



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA TELEKOMUNIKACIJE

Praćenje objekata na video zapisu korištenjem izotropskih jezgara

ZAVRŠNI RAD
- DRUGI CIKLUS STUDIJA -

Autor:
Nermina Ahmić

Mentori:
**doc. dr Emir Turajlić,
dipl. ing. el.** **doc. dr Emir Sokić,
dipl. ing. el.**

Sarajevo,
septembar 2017.

Sažetak

Praćenje objekata na video zapisu može biti koristan alat u industrijskom okruženju, gdje nalazi primjenu za ekstrakciju trajektorije kretanja mehaničkih dijelova industrijske opreme. Rezultati se potom mogu iskoristiti za utvrđivanje eventualnih nepravilnosti i pravovremeno djelovanje radi otklanjanja istih.

Korištenje izotropskih jezgara u algoritmu pomjeranja srednje vrijednosti je jedna od najpopularnijih metoda razvijenih za ove svrhe. Cilj ovog rada je ispitivanje preciznosti spomenute metode, kao i njene fuzije sa Kalmanovim filterom, SURF algoritmom i ekstrakcijom pojedinih značajki objekta koji se prati.

Analiza je provedena nad dvije skupine video zapisa, pri različitim ambijentalnim uslovima i putanjama kretanja objekata. Pokazano je da implementirani algoritam uvodi poboljšanja u odnosu na tradicionalni, te da ima visoku robusnost. Fuzijom sa drugim metodama se može ostvariti preciznija ekstrakcija trajektorije, ali poboljšanje ili pogoršanje rezultata takvom kombinacijom zavisi od konkretne primjene.

Ključne riječi: praćenje objekata, izotropska jezgra, pomjeranje srednje vrijednosti

Abstract

Tracking of the objects on the video sequence can be a useful tool in the industrial environment, where it is commonly used for the trajectory extraction of mechanical parts of the industrial equipment. The results can be further processed in order to detect and correct potential malfunctions of the equipment.

Application of the mean-shift algorithm using isotropic kernels is one of the most popular methods in the field of research. The aim of this master thesis is to analyze the accuracy of the aforementioned method, as well as its fusion with Kalman filter, SURF algorithm and the objects' feature extraction.

The analysis is conducted using two sets of video sequences, with different environmental conditions and trajectories of the tracking object. It is shown that the implemented algorithm is highly robust and it introduces improvements compared to the traditional one. Its fusion with other methods can achieve better trajectory extraction, but the results generally depend on the specific application.

Keywords: object tracking, isotropic kernel, mean-shift

Teme za završni rad

studenata II ciklusa studija koji studiraju na ETF-u u skladu sa principima Bolonjskog procesa na Odsjeku za telekomunikacije (ak.g. <2016/17>)

Tema br.1: Praćenje objekata na video zapisu korištenjem izotropskih jezgara

Tema:

Veoma često se u aplikacijama računarske vizije pojavljuje potreba za praćenjem objekata na video snimcima (video nadzor, prepoznavanje lica, autonomno upravljanje, objektno-bazirana kompresija videa i sl.). Reprezentacija i lokalizacija objekta na snimku nije trivijalan zadatak obzirom na promjenu pojave objekta koji se prati (promjena boje, osvjetljenja, oblika itd.). Generalno postoje dva pristupa u rješavanju ovog problema: reprezentacija i lokalizacija objekta (prostorno maskiranje objekta i korelacijske mjere sličnosti), odnosno filtriranje i asocijacija podataka (varijante Kalmanovog filtera i čestičnih filtera). U ovom radu će se koristiti metode praćenja objekta korištenjem reprezentacije i lokalizacije objekta, i to korištenjem tzv. izotropskih jezgara (eng. isotropic kernel).

Postavka zadatka:

U okviru rada potrebno je:

- dati detaljan pregled literature iz oblasti praćenja objekata na video zapisima, s akcentom na metode koje se izvršavaju u (skoro) realnom vremenu,
- izvršiti komparativnu analizu najvažnijih pristupa praćenju objekata zasnovanih na pomjeranju srednje vrijednosti (tzv. mean-shift pristupi),
- pokazati prednosti i nedostatke fuzije spomenutih algoritama sa Kalmanovim filterom,
- napraviti sofversku implementaciju algoritama korištenih u radu,
- verificirati implementaciju algoritama ekstrakcije trajektorije na sintetičkim snimcima korištenjem linearног enkodera kao referentnog elementa.

Koncept i metode rješavanja:

Rad se treba sastojati iz sljedećih cjelina:

- Pregled literature i opis najznačajnijih metoda,
- Praćenje zasnovano na izotropskim jezgrima i pomjeranju srednje vrijednosti (meanshift, CAMshift i dr.),
- Korištenje Kalmanovog filtera za praćenje pokretnih objekata,
- Implementacija korištenih algoritama u MATLAB i/ili C++/OpenCV okruženju,
- Eksperimentalna analiza (brzina izvršavanja, tačnost praćenja, osjetljivost algoritama).

Osnovna literatura:

- [1] Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Kernel-based object tracking." *IEEE Transactions on pattern analysis and machine intelligence* 25.5 (2003): 564-577.
- [2] Sonka, Milan, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.
- [3] Russ, John C. *The image processing handbook*. CRC press, 2016.
- [4] Costa, Luciano da Fontoura Da, and Roberto Marcondes Cesar Jr. *Shape analysis and classification: theory and practice*. CRC Press, Inc., 2000.

- [5] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." *IEEE Transactions on pattern analysis and machine intelligence* 24.5 (2002): 603-619.
- [6] Weng, Shiu-Ku, Chung-Ming Kuo, and Shu-Kang Tu. "Video object tracking using adaptive Kalman filter." *Journal of Visual Communication and Image Representation* 17.6 (2006): 1190-1208.
- [7] Ristic, Branko, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Vol. 685. Boston: Artech house, 2004.

Univerzitet u Sarajevu
Naziv fakulteta/akademije _____
Naziv odsjeka i/ili katedre _____
Predmet _____

Izjava o autentičnosti radova

Seminarski rad, završni (diplomski odnosno magistarski) rad za I i II ciklus studija i integrirani studijski program I i II ciklusa studija, magistarski znanstveni rad i doktorska disertacija¹

Ime i prezime _____
Naslov rada _____
Vrsta rada _____
Broj stranica _____

Potvrđujem:

- da sam pročitao/la dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;
- da sam svjestan/na univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznaceno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio/la prisustvo citiranog ili parafraziranog materijala i da sam se referirao/la na sve izvore;
- da sam dosljedno naveo/la korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio/la svaku pomoć koju sam dobio/la pored pomoći mentora/ice i akademskih tutora/ica.

Mjesto, datum _____

Potpis _____

¹U radu su korišteni slijedeći dokumenti: *Izjava autora* koju koristi Elektrotehnički fakultet u Sarajevu; *Izjava o autentičnosti završnog rada* Centra za interdisciplinarnе studije – master studij „Evropske studije”, *Izjava o plagijarizmu* koju koristi Fakultet političkih nauka u Sarajevu.

Sadržaj

Popis slika	ix
1 Uvod	1
1.1 Pregled stanja u oblasti istraživanja	2
1.1.1 Širi pregled stanja u oblasti istraživanja	2
1.1.2 Pregled tehnika baziranih na filtriranju i asocijaciji podataka	4
1.1.3 Pregled tehnika baziranih na reprezentaciji i lokalizaciji objekta	5
1.1.4 Hibridne tehnike	6
1.2 Motivacija istraživanja	7
2 Procesiranje video zapisa u cilju praćenja objekata	11
2.1 Preprocesiranje	13
2.1.1 Filtriranje	14
2.1.2 Morfološke operacije	15
2.2 Ekstrakcija značajki	15
2.2.1 Model boja	16
2.2.2 Histogram	17
2.2.3 Tekstura	19
2.2.4 Oblik	20
2.3 Detektori i deskriptori značajki	21
3 Praćenje zasnovano na izotropskim jezgrima i pomjeranju srednje vrijednosti	24
3.1 Prostorno filtriranje (maskiranje)	24
3.1.1 Izotropska jezgra	25
3.2 Algoritam pomjeranja srednje vrijednosti	26
3.3 Komparativna analiza metoda zasnovanih na pomjeranju srednje vrijednosti	29
4 Korištenje Kalmanovog filtera za praćenje pokretnih objekata	34
4.1 Kalmanov filter	34
4.2 Prednosti i nedostaci fuzije Kalmanovog filtera sa metodom pomjeranja srednje vrijednosti	35
5 Praktična realizacija algoritma za praćenje objekata na video zapisu	39
5.1 Obrada selekcije objekta koji se prati	41
5.1.1 Preprocesiranje	41
5.1.2 Ekstrakcija značajki	41
5.1.3 Postprocesiranje	43
5.2 Obrada frejmova pri pretrazi objekta koji se prati	44
5.2.1 Preprocesiranje	44

5.3	Ekstrakcija i obrada značajki	46
6	Eksperimentalna analiza	53
6.1	Analiza tačnosti praćenja i osjetljivosti implementiranih algoritama	53
6.1.1	Eksperiment 1: Kretanje robotske ruke	54
6.1.2	Eksperiment 2: Kretanje modela za simulaciju uklopnog mehanizma VN prekidača	60
6.2	Analiza brzine izvršavanja algoritma za praćenje objekata na video zapisu i diskusija rezultata	64
6.3	Diskusija rezultata	66
7	Zaključak i smjernice za dalje istraživanje	68
Literatura		69

Popis slika

1.1	Primjer ciljnih rezultata ekstrakcije putanje kretanja. (a) Maketa za simulaciju kretanja uklopnog mehanizma VN prekidača (zeleni marker je korišten za praćenje kretanja). (b) Ekstraktovana putanja kretanja.	8
1.2	Primjer ciljnih rezultata ekstrakcije putanje kretanja. (a) Robotska ruka, označena višebojnim markerom čije se kretanje prati. Crna linija prikazuje putanju kojom se robot kreće. (b) Ekstraktovana putanja kretanja robotske ruke.	9
2.1	(a) Originalna slika; (b) projekcija slike na diskretnu fotoreceptorsknu mrežu; (c) diskretizirana slika sa kvantiziranim vrijednostima boje; (d) numeričke vrijednosti elemenata slike (pixela) [1].	12
2.2	(a) Prikaz originalne slike "lena.jpg"; (b) Prikaz monohromatskog ekvivalenta slike "lena.jpg".	12
2.3	Princip rada različitih tipova filtera. (a) <i>Mean</i> filter; (b) <i>Median</i> filter.	13
2.4	Rezultat filtriranja u OpenCV okruženju, za maksimalnu veličinu kernela jednaku 31. (a) Filtriranje po medijani. (b) Filtriranje po srednjoj vrijednosti.	14
2.5	Rezultat sukcesivne primjene morfoloških operacija dilatacije i erozije u postupku pretprocesiranja slike [2].	16
2.6	Grafički prikaz HSV modela boja [3].	17
2.7	(a) Rezultat konverzije iz RGB u HSV prostor boja; (b) Rezultat konverzije iz RGB u binarni prostor boja.	17
2.8	(a) Monohromatski ekvivalent slike "lena.jpg". (b) Rezultat ekvalizacije histograma.	18
2.9	Prikaz histograma boje za sliku "lena.jpg". (a) Originalni histogram. (b) Ekvalizirani histogram.	18
2.10	Primjer formiranja GLCM matrice za raspored piksela pod uglovima od 0° i 45° [4].	20
2.11	Rezultat detekcije kontura primjenom <i>Canny</i> detektora rubova u OpenCV okruženju, za prag jednak 100 (od maksimalno 255).	21
2.12	Primjer rezultata pronalaženja sličnosti između dvije slike korištenjem SIFT algoritma sa 16 ključnih tačaka [5].	22
2.13	Primjer rezultata pronalaženja sličnosti između dvije slike korištenjem SURF algoritma u OpenCV okruženju.	22
3.1	Nagib izotropske Gaussove funkcije, indiciran strelicama koje predstavljaju gradijentne vektore [6].	25
3.2	Prikaz tipova jezgara. (a) Radijalno simetrično izotropsko jezgro, kružnog oblika; (b) Simetrično jezgro anizotropskog, pravougaonog oblika; (c) Simetrično jezgro anizotropskog, eliptičnog oblika; (d) Asimetrično jezgro koje prati oblik objekta od interesa [7].	26

3.3 Ilustrativni prikaz iterativnog <i>mean-shift</i> postupka.	27
3.4 Usporedba rezultata primjene KBOT algoritma i njegovog unaprjeđenja [8]. (a) Detekcija ciljnog objekta KBOT algoritmom, objekat je označen plavom elipsom; (b) Detekcija ciljnog objekta unaprijeđenim KBOT algoritmom, objekat je označen crvenom elipsom.	30
3.5 Primjer rezultata postignutog praćenjem objekta na video zapisu upotrebom asimetričnog jezgra [7].	31
3.6 Usporedba rezultata primjene tradicionalnog <i>mean-shift</i> algoritma i algoritma koji pored informacije o boji koristi i informacije o teksturi i obliku ciljnog objekta. [9]. (a) Prvi frejm na kom je ciljni objekat označen crvenim pravougaonikom; (b) Detekcija ciljnog objekta <i>mean-shift</i> algoritmom na 460. frejmu videa i površina sličnosti ciljnog i pronađenog objekta; (c) Detekcija ciljnog objekta algoritmom koji koristi informacije o boji, teksturi i obliku objekta na 460. frejmu videa i površina sličnosti ciljnog i pronađenog objekta.	32
4.1 Blok dijagram algoritma koji koristi fuziju <i>mean-shift</i> metode i Kalmanovog filtera [10].	36
4.2 Primjer rezultata fuzije Kalmanovog filtera i <i>mean-shift</i> algoritma pri praćenju objekta koji prolazi kroz dugotrajnu potpunu okluziju [8].	37
5.1 Primjer rezultata ekstrakcije ivica radi dobivanja informacija o obliku objekta koji se prati. (a) Označeni objekat za praćenje. (b) Ekstrakcija ivica objekta u slučaju njegove stacionarnosti. (c) Prikaz objekta pri ubrzanim kretanju. (d) Ekstrakcija ivica objekta kada se on ubrzano kreće.	39
5.2 Dijagram aktivnosti implementiranog algoritma za praćenje objekata na video zapisu.	40
5.3 Dijagram aktivnosti obrade korisničke selekcije objekta koji se želi pratiti.	42
5.4 Prikaz generiranog histograma boje označenog objekta za praćenje. (a) Označeni objekat. (b) Histogram boje.	43
5.5 Usporedba rezultata dobivenih sa i bez primjene morfoloških operacija. Prikazana je projekcija histograma boje na analizirani frejm za slučajevе: (a) praćenje statičnog objekta bez primjene morfoloških operacija; (b) praćenje objekta koji se brzo kreće, pri promjeni osvjetljenja na video zapisu, bez primjene morfoloških operacija; (c) praćenje statičnog objekta uz primjenu kombinacije erozije i dilatacije; (d) praćenje objekta koji se brzo kreće, pri promjeni osvjetljenja na video zapisu, uz primjenu kombinacije erozije i dilatacije.	45
5.6 Primjer rezultata filtriranja po srednjoj vrijednosti u cilju detekcije objekta koji sadrži više od jedne boje korištenjem CamShift algoritma. (a) Frejm sa selektovanim višebojnim objektom označenim crvenom elipsom. (b) Zamagljenje frejma na osnovu dimenzija označenog objekta za praćenje.	48
5.7 Usporedba praćenja višebojnog objekta korištenjem tradicionalnog CamShift algoritma i njegove modifikacije. (a) Originalni frejm sa označenim objektom za praćenje. (b) Detekcija objekta uz pomoć prethodnog filtriranja po srednjoj vrijednosti. (c) Nemogućnost detekcije višebojnog objekta bez filtriranja. (d) Detekcija žutih objekata korištenjem modificiranog CamShift algoritma. (e) Detekcija plavih objekata korištenjem modificiranog CamShift algoritma.	49

