemo ugotoviti gostoto verjetnosti. Z namenom preštetja točk, ki spadajo v področje, definiramo funkcijo

$$k(\mathbf{u}) = \begin{cases} 1, & |u_i| \leq 1/2 \quad i = 1, \dots, D \\ 0, & \text{drugače} \end{cases}, \quad (\text{A.17})$$

ki opisuje enotsko kocko v koordinatnem izhodišču. Funkcija $k(\mathbf{u})$ je primer jedrne funkcije, ki jo v danem kontekstu imenujemo tudi *Parzen-ovo okno*. Z njo zapišemo število točk znotraj hiper-kocke kot

$$K = \sum_{i=1}^N k\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right), \quad (\text{A.18})$$

ko pa vstavimo enačbo (A.18) v enačbo (A.16), dobimo ocenjeno gostoto v točki \mathbf{x}

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h^D} k\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right), \quad (\text{A.19})$$

kjer h označuje velikost kocke, h^D pa posledično volumen V .

Problem uporabe kockastega jedra so nezveznosti, ki se pojavijo na robovih kocke. Podoben problem srečamo tudi pri uporabi histogramov. Zato se bolj pogosto uporablja bolj gladek model gostote, ki uporablja Gaussovo jedro. V primeru enodimenzionalnega problema (A.16) zapišemo kot

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{\sqrt{2\pi h^2}} e^{-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2h^2}}, \quad (\text{A.20})$$

kjer h predstavlja standardni odklon Gauss-ovih jeder. Posplošeno na več dimenzij pa kot

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{\sqrt{(2\pi)^d |\mathbf{H}|}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)^T \mathbf{H}^{-1}(\mathbf{x}-\mathbf{x}_i)}, \quad (\text{A.21})$$

kjer \mathbf{H} predstavlja kovariančno matriko oziroma večdimenzionalno posplošitev parametra h , d pa število dimenzij.

Na Sliki A.3 je prikazan primer ocene funkcije gostote verjetnosti v enodimenzionalnem primeru za različne vrednosti h . Opazno je, da h igra vlogo parametra glajenja in da je treba iskati kompromis med preobčutljivostjo majhne vrednosti h ter preveč intenzivnim glajenjem večje vrednosti h . Optimalno vrednost h lahko ocenimo s Parzen-ovo okensko metodo [45]. Problem te metode je, da je računsko zahtevna, saj čas, ki ga potrebuje za oceno, narašča linearno z velikostjo množice podatkov.

