Univerza v Ljubljani

Fakulteta za elektrotehniko

Matej Kristan

Sekvenčne Monte Carlo metode za sledenje oseb v računalniškem vidu

Magistrsko delo

Mentor: prof. dr. Stanislav Kovačič

Ljubljana, september 2005

Staršem, Igorju in Tanji, ter sestri Katji.

Zahvala

Rad bi se zahvalil prof. dr. Stanislavu Kovačiču, ki mi je med podiplomskim študijem nudil stalno razpoložljivost in mentorstvo pri izdelavi magistrske naloge. Zahvalil bi se rad staršem in Katji ter Urši, ki so me spodbujali, mi vedno stali ob strani in prenašali moj včasih malce naporen karakter (človek pač postane nekoliko avtističen ob vseh teh formulah...). Posebej bi se zahvalil očetu Igorju za nekatere predloge, ki so se izkazali za zelo uporabne pri mojem strokovnem delu zadnjih dveh let. Urši se zahvaljujem za vse dodane ter odvzete vejice in pike, ki so magistrsko nalogo približale pravilom slovenske slovnice.

Vsi preizkusi v magistrski nalogi so bili izvedeni na video posnetkih, ki so rezultat vztrajnosti in naporov Gorana Vučkoviča ter Marte Bon – obema se lepo zahvaljujem.

Zahvalil bi se še rad dr. Janezu Peršu za uporabne nasvete ter predloge in obilico motivacije med podiplomskim študijem, ter kolegoma Roku Žitku in Mateju Peršetu za številne izmenjave izkušenj in idej. Posebej se zahvaljujem Štefanu Kirnu za vse nasvete, ki so močno izboljšali moje znanje programiranja. Marku se zahvaljujem za rjuhe, ki mi jih je podaril na dan mojega diplomiranja pred dvemi leti in na katerih sem včasih spal med podiplomskim študijem. Zahvalil bi se rad tudi vsem, ki so mi pri študiju kakorkoli pomagali, pa tu niso omenjeni.

Povzetek

Sledenje oseb je le del sicer širokega področja računalniškega vida, ki je bil v zadnjih dvajsetih letih deležen izredno velike pozornosti s strani raziskovalcev. Zanimiv pogled na problem sledenja oseb izhaja iz teorije vodenja in obravnava sledeni objekt kot dinamični sistem, katerega stanje je skrito, na voljo pa so le meritve o trenutnem stanju. Klasični prijem reševanja tega problema je bila v preteklosti uporaba Kalmanovega filtra in njegovih različic. Ti pristopi v glavnem predpostavljajo Gaussov linearen model dinamičnega procesa ter meritvenega postopka, kar pa je pogosto prestroga omejitev pri modeliranju realnih procesov. V poznih devetdesetih so na različnih področjih znanosti raziskave sekvenčnih Monte Carlo metod obrodile orodja za učinkovito reševanje problemov takega časovnega ocenjevanja dinamičnih sistemov. Glavna prednost slednjih pred Kalmanovim filtrom je, da ne vsiljujejo tako strogih predpostavk o opazovanem procesu, njihova implementacija pa je relativno preprosta. V računalniškem vidu so sekvenčne Monte Carlo metode, znane tudi po imenu filtri z delci, postale zelo prilubljene na področju sledenja oseb po objavi algoritma Condensation. Pričujoča magistrska naloga je posvečena problemu sledenja oseb v računalniškem vidu s pomočjo sekvenčnih Monte Carlo metod, aplikacijo katerih smo prikazali na primeru sistema za sledenje večih igralcev v moštvenih igrah. V nalogi smo najprej sledenje umestili v kontekst statističnega ocenjevanja in predstavili glavne dele Monte Carlo metod za reševanje takih problemov. Znani algoritem Condensation, ki predstavlja osrednji del naše aplikacije, smo podali kot posebno različico sekvenčnih Monte Carlo metod in preprosto implementacijo prikazali na algoritmu za sledenje enega igralca v športni igri. Z umestitvijo športne igre v koncept zaprtega sveta smo določili predpostavke, katerim sledijo tipične športne igre pri dani aplikaciji. Na podlagi teh predpostavk smo razvili robustnejši sledilnik za sledenje enega igralca v zaprtem svetu in ga nato razširili v sledilnik za sledenje večih igralcev. Predstavili smo dve različici algoritma za sledenje večih igralcev v zaprtem svetu. Nato smo izvedli vrsto različnih poskusov za vrednotenje predlaganih sledilnikov, rezultate smo komentirali in na koncu izbrali najbolj primeren sledilnik za sledenje večih igralcev. Nakazali smo tudi smernice za nadaljnji razvoj aplikacije za sledenje igralcev z uporabo sekvenčnih Monte Carlo metod.

Ključne besede:
Sekvenčne Monte Carlo metode, filtri z delci, dinamični sistemi, računalniški vid, sledenje večih tarč, zaprti svet, barvni histogrami, moštveni športi

Abstract

People tracking is a part of a broad domain of computer vision, that has received a great attention from researchers over the last twenty years. An interesting aspect of the problem of tracking originates from the field of control theory and considers the object being tracked as a dynamical system with a hidden state, of which only the current measurements are available and observed. The classical methods that were used in the past to tackle this problem employed Kalman filters and their derivatives. These generally assume a Gaussian linear dynamical and measurement model – assumptions, which are usually too restrictive for the majority of natural processes. In the late 90's, the advances in the sequential Monte Carlo methods on various fields of science gave rise to a family of methods that effectively deal with problems of this kind. Their main advantage over the Kalman filter is that they do not impose as restrictive assumptions and can be relatively easily implemented. In computer vision, the sequential Monte Carlo methods, also known as particle filters, became extremely popular with the introduction of the CONDENSATION algorithm. Since then, a body of literature has been published regarding these methods. This thesis is dedicated to the problem of tracking people by means of sequential Monte Carlo methods, application of which is demonstrated on a system for tracking players in team sports. We first consider the problem of tracking in the context of statistical estimation and present the main parts of the Monte Carlo solutions. The well known CONDENSATION algorithm, which comprises the central part of all the trackers presented here, is introduced as a sequential Monte Carlo method and a simple algorithm to track one player is presented. By considering a team sport in the context of a closed world, a set of assumptions that depicts a typical match is derived. Following these assumptions, a more robust single-player tracker is developed and then extended to the case of multiple players. Finally, two variants of trackers for tracking multiple players in the closed worlds are presented. A number of experiments are reported to evaluate the performance of the trackers and based on the results, the most suitable multi-player tracker is chosen. We also point out some guidelines for future development of the application for tracking multiple players.

KEY WORDS:
Sequential Monte Carlo methods, particle filters, dynamical systems, computer vision, multi-target tracking, closed world, color histograms, team sports

Vsebina

Akronimi

pdf *funkcija porazdelitve gostote verjetnosti*

n.v. *naključni vektor*

cdf *funkcija porazdelitve kumulativne gostote verjetnosti*

i.i.d. *neodvisni in identično porazdeljeni*

MC *Monte Carlo*

SIS *sekvenčno prioritetno vzorčenje*

SIR *vzorčenje s prevzorčenjem*

CONDENSATION *conditional density propagation*

ZS *zaprti svet*

MSE *minimalna kvadratična napaka*

AIC *Akaikejev informacijski kriterij*

MLE *postopek največjega verjetja*

MHT *sledenje večih hipotez*

JPDA *hkratna verjetnostna asociacija podatkov*

MAP *maksimalna a posteriori*

NN *najbližji sosed*

ZS *zaprti svet*

n.s. *naključna spremenljivka*

Poglavje 1

Uvod

Sledenje oseb z metodami računalniškega vida je veja sicer širokega področja računalniškega vida, ki je bila v zadnjih dvajsetih letih deležna izredno velike pozornosti s strani raziskovalcev. Možnosti aplikativne uporabe teh metod je zelo veliko; npr. že samo avtomatsko nadzorovanje vključuje številne uporabe v varovanju parkirišč, sob, notranjosti stavb, bankomatov, itd. V vojaški industriji so med bolj zanimivimi možnosti aplikacije nadzorovanja bojnega polja, sledenja letal, zemeljskih vozil, ladij, ter nenazadnje aplikacije vodenja izstrelkov in sledenja tarče. V športu avtomatsko sledenje z računalniškim vidom omogoča npr. objektiven vnos podatkov pri opazovanju gibanja igralca po igrišču ali pa gibanja le dela njegovega telesa; npr. biomehanska analiza zamaha roke pri metu kopja. Možnost uporabe avtomatskega sledenja z računalniškim vidom najdemo celo v študijah družabnega vedenja insektov, kjer avtomatsko sledenje nudi izhodišče za analizo interakcij med družabnimi bitji in omogoča raziskave *naravnih* večagentnih sistemov.

Zaradi zelo široke uporabe metod računalniškega vida je bilo razvitih tudi mnogo algoritmov in pristopov k sledenju. Eden od bolj zanimivih pristopov, ki izhaja iz teorije vodenja, temelji na Bayesovem rekurzivnem ocenjevanju a posteriori porazdelitve stanja tarče. V preteklosti je mnogo različnih avtorjev omenjeni rekurzivni izračun relativno uspešno implementiralo z uporabo tako imenovanega Kalmanovega filtra [1]. Problem Kalmanovega filtra pa je v predpostavki, da sta opazovan proces in meritveni postopek Gaussova in linearna. Ker te predpostavke veljajo le za majhen delež naravnih procesov, so bile slednje pogosto prestroge, rezultat pa je bil slabo sledenje. V ta namen so raziskovalci predlagali mnogo rešitev, ki so nekoliko omilile strogost predpostavk. Pred kratkim, v začetku devetdesetih let, pa so se na različnih področjih raziskav pojavile tako imenovane sekvenčne Monte Carlo metode, znane tudi po imenu filtri z delci in se uveljavile kot učinkovito orodje za aproksimativno reševanje prej omenjene Bayesove rekurzije. Za razliko od predhodnic, ki so temeljile na Kalmanovem filtru, slednje metode ne predpostavljajo linearnosti in Gaussovosti, njihova implementacija pa je relativno preprosta. V računalniškem vidu so

sekvenčne Monte Carlo metode postale priljubljene v poznih devetdesetih po objavi algoritma CONDENSATION [2], kar je sprožilo plaz raziskav in aplikacij na področju sledenja oseb in analize gibanja z uporabo video posnetkov.

1.1 Motivacija

Računalniško podprta analiza gibanja na podlagi video posnetkov postaja zadnje čase vedno bolj privlačno orodje za analizo športnih iger. Motivacija za uporabo take analize je potreba po objektivni obdelavi ter analizi zelo velikih količin podatkov, razlog za povečan interes pa je predvsem pocenitev računalniške opreme za zajem in obdelavo videa ter povečanje zmogljivosti osebnih računalnikov – torej, finančna dostopnost zmogljivih sistemov širšemu krogu uporabnikov.

Zaradi splošne priljubljenosti nekaterih športov so uspešni klubi, kot tudi posamezni športniki, deležni precejšne slave in s tem tudi financiranj s strani sponzorjev. Pritok financ v klube ter tekmovalna narava športnikov spodbujata težnjo po boljših rezultatih in iskanje alternativnih metod za izboljšanje igre. Najbolj očitne metode so na primer:

- **Načrtovanje treninga:** *Koliko kondicije in moči potrebuje športnik za dobro igranje? Kakšne vrste mišičnega tkiva in v koliki meri ga je potrebno razviti (naj si bo to hitre mišice za eksplozivne odzive ali vzdržljive, močne, vendar počasne mišice)?*
- **Taktična analiza:** *Kateri del igralnega polja mora določeni igralec v moštvenem športu pokrivati za najboljše rezultate? Kakašne so koreografije napadov in obramb nasprotnika? Kakšne so optimalne koreografije za dosego zmage? Katere so najpogostejše/najusodnejše taktične napake igralca/moštva med tekmo?*

Prva točka se nanaša na statistike pretečenih razdalj, dinamike hitrosti in pospeška igralcev med tekmo – lastnosti, ki se neposredno tičejo v času izmerjenega položaja igralca na igrišču. Druga točka je v smislu avtomatske analize precej bolj zahtevna, saj se nanaša na semantiko igre, iskanje vzorcev v sekvenci gibanja igralca/moštev in vključuje veliko ekspertnega znanja.

Trenutno je uporaba računalniško podprte analize gibanja igralcev osredotočena na moštvene športe, predvsem na nogomet. Razlog je v velikih cenah storitev podjetji na tem področju. Kot primer navedimo le sistem **ProZone** [3], katerega cena namestitve znaša približno £100.000, vsako naknadno analizo pa podjetje dodatno zaračuna [4]. Tako velike zneske si lahko privoščijo le premožnejši klubi oziroma njihovi financerji. Razlogi za veliko ceno storitev ponudnikov na področju analize gibanja so različni. Nekateri sistemi npr.

temeljijo na sledenju z oddajniki, katere pričvrstijo na igralce, taka tehnologija pa je seveda zelo draga. Spet drugi sistemi temeljijo na sledenju s pomočjo video posnetkov tekem, vendar vključujejo ročno označevanje igralcev, kar je zopet lahko drago. Verjetno najbolj očiten razlog za visoke cene je majhno število ponudnikov, vsak pa svoje storitve v glavnem prodaja le parim velikim športnim klubom. Ti klubi so tako močno vezani na ponudnike, da nekateri celo znotraj kluba zaposljujejo osebe za komunikacijo s njimi.

Podrobnejšo analizo tržne smotrnosti sistemov za analizo gibanja igralcev najdemo v [4], v kateri so avtorji podali pregled trenutnih ponudikov omenjenih storitev in ugotovili, da tržno nišo predstavljajo cenejši sistemi za analizo gibanja, katere bi lahko po nižjih cenah prodajali večjemu številu manjših klubov.

Ob koncu leta 2003 je k Laboratoriju za slikovne tehnologije na Fakulteti za elektrotehniko Univerze v Ljubljani pristopilo podjetje IspireWorks, ki je naročilo razvoj sistema za analizo moštvenih iger na podlagi video posnetkov. Ker je osnova za vsakršno analizo gibanja zmožnost pridobivanja položajev igralcev, je prvi korak razvoj sistema za sledenje igralcev v športni igri. Temu koraku je posvečena pričujoča magistrska naloga.

1.2 Zgradba magistrske naloge

V magistrski nalogi bomo predstavili sistem za sledenje igralcev v športnih igrah, ki temelji na sekvenčnih Monte Carlo metodah za sledenje oseb. V poglavju 2 bomo sledenje predstavili v kontekstu stohastičnega ocenjevanja in razložili matematično ozadje sekvenčnih Monte Carlo metod. Ob koncu tega poglavja bomo v metode umestili tudi zelo znan algoritem CONDENSATION, na katerem tudi temeljijo vsi v tej nalogi predstavljeni sledilniki. V nadaljevanju bomo problematiko sledenja v športni igri obravnavali kot aplikacijo v delno kontroliranem okolju, aplikacijo za analizo procesa, ki se v zelo širokem smislu podreja nekaterim zakonitostim. Športno igro bomo zato v poglavju 3 opisali s tako imenovanimi predpostavkami zaprtega sveta. V poglavju 4 bomo izbrali ustrezne vizualne značilnice za predstavitev igralca v slikah in z upoštevanjem napreprostejših predpostavk zaprtega sveta zgradili sledilnik za sledenje enega igralca. Slednjega bomo v poglavju 5 z upoštevanjem ostalih predpostavk nadgradili in predstavili kompleksnejši sledilnik za sledenje enega igralca v zaprtem svetu. Problem sledenja večih igralcev bomo v poglavju 6 umestili v kontekst sledenja večih tarč in sledilnik za enega igralca iz poglavje 5 razširili na poljubno število igralcev. Predlagane sledilnike bomo primerjali na podlagi rezultatov poskusov v poglavju 7 in nazadnje magistrsko nalogo zaključili s poglavjem 8.

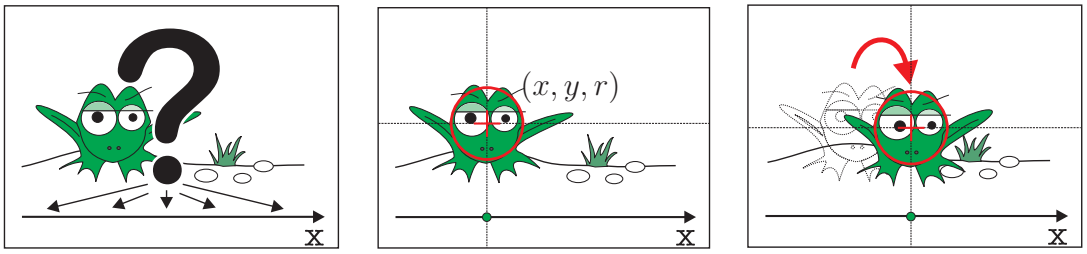
Poglavje 2

Sledenje kot problem stohastičnega ocenjevanja

Kadar govorimo o "sledenju" se moramo zavedati, da je ta beseda zelo splošna in je predvsem odvisna od pojma *zanimive informacije*. Le-ta je odvisna od tarče in lahko zajema različne lastnosti: v športu nas lahko zanima položaj igralca na igrišču ali npr. naklon kopja pri metu; v letalstvu lahko predstavlja zanimiva informacija smer, iz katere prihaja letalo; v ekonomiji je to lahko trenutna vrednost določenega blaga, itd. Sledenje si lahko torej predstavljamo kot postopek pridobivanje/vzdrževanja zanimive informacije o tarči, le-ta pa je pogosto le majhna podmnožica vse informacije, ki jo tarča zajema. S tega vidika je smiselno predstaviti tarčo z najmanjšo podmnožico vse informacije, ki še zajema zanimivo informacijo in hkrati zagotavlja primerno sledenje; to pa pomeni aproksimacijo. Tarčo tako aproksimiramo, zapišemo, z matematičnim modelom, katerega dimenzionalnost ter parametri so odvisni od prej omenjene podmnožice informacije. To idejo ponazarja primer 1.

Primer 1. *Zanima nas položaj žabice na sliki 2.1a vzdolž horizontalne osi. Problem sledenja se prevede na modeliranje žabice z npr. krogom ter določanje centra kroga vzdolž te osi v vsakem časovnem koraku. Parametri* (x, y, r) *kroga predstavljajo zadostno informacijo, s katero lahko napnemo krog na žabico ter omogoča uspešno sledenje, medtem ko le koordinata centra kroga* x *predstavlja zanimivo informacijo.*

Dejansko stanje tarče (npr. parametri modela žabice v sliki 2.1 je v resnici neznano, skrito, in ga je potrebno določiti preko procesa zaznavanja s takšnimi ali drugačnimi senzorji (npr. kamera, mikrofon, radar, meter,...). Ti instrumenti skupaj z modelom, ki je aproksimacija, navadno vnašajo določeno stopnjo negotovosti v meritve; zato v nekem časovnem koraku stanje objekta glede na meritve lahko zavzame ne enega, pač pa množico bolj ali manj verjetnih konfiguracij. To pomeni, da stanje tarče v nekem trenutku ne moremo predstaviti le z eno samo realizacijo parametrov modela, pač pa s porazdelitvijo,



(a) (b) (c)

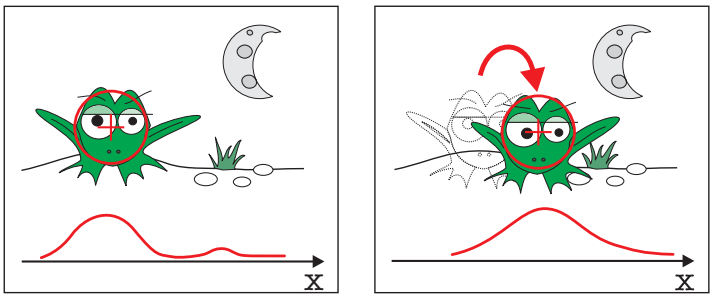
Slika 2.1: *Problem določanja položaja žabice vzdolž horizontalne osi (a). Žabico modeliramo s krogom, torej tremi parametri (x, y, r) (b) in (c). Trije parametri predstavljajo informacijo potrebno za sledenje, medtem ko le parameter x predstavlja zanimivo informacijo, to pa je položaj vzdolž horizontalne osi.*

ki upoštevajoč meritve vsakemu možnemu stanju pripiše neko gostoto verjetnosti. Skrito stanje je moč potem oceniti z npr. operatorjem matematičnega upanja na tej porazdelitvi ali pa kakšnim drugim operatorjem. To slednje je prikazano s primerom 2.

Primer 2. *Zanima nas položaj žabice v slikah 2.2(a,b). Žabico zopet modeliramo s krogom. Sedaj upoštevamo negotovost meritev ter dejstvo, da krog ni enolična ponazoritev žabice, in dobimo ne le ene, pač pa množico možnih položajev vzdolž horizontalne osi, med katerimi so nekateri bolj, drugi pa manj možni. Dobimo torej tako imenovano a posteriori funkcijo gostote verjetnosti stanj žabice glede na meritve, ki predstavlja porazdelitev verjetnosti izmerjenega položaja žabice vzdolž horizontalne osi. Z operatorjem matematičnega upanja na tej porazdelitvi lahko dobimo pričakovano stanje žabice, ki je v slikah 2.2(a,b) prikazano z rdečim krogom.*

Iz zgornjih primerov torej lahko potegnemo sklep: zanimivo informacijo o tarči v nekem časovnem trenutku ne zajema le ena sama realizacija stanja tarče, pač pa njena *funkcija porazdelitve gostote verjetnosti* (pdf) *(angl. probability density function)*. Vzdrževanje te porazdelitve skozi čas imenujemo sledenje.

V sledečih podpoglavjih bomo predstavili princip v času rekurzivnega izračunavanja zgoraj omenjenih funkcij gostot verjetnosti stanj, ki je povzet po [5, 6, 7]. Najprej bomo predstavili splošen model zapisa stanja tarče, sledil bo matematični zapis problema rekurzivnega ocenjevanja pripadajočih porazdelitev, potem pa bomo predstavili razred aproksimacijskih metod, ki so se v zadnjem času izkazale kot izredno uporabno orodje reševanja problemov sledenja. Med temi metodami bomo nazadnje izbrali algoritem, ki bo služil za osnovo vseh v tem magistrskem delu opisanih sledilnikov.



(a) (b)

Slika 2.2: *Meritveni postopek skupaj z modeliranjem vnaša negotovost v ocenjevanje stanja žabice, zato je informacija o skritem stanju podana s funkcijo gostote verjetnosti stanj. Te gostote so skicirane pod žabicami v slikah (a) in (b).*

2.1 Zapis v prostoru stanj

Če želimo tarčo slediti, jo moramo modelirati, oziroma zapisati s parametričnim modelom. Ko temu modelu določimo vrednosti parametrov, rečemo, da je tarča zavzela neko stanje, prostor vseh konfiguracij parametrov modela pa imenujemo prostor stanj. V tem poglavju bomo predstavili določen zapis stanj tarče in nekatere predpostavke, na katere se bomo sklicevali vseskozi magistrsko nalogo.

Naj bo $\mathcal{X}$ prostor stanj v katerem je definiran model tarče. Dejanskega stanja tarče ne poznamo (je prikrito) in ga ob času $t \in \mathbb{N}$ opisuje *naključni vektor* (n.v.) $\mathbf{x}_t \in \mathcal{X}$. Neopažen signal (prikrita stanja) modeliramo z *Markovovim procesom prvega reda* z začetno porazdelitvijo $p(\mathbf{x}_0)$ ter porazdelitvijo prehajanja stanj $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Informacije o stanju tarče pridobivamo z merjenjem in ob času t izvedeno meritev opisuje n.v. $\mathbf{y}_t \in \mathcal{Y}$, ker je $\mathcal{Y}$ prostor meritev. Sekvenci stanj in meritev do trenuka t označimo z

$$\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, ..., \mathbf{x}_t\} \text{ in } \mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, ..., \mathbf{y}_t\}.$$

Če upoštevamo Markovovo lastnost in predpostavimo pogojno neodvisnost trenutne meritve od vseh preteklih stanj ter meritev [1], lahko pišemo:

$$\mathbf{x}_t \perp (\mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-2})|\mathbf{x}_{t-1} \quad \Rightarrow \quad p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{2.1}$$

$$\mathbf{y}_t \perp (\mathbf{y}_{1:t-1}, \mathbf{x}_{0:t-1})|\mathbf{x}_t \quad \Rightarrow \quad p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{x}_{0:t}) = p(\mathbf{y}_t|\mathbf{x}_t), \tag{2.2}$$

kjer smo z $\perp$ označili neodvisnost.

[1] Pogojno neodvisnost si lahko razlagamo takole: če poznamo trenutno stanje tarče, potem lahko pri tem stanju izvedemo meritev. Trenutna meritev je res načeloma odvisna od sekvence preteklih stanj ter preteklih meritev, vendar ker poznamo trenutno stanje, je meritev odvisna le od slednjega in tako neodvisna od vseh ostalih meritev ter stanj. Pravimo, da je trenutna meritev *pogojno* neodvisna od sekvenc preteklih meritev, ter stanj pri *danem* trenutnem stanju.

Izraz (2.1) pomeni, da je trenutno stanje odvisno le od njegovega direktnega predhodnika, medtem ko je sklicujoč na izraz (2.2), trenutna meritev neodvisna od vseh preteklih meritev ter stanj pri danem trenutnem stanju.

2.2 Bayesovo rekurzivno filtriranje

Ker sledimo objekt na podlagi opravljanja meritev, je ob trenutku t, po opravljenih meritvah, vsa zanimiva informacija o tarči vsebovana v njeni a posteriorni porazdelitvi stanj $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$.

Ob časovnem koraku t, torej poznamo a posteriori pdf $p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$ iz prejšnjega časovnega koraka, in ko postanejo trenutne meritve $\mathbf{y}_t$ na voljo, želimo izračunati trenutno a posteriori pdf $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. Tak rekurzivni izračun je možno zapisati z Bayesovim rekurzivnim filtrom.

Vsako a posteriori pdf $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ lahko po Bayesovem pravilu zapišemo kot

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{x}_{0:t}, \mathbf{y}_{1:t})}{p(\mathbf{y}_{1:t})}. \tag{2.3}$$

Števec v zgornjem izrazu lahko dalje faktoriziramo po pravilu veriženja v

$$p(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{x}_{0:t})p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1}), \tag{2.4}$$

imenovalec pa v

$$p(\mathbf{y}_{1:t}) = p(\mathbf{y}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1}). \tag{2.5}$$

Člen $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t-1})$ v enačbi (2.4) lahko dalje faktoriziramo in z uporabo predpostavke o pogojni neodvisnosti stanj (enačba 2.1) dobimo

$$\begin{aligned} p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t-1}) &= p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) \\ &= p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}). \end{aligned} \tag{2.6}$$

Z upoštevanjem enačb (2.4-2.6) se izraz za aposteriori pdf (2.3) spremeni v

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{x}_{0:t})p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})p(\mathbf{y}_{1:t-1})}. \tag{2.7}$$

Če še v imenovalcu in števcu pokrajšamo člen $p(\mathbf{y}_{1:t-1})$, ter upoštevamo pogojno neodvisnost meritev (enačba 2.2), končno dobimo

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}. \tag{2.8}$$

Zgornji izraz predstavlja rekurzivni izračun a postetriori porazdelitve $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, saj na desni strani nastopa $p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$, ki je ravno a posteriori porazdelitev stanj v prejšnjem časovnem koraku $(t-1)$.

V kasnejših poglavjih nas bo predvsem zanimala marginalna pdf $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, ki ji pravimo *filtrirajoča porazdelitev (angl. filtering distribution)* in predstavlja a posteriori pdf *trenutnega stanja* objekta. To porazdelitev dobimo preprosto z integriranjem $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ preko sekvence preteklih stanj, katero označimo z $\mathbf{x}_{0:t-1}$:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})d\mathbf{x}_{0:t-1}. \tag{2.9}$$

Z upoštevanjem relacije (2.8) dobimo

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{0:t-1}, \tag{2.10}$$

kar lahko zapišemo v obliki

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{t-1})(\int p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{0:t-2})d\mathbf{x}_{t-1}$$

in ker velja marginalizacija

$$\int p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{0:t-2} \triangleq p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}),$$

se izraz iz enačbe (2.10) spremeni v

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}. \tag{2.11}$$

Tako smo prišli do pomembne oblike za rekurzivni izračun a posteriori pdf trenutnega stanja tarče.

Rekurziji v enačbah (2.8,2.11) sta bolj akademske narave, saj v splošnem analitična rešitve ne obstajajo. Za zelo omejeno množico primerov, ko predpostavimo linearen dinamični in meritveni model, ter normalnost vseh nastopajočih porazdelitev, pa rešitev rekurzije v enačbi (2.11) obstaja in se imenuje *Kalmanov filter* [1]. Na žalost so te omejitve primerne le za manjšino vseh praktičnih problemov, kajti v splošnem je potrebno rekurzije aproksimirati. Od sredine šestdesetih let dalje je bilo tako mnogo raziskovalnega dela posvečenega aproksimiranju teh porazdelitev in rekurzij, kar je med drugim rodilo številne rešitve kot so: *razširjeni Kalmanov filter (angl. extended Kalman filter)* (glej [8]), *filter vsote Gausov (angl. Gaussian sum filter)* [9] ter *mrežne metode (angl. grid based methods)* [10].

Šele v poznih osemdesetih je z velikim povečanjem računske moči računalnikov prišlo do hitrega napredovanja v uporabi numerične integracije v Bayesovem filtriranju [11]. Rezultat raziskav so bile Monte Carlo metode, katere bomo opisali v naslednjem podpoglavju. Glavna prednost slednjih je v tem, da ne predpostavljajo nikakršne linearnosti ali normalnosti nastopajočih porazdelitev in so relativno preprosto izračunljive.

2.3 Monte Carlo pristop k filtriranju

Na podlagi funkcije a posteriori porazdelitve stanj lahko ocenimo trenutno stanje, oziroma sekvenco stanj, preprosto z operatorjem matematičnega upanja na tej porazdelitvi. Taka ocena namreč minimizira srednjo kvadratično napako na porazdelitvi. V splošnem nas sicer zanimajo integrali tipa

$$I(f_t) = \langle f_t(\mathbf{x}_{0:t}) \rangle_{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \triangleq \int f_t(\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \tag{2.12}$$

integrabilnih funkcij $f_t : \mathcal{X}^{t+1} \to \mathbb{R}^{n_{f_t}}$.

Spomnimo, da je tudi marginalizacija iz enačbe (2.11) pravzaprav integral na a posteriori porazdelitvi stanj. To nas navaja na dejstvo, da lahko probleme širjenja iskanih porazdelitev skozi čas rešujemo z *Monte Carlo* (MC) metodami.

2.3.1 Popolno Monte Carlo vzorčenje

Predpostavimo, da lahko iz porazdelitve $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ vzorčimo N *neodvisnih in identično porazdeljenih* i.i.d. *(angl. independent and identically distributed)* vzorcev ali tako imenovanih *delcev (angl. particles)*, ki jih zapišemo z množico $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^{N}$. Empirično oceno te porazdelitve predstavljene z delci zapišemo kot

$$P_N(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\mathbf{x}_{0:t}^{(i)}}(\mathbf{x}_{0:t}), \tag{2.13}$$

kjer je $\delta_{\mathbf{x}_{0:t}^{(i)}}(\cdot)$ Dirakov delta usrediščen na delcu $\mathbf{x}_{0:t}^{(i)}$. Taki empirični oceni včasih tudi rečemo *naključna mera (angl. random measure)*. Ocena integrala $I(f_t)$ iz enačbe (2.12) se tako z uporabo naključne mere glasi:

$$I_N(f_t) = \int f_t(\mathbf{x}_{0:t}) P_N(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} = \frac{1}{N} \sum_{i=1}^{N} f_t(\mathbf{x}_{0:t}^{(i)}). \tag{2.14}$$

Če je a posteriori varianca $f_t(\mathbf{x}_{0:t})$ omejena, potem zaradi močnega zakona velikih števil velja

$$I_N(f_t) \xrightarrow[N\to\infty]{s.s.} I(f_t) \tag{2.15}$$

kjer s.s. pomeni skoraj sigurno ali z verjetnostjo ena [5]. To pomeni, da lahko s povečevanjem števila delcev N integral $I(f_t)$ poljubno dobro aproksimiramo z $I_N(f_t)$.

Žal ponavadi ne moremo učinkovito vzorčiti iz $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, saj je le-ta v splošnem multivariatna, nestandardna in znana le do proporcionalnega faktorja natančno [7]. Klasični pristop k reševanju tega problema je uporaba prioritetnega vzorčenja.

2.3.2 Uporaba prioritetnega vzorčenja

Prioritetno vzorčenje (angl. importance sampling) je klasična operacija pri MC integriranju, kadar je potrebno oceniti integral preko neke porazdelitve, ki jo v splošnem težko vzorčimo in jo lahko točkovno računamo vsaj do proporcionalne konstante natančno. Imenujmo to porazdelitev *primarna porazdelitev*. Glavna ideja je vpeljati novo porazdelitev, ki jo lažje vzorčimo in jo lahko točkovno izračunamo vsaj do proporcionalne konstante natančno. Slednjo porazdelitev imenujemo *prioritetna funkcija* ali *prioritetna porazdelitev*, *(angl. importance function; proposal distribution)* in zahtevamo, da njena podpora *(angl. support)* vsebuje podporo primarne porazdelitve. To preprosto pomeni, da je prioritetna funkcija različna od nič povsod kjer je različna od nič tudi primarna funkcija.

Naj bo $\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ prioritetna funkcija, ki je v splošnem lahko odvisna od trenutnega stanja in meritve $\mathbf{x}_t$, $\mathbf{y}_t$ ter zgodovine $\mathbf{x}_{0:t-1}$,$\mathbf{y}_{1:t-1}$, in naj vsebuje podporo primarne porazdelitve $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. Prioritetno funkcijo $\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ vzorčimo in naj množica $\{\mathbf{x}_{0:t}^{(i)}\}_{i=1}^{N}$ predstavlja N i.i.d. tako dobljenih delcev. Potem se MC aproksimacija integrala $I(f_t)$ (enačba 2.12) lahko glasi

$$\hat{I}_N(f_t) = \sum_{i=1}^{N} f_t(\mathbf{x}_{0:t}^{(i)})\tilde{w}_t^{(i)}, \tag{2.16}$$

Kjer so $\tilde{w}_t^{(i)}$ *normirane prioritetne uteži (angl. normalized importance weights)* definirane kot

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}, \tag{2.17}$$

in so $w_t^{(i)}$ *prioritetne uteži (angl. importance weights)* definirane z izrazom

$$w_t^{(i)} \propto \frac{p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}{\pi(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}. \tag{2.18}$$

Zaradi preglednosti smo nekoliko bolj natančen opis prioritetnega vzorčenja umaknili v Dodatek A.

Ker je prioritetno vzorčenje v splošnem MC metoda integriranja za aproksimacijo integrala (2.16), še vedno velja $\hat{I}_N(f_t) \xrightarrow[N\to\infty]{s.s.} I(f_t)$ [5].

Na izraz (2.16) lahko gledamo kot integracijo preko neke porazdelitve

$$\hat{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\mathbf{x}_{0:t}^{(i)}}(\mathbf{x}_{0:t}), \tag{2.19}$$

ki je aproksimacija porazdelitvi $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. Ocena $\hat{I}_N(f_t)$ je potem integral funkcije f_t preko empirične mere $\hat{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$:

$$\hat{I}_N(f_t) = \int f_t(\mathbf{x}_{0:t})\hat{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})d\mathbf{x}_{0:t}. \tag{2.20}$$

Ker velja $\hat{I}_N(f_t) \xrightarrow[N\to\infty]{s.s.} I(f_t)$, lahko rečemo, da porazdelitev $\hat{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ konvergira proti pravi $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ v smislu integralov. To pa pomeni, da lahko metodo prioritetnega vzorčenja uporabimo za generiranje delcev, s katerimi aproksimiramo porazdelitev $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, saj potrebujemo le tako reprezentacijo porazdelitve, ki omogoča dovolj dobro oceno integralov.

Pričujoč zapis prioritetnega vzorčenja še vedno ni primeren za uporabo v sekvenčnem filtriranju, saj postane s časom računsko potraten [5]. Rešitev je v zapisu prioritetne funkcije v rekurzivni obliki.

2.4 Sekvenčno prioritetno vzorčenje

Problem računske kompleksnosti zgoraj omenjenega določanja porazdelitve lahko rešimo tako, da na primeren način faktoriziramo prioritetno funkcijo:

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})\pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}). \tag{2.21}$$

Upoštevajoč, da je pri danem $\mathbf{x}_{0:t}$ imenovalec v izrazu (2.8) konstanten, lahko pišemo

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}). \tag{2.22}$$

Oba zgornja izraza vstavimo v enačbo (2.18) in dobimo

$$w_t^{(i)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})} \frac{p(\mathbf{x}_{0:t-1}^{(i)}|\mathbf{y}_{1:t-1})}{\pi(\mathbf{x}_{0:t-1}^{(i)}|\mathbf{y}_{1:t-1})}. \tag{2.23}$$

Drugi ulomek v zgornji enačbi je ravno proporcionalen uteži v prejšnjem časovnem koraku (enačba 2.18). Upoštevajoč to, končno dobimo zapis uteži prioritetnega vzorčenja v rekurzivni obliki:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}. \tag{2.24}$$

Če sedaj izberemo prioritetno funkcijo v obliki

$$\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t}) = \pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t), \tag{2.25}$$

potem ta postane odvisna le od preteklega stanja $\mathbf{x}_{t-1}$ in trenutne meritve $\mathbf{y}_t$. Tak zapis je še posebej primeren, kadar nas ob vsakem časovnem koraku zanima le ocena marginalne porazdelitve $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. V takih primerih je potrebno shraniti le trenutne vrednosti vzorčenih stanj $\mathbf{x}_t^{(i)}$ in lahko zavržemo sekvence $\mathbf{x}_{0:t-1}^{(i)}$ ter zgodovino meritev $\mathbf{y}_{1:t-1}$ [6]. Izračun uteži potem lahko zapišemo kot

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}, \tag{2.26}$$

marginalno a posteriori porazdelitev $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, pa z novo naključno mero

$$\hat{P}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t). \tag{2.27}$$

MC ocenjevanje a posteriori porazdelitve stanj, ki bazira na takem rekurzivnem prioritetnem vzorčenju, se imenuje *sekvenčno prioritetno vzorčenje* (SIS) *(angl. Sequential Importance Sampling)* in je opisano z algoritmom 2.1.

Vhod:

- A posteriori porazdelitev stanj iz prejšnjega časovnega koraka $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \approx \{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$

Izhod:

- Nova a posteriori porazdelitev stanj $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$

---

1. Za $i = 1 : N$,
   - Vzorči nov delec: $\mathbf{x}_t^{(i)} \sim \pi(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$.
   - Določi utež: $\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}$.
2. Za $i = 1 : N$,
   - Normaliziraj uteži delcev: $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}$.
3. Dobljena mera predstavlja empirični približek pravi porazdelitvi $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N} \approx p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

Algoritem 2.1: *SIS fiter za ocenjevanje* $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$.

Degeneracija sekvenčnega prioritetnega vzorčenja

Problem SIS filtra je v tem, da v praktičnih primerih že po nekaj iteracijah vse uteži razen ene postanejo zelo majhne, kar se odraža v slabi oceni iskane a posteriori porazdelitve. To je v literaturi znano kot *problem degeneracije (angl. problem of degeneracy)* delcev. V glavnem avtorji [7, 6, 12] navajajo dva principa reševanja degeneracije:

1. Primerna izbira prioritetne funkcije

2. Uporaba *prevzorčenja (angl. resampling)*

Izbira prioritetne funkcije

Idealna izbira prioritetne funkcije bi bila jasno kar iskana a posteriori $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, vendar pa te ne poznamo. Naravna izbira prioritetne funkcije zato temelji na funkciji, ki minimizira varianco prioritetnih uteži. Tak način izbire prioritetne funkcije povzroči dodatne probleme, ki pa ne spadajo v okvir tega magiterija; podrobnejšo razpravo o tem najdemo v [7].