5.8 Prikaz praćenja objekta na video zapisu uključenjem dodatnih metoda u CamShift algoritam. (a) Uvećani prikaz oznaka detektovanog objekta CamShift algoritmom (crvena elipsa) i njegovom kombinacijom sa ekstrakcijom značajki teksture (zeleni pravougaonik). (b) Detektovane i uparene karakteristične tačke primjenom SURF algoritma nad prozorom pretrage određenim CamShift algoritmom. (c) Uvećani prikaz oznaka detektovanog objekta CamShift algoritmom (crvena elipsa) i njegovom fuzijom s Kalmanovim filterom (tirkizni pravougaonik).	50
6.1 Prikaz markera korištenih za praćenje kretanja robotske ruke. (a), (b) Markeri sa više boja. (c) Jednobojni markeri, koji su osim za praćenje korišteni i za korekciju eventualnog pomjeranja kamere.	54
6.2 Primjer problema koje je potrebno riješiti za uspješno praćenje objekata na video zapisu robotske ruke koja se kreće po trajektoriji kružnice. (a) Marker na početnom frejmu. (b) Promjena osvjetljenja. (c) Djelimična okluzija.	55
6.3 Usporedba zadane i ekstraktovane trajektorije kretanja (kružnica) po x i y osi, korištenjem modificiranog CamShift algoritma, te njegove kombinacije sa ekstrakcijom tekture, Kalmanovim filterom i SURF algoritmom.	56
6.4 Usporedba zadane i ekstraktovane trajektorije kretanja (lemniskata) po x i y osi, korištenjem modificiranog CamShift algoritma, te njegove kombinacije sa ekstrakcijom tekture, Kalmanovim filterom i SURF algoritmom.	57
6.5 Ekstraktovana putanja kretanja robotske ruke koja formira lemniskatu.	58
6.6 Primjer problema koje je potrebno riješiti za uspješno praćenje objekata na video zapisu robotske ruke koja se kreće po trajektoriji kvadrata. (a) Marker na početnom frejmu. (b) Potpuna okluzija. (c) Zamagljenje (<i>blurring</i>) uslijed brzog kretanja.	59
6.7 Usporedba zadane i ekstraktovane trajektorije kretanja (kvadrat) po x i y osi, korištenjem modificiranog CamShift algoritma, te njegove kombinacije sa ekstrakcijom tekture, Kalmanovim filterom i SURF algoritmom.	60
6.8 Analizirani snimci modela za simulaciju uklopnog mehanizma VN prekidača sa različitim nivoima osvjetljenja: (a) izraženje osvjetljenje; (b) slabije osvjetljenje.	61
6.9 Prikaz modela VN prekidača čije se kretanje prati. (a) Isječak sa video zapisa na kom je prikazan isklop. (b) Isječak sa video zapisa na kom je prikazan uklop.	61
6.10 Ekstraktovana trajektorija kretanja modela za simulaciju uklopnog mehanizma VN prekidača.	62
6.11 Ekstraktovana trajektorija kretanja modela VN prekidača pri isklopu.	63
6.12 Ekstraktovana trajektorija kretanja modela VN prekidača pri uklopu.	64
6.13 Dijagram broja procesiranih frejmova u jedinici vremena za analizirane video zapise.	65

Poglavlje 1

Uvod

Estimiranje pozicije objekata tokom vremena često se koristi u praksi, bilo da se pod terminom "objekat" podrazumijeva živo biće, nestatični element ili statični element čija se pozicija naizgled mijenja zbog promjene ugla posmatranja istog. U zavisnosti od primjene, dobivena trajektorija promjene pozicije može se koristiti za analizu kretanja objekta od interesa, ili kao polazna osnova za automatizaciju industrijskih procesa. Estimacija kretanja kroz sekvencu slika naziva se praćenjem na video zapisu (engl. *video tracking*) [11] i vrši se uspostavljanjem korelacije između piksela susjednih slika, koje se najčešće u literaturi nazivaju okvirima (frejmovima) videa.

Zahvaljujući načinu na koji ljudsko oko percipira sliku, trajektorije kretanja pojedinih objekata kroz video zapis su najčešće vizuelno lahko uočljive. Međutim, kada je u pitanju računarska analiza, koja podrazumijeva zapisivanje pojedinih tačaka trajektorije u vidu prostornih koordinata, proces se značajno usložnjava. Prilikom digitalizacije, dolazi do gubitka informacija uslijed uzorkovanja i kvantizacije, zbog čega je kvalitet slike lošiji. Također, tokom samog procesa snimanja, neizbjegno dolazi do pojave šuma, koji otežava izdvajanje pojedinosti sa prikazane slike. Uz navedene početne poteškoće koje se odnose na svaku sastavnu sliku videa, najčešće se javlja i problem promjene osvjetljenja tokom vremena. Ova promjena često uzrokuje da u računarskoj viziji srodni pikseli imaju nizak nivo korelacije. To se dešava jer se analiza obično vrši korištenjem modela boja koji predstavlja piksele kao kombinaciju tri boje u odgovarajućem omjeru, a što je znatno drugačiji princip vizualizacije u odnosu na ljudsku percepciju. Piksela čije se osvjetljenje mijenja tokom vremena može imati značajnu promjenu omjera boja kojim je predstavljen, što dovodi do toga da se u računarski ne percipira kao ista tačka.

Preostali problemi koji se mogu pojaviti vezuju se za specifičnosti samog objekta koji se prati. Identifikacija može biti otežana ukoliko postoji neki sličan objekat koji se pojavljuje na video zapisu. Pritom, objekti ne moraju imati sličan oblik ili teksturu, već je i sličnost njihovih boja dovoljna kako bi praćenje bilo otežano. Ovo se naročito odnosi na bliske boje ciljnog objekta i podloge, što uzrokuje "stapanje" ivičnih piksela objekta s pikselima u njihovoј okolini. Na ovaj način se djelimično ili u potpunosti gubi informacija o obliku, u zavisnosti od mjere sličnosti spomenutih boja. Oblik objekta se sam po sebi može mijenjati kroz video sekvencu ukoliko se radi o rotirajućem objektu, te bilo kom objektu koji tokom vremena prolazi (jednim svojim dijelom) iza statičnih objekata ili kog u nekim vremenskim trenucima djelimično sakrivaju drugi pokretni objekti. Pored navedenog, približavanje i udaljavanje objekta od kamere također može imati otežavajuće djelovanje, zbog vizuelne promjene veličine objekta tokom vremena.

Usljed navedenih poteškoća koje se javljaju u računarskom praćenju objekata na video za-

pisu, dolazi do potrebe za razvojem algoritma koji će uspjeti prevladati barem većinu problema i na taj način ostvariti zadovoljavajuće performanse za potrebe aplikacije za koju je razvijen, uz rad u (približno) realnom vremenu. Ovo poglavlje ima za zadatak dati pregled postojećih algoritama u oblasti praćenja na video zapisu, dok će se u narednim poglavljima detaljnije obraditi metode koje su relevantne za ovaj rad.

1.1 Pregled stanja u oblasti istraživanja

Praćenje objekata na video zapisu je složen problem za čije rješavanje postoje dva pristupa: reprezentacija i lokalizacija objekta, te filtriranje i asocijacija podataka. U nastavku teksta izdvojene su karakteristike oba pristupa i dat je pregled postojećih algoritama u ovoj oblasti.

1.1.1 Širi pregled stanja u oblasti istraživanja

Kako bi se omogućilo praćenje određenog objekta na video zapisu, neophodno je posjedovati dovoljno informacija koje ga opisuju. U zavisnosti od vrste tih informacija, metode za praćenje se mogu podijeliti u četiri kategorije: algoritmi bazirani na ekstrakciji značajki (koriste eks-traktovane informacije o objektu), bazirani na apriornom znanju (posjeduju unaprijed poznate informacije o objektu), gradijentno bazirani (lociraju ciljni objekat minimizacijom funkcije ci-jene) i bazirani na učenju, odnosno treniranju (primjenjuju algoritme za prepoznavanje uzorka u cilju učenja modela objekta) [12]. Fokus algoritama relevantnih za ovaj rad stavlja se na metode zasnovane na ekstrakciji značajki (engl. *feature extraction*), zbog čega će se njima posvetiti više pažnje u okviru ovog poglavlja, dok će se algoritmi koji pripadaju preostalim kategorijama razmatrati u kombinaciji sa metodama baziranim na ekstrakciji značajki, u okviru potpoglavlja koje daje pregled hibridnih algoritama.

Ekstrakcija značajki se može podijeliti na tri nivoa: niski (boja, fotometrijska invarijantnost boje, gradijent, laplasijan, pokret), srednji (ivice, tačke i regije od interesa, uniformne regije) i visoki (modeli pozadine i modeli objekata) [11]. Većina aplikacija razvijenih za praćenje objekata zadržava se samo na jednoj od značajki koje spadaju u kategoriju najnižeg nivoa, što stavlja zahtjev za njihovom visokom preciznošću kako bi performanse bile zadovoljavajuće.

Modeliranje objekta se najčešće vrši na osnovu funkcije gustoće vjerovatnoće (engl. *Probability Density Function - PDF*) boje [9, 10, 13–16], ali se mogu koristiti i informacije o teksturi [9, 13], obliku [9], rubovima [13], kao i kombinacija navedenih informacija [17]. Za opis vje-rovatnoće gustoće raspodjele značajki objekta mogu se koristiti parametarski i neparametarski modeli. Parametarski modeli vrše segmentaciju scene i opisivanje svake izdvojene regije sa određenim brojem parametara, dok neparametarski svakom pikselu dodjeljuju korespondenciju ili vektor toka [11] i omogućavaju reprezentaciju složenih gustoća raspodjele. Samim time, ne-parametarski modeli, kao što su modeli zasnovani na histogramu ili gustoći jezgra, imaju znatno veću računarsku kompleksnost u odnosu na parametarske (npr. Gaussova funkcija ili mješavina Gaussovih funkcija) [18].

Razlog zbog kog se obično analizira samo PDF boje umjesto složenije ekstrakcije značajki je računarska kompleksnost algoritma i dužina vremena izvršavanja. Teži se ka što kraćem vremenu procesiranja u cilju osposobljavanja algoritma za rad u realnom vremenu, što aplikacije najčešće zahtijevaju. Međutim, treba napomenuti da je nekada moguće prevazići dugotrajno procesiranje, uprkos korištenju složenijih značajki objekta, kao što su njegove ivice [19] ili raz-dvajanje objekata od pozadine [20, 21]. Ipak, nedostatak metoda koje vrše ovo razdvajanje je nemogućnost potpunog oduzimanja podloge, što je problem koji se javlja uslijed malih pokreta određenih objekata, pojave sjenki, prisustva šuma, kao i varijabilnog osvjetljenja. Izdvajanje

složenijih značajki naročito je zastupljeno u aplikacijama koje koriste automatsko praćenje pokretnih objekata, odnosno koje imaju za zadatak detektovati pokrete, bez apriornog označavanja objekta čije se kretanje želi pratiti kroz video sekvencu. Fokus ovog rada je na metodama koje koriste interaktivne strategije praćenja objekata, gdje korisnik vrši selekciju objekta, nakon čega algoritam na osnovu izdvojenih značajki vrši njegovo praćenje. Iz tog razloga, metode namijenjene za automatsko praćenje neće biti detaljno analizirane u okviru rada.

Nakon što se izvrši označavanje objekta za praćenje, otvara se širok spektar problema koje je potrebno riješiti kako bi postupak bio uspješan. Time se dodatno usložnjava zahtjev za procesiranje u realnom vremenu. Problemi koji se javljaju se u prvom redu odnose na promjene izgleda objekta u video sekvenci. Pokretni objekti se mogu tokom vremena mijenjati, što u najjednostavnijem smislu podrazumijeva njihovu rotaciju ili translaciju, dok složenije promjene obuhvataju deformaciju objekta ili okluziju (prikrivanje objekta).

Za rješenje problema praćenja u slučaju da je objekat podvrgnut afnim transformacijama, može se koristiti mјera affine sličnosti između regija okvira video sekvence, te estimacija transformacije parametara pokretnog objekta [14]. Ova procedura omogućava proračun vektora translacije, ugla rotacije, faktora skaliranja i faktora smicanja, te pokazuje znatno bolje performanse u odnosu na popularnu metodu pomjeranja srednje vrijednosti (engl. *mean-shift*) [22], ali ne omogućava izvršavanje u realnom vremenu.

Termin okluzija obuhvata djelimično ili potpuno sakrivanje objekta koji se prati nekim drugim statičnim ili dinamičnim objektom. Čak i u uslovima stacionarnosti svih objekata, okluzija može nastupiti u slučaju promjene ugla snimanja, odnosno korištenja pokretne kamere. Ukoliko se za praćenje objekta i u slučaju okluzije koriste iste informacije kao i u njenom odsustvu, performanse algoritma mogu biti narušene. *Mean-shift* procedura, koja je od naročitog značaja za ovaj rad, ima upravo taj nedostatak, s obzirom na to da radi na principu maksimiziranja sličnosti ciljnog modela objekta i mogućeg modela objekta (kandidata). Kako bi se izbjegao ovaj problem, može se uvesti određeni broj tačaka za praćenje [16] unutar prozora praćenja, kojim je označen model ciljnog objekta. Na ovaj način se omogućava da se u slučaju parcijalne okluzije prate samo one tačke objekta koje nisu zaklonjene nekim drugim objektom. Drugi pristup rješavanju problema parcijalne okluzije je detekcija njenog postojanja, te rekonstrukcija oblika objekta koji se prati [17] ili predikcija lokacije na kojoj se on treba nalaziti [23], što se može koristiti i u slučaju potpune okluzije. Međutim, ovaj pristup zahtijeva postojanje informacije o konturi objekta, čime se usložnjava algoritam i nije pogodan za aplikacije koje zahtijevaju rad u realnom vremenu.

Tokom vremena, moguća je i promjena osvjetljenja u videu, što može predstavljati problem u algoritmima koji se oslanjaju isključivo na informaciju o boji objekta koji se prati. Iz tog razloga se često koristi konverzija iz RGB (engl. *Red, Green, Blue*) u HSI (engl. *Hue, Saturation, Intensity*) [15] ili HSV (engl. *Hue, Saturation, Value*) [24] model boja. Naime, HSI model bolje opisuje ljudski vizuelni sistem i omogućava zadržavanje informacije o tonu boje, dok se promjenom luminanse mijenja njeno zasićenje i intenzitet. S druge strane, RGB model sadrži samo informacije o nivou crvene, zelene i plave boje, te je kao takav nepogodan za praćenje objekata na temelju informacije o boji u uslovima promjenjive luminanse. Međutim, ukoliko se pored informacije o boji koriste i informacije o teksturi i konturi objekta, korištenje HSI umjesto RGB modela ne pokazuje poboljšanje performansi praćenja objekata [17]. Zbog toga se neki algoritmi zadržavaju na RGB modelu boja, uzimajući u obzir dodatne informacije, kao što su prostorne značajke [14], tekstura objekta, njegovi rubovi [17] i sl., kako bi obezbijedili dobre performanse i u uslovima promjenjivog osvjetljenja.

Dodatna poteškoća za algoritme bazirane isključivo na korištenju informacije o boji je i potreba za praćenjem objekta koji ima sličnu boju kao podloga ili neki od drugih objekata u

sceni koju video prikazuje. Poboljšanje se može postići omogućavanjem korelacije prostornih i značajki boja [14], na račun vremena izvršavanja algoritma. S druge strane, ovakav pristup je pokazao veću robusnost u odnosu na *mean-shift* [22] algoritam, kao i na čestični *CONDENSATION* algoritam [25], čiji će principi rada biti objašnjeni u narednim potpoglavljiima.

Nakon što je izvršen kratak pregled osnovnih pojmoveva i problema s kojima se susreću algoritmi za praćenje objekata u realnom vremenu, nešto detaljniji osvrt na funkcionalnost pojedinih skupina algoritama obraditi će se u okviru narednih potpoglavlja.

1.1.2 Pregled tehnika baziranih na filtriranju i asocijaciji podataka

Praćenje objekta se može poistovijetiti sa estimacijom stanja objekta uzimajući u obzir rezultate mjerena njegovih parametara [26]. Imajući to u vidu, ukoliko je šum u razmatranom sistemu Gaussove raspodjele, najpovoljnija estimacija stanja objekta postiže se korištenjem Kalmanovog filtera [15], ali se često koriste i čestični filteri [25, 27], naročito u slučaju da nije moguće usvojiti inherentne pretpostavke za Kalmanovo filtriranje. Pored navedenih, u nekim aplikacijama koriste se Beneš i Daum filteri [28]. Nakon izvršenog filtriranja, slijedi proces asocijacije podataka. On zahtijeva pronađenje pozicije pokretnog objekta i u uslovima postojanja većeg broja, prema metričkoj sličnosti, podjednako mogućih ciljnih kandidata [29].