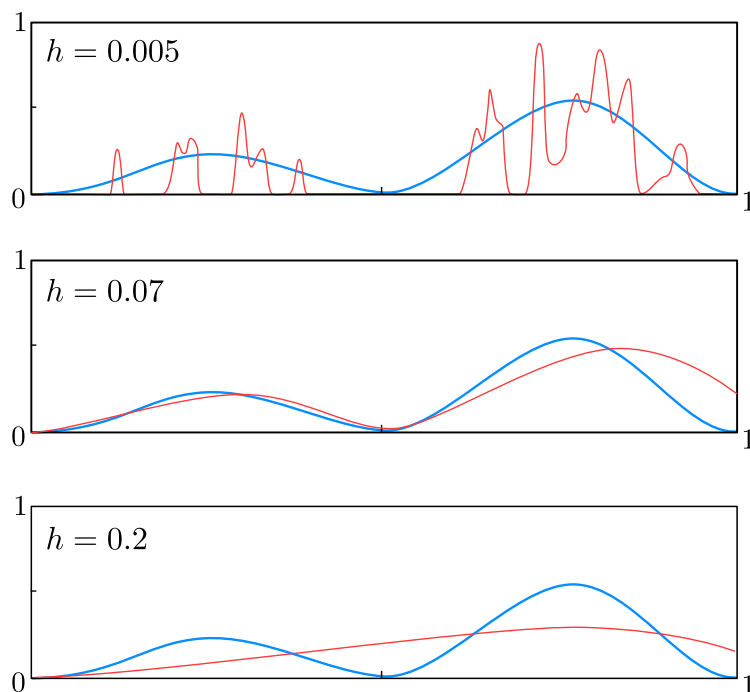
A.4 Bayesov rekurzivni filter

Delovanje Kalmanovega filtra najlažje pojasnimo z njegovo umestitvijo v širši krog Bayesovih rekurzivnih filtrov. Bayesovi rekurzivni filtri so matematične metode, katerih namen je, da za časovno zaporedje šumnih meritev določijo vrednosti, ki so načeloma bolj točne kot same meritve. V tem razdelku je torej na kratko izpeljan izvor Kalmanovega filtra, na koncu pa so predstavljene tudi njegove teoretične osnove s poudarkom na uporabi v aplikacijah sledenja.

A.4.1 Nelinearni filtri

Koncept nelinearnih filtrov je v statističnih in inženirskih krogih preučevan že več kot tri desetletja. Nelinearne filtre uporabljamo, ko imamo opravka z zaporednim ocenjevanjem stanja dinamičnega sistema na podlagi zaporedja šumnih meritev [49]. Pri tem je ponavadi predpostavljena diskretna formulacija časa, stanje pa se modelira kot vektor v prostoru vseh možnih stanj. Vektor stanja vsebuje vse informacije, ki v danem trenutku opišejo stanje sistema. V primeru sledenja vektor opisuje kinetične lastnosti objekta, ki mu sledimo. Vektorji meritev opisujejo posamezne šumne meritve, ki so neposredno povezane z vektorjem stanja. Vektor meritve je ponavadi nižje dimenzionalen kot vektor stanja.

Da lahko analiziramo dinamični sistem ter sklepamo o njegovih spremembah, moramo vedeti vsaj dve stvari. Kot prvo moramo poznati model, ki opisuje razvoj stanja sistema



Slika A.3: Ilustracija ocene funkcije gostote verjetnosti za različne vrednosti h . Modra krivulja prikazuje nepoznano funkcijo gostote, rdeča pa oceno za različne vrednosti parametra h . Slika je povzeta po [5].

v času. V tem primeru govorimo o modelu dinamike sistema. Poleg tega potrebujemo tudi model, ki povezuje meritve s stanjem sistema oziroma meritveni model. Eden izmed načinov formulacije teh dveh sistemov je verjetnostna formulacija, ki omogoča uporabo metod Bayesove statistike. Taka formulacija nudi natančen in splošen okvir za probleme ocenjevanja stanja zelo raznolikih sistemov.

V okviru Bayesove statistike je naš cilj pri ocenjevanju stanja sistema določitev a posteriorne verjetnostne porazdelitve stanja, temelječe na vseh danih podatkih oziroma meritvah. Ker taka verjetnostna porazdelitev vsebuje celotno dostopno informacijo v statistični obliki, jo lahko smatramo za popolno rešitev problema ocenjevanja, na podlagi nje pa lahko določimo tudi najverjetnejše stanje sistema ter natančnost ocene.

V primeru, ko je v vsakem časovnem koraku dostopna nova meritev, zahtevana pa nova ocena stanja, lahko uporabimo *rekurziven filter*. Rekurzivni filter je pristop, ki, za razliko od paketne obdelave, vhodne podatke obdeluje zaporedno, brez potrebe po ponovni uporabi starih podatkov. To pomeni, da lahko uporabljene meritve po njihovi uporabi zavržemo. Rekurzivni filter je sestavljen iz dveh korakov: napovedi (ang. *prediction*) in posodobitve (ang. *update*). V okviru napovedi se z uporabo modela dinamike sistema na podlagi stanja v prejšnjem časovnem koraku določi nova verjetnostna porazdelitev vektorja stanja sistema. Ker je sprememba stanja ponavadi podvržena nepoznanim vplivom, napoved verjetnostno porazdelitev ustrezno deformira in razširi. Posodobitev dobljeno verjetnostno porazdelitev

stanja z uporabo nove meritve na podlagi Bayesovega teorema popravi. Tipično se verjetnostna porazdelitev ob dostopnosti informacij nove meritve skrči.