Pomemben razred filtrov nastopi, če za prioritetno funkcijo izberemo kar a priori porazdelitev prikritega Markovovega modela sistema

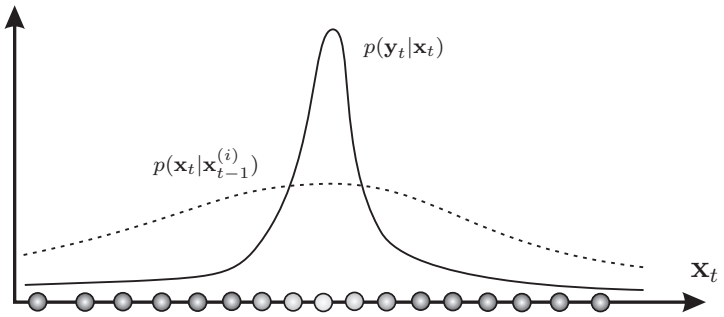
$$\pi(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{2.28}$$

in izračun prioritetnih uteži (enačba 2.26) se tako poenostavi v

$$w_t^{(i)} \propto w_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}). \tag{2.29}$$

Ta pristop je precej popularen v praktični rabi, saj je vzorčenje iz $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ ponavadi enostavno, prav tako pa je zelo lahek tudi izračun uteži.

Zavedati se je treba, da lahko taka izbira prioritetne funkcije povzroči slabo obnašanje filtra. Kadar je funkcija verjetja $p(\mathbf{y}_t|\mathbf{x}_t)$ precej bolj usločena kot $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, je rezultat vzorčenja iz $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ mnogo delcev, ki se nahajajo na področju kjer je vrednost $p(\mathbf{y}_t|\mathbf{x}_t)$ zelo majhna (slika 2.3). Torej generiramo le malo delcev na območju z večino gostote verjetnosti in večino delcev povsod drugje. Kot rezultat bo zelo veliko delcev imelo nizke uteži; prav slednji situaciji pa se želimo ogniti. Rečeno drugače, v primerih, ko je varianca meritev precej manjša od variance sistemskega šuma, je taka izbira prioritetne funkcije slaba.



Slika 2.3: *Degeneracija delcev zaradi vpliva a priori funkcije $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. Vzorčenje iz $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ generira večino vzorcev v predelu, kjer ima $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ nizke vrednosti; kot rezultat bo imela večina delcev zelo nizko utež.*

Uporaba prevzorčenja

Drugi princip ublažitve degeneriranja prioritetnih uteži je uporaba prevzorčenja. Spomnimo, da empirična porazdelitev

$$\hat{P}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t)$$

predstavlja aproksimacijo pravi porazdelitvi $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. Prevzorčenje je generiranje novih N delcev iz porazdelitve $\hat{P}(\mathbf{x}_t|\mathbf{y}_{1:t})$ na tak način, da omilimo degeneracijo, dobljeni nabor delcev pa še vedno aproksimira $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. Postopek prevzorčenja opisuje preslikava

$$\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N} \rightarrow \{\tilde{\mathbf{x}}_t^{(i)}, \frac{1}{N}\}_{i=1}^{N}, \tag{2.30}$$

$$P_r(\tilde{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(i)}) = w_t^{(i)}, \tag{2.31}$$

kar pomeni, da generiramo novo naključno mero tako, da izmed obstoječih N vzorcev $\{\mathbf{x}_t^{(i)}\}_{i=1}^{N}$ izberemo novih N vzorcev $\{\tilde{\mathbf{x}}_t^{(i)}\}_{i=1}^{N}$, kjer vsak vzorec $\tilde{\mathbf{x}}_t^{(i)}$ izberemo z verjetnostjo $w_t^{(i)}$.

Ker prevzorčenje pomeni vzorčenje iz empirične porazdelitve $\hat{P}(\mathbf{x}_t|\mathbf{y}_{1:t})$, imajo vsi dobljeni delci uteži enake $\frac{1}{N}$ (enačba 2.30) in so približno porazdeljeni po $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

Operacija prevzorčenja prepreči degeneracijo SIS algoritma s tem, da generira novo naključno mero,

$$\hat{P}_N(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\tilde{\mathbf{x}}_t^{(i)}}(\mathbf{x}_t), \tag{2.32}$$

ki je ekvivalent $\hat{P}(\mathbf{x}_t|\mathbf{y}_{1:t})$ tako, da razmnoži tiste delce, ki imajo večje uteži in zavrže tiste z nizkimi. Na tak način skuša nekako *zgostiti* delce v področja $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ z večjo gostoto verjetnosti.

Prevzorčenje potrebujemo predvsem takrat, ko postane množica delcev degenerirana. Za merjenje degeneracije se navadno uporablja ocena *efektivne velikosti vzorca (angl. effective sample size)* [7]

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (w_t^{(i)})^2}, \tag{2.33}$$

ki jo v [13] imenujejo tudi *diagnostika preživetja (angl. survival diagnostic)*. Ideja je torej preprosta: ko $\hat{N}_{eff}$ pade pod določeno vrednost N_{thres}, se izvede operacija prevzorčenja.

Obstaja mnogo učinkovitih metod prevzorčenja kot so npr. slojno vzorčenje *(angl. stratified sampling)* in residualno vzorčenje *(angl. residual sampling)*

[14]; *sistematično* vzorčenje *(angl. systematic resampling)* [15]; *deterministično* vzorčenje, *(angl. deterministic resampling)* [13], itd. V naših implementacijah, ki bodo predstavljene kasneje v magistrskem delu, bomo uporabljali deterministično vzorčenje, saj je implementacija preprosta, kompleksnost algoritma pa je O(N). Implementacijo determinističnega prevzorčenja prikazuje algoritem 2.2.

Vhod:

- A posteriori $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$

Izhod:

- Prevzorčena a posteriori $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \{\tilde{\mathbf{x}}_t^{(i)}, \frac{1}{N}\}_{i=1}^N$

---

1. Generiraj kumulativno funkcijo delcev po enačbi:
   - $c^{(i)} = \sum_{j=1}^{i} w_t^{(j)}$.
2. Inicializiraj: $j = 1$
3. Za $i = 1 : N$,
   - Dokler $(\frac{i}{N} > c^{(j)}) : j++$.
   - Izberi $\tilde{\mathbf{x}}_t^{(i)} = \mathbf{x}_t^{(j)}$ in postavi $w_t^{(i)} = \frac{1}{N}$.

Algoritem 2.2: *Deterministično prevzorčenje.*

Čeprav operacija prevzorčenja zmanjšuje problem degeneracije, pa vnaša nove probleme. Prvič, omejuje možnost paralelizacije filtra in drugič, po prevzorčenju so delci z visokimi utežmi statistično mnogokrat izbrani. Slednje lahko pripelje do zmanjšanja diverzifikacije med vzorci, saj nova množica vzorcev vsebuje mnogo repliciranih delcev. Ta pojav, ki se v literaturi imenuje *osiromašenje vzorcev (angl. sample impoverishment)*, pride močno do izraza v primerih, ko je procesni šum majhen [6] in se po nekaj iteracijah celotni nabor delcev sesuje v en sam delec. Kljub tem težavam, je potreba po odpravi degeneracije filtra tako pomembna, da je operacija prevzorčenja postala del praktično vsake implementacije SIS filtra.

2.4.1 Generični filter z delci

Če za opazovanje degeneracije množice delcev uporabimo efektivno velikost vzorca $\hat{N}_{eff}$ (enačba 2.33), ter za rekurzivno ocenjevanje a posteriori porazdelitve

$p(\mathbf{x}_t|\mathbf{y}_{1:t})$ SIS filter, dobimo zelo uporaben filter z delci, ki se imenuje *generični filter z delci (angl. generic particle filter)*, in je podan z algoritmom v sliki 2.3.

Vhod:

- A posteriori porazdelitev stanj iz prejšnjega časovnega koraka $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \approx \{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$

Izhod:

- Nova a posteriori porazdelitev stanj $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$

---

1. Izračunaj $\hat{N}_{eff}$ po enačbi (2.33).
2. Če $\hat{N}_{eff} > \hat{N}_{thres}$
   Prevzorči $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$ z algoritmom 2.2.
   - $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N} \rightarrow \{\tilde{\mathbf{x}}_{t-1}^{(i)}, \frac{1}{N}\}_{i=1}^{N}$
3. Za $i = 1 : N$,
   - Vzorči nov delec: $\mathbf{x}_t^{(i)} \sim \pi(\mathbf{x}_t|\tilde{\mathbf{x}}_{t-1}^{(i)}, \mathbf{y}_t)$
   - Določi utež: $\tilde{w}_t^{(i)} = \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}$
4. Za $i = 1 : N$,
   - Normiraj uteži delcev: $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}$
5. Dobljena mera predstavlja empirični približek pravi porazdelitvi: $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N} \approx p(\mathbf{x}_t|\mathbf{y}_{1:t})$

Algoritem 2.3: *SIS fiter za ocenjevanje* $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$.

V tretjem koraku algoritma 2.3 vidimo, da pri izračunu novih uteži $w_t^{(i)}$ ne nastopajo več stare $w_{t-1}^{(i)}$. To je zaradi prevzorčenja, saj imajo vsi delci po prevzorčenju enake uteži.

2.5 Algoritem Condensation

Posebna oblika filtra z delci se je v zadnjem desetletju v računalniškem vidu še posebej uveljavila kot popularna metoda sledenja. Gre za specifično obliko generičnega filtra z delci, pri kateri upoštevamo:

1. Za prioritetno funkcijo vzamemo dinamični model sistema:

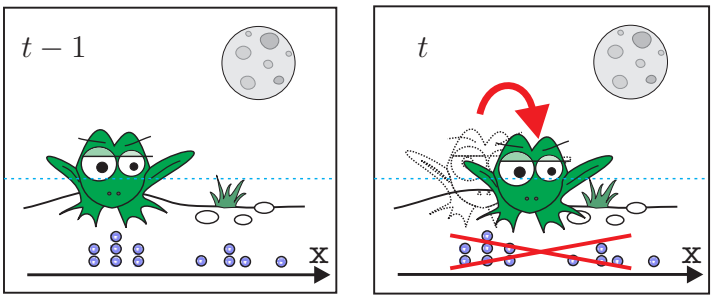
$$\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

2. Operacijo prevzorčenja uporabimo v vsaki časovni iteraciji (ekvivalentno $N_{thres} = \infty$).

Rezultat je filter, ki se je v literaturi najprej pojavil pod imenom Bayesov Bootstrap filter [16], znan tudi kot *vzorčenje s prevzorčenjem* (SIR) *(angl. Sampling Importance Resampling)* [6]. Kasneje se je enak fiter v računalniškem vidu pojavil pod imenom algoritem CONDENSATION [2]. Kot zanimivost omenimo, da CONDENSATION v originalni publikaciji [2] ni bil predstavljen kot različica filtra z delci, pač pa specifično kot princip pridobivanja a posteriori porazdelitve stanj ob nekem časovnem koraku in uporaba te a posteriori kot a priori v naslednjem časovnem koraku.

V računalniškem vidu je algoritem CONDENSATION zelo pomemben, saj je prav ta sprožil izreden interes za filtre z delci na tem področju. Z umestitvijo filtra v sekvenčne MC metode se je v literaturi računalniškega vida pojavilo mnogo implementacij tega ter drugačnih filtrov z delci. Splošna implementacija filtra z delci je prikazana v algoritmu 2.4.

Ker CONDENSATION algoritem predstavlja jedro vseh sledilnikov, ki jih bomo predstavili v tem magistrskem delu, bomo zaradi jasnosti prikazali eno iteracijo tega algoritma na spodnjem primeru, s katerim bomo tudi zaključili to poglavje.



(a) (b)

Slika 2.4: *Ilustracija problema v primeru 3: Zanima nas položaj žabice vzdolž horizontalne osi (označeno z modro črtkano črto). Znana je aproksimacija a posteriori porazdelitve $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ iz prejšnjega časovnega koraka $(t-1)$, ki je v (a) prikazana s kroglicami pod žabico; več kroglic v stolpcu nakazuje večjo gostoto verjetnosti v tem področju. V trenutnem časovnem koraku t nas zanima aproksimacija nove a posteriori $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, ki pa je različna od predhodne (b).*

Vhod:

- A posteriori porazdelitev stanj iz prejšnjega časovnega koraka $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \approx \{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$

Izhod:

- Nova a posteriori porazdelitev stanj $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$

---

1. Prevzorči $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$ z algoritmom 2.2.
   - $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N} \rightarrow \{\tilde{\mathbf{x}}_{t-1}^{(i)}, \frac{1}{N}\}_{i=1}^{N}$
2. Za $i = 1 : N$,
   - Vzorči nov delec: $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\tilde{\mathbf{x}}_{t-1}^{(i)})$
   - Določi utež: $\tilde{w}_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$
3. Za $i = 1 : N$,
   - Normaliziraj uteži delcev: $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}$
4. Dobljena mera predstavlja empirični približek pravi porazdelitvi: $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N} \approx p(\mathbf{x}_t|\mathbf{y}_{1:t})$

Algoritem 2.4: *algoritem* CONDENSATION.

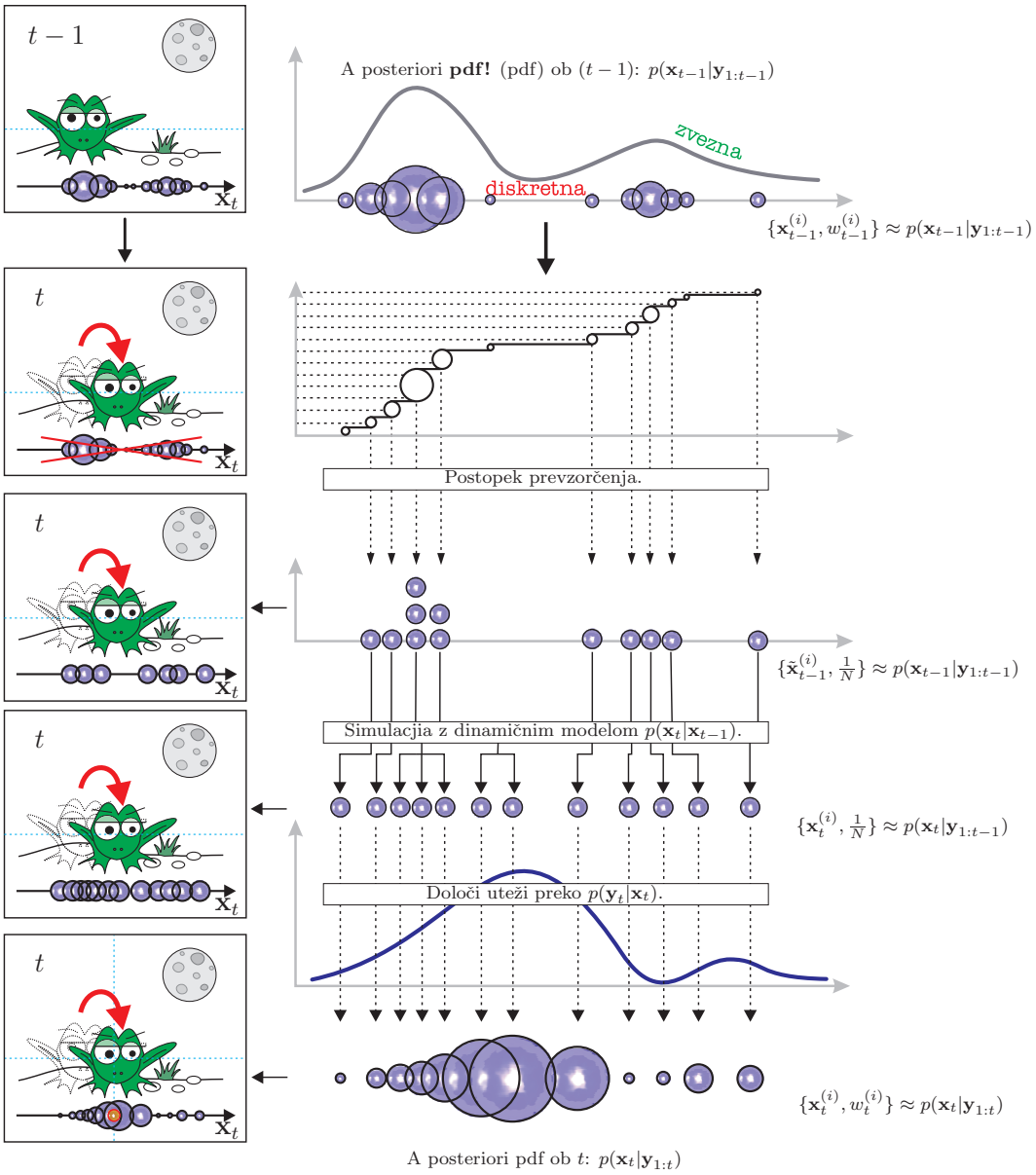
Primer 3. *Zopet vzemimo primer ocenjevanja položaja žabice vzdolž horizontalne osi v dveh zaporednih časovnih korakih in sicer* $(t-1)$ *ter* t *(slika 2.4). Ob času* t *poznamo empirični približek porazdelitvi* $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$*, ki je* $\{\boldsymbol{x}_{t-1}^{(i)}, w_(t-1)^{(i)}\}_{i=1}^{N}$*, zanima pa nas porazdelitev* $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$*. Slika 2.5 prikazuje eno iteracijo algoritma 2.4:*

- *Najprej delce prevzorčimo z deterministčnim prevzorčenjem (algoritem 2.2). Dobljena empirična porazdelitev* $\{\tilde{\mathbf{x}}_{t-1}^{(i)}, \frac{1}{N}\}$ *je še vedno aproksimacija* $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$*. Delci z višjimi utežmi so v tej porazdelitvi razmnoženi, medtem ko so nekateri z nizkimi zavrženi.*

- *Nato preko dinamičnega modela* $p(\boldsymbol{x}_t|\tilde{\boldsymbol{x}}_{t-1}^{(i)})$ *simuliramo prehod vseh* N *delcev* $\tilde{\boldsymbol{x}}_{t-1}^{(i)}$ *v nova stanja. Dobljena empirična porazdelitev* $\{\boldsymbol{x}_t^{(i)}, \frac{1}{N}\}$ *je aproksimacija porazdelitvi* $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$*, ki je predikcija a posteriori porazdelitve ob času* t*.*

- *Vsem delcem določimo uteži preko* $p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)})$ *in jih normaliziramo. Dobljeni nabor delcev* $\{\boldsymbol{x}_t^{(i)}, w_t^{(i)}\}$ *je že empirični približek porazdelitvi* $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$.

Za primer smo na dobljeni porazdelitvi izračunali še ocenjeno povprečno stanje tarče kot $\hat{\boldsymbol{x}}_t \approx \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_t^{(i)}$, *in je prikazano v sliki 2.5 z oranžnim krogcem levo spodaj.*



Slika 2.5: *Ilustracija ene časovne iteracije algoritma* CONDENSATION *na primeru sledenja žabice v zaporednih slikah 2.4(a,b). A posteriori porazdelitev stanj žabice smo predstavili z dvanajstimi delci (N=12), vsak delec, oziroma vzorčeno stanje pa smo označili s krogci. Center krogca predstavlja stanje (položaj žabice vzdolž x osi), njegov polmer pa označuje utež delca; večji ko je radij, večjo utež ima delec. Povprečno stanje smo na spodnji levi sliki žabice označili z oranžnim krogcem.*

Poglavje 3

Zaprti svet

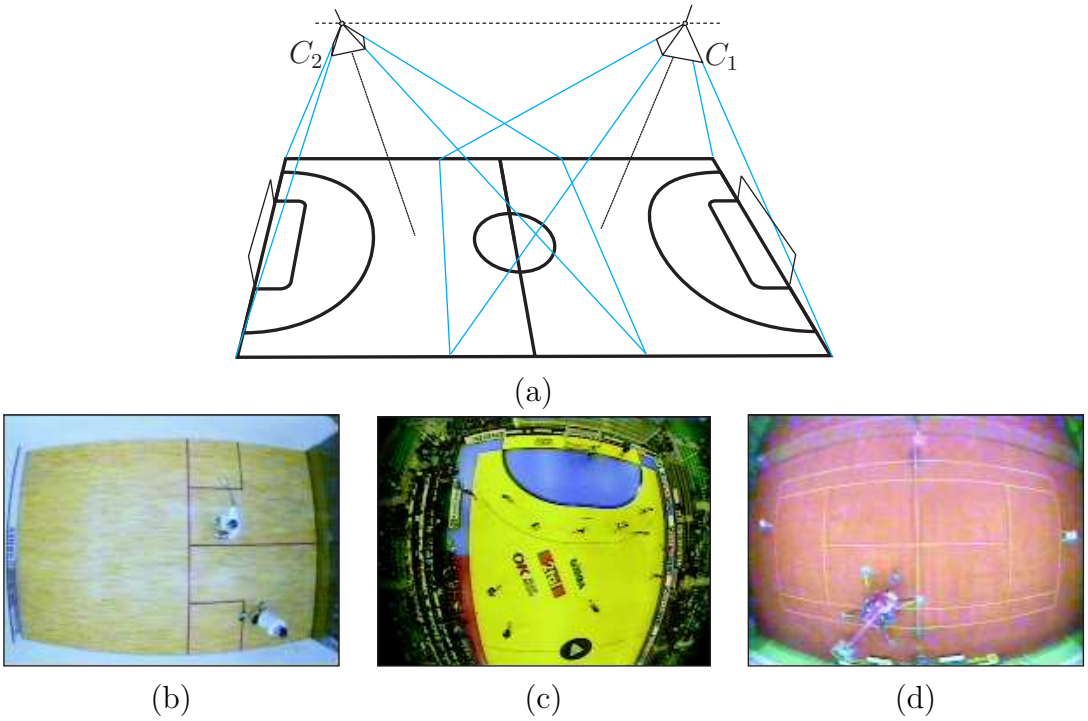
Problem sledenja smo v prejšnjem poglavju predstavili v kontekstu stohastičnega ocenjevanja. Ta pogled nam služi za izdelavo statističnega algoritma za sledenje objektov v video posnetkih. Naša naloga je sicer nekoliko bolj specifična kot le sledenje poljubnih objektov v zaporedju slik: rešiti je potrebno problem sledenja igralcev v športni igri. Kakor vsaka aplikacija, ima tudi sledenje igralcev v športu svoje zahteve, predpostavke ter omejitve. V preteklosti so se športne igre že izkazale kot primerne za opis v kontekstu tako imenovanega *zaprtega sveta* (ZS) *(angl. closed world)*, termin, ki sta ga leta 1995 vpeljala Intille in Bobick [17] pri sledenju igralcev v ameriškem nogometu. V tem poglavju bomo zato najprej na kratko opisali zajem videa tipične športne igre in idejo zaprtega sveta, potem pa bomo predstavili konkretno problematiko sledenja v zaprtem svetu.

3.1 Postavitev kamer

Naša naloga je sledenje igralcev v športni igri, z namenom pridobivanja položajev igralcev na igrišču v vsakem časovnem trenutku. Postavitev kamer na igriču močno vpliva na kvaliteto informacije, ki jo pridobivamo s sledenjem. Ker nas zanima le translatorno gibanje na igrišču, torej, nas ne zanima komponenta gibanja pravokotno na tla, je najbolj smiselna postavitev kamer taka, da so kamere nameščene visoko nad igriščem in so njihove optične osi kar se da pravokotne na tla. Ta postavitev kamer se je izkazala za primerno v preteklih aplikacijah sledenja [18] in jo prikazuje slika 3.1(a); tipične slike s tako postavljeno kamero prikazujejo slike 3.1(b-d).

3.2 Koncept zaprtega sveta

Glavna premisa koncepta zaprtega sveta je, da za neko območje v prostoru in času obstaja določen kontekst, s katerim lahko slednjega razložimo. Območje



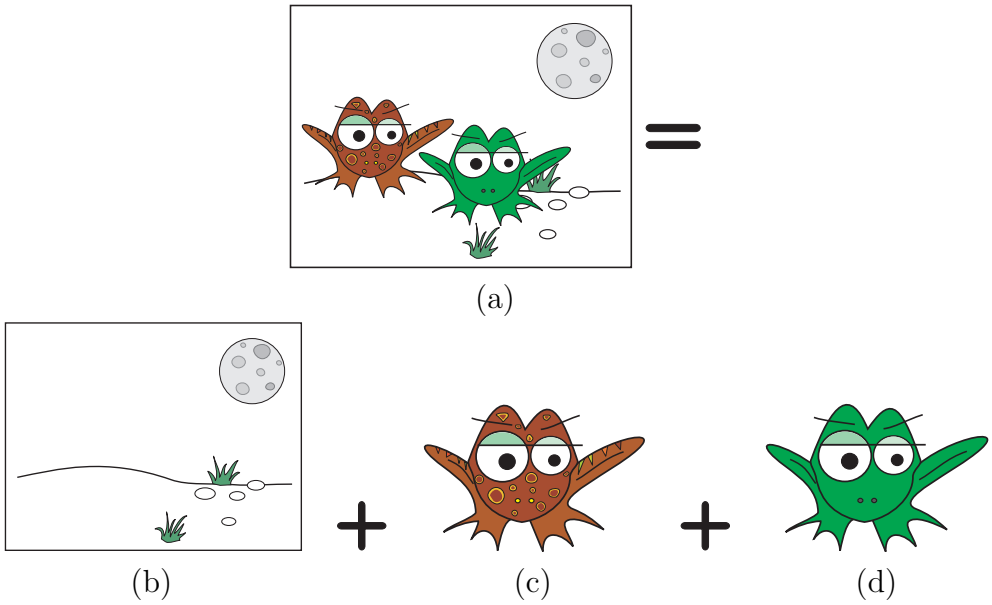
Slika 3.1: *Skica namestitve kamer za sledenje igralcev v športnih igrah (a) in tipične slike zajete pri squashu, (b); rokometu, (c) in tenisu (d)*

imenujemo zaprti svet, kontekst pa je meja v prostoru znanja, zunaj katere znanje ne pripomore k reševanju problema sledenja. Zaradi jasnosti bomo pravkar vpeljane termine predstavili na primeru 4 spodaj.

Primer 4. *Recimo, da opazujemo neko področje, znotraj katerega se gibljeta dve žabici različnih barv, katerih položaj nas zanima. To območje imenujemo zaprti svet, saj obstaja kontekst, s katerim lahko vsako konfuguracijo žabic v sliki pojasnimo. Kontekst, s katerim lahko pojasnimo zaprti svet, je slika ozadja (slika 3.2b); ter sliki žabic (slika 3.2c,d). Določena superpozicija teh slik pojasnjuje zaprti svet v nekem trenutku.*

3.2.1 Športna igra kot zaprti svet

Športne igre so že same po sebi navadno omejene z bolj ali manj strogimi pravili in so zato zelo primerne za predstavitev v zaprtem svetu: zanimivo dogajanje je omejeno z mejami igrišča, zato je potrebno opazovati le dogajanje znotraj teh meja, prav tako je v naprej znano število igralcev na igrišču, njihovi izgledi, itd. Z znano postavitvijo kamer, ki smo jo opisali v začetku tega poglavja ter lastnostmi



Slika 3.2: *Primer zaprtega sveta. Slike žabic (c,d) ter slika ozadja (b) tvorijo tvorijo kontekst, s katerim lahko pojasnimo sliko (a).*

športne igre, lahko sedaj predstavimo problem sledenja s skupino predpostavk zaprtega sveta:

1. Kamera, ki opazuje igrišče, je statična in je nameščena visoko nad igriščem, z optično osjo približno pravokotno na igrišče.
2. Dogajanje, ki ga opazuje kamera, je zaprti svet.
3. Igrišče je omejeno in njegov vizualni model lahko določimo.
4. Število igralcev na igrišču je znano.
5. Med igralci lahko pride do kratkotrajnih trkov ali delnih zakrivanj.
6. Izgledi igralcev so znani na začetku ter se s časom spreminjajo.
7. Osvetljenost je neenakomerna preko igrišča ter se s časom spreminja.

Zgornje predpostavke nam služijo kot nabor pravil in zahtev, znotraj katerih je sledenje v konkretnem problemu uspešno. Torej upoštevanje tega nabora bo pripeljalo do delujoče aplikacije za sledenje v športnih igrah.

Poglavje 4

Referenčni sledilnik za sledenje enega igralca

V tem poglavju bomo z upoštevanjem le nekaterih predpostavk zaprtega sveta zgradili sledilnik za sledenje enega igralca. Ker bo ta sledilnik v nadaljevanju služil za izgradnjo bolj kompleksnega, ki bo upošteval več predpostavk, ga imenujemo *referenčni* sledilnik.

4.1 Vizualna značilnica igralca

Video posnetki, oziroma sekvence slik, predstavljajo množico informacije, na podlagi katere želimo pridobivati določeno znanje o sledenem objektu. V našem primeru nas zanima položaj igralca na igrišču. Ker je kamera senzor, lahko na pridobljene slike gledamo kot signale, ki vsebujejo lokalne vizualne značilnosti opazovane scene. Kadar opazujemo igrišče na katerem se nahajajo igralci, so ti signali rezultat odziva na vizualne lastnosti igrišča ter igralcev. To pomeni, da se znotraj signala nahajajo odseki, ki imajo značilnosti vizualnih odzivov na igralca. Lokalizacija teh področij v sekvenci slik omogoča pridobivanje *zanimive informacije*, postopek, kateremu smo v poglavju 2 rekli sledenje. Prvi korak k sledenju igralcev je torej zapis, oziroma predstavitev igralca z nekimi čim bolj stabilnimi in merljivimi značilnicami, ki omogoča kar se da robustno razločevanje igralca od ostalih elementov v sliki.

Kadar sledimo objektom z zelo značilnimi oblikami in se teh oblik predhodno lahko naučimo, je zelo primerno take objekte predstaviti s konturnimi značilnicami. Gre za predstavitev sledenega objekta z majhno množico kontrolnih točk, ki so nanizane vzdolž njegove silhuete in povezane z zlepki *(angl. spline)* [19]. Statistične modele teh oblik, ki vsebujejo tudi dinamične lastnosti objekta v literaturi imenujejo aktivne konture *(angl. active contours)* [19]. Uporaba tovrstnih modelov se je izkazala za uporabno v aplikacijah nadzorovanja [20, 21, 22], kjer so avtorji z v naprej zgrajenimi modeli obrisov ljudi sledili pešce. V [2,13]

avtorji z aktivnimi konturami uspešno sledijo obrisom glave enega ali večih ljudi, v [23] pa je prikazana aplikacija sledenja ptic v letu. Konture so navadno zelo občutljive na šum, uspešnost sledenja z njimi pa je poleg tega močno odvisna od možnih oblik, ki jih sledeni objekt lahko zavzame.

V aplikaciji sledenja ljudi v nogometu [24] so avtorji ugotovili, da so konture neprimerne za predstavitev igralcev, saj le-ti med tekom preveč spreminjajo obliko. Za predstavitev stranskega izgleda igralcev zato uporabljajo pet tako imenovanih multiresolucijskih jeder, katera so avtomatsko določili na podlagi velike množice ročno označenih igralcev. Podobno so na podlagi valčne transformacije v [25] avtorji prikazali princip izgradnje jedra za lokalizacijo pešcev v slikah. Perš in Kovačič v [26] sledita igralce rokometa na podlagi odziva na 14 ročno določenih dvonivojskih jeder in vpeljeta adaptivno učenje na odziv. V znani aplikaciji nadzorovanja W^4 [27] avtorji predstavijo ljudi s tako imenovanimi dinamičnimi teksturno-časovnimi predlogami. Te predloge med sledenjem gradijo sproti in skušajo z njimi doseči tako predstavitev, ki vsebuje barvni izgled, kakor tudi eksplicitno informacijo o obliki sledene osebe. Nekoliko preprostejše barvne predloge igralcev v [28] avtorji uporabljajo za sledenje v ameriškem nogometu: najprej igralca lokalizirajo, izločijo ozadje in dobijo predlogo, s katero igralca iščejo v naslednji sliki. Enak princip uporabljajo v [29] za sledenje igralcev v nogometu. V [30] si avtorji pomagajo pri sledenju igralcev hokeja z uporabo kaskadnega algortima AdaBoost, katerega naučijo na veliki bazi ročno označenih igralcev.

Kadar sledimo ljudi v barvnih posnetkih, je lahko dobra značilnost že npr. barva obleke. V [31] avtorji detektirajo in sledijo otroke v sobi na podlagi njihove povprečne barve. Povprečno barva se je izkazala za zelo uporabno tudi pri sledenju v težavnejših okoljih. V [26] avtorji sledijo igralcem rokometa in vsakega igralca predstavijo z njegovo povprečno barvo. V [32,33] avtorji modelirajo igralca nogometa z dvema povprečnima barvama in sicer barvo majice ter barvo hlač. V aplikaciji nadzorovanja [20] modelirajo izgled osebe z večimi segmenti, kjer barvo vsakega segmenta zapišejo s povprečno vrednostjo in varianco. V [34] so avtorji sledenje roke s konturami izboljšali z modeliranjem barve kože z Gaussovo porazdelitvijo v RGB barvnem prostoru.

McKenna in Raja sta v [35] sledeni objekt predstavila z mešanico Gaussov v barvnem prostoru HS. Tako predstavitev so v [36] uporabili v aplikaciji nadzorovanja, medtem ko so avtorji v [23] mešanico Gaussov v RGB prostoru uporabili pri sledenju ptic v letu.

Neparametrični zapis barve objektov omogočajo barvni histogrami. Ta princip sta Swain in Ballard v [37] uporabila za poizvedovanje po slikovnih bazah. Barvni histogrami so tradicionalen način zapisa statistike nizkonivojskih barvnih lastnosti objektov. Lahko so določeni kot npr. ločene porazdelitve po posameznem barvnem kanalu ali pa kot porazdelitev preko celotnega barvnega

prostora. Taka predstavitev objektov se je izkazala za izredno primerno v številnih aplikacijah sledenja v športu [30, 38, 39, 40, 41], nadzorovanja [42, 43], sledenja obrazov [24] ter področij barve kože [44]. Razlog je v tem, da barvni histogrami nudijo preprost in robusten zapis barvnih lastnosti sledenih objektov, kadar je barva *stabilna* značilnica.

V naši aplikaciji so igralci navadno predstavljeni z majhnim številom slikovnih elementov, njihovi dresi so lahko večbarvni, njihove silhuete pa se hitro spreminjajo in niso vedno dobro razločljive. Upoštevajoč ta dejstva, nudijo barvni histogrami najbolj primereno predstavitev igralcev. V tem poglavju bomo zato bolj podrobno predstavili tak zapis objekta ter izbrali mero podobnosti za primerjanje histogramov.

4.1.1 Predstavitev z barvnim histogramom

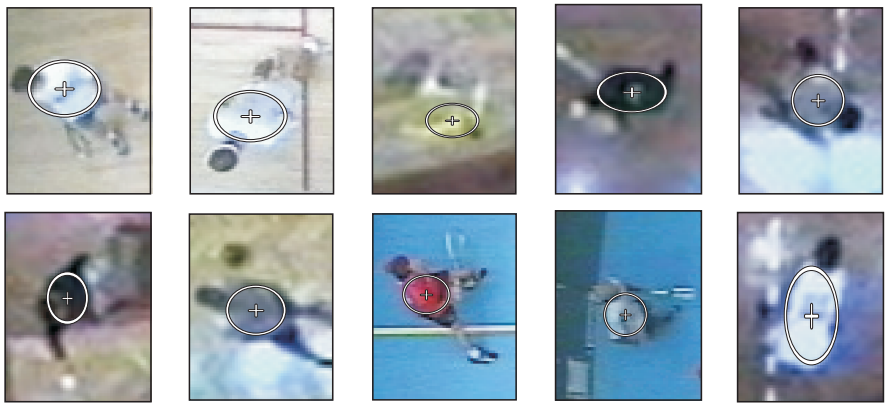
Histogrami predstavljajo teksturno lastnost neke množice slikovnih elementov, ki se nahajajo znotraj določenega območja. To območje je v splošnem lahko poljubne oblike in zato je ustrezen zapis tega področja pomemben del pri uporabi histogramov. V idealnem primeru je področje omejeno z igralčevo silhueto in vsi slikovni elementi znotraj tega področja pripadajo igralcu. Zares je taka predstavitev območja visokodimenzionalna in neizvedljiva v praktičnih situacijah, s katerimi se srečujem v naših aplikacijah. V poglavju 2, ko smo govorili o zanimivi informaciji, smo zaključili, da imamo v praktičnih primerih vedno opravka z aproksimacijo tarče. To pomeni, da potrebujemo tak nizkodimenzionalen model področja, ki je sposoben zadovoljivo opisati številne oblike igralca na igrišču. Na tipičnih slikah igralcev (slika 5.1) vidimo, da pokončna elipsa[1] primerno opiše vsa področja, ki vsebujejo igralce v različnih položajih. Z elipso parametrizirano območje označimo z $E = \{x_E, y_E, a_E, b_E\}$, kjer je $\{x_E, y_E\}$ center elipse, $\{a_E, b_E\}$ pa njena širina ter višina (slika 4.2).

Ker z elipso le približno opišemo obliko igralca, se v njeni notranjosti ob robu včasih pojavljajo slikovni elementi, ki ne pripadajo igralcu, pač pa ozadju (slika 5.1). Če torej predpostavimo, da slikovni elementi, ki so bližje centru pripadajo tarči z večjo gotovostjo kot tisti, bliže robu, moramo histogram znotraj področja vzorčiti *uteženo* in sicer tako, da pripišemo slikovnim elementom pri robu elipse manjše uteži, kot tistim blizu centra.

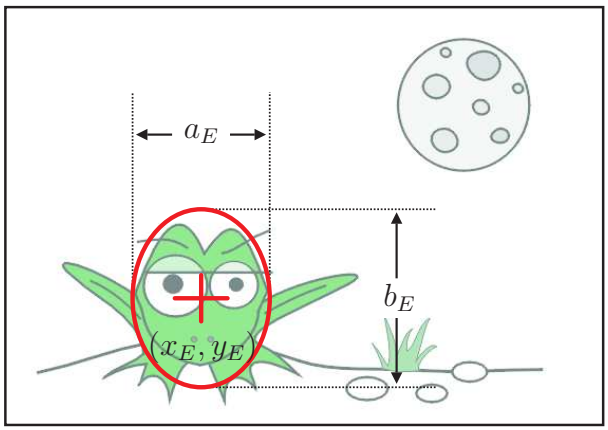
Primer takega jedra na področju E opisuje funkcija $k_E : \mathbb{R}^2 \to \mathbb{R}$, definirana s predpisom

$$k_E = \begin{cases} 1 - \|\frac{(x_E,y_E)-(u_x,u_y)}{f_E}\|^2 & ; \quad \|\frac{(x_E,y_E)-(u_x,u_y)}{f_E}\|^2 < 1 \\ 0 & ; \quad sicer \end{cases}, \tag{4.1}$$

[1] Elipsi pravimo *pokončna* zato, ker parameter kota ni prosta spremenljivka in je taka elipsa enolično določena le z njenim centrom, širino ter višino.



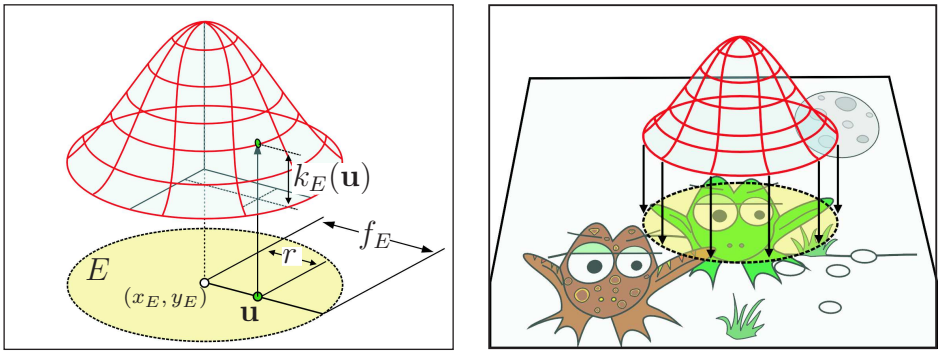
Slika 4.1: *Opis področij za zajem histograma z elipsami.*



Slika 4.2: *Parametrizacija elipse.*

kjer je $\mathbf{u} = (u_x, u_y)$ neka točka znotraj področja E, (x_E, y_E) je center tega področja, $\|\cdot\|$ označuje l_2 normo, f_E pa je razdalja od centra elipse do njenega roba v smeri točke $\mathbf{u}$. Skico utežnega jedra definiranega z izrazom (4.1) prikazuje slika 4.3(a), sicer pa je to le eno od možnih jeder; v [40] npr. kot alternativo predlagajo Epanechnikovo jedro.