Kalmanov filter se često koristi zbog mogućnosti optimalne rekurzivne estimacije stanja dinamičkog sistema [29] i rada u realnom vremenu. Zahvaljujući ovim osobinama, on omogućava predikciju pozicije objekta u narednim okvirima, zbog čega ima jako dobre performanse u uslovima okluzije. Stepen okluzije se može koristiti za adaptivnu prilagodbu kovarijanse greške Kalmanovog filtera [15], čime se postižu bolje performanse u odnosu na konvencionalno filtriranje. Osnovni nedostatak algoritama zasnovanih na korištenju ovog filtera je pretpostavka da su sistem i njegove opservacije linearni, te da je šum u sistemu i u mjerenjima Gaussov [30], što najčešće nije slučaj.

Usljed navedenih ograničenja Kalmanovog filtera, implementiran je generalizirani pristup suboptimalnih, čestičnih filtera. On ujedno predstavlja i najpopularniju tehniku baziranu na filtriranju i asocijaciji podataka. Čestični filteri omogućavaju reprezentaciju vrlo složenih gustoća raspodjele, zbog čega je estimacija gustoće jezgra [20] vrlo popularna i fleksibilna metoda. S druge strane, nedostatak čestičnih filtera leži u njihovim velikim memorijskim zahtjevima, s obzirom na to da se gustoća raspodjele reprezentuje najčešće velikim skupom slučajnih uzoraka. Ovi filteri koriste sekvencijalnu Monte Carlo estimaciju baziranu na reprezentaciji gustoće raspodjele u vidu tačke, odnosno čestice [28].

Prvi algoritam koji je zasnovan na čestičnim filterima je implementiran još u prošlom stoljeću [25], pod nazivom CONDENSATION. On koristi vizualne opservacije i dinamičke modele ciljnog objekta, čime ostvaruje robusnost i znatno bolje performanse u odnosu na Kalmanov filter, onda kada se analizira slika ili frejm na kom je teško izdvojiti objekat koji se prati od podloge i ostatka prikazanih objekata.

Uzorkovanje predstavlja najveći problem čestičnih filtera, naročito u slučaju problema visoke dimenzionalnosti, zbog čega algoritmi bazirani na ovoj metodi obično zahtijevaju veliko vrijeme izvršavanja. Ono može biti skraćeno korištenjem čestičnih filtera baziranih na upotrebi jezgra [27, 31, 32], čime se smanjuje broj čestica kojim se operira, a samim time i računarska kompleksnost. Dodatno reduciranje kompleksnosti može se izvesti inkrementalnom aproksimacijom čestica i kombiniranjem algoritama baziranih na čestičnim filterima sa metodama baziranim na reprezentaciji i lokalizaciji objekta. Ovoj kombinaciji je više riječi posvećeno u potpoglavlju koje prezentira hibridne algoritme.

1.1.3 Pregled tehnika baziranih na reprezentaciji i lokalizaciji objekta

Reprezentacija podrazumijeva modeliranje objekta na osnovu njegovih karakteristika. U tu svrhu se koristi prostor značajki (*engl. feature space*) [13, 22], koji je nezavisan od aplikacije, što je njegova osnovna prednost. U ovom prostoru se definira funkcija gustoće vjerovatnoće (*engl. Probability Density Function - PDF*) značajki objekta koji se posmatra.

Lokalizacija obuhvata izbor funkcije sličnosti između modela ciljnog objekta (objekta koji se prati) i modela kandidata (regija narednih okvira videa koji mogu predstavljati objekat koji se prati), te mjeru sličnosti kojom će se izvršiti izbor najboljeg kandidata. Lokalizacijske metode se mogu svrstati u dvije skupine: one koje koriste jednu hipotezu i one koje koriste više lokalizacijskih hipoteza, odnosno evaluiraju istovremeno više kandidata [11]. Drugu skupinu metoda obično koriste čestični filteri, na koje je akcenat stavljen u prethodnom potpoglavlju.

Tehnika bazirana na reprezentaciji i lokalizaciji objekta koja ima najčešću primjenu u praksi je pomjeranje srednje vrijednosti, tj. ranije spomenuti *mean-shift* algoritam [22]. On koristi histogram boje za modeliranje objekta, te pomjera regiju objekta u smjeru najvećeg gradijenta gustoće raspodjele. Naime, prostor značajki se sastoji od kvantitativnih osobina slike koje se mapiraju u tačke, zbog čega je gustoća raspodjele najveća na lokaciji koja odgovara značajkama od interesa [30]. Pretraga objekta počinje postavljanjem inicijalnog prozora na bilo koju poziciju unutar okvira, nakon čega se određuje centroid prozora, te se formira novi prozor pomjeren na lokaciju centroida. Ovaj proces se ponavlja dok ne dođe do konvergencije ka maksimumu gustoće raspodjele. Parametri koji određuju brzinu konvergencije, a samim tim i vrijeme izvršavanja, su veličina i oblik prozora, tj. jezgra. Tradicionalni algoritmi koriste radikalno simetrična jezgra, najčešće normalno (Gaussovo) jezgro i Epanechikovo jezgro [13, 33].

Ograničenja *mean-shift* algoritma su konstantna veličina jezgra i korištenje izotropskih radikalno simetričnih jezgara. Nepromjenjiva veličina jezgra predstavlja problem zbog promjena u veličini objekta koji se prati, a koje najčešće nastupaju uslijed približavanja ili udaljavanja objekta od kamere. Kako bi se riješio ovaj problem, mogu se koristiti asimetrična jezgra sa adaptivnom promjenom veličine i orientacije [7, 34] ili izotropska jezgra uz dodatak korištenja centroida objekata kao centralne komponente za praćenje [35]. U slučaju da objekt nije izotropskog oblika, izotropsko jezgro obuhvata regije okvira koje ne pripadaju objektu, što može dovesti do pogoršanja performansi praćenja. Zbog toga upotreba anizotropskih jezgara pokazuje znatno bolje performanse [36] u odnosu na izotropska, ali su sva navedena poboljšanja izvršena na račun povećane dužine izvršavanja algoritma.

Mean-shift algoritam se koristi kao polazna osnova za brojne druge metode namijenjene praćenju objekata na video zapisu. Praćenje bazirano na korištenju jezgra (*engl. Kernel-based Object Tracking - KBOT*) predstavljeno u [13] koristi izotropsko jezgro kojim se vrši prostorno maskiranje ciljnog objekta, nakon čega se primjenjuje prostorno glatka funkcija sličnosti, pogodna za optimizaciju baziranu na gradijentu gustoće raspodjele. Optimizacija se vrši primjenom *mean-shift* algoritma. Ideja korištenja izotropskog jezgra je da se prostor značajki opiše težinama jezgra, pri čemu veće vrijednosti imaju pikseli koji su bliži centru objekta, dok udaljeniji imaju niže vrijednosti, odnosno manji utjecaj. Ovakva dodjela težinskih faktora je izvršena korištenjem pretpostavke da pikseli na većem rastojanju od centra objekta imaju veću vjerovatnoću okluzije i smanjenu mogućnost diferencijacije od drugih objekata ili podloge. Kao mjera sličnosti između objekata u različitim okvirima koristi se metrika izvedena iz Bhattacharyya koeficijenta. Najvjerojatnija lokacija ciljnog objekta u trenutnom okviru je ona za koju navedeni koeficijent ima najveću vrijednost.

KBOT omogućava rad u realnom vremenu i ima jako dobre performanse, međutim, njegova ograničenja su nepromjenjivost modela ciljnog objekta i nesigurna metoda adaptacije skaliranja. Ovi nedostaci mogu se prevazići ažuriranjem modela objekta [18] i estimacijom optimalnih

razmjera objekta [8], uz održanje mogućnosti rada u realnom vremenu. Pored navedenih ograničenja, kao i svi algoritmi zasnovani na pomjeranju srednje vrijednosti, KBOT koristi pretpostavku da ciljni objekti imaju mali pomak između sukcesivnih okvira, odnosno, da se sporo kreću. Kako bi se omogućile njegove dobre performanse i u uslovima praćenja objekta koji se brzo kreće, moguće je koristiti pristup zasnovan na Gaussovim piramidama [33]. Isti okviri različitih rezolucija grade piramidu, pri čemu se za pretragu objekta koristi fiksna veličina jezgra, te se počinje od najniže rezolucije. Na taj način se na nižim rezolucijama omogućava detekcija objekta sa velikim pomakom u odnosu na prethodni okvir, ali i povećanje brzine rada algoritma. Navedeni problem može se riješiti i primjenom Kalmanovog filtera [10], ali ovo rješenje koristi pretpostavku da je poznat model kretanja ciljnog objekta i ima veću računarsku kompleksnost.

Iako sam KBOT ne zahtijeva dugo vremena za izvršavanje, implementirane su i određene metode za smanjenje njegove računarske kompleksnosti, koja zavisi od broja iteracija po okviru pri pretrazi ciljnog kandidata. Autori rada [37] predlažu da se pored maksimalnog broja iteracija, nakon čijeg se dostizanja završava pretraga ukoliko prethodno nije pronađen ciljni kandidat, generira i vrijednost praga nakon kojeg prestaje pretraga. Ovaj prag označava maksimalnu udaljenost između ciljnog objekta i kandidata, te se pretraga završava ukoliko je njihova udaljenost manja od praga. Također, umjesto eliptičkog modela objekta koji koristi KBOT i normalizira ga na kružni model, autori u radu [37] implementiraju kvadratni model kojim se smanjuje računarska kompleksnost s obzirom na to da uslijed digitalizacije nije moguće realizirati stvarni eliptički model, već se uvijek formira njegova aproksimacija.

Adaptacija algoritma pomjeranja srednje vrijednosti koja je prvobitno namijenjena za poboljšanje praćenja lica osoba, a koja se često koristi u praksi je *Continuously Adaptive Mean Shift Algorithm* (CamShift), implementiran od strane Intel korporacije [38]. Za razliku od *mean-shift* algoritma koji je baziran na statičkim raspodjelama, CamShift koristi kontinualno adaptivne raspodjele vjerovatnoća i HSV model boja, koji upotrebljava za modeliranje objekta korištenjem histograma boje. Prednost CamShift-a je u njegovoj niskoj kompleksnosti, ali kao i *mean-shift* algoritam, on ima loše performanse u uslovima potpune okluzije. Ta činjenica upućuje na to da se i problem može riješiti na isti način, odnosno upotrebom Kalmanovog filtera [24], uz održanje performansi u realnom vremenu, čemu je više riječi posvećeno u jednom od narednih poglavlja. Okluzija, kao i problemi praćenja objekta u slučaju promjene njegove orientacije ili razmjere, te sadržavanja većeg broja nijansi boja, mogu se riješiti upotrebom proširenog CamShift algoritma [33], koji vrši proračun centra mase objekta i koristi težirani histogram za lokalizaciju ciljnog objekta.

1.1.4 Hibridne tehnike

S obzirom na to da tehnike bazirane na reprezentaciji i lokalizaciji, kao i tehnike bazirane na filtriranju i asocijaciji imaju svoje prednosti i nedostatke, razvijeni su brojni algoritmi koji koriste kombinaciju navedenih pristupa, te kombinaciju metoda baziranih na učenju, ekstrakciji značajki, posjedovanju apriornih informacija i gradijentno baziranih metoda. U nastavku teksta će se navesti samo neke od njih, dok će se hibridne tehnike zasnovane na kombinaciji algoritma pomjeranja srednje vrijednosti i Kalmanovog filtera detaljnije obraditi u jednom od narednih poglavlja ovog rada.

Najčešća kombinacija metoda koja se koristi je fuzija Kalmanovog filtera sa *mean-shift* algoritmom. U usporedbi pojedinačnog rada svake od navedenih metoda, pokazano je da upotreba Kalmanovog filtera daje bolje performanse u odnosu na algoritam pomjeranja srednje vrijednosti u prisustvu šuma i složenijih promjena stanja objekta, dok oba pristupa imaju iste performanse u slučaju translacije objekta [39]. Kada se izvrši njihova fuzija, smanjuje se regija

pretrage objekta zahvaljujući Kalmanovom filteru koji omogućava estimaciju moguće pozicije ciljnog objekta, te se poboljšavaju performanse algoritma baziranog na pomjeranju srednje vrijednosti, naročito u slučaju kada se analiziraju slike u kojima je objekat slične boje kao podloga ili neki drugi objekat [40], s obzirom na to da histogram boje ne obezbjeđuje veliku robusnost. Pritom se zadržava mogućnost rada u realnom vremenu. Kalmanov filter se koristi i u kombinaciji sa CamShift algoritmom [24], te omogućava poboljšanje njegovog djelovanja u prisustvu okluzija, kao i promjena oblika i orientacije objekta koji se prati.

Pored toga, *mean-shift* i CamShift se mogu koristiti i u kombinaciji sa čestičnim filterima [31, 41], čime se iskorištavaju prednosti vrlo složene metode estimacije gustoće jezgra, uz smanjenje memorijskih zahtjeva. Navedenom kombinacijom se poboljšava adaptacija razmjera objekta u odnosu na tradicionalne algoritme, povećava efikasnost uzorkovanja čestičnih filtera, te se reducira njihova računarska kompleksnost.

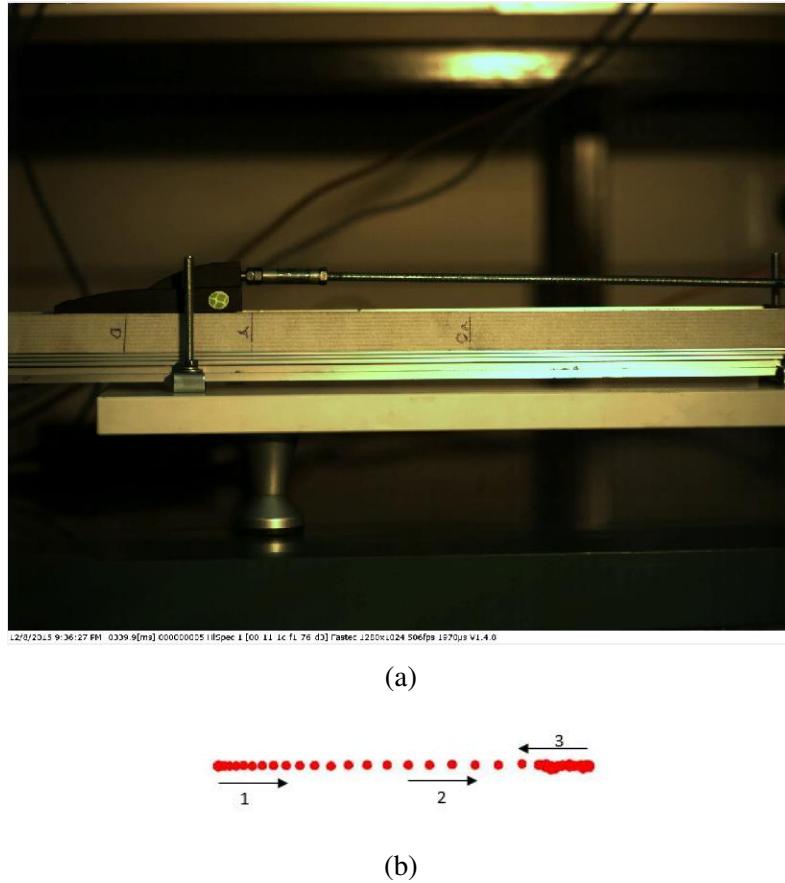
Ukoliko se udruže gradijentno bazirana metoda sume kvadratnih razlika (engl. *Sum-of-Squared Differences - SSD*) i metoda pomjeranja srednje vrijednosti bazirana na informaciji o boji objekta koji se prati, algoritam pokazuje bolje performanse u odnosu na tradicionalni *mean-shift*, pritom zadržavajući mogućnost rada u realnom vremenu [16]. Dok *mean-shift* metoda nema dobre performanse ukoliko se objekat brzo kreće, tj. ako je njegov pomak između suksesivnih okvira prevelik, performanse SSD metode su narušene zahtjevom za brzim ažuriranjem modela, s obzirom na kratkotrajnu memoriju ove metode. Međutim, prednost ovakve kombinacije dva različita pristupa u praćenju objekata je u tome da oni međusobno kompenziraju svoje nedostatke. Kako vrijeme izvršavanja algoritama koji koriste sintezu dvije ili više metoda većinom zavisi od one metode koja ima najveću računarsku kompleksnost, u slučaju korištenja *mean-shift* i SSD metode, složenost zavisi od veličine prostora pretrage koju koristi SSD.