A.4.2 Osnovni problem

Z namenom formalne definicije nelinearnih rekurzivnih filtrov definirajmo stanje sistema kot $\mathbf{x}_t \in \mathbb{R}^N$, kjer je N dimenzija prostora stanj, t pa indeks časovnega koraka. Stanje se spreminja po verjetnostnemu modelu

$$\mathbf{x}_t = \mathbf{f}_{t-1}(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}), \quad (\text{A.22})$$

kjer \mathbf{f}_{t-1} predstavlja poznano, potencialno ne-linearno funkcijo stanja \mathbf{x}_{t-1} ter šuma procesa \mathbf{v}_{t-1} . Šum procesa (ang. *process noise*) opisuje vse možne napake pri modeliranju ter nepredvidene vplive na stanje sistema. Cilj nelinearnega filtriranja je rekurzivno ocenjevanje stanja \mathbf{x}_t na podlagi meritve $\mathbf{z}_t \in \mathbb{R}^M$, ki je s stanjem povezana na podlagi funkcije

$$\mathbf{z}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{w}_t), \quad (\text{A.23})$$

kjer \mathbf{h}_t predstavlja poznano, potencialno ne-linearno funkcijo stanja \mathbf{x}_t ter šuma meritev (ang. *measurement noise*) \mathbf{w}_t . Za \mathbf{v}_{t-1} in \mathbf{w}_t predpostavljamo, da sta vzorčena iz medsebojno neodvisnih verjetnostnih porazdelitev, ki sta neodvisni tudi od verjetnostne porazdelitve stanja.

Naloga nelinearnega filtra je določitev verjetnostne porazdelitve $p(\mathbf{x}_t|\mathbf{Z}_t)$, kjer z \mathbf{Z}_t označimo zaporedje meritev za $t = 1, \dots, t$. Le-to dobimo rekurzivno z uporabo korakov, omenjenih v razdelku A.4.1. Predpostavimo, da na začetku časovnega koraka t poznamo $p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1})$, torej oceno iz prejšnjega časovnega koraka. V okviru napovedi uporabimo model (A.22), z uporabo katerega dobimo napovedano verjetnostno gostoto z uporabo Chapman-Kolmogorov-e enačbe:

$$p(\mathbf{x}_t|\mathbf{Z}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1})d\mathbf{x}_{t-1}. \quad (\text{A.24})$$

V (A.24) je že upoštevano, da (A.22) opisuje Markovski proces prvega reda, zato velja $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{Z}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Verjetnostno porazdelitev $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ definirata model dinamike sistema (A.22) ter poznana statistika šuma procesa \mathbf{v}_{t-1} .

V okviru koraka posodobitve uporabimo novo meritev \mathbf{z}_t v Bayesovem pravilu, s katerim posodobimo napovedano verjetnostno gostoto

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Z}_{t-1})}{p(\mathbf{z}_t|\mathbf{Z}_{t-1})}, \quad (\text{A.25})$$

kjer je normalizacijska konstanta $p(\mathbf{z}_t|\mathbf{Z}_{t-1})$ definirana kot

$$p(\mathbf{z}_t|\mathbf{Z}_{t-1}) = \int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Z}_{t-1})d\mathbf{x}_t \quad (\text{A.26})$$

ter je odvisna od $p(\mathbf{z}_t|\mathbf{x}_t)$, ki jo definirata meritveni model (A.23) in poznana statistika šuma meritve \mathbf{w}_t .

Problem, ki nastopi, je, da je rekurzivno ocenjevanje posteriorne verjetnostne porazdelitve, opisano v (A.24) in (A.25), samo konceptualna rešitev. S tem je mišljeno, da ocen v splošnem ne moremo izračunati analitično. Splošna obravnava bi namreč zahtevala hrambo celotne verjetnostne porazdelitve, kar je v najsplošnejšem primeru enakovredno hrambi neskončno-dimenzionalnega vektorja. Analitični pristop je torej omejen na specifične podrazrede. Enega izmed njih obravnava tudi Kalmanov filter, opisan v naslednjem razdelku.