Kadar v naprej vemo, kateri slikovni elementi na sliki zagotovo ne pripadajo tarči, je smiselno to informacijo uporabiti pri gradnji histogramov. Pravimo, da poznamo dvonivojsko a priori masko $M : \mathbb{N}^2 \rightarrow \{0, 1\}$, ki je ilustrirana v sliki 4.4(b). Utež slikovnega elementa $\mathbf{u}$ je torej nič, če je vrednost $M(\mathbf{u})$ enaka nič, in je sicer določena z utežnim jedrom po enačbi (4.1) (slika 4.4c).



(a) (b)

Slika 4.3: *Ilustracija utežnega jedra. Oblika jedra je taka, da slikovnim elementom, ki so bliže centru, pripisuje večje uteži kot tistim, ki ležijo dlje.*

(a) (b) (c)

Slika 4.4: *Ilustracija gradnje histograma znotraj eliptičnega področja E (a). Slikovnim elementom, ki niso vidni skozi a priori masko, (b) pripišemo utež nič, ostalim pa določa utež jedro (c).*

Na sliki $A(\mathbf{u})$ znotraj eliptičnega področja E vzorčen histogram $\mathbf{h}_A = (h_{1A}, h_{2A}, ..., h_{iA}, ..., h_{mA})$ z m celicami definiramo kot

$$h_{iA} = f_h \sum_{\mathbf{u} \in E} k_E(\mathbf{u}) M(\mathbf{u}) \delta_i(b(\mathbf{u})), \tag{4.2}$$

kjer je h_{iE} i-ta celica histograma, $k_E(\cdot)$ je funkcija utežnega jedra (En. 4.1), $M(\cdot)$ predstavlja funkcijo a priori maske, $\delta_i(\cdot)$ je Kroneckerjev delta usrediščen na i-ti celici, funkcija $b : \mathbb{N}^2 \rightarrow \{1, ..., m\}$ asociira barvo slikovnega elementa na lokaciji $\mathbf{u}$ s celico z indeksom $b(\mathbf{u})$ v histogramu $\mathbf{h}_E$, f_h pa je normalizacijski faktor, določen tako, da je vsota vseh celic v histogramu enaka ena:

$$f_h = \frac{1}{\sum_{\mathbf{u} \in E} k_E(\mathbf{u}) M(\mathbf{u})}. \tag{4.3}$$

Barvni prostor in število celic v histogramu

Pomemben korak, ki ga je potrebno narediti kadar govorimo o uporabi barvnih histogramov, je izbira barvnega prostora ter število celic v histogramu. Izbira barvnega prostora je dostikrat pogojena s področjem uporabe: npr. pri sledenju obrazov sta najbolj popularna barvna prostora HSV ter Lab, saj je znano, da je večina barvnih odtenkov kože dobro lokalizirana v določenem delu teh prostorov. Dobra lastnost teh prostorov je tudi ločitev svetlosti od barve, saj tako nudi svetlostno neobčutljive predstavitve. To lastnost barvnega prostora HSV izrabljajo v [43] za sledenje objektov, v [30] pa za sledenje igralcev hokeja. Za sledenje obrazov v [45] uporabljajo RGB barvni prostor, uspešno uporabo tega prostora pa navajajo tudi v [42, 39, 46, 40, 41] za sledenje ljudi. Upoštevajoč izkušnje teh avtorjev smo izbrali RGB barvni prostor za gradnjo histogramov.

Drugi pomemben parameter pri gradnji histogramov je število celic na barvni kanal. Z izbiro večjih celic lahko npr. naredimo predstavitev z barvnim histogramom robustno na spremembe v svetlosti ter barvi. Parameter števila celic nekoliko variira med avtorji npr. v [46] predlagajo 32 celic na barvni kanal, v [45] 10, v [40] pa 8. Največ primerov uspešnega sledenja, ki so sorodni naši problematiki navajajo v [40], zato smo se odločili za RGB barvne histograme z osmimi celicami na barvni kanal.

Naj bo $\mathbf{u}$ nek slikovni element v sliki z barvo (r, g, b), kjer so vrednosti r, g, b cela števila na intervalu $[0, 255]$. Trojček $(r', g', b') \in \mathbb{Z}^{+3}$ je definiran kot

$$(r', g', b') = \lfloor (r, g, b) \frac{8}{256} \rfloor,$$

kjer je $\lfloor \cdot \rfloor$ operator zaokrožitve navzdol in preslikava $b(\mathbf{u})$ iz enačbe (4.2) se potem glasi

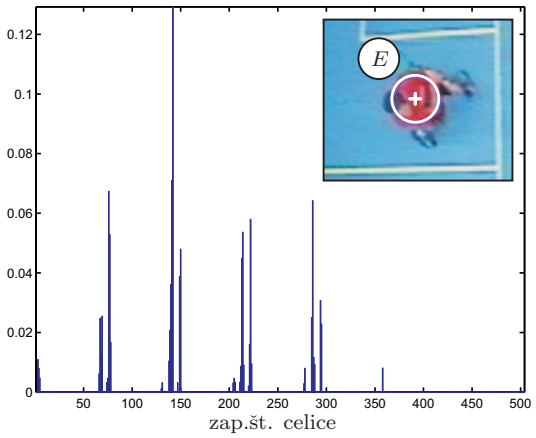
$$b(\mathbf{u}) = 1 + 64b' + 8g' + r'. \tag{4.4}$$

Primer RGB histograma igralca squasha z $(8 \times 8 \times 8)$ celic prikazuje slika 4.5.

Do sedaj smo govorili o tem, kako neko področje opisati z barvnim histogramom. Za potrebe sledenja na podlagi histogramov, pa je potrebno določiti še mero podobnosti, oziroma različnosti med histogrami.

Mera razdalje med histogrami

Zaradi razširjenosti uporabe barvnih histogramov v računalniškem vidu je bilo razvitih mnogo mer za primerjavo histogramov. Opis nekaterih mer v aplikacijah poizvedovanja po slikovnih bazah in segmentacji video posnetkov najdemo v [47, 37, 48], v aplikacijah sledenja na podlagi barvnih histogramov, pa se je v zadnjem času uveljavila mera, ki temelji na statističnem pristopu. Histograme si namreč lahko predstavljamo kot funkcije porazdelitve verjetnosti med m-elementi, kjer je



Slika 4.5: *Slika igralca squasha in primer barvnega RGB histograma znotraj elipse E z $(8 \times 8 \times 8)$ celicami.*

m dimenzija histograma. Ugotavljanje podobnosti med dvema histogramoma je torej ekvivalentno testiranju konsistentnosti dveh verjetnostnih porazdelitev. Mer za preverjanje konsistentnosti porazdelitev je sicer mnogo, v aplikaciji sledenja objektov pa so avtorji v [46] za primerjanje histogramov predlagali uporabo zelo znanega koeficienta Bhattacharyya [49]. Za dva histograma $\mathbf{h}_A = (h_{1A}, ..., h_{mA})$ in $\mathbf{h}_B = (h_{1B}, ..., h_{mB})$ z enakim številom celic m je koeficient Bhattacharyya definiran kot

$$\rho(\mathbf{h}_A, \mathbf{h}_B) = \sum_{i=1}^{m} \sqrt{h_{iA} h_{iB}}. \tag{4.5}$$

V zadnjih petih letih, po objavi članka [46], je koeficienta Bhattacharyya ter z njim predstavljena mera v [46] postal klasičen ter praktično najpopularnejši pristop k primerjanju barvnih histogramov v aplikacijah sledenja, kot so na primer [43,40,41,30]. Ideja uporabe tega koeficienta za primerjanje histogramov je seveda starejša; v [50] so npr. avtorji analizirali njegove lastnosti ter ga predlagali kot absolutno metriko med histogrami.

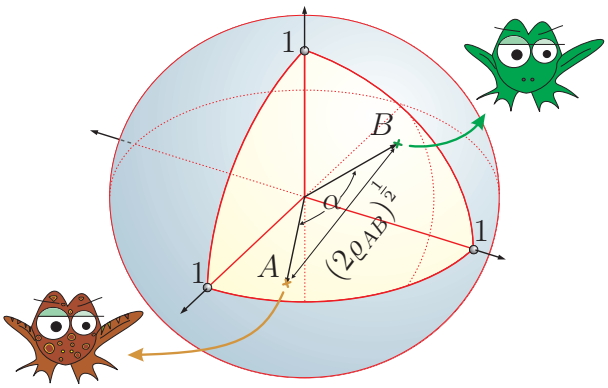
Razlogi za priljubljenost tega koeficienta, ki jih v glavnem navajajo v literaturi so sledeči:

- Njegova povezanost z zgornjo mejo verjetnosti napake Bayesovega razvrščanja [46].
- Dejstvo, da je omejen na interval $[0, 1]$ [43].
- V nasprotju Kullback-Leiblerjevo divergenco koeficient $\rho(\cdot, \cdot)$ ne trpi zaradi singularnosti pri praznih celicah [43].

Kot zanimivost naj omenimo, da je bila prvotna interpretacija koeficienta Bhattacharyya geometrijske narave [49]: V izvirnem delu je Bhattacharyya namreč izpostavil, da si lahko $(\sqrt{h_{1A}}, ..., \sqrt{h_{mA}})$ in $(\sqrt{h_{1B}}, ..., \sqrt{h_{mB}})$ predstavljamo kot smerne kosinuse dveh enotskih vektorjev

$$\mathbf{h}'_A = (\sqrt{h_{1A}}, ..., \sqrt{h_{mA}}) \text{ in } \mathbf{h}'_B = (\sqrt{h_{1B}}, ..., \sqrt{h_{mB}})$$

v m-dimenzionalnem ortogonalnem prostoru. Koeficient Bhattacharyya je potem enak skalarnemu produktu med vektorjema $\mathbf{h}'_A$ ter $\mathbf{h}'_B$, oziroma, ker sta enotska, kar kosinusu kota α, ki ga vektorja oklepata, $\langle \mathbf{h}'_A, \mathbf{h}'_B \rangle = cos\alpha$. Vrednosti kota α so očitno omejene na interval $[0, \pi/2]$ in torej sledi, da vrednost koeficienta $\rho(\mathbf{h}_A, \mathbf{h}_B)$ zavzame vrednost nič, ko sta vektorja $\mathbf{h}'_A$ in $\mathbf{h}'_B$ pravokotna, oziroma zavzame vrednost ena, ko sta enaka. Skico geometrijske interpretacije podobnosti dveh histogramov prikazuje slika. 4.6.



Slika 4.6: *Ilustracija geometrijske interpretacije dveh histogramov* $\mathbf{h}_A$ *ter* $\mathbf{h}_B$ *s trodimenzionalnima vektorjema* $\mathbf{h}'_A$ *ter* $\mathbf{h}'_B$ *na enotski krogli. Kot med vektorjema je označen z* α*, medtem ko je ločna razdalja med točkama* A *in* B *označena s korenom dvakratne Hellingerjeve diskriminacije* ϱ_{AB}*. Ker sta vektorja* $\mathbf{h}'_A$ *ter* $\mathbf{h}'_B$ *enotska s komponentami omejenimi na* $[0, 1]$*, je prostor vseh možnih konfiguracij omejen na označeno osmino krogle.*

Ker je koeficient $\rho(\cdot, \cdot)$ (enačba 4.5) *mera podobnosti* z vrednostjo ena pri največji podobnosti ter vrednostjo nič pri najmanjši, smo mero različnosti definirali kot

$$\varrho(\mathbf{h}_A, \mathbf{h}_B) = 1 - \rho(\mathbf{h}_A, \mathbf{h}_B), \tag{4.6}$$

ki pa je ravno polovica zelo znane Hellingerjeve razdalje (glej npr. [51, 52]), in doseže največjo vrednost 1 ko sta histograma najbolj različna ter najnižjo vrednost 0 ko sta enaka. Zaradi lažje vizualizacije omenimo, da je geometrijsko gledano koren dvakratne Hellingerjeve razdalje $\varrho(\mathbf{h}_A, \mathbf{h}_B)$ enak razdalji med vrhovi prej definiranih vektorjev $\mathbf{h}'_A$ ter $\mathbf{h}'_B$, kar je tudi skicirano na sliki. 4.6.

Koren izraza (4.6) v literaturi pogosto imenujejo Bhattacharyyjeva razdalja in je bil izbira mnogih avtorjev aplikacij sledenja s primerjavo histogramov [46, 40, 45, 30]. Na tem mestu je smiselno poudariti, da vsi avtorji modernih algoritmov slednja (npr. [40, 45, 41, 30]) govorijo o izbiri *Bhattacharyyjeve razdalje* kot mere podobnosti med histogrami, vendar potem dejansko primerjavo histogramov izvedejo preko kvadrata te razdalje – torej Hellingerjeve diskriminacije (enačba 4.6). To seveda nikakor ne pomeni, da vsi ti prispevki vsebujejo napako, vendar pa ta detajl nikjer ni izpostavljen.

4.2 Funkcija verjetja

Funkcija verjetja *(angl. likelihood function)* je eden od sestavnih delov vsakega filtra z delci. Delcem, oziroma stanjem, prav preko te funkcije določamo uteži, ki so proporcionalne verjetnosti opazovane meritve pri določenem stanju. Potrebujemo torej neko distančno mero med *vizualnim* modelom tarče ter meritvami, pdf te mere pa predstavlja funkcijo verjetja.

Recimo, da barvni histogram $\hat{\mathbf{h}}_{ref}$ predstavlja vizualni model sledenega igralca. Ob času t filter delcev generira neko stanje $\mathbf{x}_t$ in znotraj elipse E tega stanja opravimo meritev; t.j. vzorčimo histogram $\mathbf{h}_A$ na trenutni sliki. Distančna funkcija med modelom in meritvijo je lahko mera razdalje med histogrami iz enačbe (4.6).

Različni avtorji [40, 30, 43] so ugotovili, da se ta mera porazdeljuje eksponentno, torej v obliki

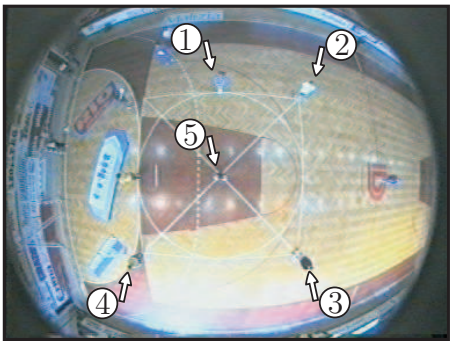
$$p(\mathbf{y}_t|\mathbf{x}_t) = \lambda exp(-\lambda\varrho(\mathbf{h}_{ref}, \mathbf{h}_A)). \tag{4.7}$$

Zanimivo je, da so avtorji v [40] in [43] uporabljali različna barvna prostora za izgradnjo histogramov in različno zrnenje barvnih prostorov, prišli pa so do podobnih zaključkov o obliki porazdelitve mere razdalje iz enačbe (4.6).

Edini prosti parameter v pdf iz enačbe (4.7) je λ, katerega smo določili s poskusom. Izbrali smo posnetek dolg 1500 slik, kjer je pet igralcev stalo na mestu (slika 4.7). Vsakega igralca smo ročno označili z elipso, in posneli sekvenco histogramov znotraj te elipse. Referenčni histogram določenega igralca smo določili kot povprečni histogram te sekvence. Za vsak histogram v sekvenci smo nato izračunali razdaljo $\varrho(\cdot,\cdot)$ do povprečnega. Na podlagi tako pridobljenih razdalj vseh igralcev smo s postopkom največjega verjetja (ML) *(angl. maximum likelihood)* določili vrednost parametra λ, ki je znašal

$$\lambda = 12.$$

S tem je pdf v enačbi (4.7) popolnoma določena.



Slika 4.7: *Pet igralcev je v sekvenci 1500 slik stalo na mestu. Z opazovanjem histogramov teh igralcev smo določili parameter modela funkcije verjetja.*

4.3 Dinamika gibanja igralca

Druga pomembna značilnost sledenega objekta, poleg vizualnih, je njegovo obnašanje ali tako imenovani *dinamični model (angl. dynamical model)*, ki zajema lastnosti gibanja. Ustrezno modeliranje te dinamike lahko precej izboljša sledenje, saj se pogosto opazovan proces podreja precej specifičnim dinamičnim lastnostim.

Ko smo predstavili filtre delcev (poglavje 2.1), smo dinamične lastnosti tarče modelirali z Markovovim modelom prvega reda v obliki porazdelitve

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}),$$

ki ji pravimo *funkcija porazdelitve verjetnosti prehajanja stanj (angl. state transition function)* ali kar dinamični model. V aplikacijah filtrov delcev se dinamiko običajno modelira s tako imenovanimi *avtoregresijskimi procesi* ARP *(angl. autoregressive process)*. To so linearni Gaussovi modeli, katerih uporaba sega v trideseta leta prejšnjega stoletja, ko so bili predstavljeni v študiji števila sončnih peg [53]. Splošna oblika avtoregresijskih modelov sledi zapisu

$$\mathbf{x}_t = \mathbf{A}_0 + \sum_{i=1}^{k} \mathbf{A}_i \mathbf{x}_{t-i} + \mathbf{B}\epsilon; \quad \epsilon \sim \mathcal{N}(\mu, \mathbf{\Lambda}), \tag{4.8}$$

kjer je $\mathcal{N}(\mu, \mathbf{\Lambda})$ normalna porazdelitev s srednjo vrednostjo μ ter kovariančno matriko $\mathbf{\Lambda}$ in je k red procesa[2].

Dinamični model tarče je v splošnem precej odvisen od tipa sledene tarče in njene predstavitve v prostoru stanj. Pri sledenju objektov z aktivnimi konturami,

[2]Na prvi pogled se zdi, da avtoregresijski modeli reda višjega od prvega ($k > 1$) kršijo Markovovo predpostavko o odvisnosti trenutnega stanja od neposredne preteklosti, vendar se izkaže, da lahko ARP poljubnega reda vedno zapišemo v obliki ARP prvega reda, preprosto z uvedbo nove spremenljivke stanja (glej npr. [19] stran 204).

npr. poznamo povprečno ali pričakovano obliko sledenega objekta. Ta oblika predstavlja mirovno stanje procesa in določa prvi člen $\mathbf{A}_0$[3] v enačbi 4.8. V [19] avtorji dinamiko deviacij od tega povprečja modelirajo s prvim ter drugim členom avtoregresijskega procesa – pravimo, da dinamiko oblike modelirajo z avtoregresijskim procesom drugega reda (ARP2) z mirovnim stanjem ($\mathbf{A}_0 \neq \mathbf{0}$).

Kadar sledimo objektom na podlagi njihove oblike, se navadno oblika podreja drugačnemu dinamičnemu procesu kot položaj. Za sledenje objektov z aktivnimi konturami tako avtorji v [19] predlagajo uporabo ARP2 in ugotavljajo, da je smiselno ločevati translatorno gibanje od dinamike oblike. Tako ločevanje oblike in translacije uporabljajo npr. v aplikaciji nadzorovanja [54] ter v [23] za sledenje ptic v letu. V obeh primerih avtorji modelirajo translatorno gibanje z dinamičnim modelom *konstantne hitrosti (angl. constant velocity dynamical model)*, ki je pravzaprav ARP2 ($k = 2$ v enačbi 4.8) brez mirovnega stanja in s parametri

$$\mathbf{A}_0 = \mathbf{0},\ \mathbf{A}_1 = 2\mathbf{I},\ \mathbf{A}_2 = -\mathbf{I}.$$

Različni avtorji so tak model gibanja uporabili za modeliranje gibanja igralcev nogometa [33, 32, 39, 40], v aplikaciji nadzorovanja s statično kamero [40, 43] in sledenjem objektov z mobilnim robotom [55].

S predpostavko o nepredvidljivosti gibanja ljudi v aplikacijah sledenja igralcev v nogometu [56, 38, 41] in hokeju [30] ter aplikacijah nadzorovanja [24, 22] avtorji modelirajo gibanje s t.i. *Gaussovimi naključnimi prehodi (angl. Gaussian random walk)*. Gre za avtoregresijske procese prvega reda (ARP1) brez stabilnega stanja, znane tudi po imenu *Brownovo gibanje (angl. Brownian motion)*, ki ga opišemo z enačbo 4.8, če postavimo $k = 1$, ter

$$\mathbf{A}_0 = \mathbf{0},\ \mathbf{A}_1 = \mathbf{I}.$$

4.3.1 Modeliranje naključnosti gibanja igralca

Narava gibanja igralcev v športnih igrah je na prvi pogled povsem nepredvidljiva. Že sama ideja igre narekuje tako gibanje, ki ga nasprotnik ne predvidi, in s tega vidika lahko rečemo, da je gibanje zares nepredvidljivo – tako rekoč naključno.

Recimo, da ob času t igralca predstavlja elipsa E, ki jo zapišemo v prostoru stanj z $\mathbf{x}_t = (x_t, y_t, a_t, b_t)$. Kot ponavadi je (x_t, y_t) center elipse, (a_t, b_t) pa je njena širina in višina. Zaradi predpostavljene naključnosti gibanja modeliramo prehod stanja ob času t z normalno porazdelitvijo, centrirano na stanju iz prejšnjega časovnega koraka $\mathbf{x}_{t-1}$:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{\Lambda}), \tag{4.9}$$

[3] Za primer določanja tega člena glej npr. [19] stran 193.

kjer je $\mathcal{N}(\cdot; \mu, \mathbf{\Sigma})$ normalna porazdelitev s srednjo vrednostjo μ in kovarianco $\mathbf{\Sigma}$. Zgornji model lahko zapišemo v generativni obliki z ARP1 kot

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \epsilon_t; \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}), \tag{4.10}$$

kjer je $\mathbf{A}_0 = \mathbf{0}$ in $\mathbf{A}_1 = \mathbf{I}$, torej z Brownovim gibanjem.

Model gibanja v enačbi (4.9) je dvoparametrični: prvi parameter (srednja vrednost) je kar prejšnje stanje, drugi parameter (kovariančna matrika) pa je nekako določen s fizičnimi lastnostmi igralca. Najprej predpostavimo, da so parametri spremenljivke stanja (x_t, y_t, a_t, b_t) neodvisne n.s., in je torej kovariančna matrika stanja diagonalna. V tipičnih posnetkih športnih iger gledamo igralce z vrha, zato lahko predpostavimo, da sta varianci parametrov x_t ter y_t v splošnem enaki. Skupno varianco označimo s σ^2_{xy}. Enako lahko sklepamo za parametre velikosti elipse a_t ter b_t in varianco teh parametrov označimo s σ^2_{ab}. Kovariančna matriko iz enačbe (4.9) je torej diagonalna matrika

$$\mathbf{\Lambda} = \begin{bmatrix} \sigma^2_{xy} & 0 & 0 & 0 \\ 0 & \sigma^2_{xy} & 0 & 0 \\ 0 & 0 & \sigma^2_{ab} & 0 \\ 0 & 0 & 0 & \sigma^2_{ab} \end{bmatrix}. \tag{4.11}$$

Parametra σ^2_{xy} ter σ^2_{ab} govorita o spremembi stanja znotraj enega časovnega koraka; npr.: 99% vseh pričakovanih sprememb vzdolž osi x (parameter x_t) znotraj enega časovnega koraka se bo nahajalo na intervalu $\pm 3\sigma_{xy}$. Ta sprememba je jasno odvisna od velikosti igralca na sliki: večji ko je igralec na sliki, z večimi slikovnimi elementi je predstavljen in več slikovnih elementov lahko znaša njegov premik. Če vrednosti parametrov σ^2_{xy} ter σ^2_{ab} obravnavamo kot proporcionalne trenutni velikosti elipse igralca lahko pišemo

$$\sigma_{xy} = H_t \cdot \alpha_{xy}, \tag{4.12}$$

$$\sigma_{ab} = H_t \cdot \alpha_{ab}, \tag{4.13}$$

kjer je H_t mera trenutne *velikosti* elipse, ki jo definiramo kot diagonalo elipsi orisanega pravokotnika

$$H_t = \sqrt{a_t^2 + b_t^2}. \tag{4.14}$$

Ker sta σ^2_{xy} ter σ^2_{ab} odvisni od trenutne velikosti H_t elipse, ki je določena s trenutnim stanjem $\mathbf{x}_t$, bomo kovariančno matriko $\mathbf{\Lambda}$ (enačba 4.11) odslej označevali z $\mathbf{\Lambda}_{\mathbf{x}_t}$.

Parametri naključnosti spremembe stanja

Dinamičnemu modelu iz enačbe (4.9) moramo v naprej določiti vrednosti proporcionalnih parametrov α_{xy} ter α_{ab} (enačba 4.12), ki odražata tipične lastnosti gibanja igralca v športni igri.

Parameter α_{xy} je vezan na statistiko hitrosti gibanja igralca med tekmo in ga je potrebno določiti preko analize gibanja igralcev po igrišču. V [57] so avtorji analizirali gibanje šestih igralcev med rokometno tekmo in izmerili najvišjo hitrost približno $8m/s$. Podobne rezultate so dobili Cambell in ostali, [58], kjer so avtorji z analizo gibanja igralcev rokometa ugotovili, da so ti le v 1% celotne tekme presegli hitrost $8m/s$. Z analizo gibanja igralcev v nogometu so avtorji v [59] določili največjo hitrost igralcev $8.3m/s$; podobno je v [60] znašala najvišja izmerjena hitrost nogometašev $7.8m/s$. Za košarko in druge podobne skupinske športe žal nismo uspeli najti ustreznih publikacij, ki bi vključevale take analize gibanja, zato smo na podlagi zgornjih rezultatov za oceno najvišje hitrosti gibanja igralca izbrali vrednost

$$v_{max} = 8.0m/s.$$

Kot zanimivost omenimo, da je pri modeliranju gibanja igralca nogometa v [61] ekvivalentna največja hitrost znašala $7.5m/s$, vendar avtorji ne navajajo nikakršne trdne osnove za ta podatek.

Ker je frekvenca zajema slik v tipičnih posnetkih $25slik/s$, zapišemo največjo hitrost kot

$$v_{max} = 0.32m/sliko.$$

Elipsa s katero sledimo igralca je običajno nekoliko večja od premera njegovega hrbta (slika 4.8) in jo ocenimo kot $H_t \approx 0.4m$. Največja hitrost zapisana z velikostjo elipse torej znaša

$$v_{max} = 0.8H_t/sliko,$$

kar pomeni, da se igralec med dvema zaporednima slikama lahko premakne največ za 80% velikosti trenutne elipse. Premike modeliramo z Gaussovo porazdelitvijo in ker se v tej porazdelitvi praktično vsi pričakovani premiki po absolutni vrednosti nahajajo znotraj intervala treh standardnih devijacij, lahko maksimalno hitrost zapišemo kot

$$v_{max} = 3\sigma_{xy}/sliko,$$

in ker je

$$\sigma_{xy} = H_t\alpha_{xy},$$

iz enakosti

$$3H_t \cdot \alpha_{xy}/sliko = 0.8H_t/sliko \tag{4.15}$$

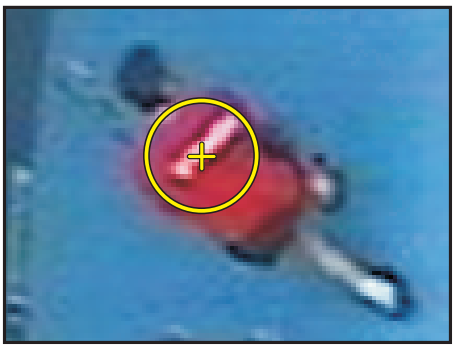
sledi ocena vrednosti proporcionalnega parametra, ki znaša

$$\alpha_{xy} = \frac{0.8}{3} \doteq \frac{1}{4}.$$

To pomeni, da je pričakovana sprememba centra v dveh zaporednih slikah po absolutni vrednosti enaka četrtini trenutne *velikosti* H_t igralca v sliki.

Za določitev parametra spremembe velikosti elipse α_{ab} žal nimamo enako dobre podlage kot v zgornjem primeru. Ker so kamere, s katerimi sledimo igralce navadno nameščene visoko nad igriščem, je glavni vzrok trenutnim spremembam velikosti elipse mahanje z rokami in nagibanje igralca. Predpostavimo, da se med dvema zaporednima slikama širina ali višina elipse lahko največ spremeni za 15% trenutne velikosti ($dH_{max} = 0.15 \cdot H_t$), kar zopet enačimo s tremi standardnimi devijacijami, in iskana vrednost parametra znaša

$$\alpha_{ab} = 0.05.$$



Slika 4.8: *Primer elipse s katero opisujemo igralca. Elipsa približno obsega igralčeva ramena in del hrbta.*

4.4 Generator naključnih števil

Filtri z delci so MC metode, katerih bistvo je generiranje množic naključnih števil in izvajanje določenih operacij nad takimi množicami (npr. simulacija pri dinamičnem modelu iz prejšnjega poglavja). To pomeni, da je eden od poglavitnih delov vsake implementacije filtrov z delci implementacija generatorja naključnih števil. V klasičnih implementacijah filtrov z delci je pridobivanje vrednosti iz poljubnih porazdelitev (npr. normalne) implementirano preko generatorja, ki generira naključna števila iz enakomerne, na nek interval omejene porazdelitve. Različne operacije nad temi naključnimi števili rezultirajo v naključnih številih z različnimi porazdelitvami.

Najbolj običajna izbira generatorja naključnih števil je seveda kar sistemska `rand()` funkcija, ki pa je skoraj vedno linearni kongregenčni generator *(angl. linear congruential generator)* in spada v skupino tako imenovanih *generatorjev psevdo naključnih števil (angl. pseudo-random number generators)*. Ti generatorji

so sicer hitri, vendar problematični pri uporabi v visokodimenzionalnih problemih [62]. V MC literaturi zato predlagajo uporabo generatorjev *kvazi naključnih števil (angl. quasi-random number generators)*. Ti generatorji ne temeljijo na generiranju sekvenc popolnoma naključnih števil/točk (pravzaprav sekvence niti niso naključne), pač pa na generiranju takih števil/točk v prostoru, ki so si maksimalno narazen. Na tak način kvazi naključni generatorji *bolj enakomerno* vzorčijo prostor, rezultat pa je manjša varianca ocene MC metode [62]. S temi argumenti so npr. avtorji v [22] motivirali uporabo Sobolovega generatorja kvazi naključnih števil za sledenje oseb z aktivnimi konturami. Kolikor je znano avtorju magistrske naloge, je to edina ali pa redka publikacija iz področja sledenja s filtri delcev na podlagi vizualnih značilnic, ki namenja posebno pozornost naključnim generatorjem. Praktično nobena druga publikacija na tem področju ne omenja implementacije naključnega generatorja iz česar upravičeno sklepamo, da večina avtorjev v svojih implementacijah uporablja sistemsko funkcijo `rand()`.

Problem sistemske funkcije `rand()` je v tem, da so njene implementacije lahko različne od sistema do sistema ali celo med različnimi verzijami uporabljenega programskega jezika. Zanimivo razpravo o ANSI C in podobnih `rand()` funkcijah najdemo v [62](stran 279). Kot zanimivost, eden od avtorjev [62] se celo spominja, da mu je svetovalec iz centra za programersko podporo IBM-ovih osrednjih *(angl. mainframe)* računalnikov o njihovi sistemski implementaciji generatorja naključnih števil rekel: "Garantiramo, da je vsako posamezno generirano število naključno zase, ne garantiramo pa, da je več kot eno naključno." – avtorji poudarjajo nerazumljivost stavka.

Ker želimo v magistrskem delu predstaviti razvoj aplikacije, ki bazira na MC metodah in s tem na generatorju naključnih števil, se zdi smiselno določiti lastni generator ter v tem smislu implementacijo narediti neodvisno od sistema, verzij programskega jezika, itd. Za generator naključnih števil smo izbrali tako imenovani *Mersennov Vrtinec (angl. Mersenne Twister)* [63], saj je hiter, generira naključna števila z zelo veliko periodo, je primeren za generiranje števil v visokodimenzionalnem prostoru in prestaja tako imenovane *DieHard* preizkuse[4].

4.5 Implementacija referenčnega sledilnika

V poglavju 2 smo umestili sledenje v problematiko stohastičnega ocenjevanja. Ker kamera ni nič drugega kot senzor za pridobivanje določene informacije, lahko Monte Carlo postopke iz poglavja 2 direktno uporabimo za sledenje igralca v video posnetkih. V računalniškem vidu se je kot popularen filter delcev za sledenje na podlagi vizualne informacije uveljavil algoritem CONDENSATION, katerega smo opisali v poglavju 2.5. Implementacija tega filtra je relativno preprosta, saj

[4] DieHard preizkusi so skupina statističnih testov za merjenje kvalitete nabora naključnih števil, katere je več let razvijal George Marsaglia. Skupaj so smatrani kot najstrožji znani testi.

zahteva le določitev funkcije verjetja meritev in dinamični model. Slednja smo skupaj z vizualno značilnico določili v prejšnjih treh podpoglavjih, v tem pa bomo predstavili sledilnik za sledenje enega igralca. Ta sledilnik imenujemo referenčni, saj bo v nadaljevanju služil za razvoj bolj specifičnega sledilnika.

Sledenje z referenčnim sledilnikom poteka na sledeč način. V sliki pred začetkom sledenja najprej ročno označimo igralca. Obliko začetne elipse je potrebno izbrati v naprej, prav tako pa je dobro določiti še največjo in najmanjšo možno elipso. Slednji določimo z največjim in najmanjšim krogom s katerim v danem posnetku lahko opišemo igralca. S tem vnesemo dodatno znanje o sledenem objektu v sledilnik, oziroma določimo del prostora stanj, znotraj katerega bomo sledili objekt. Začetno stanje označimo npr. z $\hat{\mathbf{x}}_0$ in predstavlja pričakovano stanje igralca ob času $t = 0$. Izhodiščno porazdelitev delcev $p(\mathbf{x}_0)$ lahko generiramo tako, da delce vzorčimo kar iz dinamičnega modela

$$\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t|\hat{\mathbf{x}}_0),$$

kjer je $\sim$ operator vzorčenja n.v. iz porazdelitve, in vse uteži delcev postavimo na enako vrednost

$$w_0^{(i)} = \frac{1}{N},$$

kjer je N število vseh delcev v filtru.

Referenčni histogram $\mathbf{h}_{ref}$ vzorčimo znotraj elipse stanja $\hat{\mathbf{x}}_0$. Ta histogram predstavlja barvni model igralca in sledenje si potem lahko v grobem predstavljamo kot iskanje elipse s čim bolj podobnim histogramom.

Ob vsakem časovnem trenutku t izvedemo iteracijo algoritma CONDENSATION in dobimo aproksimacijo porazdelitve $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. Trenutno stanje $\hat{\mathbf{x}}_t$ lahko ocenimo kot *minimalno srednjo kvadratično napako* MSE *(angl. minimum square error)* oceno na tej porazdelitvi[5]; to pa je povprečna vrednost delcev

$$\hat{\mathbf{x}}_t = \sum_{i=1}^{N} w_t^{(i)} \mathbf{x}_t^{(i)}.$$

Implementacijo referenčnega sledilnika prikazuje algoritem 4.1, primer sledenja igralca squasha s tem algoritmom pa slika 4.9.

[5] Trenutno stanje bil lahko sicer ocenili tudi na kak drug način, npr. kot maksimalno a posteriori (MAP) oceno na porazdelitvi, ki je kar delec z najvišjo utežjo.

Vhod:

- A posteriori porazdelitev stanj iz prejšnjega časovnega koraka $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \approx \{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$

Izhod:

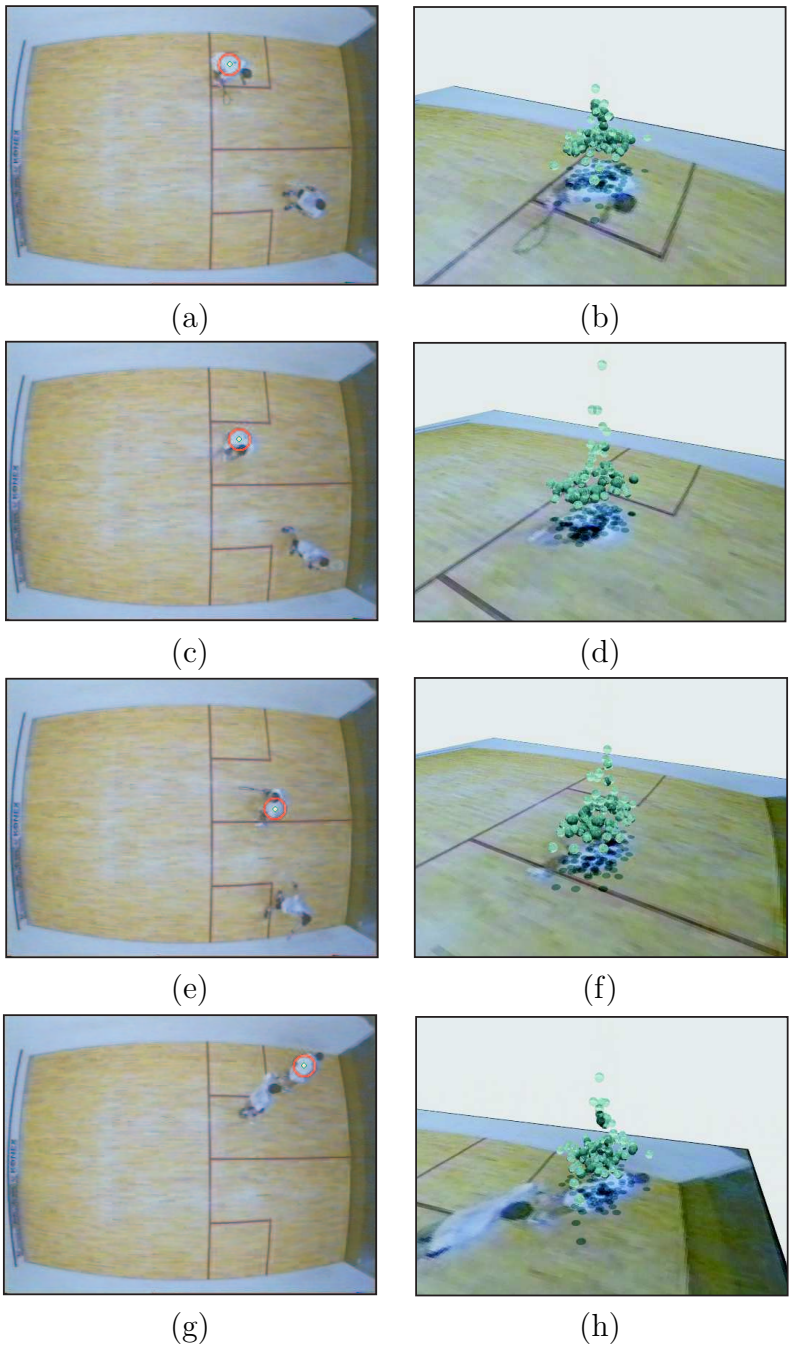
- Nova a posteriori porazdelitev stanj $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$

1. Prevzorči $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$ z determinističnim vzorčenjem (algoritem 2.2),
$$\{\tilde{\mathbf{x}}_{t-1}^{(i)}, \tfrac{1}{N}\}_{i=1}^{N} \approx p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}).$$
2. For i = 1 ... N
   (a) Simuliraj prehod delcev z dinamičnim modelom (enačba 4.10)
   $$\mathbf{x}_t^{(i)} = \tilde{\mathbf{x}}_{t-1}^{(i)} + \epsilon_t; \quad \epsilon_t \sim \mathcal{N}(\cdot; \mathbf{0}, \mathbf{\Lambda}_{\mathbf{x}_{t-1}})$$
   (b) Vzorči histogram $\mathbf{h}^{(i)}$ znotraj elipse stanja $\mathbf{x}_t^{(i)}$ na trenutni sliki.
   (c) Izračunaj razdaljo $\varrho(\mathbf{h}_{ref}, \mathbf{h}^{(i)})$ (enačba 4.6) in določi utež preko funkcije verjetja $p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$ iz enačbe (4.7):
   $$\tilde{w}_t^{(i)} = \frac{1}{\lambda} e^{-\frac{1}{\lambda}\varrho(\mathbf{h}_{ref}, \mathbf{h}^{(i)})}.$$

   End For
3. Normiraj uteži delcev:
   - For i = 1 ... N
   $$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}$$

     End For

Algoritem 4.1: *Implementacija referenčnega sledilnika za sledenje enega igralca.*



Slika 4.9: *Sledenje enega igralca squasha z algoritmom 4.1, kjer smo uporabili $N = 100$ delcev. Elipsa v slikah (a,c,e,g) prikazuje trenutno MSE oceno stanja igralca. V slikah (b,d,f,h) smo vizualizirali a posteriori porazdelitev delcev s kroglicami nad sliko. Vsaka kroglica prikazuje center elipse stanja (delca), uteži delcev pa so prikazane z višino kroglice; višja ko je kroglica, večja je utež delca.*

Poglavje 5

Razvoj sledilnika za sledenje enega igralca v zaprtem svetu

5.1 Gradnja modela ozadja

Sledenje lahko v splošnem obravnavamo kot problem lokalizacije objekta na podlagi neke informacije, medtem ko je določitev te informacije odvisna od področja uporabe. V našem primeru gre za razvoj aplikacije v delno kontroliranem okolju, ki sledi nekaterim omejitvam ter predpostavkam, katere smo v poglavju 3 podali kot predpostavke zaprtega sveta. Tako je po prvi predpostavki zaprtega sveta (ZS1) kamera statična, kar pomeni, da lahko generiramo model praznega igrišča, ki ga bomo v nadaljevanju imenovali kar *slika/model ozadja*. Po zadnji, četrti, predpostavki (ZS4) osvetljenost igrišča ni časovno konstantna, kar gotovo vpliva na kakovost modela, vendar tudi relativno slab model ozadja nudi vsaj nekaj a priori informacije o njegovem dejanskem izgledu.