Pored ograničenja navedenih u okviru potpoglavlja u kom se opisuju metode bazirane na reprezentaciji i lokalizaciji objekta, algoritmi bazirani na pomjeranju srednje vrijednosti imaju i nedostatak u tome da se model objekta koji se prati gradi na osnovu samo jedne slike, te postoje poteškoće u adaptivnom ažuriranju modela. Ovaj nedostatak se može prevazići primjenom treniranja korištenjem mašina vektorske podrške (engl. *support vector machines - SVM*) [42]. Iako su metode koje obuhvataju treniranje obično vrlo složene za realizaciju, algoritam implementiran u radu [42] ima istu računarsku kompleksnost kao i standardni algoritmi zasnovani na pomjeranju srednje vrijednosti, te bolje performanse u odnosu njih, kao i u odnosu na algoritme bazirane na čestičnom filtriranju. S druge strane, algoritmi koji ne koriste pomjeranje srednje vrijednosti, a zasnovani su na korištenju SVM-a teže ostvaruju rad u realnom vremenu [43], iako je to moguće ostvariti u nekim slučajevima, zavisno od broja značajki koje se koriste za praćenje.

S obzirom na brojne probleme koje je potrebno riješiti kako bi se ostvarilo idealno praćenje objekta na video zapisu, u literaturi je moguće pronaći veliki broj kombinacija postojećih pristupa i njihovih unaprjeđenja. Kako nije pronađeno optimalno rješenje, istraživanja na ovu temu su još uvijek aktuelna.

1.2 Motivacija istraživanja

Kako je ranije navedeno, praćenjem objekata na video zapisu može se izvršiti ekstrakcija trajektorije kretanja, na osnovu koje je moguće provesti dalju analizu u različitim naučnim poljima. U slučaju praćenja pokreta ljudskih ruku, moguće je prevesti znakovni jezik [44] u odgovarajuće riječi. Ako se analizira putanja kojom se kreću jedinke neke sorte, najčešće životinje [45], moguće je uočiti sličnosti u kretanju i na osnovu njih izvesti zaključke o ponašanju pri određenim

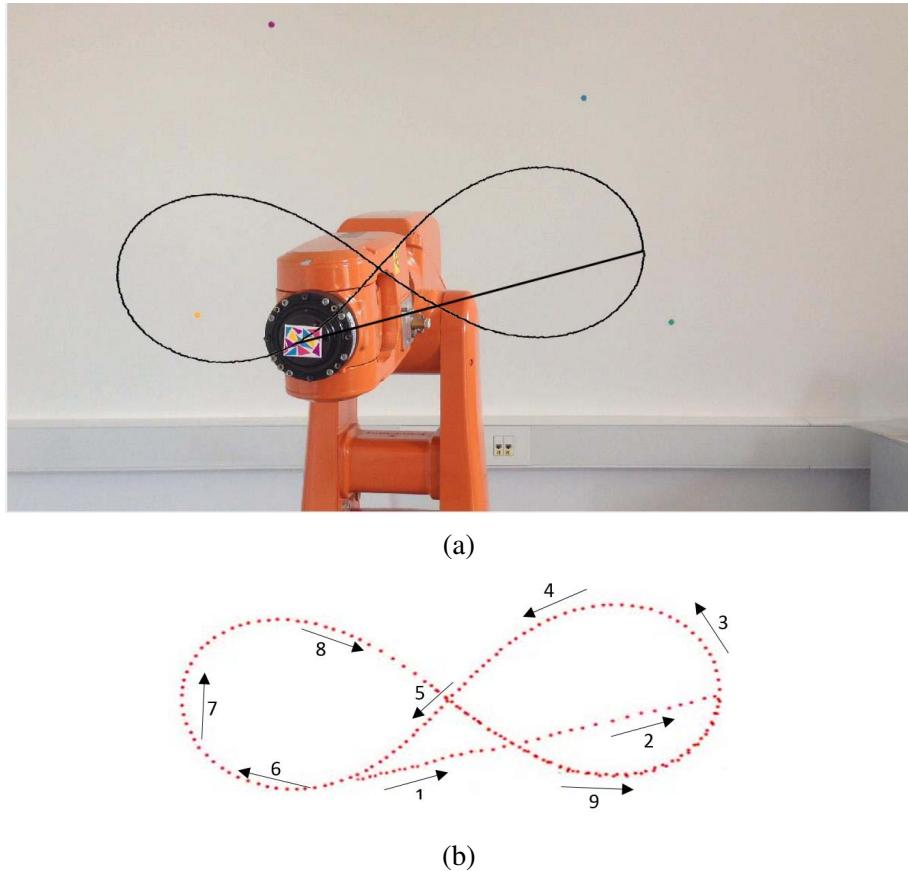


Slika 1.1: Primjer ciljnih rezultata ekstrakcije putanje kretanja. (a) Maketa za simulaciju kretanja uklopnog mehanizma VN prekidača (zeleni marker je korišten za praćenje kretanja). (b) Ekstraktovana putanja kretanja.

uslovima ili izvršiti predikciju budućeg kretanja. Analogno, u industrijskom okruženju mogu se uočiti pravilnosti u trajektoriji objekta koji se prati, ukoliko se proces ispravno izvršava, i na osnovu toga automatizirati isti [46], ili vršiti nadzor i eventualno detektovati neispravan rad, radi mogućnosti popravka.

Da bi industrijski sistem kao cjelina bio funkcionalan, neophodno je ostvariti raspoloživost, pouzdanost i efikasnost svih njegovih elemenata. Međutim, donedavno je bilo uobičajeno da se dijagnostika opreme provodi u pravilnim vremenskim intervalima, koji su za visokonaponske (VN) prekidače bili mjereni u godinama [47]. Radi što boljih performansi, javlja se potreba za bržom detekcijom problema, a kao rješenje se nameće stalni nadzor elemenata u industrijskom okruženju. Nadzorom putem statične kamere moguće je prikupiti potrebne informacije i obraditi snimke, te doći do zaključka o ispravnosti rada objekta od interesa. Pored navedenog, u slučaju visokonaponskih prekidača, mehanička ispitivanja pogonskog mehanizma zahtijevaju mjerjenje brzih mehaničkih pojava [48], zbog čega je ključno riješiti problem tačnosti praćenja objekta. Zahtjev za radom u realnom vremenu, koji aplikacije za praćenje objekata najčešće imaju, pada u drugi plan, ali je svakako cilj postići što bržu obradu video zapisa.

Ideja korištenja računarske obrade slike u svrhu analize rada industrijskog sistema i ispitivanja mehaničkih karakteristika njegovih pojedinih dijelova u svrhu pomoći pri projektovanju mehaničkog sistema potiče iz činjenice da se određena kretanja ne mogu izmjeriti konvencionalnim senzorima, jer vrlo brze prelazne mehaničke pojave iziskuju visoku osjetljivost sen-



Slika 1.2: Primjer ciljnih rezultata ekstrakcije putanje kretanja. (a) Robotska ruka, označena višebojnim markerom čije se kretanje prati. Crna linija prikazuje putanju kojom se robot kreće. (b) Ekstraktovana putanja kretanja robotske ruke.

zora [48] ili ne postoji dovoljno prostora za njihovu montažu. Primjer ovakvih fenomena su (ne)elastična sabijanja metalnih opruga, aksijalne i torzionalne vibracije, postojanje zazora na spojevima u kinematskom lancu i sl. U navedenim slučajevima, analiza slike predstavlja rješenje, a omogućava određivanje relativne ili apsolutne trajektorije objekta ili njegovog određenog dijela. Analizom dobivene putanje mogu se detektovati slabe tačke konstrukcije, te predvidjeti eventualna oštećenja mehaničkog sistema.

Za potrebe industrijskih aplikacija, do sada su razvijeni algoritmi za praćenje objekata koji koriste mašinsko učenje pri procesu praćenja na video snimku s pokretne kamere (pričvršćene na kacigu radnika u industrijskom okruženju) [49], oni koji koriste određene značajke objekta [50], te polje deskriptora [51, 52]. Spomenuti algoritmi razvijeni su za potrebe specifičnih aplikacija, pa se kao takvi oslanjaju na rješavanje problema koji su ključni za njih i koriste poznate informacije o konkretnim objektima od interesa. Fokus rada [52] stavljen je na rješavanje problema okluzije, dok se u radu [49] nastoji ostvariti što je manje moguće vrijeme potrebno za procesiranje frejmova. Algoritam opisan u [50] vrši detekciju svih objekata kružnog oblika, a često se u aplikacijama koristi i boja kao značajka na osnovu koje se automatski detektuje objekat od interesa. Uobičajeno je da se u industrijskom okruženju razvijaju namjenski algoritmi koji uvažavaju karakteristike okruženja, kao što je to slučaj sa spomenutim algoritmima.

Prethodno navedena rješenja koriste složenije pristupe u odnosu na praćenje bazirano na upotrebi izotropskih jezgara, te se fokusiraju na probleme koji se najčešće ne javljaju u okruženju od interesa za ovaj rad. Složenost pristupa rješavanju problema često uzrokuje duže vrijeme

izvršavanja algoritma, ali i ukoliko ovo nije slučaj, takva rješenja su obično skuplja u odnosu na ona jednostavnijeg pristupa. Namjena algoritma implementiranog u okviru ovog rada je u prvom redu ekstrakcija hodograma mehaničkih dijelova modela visokonaponskih prekidača sa video snimka brze kamere, te hodograma robotske ruke. U oba slučaja, kamera je statična, objekti od interesa su različitih boja i oblika, te ne dolazi do okluzije. Primjeri ciljnih rezultata dati su na slikama 1.1 i 1.2, gdje su prikazane maketa za simulaciju kretanja uklopnog mehanizma VN prekidača i robotska ruka, te izvedene putanje njihovih kretanja.

Industrijsko okruženje samo po sebi stvara teže uslove za praćenje objekata na video zapisu, s obzirom na to da je osvjetljenje najčešće nehomogeno, te promjenjivo tokom vremena, tekstura objekata nije izražena, te se objekti od interesa nerijetko "stapaju" s podlogom po kojoj se kreću. Kako je fokus rada na algoritmima zasnovanim na korištenju izotropskog jezgra, potrebno je eliminirati postojeće nedostatke CamShift algoritma, koji nije u mogućnosti vršiti praćenje višebojnih objekata, za čim se javlja potreba u konkretnoj primjeni. U narednim poglavljima izvršena je analiza navedenih problema i data su primijenjena rješenja za iste.

* * *

U okviru ovog poglavlja dat je pregled literature na temu praćenja objekata na video zapisu, te motivacija za provedeno istraživanje. Metode bazirane na reprezentaciji i lokalizaciji objekta, kao i metode zasnovane na filtriranju i asocijaciji podataka imaju svoje prednosti i nedostatke, zbog čega izbor optimalne metode zavisi od potreba aplikacije koja će je upotrebljavati. Naredna poglavlja će se fokusirati na metode bazirane na pomjeranju srednje vrijednosti i njihovoj fuziji sa Kalmanovim filterom, pri čemu će biti detaljnije objašnjen princip navedenih metoda.

Poglavlje 2

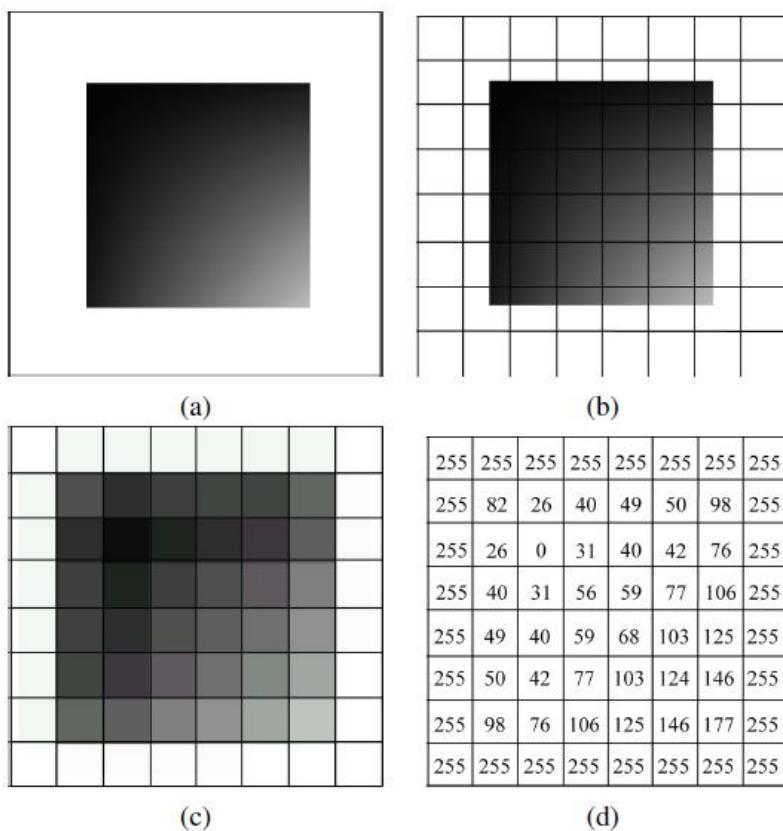
Procesiranje video zapisa u cilju praćenja objekata

Na samom početku, potrebno je upoznati se sa osnovnim pojmovima i elementima obrade slike, koja predstavlja osnovu svakog video zapisa. Sve tehnike obrade slike primjenjuju se i na videu, s tim da se kao dodatni zahtjev stavlja smanjenje računarske kompleksnosti, s obzirom na to da se u samo jednoj sekundi videa treba obraditi najmanje 25 slika, što za cilj postavlja njihovo odgovarajuće procesiranje u navedenom periodu.

Reprezentacija slike na računaru je složen proces, koji se grubo može podijeliti na dva nivoa: procesiranje slike i njen razumijevanje [30]. Slika se matematski može opisati kao funkcija dvije promjenjive $f(x,y)$, gdje x i y označavaju prostorne koordinate u ravni. Kako bi se moglo provoditi računarsko procesiranje, neophodno je izvršiti proces digitalizacije. U tu svrhu, originalna slika se prvo uzorkuje (engl. *sampling*), čime se prevodi u konačan broj elemenata, nakon čega se vrši kvantizacija, gdje se svakom elementu dodjeljuje odgovarajuća vrijednost, u zavisnosti od varijacije intenziteta svjetlosti po površini koja je predstavljena na slici. Treba napomenuti da je aproksimacija originalne slike bolja što je veći broj kvantizacionih nivoa, te ako je veća frekvencija uzorkovanja, tj. rezolucija. Međutim, s povećanjem broja uzoraka i nivoa kvantizacije, povećava se i složenost same digitalne slike, te računarska obrada zahtijeva više vremena.

Sastavni elementi digitalne slike nazivaju se pikselima, a vrijednosti koje mogu poprimiti leže u rasponu od 0 do 255. Vrijednost 0 se dodjeljuje potpuno crnoj boji, koja apsorbuje svjetlost svih talasnih dužina, dok 255 odgovara bijeloj boji, odnosno potpunoj refleksiji. Između ove dvije granične vrijednosti nalaze se tonovi sive boje. S obzirom na to da svaki piksel može poprimiti jednu od ukupno 256 vrijednosti, može se zaključiti da je za njegovu reprezentaciju potrebno 8 bita, odnosno jedan bajt. Kao osnovna memorijska jedinica, bajt je pogodan za računarsku obradu, a pritom je i dovoljan za ljudsku percepciju slike bez degradacije. Naime, proširenjem opsega vrijednosti koje može poprimiti svaki piksel, ljudsko oko neće percipirati veći kvalitet slike. Primjer procesa uzorkovanja, kvantizacije i numeričke reprezentacije piksela slike dat je na slici 2.1, preuzetoj iz [1].