A.4.3 Kalmanov filter

Kalmanov filter [28] je imenovan po madžarsko-ameriškem elektroinženirju in matematiku Rudolfu E. Kalmanu. Posebnost Kalmanovega filtra je, da predpostavlja, da a posteriori verjetnostna porazdelitev v vsakem koraku spada v družino Gaussovih porazdelitev in je tako enolično določena z dvema parametroma: srednjo vrednostjo in kovarianco. Taka omejitev omogoča hitro in preprosto računanje in posodabljanje, zaradi česar je Kalmanov filter uporabljan v sistemih od avtopilotov ter satelitske navigacije do napovedovanja vremena.

Če je $p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1})$ Gauss-ova porazdelitev, velja, da je Gausso-va porazdelitev tudi $p(\mathbf{x}_t|\mathbf{Z}_t)$, v kolikor veljajo naslednje predpostavke [49]:

- \mathbf{v}_{t-1} in \mathbf{w}_t sta vzorčeni iz Gauss-ovih porazdelitev s poznanimi parametri,
- $\mathbf{f}_{t-1}(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})$ je poznana linearna funkcija,
- $\mathbf{h}_t(\mathbf{x}_t, \mathbf{w}_t)$ je poznana linearna funkcija.

Predpostavimo torej, da lahko (A.22) in (A.23) zapišemo kot

$$\mathbf{x}_t = \mathbf{F}_{t-1}\mathbf{x}_{t-1} + \mathbf{v}_{t-1} \quad (\text{A.27})$$

ter

$$\mathbf{z}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{w}_t, \quad (\text{A.28})$$

kjer je \mathbf{F}_{t-1} matrika velikosti $N \times N$, \mathbf{H}_t pa matrika velikosti $M \times N$. Vektorja šuma \mathbf{v}_{t-1} in \mathbf{w}_t sta vzorčena na podlagi medsebojno neodvisnih Gauss-ovih porazdelitev z nično srednjo vrednostjo ter kovariancama \mathbf{Q}_{t-1} ter \mathbf{R}_t . Indeksi na vseh štirih matrikah nakazujejo, da se lahko matrike tekom časa načeloma tudi spreminjajo.

Algoritem Kalmanovega filtra, izpeljan z uporabo (A.24) in (A.25), lahko zapišemo kot naslednje tri rekurzivne odnose:

$$p(\mathbf{x}_{t-1}|\mathbf{Z}_{t-1}) = \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mathbf{x}}_{t-1|t-1}, \mathbf{P}_{t-1|t-1})$$

$$p(\mathbf{x}_t|\mathbf{Z}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t-1}, \mathbf{P}_{t|t-1})$$

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \mathbf{P}_{t|t}),$$

kjer $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{P})$ označuje Gauss-ovo porazdelitev z argumentom \mathbf{x} , srednjo vrednostjo \mathbf{m} ter kovarianco \mathbf{P} . Omenjene srednje vrednosti ter kovariance se zračunajo po naslednjih formulah:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_{t-1}\hat{\mathbf{x}}_{t-1|t-1}$$

$$\mathbf{P}_{t|t-1} = \mathbf{Q}_{t-1} + \mathbf{F}_{t-1}\mathbf{P}_{t-1|t-1}\mathbf{F}_{t-1}^T$$

$$\mathbf{x}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1})$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{S}_t\mathbf{K}_t^T,$$

kjer je

$$\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t$$

kovarianca inovacijskega člana $\nu_t = \mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}$,

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1}$$

pa je Kalmanova pridobitev (ang. *Kalman gain*). Če povzamemo, lahko rečemo, da Kalmanov filter torej rekurzivno računa srednjo vrednost in kovarianco Gauss-ove funkcije, ki opisuje aposteriorno verjetnost $p(\mathbf{x}_t|\mathbf{Z}_t)$. V primeru, da držijo prej omenjene (zelo stroge) omejitve, lahko rečemo, da je rešitev, ki jo nudi Kalmanov filter, optimalna. V praksi, v sledenju objektov, te predpostavke skoraj nikoli ne držijo popolnoma. Kljub temu lahko Kalmanov filter uporabimo za približek oziroma oceno dejanskega položaja objekta, saj s tem zmanjšamo področje iskanja ter tako pospešimo sledenje.