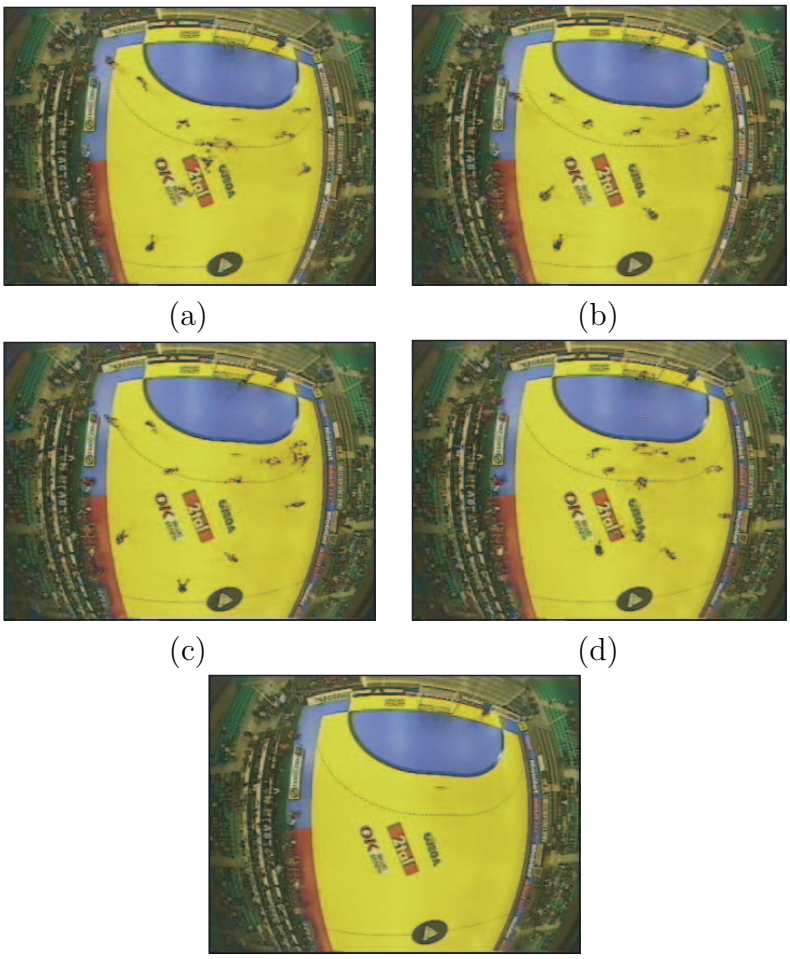
Uporaba slike ozadja v sledenju spada med nizkonivojske operacije na katerih temeljijo mnogi sledilniki (npr. [20, 27, 42, 31, 26, 33, 32]). Sledilnik, ki bo predstavljen v tem poglavju, uporablja sliko ozadja na več različnih nivojih, zato bomo na tem mestu predstavili njeno izgradnjo.

Robusten in večkrat potrjeno uspešen (glej npr. [27, 26, 64]) način izgradnje slike ozadja v posnetkih s statično kamero je pristop s časovno mediano. Postopek poteka tako, da iz sekvence slik $\mathbf{A}_t = \{A_t(\cdot)\}_{t=1}^{T}$ naključno izberemo množico N_s-tih slik $\{S_i(\cdot)\}_{i=1}^{N_s}$ (npr. $N_s = 25$), ter za vsak slikovni element na položaju $\mathbf{u} = (x, y)$, za vsak barvni kanal posebej izračunamo mediano sivinskih vrednosti preko vseh slik v množici. Zgoščeno lahko postopek izračuna slike ozadja $B(\cdot)$ z uporabo časovne mediane zapišemo kot

$$B(\mathbf{u})_z = median[\{S_i(\mathbf{u})_z\}_{i=1}^{N_s}]; \quad z \in \{r, g, b\}$$

kjer smo z **u** označili določen slikovni element, parameter z označuje rdeči, zeleni ali modri barvni kanal, operator $median[\{\cdot_i\}_{i=1}^{N_s}]$ pa je mediana na množici N_s-tih vrednosti.

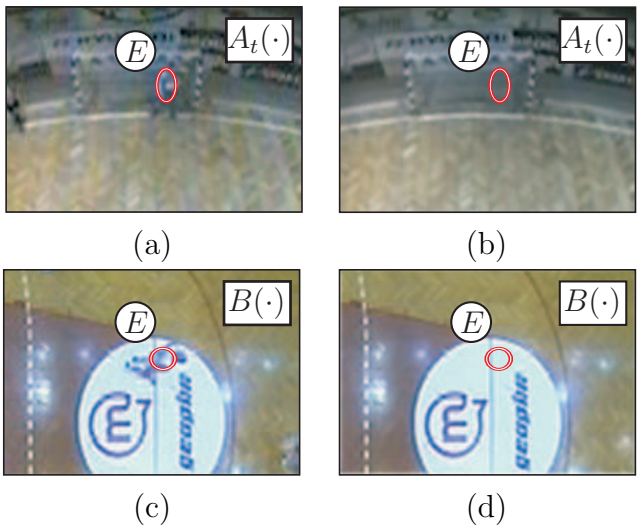
Slika 5.1 prikazuje primere naključno izbranih slik iz video posnetka igre rokometa ter izračunano sliko ozadja z uporabo časovne mediane.



Slika 5.1: *Primeri slik iz sekvence posnetka rokometne tekme (a,b,c,d) ter slika ozadja (e) izračunana s postopkom časovne mediane.*

5.2 Relativna mera prisotnosti tarče

Lokalizacija igralca le na podlagi trenutne slike ter znane igralčeve teksture postane dokaj nerobustna, ko se igralec teksturno[1] bistveno ne razlikuje od njegove okolice. Dva taka primera sta ilustrirana v sliki 5.2, kjer slika 5.2(a) prikazuje slabo vidnega igralca rokometa pred golom, slika 5.2(c) pa belega igralca košake na belem delu igrišča. Oba igralca smo ročno označili z elipsami ter

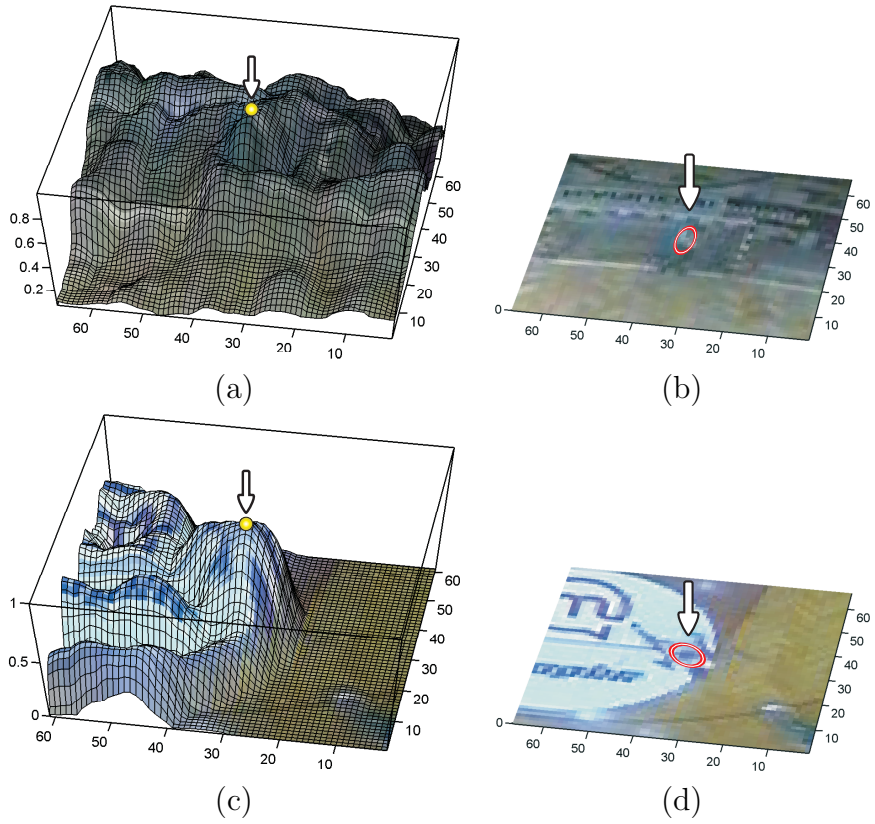


Slika 5.2: *Primer, ko se igralec v trenutni sliki $A_t(\cdot)$ nahaja na teksturno podobnem ozadju (a,c). Igralca sta v obeh slikah (a,c) označena z elipsami E, enaka področja pa so tudi označena na slikah ozadja $B(\cdot)$ (b,d).*

določili njuna referenčna histograma. Nato smo v njuni okolici na regularni mreži vzorčili histograme znotraj enakih elips ter izmerili razdaljo $\varrho(\cdot,\cdot)$ (enačba 4.6) med vsakim vzorčenim ter referenčnim histogramom. Sliki 5.3(a,c) prikazujeta rezultat tega postopka, kjer smo zaradi boljše predstavitve namesto razdalj $\varrho(\cdot,\cdot)$ prikazali vrednosti $1-\varrho(\cdot,\cdot)$; torej višje vrednosti v grafih (sliki 5.3a,c) nakazujejo večjo podobnost z referenčnim histogramom, nižje pa manjšo. Iz slike 5.3(a) je očitno, da je igralec rokometa praktično nerazločljiv od okolice, medtem ko je igralec košarke (slika 5.3c) nekoliko bolj, vendar še vedno slabo.

Lokalizacijo lahko naredimo nekoliko bolj robustno, če imamo na voljo kakšno a priori znanje, kot je npr. slika ozadja. Naj bo $\hat{\mathbf{h}}_t$ referenčni histogram sledene tarče. Znotraj neke elipse E na trenutni sliki $A_t(\cdot)$ vzorčimo histogram $\mathbf{h}_A$, medtem ko na sliki ozadja $B(\cdot)$ vzorčimo znotraj iste elipse histogram $\mathbf{h}_B$. Razdaljo med referenčnim histogramom ter $\mathbf{h}_A$ po enačbi (4.6) označimo z

[1]Kadar se barvni histogram igralca ne razlikuje močno od lokalnih histogramov v njegovi okolici.



Slika 5.3: *Sliki (b,d) prikazujeta dva primera, ko se igralec teksturno ne razlikuje močno od okolice. To dejstvo je ilustrirano v slikah (a,c), kjer so izrisane podobnosti* $(1 - \varrho(\cdot, \cdot))$ *med referenčnim ter lokalnimi histogrami: višja izrisana vrednost pomeni večjo podobnost. Kroglici v slikah (a,c) označujeta položaj igralcev in vidimo, da je precej neizražen.*

$$\hat{\varrho}_A = \varrho(\hat{\mathbf{h}}_t, \mathbf{h}_A),$$

razdaljo med referenčnim ter $\mathbf{h}_B$ pa z

$$\hat{\varrho}_B = \varrho(\hat{\mathbf{h}}_t, \mathbf{h}_B).$$

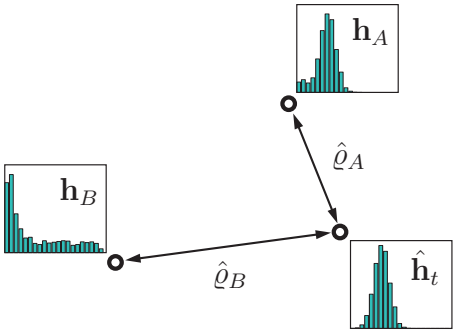
Predpostavimo, da manjša ko je razdalja $\hat{\varrho}_A$, bolj je histogram na trenutni sliki podoben referenčnemu in bolj verjetno se znotraj elipse E nahaja iskana tarča. Vendar, če je razdalja $\hat{\varrho}_B$ med referenčnim histogramom ter ozadjem $\mathbf{h}_B$ prav tako majhna, pa več ne moremo z veliko gotovostjo trditi o prisotnosti tarče, saj je na tem mestu že a priori razdalja do reference majhna.

Če si sedaj predstavljamo referenčni histogram $\hat{\mathbf{h}}_t$ kot točko v prostoru razdalj $(\hat{\varrho}_A, \hat{\varrho}_B)$, (slika 5.4), potem lahko definiramo *relativno* oddaljenost histograma $\mathbf{h}_A$

od $\hat{\mathbf{h}}_t$ z upoštevanjem ozadja $\mathbf{h}_B$ kot

$$D_{rel}(\mathbf{h}_A, \hat{\mathbf{h}}_t; \mathbf{h}_B) = \frac{\varrho(\hat{\mathbf{h}}_t, \mathbf{h}_A)}{\sqrt{\varrho^2(\hat{\mathbf{h}}_t, \mathbf{h}_A) + \varrho^2(\hat{\mathbf{h}}_t, \mathbf{h}_B)}}. \tag{5.1}$$

Zgornjo razdaljo imenujemo relativna razdalja med $\mathbf{h}_A$ in $\hat{\mathbf{h}}_t$, saj upošteva znanje o teksturi na ozadju in se pri konstantnih $\mathbf{h}_A$ in $\hat{\mathbf{h}}_t$ spreminja glede na $\mathbf{h}_B$.

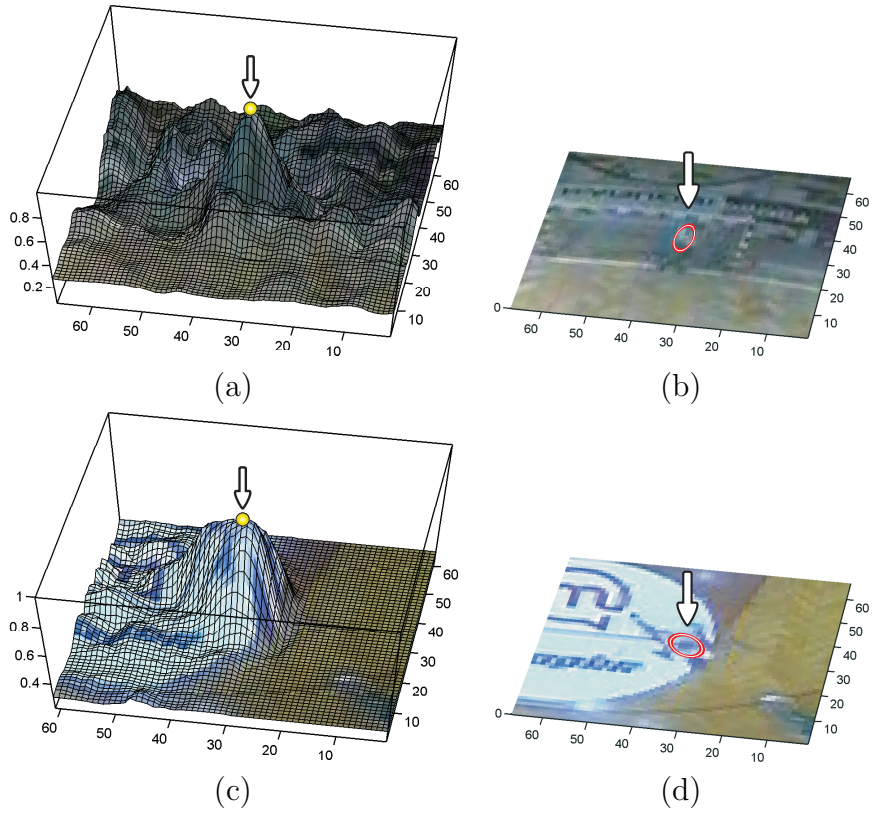


Slika 5.4: *Slika prikazuje tri histograme: referenčni histogram $\hat{\mathbf{h}}_t$, histogram na trenutni sliki $\mathbf{h}_A$, ter histogram na sliki ozadja $\mathbf{h}_B$ so ilustrirani kot točke v prostoru. Razdalje med histogrami $\mathbf{h}_A$ in $\mathbf{h}_B$ ter referenčnim $\hat{\mathbf{h}}_t$ smo označili z $\hat{\varrho}_A$ in $\hat{\varrho}_B$.*

Robustnost nove mere relativne razdalje $D_{rel}(\cdot,\cdot;\cdot)$ je ilustrirana v sliki 5.5, kjer smo zopet ročno označili igralca, določili njegov referenčni histogram ter izračunali vrednosti razdalje $D_{rel}(\cdot,\cdot;\cdot)$ do sosednjih področij na regularni mreži. Zaradi preglednosti smo kakor v sliki 5.3 namesto razdalje $D_{rel}(\cdot,\cdot;\cdot)$ izrisali vrednosti mere podobnosti $1 - D_{rel}(\cdot,\cdot;\cdot)$. Po primerjavi izraženosti vrhov v slikah 5.5 in slikah 5.3 je robustnost mere $D_{rel}(\cdot,\cdot;\cdot)$ očitna, saj sta vrhova v sliki 5.5 precej bolj izražena kot v sliki 5.3.

5.3 Uporaba maske za izločanje ozadja

Relativna razdalja med histogrami omogoča precej robustno lokalizacijo igralca tudi takrat, ko se ta nahaja na teksturno podobni podlagi. Vendar, ker ta razdalja temelji na barvnih histogramih, posledično ne vsebuje prostorske informacije o teksturi igralca in lokalizacija lahko kljub vsemu odpove. Ta problem skušamo omiliti tako, da generiramo binarno masko, ki vsakemu slikovnemu elementu v trenutni sliki dodeli vrednost nič, če ta pripada ozadju, in ena sicer. Klasičen način generiranja binarne maske $M_{Dt}(\cdot)$ je odštevanje slike ozadja $B(\cdot)$ od



Slika 5.5: *Prikaz podobnosti med referenčnim histogramom igralca ter okolico, če upoštevamo še teksturne lastnosti ozadja (a,c). Kroglici v slikah (a,c) prikazujeta položaj igralcev iz slik (b,d). Izraženost vrhov podobnosti je v obeh primerih občutno večja, kot če ne upoštevamo ozadja.*

trenutne slike $A_t(\cdot)$ ter upragovanje te razlike z nekim pragom κ, oziroma:

$$M_{D_t}(\mathbf{u}) = \begin{cases} 1 & ; \quad \|A_t(\mathbf{u}) - B(\mathbf{u})\| \geq \kappa \\ 0 & ; \quad sicer \end{cases}, \tag{5.2}$$

kjer $\|\cdot\|$ označuje l_2 normo in $\mathbf{u}$ položaj nekega slikovnega elementa. Slikovnim elementom $\mathbf{u}$, katerih vrednost maske je $M_D(\mathbf{u}) = 1$ pravimo *vidni*, ostalim pa *nevidni* slikovni elementi. Primer maske, kjer so vidni slikovni elementi označeni z belo barvo, je prikazan v sliki 5.6(e).

Po predpostavkah zaprtega sveta šest in sedem (ZS6,7) barva ter osvetlitev igrišča nista homogeni, osvetlitev pa se še s časom lahko spreminja. To ter dejstvo, da se igralčev izgled prav tako med gibanjem spreminja, onemogoča določitev za vse igrišče enake ter časovno konstantne pragovne vrednosti κ, kakor je to storjeno v npr. [64, 18].

Očitno je pragovna vrednost lastna igralcu ter časovno spremenljiva, zato jo bomo odslej označevali s κ_t, vendar posebej poudarimo, da ta vrednost, kakor tudi maska, pripadata le točno določenemu igralcu in ne vsem.

Če predpostavimo, da je igrišče dovolj homogeno, in je frekvenca zajemanja slik dovolj velika, lahko pragovno vrednost κ_t ob času t ocenimo kot pragovno vrednost dobljeno iz predhodnega časovnega koraka:

$$\kappa'_t = \kappa_{t-1},$$

kjer smo s $(\cdot)'$ poudarili, da gre le za oceno. Ideja je torej ob času $t-1$, po končani iteraciji sledenja, določiti ustrezni prag κ_{t-1} preko znanega položaja objekta v sliki ter ta prag uporabiti pri generiranju nove maske za izločevanje ozadja ob času t.

Določevanje pragovne vrednosti za generiranje maske

Časovno spreminjanje pragovne vrednosti κ_t torej realiziramo preko zaporednega ocenjevanja in korekcije. Sedaj se pojavi vprašanje, kako po končani iteraciji sledenja določiti ustrezno pragovno vrednost, ki bo v naslednjem trenutku služila za novo upragovanje. To storimo tako, da na podlagi ocenjenega položaja objekta določimo tako pragovno vrednost, ki zadosti nekemu predpisanemu kriteriju. S tem se sicer izognemo predpisu pragovne vrednosti, ki je spremenljiva, vendar moramo sedaj določiti kriterij, ki temelji na takih lastnostih igralca, ki so na nek način manj občutljive na spremenljivost osvetlitve in barve.

Z opazovanjem elips igralcev, ko so se ti gibali na njim podobnem ozadju smo opazili, da elipse vedno vsebujejo nek delež slikovnih elementov, ki pripadajo ozadju ali pa pripadajo igralcem, vendar so zelo podobni ozadju. Predpostavimo, da se v takih situacijah znotraj elips igralcev vedno nahaja nek delež η_0 slikovnih elementov, ki bi jih lahko pripisali ozadju.

Če torej ob času $t-1$ poznamo položaj igralca na igrišču in ga opišemo z neko elipso E, potem je ocena pragovne vrednosti v novem časovnem trenutku t tak $\kappa'_t(\eta_0)$, ki v trenutku $t-1$ generira tako masko $M_{D_{t-1}}(\cdot)$, da se znotraj elipse E nahaja delež η_0 slikovnih elementov z vrednostjo $M_{D_{t-1}}(\mathbf{u}) = 0$.

Postopek določevanja pragovne vrednosti je sledeč: Naj $\mathbf{d}_E = \{d_{iE}\}_{i=1}^{m_D}$ označuje histogram diferenc z m_D celicami med trenutno sliko $A_t(\cdot)$ in modelom ozadja $B(\cdot)$ znotraj elipse E. V histogramu $\mathbf{d}_E$ je i-ta celica definirana kot

$$d_{iE} = \frac{1}{a_E} \sum_{\mathbf{u} \in E} \delta_i(b(\mathbf{u})), \tag{5.3}$$

kjer je a_E število vseh slikovnih elementov v elipsi E in je $b(\mathbf{u})$ funkcija razdalje med slikovnimi elementi $\mathbf{u}$ na modelu ozadja in trenutni sliki

$$b(\mathbf{u}) = \|A_t(\mathbf{u}) - B(\mathbf{u})\|.$$

Na podlagi histograma razdalj $\mathbf{d}_E$ izračunamo kumulativni histogram $\mathbf{c}_E = \{c_{iE}\}_{i=0}^{m_C-1}$, kjer je i-ta celica definirana kot

$$c_{iE} = \sum_{j=0}^{i} d_{jE}. \tag{5.4}$$

Iskana pragovna vrednost $\kappa_t(\eta_0)$ je tisti najmanjši i, pri katerem vrednost celice $c_{(i+1)E}$ v kumulativnem histogramu preseže predpisano vrednost parametra nevidnosti η_0:

$$\kappa_t(\eta_0) = i \quad ; \quad c_{iE} < \eta_0 \leq c_{(i+1)E}. \tag{5.5}$$

Določanje pragovne vrednosti smo ilustrirali s primerom 5, kjer je izbrana vrednost parametra $\eta_0 = 0.25$.

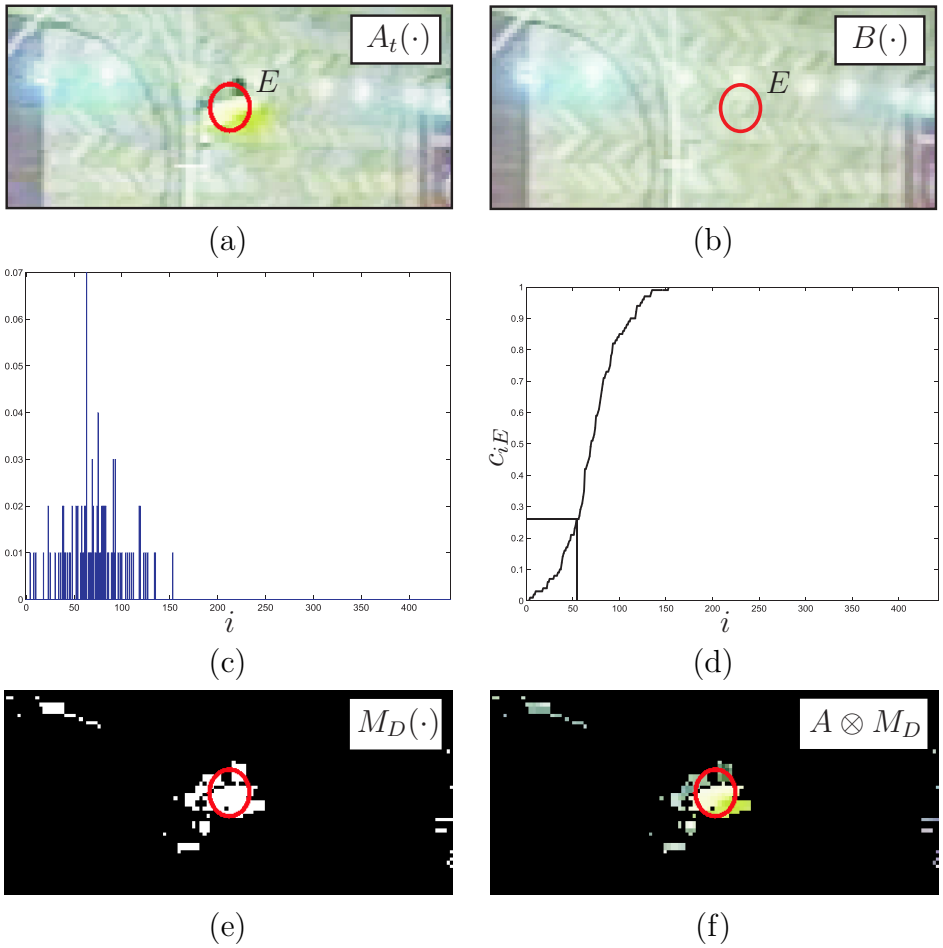
Primer 5. *Recimo, da ob času $t-1$ poznamo stanje igralca, ki ga opisuje elipsa E (slika 5.6a) in model ozadja (slika 5.6b). Želimo določiti tak prag $\kappa_{t-1}(\eta_0)$, da bo znotraj elipse E 25% nevidnih slikovnih elementov ($\eta_0 = 0.25$); iščemo torej $\kappa_t(0.25)$. Najprej zgradimo histogram diferenc $\mathbf{d}_E$, kjer je število celic $m_D = 444$ (slika 5.6c) in kumulativni histogram $\mathbf{c}_E$ (slika 5.6d). Iz slednjega odčitamo indeks celice, katere vsebina preseže vrednost $\eta_0 = 0.25$ in dobimo pragovno vrednost $\kappa_{t-1}(0.25) = 55$. S tako dobljenim pragom generirano masko ob času $t-1$ prikazuje slika 5.6(e), medtem ko slika 5.6(f) prikazuje maskirano sliko igralca.*

Kdaj generirati masko?

Generiranje maske potrebujemo predvsem v situacijah, ko je igralec slabo razločljiv od ozadja, oziroma le v primerih, ko je razdalja $\varrho(\mathbf{h}_A, \mathbf{h}_B)$ (enačba 4.6) med histogramom $\mathbf{h}_A$ znotraj elipse ocenjenega stanja igralca E na trenutni sliki $A_t(\cdot)$ in histogramu $\mathbf{h}_B$ znotraj iste elipse na modelu igrišča $B(\cdot)$ dovolj majhna. Pragovno vrednost razdalje ϱ_T nad katero upragovanje ni potrebno smo določili tako, da smo ročno označili 119 igralcev, ko so se ti nahajali po našem mnenju na njim podobnem ozadju in izračunali razdalje med njimi ter ozadjem. Pragovno razdaljo ϱ_T smo ocenili kot povprečno vrednost teh razdalj:

$$\varrho_T = 0.8.$$

S poskušanjem smo ugotovili, da te igralce dobro ločimo od ozadja takrat, ko generiramo tako masko, da se znotraj elipse igralca nahaja približno ena četrtina slikovnih elementov, ki jih maska pripiše ozadju; torej, $\eta_0 = 0.25$.



Slika 5.6: *Primer določanja pragovne vrednosti* $\kappa_{t-1}(\eta_0)$ *pri vrednosti parametra* $\eta_0 = 0.25$. *Igralca ter njegovo elipso E prikazuje slika (a), model ozadja slika (b), histogram diferenc znotraj elipse E prikazuje slika (c), kumulativni histogram diferenc pa je prikazan v sliki (d). Pri vrednosti* $c_{iE} < 0.25 \leq c_{(i+1)E}$ *odčitamo vrednost* $i = 55$ *in dobimo prag* $\kappa_{t-1}(0.25) = 55$. *Primer s tem pragom upragovane slike razlike ob času* $(t-1)$ *prikazuje (e), sliko igralca s superponirano masko pa (f). V naslednjem časovnem koraku, torej generiramo masko z ocenjeno pragovno vrednostjo* $\kappa'_t(0.25) = 55$.

Povzetek generiranja maske

Za konec povzemimo postopek generiranja maske. Po iteraciji sledenja ob času $t-1$ ocenimo elipso E trenutnega stanja igralca. Znotraj elipse E izračunamo histogram $\mathbf{h}_A$ na trenutni sliki $A_t(\cdot)$ in histogram $\mathbf{h}_B$ na modelu ozadja $B(\cdot)$. Če njuna razdalja $\varrho(\mathbf{h}_A, \mathbf{h}_B)$ (enačba 4.6) preseže vrednost $\varrho_T = 0.8$, potem v naslednjem časovnem koraku t maske ne generiramo ($\kappa'_{t+1}(0.25) = 0$), sicer pa novi prag ocenimo preko trenutnega kumulativnega histograma diferenc in postavimo $\kappa'_{t+1}(0.25) = \kappa_t(0.25)$.

5.4 Prilagodljiv barvni model igralca

Pred začetkom sledenja določimo barvni model ali referenčni histogram igralca tako, da ga ročno označimo. Sledenje v vseh naslednjih slikah se potem prevede na problem lokalizacije s tem referenčnim histogramom in uspešnost lokalizacije je neposredno odvisna od tega, kako dobro referenčni histogram opisuje igralčev trenutni izgled.

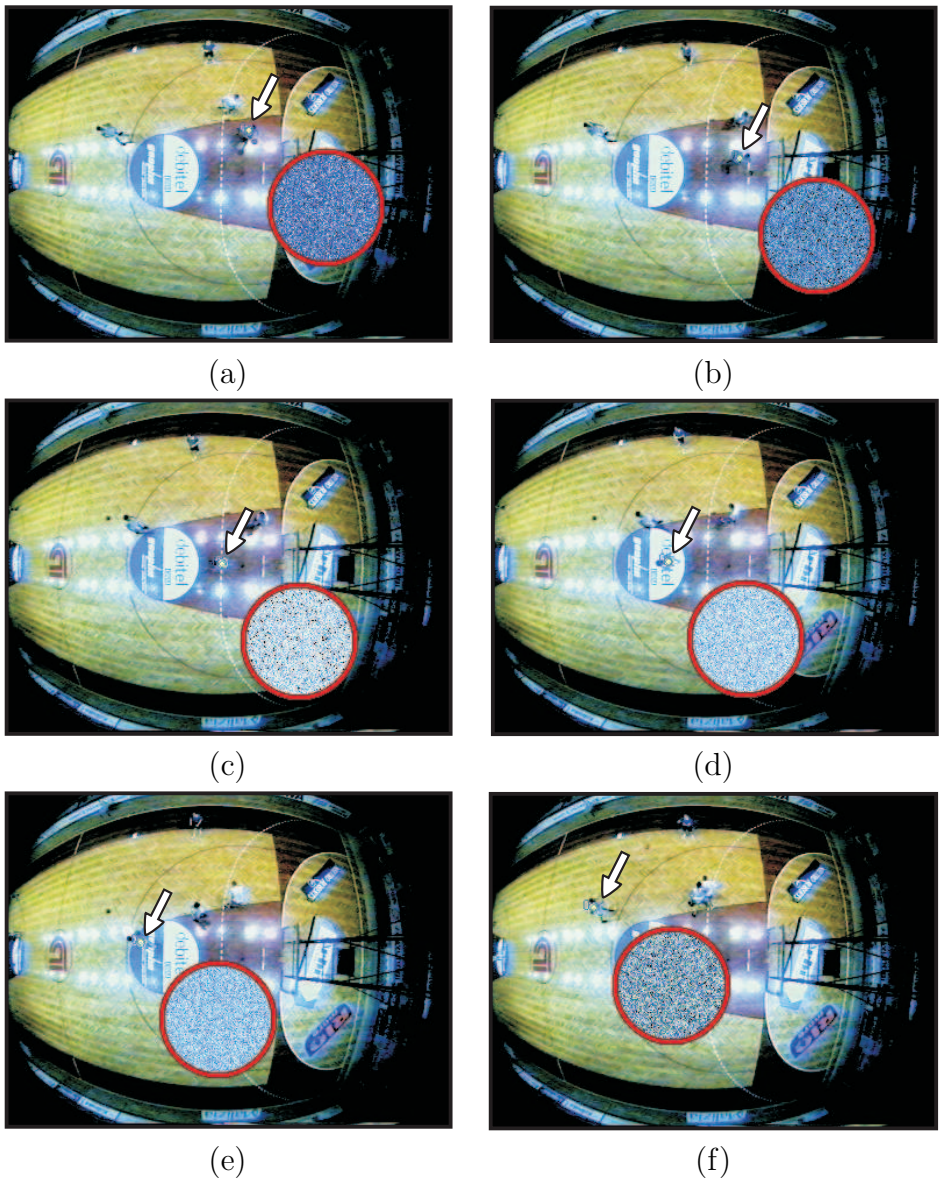
Sedaj se pojavi vprašanje, ali je smiselno vseskozi sledenje uporabljati nespremenljiv referenčni histogram. Dejstvo je, da igralčeva tekstura ponavadi ni enaka na različnih delih njegovega dresa (npr. nosi črno majico z belo številko na hrbtu), zato se njegov izgled nekoliko spreminja že samo pri nagibanju. Poleg tega je osvetlitev igrišča časovno ter krajevno neenakomerna (sedma predpostavka, ZS7) in zato sklepamo, da se resnični igralčev izgled, oziroma histogram, že a priori s časom spreminja (šesta predpostavka, ZS6). To spremenljivost igralčevega izgleda smo prikazali v sliki 5.7, kjer so vizualizirani trenutni histogrami igralca na različnih mestih pri preteku igrišča.

Upoštevajoč rezultate v slikah 5.7 zaključimo, kakor mnogo avtorjev pred nami [40,65,66,67,36,44], da je potrebno model tarče, t.j. referenčni histogram igralca, s časom adaptirati. Naj $\hat{\mathbf{h}}_t$ označuje referenčni histogram s katerim ob času t lokaliziramo nekega igralca. Po končani iteraciji sledenja ob času t ocenimo novo stanje igralca ter znotraj elipse tega stanja na trenutni sliki vzorčimo histogram $\mathbf{h}_t$. Novi referenčni histogram za lokalizacijo igralca ob času $t+1$ označimo s $\hat{\mathbf{h}}_{t+1}$ in kakor v [40, 44] sledi enačbi za adaptacijo

$$\hat{\mathbf{h}}_{t+1} = (1-\alpha)\hat{\mathbf{h}}_t + \alpha\mathbf{h}_t, \tag{5.6}$$

kjer α predstavlja faktor s katerim trenutni vzorčeni histogram igralca $\mathbf{h}_t$ vpliva na novi model $\hat{\mathbf{h}}_{t+1}$.

Uspešnost sledenja je precej odvisna od parametra α: vrednost $\alpha = 0$ pomeni "brez adaptacije", kar po peti in šesti predpostavki ZS(5,6), ki govorita o spremenljivosti izgleda, pomeni a priori slabo sledenje; prav tako je neprimerna sto odstotna adaptacija ($\alpha = 1$), saj sledenje postane preveč občutljivo na



Slika 5.7: *Vizualizacija histograma igralca pri teku preko igrišča. Slike prikazuejo igralca na igrišču (označeno z belo puščico) in vzorec z enakim histogramom kot ga ima igralec v trenutni sliki. V vsaki sliki smo igralca ročno označili in posneli njegov barvni histogram. Vzorec smo potem generirali tako, da smo barvni histogram tretirali kot porazdelitev in jo vzorčili z inverzno metodo (Dodatek B). Časovno si slike sledijo: (a,b,c,d,e,f)*

morebitne kratkotrajne motnje. Poleg tega je referenčni histogram smiselno bolj adaptirati takrat, ko se igralec zares nahaja v elipsi trenutno ocenjenega stanja. V poglavju 4.1 smo predstavili mero prisotnosti tarče, ki upošteva sliko ozadja, in to mero lahko uporabimo za vodilo intenzivnosti adaptacije referenčnega histograma.

Recimo, da ob času t ocenimo elipso trenutnega stanja igralca. Znotraj te elipse na trenutni sliki $A_t(\cdot)$ vzorčimo histogram $\mathbf{h}_t$, in na sliki ozadja $B(\cdot)$ histogram $\mathbf{h}_B$. Časovno spremenljiv faktor adaptacije z upoštevanjem prisotnosti igralca zapišemo kot

$$\alpha_t = \alpha_{max} \cdot (1 - D_{rel}(\mathbf{h}_t, \hat{\mathbf{h}}_t; \mathbf{h}_B)), \tag{5.7}$$

kjer je $D_{rel}(\mathbf{h}_t, \hat{\mathbf{h}}_t; \mathbf{h}_B)$ relativna razdalja med referenčnim in vzorčenim histogramom (enačba 5.1), α_{max} določa največjo možno vrednost adaptacije, enačba časovno spremenljive adaptacije referenčnega histograma, ki upošteva prisotnost igralca pa se potem glasi

$$\hat{\mathbf{h}}_{t+1} = (1 - \alpha_t)\hat{\mathbf{h}}_t + \alpha_t \mathbf{h}_t. \tag{5.8}$$

Najvišja možna adaptacija referenčnega histograma znotraj enega časovnega trenutka je po enačbi (5.7) odvisna od parametra α_{max}, ki je konstanten vseskozi sledenje. Zadnji korak pri modeliranju spremenljive adaptacije je določitev primerne vrednosti tega parametra.