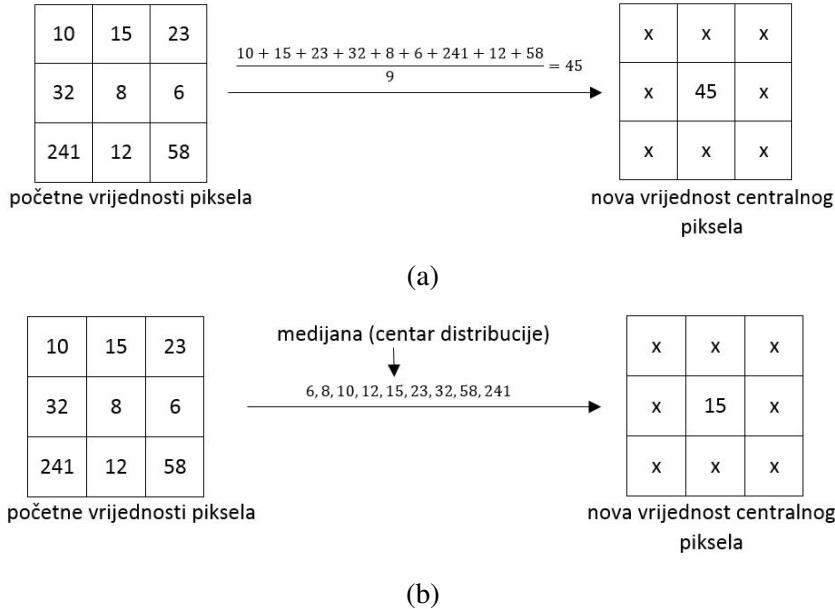
Za računarsku obradu slike su monohromatske slike od mnogo većeg značaja u odnosu na slike u boji, te obrada slike u boji najčešće podrazumijeva prвobitno prevođenje u monohromatski ekvivalent. Ova operacija spada u dio obrade slike koji se u literaturi naziva pretprocesiranjem (engl. *preprocessing*), a koja obuhvata i procese kao što su normalizacija sadržaja slike, odstranjivanje šuma, te morfoloшко procesiranje. Nakon ovih postupaka, koji se provode kao osnova za dalju obradu i izdvajanje parametara od interesa, vrši se segmentacija slike. Metode koje su obuhvaćene segmentacijom su određivanje praga, te detekcija tačaka, linija i rubova.



Slika 2.1: (a) Originalna slika; (b) projekcija slike na diskretnu fotoreceptorsku mrežu; (c) diskretizirana slika sa kvantiziranim vrijednostima boje; (d) numeričke vrijednosti elemenata slike (piksela) [1].



Slika 2.2: (a) Prikaz originalne slike "lena.jpg"; (b) Prikaz monohromatskog ekvivalenta slike "lena.jpg".



Slika 2.3: Princip rada različitih tipova filtera. (a) *Mean* filter; (b) *Median* filter.

Nakon toga, moguće je izvršiti ekstrakciju makroskopskih i mikroskopskih obilježja slike.

Svrha procesiranja slike je obično izolacija objekta od interesa, odnosno potiskivanje informacija o ostalim, manje bitnim objektima koji su prikazani. Termini koji se vežu uz postupak izolacije objekata su regija i oblik. Regija je povezani podskup tačaka slike, dobiven procesom segmentacije [53]. Regije se mogu podijeliti na unutrašnje i vanjske, pri čemu se pod unutrašnjom podrazumijeva ograničena regija koja nema šupljinu u unutrašnjosti, a pod vanjskom njen komplement, neograničena regija sa unutrašnjom šupljinom. Oblik je intuitivno jasan pojam, ali ga je teško definirati riječima. U literaturi se često njegova definicija pojednostavljuje poistovjećenjem sa bilo kojim povezanim skupom tačaka [1]. Dva najvažnija deskriptora oblika su kontura i skeleton [53], koji se koriste u različitim klasifikacijskim procedurama. Kontura predstavlja rub ili liniju kojom je određen ili ograničen oblik. S druge strane, skeleton je središnja linija karakteristika objekta, koja se obično dobiva iterativnim odstranjivanjem graničnih piksela, sve dok ne preostanu samo oni čijim bi se uklanjanjem prekinula povezanost [54].

U nastavku poglavlja detaljnije će se predstaviti koncept analize slike u boji i obraditi će se neke od osnovnih operacija računarske obrade slike, sa osvrtom na njihovu realizaciju korištenjem OpenCV okruženja. Za svrhu prikaza rada različitih metoda obrade slike u spomenutom okruženju, korištena je poznata slika "lena.jpg", koja je prikazana na slici 2.2.

2.1 Preprocesiranje

Preprocesiranje signala se vrši radi lakše ekstrakcije značajki koje će se koristiti u daljoj analizi. Dakle, signal se u okviru ovog postupka priprema za dalju obradu. U slučaju video zapisa, preprocesiranje se vrši na nivou frejma, odnosno slike. Uobičajeno je da se nad slikom izvode operacije filtriranje (najčešće u svrhu potiskivanja šuma), normalizacija podataka, morfološko procesiranje i sl., međutim, koji od navedenih postupaka će se provesti i na koji način, zavisi od konkretnog slučaja.

U okviru ovog potpoglavlja, bit će opisane tehnike za potiskivanje šuma i neke od morfolo-



Slika 2.4: Rezultat filtriranja u OpenCV okruženju, za maksimalnu veličinu kernela jednaku 31. (a) Filtriranje po medijani. (b) Filtriranje po srednjoj vrijednosti.

ških operacija koje su relevantne za algoritam implementiran u okviru ovog rada.

2.1.1 Filtriranje

Prilikom procesa skeniranja slike često dolazi do njene degradacije u vidu pojave šuma. Šum se javlja i na digitalnim fotografijama uslijed različitih faktora, kao što su prašina ili vodene čestice na objektivu fotoaparata ili kamere i sl. Kako bi se poboljšao kvalitet slike, potrebno je smanjiti količinu šuma u što većoj mjeri. Tehnika koja se često koristi u tu svrhu je zaglađivanje (engl. *smoothing*), koja podrazumijeva proračun srednje vrijednosti piksela koji se nalaze u proizvoljno velikoj okolini u odnosu na razmatrani i dodjelu proračunate vrijednosti istom. Nedostatak ove metode je pojava degradacije oštih rubova slike, koji nose značajne informacije. Kako bi se spriječila ova pojava, pri usrednjavanju je potrebno uzeti u obzir samo tačke bliske posmatranoj, jer je na taj način povećana vjerovatnoća razmatranja tačaka istih ili sličnih karakteristika [30]. Kao rezultat, ova metoda najčešće daje zamagljenu (engl. *blurred*) sliku, a zamagljenje je manje što se uža okolina za usrednjavanje uzima u obzir. Filter koji se primjenjuje u ovu svrhu najčešće se naziva *mean* ili *averaging* filterom. Primjer njegovog rada prikazan je na slici 2.3a.

Sve metode za potiskivanje šuma se generalno mogu podijeliti u kategorije linearnih i nelinearnih metoda. Linearno potiskivanje šuma u prostornom domenu podrazumijeva formiranje tzv. maski za odstranjivanje šuma, koje predstavljaju filtere napisane u vidu matrica. Kako se šum nalazi na visokim frekvencijama, ove matrice su niskopropusni filteri. Filtriranje po medijani (engl. *median filtering*) je najčešće primjenjivana metoda iz skupine nelinearnih tehnika odstranjivanja šuma. Kao i metoda zamagljenja, temelji se na ideji da se posmatrani piksel slike, prepoznat kao šum, zamjeni pikselom čija je vrijednost jednaka medijani njegovih susjednih tačaka, tj. vrijednosti koja određuje sredinu distribucije piksela obuhvaćenih maskom za filtriranje. Razlika u odnosu na metodu zamagljenja je u tome da je u ovom slučaju degradacija ivica mnogo manja, zbog čega se filtriranje može iterativno ponavljati [30]. Primjer funkcioniranja *median* filtera prikazan je na slici 2.3b. Često se u literaturi ovaj tip filtriranja naziva filtriranjem po srednjoj vrijednosti. Međutim, ovdje će se usvojiti termin filtriranja po medijani, s obzirom na to da će se koristiti i *averaging*, tj. *mean* filter.

Primjer rezultata filtriranja po medijani u OpenCV okruženju dat je na slici ??, dok se filtriranjem po srednjoj vrijednosti dobiva rezultat prikazan na slici ???. OpenCV za ovu svrhu implementira funkcije *medianBlur* i *blur*, respektivno. U prvoj navedenoj funkciji se koristi parametar označen kao *MAX_KERNEL_LENGTH* za navođenje linearne veličine aperture (neparan broj), te parametar u kom se prosljeđuju dimenzije matrice koja se izdvaja oko posmatranog piksela, a čije se usrednjavanje vrši. Treba napomenuti da funkcija *medianBlur* može direktno izvršiti *median* filtriranje na slikama u boji, što u drugim okruženjima najčešće nije slučaj. Naprimjer, MATLAB ne nudi tu mogućnost, već zahtijeva prvo bitno filtriranje za svaku od tri matrice boja ponaosob. Slično filtriranju po medijani, i za filtriranje po srednjoj vrijednosti se definira veličina jezgra za usrednjavanje i provodi se prethodno opisana procedura.

2.1.2 Morfološke operacije

Morfološke operacije obuhvataju skup postupaka koji se vrše radi olakšavanja procesa ekstrakcije značajki objekta, kao što su njegove granice i skelet [55]. Najčešće se upotrebljavaju operacije dilatacije (engl. *dilation*), erozije (engl. *erosion*), morfološkog otvaranja (engl. *morphological opening*), te morfološkog zatvaranja (engl. *morphological closing*).

Usljed slabog ili promjenjivog osvjetljenja na video zapisu, otežana je detekcija objekta korištenjem informacije o njegovoj boji. S druge strane, ukoliko se vrši detekcija oblika objekta, dodatnu poteškoću predstavlja pojava njegovog brzog kretanja, čime se vizuelno gubi informacija o obliku. Kako bi se omogućilo naglašavanje značajki objekta, poželjno je koristiti kombinaciju metoda dilatacije i erozije.

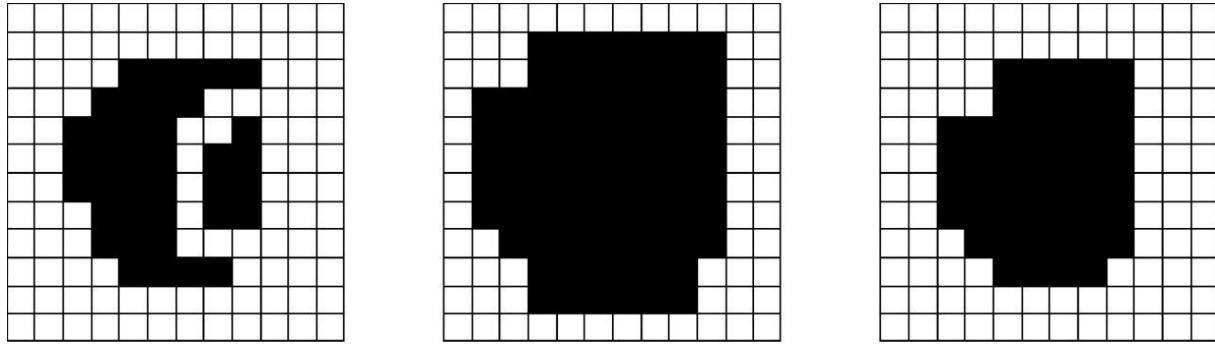
Zadatak dilatacije je podebljavanje, tj. proširenje oblika prikazanih na slici, dok je erozija njoj suprotna operacija, koja uzrokuje smanjenje veličine prikazanih objekata. Dilatacija se često upotrebljava za naglašavanje oblika objekta. S druge strane, erozija se najčešće primjenjuje radi eliminacije nebitnih detalja sa binarne slike [56], koji obično predstavljaju šum.

Morfološke operacije se provode korištenjem tzv. strukturnog elementa. Rezultat procesiranja zavisi od oblika, veličine i referentne tačke, tj. tačke usidrenja (engl. *anchor point*) odabranog elementa. Njegovom translacijom preko sastavnih piksela slike i konvolucijom s originalnom slikom se omogućava postizanje željenih rezultata. U slučaju dilatacije, strukturni element se ponaša kao operator lokalnog maksimuma. Vrši se pretraga najveće vrijednosti piksela iz skupa piksela obuhvaćenih strukturnim elementom, nakon čega se vrijednost tačke usidrenja zamjenjuje maksimalnom pronađenom vrijednošću. Slično, strukturni element se u slučaju erozije ponaša kao operator lokalnog minimuma i postupak se vrši u skladu sa analogijom prethodno opisanoj proceduri.

Navedene operacije mogu se ponavljati i više puta jedna za drugom, kako bi se postigli što bolji rezultati. Jedan od primjera prvo bitne primjene dilatacije, a potom erozije, prikazan je na slici 2.5.

2.2 Ekstrakcija značajki

Nakon provođenja preprocesiranja, slika (frejm videa) je spremna za dalju obradu. Kako bi se mogla provesti dalja analiza, potrebno je izdvojiti značajke od interesa. S obzirom na to da se algoritam za praćenje objekata koji je implementiran u okviru ovog rada zasniva na pretrazi po boji, naročito je interesantno analizirati modele boja koji se mogu primijeniti u obradi slike, kao i histograme boja. Pored informacije o boji, često se koriste i značajke teksture i oblika, zbog čega će se u nastavku teksta nešto detaljnije obraditi svaka od spomenutih karakteristika.



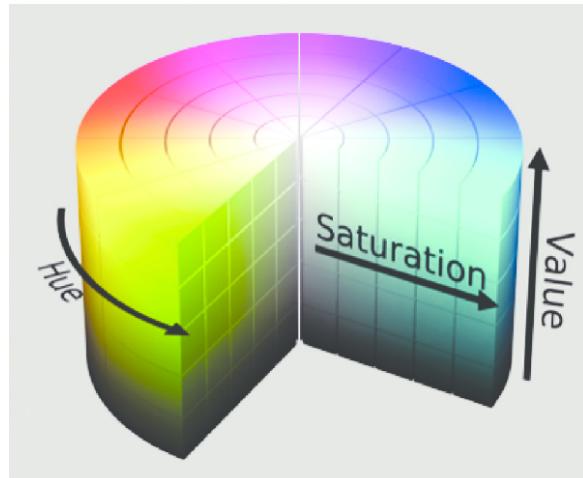
Slika 2.5: Rezultat sukcesivne primjene morfoloških operacija dilatacije i erozije u postupku preprosiranja slike [2].

2.2.1 Model boja

Pri analizi slika u boji, svjetlost se mjeri na način da se svakoj tački uređenog para prostornih koordinata (x,y) pridružuje uređena trojka (r,g,b), pomoću koje je moguće predstaviti sve boje iz spektra [57]. Oznake r , g i b se odnose na crvenu (engl. *red*), zelenu (engl. *green*) i plavu (engl. *blue*) boju, respektivno. Ove tri boje se koriste u računarskoj reprezentaciji slike iz razloga što ljudsko oko pored receptorskih ćelija za percepciju intenziteta svjetlosti (štapići), koji su zaduženi za raspoznavanje različitih monohromatskih tonova, posjeduje i tri vrste ćelija za percepciju boje (čunjići), i to upravo za crvenu, zelenu i plavu boju. Miješanjem navedenih boja u odgovarajućem omjeru, mogu se dobiti sve nijanse boja koje ljudsko oko može percipirati.

Prostor boje je matematička reprezentacija boje [58], te postoji nekoliko prostora, koji se koriste u različite svrhe. Kako je već spomenuto, za prikaz slike u boji računarska grafika koristi RGB (engl. *Red, Green, Blue*) model. Ovaj model je aditivan, tj. zasniva se na dodavanju primarnih boja svjetlosti, dok je nasuprot njemu CMY (engl. *Cyan, Magenta, Yellow*) model subtraktivni, te koristi sekundarne boje tirkiznu, grimiznu i žutu. Kako se mijesanjem sekundarnih boja svjetlosti ne može proizvesti potpuno crna boja, u svrhu printanja u boji, formiran je model koji dodaje crnu boju, a označen je sa CMYK (engl. *Cyan, Magenta, Yellow, Black*). Razlog zbog kog se RGB koristi za računarski prikaz, a CMYK za printanje je taj da monitori emituju svjetlost, dok je tinta reflektuje [59]. U video sistemima se koriste modeli koji razdvajaju komponentu osvjetljenja od komponente boje, a to su YIQ, YUV i YCbCr. YIQ i YUV su analogni modeli koji se koriste u dva različita sistema, dok je YCbCr digitalni standard. Komponenta Y u spomenutim modelima označava osvjetljenje, odnosno luminantnu komponentu. Sa I i Q su označene međusobno okomite komponente u fazi (engl. *in-phase*) i u kvadraturi (engl. *quadrature*), respektivno. U i V su hrominantne komponente, tj. komponente boje, što vrijedi i za Cb i Cr, s tim da Cb označava komponentu plave, a Cr komponentu crvene boje. Svi navedeni modeli su prilagođeni računarskoj reprezentaciji slike u boji, a ne ljudskom vizuelnom sistemu. Modeli koji prikazuju sliku na način koji je sličniji ljudskoj percepciji su HSI (engl. *Hue, Saturation, Intensity*), HSB (engl. *Hue, Saturation, Brightness*) i HSV (engl. *Hue, Saturation, Value*), koji obuhvataju nijansu, zasićenje i intenzitet, odnosno svjetlinu i vrijednost, respektivno. Nijansa označava gdje se određena boja nalazi u spektru boja, zasićenje koliko je ta boja čista, tj. bez primjesa, a intenzitet/svjetlina/vrijednost koliko je svijetla ta boja. Grafički se HSV model može prikazati u vidu cilindra, kao na slici 2.6.