Dodatek B

Uporabljeno programje

B.1 Odprtokodna orodja in knjižnice

Pri izdelavi magistrske naloge so nam bila v veliko pomoč razna odprtokodna orodja. Zato se nam zdi prav, da jih, z namenom zahvale vsem ljudem, ki jih ustvarjajo, vsaj navedemo.

- Ubuntu (<http://www.ubuntu.com/>)
- OpenCV (<http://opencv.willowgarage.com/wiki/>)
- L^AT_EX (<http://www.latex-project.org/>)
- Inkscape (<http://www.inkscape.org/>)
- Subversion (<http://subversion.tigris.org/>)
- Processing (<http://processing.org/>)
- Multivalent (<http://multivalent.sourceforge.net/>)

B.2 Izvorna koda sledilnikov

V okviru eksperimentov, opisanih v razdelku 4.5, so bili posredno ali neposredno uporabljeni naslednji skupki izvorne kode:

- Zahvaljujemo se Braisu Martinezu in Xavierju Binefaju, za dostop do njune kode sledilnika, opisanega v [42], na podlagi katere smo lahko izvedli primerjave.
- Implementacija sledilnika, ki uporablja jato značilnic, je delno osnovana na konceptih iz projekta HandVU:
<http://www.movesinstitute.org/~kolsch/HandVu/HandVu.html>
- Uporabljena implementacija sledilnika, ki uporablja filter z delci, temelji na kodi, ki jo je napisal Sebastien Paris in je dostopna na:
<http://www.mathworks.com/matlabcentral/fileexchange/17960>

Literatura

- [1] A. Adam, E. Rivlin, in I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, strani 798–805. IEEE Computer Society, 2006.
- [2] V. Badrinarayanan, F. Le Clerc, L. Oisel, in P. Perez. Geometric Layout Based Graphical Model for Multi-Part Object Tracking. Objavljeno v *International Workshop on Visual Surveillance*, 2008.
- [3] V. Badrinarayanan, P. Perez, F. Le Clerc, in L. Oisel. Probabilistic Color and Adaptive Multi-Feature Tracking with Dynamically Switched Priority Between Cues. Objavljeno v *IEEE International Conference on Computer Vision*, strani 1–8, 2007.
- [4] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.
- [5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Oktober 2007.
- [6] P. Brasnett, L. Mihaylova, D. Bull, in N. Canagarajah. Sequential Monte Carlo tracking by fusing multiple cues in video sequences. *Image and Vision Computing*, 25(8):1217–1227, 2007.
- [7] C. Bregler, A. Hertzmann, in H. Biermann. Recovering Non-Rigid 3D Shape from Image Streams. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, strani 690—696, 2000.
- [8] G. J. Brostow in R. Cipolla. Unsupervised Bayesian Detection of Independent Motion in Crowds. Objavljeno v *IEEE Conference on Computer Vision and Pattern Recognition.*, del 1, strani 594–601, Los Alamitos, CA, USA, 2006. IEEE Computer Society.
- [9] A. Buchanan in A. Fitzgibbon. Combining local and global motion models for feature point tracking. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, strani 1–8, 2007.
- [10] W.-Y. Chang, C.-S. Chen, in Y.-P. Hung. Tracking by Parts: A Bayesian Approach With Component Collaboration. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):375–388, 2009.