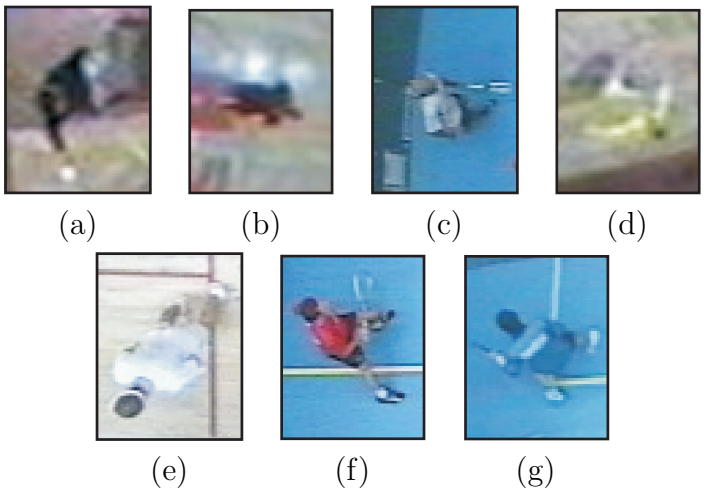
5.4.1 Določanje parametra maksimalne adaptacije

Z referenčnim sledilnikom iz poglavja 4 smo posledili sedem igralcev rokometa in squasha (slika 5.8). V izbranih sekvencah so se igralci gibali na njim dovolj različnem ozadju in med njimi ni prihajalo do trkov ali zakrivanj, tako da so bile spremembe v izgledu le posledica nagibanja in premikanja. Na tak način smo uspeli dobiti veliko množico dobro ocenjenih stanj (približno 500 stanj na igralca). Vsa stanja smo še vizualno preverili ter ugotovili, da sledenje v izbranih sekvencah ni nikoli odpovedalo. Za vsakega igralca smo nato na podlagi dobljenih stanj posneli sekvenco histogramov na trenutnih slikah ter ozadju; sekvenco K_i-tih parov histogramov i-tega igralca označimo s

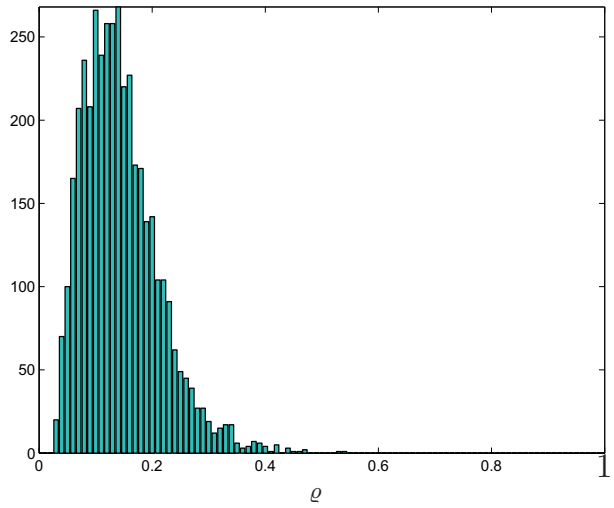
$$\mathbf{h}^{(i)}_{1:K_i} = \{\mathbf{h}^{(i)}_{A_k}, \mathbf{h}^{(i)}_{B_k}\}_{k=1}^{K_i}.$$

Vsem igralcem smo na podlagi zajetih histogramov določili povprečne histograme. Povprečni histogram i-tega igralca $\tilde{\mathbf{h}}^{(i)} = \{\tilde{h}^{(i)}_j\}_{j=1}^{K_i}$ je definiran s povprečjem trenutnih histogramov

$$\tilde{h}^{(i)}_j = \frac{1}{K_i} \sum_{k=1}^{K_i} h^{(i)}_{A_k j}, \tag{5.9}$$



Slika 5.8: *Slike igralcev rokometa in squasha uporabljenih za določitev parametra adaptacije.*



Slika 5.9: *Histogram relativnih razdalj trenutnih histogramov igralcev od njihovih povprečnih histogramov.*

kjer je $h^{(i)}_{A_{kj}}$ j-ta celica k-tega histograma v sekvenci histogramov i-tega igralca. Za vsakega igralca smo simulirali časovno adaptacijo z algoritmom 5.1. Na vsakem koraku smo določili razdaljo $\varrho(\cdot,\cdot)$ (enačba 4.6) med trenutnim referenčnim ter povprečnim histogramom in tako dobili množico razdalj. Porazdelitev dobljenih diferenc $\varrho(\cdot,\cdot)$ vseh igralcev pri popolni adaptaciji ($\alpha_t = 1$) prikazuje slika 5.9.

Na varianco v tej porazdelitvi lahko gledamo kot na *šumenje* razdalj med referenčnimi ter povprečnimi histogrami. Ker želimo določiti tak parameter

Vhod:

- Sekvenca parov histogramov igralca $\mathbf{h}_{1:K} = \{\mathbf{h}_{A_k}, \mathbf{h}_{B_k}\}_{k=1}^{K}$
- Povprečni histogram $\tilde{\mathbf{h}}$
- Parameter maksimalne adaptacije α_{max}

Izhod:

- Sekvenca diferenc med trenutnimi referenčnimi ter povprečnim histogramom $\mathbf{\Theta}_{1:K}$

1. Inicializiraj: $\hat{\mathbf{h}}_2 = \mathbf{h}_{A_1}$, $\mathbf{\Theta}_1 = \{\varrho(\hat{\mathbf{h}}_2, \tilde{\mathbf{h}})\}$
2. For $k = 2 : K$
   - $\alpha_k = \alpha_{max}(1 - D_{ref}(\hat{\mathbf{h}}_k, \mathbf{h}_{A_k}; \mathbf{h}_{B_k}))$
   - $\hat{\mathbf{h}}_{k+1} = (1 - \alpha_k)\hat{\mathbf{h}}_k + \alpha_k \mathbf{h}_{A_k}$
   - $\mathbf{\Theta}_{1:k} = \{\mathbf{\Theta}_{1:k-1}, \varrho(\hat{\mathbf{h}}_k, \tilde{\mathbf{h}})\}$

   End for

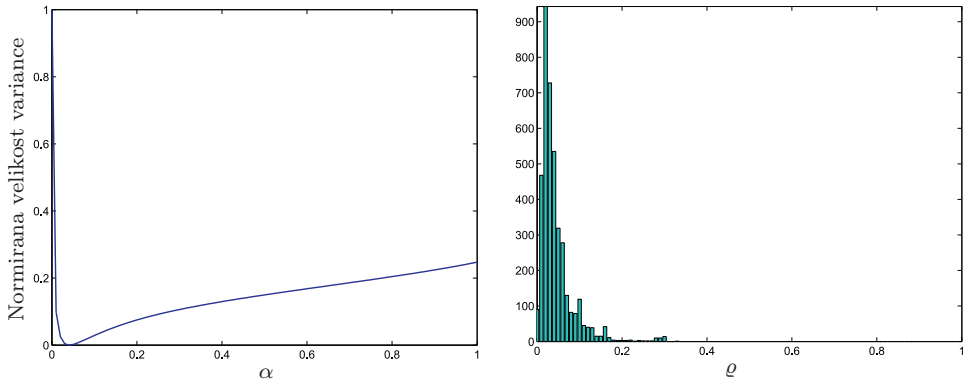
Algoritem 5.1: *Algoritem za simulacijo adaptacije refrenčnega histograma enega igralca in izračun trenutnih diferenc med referenčnim ter povprečnim histogramom.*

maksimalne adaptacije α_{max}, da bodo referenčni histogrami čim bolj robustni glede na šumenje, t.j. da se bodo adaptirali na počasne spremembe, smo za α_{max} izbrali tisto vrednost, ki je minimizirala varianco v porazdelitvi diferenc vseh igralcev. Odvisnost variance od parametra α_{max} prikazuje slika 5.10(a), v kateri vidimo, da je bila varianca minimizirana pri približno 5% adaptaciji ($\alpha_{max} = 0.05$). Porazdelitev diferenc vseh igralcev pri taki adaptaciji je prikazana v sliki 5.10(b).

Z naslednjim primerom bomo skušali nekoliko osvetliti pomen vrednosti parametra maksimalne adaptacije α_{max}. Zaradi enostavnosti bo primer temeljil na enačbi konstantne vrednosti adaptacije (enačba 5.8), kjer smo vrednost parametra konstantne adaptacije α določili kot povprečno vrednost parametra α_t, ki smo jo izmerili v zgornjem poskusu pri $\alpha_{max} = 0.05$, in je znašala $\alpha = 0.045$.

Primer 6. *Denimo, da ob času $t = 0$ poznamo referenčni model igralca $\hat{\boldsymbol{h}}_0$. V naslednjem trenutku prižgemo barvne luči in opazujemo igralca. Histogram novega izgleda označimo s $\boldsymbol{h}_C$ in naj bo konstanten za vsak $t > 0$. Model $\hat{\boldsymbol{h}}_t$ se prične adaptirati na trenutni histogram $\boldsymbol{h}_C$ po enačbi*

$$\hat{\boldsymbol{h}}_t = (1 - \alpha)\hat{\boldsymbol{h}}_{t-1} + \alpha \boldsymbol{h}_C.$$



(a) (b)

Slika 5.10: *Graf odvisnosti variance diferenc od parametra maksimalne adaptacije α_{max} (a). Vrednosti v grafu so normalizirane na interval $[0, 1]$, minimum pa graf doseže pri $\alpha_{max} = 0.045$, kar pomeni pet procentno adaptacijo ($\alpha_{max} \approx 5\%$). Porazdelitev diferenc pri taki vrednosti maksimalne adaptacije prikazuje (b).*

Ker sta α in $\boldsymbol{h}_C$ konstantna, lahko referenčni histogram ob času t zapišemo kot

$$\hat{\boldsymbol{h}}_t = (1-\alpha)^t \hat{\boldsymbol{h}}_0 + (1-(1-\alpha)^t)\boldsymbol{h}_C. \tag{5.10}$$

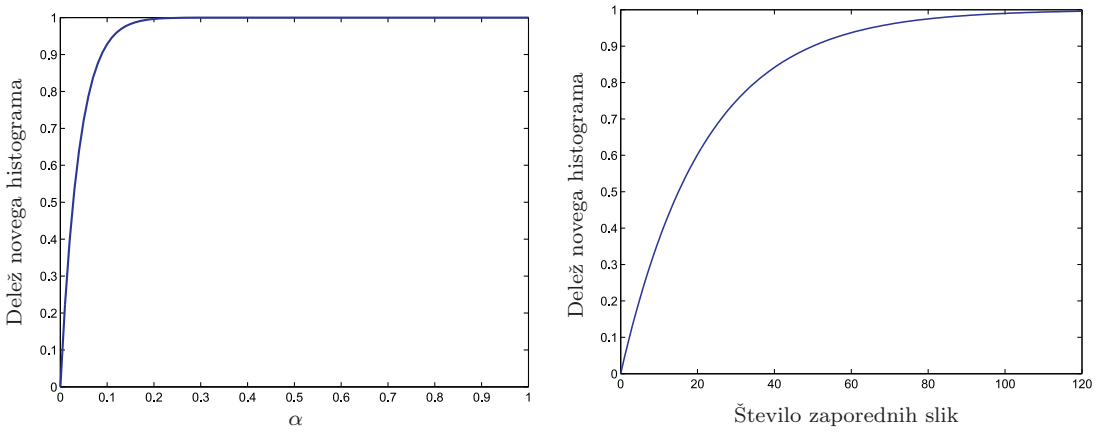
Recimo, da nas zanima delež novega histograma $\boldsymbol{h}_C$ v referenčnem histogramu $\hat{\boldsymbol{h}}_t$ po 25-ih zaporednih slikah[2] pri parametru adaptacije $\alpha = 0.045$. Po enačbi (5.10) sledi

$$\begin{aligned} \hat{\boldsymbol{h}}_t &= (1-0.045)^{25}\hat{\boldsymbol{h}}_0 + (1-(1-0.045)^{25})\boldsymbol{h}_C \\ &\approx 0.32\hat{\boldsymbol{h}}_0 + 0.68\mathbf{h}_C, \end{aligned}$$

kar pomeni, da bo po 25-ih korakih referenčni histogram $\hat{\boldsymbol{h}}_t$ vseboval približno eno tretjino referenčnega histograma ob $t = 0$, $\hat{\boldsymbol{h}}_0$, in dve tretjini novega histograma $\mathbf{h}_C$.

Delež histograma $\mathbf{h}_C$ v $\hat{\boldsymbol{h}}_t$ po 25-ih slikah v odvisnost od parametra α prikazuje slika 5.11(a), kjer vidimo, da je pri vrednosti $\alpha = 0.2$ po 25-ih zaporednih slikah referenčni histogram praktično enak trenutnem histogramu; pri tej vrednosti adaptacije se torej referenčni histogram v eni sekundi popolnoma prilagodi na stopničasto spremembo. Slika (b) prikazuje delež novega histograma pri konstantni adaptaciji $\alpha = 0.045$ v odvisnosti od časa. Vidimo, da se pri $\alpha = 0.045$ referenčni histogram adaptira na nenadno spremembo v 120 časovnih korakih, kar znaša približno 5 sekund.

[2] Trajanje 25-ih zaporednih slik je v vseh naši poskusih ekvivalentno eni sekundi posnetka.



(a) (b)

Slika 5.11: *Delež novega histograma v referenčnem po 25-ih slikah v odvisnost od parametra α_{max} (a) in delež novega histograma v referenčnem pri vrednosti parametra $\alpha_{max} = 0.04$ v odvisnosti od števila zaporednih slik (b).*

5.5 Nova funkcija verjetja stanj

V tem poglavju bomo predstavili novo funkcijo verjetja stanj, katero bomo uporabili za sledenje igralcev v zaprtem svetu.

Recimo, da barvni histogram $\hat{\mathbf{h}}_t$ predstavlja vizualni model sledenega igralca. Ob času t filter delcev generira neko stanje $\mathbf{x}_t$ in znotraj elipse E tega stanja opravimo meritev; npr. vzorčimo histogram $\mathbf{h}_A$ na trenutni sliki ter histogram $\mathbf{h}_B$ na sliki ozadja. Distančna funkcija med modelom in meritvijo je lahko relativna razdalja med histogrami, ki smo jo določili v enačbi (5.1). Vendar, ker smo v poglavju 5.3 določili postopek za dinamično generiranje maske, lahko tudi to informacijo uporabimo v distančni funkciji; na primer, če se znotraj elipse E na maski nahaja le malo *vidnih* slikovnih elementov, potem manj verjamemo v prisotnost igralca v stanju $\mathbf{x}_t$, kot če se jih veliko. Distančno funkcijo $z(\mathbf{x}_t)$, ki jo imenujemo tudi mera prisotnosti, zato definiramo kot

$$z(\mathbf{x}_t) \quad = \quad \beta^{-1} \cdot D_{rel}(\mathbf{h}_A, \hat{\mathbf{h}}_t; \mathbf{h}_B), \tag{5.11}$$

kjer parameter β določa delež *vidnih* slikovnih elementov znotraj elipse E v maski $M(\cdot)$ in sledi enačbi

$$\beta = \frac{1}{a_E} \sum_{\mathbf{u} \in E} M(\mathbf{u}), \tag{5.12}$$

kjer je a_E število vseh slikovnih elementov v elipsi E in $\mathbf{u}$ poljuben slikovni element v E.

5.5.1 Določevanje pdf mere prisotnosti

Za določevanje uteži stanj z uporabo mere $z(\cdot)$ (enačba 5.11) moramo najprej določiti model njene pdf. Ker fizikalnega modela generiranja vrednosti te mere ne poznamo, smo se poslužili empiričnega pristopa z načrtovanim poskusom.

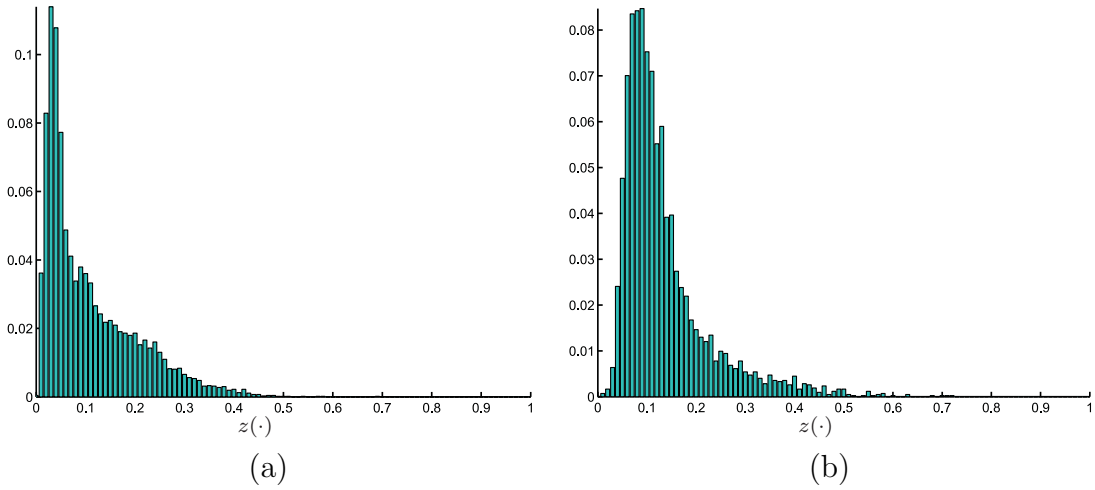
Če gledamo na igralce in njihova stanja v posnetkih kot na generatorje vrednosti mere $z(\cdot)$, potem je smiselno poskus načrtovati tako, da zajamemo neke tipične globalne lastnosti teh generatorjev. Poglavitna lastnost je npr. kolik delež tekme se igralec v povprečju giblje z določeno hitrostjo. Ta lastnost na prvi pogled nima nikakršne zveze z vizualnimi lastnostmi, ki generirajo mero prisotnosti, vendar če opazujemo izgled igralca v zaporednih slikah, ugotovimo, da si je igralec bolj podoben v zaporedju slik kadar stoji na miru ali se giblje počasi, kot takrat ko hitro teče.

V [68] so avtorji analizirali gibanje igralcev rokometa in ugotovili, da igralci približno štirideset odstotkov tekme pretečejo zelo hitro, ostalih šetdeset odstotkov njihovega gibanja pa predstavlja počasno premikanje. Poleg tega avtorji še ugotavljajo, da ta lastnost velja tudi za sorodne športne igre kot sta npr. nogomet in košarka ter pri tem navajajo raziskave drugih avtorjev.

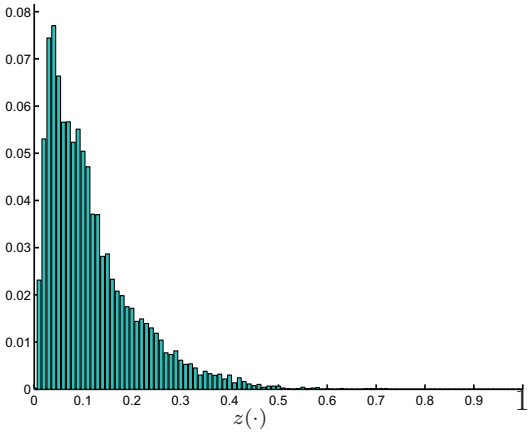
Na podlagi teh ugotovitev smo poskus ločili v dva dela. Najprej smo sledili šestim igralcem rokometa, ko so ti stali na mestu, potem pa še petim igralcem rokometa in trem squasha, ko so se gibali hitro. Ker referenčni sledilnik iz poglavja 4 ni bil sposoben slediti tem sekvencam, smo ga razširili z generiranjem maske za izločevanje ozadja (poglavje 5.3). S tako spremenjenim sledilnikom smo sledili igralce in izbrali sekvence, ko sledenje ni odpovedalo. Med sledenjem smo beležili vrednosti mere $z(\hat{\mathbf{x}}_t)$ na ocenjenih povprečnih stanjih $\hat{\mathbf{x}}_t$. Na tak način smo posneli dve množici meritev: 4239 vrednosti mere pri hitrem gibanju igralcev ter 7300 vrednosti, ko so igralci stali na mestu; število meritev v eni in drugi množici smo izbrali v skladu z zgornjimi ugotovitvami o gibanju igralcev in so približno v razmerju $4 : 6$. Histograme vrednosti mere $z(\hat{\mathbf{x}}_t)$ obeh množic prikazuje slika 5.12.

Slika 5.12(b) prikazuje porazdelitev mere, ko so se igralci gibali hitro, in kot smo pričakovali, je modus porazdelitve očitno izmaknjen iz vrednosti nič. Vzrok temu pojavu je lahko to, da pri hitrem gibanju obstaja majhna možnost da ostane igralčev izgled v dveh zaporednih slikah enak, in večja možnost da se nekoliko spremeni. Enak, vendar občutno manjši efekt izmaknjenosti porazdelitve opazimo pri porazdelitvi mere v sliki 5.12(a), ko so igralci stali na mestu. Do te izmaknjenosti po našem mnenju v glavnem pride zaradi šumenja v sliki, spremenljivosti segmentacije z uporabo maske ter zaradi premikanja igralca na mestu.

Kot smo že omenili, razmerje med številom meritev v prvi ter drugi množici odgovarja razmerju med počasnim ter hitrim gibanjem igralcev med tekmo [68].



Slika 5.12: *Histograma predstavljata empirično dobljeno porazdelitev mere prisotnosti med stojo na mestu (a) in hitrim gibanjem (b) igralcev rokometa ter squasha.*



Slika 5.13: *Histogram prikazuje porazdelitev mere, kjer smo združili meritve pri hitrem gibanju ter stanjem na mestu in modelira globalno porazdelitev mere med tipično tekmo.*

Obe množici meritev smo zato združili v eno množico, ki jo bomo v nadaljevanju imenovali *skupne meritve*, ter zgradili skupni histogram (slik 5.13). Ta histogram predstavlja globalno porazdelitev mere, ki jo pričakujemo pri sledenju igralcev med tipično tekmo, in zapis te porazdelitve s parametričnim modelom je iskana funkcija verjetja.

Izbira parametričnega modela skupnega histograma

Ker zopet nimamo močnega fizikalnega ozadja na katerega bi se oprli pri izbiri parametričnega modela skupnega histograma, smo izbrali štiri poskusne modele:

I. Eksponentna porazdelitev: $f(x;b) = \frac{1}{b}e^{-\frac{x}{b}}$

II. Gama porazdelitev: $f(x;a,b) = \frac{1}{b^a\Gamma(a)}x^{a-1}e^{-\frac{x}{b}}$

III. Inverzna gama porazdelitev: $f(x;a,b) = \frac{b^a}{\Gamma(a)}x^{-a-1}e^{-\frac{b}{x}}$

IV. Normalna porazdelitev s srednjo vrednostjo nič: $f(x;b) = \frac{1}{b\sqrt{2\pi}}e^{-\frac{x^2}{2b^2}}$

Slavni rek Georgea Boxa[3] pravi: "Vsi modeli so sicer napačni, vendar pa so nekateri med njimi uporabni." [69]. Na voljo imamo torej nekaj modelov med katerimi nobeden ni nujno pravi, želimo pa izbrati tistega, ki kar najbolje pojasni podatke. Drugače rečeno, z izbiro modela skušamo minimizrati izgubo informacije pri modeliranju podatkov. Klasični prijem pri takih problemih je uporaba Akaikejevega informacijskega kriterija (AIC) [70], ki združuje princip največjega verjetja *(angl. maximum likelihood)* in Kullback-Leiblerjeve informacije [71], ter hkrati vsebuje intuicijo Occamove želetke *(angl. Occam razor)*. Podrobnejšo razpravo o tem kriteriju in uporabi v problemih izbire modelov najdemo npr. v [69], tu pa bomo navedli le najpomembnejše enačbe za izbiro modela in njihov pomen. Enačba za izračun AIC se glasi

$$AIC = -2log(\mathcal{L}(\hat{\theta}|podatki)) + 2K, \tag{5.13}$$

kjer je $\mathcal{L}(\hat{\theta}|podatki)$ verjetje modela glede na podatke, s $\hat{\theta}$ označimo oceno parametrov modela po *postopku največjega verjetja* MLE *(angl. maximum likelihood estimate)*, K pa je število ocenjenih parametrov v modelu.

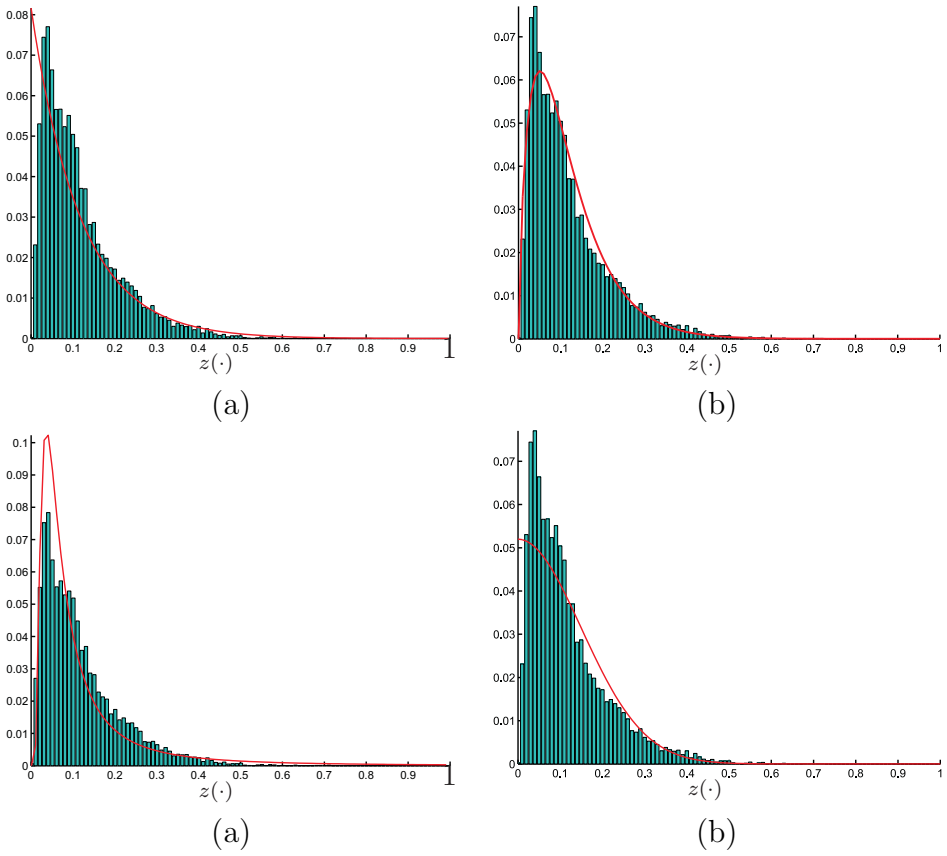
Primerjava modelov poteka tako, da za vsak i-ti model izračunamo Δ_i po enačbi

$$\Delta_i = AIC_i - AIC_{min},$$

kjer je AIC_i vrednost AIC i-tega modela, medtem ko je AIC_{min} najnižja vrednost AIC med vsemi modeli. Na podlagi teh vrednosti izračunamo Akaikejeve uteži w_{AIC_i} po enačbi

$$w_{AIC_i} = \frac{e^{-\frac{\Delta_i}{2}}}{\sum_j e^{-\frac{\Delta_j}{2}}}. \tag{5.14}$$

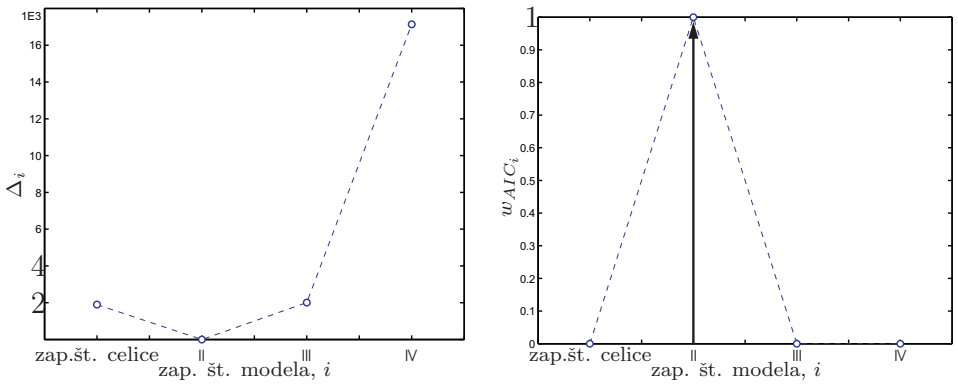
[3]George Box (1919-) je eden od najbolj vplivnih avtorjev na področju eksperimentalne statistike in načrtovanja poskusov.



Slika 5.14: *Slike prikazujejo histograme vrednosti mere prisotnosti ter grafe različnih modelov pri MLE ocenah parametrov. Grafi si sledijo: model I, (a); model II, (b); model III, (c); model IV (d).*

Te uteži predstavljajo verjetnost, da je i-ti model pravi model med vsemi izbranimi; torej, i-ti model je z verjetnostjo w_{AIC_i} najboljši model med vsemi preizkušanimi modeli.

Vsem štirim modelom porazdelitve smo po postopku *postopek največjega verjetja* (MLE) določili vrednosti parametrov na podlagi skupnih meritev distančne funkcije $z(\mathbf{x}_t)$. Pri teh parametrih smo za vsak i-ti model izračunali Δ_i ter utež w_{AIC_i}. Slike histogramov skupnih meritev ter grafe izračunanih modelov prikazuje slika 5.14, kjer že na pogled opazimo, da se gama porazdelitev najbolje prilega histogramu. To izbiro še potrdimo na podlagi grafov vrednosti Δ_i in w_{AIC_i} v sliki 5.15, iz katerih razberemo, da ima model z gama porazdelitvijo utež w_{AIC_2} praktično ena.



(a) (b)

Slika 5.15: *Vrednosti Δ_i v grafu (a) nakazujejo, da je izbira modela II (gama porazdelitev) najbolj primerna. To potrdijo tudi AIC uteži (b), kjer vidimo, da obstaja skoraj 100% šansa, da je model II najbolj primeren med vsemi štirimi modeli.*

Na podlagi rezultatov v slikah 5.14 in 5.15) smo za model porazdelitve mere $z(\cdot)$ izbrali gama porazdelitev

$$f(z; \hat{\alpha}, \hat{\beta}) = \frac{1}{\hat{\beta}^{\hat{\alpha}}\Gamma(\hat{\alpha})} x^{\hat{\alpha}-1} e^{-\frac{x}{\hat{\beta}}}, \tag{5.15}$$

kjer sta $\hat{a}$ in $\hat{b}$ MLE oceni parametrov na podlagi skupne množice meritev z vrednostmi

$$\hat{\alpha} = 1.769, \quad \hat{\beta} = 0.066,$$

ter 95% intervalom zaupanja:

$$p(1.719 < \hat{\alpha} < 1.818) = 0.95$$

in

$$p(0.064 < \hat{\beta} < 0.068) = 0.95.$$

Sedaj lahko definiramo novo funkcijo verjetja, ki upošteva predpostavke zaprtega sveta, kot

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto z(\mathbf{x}_t)^{\hat{\alpha}-1} e^{-\frac{z(\mathbf{x}_t)}{\hat{\beta}}}, \tag{5.16}$$

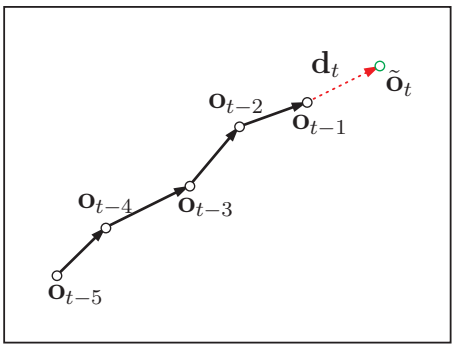
kjer je $z(\mathbf{x}_t)$ mera definirana z enačbo (5.11).

5.6 Nenaključnost gibanja igralca

V poglavju 4.3 smo si gibanje igralca razlagali le iz stališča njegovega namena, ki je obnašati se na nepredvidljiv način. Želeti nekaj, pa sicer dostikrat ne pomeni

to dejansko storiti; namreč kako si igralec predstavlja gibanje, ki ni predvidljivo, je le pojem ali želja. Na primer, ko želi igralec hitro priti iz točke T_1 v točko T_2, bo verjetno izbral tako pot, kjer izgubi najmanj energije, jo najhitreje prepotuje ter se hkrati izogne nasprotnikom. Takih poti je ponavadi več in izbira ene je že lahko tisto naključje o katerem govorimo. Vendar, ko enkrat igralec teče po tej poti z veliko hitrostjo, pa ne more spremeniti smeri gibanja povsem nenadno in poljubno zaradi inercije, porabe energije in navsezadnje, ker je njegov namen lahko čim hitreje to pot preteči. V takih primerih je lahko gibanje povsem predvidljivo. Nekoliko drugače je, kadar igralec zopet naleti na nasprotnika in se mu recimo želi ogniti: takrat je prav tako omejen z enakimi vzroki kot prej, vendar gibanje postane nekoliko bolj naključno, saj je njegov namen izogniti se nasprotniku in to čim bolj nepredvidljivo.

Igralec se torej obnaša povsem naključno le ob določenih situacijah in še takrat je omejen s porabo energije, močjo in inercijo. Predpostavimo, da je frekvenca vzorčenja slik dovolj velika, da se v nekem majhnem zaporedju slik spremembe stanja bistveno ne razlikujejo. Potem lahko predvidljiv del gibanja ob času t modeliramo z deterministično[4] spremembo $\mathbf{d}_t = (d_{x_t}, d_{y_t}, d_{a_t}, d_{b_t})$ (slika 5.16).



Slika 5.16: *Ilustracija ocenjenih stanj* $\mathbf{o}_k$. *Ocena spremembe stanja* $\mathbf{o}_{t-1}$ *v iskano stanje je označena z* $\mathbf{d}_t$, *predikcija stanja pa z* $\tilde{\mathbf{o}}_t$.

Dinamični model iz enačbe (4.9) se spremeni v

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1} + \mathbf{d}_t, \mathbf{\Lambda}_{\mathbf{x}_{t-1}}), \tag{5.17}$$

kjer je $\mathcal{N}(\cdot; \mu, \mathbf{\Sigma})$ normalna porazdelitev s srednjo vrednostjo μ in kovariančno matriko $\mathbf{\Sigma}$. Model zopet lahko zapišemo v generativni obliki kot

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{d}_t + \epsilon_t; \quad \epsilon_t \sim \mathcal{N}(\cdot; \mathbf{0}, \mathbf{\Lambda}_{\mathbf{x}_{t-1}}). \tag{5.18}$$

[4] Pravimo, da je sprememba deterministična, ker je popolnoma določena z naborom preteklih stanj.

Ocenjevanje spremembe stanja

Časovno konstantni premik $\mathbf{d}_t$ modelira gladko gibanje in ga ocenimo preko preteklih *glajenih* povprečnih stanj objekta. Naj

$$\mathbf{o}_{t-T:t-1} = \{\mathbf{o}_k\}_{k=t-T}^{t-1} \tag{5.19}$$

predstavlja nabor T-tih preteklih glajenih stanj $\mathbf{o}_k = (o_{x_k}, o_{y_k}, o_{a_k}, o_{b_k})$ tarče in naj

$$\pi_{t-T:t-1} = \{\pi_{\mathbf{o}_k}\}_{k=t-T}^{t-1} \tag{5.20}$$

predstavlja nabor pripadajočih uteži; utež $\pi_{\mathbf{o}_k}$, npr. odraža zaupanje v ocenjeno stanje $\mathbf{o}_k$.

Trenutno spremembo $\mathbf{d}_t$ ocenimo kot uteženo vsoto diferenc med preteklimi glajenimi stanji po enačbi

$$\mathbf{d}_t = c_o^{-1} \sum_{k=t-T}^{t-1} (\mathbf{o}_k - \mathbf{o}_{k-1}) G_k(t), \tag{5.21}$$

kjer so $G_k(t)$ uteži diferenc med zaporednimi stanji in je c_o je normirni faktor

$$c_o = \sum_{k=t-T}^{t-1} G_k(t).$$

Utež $G_k(t)$ določa vpliv diference $(\mathbf{o}_k - \mathbf{o}_{k-1})$ na izračun spremembe $\mathbf{d}_t$. V tej uteži se mora odražati tako zaupanje v samo pravilnost stanj $\mathbf{o}_k$ ter $\mathbf{o}_{k-1}$ kot tudi časovna komponenta, saj morajo diference v bližnji preteklosti a priori bolj vplivati na oceno spremembe kot tiste v daljni. S tem v mislih definiramo utež $G_k(t)$ kot produkt uteži posameznih stanj ter Gaussove funkcije, ki modelira a priori dodeljevanje uteži:

$$G_k(t) = \pi_k \pi_{k-1} e^{-\frac{1}{2}\frac{(k-t)^2}{\sigma_o^2}}. \tag{5.22}$$

Zaradi Gaussove funkcije v enačbi (5.22) so uteži diferenc kasnejših od $3\sigma_o$ zanemarljive in zato je potrebno spremembo stanja v praktičnih primerih ocenjevati le na sekvenci zadnjih $T = 3\sigma_o + 1$ stanj.

S predpostavko, da igralec v pol sekunde ne more bistveno spremeniti svojega stanja, smo izbrali tako vrednost T, ki ustreza časovnemu razmaku polovice sekunde. Pri frekvenci zajema slik $25 slik/s$ upoštevamo le zadnjih trinajst stanj, $T = 13$, parameter Gaussove funkcije v enačbi (5.22) pa znaša $\sigma_o = 4.3$[5]

[5]Kot zanimivost omenimo, da so avtorji v [18] sledili igralce rokometa v posnetkih s frekvenco 25slik/s. Študirali so vplive filtriranja pridobljenih trajektorij z Gaussovim filtrom in ugotovili, da dosežejo največjo natančnost ocene hitrosti pri širini filtra $6\sigma = 25$, torej pri $\sigma \doteq 4.2$, kar pa je približno parameter naše Gaussove funkcije.

Izračun glajenih stanj

Ob času t, po končani iteraciji filtra delcev, določimo aproksimacijo porazdelitvi $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ in povprečno stanje tarče po *minimalna kvadratična napaka* (MSE) ocenimo kot

$$\hat{\mathbf{x}}_t = \sum_{i=1}^{N} w_t^{(i)} \mathbf{x}_t^{(i)}. \tag{5.23}$$

Predikcijo glajenega stanja $\tilde{\mathbf{o}}_t$ določimo kot vsoto preteklega glajenega stanja $\mathbf{o}_{t-1}$, ter trenutne spremembe $\mathbf{d}_t$ po enačbi

$$\tilde{\mathbf{o}}_t = \mathbf{o}_{t-1} + \mathbf{d}_t. \tag{5.24}$$

Povprečnemu stanju $\hat{\mathbf{x}}_t$ ter *glajeni* predikciji $\tilde{\mathbf{o}}_t$ določimo uteži, ki odražajo pravilnost ocen preko funkcije verjetja

$$w_{\hat{\mathbf{x}}_t} = p(\mathbf{y}_t|\hat{\mathbf{x}}_t), \quad w_{\tilde{\mathbf{o}}_t} = p(\mathbf{y}_t|\tilde{\mathbf{o}}_t) \tag{5.25}$$

in novo glajeno stanje $\mathbf{o}_t$ določimo kot uteženo vsoto teh ocen

$$\mathbf{o}_t = \frac{\tilde{\mathbf{o}}_t \cdot w_{\tilde{\mathbf{o}}_t} + \hat{\mathbf{x}}_t \cdot w_{\hat{\mathbf{x}}_t}}{w_{\tilde{\mathbf{o}}_t} + w_{\hat{\mathbf{x}}_t}}. \tag{5.26}$$

Dobljenemu stanju $\mathbf{o}_t$ nato pripadajočo utež $\pi_{\mathbf{o}_t}$ določimo s funkcijo verjetja

$$\pi_{\mathbf{o}_t} = p(\mathbf{y}_t|\mathbf{o}_t). \tag{5.27}$$

5.7 Implementacija sledilnika za sledenje enega igralca

Problem sledenja smo v poglavju 2 umestili sledenje v področje stohastičnega ocenjevanja in predstavili popularne Monte Carlo algoritme za reševaje takih problemov. V poglavju 3 smo nato športno igro predstavili v kontekstu zaprtega sveta in podali relavantne predpostavke s katerimi je igra v tem svetu definirana. Dalje smo v poglavju 4 predstavili referenčni sledilnik, katerega smo v tem poglavju z upoštevanjem mnogih predpostavk zaprtega sveta razširili in določili sestavne dele sledilnika za sledenje enega igralca v zaprtem svetu. Zaradi jasnosti še enkrat povzemimo te sestavne dele:

1. Model ozadja generiramo z uporabo časovne mediane preko vseh slikovnih elementov v sliki (poglavje 5.1).
2. Igralca v sliki predstavimo z barvnim RGB histogramom z osmimi celicami na barvni kanal. Histograme vzorčimo uteženo in preko maske (poglavje 4.1.1).

3. Ker poznamo model ozadja, lahko histograme primerjamo glede na teksturo v sliki ozadja. Primerjavo izvedemo preko relativne razdalje med histogrami (poglavje 5.2).

4. Histogrami predstavljajo teksturo le kot statistiko pojavljanja določenih odtenkov barve, vendar ne vsebujejo prostorske informacije. Zato smo v poglavju 5.3 predstavili princip generiranja maske za izločanje ozadja, ki temelji na dinamičnem upragovanju razlike med trenutno sliko in sliko ozadja.

5. Po predpostavkah ZS(6,7) se izgledi igralcev s časom lahko spreminjajo, in ker po ZS(5) lahko pride do kratkotrajnih trkov in okluzij smo v poglavju 5.4 vpeljali model adaptacije izgleda igralca, z močjo adaptacije odvisno od mere prisotnosti igralca.