Postoji međusobna zavisnost između prostora boja, zbog čega je iz RGB informacija sadržanih na računaru ili nekom drugom digitalnom uređaju moguće izvesti bilo koji prostor boja. Kako je HSV model najsličniji ljudskom vizualnom sistemu, a RGB se uglavnom koristi za



Slika 2.6: Grafički prikaz HSV modela boja [3].



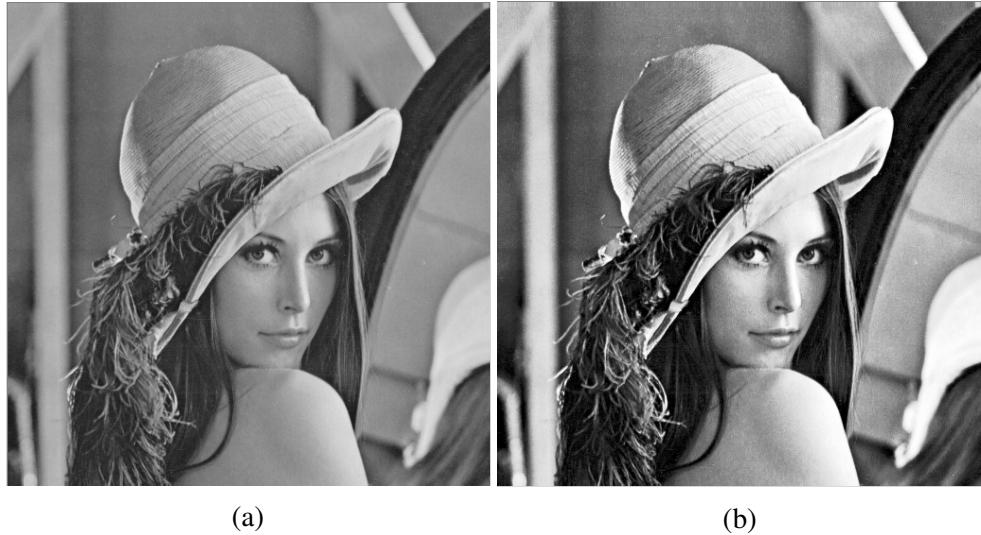
Slika 2.7: (a) Rezultat konverzije iz RGB u HSV prostor boja; (b) Rezultat konverzije iz RGB u binarni prostor boja.

računarsku reprezentaciju slike, najčešća konverzija koja se vrši je iz RGB u HSV. U OpenCV okruženju je ovo moguće izvršiti primjenom opcije `CV_BGR2HSV` u okviru funkcije `cvtColor`. Rezultat je dat na slici 2.7a.

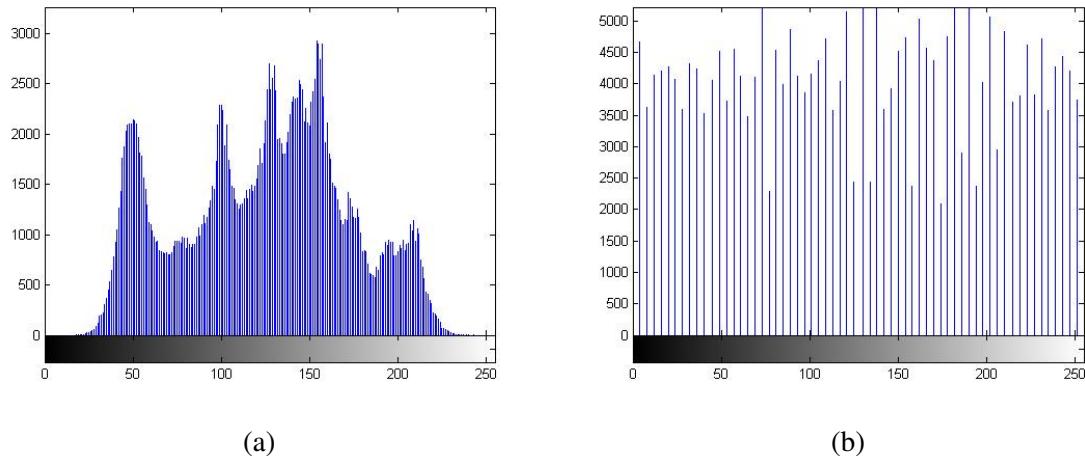
Često se u analizi slike javlja i potreba za konverzijom slike u boji u binarnu sliku, gdje pikseli mogu imati samo jednu od dvije vrijednosti: 0 (crna boja) i 1 (bijela boja). U OpenCV okruženju, ovo se vrši upotrebom `CV_BGR2GRAY` u okviru funkcije `cvtColor`, te naknadnom primjenom `THRESH_BINARY` u sklopu funkcije `threshold`. Primjer rezultata prikazan je na slici 2.7b. Ukoliko je cilj iz slike u boji dobiti samo njen monohromatski ekvivalent, nema potrebe za primjenom prethodno spomenute funkcije pragovanja.

2.2.2 Histogram

U cilju izdvajanja značajki (engl. *features*) objekta prikazanog na slici, često se javlja potreba za povećanjem kontrasta, kako bi se naglasile korisne informacije koje je potrebno izdvojiti.



Slika 2.8: (a) Monohromatski ekvivalent slike "lena.jpg". (b) Rezultat ekvalizacije histograma.



Slika 2.9: Prikaz histograma boje za sliku "lena.jpg". (a) Originalni histogram. (b) Ekvalizirani histogram.

Povećanjem kontrasta povećavaju se vrijednosti svijetlih, a istovremeno smanjuju vrijednosti tamnih piksela, čime oni postaju još svjetlijci, odnosno tamniji, respektivno.

Histogram slike je grafik koji prikazuje broj piksela za svaki nivo intenziteta svjetlosti koji je prisutan na slici, što ga čini dobrom indikatorom zastupljenog kontrasta [54]. Ukoliko histogram prekriva cijeli dinamički opseg (ako pikseli obuhvataju cijeli skup vrijednosti od 0 do 255), kontrast je izražen na slici. S druge strane, kontrast je loš ukoliko je na histogramu obuhvaćen samo dio dinamičkog opsega. Normalizacija kontrasta vrši se ekvalizacijom histograma, koja podrazumijeva proširenje postojećih vrijednosti piksela na cijeli dinamički opseg.

U OpenCV okruženju se ekvalizacija histograma vrši pomoću funkcije *equalizeHist*. Njenom primjenom nad slikom "lena.jpg", dobiva se rezultat prikazan na slici 2.8b. Izgled histograma za taj slučaj je dat na grafiku 2.9b, dok graf 2.9a prikazuje histogram originalne slike u sivim tonovima.

2.2.3 Tekstura

Tekstura objekta je još jedna od značajki koje ga opisuju, a na osnovu koje je moguće detektovati objekat na slici. Samim tim, nameće se zaključak da se primjenom detekcije objekta po teksturi na sukcesivnim frejmovima videa može postići praćenje trajektorije objekta na video zapisu.

U računarskoj analizi slike, informacije o teksturi se mogu dobiti iz statističke raspodjele kombinacija intenziteta piksela na specificiranim pozicijama, relativnim jedna u odnosu na drugu. Zavisno od broja obuhvaćenih piksela, razlikuju se statistike prvog, drugog i višeg reda [60]. Uobičajeno je korištenje statistike drugog reda, poznate pod skraćenicom GLCM (engl. *Gray-Level Co-occurrence Matrix*). Ova metoda se zasniva na vezi teksture sa statističkom distribucijom tonova sive boje [61].

Matematski se GLCM može definirati kao funkcija vjerovatnoće F nivoa sive boje i koji se pojavljuje na slici na udaljenosti d , pod uglom θ u odnosu na nivo sive boje j u tačkama x i y , kao što je prikazano relacijom 2.1 [61]. U navedenoj relaciji su sa Δx i Δy označene udaljenosti između razmatranih piksela po x i y osi, respektivno.

$$C_{\Delta x, \Delta y}(i, j) = \sum_{x=1}^n \sum_{y=1}^m \begin{cases} 1, & \text{za } F(x, y) = i \wedge F(x + \Delta x, y + \Delta y) = j \\ 0, & \text{inače} \end{cases} \quad (2.1)$$

Razmatraju se četiri rasporeda piksela, odnosno, njihove relativne pozicije pod uglovima od 0° , 45° , 90° i 135° . Ugao od 0° podrazumijeva da se dva piksela nalaze jedan do drugog (desno/lijevo u odnosu na susjedni piksel). Pod uglom od 45° se nalaze dvije tačke od kojih je jedna pozicionirana gore desno ili dolje lijevo u odnosu na drugu. 90° ukazuje na to da se pikseli nalaze jedan ispod/iznad drugog, tj. da su pozicionirani vertikalno jedan u odnosu na drugi, dok ugao od 135° podrazumijeva da su pikseli razmješteni tako da se jedan nalazi gore lijevo ili dolje desno u odnosu na drugi. Slika 2.10 prikazuje primjer formiranja GLCM matrice za slučaj razmatranja pozicija piksela pod uglovima od 0° i 45° . Za svaki par piksela (i, j) se na odgovarajućoj poziciji u matrici upisuje broj puta koliko se oni nalaze u razmatranom rasporedu. U slučaju analize horizontalnog rasporeda uređenog para $(1, 1)$, uočava se da se dva puta pojavljuju pod uglom od 0° , što je označeno crvenom bojom. Uređeni par $(1, 3)$ se u ovom rasporedu javlja samo jednom i naznačen je plavom bojom, dok se uočavaju dva ponavljanja uređenog para $(2, 5)$ pod uglom od 45° , što je označeno zelenom bojom na slici.

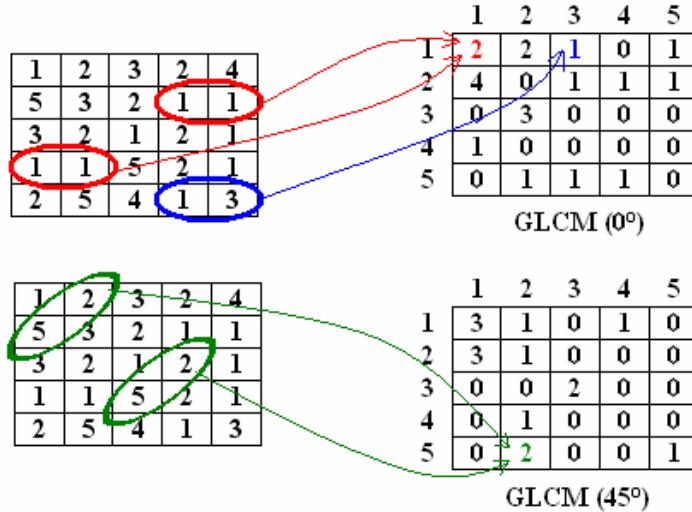
Na osnovu koeficijenata c_{ij} može se izvršiti proračun kontrasta, entropije, energije (uniformnosti) i homogenosti, na način prikazan u relacijama 2.2-2.5, respektivno [60]. Proračunom kontrasta mjeru se lokalne varijacije, entropija je mjera statističke slučajnosti, energija predstavlja sumu kvadratnih vrijednosti elemenata GLCM matrice, dok homogenost mjeri raspodjelu elemenata u odnosu na glavnu dijagonalu iste matrice [4]. Usporedbom razlike između vrijednosti ovih veličina za dvije slike, može se utvrditi njihova teksturalna sličnost.

$$C = \sum_i \sum_j (i - j)^2 c_{ij} \quad (2.2)$$

$$E = \sum_i \sum_j c_{ij} \log c_{ij} \quad (2.3)$$

$$U = \sum_i \sum_j c_{ij}^2 \quad (2.4)$$

$$H = \sum_i \sum_j \frac{c_{ij}}{i + |i - j|} \quad (2.5)$$



Slika 2.10: Primjer formiranja GLCM matrice za raspored piksela pod uglovima od 0° i 45° [4].

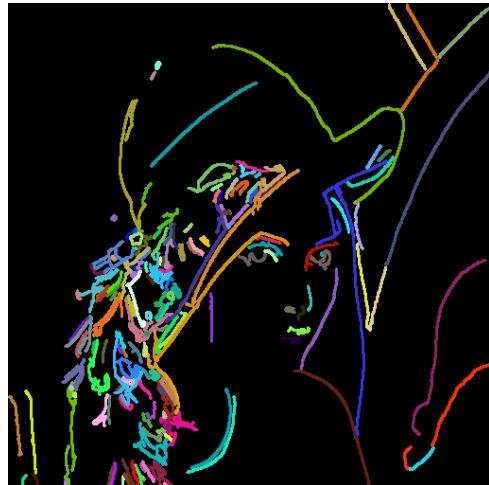
Dimenzije GLCM matrice jednake su broju nivoa sive boje na razmatranoj slici. S obzirom na to, ove matrice najčešće sadrže veliki broj podataka i složene su za obradu. S druge strane, obimni podaci su bitni za izvođenje pouzdanih statističkih zaključaka, tj. estimaciju združene gustoće vjerovatnoće piksela razmatranih sivih tonova. Zbog toga je potrebno pronaći kompromis između brzine obrade i tačnosti rezultata. Za smanjenje dimenzija GLCM matrice se vrši reduciranje broja kvantizacionih nivoa, te se koriste veći prozori mjerena (engl. *measurement window*). Ovim postupcima se vrši degradacija kvaliteta rezultata GLCM algoritma, dok se proračuni izvršavaju brže.

2.2.4 Oblik

Jedan od načina detekcije konture objekta prikazanog na slici je da se odrede njegove granice na osnovu detektovanih rubova objekta. Drugi način je apriorno poznavanje krajnjih tačaka (engl. *end points*) konture, gdje se iterativnim postupkom formiraju particije objekta i traži najizraženiji rub, okomit na liniju koja povezuje krajnje tačke svake particije. Detektovani najizraženiji rub na okomici bliskoj pravoj koja povezuje krajnje tačke razmatrane particije se prihvata kao novi element konture, nakon čega se postupak ponavlja [30]. Metode segmentacije koje zahtijeva ovaj proces su detekcija tačaka, linija i rubova.

Česta metoda koja se koristi za detekciju rubova je *Canny* detekcija (engl. *Canny edge detection*). Ova metoda je optimalna po tri kriterija: kriterij detekcije (bitni rubovi se moraju uzeti u obzir), lokalizacijski kriterij (udaljenost između stvarne i locirane pozicije ruba mora biti minimalna) i kriterij jedinstvenog odziva (višestruke mogućnosti se svode na jedan rub) [30]. Na slici 2.11 su prikazani rezultati dobiveni primjenom ove metode u OpenCV okruženju.

U programu prilagođenom OpenCV okruženju, potrebno je prvo bitno izvršiti pretvorbu slike u njen monohromatski ekvivalent, nakon čega se mogu detektovati *Canny* rubovi istoimenom funkcijom, a potom konture funkcijom *findContours*, nakon čega se one iscrtavaju različitim bojama.



Slika 2.11: Rezultat detekcije kontura primjenom *Canny* detektora rubova u OpenCV okruženju, za prag jednak 100 (od maksimalno 255).