- [11] R. T. Collins, Y. Liu, in M. Leordeanu. Online Selection of Discriminative Tracking Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.
- [12] D. Comaniciu, V. Ramesh, in P. Meer. Real-time tracking of non-rigid objects using mean shift. Objavljeno v *IEEE Conference on Computer Vision and Pattern Recognition*, del 2, strani 142–149, 2000.
- [13] D. Comaniciu, V. Ramesh, in P. Meer. Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003.
- [14] A. Doucet, N. de Freitas, in N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 1 edition, Junij 2001.
- [15] Z. Fan, M. Yang, in Y. Wu. Multiple Collaborative Kernel Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1268–1273, 2007.
- [16] M. A. Fischler in R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [17] D. A. Forsyth in J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, Avgust 2002.
- [18] A. Fossati, E. Arnaud, R. Horaud, in P. Fua. Tracking Articulated Bodies using Generalized Expectation Maximization. Objavljeno v *CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*, Anchorage, AK, USA, 2008.
- [19] Y. Freund in R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Objavljeno v *European Conference on Computational Learning Theory*, strani 23–37. Springer-Verlag, 1995.
- [20] J. D. Friedenbergr in G. Silverman. *Cognitive Science: An Introduction to the Study of Mind*. Sage Publications, Inc, 1 edition, September 2005.
- [21] H. Grabner in H. Bischof. On-line Boosting and Vision. Objavljeno v *IEEE Conference on Computer Vision and Pattern Recognition*, del 1, strani 260–267, Junij 2006.
- [22] H. Grabner, M. Grabner, in H. Bischof. Real-Time Tracking via On-line Boosting. Objavljeno v *British Machine Vision Conference*, 2006.
- [23] G. D. Hager, M. Dewan, in C. V. Stewart. Multiple kernel tracking with SSD. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, del 1, strani 790–797, 2004.
- [24] C. Harris in M. Stephens. A Combined Corner and Edge Detection. Objavljeno v *Alvey Vision Conference*, strani 151, 147, 1988.

-
- [25] J. Hoey. Tracking using flocks of features, with application to assisted handwashing. Objavljeno v *Proceedings of British Machine Vision Conference*, 2006.
- [26] M. Irani. Multi-Frame Optical Flow Estimation Using Subspace Constraints. Objavljeno v *IEEE International Conference on Computer Vision*, 1999.
- [27] M. Isard in J. MacCormick. BraMBLe: a Bayesian multiple-blob tracker. Objavljeno v *IEEE International Conference on Computer Vision*, del 2, strani 34–41, 2001.
- [28] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the American Society of Mechanical Engineers, Journal of Basic Engineering*, 82:34–45, 1960.
- [29] Z. Khan, T. Balch, in F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1805–1819, 2005.
- [30] M. Kölsch in M. Turk. Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration. Objavljeno v *Conference on Computer Vision and Pattern Recognition Workshop*, del 10, stran 158, Washington, DC, USA, 2004. IEEE Computer Society.
- [31] M. Kristan in A. Leonardis. Multivariate Online Kernel Density Estimation. Objavljeno v *Computer Vision Winter Workshop*, stran accepted, 2010.
- [32] M. Kristan, J. Perš, S. Kovačič, in A. Leonardis. A Local-motion-based probabilistic model for visual tracking. *Pattern Recognition*, 2008.
- [33] M. Kristan, J. Perš, A. Leonardis, in S. Kovačič. A hierarchical dynamic model for tracking in sports. Objavljeno v *Electrotechnical and Computer Science Conference*, September 2007.
- [34] M. Kristan, J. Perš, A. Leonardis, in S. Kovačič. Probabilistic tracking using optical flow to resolve color ambiguities. Objavljeno v Michael Grabner in Helmut Grabner, editors, *Computer Vision Winter Workshop*, strani 3–10, 2007.
- [35] S. Kullback in R. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):86, 79, 1951.
- [36] T. Kurata, T. Okuma, M. Kourogi, in K. Sakaue. The Hand Mouse: GMM hand-color classification and mean shift tracking. Objavljeno v *IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, strani 119–124, 2001.
- [37] J. S. Kwon in K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive Basin Hopping Monte Carlo sampling. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, strani 1208–1215, 2009.