6. Sledenje je realizirano z algoritmom CONDENSATION, kateremu je potrebno določiti funkcijo verjetja in dinamični model stanj.

7. V poglavju 5.5 smo na podlagi poskusov funkcijo verjetja modelirali z dvo parametrično gama porazdelitvijo mere prisotnosti.

8. Ker predpostavljamo, da gibanje igralca ni povsem naključno (oziroma premik ob času t ni popolnoma poljuben, če poznamo obnašanje v bližnji preteklosti), smo določili dinamični model spremembe stanj z deterministično in naključno komponento.

Sledilnik inicializiramo tako, da ročno označimo sledenega igralca in vzorčimo njegov histogram. Ta referenčni histogram je vizualni model na podlagi katerega sledilnik v vsaki sliki lokalizira igralca ter se s časom adaptira spremembam v igralčevem izgledu. Iteracija sledenja poteka po algoritmu 5.2. Zaradi jasnosti smo implementacijo algoritma CONDENSATION, ki uporablja predlagano funkcijo verjetja in dinamični model prikazali z algoritmom 5.3. Parametri algoritma, ki so stalni in neodvisni od tipa posnetka so navedeni v tabeli 5.1. Edini parametri, ki jih je posnetkih potrebno določiti ročno, so omejitve v velikosti elips s katerimi sledimo igralce; t.j. največja in najmanjša možna velikost elipse. Kadar pa so posnetki na katerih sledimo igralca kalibrirani[6], lahko te parametre določimo avtomatsko; npr.: povprečna velikost elipse je krog s premerom 0.4m, največjo ter najmanjšo pa recimo določimo s $\pm 20\%$ te velikosti.

[6]S tem mislimo na kalibracijo kamere; torej, kadar poznamo preslikavo iz slike v koordinatni sistem igrišča in obratno.

Vhod:

- A posteriori porazdelitev stanj iz prejšnjega časovnega koraka predstavljena z N delci $p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}) \approx \{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$
- Nabor zadnjih $T = 13$ glajenih stanj $\mathbf{o}_{t-T:t-1} = \{\mathbf{o}_k, \pi_k\}_{k=t-T}^{t-1}$
- Referenčni histogram igralca $\hat{\mathbf{h}}_t$
- Pragovna vrednost za generiranje maske s sliko razlike κ_t'

Izhod:

- Nova a posteriori porazdelitev stanj $p(\mathbf{x}_t|\mathbf{y}_t) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$
- Novi nabor zadnjih $T = 13$ glajenih stanj $\mathbf{o}_{t-T+1:t} = \{\mathbf{o}_k, \pi_k\}_{k=t-T+1}^{t}$
- Novi referenčni histogram igralca $\hat{\mathbf{h}}_{t+1}$
- Nova pragovna vrednost za generiranje slike razlike κ_{t+1}'

1. Trenutno sliko razlike upraguj s pragovno vrednostjo κ_t' in določi masko $M_D(\cdot)$.
2. Enači masko za sledenje z masko slike razlike $M(\cdot) = M_D(\cdot)$.
3. Preko zadnjih $T = 13$ glajenih stanj določi trenutni premik $\mathbf{d}_t$ (enačba 5.21).
4. Z iteracijo algoritma CONDENSATION (algoritem. 5.3) določi novo a posteriori porazdelitev stanj $p(\mathbf{x}_t|\mathbf{y}_t) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$.
5. Na dobljeni porazdelitvi stanj oceni povprečno stanje.
6. Določi novo glajeno stanje $\mathbf{o}_t$ ter njegovo utež π_t (poglavje 5.6).
7. Z oceno prisotnosti igralca znotraj elipse glajenega stanja $\mathbf{o}_t$ določi faktor adaptacije α_t (enačba 5.7).
8. Vzorči histogram $\mathbf{h}_{\mathbf{o}_t}$ znotraj eliptičnega področja glajenega stanja $\mathbf{o}_t$.
9. Adaptiraj referenčni histogram $\hat{\mathbf{h}}_{t+1} = (1 - \alpha_t)\hat{\mathbf{h}}_t + \alpha_t \mathbf{h}_{\mathbf{o}_t}$
10. Če je potrebno, oceni novo pragovno vrednost κ_{t+1}' za generiranje slike razlike (poglavje 5.3) znotraj elipse stanja $\mathbf{o}_t$.

Algoritem 5.2: *Iteracija sledenja enega igralca v zaprtem svetu.*

Število celic v RGB histogramu	$m = 8 \times 8 \times 8$
Prag podobnosti za generiranje maske	$\varrho_T = 0.8$
Število nevidnih slikovnih elementov	$\eta_0 = 0.25$
Parameter maksimalne adaptacje	$\alpha_{max} = 0.05$
Parametri gama porazdelitve v funkciji verjetja	$(\hat{\alpha}, \hat{\beta}) = (1.769, 0.066)$
Parametri naključnega dela dinamičnega modela	$(\alpha_{xy}, \alpha_{ab}) = (0.25, 0.05)$
Vrednost parametra za izračun glajenih stanj	$\sigma_o = 4.3$ (oziroma $T = 13$)

Tabela 5.1: *Parametri sledilnika enega igralca v zaprtem svetu.*

Vhod:

- A posteriori porazdelitev stanj iz prejšnjega časovnega koraka $p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}) \approx \{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$
- Trenutni konstantni premik $\mathbf{d}_t$

Izhod:

- Nova a posteriori porazdelitev stanj $p(\mathbf{x}_t|\mathbf{y}_t) \approx \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$

1. Prevzorči z deterministčnim vzorčenjem (algoritem 2.2),
$$p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}) \approx \{\tilde{\mathbf{x}}_{t-1}^{(i)}, \tfrac{1}{N}\}_{i=1}^{N}.$$
2. Simuliraj vse delce preko dinamičnega modela $p(\mathbf{x}_t^{(i)}|\tilde{\mathbf{x}}_{t-1}^{(i)})$ iz enačbe (5.17),
$$\mathbf{x}_t^{(i)} = \tilde{\mathbf{x}}_{t-1}^{(i)} + \mathbf{d}_t + \epsilon_t; \quad \epsilon_t \sim \mathcal{N}(\cdot; \mathbf{0}, \mathbf{\Lambda}_{\mathbf{x}_{t-1}}).$$
3. Določi simuliranim delcem uteži s funkcijo verjetja (En 5.15),
$$\tilde{w}_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_t^{(i)}).$$
4. Normiraj uteži,
$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_i \tilde{w}_t^{(i)}}.$$

Algoritem 5.3: *Implementacija algoritma* Condensation *za sledenje igralca v zaprtem svetu.*

Poglavje 6

Sledenje večih igralcev

Problem sledenja večih ljudi spada v področje sledenja večih tarč *(angl. multi target tracking)*. V primeru sledenja ene tarče je sistem, ki ga opazujemo tarča sama, stanje tega sistema pa lahko zapišemo z npr. položajem in obliko te tarče. Pri sledenju večih tarč stanje opazovanega sistema ne sestavlja več le ena sama tarča, pač pa vse tarče skupaj. Trenutno stanje sistema torej lahko zapišemo kot množico stanj posameznih tarč. Klasični pristopi sledenja večih tarč temeljijo na detekciji kandidatov (meritev) in asociaciji teh kandidatov s sledenimi tarčami. Med njimi sta najbolj popularna pristopa t.i. *sledenje večih hipotez* (MHT) *(angl. multi hypothesis tracking)* in *hkratna verjetnostna asociacija podatkov* (JPDA) *(angl. joint probabilistic data association)* [72].

Ideja MHT filtrov je asociacija vsake meritve z eno od obstoječih trajektorij sledenih tarč, vendar ker te asociacije niso vedno enolične, se sproti generira ter ohranja več najbolj verjetnih hipotez. Drugačen pristop uporabljajo JPDA filtri, kjer vsako meritev asocirajo z vsako tarčo glede na verjetnost, da je tarča generirala določeno meritev. Uspešnost JPDA filtra z delci so avtorji v [73] prikazali na primeru sledenja večih ljudi z robotom.

Oba zgornja pristopa v osnovi zahtevata izdatno preiskovanja vseh možnih asociacij med meritvami ter tarčami in sta zato lahko računsko zelo potratna. Hue in ostali v [74] zato vektor asociacij tretirajo kot stohastično spremenljivko, katere vrednosti določajo s pomočjo *Gibbsovega vzorčevalnika (angl. Gibbs sampler)* [75].

Drugi možni pristop k sledenju večih tarč je sestaviti stanja posameznih tarč v eno skupno stanje in uporabiti klasične pristope za sledenje enega stanja sistema, kjer je trenutno stanje sistema torej določeno s trenutno konfiguracijo vseh tarč. V aplikaciji filtrov z delci to pomeni, da je en delec predstavljen z realizacijo položajev vseh sledenih tarč, in se njegova utež izračunava na podlagi vseh tarč skupaj. Tak pristop so [56] uporabili za sledenje znanega števila igralcev nogometa. Pérez in ostali [43] so uspešno uporabo skupnih stanj prikazali na primeru sledenja dveh ljudi. V [54] so avtorji skupno stanje

razširili s spremenljivko števila slednih objektov in tako razvili sledilnik za sledenje poljubnega števila ljudi v sobi. Eksplicitno tretiranje podatkov v sliki med okluzijo dveh tarč so v [76] realizirali s tako imenovanim *verjetnostnim principom izločanja (angl. probabilistic exclusion principle)*.

Slabost pristopov sledenja s skupnim stanjem je v tem, da slaba ocena le ene tarče v delcu lahko pokvari celotni delec, t.j. povzroči nizko utež delca. To pomeni, da je za uspešno sledenje potrebno zelo veliko število delcev, kar pa je lahko računsko potratno.

Alternativa sledenju s skupnim stanjem je sledenje vsake tarče posebej s svojim fitrom. Tu se pojavi nov problem, saj je znano, da delci v filtru težijo k mestom z večjim verjetjem. Slednja lastnost povzroči odpoved sledenja predvsem, ko sledimo enakim, ali podobnim objektom med katerimi pogosto prihaja do trkov. Takoj po trku dveh podobnih objektov je navadno eden bolj podoben modelu tarče kot drugi. Funkciji verjetja obeh filtrov imata torej na mestu enega od objektov večje vrednosti. Oba filtra zgostita delce v tem področju in v nadaljevanju oba sledita le enemu objektu.

Zaradi nizke računske zahtevnosti je torej sledenje večih tarč z ločenimi filtri privlačno, vendar je zelo občutljivo na trke in okluzije med tarčami. V [36, 32] so zato v aplikaciji sledenja igralcev v nogometu vpeljali t.i. *verjetnost okluzije (angl. occlusion alarm probability)*. Ideja je v tem, da vsakega igralca sledijo s svojim filtrom, medtem ko delce v filtrih dodatno utežujejo glede na bližino ocenjenih stanj ostalih igralcev; to pomeni, da tiste delce, ki so bliže kateremukoli od ostalih igralcev, dodatno utežijo z neko nizko utežjo. Podoben mehanizem uporabljajo v [77] pri sledenju mravelj v terariju, kjer delce v fitrih dodatno utežujejo na podlagi markovovega naključnega polja *(angl. markov random field)*. Nekoliko drugačen pristop so izbrali avtorji v [39], kjer igralcem nogometa sledijo z ločenimi fitri, probleme okluzije med igralci pa razrešujejo z uporabo predlog igralcev, katere si zapomnijo pred trkom. V [30] avtorji sledijo igralcem hokeja z ločenimi filtri, in vzdrževanje delcev na pravih lokacijah implementirajo z uporabo algoritma [43] in algoritma Adaboost.

Kadar sledimo enakim objektom, oziroma objektom, ki generirajo podobne značilnice, lahko uporabimo navaden filter z delci, ki ocenjuje a posteriori porazdelitev stanja ene tarče. V primeru večih tarč to preprosto pomeni, da ima ocenjena a posteriori porazdelitev več modusov, vsak modus pa odgovarja določeni tarči. Tak filter so implementirali v [55], kjer sledijo spremenljivemu številu objektov z mobilnim robotom, moduse v a posteriori porazdelitvi pa določajo z metodami *lokalnega rojenja (angl. local clustering)*. Problem, ki ga tovrstni filtri trpijo je, da delci z operacijami prevzorčenja hitro migrirajo k tistim modusom v porazdelitvi, ki so večji; rezultat je jasno izguba tarč, ki so predstavljene z majhnimi modusi. To problematiko so Vermaak in ostali obdelali

v [43] in predstavili rešitev. Predlagani algoritem so z uspehom preizkusili na primeru sledenja večih enakih igralcev v nogometu.

Problematiko sledenja večih tarč lahko gledamo tudi iz vidikov, ki ne temeljijo na teoriji statističnega ocenjevanja. Kot primer navedimo le nedavno predstavljen pristop, ki temelji na Viterbijevem algoritmu [38]. Glavna predpostavka slednjega algoritma je, da v vsaki sliki lahko detektiramo množico možnih položajev vseh tarč, med katerimi so nekateri položaji lahko rezultat šuma. Avtorji določijo nabor pravil po katerih se lahko sekvence položajev *vedejo* in potem iščejo *maksimalno a posteriori* MAP *(angl. maximal a posteriori)* rešitev poti skozi sekvenco možnih detektiranih položajev. Druge pristope sledenja večih tarč lahko najdemo npr. v [26, 64, 27, 78, 29, 21, 42, 31]

Pričujoča magistrska naloga se zadeva statističnih pristopov sledenja večih igralcev v športni igri. Ker smo že v poglavju 3 umestili športno igro v kontekst zaprtega sveta, bomo v tem duhu predstavili še sledenje večih igralcev.

6.1 Sledilnik za sledenje večih igralcev v zaprtem svetu

Sledilnik enega igralca je torej v splošnem neprimeren za uporabo pri sledenju večih igralcev, saj ne upošteva določenih dejstev ekipne igre in eno od teh je gotovo to, da med igro lahko prihaja do trkov med podobnimi igralci. Po predpostavki ZS1, ki govori o postavitvi kamer, gledamo igralce z vrha. V večini ekipnih športov kot so košarka, rokomet in nogomet redkokdaj eden igralec pristane na drugem, oziroma se takrat igra prekine. To pomeni, da so med regularnim delom igre vsi igralci skoraj vedno vsaj delno vidni kameri: npr. kadar sta si dva igralca zelo blizu ponavadi eden zakriva drugega, vendar je slednji zaradi postavitve kamere vsaj deloma viden. Sedaj lahko določimo dodatne predpostavke zaprtega sveta za sledenje večih igralcev:

- Med igro lahko prihaja do dolgotrajnih trkov med podobnimi igralci.
- V nekem časovnem koraku dva igralca ne moreta zasesti enakega položaja na igrišču.

Spomnimo se, da pri ocenjevanju funkcije verjetja stanj uporabljamo masko s pomočjo katere izločamo slikovne elemente, ki pripadajo ozadju. Na tak način bi lahko tudi izločali področja slikovnih elementov, ki ne pripadajo sledenemu igralcu: pred iteracijo sledenja določenega igralca lahko na podlagi trenutnega znanja o ostalih igralcih generiramo masko za izločevanje tistih področji na sliki, ki bolj verjetno pripadajo ostalim igralcem. To masko potem skupaj z masko za izločanje ozadja (poglavje 5.3) uporabimo v trenutni iteraciji sledenja. Na tak način lahko vsakega igralca sledimo s samostojnim sledilnikom za sledenje enega

igralca, in informacijo o ostalih igralcih med sledilniki prenašamo preko mask za izločanje igralcev.

V nadaljevanju bomo v prvih dveh podpoglavjih predstavili dva načina generiranja maske za izločanje igralcev. Oba pristopa bosta temeljila na idealiziranem primeru, ko poznamo prava trenutna stanja vseh igralcev. V tretjem podpoglavju bomo nato na podlagi teh pristopov predstavili dva algoritma za sledenje večih igralcev v primerih, ko prava trenutna stanja niso znana.

6.1.1 Maskiranje z uporabo Voronoievih področji

Recimo, da ob času t poznamo stanja vseh igralcev na igrišču. Množico stanj označimo z

$$\mathbf{X}_t = \{{}^{(j)}\mathbf{x}_t\}_{j=1}^{N_p},$$

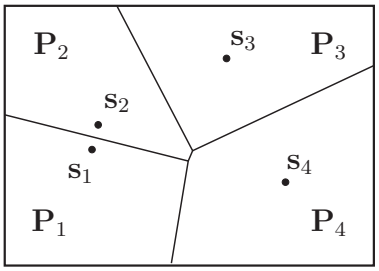
kjer je ${}^{(j)}\mathbf{x}_t = ({}^{(j)}x_t, {}^{(j)}y_t, {}^{(j)}a_t, {}^{(j)}b_t)$ stanje j-tega igralca, N_p pa je število vseh igralcev. V trenutni sliki je j-ti igralec predstavljen z neko množico slikovnih elementov, ki jo označimo s ${}^{(j)}\mathbf{K}$. Če predpostavimo, da je vsak slikovni element iz te množice $\mathbf{u} = (x, y), \mathbf{u} \in {}^{(j)}\mathbf{K}$ prostorsko najbližji centru $({}^{(j)}x_t, {}^{(j)}y_t)$ elipse stanja j-tega igralca ${}^{(j)}\mathbf{x}_t$, potem lahko trenutno sliko segmentiramo po pravilu *najbližji sosed* (NN) *(angl. nearest neighbor)* v N_p takih paroma disjunktivnih področji, da j-to področje vsebuje le množico ${}^{(j)}\mathbf{K}$. Ta predpostavka sicer ne drži kadar so elipse stanj igralcev zelo podolgovate (t.j. kadar je ena os elipse precej večja od druge) in so si hkrati igralci zelo blizu, vendar je tak dogodek zelo neobičajen in redek.

Delitev prostora, ki doseže zgoraj omenjeno topologijo je znani Voronoiev diagram (glej npr. [79] stran 178). Voronoiev diagram med N_p točkami ali semeni je popolnoma definiran z množico semen $\mathbf{S} = \{{}^{(j)}\mathbf{s}\}_{j=1}^{N_p}$ in generira množico neprekrivajočih se konveksnih področji/parcel $\mathbf{P} = \{{}^{(j)}\mathbf{P}\}_{j=1}^{N_p}$ tako, da vsako področje vsebuje tiste točke prostora, ki so najbližje njegovemu semenu. Primer Voronoievega diagrama med štirimi semeni prikazuje slika 6.1.

Če torej poznamo trenutna stanja igralcev na igrišču, lahko generiramo masko za I-tega igralca po algoritmu 6.1. Primer maske enega igralca z uporabo Voronoievih področji prikazuje slika 6.2.

6.1.2 Maskiranje s skrivanjem stanj igralcev

Drugi postopek generiranja maske za izločanje igralcev je nekoliko bolj preprost. Zopet recimo, da ob času t poznamo stanja vseh igralcev na igrišču. Množico trenutnih stanj označimo z $\mathbf{X}_t = \{{}^{(j)}\mathbf{x}_t\}_{j=1}^{N_p}$, kjer je ${}^{(j)}\mathbf{x}_t = ({}^{(j)}x_t, {}^{(j)}y_t, {}^{(j)}a_t, {}^{(j)}b_t)$ stanje j-tega igralca in je N_p število vseh igralcev. Če predpostavimo, da so



Slika 6.1: *Voronoiev diagram med štirimi točkami/semeni* $\{\mathbf{s}_i\}_{i=1}^4$ *generira po pravilu NN take štiri parcele* $\{\mathbf{P}_i\}_{i=1}^4$*, da je vsaka točka znotraj neke parcele najbližja semenu te parcele.*

Vhod:

- Množica trenutnih stanj N_p-tih igralcev $\mathbf{X}_t = \{{}^{(j)}\mathbf{x}_t\}_{j=1}^{N_p}$, kjer je ${}^{(j)}\mathbf{x}_t = ({}^{(j)}x_t, {}^{(j)}y_t, {}^{(j)}a_t, {}^{(j)}b_t)$ trenutno stanje j-tega igralca.
- Indeks igralca I.

Izhod:

- Maska I-tega igralca ${}^{(I)}M_{V_t}$ za zakrivanje ostalih igralcev.

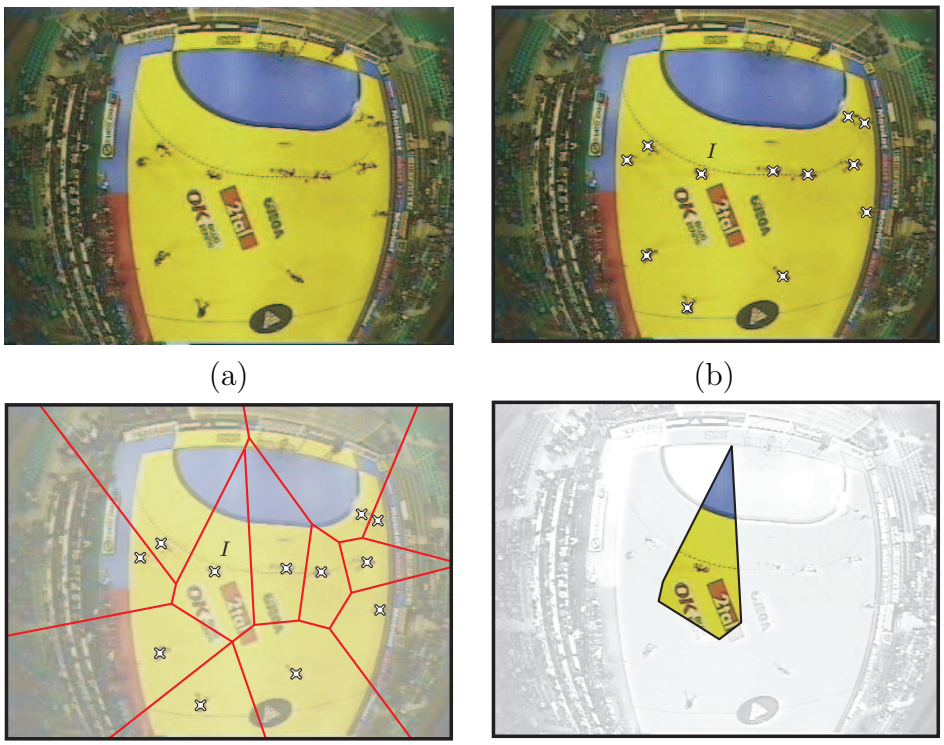
---

1. Zgradi množico semen $\mathbf{S} = \{{}^{(j)}\mathbf{s}\}_{j=1}^{N_p}$, kjer je ${}^{(j)}\mathbf{s} = ({}^{(j)}x_t, {}^{(j)}y_t)$.
2. Na podlagi semen zgradi Voronoieve parcele $\mathbf{P} = \{{}^{(j)}\mathbf{P}\}_{j=1}^{N_p}$.
3. Za vsak slikovni element $\mathbf{u}$ v trenutni sliki zgradi masko po sledečem pravilu

$$ {}^{(I)}M_{V_t}(\mathbf{u}) = \begin{cases} 1 & ; \quad \mathbf{u} \in {}^{(I)}P \\ 0 & ; \quad sicer \end{cases} . \tag{6.1} $$

Algoritem 6.1: *Generiranje maske I-tega igralca z uporabo Voronoievih parcel.*

vsa stanja v množici $\mathbf{X}_t$ res prava, potem vse igralce razen I-tega izločimo tako, da generiramo masko ${}^{(I)}M_{S_t}$, kjer zakrijemo vse elipse stanj iz množice $\mathbf{X}_t$, razen elipse I-tega stanja. Postopek izgradnje maske je opisan z algoritmom 6.2, primer maske enega igralca s skrivanjem stanj pa prikazuje slika 6.3.



(a) (b)

(c) (d)

Slika 6.2: *Primer izgradnje mask z Voronoievim diagramom. Sliki (a,b) prikazujeta igralce rokometa na igrišču. Voronoieva maska (c,d) dodeli I-temu igralcu v sliki področje katerega točke so najbližje centru stanja I-tega igralca glede na ostale igralce. To področje je posebej označeno v sliki (d).*

6.2 Implementacija sledilnika za sledenje večih igralcev

S pomočjo postopkov izgradnje mask, predstavljenih v prejšnjih dveh poglavjih, lahko za določenega igralca izločimo tista področja, ki v nekem smislu pripadajo ostalim igralcem. To pomeni, da z maskami vsakemu igralcu *simuliramo* zaprti svet, in torej lahko potem igralca sledimo s sledilnikom za sledenje enega igralca v zaprtem svetu (poglavje 5).

Težava je v tem, da pri gradnji mask predpostavljamo, da že pred iteracijo sledenja ob času t poznamo trenutna stanja vseh igralcev $\mathbf{X}_t = \{{}^{(j)}\mathbf{x}_t\}_{j=1}^{N_p}$. V resnici teh stanj sicer ne poznamo, vendar jih lahko ocenimo. Za j-tega igralca, na primer, lahko ocenimo trenutno stanje ${}^{(j)}\tilde{\mathbf{x}}_t$ preko njegovega glajenega stanja ${}^{(j)}\mathbf{o}_{t-1}$ iz predhodnega časovnega trenutka in ocene trenutnega premika ${}^{(j)}\mathbf{d}_t$ (glej poglavje 5.6). Začetno oceno množice stanj vseh igralcev potemtakem zapišemo kot

$$\tilde{\mathbf{X}}_t = \{{}^{(j)}\tilde{\mathbf{x}}_t\}_{j=1}^{N_p} \quad ; \quad {}^{(j)}\tilde{\mathbf{x}}_t = {}^{(j)}\mathbf{o}_{t-1} + {}^{(j)}\mathbf{d}_t. \tag{6.3}$$

Vhod:

- Množica trenutnih stanj N_p-tih igralcev $\mathbf{X}_t = \{{}^{(j)}\mathbf{x}_t\}_{j=1}^{N_p}$, kjer je ${}^{(j)}\mathbf{x}_t = ({}^{(j)}x_t, {}^{(j)}y_t, {}^{(j)}a_t, {}^{(j)}b_t)$ trenutno stanje j-tega igralca.
- Indeks igralca I.

Izhod:

- Maska I-tega igralca ${}^{(I)}M_{S_t}$ za zakrivanje ostalih igralcev.

1. Za vsak slikovni element $\mathbf{u}$ v trenutni sliki zgradi masko po sledečem pravilu

$$ {}^{(I)}M_{S_t}(\mathbf{u}) = \begin{cases} 0 & ; \quad \mathbf{u} \in {}^{(j)}E, j \neq I \\ 1 & ; \quad sicer \end{cases}, \tag{6.2} $$

   kjer je ${}^{(j)}E$ elipsa stanja j-tega igralca.

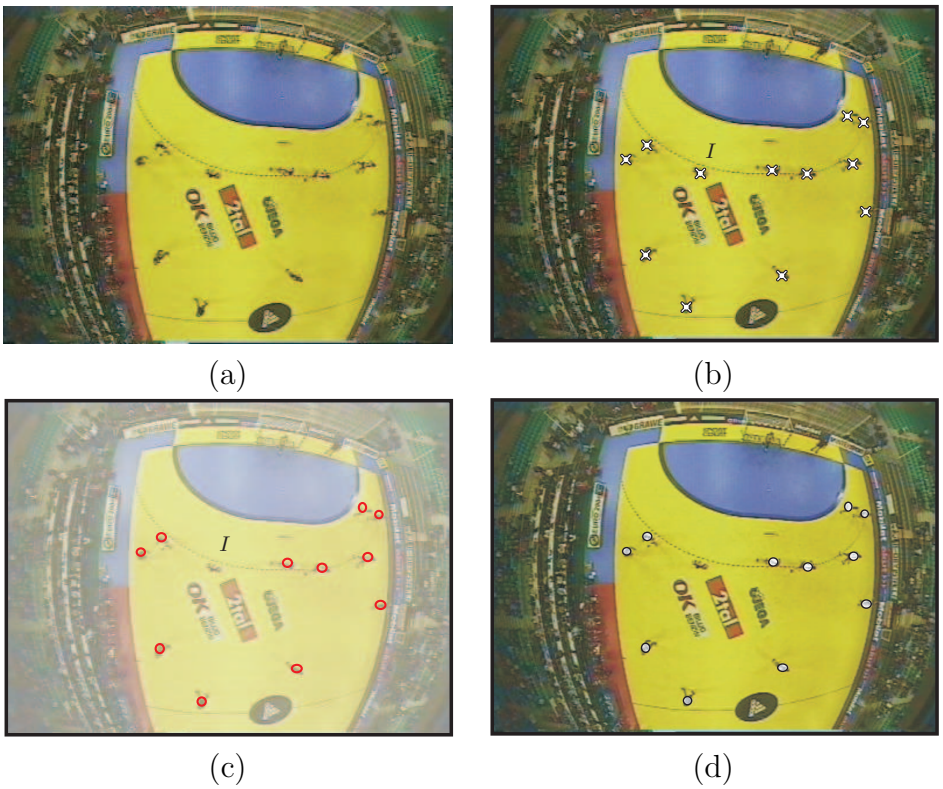
Algoritem 6.2: *algoritem za generiranje maske I-tega igralca s skrivenjem stanj.*

S to oceno lahko zgradimo masko za npr. j-tega igralca. Po končani iteraciji sledenja j-tega igralca dobimo boljšo oceno trenutnega stanja ${}^{(j)}\tilde{\mathbf{x}}_t$, ki je lahko npr. glajeno stanje ${}^{(j)}\mathbf{o}_t$, in množico stanj $\tilde{\mathbf{X}}_t$ potem izboljšamo tako, da s tem stanjem nadomestimo prejšnjo oceno, t.j. ${}^{(j)}\tilde{\mathbf{x}}_t = {}^{(j)}\mathbf{o}_t$. Ta postopek ponavljamo dokler ne izvršimo iteracij sledenja vseh igralcev.

Sledilnik za sledenje N_p-tih igralcev v zaprtem svetu zapišemo kot množico ločenih sledilnikov za enega igralca v zaprtem svetu. Trenutno znanje o vseh igralcih zapišemo kot množico

$$ \{{}^{(j)}p(\mathbf{x}_t|\mathbf{y}_{1:t-1}), {}^{(j)}p(\mathbf{x}_t|\mathbf{x}_{t-1}), {}^{(j)}\mathbf{o}_{t-T:t-1}{}^{(j)}\hat{\mathbf{h}}_t, {}^{(j)}\kappa'_t\}_{j=1}^{N_p}, \tag{6.4} $$

kjer z ${}^{(j)}(\cdot)$ označimo j-tega igralca, ${}^{(j)}p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ je njegova a posteriori porazdelitev stanj iz prejšnjega časovnega koraka, ${}^{(j)}p(\mathbf{x}_t|\mathbf{x}_{t-1})$ je njegov dinamični model, ${}^{(j)}\mathbf{o}_{t-T:t-1}$ pa je množica preteklih $T = 13$ glajenih stanj. Oceno barvnega modela igralca predstavlja njegov barvni histogram ${}^{(j)}\hat{\mathbf{h}}_t$, ${}^{(j)}\kappa'_t$ pa je v prejšnjem časovnem koraku ocenjen prag za izločanje ozadja. Implementacijo sledilnika za sledenje večih igralcev v zaprtem svetu z uporabo Voronoievih parcel prikazuje algoritem 6.3, medtem ko implementacijo sledenja večih igralcev v zaprtem svetu z uporabo skrivanja stanj, algoritem 6.4. Iz slik je očitno, da se sledilnika razlikujeta le v načinu izgradnje maske za generiranje zaprtih svetov posameznih sledilnikov.



(a) (b)

(c) (d)

Slika 6.3: *Primer izgradnje mask s postopkom skrivanja stanj. Sliki (a,b) prikazujeta igralce rokometa na igrišču. V sliki (c) so označena trenutna stanja igralcev z elipsami. Dobljena maska za I-tega igralca (d) skrije stanja vseh igralcev razen stanja, ki pripada I-temu.*

Vhod:

- A posteriori porazdelitve stanj vseh igralcev iz prejšnjega časovnega koraka $\{^{(j)}p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \approx {}^{(j)}\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}\}_{j=1}^{N_p}$
- Nabori zadnjih $T = 13$ glajenih stanj vseh igralcev $\{^{(j)}\mathbf{o}_{t-T:t-1}\}_{j=1}^{N_p}$
- Referenčni histogrami igralcev $\{^{(j)}\hat{\mathbf{h}}_t\}_{j=1}^{N_p}$
- Pragovne vrednosti za generiranje slike razlike $\{\kappa'_t\}_{j=1}^{N_p}$

Izhod:

- Nove a posteriori porazdelitve stanj $\{^{(j)}p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx {}^{(j)}\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}\}_{j=1}^{N_p}$
- Novi nabori zadnjih $T = 13$ glajenih stanj $\{^{(j)}\mathbf{o}_{t-T+1:t}\}_{j=1}^{N_p}$
- Novi referenčni histogrami igralcev $\{^{(j)}\hat{\mathbf{h}}_t\}_{j=1}^{N_p}$
- Nove pragovne vrednosti za generiranje slike razlike $\{\kappa'_{t+1}\}_{j=1}^{N_p}$

1. Inicializiraj množico ocenjenih stanj igralcev $\tilde{\mathbf{X}}_t = \{^{(j)}\tilde{\mathbf{x}}_t\}_{j=1}^{N_p}; {}^{(j)}\tilde{\mathbf{x}}_t = {}^{(j)}\mathbf{o}_{t-1} + {}^{(j)}\mathbf{d}_t$
2. Za vsak $j = 1 : N_P$ izvedi iteracijo
   - Z množico centrov $\tilde{\mathbf{X}}_t$ določi masko za izločanje igralcev ${}^{(j)}M_{V_t}$ z algoritmom 6.1.
   - Trenutno sliko razlike upraguj s pragovno vrednostjo κ'_t in določi masko ${}^{(j)}M_D$ (poglavje 5.3).
   - Določi masko za sledenje j-tega igralca ${}^{(j)}M_t = {}^{(j)}M_D \cap {}^{(j)}M_{V_t}$
   - Z iteracijo algoritma Condensation (algoritem 5.3) določi novo a posteriori porazdelitev stanj ${}^{(j)}p(\mathbf{x}_t|\mathbf{y}_t) \approx {}^{(j)}\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$.
   - Na dobljeni a posteriori porazdelitvi stanj oceni povprečno stanje in določi novo glajeno stanje ${}^{(j)}\mathbf{o}_t$ ter njegovo utež ${}^{(j)}\pi_t$ (poglavje 5.6).
   - Z oceno prisotnosti igralca znotraj elipse glajenega stanja določi faktor adaptacije α_t (enačba 5.7).
   - Vzorči histogram $\mathbf{h}_{\mathbf{o}_t}$ znotraj eliptičnega področja glajenega stanja ${}^{(j)}\mathbf{o}_t$.
   - Adaptiraj referenčni histogram ${}^{(j)}\hat{\mathbf{h}}_{t+1} = (1 - \alpha_t){}^{(j)}\hat{\mathbf{h}}_t + \alpha_t\mathbf{h}_{\mathbf{o}_t}$.
   - Če je potrebno, oceni novo pragovno vrednost ${}^{(j)}\kappa'_{t+1}$ za generiranje slike razlike (poglavje 5.3) znotraj elipse stanja ${}^{(j)}\mathbf{o}_t$.
   - Posodobi množico stanj $\tilde{\mathbf{X}}_t$ s trenutnim glajeni stanjem: ${}^{(j)}\tilde{\mathbf{x}}_t = {}^{(j)}\mathbf{o}_t$.

Algoritem 6.3: *Iteracija sledenja večih igralcev v zaprtem svetu z uporabo Voronoievih parcel.*

Vhod:

- A posteriori porazdelitve stanj vseh igralcev iz prejšnjega časovnega koraka $\{^{(j)}p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \approx {}^{(j)}\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}\}_{j=1}^{N_p}$
- Nabori zadnjih $T = 13$ glajenih stanj vseh igralcev $\{^{(j)}\mathbf{o}_{t-T:t-1}\}_{j=1}^{N_p}$
- Referenčni histogrami igralcev $\{^{(j)}\hat{\mathbf{h}}_t\}_{j=1}^{N_p}$
- Pragovne vrednosti za generiranje slike razlike $\{\kappa'_{t+1}\}_{j=1}^{N_p}$

Izhod:

- Nove a posteriori porazdelitve stanj $\{^{(j)}p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx {}^{(j)}\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}\}_{j=1}^{N_p}$
- Novi nabori zadnjih $T = 13$ glajenih stanj $\{^{(j)}\mathbf{o}_{t-T+1:t}\}_{j=1}^{N_p}$
- Novi referenčni histogrami igralcev $\{^{(j)}\hat{\mathbf{h}}_t\}_{j=1}^{N_p}$
- Nove pragovne vrednosti za generiranje slike razlike $\{\kappa'_t\}_{j=1}^{N_p}$

1. Inicializiraj množico ocenjenih stanj igralcev $\tilde{\mathbf{X}}_t = \{^{(j)}\tilde{\mathbf{x}}_t\}_{j=1}^{N_p}; {}^{(j)}\tilde{\mathbf{x}}_t = {}^{(j)}\mathbf{o}_{t-1} + {}^{(j)}\mathbf{d}_t$
2. Za vsak $j = 1 : N_P$ izvedi iteracijo
   - Z množico centrov $\tilde{\mathbf{X}}_t$ določi masko za izločanje igralcev ${}^{(j)}M_{S_t}$ z algoritmom 6.2.
   - Trenutno sliko razlike upraguj s pragovno vrednostjo κ'_t in določi masko ${}^{(j)}M_D$ (poglavje 5.3).
   - Določi masko za sledenje j-tega igralca ${}^{(j)}M_t = {}^{(j)}M_D \cap {}^{(j)}M_{S_t}$
   - Z iteracijo algoritma Condensation (algoritem 5.3) določi novo a posteriori porazdelitev stanj ${}^{(j)}p(\mathbf{x}_t|\mathbf{y}_t) \approx {}^{(j)}\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N}$.
   - Na dobljeni a posteriori porazdelitvi stanj oceni povprečno stanje in določi novo glajeno stanje ${}^{(j)}\mathbf{o}_t$ ter njegovo utež ${}^{(j)}\pi_t$ (poglavje 5.6).
   - Z oceno prisotnosti igralca znotraj elipse glajenega stanja določi faktor adaptacije α_t (enačba 5.7).
   - Vzorči histogram $\mathbf{h}_{\mathbf{o}_t}$ znotraj eliptičnega področja glajenega stanja ${}^{(j)}\mathbf{o}_t$.
   - Adaptiraj referenčni histogram ${}^{(j)}\hat{\mathbf{h}}_{t+1} = (1 - \alpha_t){}^{(j)}\hat{\mathbf{h}}_t + \alpha_t\mathbf{h}_{\mathbf{o}_t}$.
   - Če je potrebno, oceni novo pragovno vrednost ${}^{(j)}\kappa'_{t+1}$ za generiranje slike razlike (poglavje 5.3) znotraj elipse stanja ${}^{(j)}\mathbf{o}_t$.
   - Posodobi množico stanj $\tilde{\mathbf{X}}_t$ s trenutnim glajeni stanjem: ${}^{(j)}\tilde{\mathbf{x}}_t = {}^{(j)}\mathbf{o}_t$.

Algoritem 6.4: *Iteracija sledenja večih igralcev v zaprtem svetu z uporabo zakrivanja stanj.*

Poglavje 7

Preizkusi sledilnikov

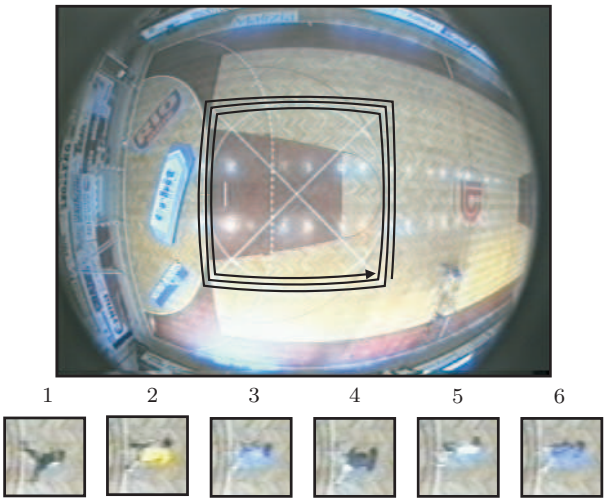
V poglavju 5 smo predstavili sledilnik za sledenje enega igralca v zaprtem svetu in ga v poglavju 6 razširili na problem sledenja večih igralcev. Predlagali smo dva principa kako povezati samostojne sledilnike za enega igralca v skupni sledilnik in s tem izboljšati sledenje. Ker je ozadje vseh predstavljenih sledilnikov Monte Carlo ocenjevanje, torej aproksimacija zveznih porazdelitev z delci, se bomo v prvem poglavju nekoliko posvetili izbiri števila delcev v sledilniku za enega igralca. Z izbranim številom delcev bomo nato primerjali delovanje sledilnika, ki upošteva predpostavke zaprtega sveta z referenčnim sledilnikom iz poglavja 4. Obnašanje obeh sledilnikov bomo analizirali s sledenjem posameznih igralcev med gibanjem v video posnetkih, ki simulirajo značilnosti tipične igre in enem težavnem video posnetku, kjer se igralec giblje po njemu podobnem ozadju. Uspešnost in probleme sledenja, ko nekatere predpostavke zaprtega sveta ne držijo, bomo prikazali s primeri sledenja igralcev v video posnetkih, zajetih s tribune športne dvorane.