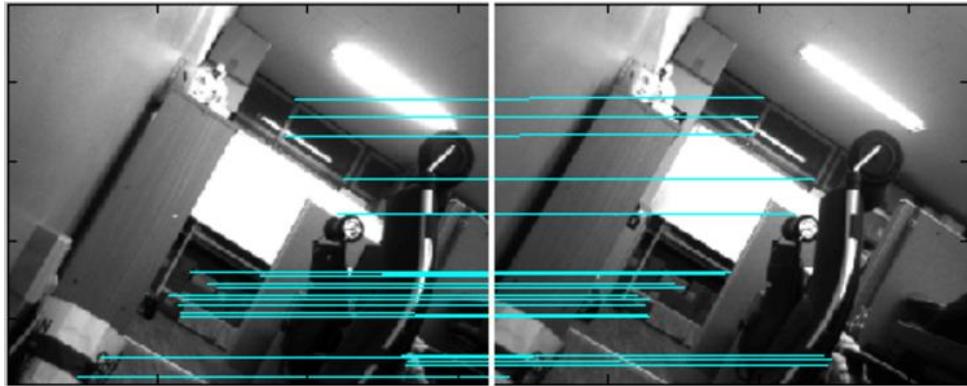
2.3 Detektori i deskriptori značajki

Detektori i deskriptori značajki se često koriste u računarskoj analizi pri procesu usporedbe dvije ili više slika, te obuhvataju dvije faze. U prvoj fazi se vrši pretraga tzv. ključnih tačaka (engl. *key points*), koje se ekstraktuju iz karakterističnih dijelova slike, kao što su rubovi prikazanih objekata. Druga faza podrazumijeva obuhvatanje piksela koji su susjedni detektovanim ključnim tačkama, nakon čega se vrši proračun deskriptora značajki za svaku od izdvojenih regija [62]. U skupinu detektora i deskriptora ubrajaju se SIFT (engl. *Scale Invariant Feature Transform*), SURF (engl. *Speed Up Robust Features*), FAST (engl. *Features from Accelerated Segment Test*) i BRIEF (engl. *Binary Robust Independent Elementary Features*). Među navedenim, najčešće primjenjuju se SURF i SIFT algoritmi.

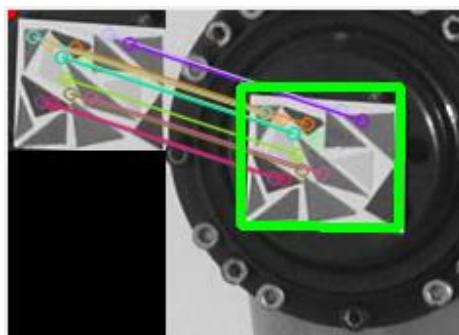
SIFT obezbeđuje invarijantnost izdvojenih značajki objekta na rotaciju i skaliranje. To znači da će, bez obzira na promjenu veličine i orientacije objekta, njegove značajke ostati iste. Algoritam se sastoji iz četiri osnovna koraka:

- detekcija ekstrema u skaliranom prostoru,
- lokalizacija ključnih tačaka,
- dodjela orientacije,
- generiranje deskriptora [63].

Pod terminom detekcije u skaliranom prostoru se podrazumijeva korištenje prozora većih ili manjih dimenzija za detekciju ključnih tačaka, u zavisnosti od količine detalja koje objekat sadrži. Što je više detalja, to je potrebno koristiti prozor manjih dimenzija. U skaliranom prostoru se potom vrši proračun razlike Gaussovih funkcija za različite vrijednosti skalirajućeg faktora σ . U zavisnosti od njegove vrijednosti, vrši se detekcija uglova i drugih karakteristika objekta različitih veličina. Pretragom lokalnih ekstremi za svaku uređenu trojku (x, y, σ) , gdje su x i y prostorne koordinate, dobiva se potencijalna ključna tačka u (x, y) za korištenu vrijednost parametra σ . Filtriranje potencijalnih ključnih tačaka radi dobivanja relevantnijih rezultata vrši se odbacivanjem tačaka niskog kontrasta. Preostalim tačkama se dodjeljuje orientacija kako bi se postigla invarijantnost na rotaciju. Ovaj postupak je baziran na pronalaženju lokalnog



Slika 2.12: Primjer rezultata pronalaženja sličnosti između dvije slike korištenjem SIFT algoritma sa 16 ključnih tačaka [5].



Slika 2.13: Primjer rezultata pronalaženja sličnosti između dvije slike korištenjem SURF algoritma u OpenCV okruženju.

gradijenta slike [63]. Magnituda i orientacija gradijenta se potom koriste u svrhu generiranja deskriptora za svaku ključnu tačku. Postupak se sastoji u tome da se oko svake od ključnih tačaka izdvaja nekoliko susjednih, te se za svaku tako formirana regiju generira orijentacioni histogram. Usporedba dvije slike onda podrazumijeva pronalaženje najsličnijih orijentacionih histograma, što odgovara detekciji istih ključnih tačaka. Primjer rezultata koji daje algoritam prikazan je na slici 2.12.

Proračuni koje podrazumijeva SIFT zahtijevaju veliko vrijeme izvršavanja algoritma, zbog čega je razvijen SURF. Ova dva algoritma imaju sličan princip rada, koji uključuje prethodno pobrojana četiri koraka navedena u sklopu analize SIFT algoritma. SURF umjesto proračuna razlike Gaussovih funkcija za detekciju ekstrema u skaliranom prostoru koristi filtriranje po srednjoj vrijednosti, što najviše ubrzava njegov rad u odnosu na SIFT. Pored toga, za lokalizaciju ključnih tačaka se koristi determinanta Hessijan matrice, koja omogućava dobre performanse i pouzdanost [63]. Za dodjelu orijentacije ključnim tačkama, izdvaja se grupa njoj susjednih piksela i vrši proračun odziva Haar-wavelet transformacije u horizontalnom i vertikalnom smjeru u odnosu na ključnu tačku. Ovo se vrši kako bi se postigla invarijantnost na rotaciju [63]. Primjer rezultata primjene SURF algoritma u OpenCV okruženju dat je na slici 2.13. S lijeve strane je prikazana slika objekta koji se pretražuje, dok je na desnoj strani data slika na kojoj ga je potrebno pronaći. Ključne tačke su označene različitim bojama i detektovani parovi tačaka su povezani odgovarajućim linijama. Pronađeni objekat je naglašen zelenim pravougaonikom.

Preostala dva spomenuta algoritma nisu toliko popularna kao SURF i SIFT, zbog čega se neće ulaziti u detalje principa njihovog rada. FAST i BRIEF algoritmi imaju simbolične skraćenice, koje ukazuju na novine koje su uvedene u odnosu na prethodno predstavljene algoritme. FAST omogućava još brže procesiranje u odnosu na SURF, što ga čini pogodnim za korištenje u aplikacijama koje zahtijevaju rad u realnom vremenu, dok BRIEF koristi deskriptor sa manje tačaka. Na taj način se zauzima manje memorijskih resursa i analiza se brže izvršava, ali su rezultati nešto slabiji.

* * *

Računarska analiza slike obuhvata širok spektar metoda za naglašavanje korisnih informacija sadržanih u slici, koje su potrebne radi dalje obrade. Primjenom različitih kombinacija postupaka mogu se ostvariti isti rezultati, ali je provođenje preprocesiranja svakako vrlo bitno za ekstrakciju relevantnih značajki. U okviru ovog poglavlja dat je kratak teorijski opis postupaka koji su primjenjeni u algoritmu implementiranom u okviru ovog rada, s osvrtom na način na koji se oni provode u OpenCV okruženju.

Poglavlje 3

Praćenje zasnovano na izotropskim jezgrima i pomjeranju srednje vrijednosti

3.1 Prostorno filtriranje (maskiranje)

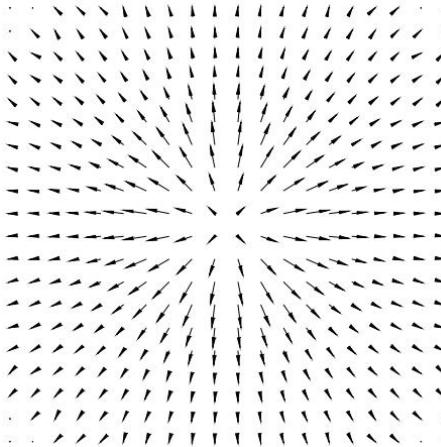
Proces filtriranja obično se vezuje za operacije koje se vrše u frekvencijskom domenu, gdje je uobičajeno da se nad signalima koji se procesiraju primjenjuje Fourierova transformacija. Termin prostorno filtriranje (engl. *spatial filtering*) se upotrebljava u literaturi kako bi se naglasila različitost u odnosu na spomenuto procesiranje, a u slučaju obrade slike, podrazumijeva direktnu primjenu filtera nad sastavnim pikselima slike [56]. Filtriranje se vrši korištenjem matrice određenih dimenzija, čiji su članovi koeficijenti koji utječu na vrijednosti piksela slike od interesa, kada se nad njima primjeni filter. Da bi se postigla simetrija oko centralnog piksela, dimenzije matrice su obično neparne. Kako se na opisani način vrši "maskiranje" piksela, matrica predstavlja masku za onaj dio slike nad kojim je primjenjena, a koji je istih dimenzija kao i sama matrica. Pored termina maska, mogu se upotrijebiti i termini filter, prozor ili jezgro (kernel) [56].

Maskiranje je iterativno i vrši se pomjeranjem matrice po slici, obuhvatajući na taj način različite skupove piksela, a postupak se ponavlja sve dok svi pikseli slike ne budu podvrgnuti filtriranju. Utjecaj maskiranja zavisi od predefinirane veze između koeficijenata kojim je definirana maska i vrijednosti piksela slike nad kojim je ista primjenjena.

U slučaju linearog filtriranja, odziv filtera jednak je sumi proizvoda koeficijenata maske i odgovarajućih piksela obuhvaćenih tom maskom, što odgovara procesu konvolucije u frekvencijskom domenu. Iz tog razloga, linearni filteri često se nazivaju i konvolucionim maskama ili konvolucionim jezgrima (engl. *convolution kernel*) [56]. Drugim riječima, rezultat linearog filtriranja za svaki pojedini piksel obuhvaćen maskom je srednja vrijednost svih obuhvaćenih piksela, multiplicirana odgovarajućim koeficijentom maske.

Nelinearno prostorno filtriranje zavisi od vrijednosti piksela obuhvaćenih maskom i ne koristi nužno sumu proizvoda koeficijenata, kao što je to slučaj sa linearnim filterima. Međutim, oba tipa prostornog maskiranja podrazumijevaju prevlačenje maske preko slike, dok se svi pikseli ne podvrgnu proceduri. Primjer nelinearnog maskiranja je filtriranje po srednjoj vrijednosti, koje je detaljnije objašnjeno u okviru prethodnog poglavlja.

Za potrebe praćenja objekta na video zapisu, prostorno maskiranje se prvenstveno koristi za lokalizaciju objekta od interesa, iako se upotrebljava i za potiskivanje šuma, kao i u ostalim aplikacijama koje koriste obradu slike. U konkretnom slučaju, njegovom primjenom se može obezbijediti naglašavanje značajki, nakon čega se najčešće formira histogram na osnovu kog će se vršiti pretraga ciljnog objekta na narednim frejmovima [13].



Slika 3.1: Nagib izotropske Gaussove funkcije, indiciran strelicama koje predstavljaju gradijentne vektore [6].

Naime, lokalne promjene intenziteta nose bitne informacije o pojavi objekta od interesa. Ove promjene se dešavaju na samom objektu, ili na granici između objekta i pozadine [11]. Uobičajena je pojava da se granični pikseli objekta djelimično "stapaju" sa okolnim pikselima, zbog čega su manje relevantni u odnosu na centralne. Ideja korištenja jezgra u ovom slučaju je upravo naglašavanje središnjih piksela, a smanjenje utjecaja rubnih, prilikom pretrage objekta na narednom frejmu. Što se tiče promjena na samom objektu, one su uzrokovane refleksijom pojedinih dijelova objekta, koji mogu praviti sjenu ili ukazivati na teksturu [11]. Ukoliko je to slučaj, i tekstura se može koristiti kao značajka na osnovu koje će se vršiti praćenje objekta.

3.1.1 Izotropska jezgra

Izotropska jezgra su simetrična jezgra čije je djelovanje isto u svim pravcima (izotropno), odakle i potiče njihov naziv. Drugim riječima, funkcija kojom je jezgro definirano se ne mijenja.

Za potrebe praćenja objekata na video zapisu, najčešće se koriste Gaussovo (normalno) $K_N(x)$ i Epanechnikovo $K_E(x)$ jezgro, kojima odgovaraju relacije 3.1 i 3.2, respektivno [30]. Normalizacijska konstanta je pritom označena sa c i uvijek ima pozitivnu vrijednost, dok je sa x označena vrijednost prostorne koordinate u jednodimenzionalnom prostoru. Pored navedenih, mogu se koristiti i uniformno, trougaono jezgro i sl. Najjednostavnije za implementaciju je uniformno jezgro, međutim, ono ne daje zadovoljavajuće performanse u većini slučajeva [33], zbog čega se ipak preferiraju složenije varijante. Kako brzina konvergencije ka ciljnoj poziciji zavisi upravo od odabranog jezgra, Epanechnikovo jezgro ima prednost u odnosu na Gaussovo, s obzirom na to da omogućava konvergenciju u konačnom broju koraka, što nije slučaj sa normalnim jezrom.

$$K_N(x) = ce^{-\frac{1}{2}\|x\|^2} \quad (3.1)$$

$$K_E(x) = \begin{cases} c(1 - \|x\|^2), & \text{za } \|x\| \leq 1 \\ 0, & \text{inače} \end{cases} \quad (3.2)$$

Nagib izotropske Gaussove funkcije je indiciran strelicama na slici 3.1, s koje se može uočiti radikalna simetrija. Prikazane strelice nazivaju se gradijentnim vektorima [6], koji u *mean-shift*



Slika 3.2: Prikaz tipova jezgara. (a) Radijalno simetrično izotropsko jezgro, kružnog oblika; (b) Simetrično jezgro anizotropskog, pravougaonog oblika; (c) Simetrično jezgro anizotropskog, eliptičnog oblika; (d) Asimetrično jezgro koje prati oblik objekta od interesa [7].

proceduri služe za pomjeranje centroida oblasti pretrage objekta prema slivu privlačnosti. Ovaj postupak će detaljnije biti objašnjen u narednom poglavljju.

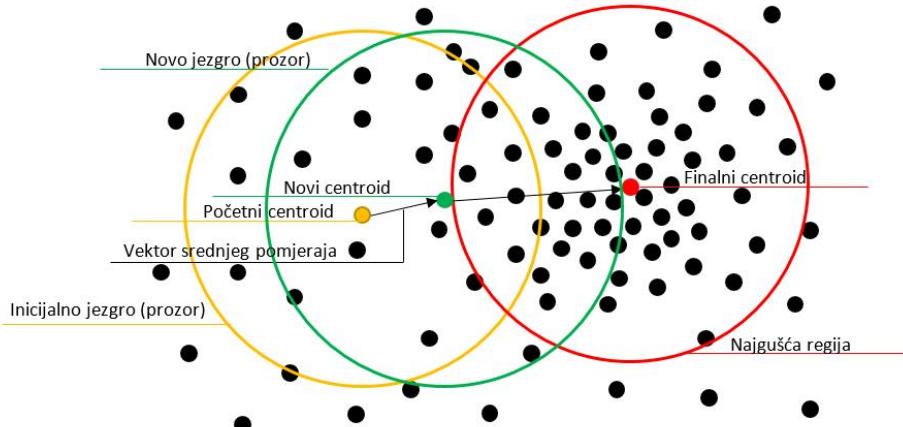
Generalno, izotropska jezgra se koriste zbog svoje jednostavnosti, ali njihova primjena ima određena ograničenja u domenu praćenja objekata na video zapisu. U prvom redu, ova ograničenja se odnose na njihovu konstantnu veličinu [7], što pogoršava performanse algoritma ukoliko dolazi do skaliranja objekta u toku videa. Drugo ograničenje je uzrokovano radijalnom simetrijom, koja ne omogućava dobro praćenje oblika objekta. Najpoželjnije bi bilo vršiti maskiranje objekta jezgrom identičnog oblika, što znači da izotropska jezgra imaju optimalno djelovanje isključivo ukoliko je i sam objekat koji se prati radijalno simetričan. Ako to nije slučaj, može se upotrijebiti jezgro manjih dimenzija kojim će biti obuhvaćen samo jedan dio objekta, bez uključenja okolnih elemenata ili podloge u proces maskiranja, ili jezgro većih dimenzija kojim će se obuhvatiti cijeli objekat ili jedan njegov dio, ali i jedan dio njegovog okruženja. Ovim se uzrokuje pomjeranje estimirane pozicije objekta od interesa i u tom slučaju anizotropska jezgra daju bolje performanse. Iz te skupine, najčešće se koristi eliptično jezgro, dok je rijetka upotreba jezgara koja u potpunosti adaptiraju svoj oblik objektu koji se prati, uslijed velike računarske kompleksnosti koju taj proces zahtijeva.