- [38] P. Li in F. Chaumette. Image Cues Fusion for Object Tracking Based on Particle Filter. Objavljeno v *International Conference on Articulated Motion and Deformable Objects*, strani 99–107, 2004.
- [39] X. Rong Li in V. Jilkov P. Survey of Maneuvering Target Tracking: Dynamic Models. *IEEE Aerospace and Electronic Systems*, 39(4):1333–1363, Oktober 2003.
- [40] J.Y. Lin, Ying Wu, in T.S. Huang. 3D model-based hand tracking using stochastic direct search method. Objavljeno v *IEEE International Conference on Automatic Face and Gesture Recognition*, strani 693—698, 2004.
- [41] B. D. Lucas in T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI). Objavljeno v *International Joint Conference on Artificial Intelligence*, strani 674–679, April 1981.
- [42] B. Martinez in X. Binefa. Piecewise affine kernel tracking for non-planar targets. *Pattern Recognition*, 41(12):3682–3691, 2008.
- [43] S. M. S. Nejhumi, J. Ho, in M. H. Yang. Visual Tracking with Histograms and Articulating Blocks. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, Anchorage, USA, 2008.
- [44] M. Özuysal, M. Calonder, V. Lepetit, in P. Fua. Fast Keypoint Recognition Using Random Ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, 2010.
- [45] E. Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
- [46] P. Pérez, C. Hue, J. Vermaak, in M. Gangnet. Color-Based Probabilistic Tracking. Objavljeno v *European Conference on Computer Vision*, del 1, strani 661–675. Springer-Verlag, 2002.
- [47] F. Porikli, O. Tuzel, in P. Meer. Covariance Tracking using Model Update Based on Means on Riemannian Manifolds. Technical report, Mitsubishi Electric Research Laboratories, 2006.
- [48] P. Pritchett in A. Zisserman. Wide Baseline Stereo Matching. Objavljeno v *IEEE International Conference on Computer Vision*, strani 754—760, 1998.
- [49] B. Ristic, S. Arulampalam, in N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, Februar 2004.
- [50] R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, Maj 1997.
- [51] R. Y. Rubinstein in D. P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer, Julij 2004.

-
- [52] B. Stenger, P. Mendonca, in R. Cipolla. Model-based hand tracking using an unscented kalman filter. Objavljeno v *British Machine Vision Conference*, Manchester, UK, September 2001.
- [53] F. Tang in H. Tao. Object tracking with dynamic feature graph. Objavljeno v *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, strani 25–32, 2005.
- [54] L. Torresani, D. B. Yang, E. J. Alexander, in C. Bregler. Tracking and modeling non-rigid objects with rank constraints. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, del 1, strani 493–500, 2001.
- [55] S. Tran in L. Davis. Robust Object Tracking with Regional Affine Invariant Features. Objavljeno v *IEEE International Conference on Computer Vision*, strani 1–8, 2007.
- [56] L. Čehovin, M. Kristan, in A. Leonardis. Tracking Non-Rigid Objects by Combining Local and Global Visual Model. Objavljeno v *Computer Vision Winter Workshop*, 2009.
- [57] P. Viola in M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. Objavljeno v *IEEE Conference on Computer Vision and Pattern Recognition*, del 1, stran 511, 2001.
- [58] C. Yang, R. Duraiswami, in L. Davis. Efficient Mean-Shift Tracking via a New Similarity Measure. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, strani 176–183. IEEE Computer Society, 2005.
- [59] M. Yang in Y. Wu. Granularity and elasticity adaptation in visual tracking. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, strani 1–8, 2008.
- [60] Z. Yin in R. Collins. On-the-fly Object Modeling while Tracking. Objavljeno v *IEEE International Conference on Computer Vision and Pattern Recognition*, strani 1–8, 2007.
- [61] X. Zhu, J. Yang, in A. Waibel. Segmenting hands of arbitrary color. Objavljeno v *IEEE International Conference on Automatic Face and Gesture Recognition*, strani 446–453, 2000.

Izjava

Izjavljam, da sem magistrsko nalogo izdelal samostojno pod vodstvom mentorja prof. dr. Aleša Leonardisa. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Ljubljana, 15. junij 2010

Luka Čehovin