Nazadnje bomo primerjali sledenje večih igralcev z uporabo algoritmov iz poglavja 6 in rezultate primerjali z rezultati sledilnika, ki ne upošteva prisotnosti ostalih igralcev.

Vsi testi, ki so opisani v sledečih poglavjih, so bili izvedeni na osebnem računalniku s procesorjem Intel Pentium IV, 2.6GHz, 2GB pomnilnika, operacijskim sistemom Windows XP in implementirani v programskem jeziku C++.

7.1 Izbira števila delcev

Uspešnost sledenja s filtri delcev je močno odvisna od števila delcev s katerimi aproksimiramo a posteriori porazdelitev stanj. Če si filtre delcev predstavljamo kot sekvenčne Monte Carlo metode za rekurzivno ocenjevanje integralov na a posteriori porazdelitvi stanj (npr. matematično upanje porazdelitve), potem je



Slika 7.1: *Šest različnih igralcev je med poskusom za določanje primernega števila delcev v filtru teklo po kvadratni poti, ki je bila narisana na igrišču. Celotna pretečena pot enega igralca je znašala tri obhode po kvadratu.*

jasno, da z naraščanjem števila delcev varianca vrednosti izračunanega integrala upada. Torej, večje ko je število delcev, boljša je npr. aproksimacija povprečnega stanja, prav tako pa sledenje redkeje popolnoma odpove. Pri velikem številu delcev lahko tudi povečamo volumen preiskovalnega prostora, hkrati pa ohranimo natančnost sledenja. To sicer pomeni, da lahko uporabljamo bolj splošen dinamični model igralcev, vendar ker je dinamika igralcev že sama po sebi značilnost sledenega objekta, se to ne zdi smiselno, kadar je dinamični model poznan.

Žal je obratno s časom izvedbe ene iteracije sledenja: več ko je delcev v filtru, počasnejše je sledenje. Časovna zahtevnost algoritma 5.2 je približno premo sorazmerna številu delcev in je torej njihovo število potrebno izbrati s kompromisom med natančnostjo in uspešnostjo sledenja na eni strani ter časovno potratnostjo na drugi. V tem poglavju si bomo zato v nadaljevanju pogledali vpliv števila delcev na uspešnost sledenja.

Za vrednotenje obnašanja sledilnika pri različnem številu delcev smo izvedli poskuse na video posnetku, v katerih so različni igralci v različnih dresih tekli po kvadratni poti, ki je bila začrtana na igrišču. Frekvenca zajema slik v video posnetku je bila 25 slik na sekundo, medtem ko so velikosti slik znašale 384×288 slikovnih elementov; igralce in pot prikazuje slika 7.1. Osvetlitev igrišča je bila vzdolž začrtane poti nehomogena in se je med gibanjem igralcev spreminjala. Posnetek je torej predstavljal kontrolirano okolje, kjer je bil sledeni igralec vedno viden in se v njegovi neposredni bližini ni nikoli nahajal še kak drug igralec.

Vsakega igralca smo sledili N_m-krat in tako za vsak trenutek dobili po N_m meritev igralčevega položaja ob tistem času. Ponovljivost sledenja j-tega igralca ob času t lahko merimo kot varianco N_m-tih trenutnih položajev in sledi enačbi

$$ {}^{(j)}r_t = \sigma_{x_t}^2 + \sigma_{y_t}^2. \tag{7.1} $$

Vrednosti $\sigma_{x_t}^2$ in $\sigma_{y_t}^2$ v zgornji enačbi sta varianci položajev v x ter y smeri

$$ \begin{aligned} \sigma_{x_t}^2 &= \frac{1}{N_m} \sum_{i=1}^{N_m} (x_t^{(i)} - \bar{x}_t)^2, \\ \sigma_{y_t}^2 &= \frac{1}{N_m} \sum_{i=1}^{N_m} (y_t^{(i)} - \bar{y}_t)^2, \end{aligned} $$

kjer je $(x_t^{(i)}, y_t^{(i)})$ i-ta ocena položaja igralca ob času t, $(\bar{x}_t, \bar{y}_t)$ pa povprečni položaj igralca ob tem času:

$$ \begin{aligned} \bar{x}_t &= \frac{1}{N_m} \sum_{i=1}^{N_m} x_t^{(i)}, \\ \bar{y}_t &= \frac{1}{N_m} \sum_{i=1}^{N_m} y_t^{(i)}. \end{aligned} $$

Zaradi preglednosti smo v zgornjih enačbah opustili oznako igralca ${}^{(j)}(\cdot)$, saj je jasno, da se enačbe nanašajo na j-tega igralca.

Ponovljivost ${}^{(j)}r$ celotne trajektorije dolžine T časovnih korakov za j-tega igralca določimo kot povprečno varianco ${}^{(j)}r_t$ preko vseh trenutkov t po enačbi

$$ {}^{(j)}r = \frac{1}{T} \sum_{t=1}^{T} {}^{(j)}r_t. \tag{7.2} $$

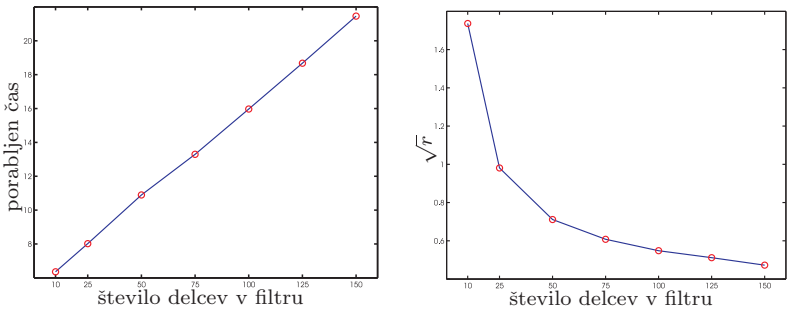
Vpliv števila delcev na sledenje smo ocenjevali tako, da smo v video posnetku pri izbranem številu delcev 30-krat sledili šestim igralcem ($N_m = 30$), za vsakega igralca po enačbi (7.2) določili varianco ${}^{(j)}r$ in ponovljivost sledenja vseh igralcev določili kot

$$ r = \frac{1}{6} \sum_{j=1}^{6} {}^{(j)}r. \tag{7.3} $$

Vrednost $r^{1/2}$ si lahko predstavljamo kot pričakovano negotovost sledilnika med sledenjem z določenim številom delcev. Hkrati smo med sledenjem beležili število odpovedi sledilnika in tiste dele trajektorij, kjer je sledenje odpovedalo nismo uporabili za izračun varianc.

Čas izvršitve iteracije sledenja pri različnem številu delcev smo z ločenim poskusom ocenili tako, da smo med sledenjem enega igralca namesto ene iteracije ob vsakem časovnem trenutku izvršili 100 iteracij in izračunali povprečni čas obdelave. Povprečje teh med sledenjem dobljenih povprečnih časov je predstavljalo oceno trajanja ene iteracije.

Rezultati poskusa so podani v sliki 7.2 in tabeli 7.1, kjer smo oceno pričakovane negotovosti podali še v deležu povprečne velikosti elipse s katero smo igralce sledili. Velikost elipse (H_t) je definirana z enačbo (4.14), izmerjena vrednost te velikosti med sledenjem pa je znašala 11 slikovnih elementov.



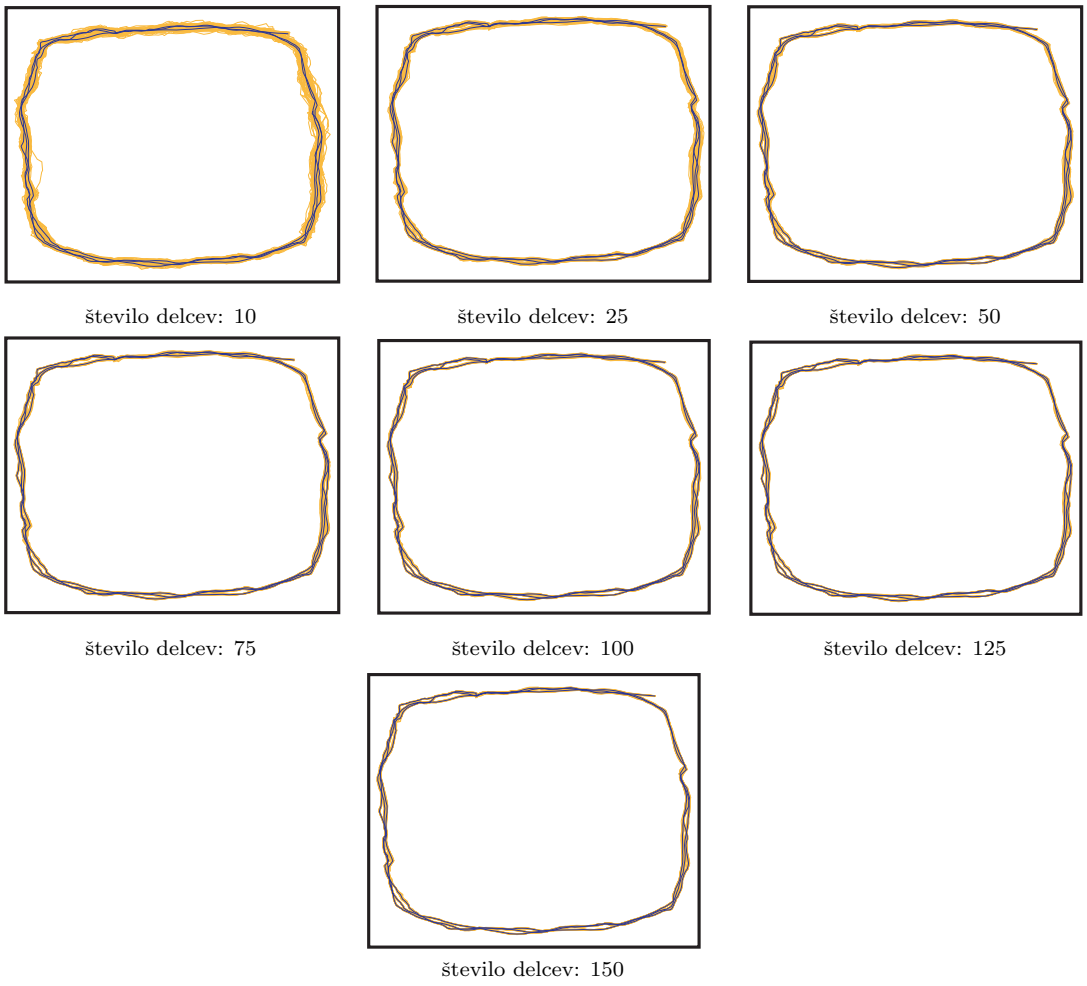
(a) (b)

Slika 7.2: *Graf porabljenega časa za eno iteracijo sledenja v odvisnosti od števila delcev (a) in potek negotovosti sledilnika v odvisnosti od števila delcev (b).*

št. delcev v filtru	št. vseh odpovedi	čas za eno iteracijo $[ms]$	$\sqrt{r}$ [slik. el.]	$\sqrt{r}$ [delež velikosti igralca]
10	7	6.4	1.74	0.16
25	0	8.0	0.98	0.09
50	0	10.9	0.71	0.06
75	0	13.3	0.61	0.06
100	0	16.0	0.55	0.05
125	0	18.6	0.51	0.05
150	0	21.5	0.47	0.04

Tabela 7.1: *Rezultati sledenja s sledilniki z različnim številom delcev.*

Slika 7.3 prikazuje trideset trajektorij igralca št. 2 iz slike 7.1 pridobljenih s sledilniki z različnim številom delcev. Že na pogled je opaziti, da varianca ocene trajektorije upada z naraščanjem števila delcev. Sledenje z desetimi delci je rezultiralo v največji varianci, s povečevanjem števila delcev pa se varianca hitro manjša. To tendenco opazimo tudi pri rezultatih v četrtem ter petem stolpcu tabele 7.1 in sliki 7.2(b), kjer vidimo, da varianca najbolj upade pri prehodu iz desetih na petindvajset delcev, in se bistveno več ne spreminja po petdesetih



Slika 7.3: *Trajektorije igralca št. 2 (slika 7.1) pridobljene s sledenjem z različnim številom delcev. Vse slike prikazujejo po trideset trajektorij, ko je igralec trikrat pretekel na igrišču narisan kvadrat (oranžno). Z modro barvo je o značena trajektorija dobljena s povprečenjem teh tridesetih. Vidimo, da razpršenost trenutnih položajev v trajektorijah upada s številom delcev v filtru.*

delcih. Sledenje je bilo najhitrejše pri najnižjem številu delcev s časom 6.4ms na iteracijo, vendar je med poskusi tudi sedemkrat odpovedalo. V primerih, ko je sledilnik vseboval več kot deset delcev, sledenje ni nikoli odpovedalo in je bila ponovljivost vedno znotraj desetih odstotkov povprečne velikosti elipse igralca (H_t). Na tem mestu naj poudarimo, da je čas ene iteracije sledilnika poleg števila delcev odvisen še od velikosti elipse s katero igralca sledimo. V trenutni implementaciji sledilnika je ta odvisnost približno premosorazmerna.

Glede na čas, ki ga sledilnik porabi za eno iteracijo, ponovljivost sledenja in število odpovedi med preizkusi, smo ugotovili, da sledenje deluje zadovoljivo s

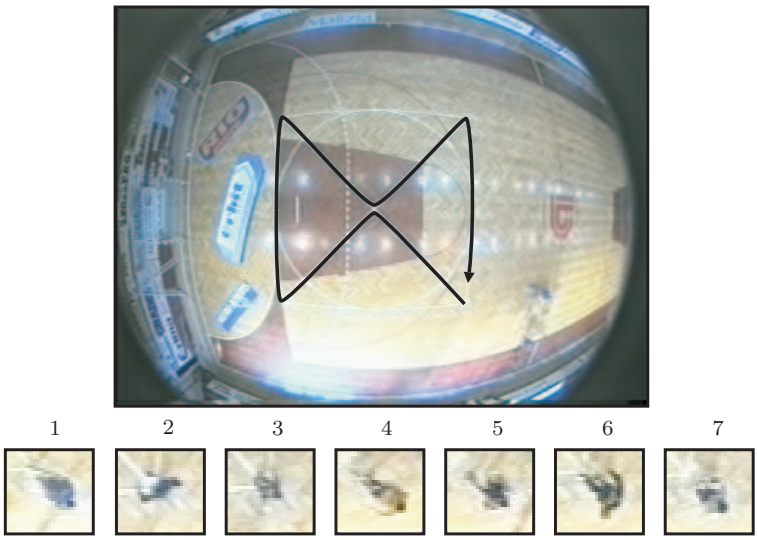
petindvajsetimi delci. Upoštevajoč to ugotovitev, smo vse poskuse sledenja v nadaljnjih poglavjih izvajali s sledilniki, ki so vsebovali po petindvajset delcev.

7.2 Uspešnost sledenja enega igralca s predpostavkami zaprtega sveta

Z izbiro števila delcev v sledilniku enega igralca v prejšnjem poglavju smo določili še zadnji parameter sledilnika enega igralca, ki upošteva predpostavke zaprtega sveta. Ta sledilnik (označimo ga z $\mathbf{S}_{zs}$) smo primerjali z referenčno različico, ki ni upoštevala predpostavk zaprtega sveta (algoritem 4.1); slednjo označimo z $\mathbf{S}_{ref}$.

7.2.1 Sledenje v kontroliranem okolju

Najprej smo teste izvedli na posnetku kjer je sedem igralcev, vsak posebej, šprintalo po začrtani poti na igrišču (slika 7.4) in zaradi oblike poti izvajali nenadne in občutne spremembe smeri gibanja.



Slika 7.4: *Slika igrišča z označeno potjo po kateri so tekli igralci in slike igralcev.*

Vsakega igralca smo petkrat sledili skozi sekvenco 250-ih slik s sledilnikoma $\mathbf{S}_{ref}$ in $\mathbf{S}_{zs}$ ter beležili število odpovedi sledenja. Število delcev v vsakem sledilniku je bilo 25, prostor elips stanj je bil omejen na $\{a, b \in [4, 7]\}$. Ostali parametri sledilnikov $\mathbf{S}_{zs}$ in $\mathbf{S}_{ref}$ so dani v tabeli 5.1.

Tabela 7.2 prikazuje število vseh odpovedi sledilnikov v petih poskusih sledenja sedmih igralcev in povprečno število odpovedi preračunano na enega

	ig.1	ig.2	ig.3	ig.4	ig.5	ig.6	ig.7	povp. število odpovedi na igralca
$\mathbf{S}_{ref}$	4	8	13	20	11	5	16	2.2
$\mathbf{S}_{zs}$	0	0	0	0	0	0	0	0

Tabela 7.2: *Število vseh odpovedi sledilnikov v petih poskusih sledenja sedmih igralcev za vsakega igralca posebej in povprečno število odpovedi na poskus, preračunano za enega igralca.*

igralca. Sledilnik $\mathbf{S}_{zs}$ med poskusi ni nikoli odpovedal, medtem ko je $\mathbf{S}_{ref}$ v povprečju 2.2-krat. To seveda ne pomeni, da je sledilnik $\mathbf{S}_{zs}$ tako robusten da uspe slediti igralca v kakršnemkoli posnetku, pač pa to, da je $\mathbf{S}_{zs}$ zgrajen na nekih predpostavkah zaprtega sveta, katerih neupoštevanje lahko privede do slabega sledenja. Rezultati v tabeli 7.2 le navajajo, da $\mathbf{S}_{zs}$ s predpostavkami zaprtega sveta dobro interpretira tipično okolje športne igre, ko se poleg sledenega igralca ne nahaja noben drug igralec.

7.2.2 Sledenje v težavnih razmerah

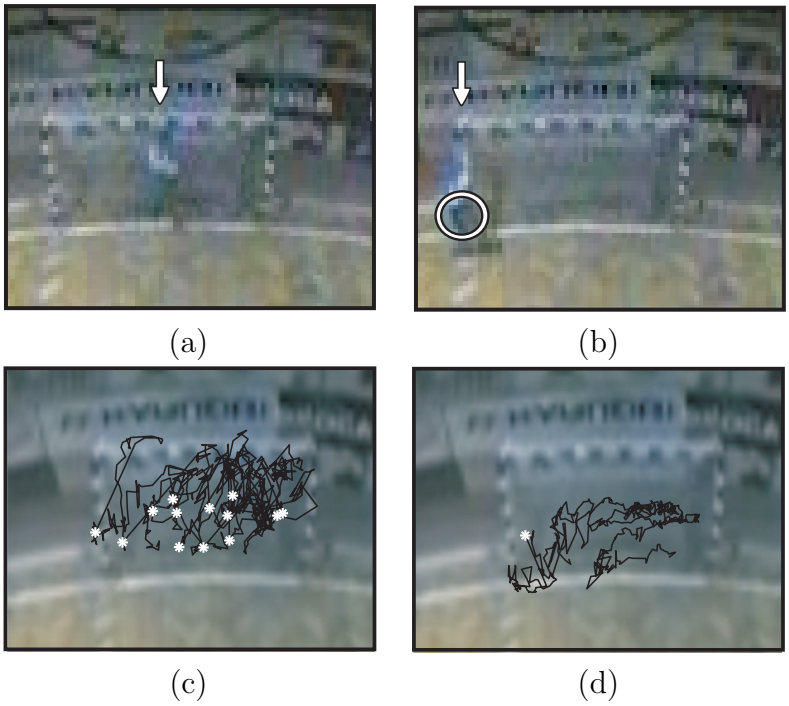
V zgornjem poskusu so se v povprečju sledeni igralci po izgledu dokaj razlikovali od ozadja. Da bi nekoliko osvetlili problematiko vpliva ozadja na sledenje, smo izvedli še poskus, ko se je igralec nahajal na njemu zelo podobnem ozadju.

V posnetku igre rokometa smo tako izbrali 600 zaporednih slik ali 24 sekund dolg izsek. V sledenjem smo sledili vratarja, ki je bil modre barve in se je nahajal pred modrim golom. Enako kakor v prejšnjem poglavju smo s sledilniki $\mathbf{S}_{zs}$ in $\mathbf{S}_{ref}$ petkrat posledili golmana in beležili odpovedi sledenja. Sliko vratarja pred golom prikazuje slika 7.5(a), sliki 7.5(c,d) pa primere dobljenih trajektorij. Iz rezultatov sledenja v tabeli 7.3 vidimo, da se je sledilnik $\mathbf{S}_{zs}$ v težavnem posnetku dobro odrezal, saj je vratarja vedno izgubil le enkrat, medtem ko ga je $\mathbf{S}_{ref}$ v povprečju kar 8.8 krat.

sledilnik	povp. št. odpovedi na sekvenco
S_{ref}	8.8
S_{zs}	1

Tabela 7.3: *Povprečno število odpovedi sledilnikov S_{ref} in S_{zs} med sledenjem golmana v sekvenci 600-tih slik, kar znaša 24 sekund posnetka.*

Glavni vzrok odpovedi sledenja s sledilnikom S_{zs} je bila situacija, ko je bil vratar tako močno podoben ozadju, da je sledilnik pričel slediti le njegove noge, ki so bile različne od ozadja, hkrati pa še vedno dovolj podobne barvnemu modelu igralca. Odpoved sledenja je prikazana v sliki 7.5(c). Sledilnik S_{ref} je imel precej



Slika 7.5: *Primer vratarja pred golom (a) in slika pri kateri je bil vratar tako podoben ozadju, da je sledilnik S_{zs} sledil le njegove noge (b). Primer trajektorije vratarja pridobljene s sledilnikom S_{ref} in S_{zs} prikazujeta sliki (c) in (d). Mesta odpovedi sledilnikov, oziroma mesta posredovanja operaterja, so označena z belimi zvezdicami.*

večje probleme s sledenjem vratarja, saj ga le s težavo razlikoval od ozadja, kar je rezultiralo v zelo slabi lokalizaciji in pogostem odpovedovanju. To je očitno v sliki 7.5(d), v kateri je dobljena trajektorija vratarja močno pošumljena.

7.2.3 Sledenje v *odprtem svetu*

Do sedaj smo vedno govorili o sledenju igralca v zaprtem svetu. Zaprti svet smo zapisali z nekimi predpostavkami na podlagi katerih smo zgradili sledilnik za sledenje enega igralca. S preizkusi smo pokazali, da omenjeni sledilnik zadovoljivo razlaga igralca v zaprtem svetu, rezultat tega pa je dobro sledenje.

Naravno vprašanje, ki se sedaj pojavi, je kako se sledilnik obnaša kadar kršimo nekatere predpostavke. Podrobnega odgovora na to vprašanje ne bomo podali, vendar pa ga bomo osvetlili z nekaj primeri. Ker kršimo predpostavke zaprtega sveta, smo to okolje že v naslovu označili z *odprtim svetom*.

Preizkuse smo izvedli na treh izsekih posnetka rokometne tekme, ki je bila posneta iz tribune s premično ročno kamero; primere slik iz posnetka prikazujejo slike 7.6(a-c).

Kršitve predpostavk zaprtega sveta v izbranih posnetkih lahko povzamemo s sledečimi alinejami:

- Zaradi spreminjanja orientacije kamere in sprememb v povečavi ni bilo mogoče s preprosto časovno mediano kakor v poglavju 5.1 zgraditi dobrega modela ozadja (slika 7.6d)[1].
- Dinamika gibanja igralcev v slikah je bila v posnetkih rezultat tako njihovega premikanja po igrišču kot tudi spreminjanja parametrov kamere (orientacija,povečava). Takega gibanja nismo upoštevali v poglavju 5.6, ko smo načrtovali dinamični model igralca.
- Delna in popolna zakrivanja so zelo pogosta zaradi same postavitve kamere, saj je optična os vse prej kot pravokotna na ravnino igrišča.
- Operater kamere je usmerjal pozornost po svoji presoji le na določene igralce, zato v vsakem trenutku vsi igralci niso prisotni v vidnem polju kamere.

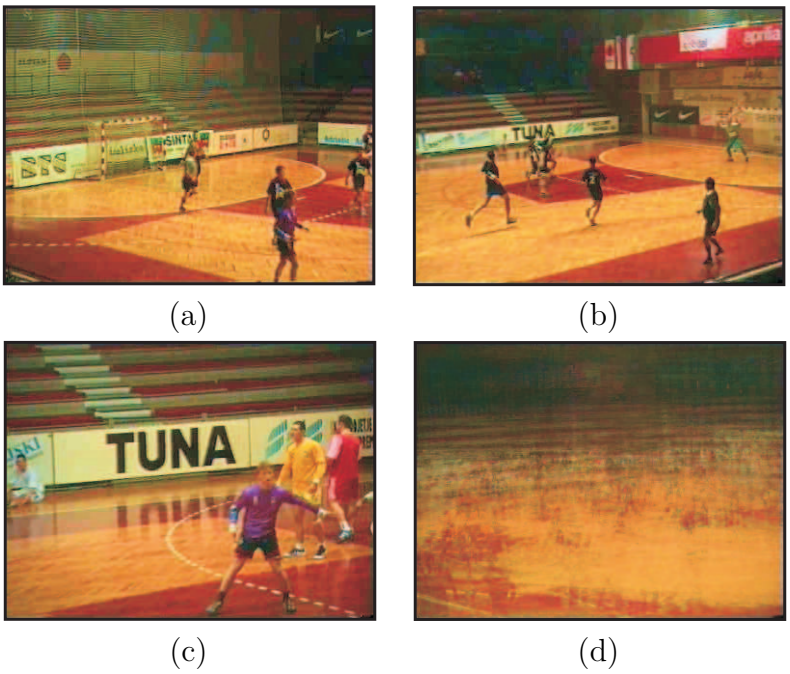
V prvem poskusu smo sledili rumenega igralca skozi sekvenco 462-tih slik (slika 7.7). V sekvenci je pogosto prihajalo do zakrivanj med igralci; taka situacija je prikazana v sliki 7.7(c), kjer je v 128-ti sliki je črni igralec popolnoma prekril rumenega. Ker je bil rumeni igralec dovolj različen od črnega, ga je sledilnik 30 slik kasneje, ko je bil ta zopet dobro viden, uspel lokalizirati (slika 7.7d). Problem se je pojavil 234 slik kasneje (slika 7.7e), ko je igralec izginil iz vidnega polja kamere in se zopet pojavil čez približno 50 slik (slika 7.7e). Kljub temu, da se sledilnik v tem času še ni adaptiral na ozadje ali kakega drugega igralca, je sledenje odpovedalo, saj sledilnik v tistem času ni preiskoval področja slike kjer je igralec vstopil.

Primer uspešne *reinicializacije*[2] prikazuje slika 7.8 kjer smo sledili violičnega igralca. V 150-ti sliki (slika 7.8b) je igralec izginil iz vidnega polja kamere in se zopet pojavil čez 23 slik (slika 7.8c). Sledilnik je uspešno zaznal igralca in sledenje se je nadaljevalo.

Velik problem predstavljajo okluzije med podobnimi igralci. Za primer smo sledili črnega igralca, ko je tekel mimo drugega črnega igralca (slika 7.9). Kmalu

[1] Model ozadja bi sicer lahko zgradili npr. s postopki registracije zaporednih slik in računanja časovne mediane (glej npr. [64, 80]), vendar želimo tu demonstrirati situacijo, ko ozadje slabo modeliramo.

[2] Reinicializacija ni popolnoma upravičen izraz, saj predlagani sledilnik ne vsebuje posebnega mehanizma za neodvisno lokalizacijo in inicializacijo igralca.



(a) (b)

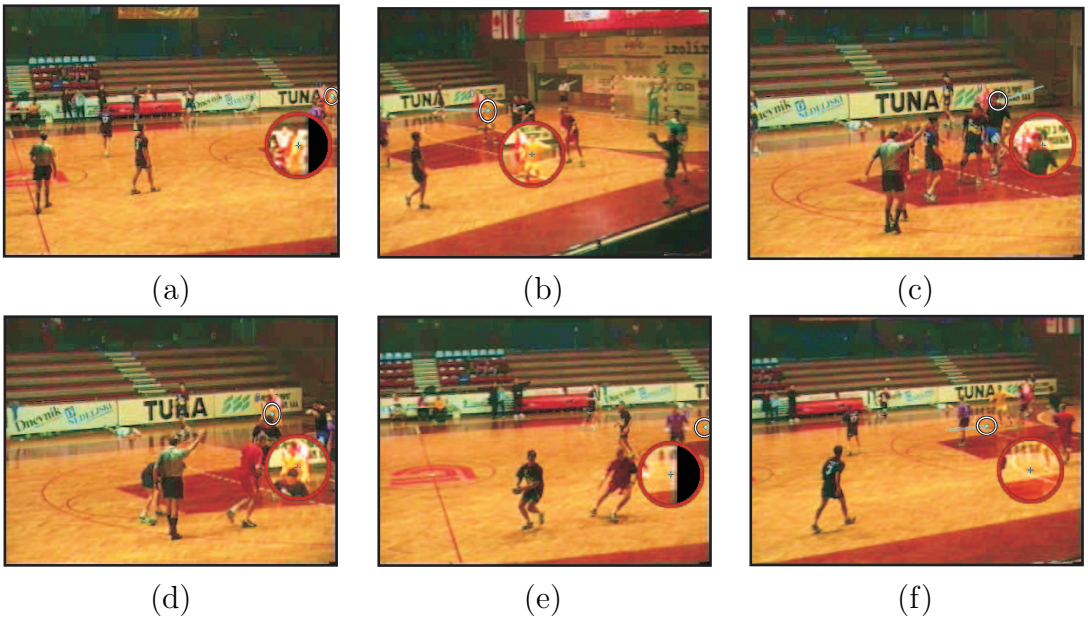
(c) (d)

Slika 7.6: *Primeri slik iz posnetka tekme zajetega s strani igrišča s premično kamero in spremenljivo povečavo (a,b,c). Rezultat je slabo modelirano ozadje (d). Sicer se zdi na prvi pogled model ozadja povsem naključen, vendar če bolje pogledamo, vidimo, da zajame dve glavni lastnosti posnetka: v zgornjem delu slike je v glavnem temna tribuna, medtem ko v spodnjem prevladuje oranžno igrišče.*

po 75-ti sliki (slika 7.9b) sta se igralca delno prekrila, napačni igralec (na sliki 7.9b je to spodnji igralec) je bil bolj podoben referenčnemu barvnemu modelu kot sledeni igralec, in sledilnik je v nadaljevanju (slika 7.9c) obstal na napačnem igralcu.

7.3 Primerjava sledilnikov večih igralcev

Sledilnik za sledenje večih igralcev v zaprtem svetu je sestavljen iz ločenih sledilnikov za enega igralca, ki skupaj simulirajo zaprte svetove vseh igralcev. V poglavju 6.2 smo tako predstavili dva principa povezovanja posameznih sledilnikov v skupnega. Prvi je temeljil na iterativni segmentaciji slike, tako da je vsakemu igralcu pripisal njemu lastno področje in skušal v tem področju igralca lokalizirati. Drugi pristop je temeljil na skrivanju ocenjenih eliptičnih področji, ki v nekem trenutku pripadajo ostalim igralcem. V tem poglavju bomo prikazali rezultate primerjave obeh sledilnikov in jih kometirali.



(a) (b) (c)

(d) (e) (f)

Slika 7.7: *Primeri slik iz video posnetka kjer smo sledili rumenega igralca. Ocena stanja igralca v vsaki sliki je označena z belo elipso, poleg pa je v rdečem okvirju izrisana še povečana slika na mestu centra elipse. Slike si časovno sledijo: slika št.1 (a), slika št.128 (b), slika št.148 (c), slika št.179 (d), slika št.411 (e), slika št.462 (f).*

(a) (b) (c)

Slika 7.8: *Primeri slik iz video posnetka kjer smo sledili violičnega igralca. Igralec je za kratek čas izginil iz vidnega polja kamere, vendar ga je sledilnik zopet uspešno sledil, ko se je igralec zopet prikazal. Ocena stanja igralca v vsaki sliki je označena z belo elipso, poleg pa je v rdečem okvirju izrisana še povečana slika na mestu centra elipse. Slike si časovno sledijo: slika št.130 (a), slika št.150 (b), slika št.173 (c).*

Sledilnike smo preizkušali na dveh posnetkih rokometne tekme, označimo ju s P_1 in P_2, in enem posnetku košarke; slednjega označimo s P_3. Tipične slike iz

(a) (b) (c)

Slika 7.9: *Primeri slik iz video posnetka kjer smo sledili črnega igralca. Igralec je v posnetku tekel mimo podobnega igralca in sledenje je odpovedalo. Ocena stanja igralca v vsaki sliki je označena z belo elipso, poleg pa je v rdečem okvirju izrisana še povečana slika na mestu centra elipse. Slike si časovno sledijo: slika št.69 (a), slika št.75 (b), slika št.92 (c).*

posnetkov prikazuje slika 7.10, ostali relavantni podatki pa so podani v tabeli 7.4.

oznaka posnetka	št. igralcev	dolžina posnetka [slik]	frekv. zajema slik [/s]	velikost slik [slik. el.]
P_1	12	469	25	384×288
P_2	12	1257	25	384×288
P_3	10	430	25	368×288

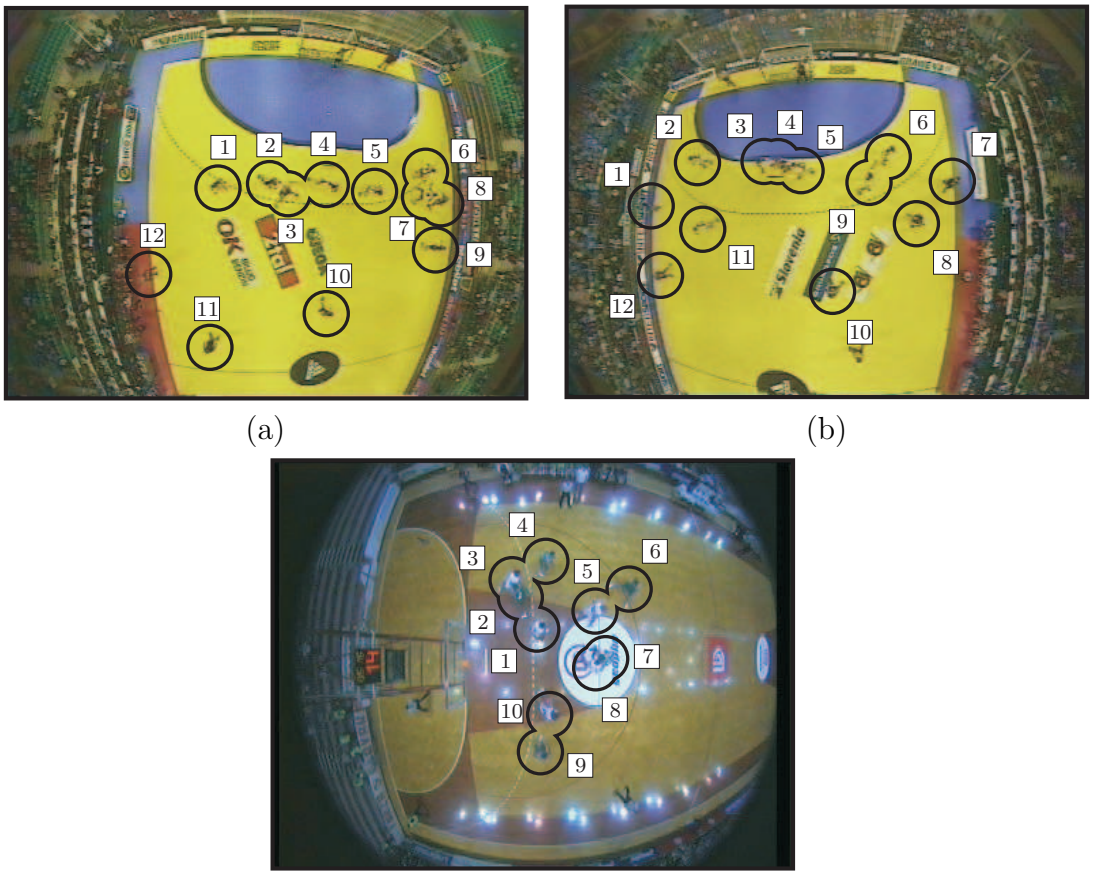
Tabela 7.4: *Podatki za posnetke rokometa (P_1, P_2) in košarke (P_3).*

V vseh treh posnetkih smo sledili igralce s tremi sledilniki, ki jih na kratko označimo z ***A***, ***B*** in ***C***:

- *Sledilnik* ***A*** je predstavljal referenco sledenja brez upoštevanja sosednosti, saj smo vsakega igralca v posnetku sledili s samostojnim sledilnikom za enega igralca (algoritem 5.2). Ta sledilnik je bil torej sestavljen iz popolnoma neodvisnih sledilnikov za enega igralca in ni vseboval mehanizmov za upoštevanje sosednosti med igralci. To pomeni, da lahko kakršnokoli boljše delovanje ostalih testiranih sledilnikov pripišemo njihovi sposobnosti upoštevanja vseh igralcev skupaj.

- *Sledilnik* ***B*** je bil skupni sledilnik, ki je uporabljal Voronoieva področja (algoritem 6.3). Od referenčnega se je razlikoval le v tem, da je vseboval mehanizem za upoštevanje večih igralcev naenkrat.

- *Sledilnik* ***C*** je bil skupni sledilnik, ki je uporabljal metodo skrivanja stanj (algoritem 6.4). Tudi ta se je od referenčnega sledilnika razlikoval le v tem, da je vseboval mehanizem za upoštevanje večih igralcev naenkrat.



Slika 7.10: *Primeri slik iz video posnetkov rokometa P_1 (a), P_2 (b) ter košarke P_3 (c). V posnetkih P_1 in P_2 smo sledili dvanajst igralcev, v posnetku P_3 pa deset. Vsi igralci so označeni s krogom in zaporedno številko.*

Število delcev v vseh sledilnikih posameznih igralcev je bilo 25, prostor velikosti elips pa je bil v posnetkih rokometa omejen na $\{a, b \in [6, 10]\}$ in v posnetku košarke na $\{a, b \in [8, 12]\}$, kjer smo z a in b označili horizontalni in vertikalni premer elipse. Ostali parametri sledilnikov so podani v tabeli 5.1.

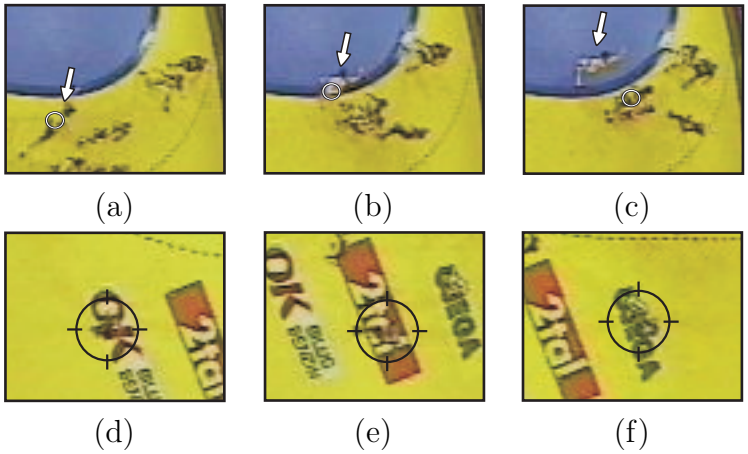
Z vsakim sledilnikom smo po petkrat sledili igralce v treh posnetkih in beležili število odpovedi. Odpoved sledilnika smo obravnavali kot dogodek, ko je ta očitno izgubil enega od igralcev in ga ni mogel več slediti. V takem primeru smo sledenje prekinili, izgubljenega igralca zopet ročno označili in nadaljevali s sledenjem. Na podlagi petih poskusov smo izračunali povprečno število odpovedi sledenja dvanajstih oziroma desetih igralcev v sekvencah P_1, P_2 in P_3. Rezultate prikazuje tabela 7.5, kjer smo zaradi boljšega razumevanja izračunali še povprečno število odpovedi sledilnikov na minuto.

Sledilnik	O_{vseh} [/*izsek*]	O_{vseh} [/*min*]
Posnetek P_1 (12 igralcev, 469 slik)		
A	24.8	38.4
B	1.2	1.8
C	1.9	3
Posnetek P_2 (12 igralcev, 1257 slik)		
A	29.2	34.8
B	7.2	8.6
C	8.2	9.8
Posnetek P_3 (10 igralcev, 430 slik)		
A	4.4	15.4
B	1	3.5
C	0	0

Tabela 7.5: *Rezultati uspešnosti sledenja v izsekih P_1, P_2 in P_3 s sledilniki **A**,**B** in **C**. V tabeli smo zapisali povprečno število intervencij (O_{vseh}) v petih poskusih sledenja z vsakim sledilnikom v vsakem posnetku. V posnetkih P_1 in P_2 smo sledili dvanajstim igralcem, v posnetku P_3 pa desetim.*

V vseh treh posnetkih je sledilnik ***A*** deloval najslabše, saj je praktično ob vsakem trku med podobnimi igralci odpovedal. Prav tako so sledilniku povzročale probleme situacije kakršno prikazujejo slike 7.11(a-c). V teh slikah se bel igralec giblje po rumeni podlagi in je zaradi vpliva podlage rumenkaste barve. Problem nastopi v sliki 7.11(b), ko se igralec premakne na modro podlago in nekoliko spremeni barvo; ker se v njegovi bližini nahaja na rumeni podlagi še en igralec iz istega moštva, sledilnik preskoči nanj in sledenje odpove. Sicer se je sledilnik obnašal precej dobro, kadar so bili podobni igralci dovolj narazen; slike 7.11(d-f) prikazujejo take primere, kjer se je igralec le malo razlikoval od ozadja, vendar sledenje vseeno ni odpovedalo.

Kar se zadeva števila odpovedi sta bila sledilnika ***B*** in ***C*** v vseh posnetkih približno izenačena. V prvih dveh posnetkih, P_1 in P_2, se je nekoliko bolje odrezal sledilnik ***B***, v tretjem, P_3, pa sledilnik ***C***. Boljše delovanje sledilnika ***B*** si lahko razlagamo kot posledico njegove robustnosti na slabo oceno trenutnih stanj igralcev: Kadar v sledilniku C izvajamo iteracijo sledenja za določenega igralca, je potrebno zelo dobro oceniti trenutne elipse ostalih igralcev, saj s temi gradimo igralčev zaprti svet. Drugače je pri sledilniku **B**, kjer za izgradnjo zaprtih svetov uporabljamo le ocene centrov igralcev ter pravilo NN. Slednji je zato na nek način neobčutljiv na napake v oceni velikosti elips. Seveda pa to pomeni tudi da lahko odpove vsakič, ko so predpostavke za uporabo pravila najbližji sosed prekršene. To je bilo zelo očitno v primeru sledenja določenega igralca košarke v izseku P_3, kjer je sledilnik ***B*** vedno na istem mestu odpovedal, medtem ko sledilnik ***C***. Ta primer je prikazan v sliki 7.12, kjer temnega igralca za trenutek popolnoma



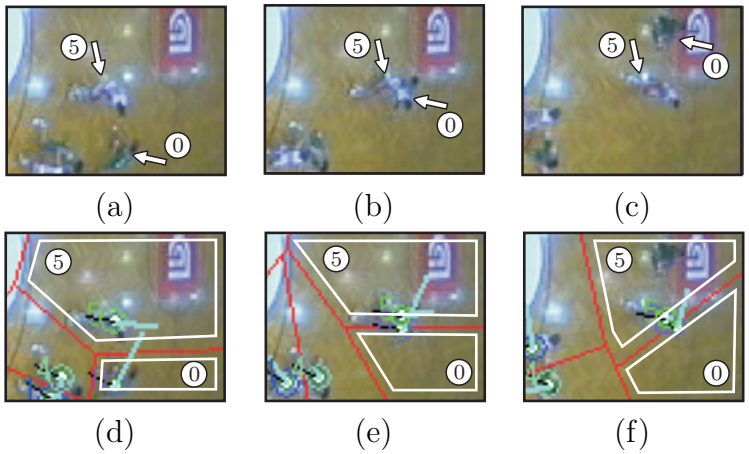
Slika 7.11: *Sledilnik **A** odpove, ko se igralcu ob prehodu na drugačno podlago spremeni barva, v njegovi bližini pa se nahaja igralec iz istega moštva (a-c). Ocenjeno stanje smo v slikah (a-c) označili z belo elipso, sledenega igralca pa s puščico; odpoved sledilnika prikazuje slika (c), kjer puščica in elipsa ne označujeta več istega igralca. Kadar so si bili podobni igralci dovolj narazen je sledilnik deloval precej dobro. Težavne primere prehoda na drugačno podlago, v katerih sledenje ni odpovedalo, prikazujejo slike (d-f), kjer smo položaj igralca v sliki označili s krogom.*

prekrije igralec iz nasprotnega moštva, po prekritju pa temni nadaljuje gibanje v taki smeri, da se več ne nahaja v oceni njegovega zaprtega sveta; sledilnik ga torej več ne *vidi* in sledenje odpove.

Za direktno primerjavo smo na podlagi rezultatov v tabeli 7.5 izračunali še pričakovano število odpovedi sledilnikov na minuto, če bi sledili dvanajst igralcev (tabela 7.6). Ker je pomembna lastnost sledilnika tudi njegova časovna potratnost, smo za vse sledilnike izmerili čas, ki je bil potreben za izvršitev ene časovne iteracije sledenja dvanajstih igralcev.

Med vsemi sledilniki je bil najhitrejši sledilnik ***A***, saj ne upošteva več igralcev naenkrat in v ta namen torej ne porablja dodatne računske moči. Kot smo že ugotovili iz tabele 7.5 ima to neupoštevanje vseh igralcev skupaj tudi slabo stran in sicer veliko število odpovedi sledilnika. Tako pričakujemo v eni minuti sledenja dvanajstih igralcev s sledilnikom ***A*** v povprečju kar 34 odpovedi.

Število pričakovanih odpovedi krepko pade z upoštevanjem vseh igralcev skupaj. V eni minuti sledenja dvanajstih igralcev s sledilnikom ***B*** pričakujemo v povprečju pet odpovedi, pri uporabi sledilnika ***C*** pa šest (tabela 7.6). Tudi ta rezultat ni presenetljiv, saj smo že v tabeli 7.5 videli, da sta sledilnika ***A*** in ***B*** dokaj izenačena kar se tiče odpovedovanja. Precejšna razlika pa je v času, ki ga sledilnika potrebujeta za eno iteracijo. Sledilnik ***B*** je porabil za procesiranje



Slika 7.12: *Primer odpovedi sledenja s sledilnikom **B**, ko med svetlim igralcem (št.5) in temnim (št.0) pride do popolne okluzije. Oba igralca smo v slikah (a-c) označili s puščicami. V slikah (d-f) smo izrisali ločilne meje med sledenimi igralci in z belimi okvirji označili Voronoieva področja za igralca št.(5) in št.(0). Ker je po okluziji področje igralca št.(0) napačno ocenjeno (f), sledilnik ne vidi več tega igralca in sledenje odpove.*

Sledilnik	$\hat{O}_{12}$ $[/min]$	T_{12} $[min]$
A	34	1.99
B	5	7.31
C	6	2.00

Tabela 7.6: *Pričakovano število odpovedi sledilnikov na minuto pri sledenju dvanajstih igralcev $\hat{O}_{12}$ in pričakovan čas T_{12}, ki ga sledilniki potrebujejo za obdelavo minute posnetka. Število odpovedi $\hat{O}_{12}$ smo izračunali iz rezultatov v tabeli 7.5, pričakovane čase T_{12} pa smo posebej določili s poskusom, kjer smo sledili dvanajst igralcev. Sledilnika **A** in **C** za obdelavo minute posnetka potrebujeta približno dve minuti, medtem ko **B** za enak posnetek potrebuje več kot sedem minut.*

ene minute dolgega posnetka namreč več kot trikrat več časa kot sledilnik ***C***. Sledenje s sledilnikom ***C*** je torej praktično enako hitro kot sledenje z ***A***, medtem ko je pričakovano število odpovedi skoraj enako kot pri ***B***. Iz tega stališča se zdi v problemih sledenja večih igralcev izbira sledilnika ***C*** najboljša, saj slednji združuje tako hitrost sledenja, kot tudi nizek odstotek odpovedovanja.

Poglavje 8

Zaključek

Sekvenčne Monte Carlo metode ali filtri z delci so statistični pristopi k ocenjevanju dinamičnih procesov, ki temeljijo na predstavitvi a posteriori pdf trenutnega stanja sistema z množico naključnih uteženih delcev. Dinamični model sistema v filtrih z delci sledi Markovovem modelu prvega reda, kar je včasih tudi vzrok za zmotno prepričanje, da so filtri z delci pravzaprav Monte Carlo metode z Markovovimi verigami *(angl. Markov Chain Monte Carlo)*; v resnici gre za Monte Carlo metode *v* Markovovi verigi. V zadnjem desetletju, po objavi algoritma CONDENSATION, so se te metode uveljavile kot uspešno orodje pri reševanju problemov sledenja na podlagi vizualne informacije.

Primer uporabe sekvenčnih MC metod za sledenje ljudi smo prikazali na sledilniku za sledenje igralcev v moštvenih športih. Športno igro smo najprej predstavili v kontekstu zaprtega sveta (ZS) z naborom pravil, ki se nanašajo na splošno igro moštvenega športa. Nato smo z izbiro dinamičnega modela igralca in njegove vizualne značilnice določili referenčni sledilnik za enega igralca, ter mu s postopki modeliranja določili vse parametre. Ta sledilnik je služil kot ilustrativni primer uporabe algoritma CONDENSATION in je v nadaljevanju služil za pridobivanje podatkov pri gradnji bolj robustnega sledilnika.

Upoštevajoč predpostavke zaprtega sveta smo ugotovili, da lahko zgradimo model ozadja ter da je to informacijo smiselno uporabiti na različnih nivojih sledenja. Tako smo določili robustno oceno prisotnosti igralca z upoštevanjem ozadja in metodo časovne adaptacije igralčevega izgleda. S predpostavko, da igralčevo gibanje ni popolnoma nakjučno, smo določili nov dinamični model igralca. Vse te izboljšave smo povezali v sledilnik za sledenje enega igralca v zaprtem svetu.

Problem sledenja večih igralcev smo umestili v problematiko sledenja večih tarč in omenili nekatere v literaturi obstoječe rešitve. Ker o športni igri govorimo kot o zaprtem svetu, smo vpeljali še dve dodatni predpostavki in v skladu s temi predpostavkami prikazali dva načina povezovanja samostojnih sledilnikov za enega igralca v skupni sledilnik za sledenje večih igralcev v zaprtem svetu. Oba

postopka sta temeljila na maskiranju področji, ki v nekem trenutku ne pripadajo sledenemu igralcu. Prvi način je temeljil na postopku generiranja Voronoievih področji, medtem ko je drugi temeljil na skrivanju stanj igralcev.

Vse predstavljene sledilnike smo nazadnje ovrednotili s številnimi preizkusi. Tako smo sledilnik za sledenje enega igralca v ZS primerjali z referenčnim in ugotovili, da deluje občutno bolje, rezultat pa je manjše število odpovedi med sledenjem. Prednosti slednjega so se še posebej izkazale v težavnem primeru sledenja vratarja pred golom. Z dodatnimi poskusi smo prikazali tudi delovanje sledilnika za enega igralca, kadar nekatere predpostavke ZS ne držijo, in ugotovili, da sledilnik deluje dokaj dobro; opozorili smo tudi na pomanjkljivosti in probleme.

Predlagana sledilnika za sledenje večih igralcev v ZS smo na treh posnetkih športne igre primerjali z delovanjem skupine neodvisnih sledilnikov za enega igralca. Rezultati so pokazali občutno boljše delovanje prvih dveh. Primerjali smo še rezultate sledenja slednjih in ugotovili, da je sledilnik, ki je temeljil na skrivanju stanj deloval najhitreje in približno enako uspešno kot tisti, ki je temeljil na Voronoievih področjih. Glede na te rezultate smo v nadaljevanju predlagali uporabo sledilnika za sledenje večih igralcev v ZS, ki je temeljil na metodi skrivanja stanj.

Kot opombo omenimo, da je hitrost izvršitve iteracije sledenja lahko nekoliko zavajujoč podatek, saj je po avtorjevih izkušnjah sledenje več kot treh igralcev naenkrat dokaj težavno za operaterja. Težavnost je še toliko večja, če se ti igralci ne gibljejo skupaj. V problematičnih situacijah mora operater nameniti zelo veliko pozornosti trkom med igralci in tega ne more početi s poljubno veliko hitrostjo. To pomeni, da sta hitrost sledenja in število igralcev, ki jih sledimo naenkrat omejena že samo zaradi človeškega faktorja. Po avtorjevih izkušnjah je operaterjevo nadzorovanje pravilnosti sledenja najlažje pri sledilniku, ki temelji na Voronoievih področjih. Vzrok je v tem, da izris Voronoievih področji nudi dobro orodje za orientacijo dogajanja na igrišču. To pomeni, da na nek način razdalje in odnose med igralci s temi področji dobro vizualiziramo ter s tem potencialnemu uporabniku olajšamo nadzorovanje pravilnosti sledenja.

8.1 Smernice za nadaljnji razvoj

Filtri z delci nudijo precej splošen pristop za integracijo različnih virov informacije v skupnega s pomočjo katerega lahko realiziramo sledenje. Glavni pogoj je le, da so ti viri medsebojno kalibrirani. To pomeni, da moramo poznati preslikave vsaj znotraj neke omejene napake, med krajevnimi komponentami teh virov[1]. Najbolj očitna razširitev bi torej bila uporaba večih statičnih kamer v vidnem spektru, saj bi bil verjetno sledeni igralec med večino trkov in zakrivanj dobro

[1] V preprostem primeru dveh kamer bi to bila npr. homografija med kamerama.

viden vsaj eni kameri. Smislena se zdi tudi uporaba kamer, ki delujejo v različnih delih spektra svetlobe; npr. ultra vijolični ali infra rdeči. Dosedanje izkušnje, ki jih imamo z uporabo kamere v infra rdečem spektru kažejo, da se pri sledenju ljudi s tako kamero pojavljajo problemi enake narave kot pri kameri, ki deluje v vidnem spektru, vendar pa ne nujno na enakih mestih. To pomeni, da skupno upoštevanje vizualnih lastnosti oseb v različnih spektrih lahko izboljšajo sledenje. Prav tako ni nobenega razloga zakaj nebi uporabili drugačne modalitete, kot so npr. zvok, ultra-zvok ali radio-frekvenčni valovi. Opomnimo, da ni strašanske potrebe za posebno natančnost informacije teh posameznih modalitet, saj bi se skupaj dopolnjevale in na tak način izboljšale sledenje. Dodatno bi lahko bili nekateri viri informacij nestatični (npr. premična kamera) in bi lahko svojo pozornost usmerjali v dele igrišča, kjer bi potrebovali več informacije za pojasnitev dogajanja.

Trenutna implementacija sledilnika za sledenje igralcev vsebuje fiter z delci, ki v vsaki iteraciji uporablja postopek prevzorčenja delcev. Kot smo že omenili, vsakokratno prevzorčenje ni potrebno in glede na nekatere avtorje celo nepriporočljivo. V nadaljnjem delu bi se zato morda bilo smiselno nekoliko posvetiti temu problemu. Prav tako bi bilo smiselno vpeljati bolj kompleksen sistem za izgradnjo ozadja, npr. tak, ki bi se časovno prilagajal spremenljivi osvetljavi na igrišču in bi bil sposoben boljšega izločanja ozadja iz trenutnih slik. Trenutno je adaptacija vizualnih modelov igralcev implementirana tako, da se vrši tudi med trki in okluzijami. V nadaljnjem delu bi bilo smiselno vpeljati algoritem, ki bi med trki igralcem avtomatično izklopil adaptacijo.

Vse te posodobitve in potencialne izboljšave algoritmov zahtevajo dodatne raziskave, preizkuse ter nenazadnje postopke za sistematično določevanje novih morebitnih parametrov. Na primer. Kakšen je vpliv uporabe različnih kombinacij modalitet na sledenje? Kako implementirati ustrezne kalibracije znotraj in preko modalitet? Kako nastaviti pragovno vrednost efektivne velikosti vzorca za proženje prevzorčenja? Kako določiti primerne statistike slikovnih elementov v modelu ozadja? Kako realizirati njegovo adaptacijo? Pri kakšni razdalji med igralci izklopiti adaptacijo? itd. V nadaljnjem delu bi bilo smiselno odgovoriti vsaj na del teh vprašanj, z njimi pa tudi zaključujemo magistrsko nalogo.

Literatura

[1] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83:95–107, 1961.

[2] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[3] Prozone. Spletni naslov: http://www.pzfootball.com/, September 2005.

[4] D. Setterwall. Computerised video analysis of football– technical and commercial possibilities for football coaching. Master's thesis, Centre for User Oriented IT Design, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Universitet Stockholms, 2003.

[5] A. Doucet, N. de Feritas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[6] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

[7] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, (10):197–208, 2000.

[8] A. H. Jazwinski. *Stochastic processes and filtering theory*. Academic Press, New York, 1970.

[9] D. L. Alspach and H. W. Sorenson. Nonlinear bayesian estimation using gaussian sum approximation. *IEEE Trans. Auto. Control*, 20:107–110, 1975.

[10] A. Pole and M. West. Efficient numerical integration in dynamic models. Technical report, Department of statistics, University of Warwick, 1988.

[11] G. Kitagawa. Non-gaussian state-space modelling of nonstationary time series. *Journal of American Statistical Association*, 82:1032–1063, 1987.

[12] S. M. Herman. *A particle filtering approach to joint passive radar tracking and target classification*. PhD thesis, University of Illinois, 2002.

[13] J. MacCormick. *Probabilistic modelling and stochastic algorithms for visual localisation and tracking*. PhD thesis, Department of Engineering Science, University of Oxford, January 2000.

[14] J. S. Liu and R. Chen. Sequential monte carlo methods for dynamical systems. *J. Amer. Statist. Assoc.*, 93:1032–1044, 1998.

[15] G. Kitagawa. Monte carlo filter and smoother for non-gaussian non-linear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[16] N. J. Gordon and D. J. Salmond an A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. volume 40, pages 107–113, April 1993.

[17] S. S. Intille and A. F. Bobick. Visual tracking using closed-worlds. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 672–678, MIT, Cambridge, MA, June 1995.

[18] J. Perš, M. Bon, S. Kovačič, M. Šibila, and B. Dežman. Observation and analysis of large-scale human motion. *Human Movement Science*, 21(2):295–311, July 2002.

[19] A. Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics,Vision,Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[20] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[21] A. Baumberg and D. Hogg. An efficient method for contour tracking using active shape models. In *IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pages 194–199. IEEE Press, 1994.

[22] Vasanth Philomin, Ramani Duraiswami, and Larry S. Davis. Quasi-random sampling for condensation. In *European Conference on Computer Vision*, volume 2, pages 134–149, London, UK. Springer-Verlag.

[23] D. Tweed and A. Calway. Tracking multiple animals in wildlife footage. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 24–27, 2002.

[24] M. Xu, J. Orwell, and G. Jones. Tracking football players with multiple cameras. In *EEE Conference on Image Processing*, pages V: 2909–2912, 2004.

[25] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 193–199, June 1997.

[26] J. Perš. Sledenje ljudi z metodami računalniškega vida. Master's thesis, Fakulteta za elektrotehniko, Univerza v Ljubljani, April 2001.

[27] I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, August 2000.

[28] S. S. Initille and A. F. Bobick. Closed-world tracking. *In Proc. IEEE International Conference on Computer Vision*, pages 672–678, June 1995.

[29] Y. Seo, S. Choi, H. Kim, and K.S. Hong. Where are the ball and players? soccer game analysis with color based tracking and image mosaick. *Proceedings of the 9th International Conference on Image Analysis and Processing*, 2:196–203, September 1997.

[30] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. volume 1, pages 28–39, 2004.

[31] S.S. Intille, J.W. Davis, and A.F. Bobick. Real-time closed-world tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 697–703, June 1997.

[32] K. Choi and Y. Seo. Probabilistic tracking of the soccer ball. In *Statistical Methods in Video Processing*, pages 50–60, 2004.

[33] H. OK, Y. Seo, and K. Hong. Multiple soccer players tracking by condensation with occlusion alarm probability. In *International Workshop on Statistically Motivated Vision Processing*, 2002.

[34] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In *European Conference on Computer Vision*, volume 1, pages 893–908, 1998.

[35] S.J. McKenna, Y. Raja, and S. Gong. Object tracking using adaptive color mixture models. In *Proceedings of Asian Conference on Computer Vision*, volume 1, pages 615–622, January 1998.

[36] C.P. Town and S.J. Moran. Robust fusion of colour appearance models for object tracking. In *Proceedings of British Machine Vision Conference*, 2004.

[37] M.J. Swain and D.H Ballard. Colour indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.

[38] F. Pitié, S.A. Berrani, R. Dahyot, and A. Kokaram. Off-line multiple object tracking using candidate selection and the viterbi algorithm. *Proc. of the IEEE International Conference on Image Processing*, September 20005.

[39] K. Choi, Y.D. Seo, and S.W. Lee. Probabilistic tracking of soccer players and ball. *Proc. 6th Asian Conference on Computer Vision*, pages 27–30, January 2004.

[40] K. Nummiaro, E. Koller-Meier, and L.J. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, January 2003.

[41] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multi-modality through mixture tracking. In *International Conference on Computer Vision*, volume 1, pages 110–116, Nice, France, June 2003.

[42] S.J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, pages 348–353. IEEE Computer Society, March 2000.

[43] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *7th European Conference on Computer Vision*, volume 1, pages 661–675. Springer-Verlag, 2002.

[44] L. Sigal, S. Sclaroff, and V. Athitsos. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 152–159, 2000.

[45] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of IEEE*, 92(3):495–513, January 2004.

[46] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. pages 142–151, 2000.

[47] A. del Bimbo. *Visual Information Retrieval*. Academic Press, San Francisco, USA, 1999.

[48] S. Liapis and G. Tziritas. Color and texture image retrieval using chromaticity histograms and wavelet frames. *IEEE Transactions on Multimedia*, 6(5):676–686, October 2004.

[49] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, February 1967.

[50] F. Aherne, N. Thacker, and P. Rockett. The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 32(4):1–7, 1997.

[51] M. Basseville. Distance measures for signal processing and pattern recognition. *Signal Processing*, 18(4):349–454, December 1989.

[52] I.J. Taneja. Refinement inequalities among symmetric divergence measures. *The Australian Journal of Mathematical Analysis and Applications*, 1(2):1–23, April 2005.

[53] H. Tong. *Non-linear time series: A dynamical system approach.* Oxford University Press, 1990.

[54] M. Bramble, Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *International Conference on Computer Vision (ICCV)*, pages 34–41, 2001.

[55] E. Koller-Meier and F. Ade. Tracking multiple objects using the condensation algorithm. *Journal of Robotics and Autonomous Systems*, 34:93–105, February 2001.

[56] C.J. Needham. *Tracking and Modelling of Team Game Interactions.* PhD thesis, School of Computing, The University of Leeds, October 2003.

[57] M. Bon, J. Perš, M. Šibila, and S. Kovačič. *Analiza gibanja igralca med tekmo*, chapter 5, page 162. Fakulteta za šport, Univerza v Ljubljani, Ljubljana, 1 edition, 2002.

[58] M. Bon, J. Perš, M. Šibila, and S. Kovačič. *Analiza gibanja igralca med tekmo*, chapter 5, page 158. Fakulteta za šport, Univerza v Ljubljani, Ljubljana, 1 edition, 2002.

[59] J. Bangsbo. The physiology of soccer: With special reference to intense intermittent exercise. *Acta Physiologica Scandinavica, Suppl.*, 619:1–155, 1994.

[60] W. S. Erdmann. Gathering of kinematic data of sport event by televising the whole pitch and track. pages 159–162, 1992.

[61] C. J. Needham and R. D. Boyle. Tracking multiple sports players through occlusion, congestion and scale. In *12th British Machine Vision Conference, BMVC01*, pages 93–102, Manchester, UK, September 2001.

[62] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, New York, NY, USA, 1992.

[63] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, 8(1):3–30, January 1998.

[64] I. Lesjak. Sledenje in analiza vizualne informacije na posnetkih športnih dogodkov, magistrsko delo. Master's thesis, Fakulteta za računalništvo in informatiko, Ljubljana, May 2001.

[65] L. Sigal, S. Sclaroff, and V. Athitsos. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. volume 2, pages 152–159, 2000.

[66] Y. Raja, S. Mckenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using colour. pages 228–233, 1998.

[67] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, October 2003.

[68] M. Bon, J. Perš, M. Šibila, and S. Kovačič. *Analiza gibanja igralca med tekmo*, chapter 5, page 159. Fakulteta za šport, Univerza v Ljubljani, Ljubljana, 1 edition, 2002.

[69] K.P. Burnham and D.R. Anderson. Multimodel inference: Understanding aic and bic in model selection. *Sociological Methods and Research*, 33(2):261–304, 2004.

[70] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, December 1974.

[71] S. Kullback and R. A. Lieber. On information and sufficiency. *Annals of Mathemathical Statistics*, 22:679–86, 1951.

[72] S. Särkkä, A. Vehtari, and J. Lampinen. Rao-Blackwellized Monte Carlo data association for multiple target tracking. In *Proceedings of the Seventh International Conference on Information Fusion*, volume 1, pages 583–590, June 2004.

[73] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 665–670, 2001.

[74] C. Hue, J.-P. Le Cadre, and P. Pérez. A particle filter to track multiple objects. In *IEEE Workshop on Multi-Object Tracking*, pages 61–68, Vancouver, Canada, July 2001.

[75] G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46:167–174, 1992.

[76] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.

[77] Z. Khan, T. Balch, and F. Dellaert. Efficient particle filter-based tracking of multiple interacting targets using an mrf-based motion model. *Intl. Conf. on Intelligent Robots and Systems*, 1:254–259, October 2003.

[78] M. Xu and T. Ellis. Partial observation vs. blind tracking through occlusion. *Proceedings of the British Machine Vision Conference*, pages 777–786, September 2002.

[79] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* A Wiley-Interscience Publication, 2nd edition, 2000.

[80] K. Okuma, J.J. Little, and D. Lowe. Automatic acquisition of motion trajectories: Tracking hockey players. *Internet Imaging V, SPIE*, 15:202–213, December 2004.

Dodatek A

Prioritetno vzorčenje

V poglavju 2.3.2 smo pokazali kako je postopek prioritetnega vzorčenja uporabljen v filtrih z delci kot metoda pridobivanja delcev iz željene porazdelitve. Sicer je ta postopek klasična metoda za ocenjevanje integralov z Monte Carlo pristopi, iz česar v poglavju 2.3.2 tudi izhajamo; zato bomo na tem mestu nekoliko bolj natančno opisali idejo za tem pristopom ter razlago prioritetnega vzorčenja kot integracijske metode.

Naj bo ($\mathbf{x} \in \mathcal{X}$) *naključna spremenljivka* (n.s.) v prostoru $\mathcal{X}$ in naj bo $p(\mathbf{x})$ njena pdf. Dalje, naj bo $f(x) : \mathcal{X} \to \mathcal{R}^n$ neka funkcija definirana na prostoru $\mathcal{X}$, ki slika v prostor realnih števil $\mathcal{R}^n$. Zanima nas vrednost integrala v obliki

$$I_f = \int_{\mathcal{X}} f(\mathbf{x})p(\mathbf{x})d\mathbf{x}. \qquad \text{(Dodatek A.1)}$$

Predpostavimo, da lahko pdf $p(\mathbf{x})$ določimo vsaj do proporcionalne konstante natančno; torej lahko določimo vrednost $P(\mathbf{x})$ tako, da velja

$$P(\mathbf{x}) = C_p \cdot p(\mathbf{x}),$$

kjer je C_p neka konstanta. Recimo, da je $P(\mathbf{x})$ porazdelitev, ki je preveč kompleksna, da bi jo lahko vzorčili direktno. Torej, analitična rešitev integrala ne obstaja ali pa je zelo težko izračunljiva, aproksimacija z uporabo klasičnega Monte Carlo vzorca pa je nemogoča, ker $p(\mathbf{x})$ ne moremo vzorčiti.

Sedaj predpostavimo preprostejšo porazdelitev $q(\mathbf{x})$, ki je različna od nič vsaj tam, kjer je različna od nič tudi $p(\mathbf{x})$. Pravimo, da $q(\mathbf{x})$ vsebuje podporo porazdelitve $p(\mathbf{x})$. Zaradi splošnosti recimo, da lahko $q(\mathbf{x})$ prav tako določimo le do proporcionalne konstante natančno; torej poznamo tak $Q(\mathbf{x})$, da velja

$$Q(\mathbf{x}) = C_q \cdot q(\mathbf{x}),$$

vendar, za razliko od $P(\mathbf{x})$ lahko $Q(\mathbf{x})$ vzorčimo enostavno.

Integral (Dodatek A.1) se ne spremeni, če notranji del pomnožimo in delimo z enako funkcijo, zato lahko pišemo:

$$\begin{aligned} I_f &= \int_{\mathcal{X}} f(\mathbf{x})q(\mathbf{x})\frac{1}{q(\mathbf{x})}p(\mathbf{x})d\mathbf{x} \\ &= \frac{1}{C_p}\int_{\mathcal{X}} f(\mathbf{x})Q(\mathbf{x})w(\mathbf{x})d\mathbf{x}, \end{aligned} \qquad \text{(Dodatek A.2)}$$

kjer smo z $w(\mathbf{x})$ označili

$$w(\mathbf{x}) = \frac{P(\mathbf{x})}{Q(\mathbf{x})}.$$

Konstanto C_p lahko zapišemo kot

$$C_p = \int_{\mathcal{X}} P(\mathbf{x})d\mathbf{x} = \int_{\mathcal{X}} P(\mathbf{x})\frac{1}{Q(\mathbf{x})}Q(\mathbf{x})d\mathbf{x} = \int_{\mathcal{X}} w(\mathbf{x})Q(\mathbf{x})d\mathbf{x}$$

in izraz (Dodatek A.2) se spremeni v

$$I_f = \frac{1}{\int_{\mathcal{X}} w(\mathbf{x})Q(\mathbf{x})d\mathbf{x}}\int_{\mathcal{X}} f(\mathbf{x})w(\mathbf{x})Q(\mathbf{x})d\mathbf{x}. \qquad \text{(Dodatek A.3)}$$

Na ta izraz lahko gledamo kot integral funkcije $\frac{1}{C_p}f(\mathbf{x})w(\mathbf{x})$ preko porazdelitve $Q(\mathbf{x})$. Ker porazdelitev $Q(\mathbf{x})$ lahko vzorčimo preprosto, integral (Dodatek A.3) rešimo s standardnim MC vzorčenjem.

Naj $\{\mathbf{x}^{(i)}\}_{i=1}^N$ predstavlja N delcev vzorčenih iz porazdelitve $Q(\mathbf{x})$. Empirični približek porazdelitvi $Q(\mathbf{x})$ zapišemo kot

$$\hat{Q}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathbf{x}^{(i)}}(\mathbf{x}),$$

kjer je $\delta_{\mathbf{x}^{(i)}}(\mathbf{x})$ Dirakov delta usrediščen na delcu $\mathbf{x}^{(i)}$ in aproksimacija integrala (Dodatek A.1) se potem glasi

$$\begin{aligned} \hat{I}_f &= \frac{1}{\int_{\mathcal{X}} w(\mathbf{x})\hat{Q}(\mathbf{x})d\mathbf{x}}\int_{\mathcal{X}} f(\mathbf{x})w(\mathbf{x})\hat{Q}(\mathbf{x})d\mathbf{x} \\ &= \frac{1}{\frac{1}{N}\sum_{i=1}^{(i)} w(\mathbf{x}^{(i)})}\frac{1}{N}\sum\nolimits_{i=1}^{N} f(\mathbf{x}^{(i)})w(\mathbf{x}^{(i)}), \end{aligned}$$

ali drugače,

$$\hat{I}_f = \sum\nolimits_{i=1}^{N} f(\mathbf{x}^{(i)})\hat{w}(\mathbf{x}^{(i)}), \qquad \text{(Dodatek A.4)}$$

kjer je

$$\hat{w}(\mathbf{x}^{(i)}) = \frac{w(\mathbf{x}^{(i)})}{\sum_{i=1}^{N} w(\mathbf{x}^{(i)})}. \qquad \text{(Dodatek A.5)}$$

Funkciji $Q(\mathbf{x})$ pravimo *prioritetna* ali *vzorčevalna* funkcija *(angl. importance function, sampler function)*. Ime *vzorčevalna* si lahko razlagamo kot dejstvo, da preko nje vzorčimo delce s katerimi ocenjujemo integral (Dodatek A.1), ime *prioritetna* pa izhaja iz dejstva, da njena oblika narekuje v katerih delih prostora $\mathcal{X}$ se bo generirala glavnina teh delcev. V literaturi filtriranja z delci se glavnem uporablja ime *prioritetna (angl. importance)* funkcija.

Aproksimacija integrala v obliki (Dodatek A.1) z uporabo prioritetnega vzorčenja je torej preprosta: določimo prioritetno funkcijo $Q(\cdot)$, ki jo vzorčimo N-krat in vsakemu vzorčenemu delcu pripišemo utež po enačbi (Dodatek A.5). Ocena integrala (Dodatek A.1) se potem izračuna po enačbah (Dodatek A.4) in (Dodatek A.5).

Dodatek B

Inverzna metoda vzorčenja

Naj bo $f : \Omega \to \mathbb{R}^+$ *funkcija porazdelitve gostote verjetnosti* (pdf) z definicijskim območjem $\Omega \subset \mathbb{R}$ in naj bo $F_X(x)$ *funkcija porazdelitve kumulativne gostote verjetnosti* (cdf) *(angl. cumulative probability density function)*

$$F_X(x) = \int_{-\infty}^{x} f(y)dy.$$

Vzorčenje iz $f(x)$ pomeni generiranje vrednosti take n.s. X, da velja

$$P(X \leq x) = F_X(x).$$

Naj bo U enakomerno porazdeljena n.s. na intervalu $[0, 1]$ in naj bo $F_X^{-1}(\cdot)$ inverzna funkcija definirana tako, da velja

$$F_X^{-1}(F_X(x)) = x.$$

Če definiramo n.s. $X = F_X^{-1}(U)$, potem je njena pdf enaka $F_X(x)$.

Dokaz. Funkcija porazdelitve kumulativne gostote verjetnosti enakomerno porazdeljene n.s. je

$$F_U(u) = P(U \leq u) = \int_{-\infty}^{u} 1dx = u,$$

cdf n.s. X, definirane kot $X = F_X^{-1}(U)$, pa je

$$F_X(x) = P(X \leq x) = P(F_X^{-1}(U) \leq x).$$

Ker je $F_X^{(\cdot)}$ monotono naraščajoča funkcija, lahko pišemo

$$F_X(x) = P(U \leq F_X(x)) = F_U(F_X(x)) = F_X(x).$$

Zgornja enačitev zaključuje dokaz. □

Simuliranje n.s. X z zvezno pdf $f(x)$ je torej preprosto: najprej generiramo vrednost u na intervalu $[0, 1]$ enakomerno porazdeljene n.s. in izračunamo $x = F_X^{-1}(u)$. Vrednost x je realizacija n.s. s pdf $f(x)$.

Za n.s. z diskretno porazdelitvijo $F(x)$ je simulacija podobna; kot prej generiramo vrednost u n.s. z enakomerno porazdelitvijo na intervalu $[0, 1]$ ter določimo tak x_k, da velja

$$F_X(x_{k-1}) < u \leq F_X(x_k),$$

kjer je $F_X(x_j)$ vrednost cdf definirane s predpisom

$$F_X(x_j) = \sum_{i=1}^{j} F(x_i).$$

Vrednost x_k je realizacija n.s. z diskretno pdf $F(x)$.

Izjava

Izjavljam, da sem magistrsko nalogo izdelal samostojno pod vodstvom mentorja prof. dr. Stanislava Kovačiča univ. dipl. inž. el. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

V Ljubljani, Ljubljana, september 2005.

Matej Kristan

Strokovni življenjepis avtorja

Matej Kristan se je rodil 30.10.1978 v Ljubljani. Po končani osnovni šoli se je vpisal na Gimnazijo Ledina, in nato leta 1997 na Fakulteto za elektrotehniko Univerze v Ljubljani, smer Avtomatika-Kibernetika. Leta 2000 se je pridružil skupini za robotski nogomet v Laboratoriju za simulacijo in vodenje, kjer je deloval na področju sledenja robotov z metodami računalniškega vida. Leta 2001 se je za kratek čas pridružil podjetju Zamisel d.o.o., kjer je aktivno sodeloval pri razvoju postopkov za avtomatsko kalibracijo laserskih CNC strojev ter pri razvoju aplikacije za učinkovito sklapljanje trodimenzionalnih trianguliranih objektov. Dodiplomski študij je zaključil leta 2003 z diplomskim delom iz področja računalniškega vida. Istega leta se je na Fakulteti za elektrotehniko Univerze v Ljubljani vpisal na podiplomski študij in se pridružil skupini v Laboratoriju za slikovne tehnologije, kjer sodeluje pri komercialnem projektu analize športnih iger. Trenutno področje njegovih raziskav so postopki sledenja ljudi z metodami računalniškega vida.

Objavljena strokovna dela

[1] M. Kristan, J. Perš, M. Perše, M. Bon, S. Kovačič. Multiple interacting targets tracking with application to team sports. In *Proceedings of International symposium on image and signal processing and analysis*, September 2005, *(poslano v objavo)*.

[2] M. Kristan, J. Perš, M. Perše, S. Kovačič. Bayes Spectral Entropy-Based Camera Focusing. In *Proceedings of 10th Computer Vision Winter Workshop*,155-164, February 2005.

[3] M. Kristan, F. Pernuš. Entropy Based Measure of Camera Focus. In *Proceedings of the 13th Electrotechnical and Computer Science Conference* B:179-182, September 2004.

[4] M. Kristan, J. Perš, M. Perše, S. Kovačič. Implementacija CONDENSATION algoritma v domeni zaprtega sveta. In *Proceedings of the 13th Electrotechnical and Computer Science Conference* B:179-182, September 2004.

[5] M. Kristan. Sledenje objektov v robotskem nogometu. *Diplomsko delo*, September 2003.

[6] M. Perše, M. Kristan, J. Perš, S. Kovačič. Avtomatsko zaznavanje osnovnih elementov košarkarske igre na podlagi trajektorij igralcev. In *Proceedings of the 14th Electrotechnical and Computer Science Conference*, September 2005, *(poslano v objavo)*.

[7] G. Klančar, M. Kristan, S. Kovačič. Robust and efficient vision system for group of cooperating mobile robots with application to soccer robots. *ISA Transactions*, 43(3): 329-342, July 2004.

[8] G. Klančar Gregor, M. Kristan, R. Karba. Wide-angle camera distortions and non-uniform illumination in mobile robot tracking. *Robotics and Autonomous Systems*, 46(2): 125 - 133, February 2004.

[9] J. Perš, M. Kristan, M. Perše, S. Kovačič. Observing Human Motion Using Far-Infrared (FLIR) Camera – Some Preliminary Studies. In *Proceedings of the 13th Electrotechnical and Computer Science Conference*, B:187-190, September 2004.

[10] M. Perše, J. Perš, M. Kristan, S. Kovačič. Physics-Based Modelling of Human Motion using Kalman Filter and Collision Avoidance Algorithm. In *Proceedings of International symposium on image and signal processing and analysis*, September 2005, *(poslano v objavo)*.

[11] M. Perše, J. Perš, M. Kristan, Stanislav Kovačič. Vrednotenje učinkovitosti Kalmanovega filtra pri sledenju ljudi. In *Proceedings of the 13th Electrotechnical and Computer Science Conference*, B:191-194, September 2004.