Radi bolje percepcije pobrojanih tipova jezgara, njihovi primjeri su dati na slici 3.2, preuzeutoj iz [7]. Treba napomenuti da su u ovom slučaju crvenim linijama označeni oblici jezgara u odnosu na objekat koji se prati, a da prilikom izvođenja algoritma ista oznaka ukazuje na regiju u kojoj se nalazi detektovani objekat, a ne na tip jezgra koji je korišten u implementaciji.

3.2 Algoritam pomjeranja srednje vrijednosti

Procedura pomjeranja srednje vrijednosti (engl. *mean-shift*) je neparametarska tehnika za analizu kompleksnog multimodalnog prostora značajki i identifikaciju klastera značajki [30], a obuhvata dva koraka koja se sukcesivno ponavljaju: proračun vektora srednjeg pomaka i translaciju jezgra u zavisnosti od proračunate vrijednosti. Ovim postupkom dolazi do konvergencije u tačku gdje estimirana vrijednost gustoće ima nulti gradijent [22], što u slučaju praćenja objekta podrazumijeva pronalaženje njegove najvjerojatnije pozicije na analiziranoj slici. Naime, gradijent estimacije gustoće jezgra daje kao rezultat vektor srednjeg pomjeranja, a njegova vrijednost jednaka nuli ukazuje na to da nema potrebe za pomjeranjem, tj. da je trenutna pozicija ujedno i najvjerojatnija.

Da bi se čitava procedura mogla izvesti, koristi se prostor značajki. Ovaj prostor se, u pogledu računarske obrade slike, sastoji od kvantitativnih osobina slike, mapiranih u tačke višedimenzionalnog prostora u kom je slika predstavljena [30]. Nakon što se izvrši mapiranje svih tačaka slike u prostor značajki, može se izvršiti analiza gustoće mapiranja. Ukoliko se dio mapiranih tačaka rasporedi u blizini jedna drugoj, one formiraju klaster. Može postojati više



Slika 3.3: Ilustrativni prikaz iterativnog *mean-shift* postupka.

klastera, što bi u kontekstu segmentacije slike odgovaralo prepoznavanju više objekata na slici, dok bi dio prostora u kom nema mapiranih tačaka odgovarao podlozi [30].

Na slici 3.3 je grafički predstavljen princip *mean-shift* procedure. Prvobitno se nasumično postavlja inicijalno jezgro, tj. prozor unutar analiziranog prostora značajki, te se određuje njegov centroid. Na spomenutoj slici, inicijalno jezgro i centroid su označeni narandžastom bojom. Nakon toga, vrši se proračun vektora srednjeg pomjeranja i na osnovu njega se translira jezgro na odgovarajuću poziciju, te mu se pridružuje novi centroid. Ovaj korak je označen zelenom bojom u navedenom primjeru. Postupak se ponavlja iterativno, sve dok ne dođe do konvergencije ka lokalnom maksimumu gustoće raspodjele mapiranih tačaka u prostoru značajki. Za taj slučaj, jezgro i odgovarajući centroid su označeni crvenom bojom na slici.

Može se uočiti da se u procesu detekcije klastera, tj. regije od interesa mogu mijenjati samo veličina i oblik jezgra. U inicijalnoj *mean-shift* proceduri se koriste isključivo radijalno simetrična jezgra, i to ona koja zadovoljavaju uslov iskazan relacijom 3.3, u kojoj je sa c označena striktno pozitivna normalizacijska konstanta, koja obezbeđuje da jezgro $K(x)$ zadovoljava uslov normiranja, a $k(x)$ se naziva profilom jezgra i definirano je samo za nenegativne vrijednosti x [22]. Gaussova i Epanechnikova jezgra, razmatrana u prethodnom poglavlju, zadovoljavaju relaciju 3.3 i najčešće se primjenjuju za potrebe algoritama zasnovanih na *mean-shift* proceduri.

$$K(x) = ck(\|x\|^2) \quad (3.3)$$

Estimacija gustoće jezgra je najpopularnija metoda za estimaciju u području prepoznavanja na osnovu paterna (engl. *pattern recognition*), gdje se često označava kao tehnika Parzenovog prozora. Multivarijantni estimator gustoće jezgra $\hat{f}(x)$ u d -dimenzionalnom prostoru R^d sa n tačaka $x_i, i = 1, \dots, n$ dat je relacijom 3.4, u kojoj je sa \mathbf{H} označena pozitivno definitna matrica propusnog opsega, dimenzija $d \times d$, izračunata u tački x , dok je sa $K(x)$ označeno jezgro [22], koje je za matricu \mathbf{H} definirano relacijom 3.5.

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_{\mathbf{H}}(x - x_i) \quad (3.4)$$

$$K_{\mathbf{H}}(x) = |\mathbf{H}|^{-\frac{1}{2}} K(\mathbf{H}^{-\frac{1}{2}} x) \quad (3.5)$$

Iz relacije 3.4 je vidljivo postojanje zavisnosti od matrice propusnog opsega \mathbf{H} . Ova matrica bi trebala biti što jednostavnijeg oblika, kako bi se smanjio broj parametara potrebnih za njen

proračun, te sama računarska kompleksnost. Zbog toga je ona najčešće dijagonalna ili proporcionalna matrici identiteta. Dijagonalna matrica u d -dimenzionalnom prostoru zahtijeva d vrijednosti parametara, dok druga navedena opcija traži samo jedan - onaj kojim će biti pomnožena matrica identiteta. S obzirom na navedeno, matrica definirana relacijom 3.6 ima prednost nad svim drugim varijantama i koristi se u *mean-shift* proceduri. Sa h je označen parametar propusnog opsega, koji uvijek ima pozitivnu vrijednost i on zapravo određuje veličinu jezgra. Uzimajući u obzir sve prethodno navedene relacije, estimator gustoće jezgra se može predstaviti relacijom 3.7 [22].

$$\mathbf{H} = h^2 \mathbf{I} \quad (3.6)$$

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (3.7)$$

Ono što čini proceduru pomjeranja srednje vrijednosti jednostavnijom i pogodnijom od ostalih rješenja jeste mogućnost identifikacije regija u kojima su najgušće raspoređene tačke značajki bez estimacije samih funkcija gustoće raspolođene. Umjesto toga, problem se svodi na estimaciju gradijenta gustoće, tj. pronalaženje $\nabla \hat{f}(x)$, iskazano relacijom 3.8 [30].

$$\nabla \hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n \nabla K\left(\frac{x-x_i}{h}\right) \quad (3.8)$$

Uzimajući u obzir vezu između jezgra $K(x)$ i njegovog profila $k(x)$, datu relacijom 3.9, može se doći do izraza u kom je omogućeno jasno razdvajanje dijela formulacije koji odgovara vektoru srednjeg pomjeranja i onog koji je proporcionalan estimatoru funkcije gustoće. Da bi spomenuta relacija bila važeća, mora postojati prvi izvod profila jezgra za svaku nenegativnu vrijednost x , osim u konačnom broju tačaka [30].

$$K\left(\frac{x-x_i}{h}\right) = c_k k\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right) \quad (3.9)$$

Označimo sada sa $g(x)$ negativni prvi izvod profila $k(x)$, tako da vrijedi $g(x) = -k'(x)$ i neka je za svako $i = 1, \dots, n$ definirano g_i zadano jednačinom $g_i = g(\|(x-x_i)/h\|^2)$. Relacija 3.8 se u tom slučaju može transformirati kako je to prikazano u 3.10 [30]. U navedenoj relaciji je lijevi član $2c_{k,d}/nh^{(d+2)}(\sum_{i=1}^n g_i)$ proporcionalan estimatoru gustoće jezgra G , dok desni član $\sum_{i=1}^n x_i g_i / \sum_{i=1}^n g_i - x$ predstavlja vektor srednjeg pomjeranja.

$$\begin{aligned} \nabla \hat{f}(x) &= \frac{2c_{k,d}}{nh^{(d+2)}} \sum_{i=1}^n (x-x_i) k'\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right) = \\ &= \frac{2c_{k,d}}{nh^{(d+2)}} \sum_{i=1}^n (x-x_i) g\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right) = \\ &= \frac{2c_{k,d}}{nh^{(d+2)}} \left(\sum_{i=1}^n g_i \right) \left(\frac{\sum_{i=1}^n x_i g_i}{\sum_{i=1}^n g_i} - x \right) \end{aligned} \quad (3.10)$$

Kada se u izraz koji odgovara vektoru srednjeg pomjeranja uvrsti i vrijednost g_i , dobiva se konačna relacija 3.11 za *mean-shift* vektor $m(x)$.

$$m(x) = \frac{\sum_{i=1}^n x_i g\left(\left|\left|\frac{(x-x_i)}{h}\right|\right|^2\right)}{\sum_{i=1}^n g_i} - x \quad (3.11)$$

Jednostavnim matematičkim operacijama se iz izraza 3.10, uzimajući u obzir značenja pojedinih članova relacije, može doći do drugačijeg oblika formulacije vektora srednjeg pomjeranja, što je pokazano kroz jednakost 3.12 [22], gdje je sa $\hat{f}_G(x)$ označen estimator gustoće izračunat sa jezgrom G, koji se definira na analogan način kao u relaciji 3.4, dok je sa c označena pozitivna konstanta.

$$m(x) = \frac{1}{2} h^2 c \frac{\nabla \hat{f}(x)}{\hat{f}_G/(x)} \quad (3.12)$$

Prethodno navedeni izraz pokazuje da je na poziciji x vektor srednjeg pomjeranja sa jezgrom G proporcionalan normaliziranoj estimaciji gradijenta gustoće proračunatoj korištenjem jezgra K, pri čemu je normalizacija izvršena estimacijom gustoće na poziciji x , dobivenom korištenjem jezgra G. Ovo dokazuje da vektor srednjeg pomjeranja uvijek upućuje na poziciju gdje je izračunat maksimalni porast gustoće [22], što u konačnici vodi do pronalaženja ciljne pozicije, tj. klastera u koji su se mapirale tačke značajki objekta. Zbog osobine da algoritam srednjeg pomjeranja vodi od nasumično odabrane pozicije do one u kojoj je najgušće mapiranje značajki, skup lokacija koje konvergiraju ka lokalnom maksimumu funkcije gustoće vjerovatnoće, tj. modu gustoće (engl. *density mode*) naziva se slivom privlačnosti (engl. *basin of attraction*) date funkcije [13]. Treba napomenuti da se tokom *mean-shift* procedure može desiti da konvergencija prestane u nekoj od stacionarnih tačaka koja nije lokalni maksimum. Zbog toga se vrši perturbacija stacionarnih tačaka za proizvoljan, mali vektor i procedura se ponavlja. Ukoliko proces ponovo konvergira na istu poziciju, stacionarna tačka je zaista lokalni maksimum i identificira sliv privlačnosti moda gustoće [30].

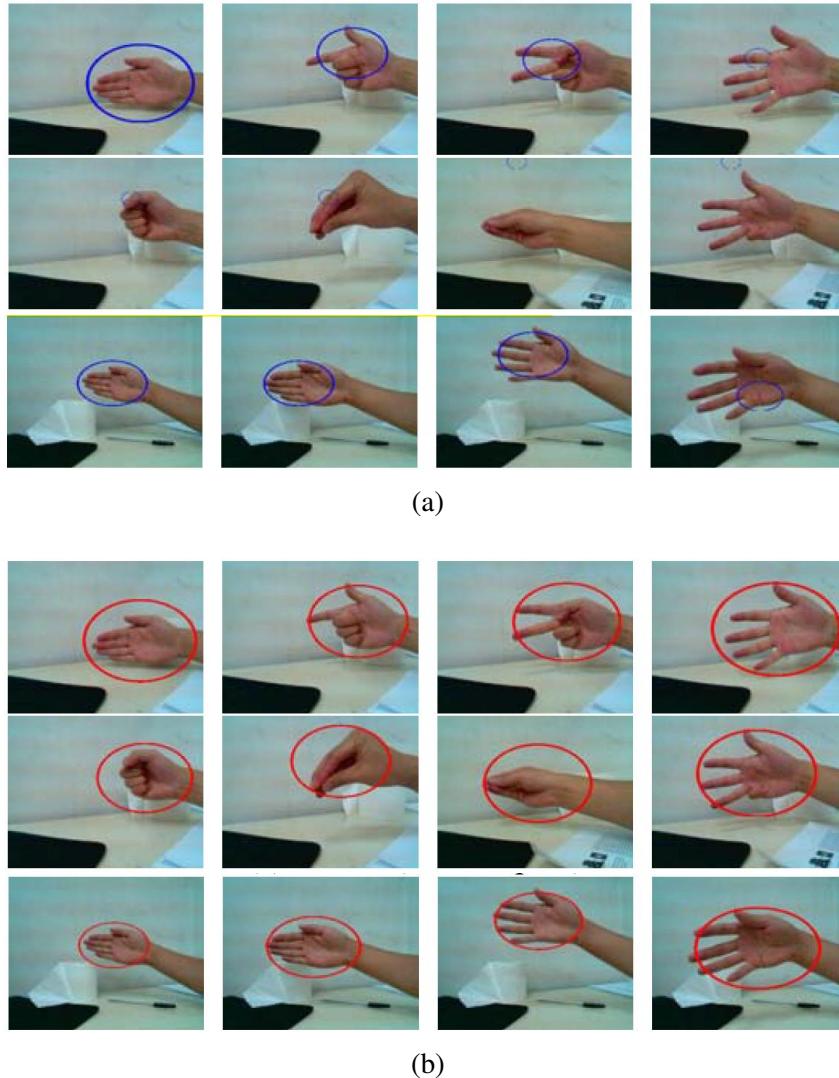
Jednostavnost *mean-shift* metode leži u tome da je potrebno odabrati samo jedan parametar, h , tj. veličinu jezgra. Međutim, ovo je ujedno i ograničenje, s obzirom na to da se odabirom prevelikog jezgra može uzrokovati povezivanje različitih slivova atrakcije, dok se korištenjem premalog jezgra može desiti da se jedan klaster podijeli na više njih [30], formirajući na taj način više slivova atrakcije od jednog.

Kako se radi o praktičnoj metodi, pogodnoj za rad u realnom vremenu, razvijeni su brojni algoritmi zasnovani na pomjeranju srednje vrijednosti. Njihova komparativna analiza bit će izvršena u narednom poglavlju.

3.3 Komparativna analiza metoda zasnovanih na pomjeranju srednje vrijednosti

Najznačajnija primjena algoritma pomjeranja srednje vrijednosti, KBOT (engl. *Kernel-Based Object Tracking*), prvi put je predstavljena u radu [13]. KBOT vrši prostorno maskiranje ciljnog objekta korištenjem izotropskog jezgra, čime se omogućava regularizacija histograma značajki kojim se predstavlja objekat i definira prostorno glatka funkcija sličnosti. Na taj način se postiže mogućnost jednostavne pretrage sličnih objekata korištenjem gradijenta, kako je opisano u prethodnom poglavlju. Dakle, problem lokalizacije objekta se svodi na pretragu sliva privlačnosti definirane funkcije.

Radi utvrđivanja sličnosti između ciljnog i potencijalno sličnih objekata, oba se prvobitno opisuju njihovim funkcijama gustoće vjerovatnoće (PDF) na osnovu kojih se grade odgovarajući histogrami. Treba napomenuti da ova metoda koristi histogram boje, ali se navodi mogućnost korištenja drugih značajki, kao što su tekstura ili oblik objekta koji se prati. Na osnovu dobivenih histograma se utvrđuje korelacija korištenjem metrike izvedene iz Bhattacharyya koeficijenta. Ovaj koeficijent jednak je kosinusu ugla između m -dimenzionalnih vektora PDF-ova



Slika 3.4: Usporedba rezultata primjene KBOT algoritma i njegovog unaprjeđenja [8]. (a) Detekcija ciljnog objekta KBOT algoritmom, objekat je označen plavom elipsom; (b) Detekcija ciljnog objekta unaprijeđenim KBOT algoritmom, objekat je označen crvenom elipsom.

razmatranih objekata, pa je cilj izvršiti njegovu maksimizaciju, jer je na taj način postignuta najveća sličnost između ciljnog objekta i kandidata. KBOT je pokazao dobre performanse pri kretanju kamere, djelimičnim okluzijama i promjeni veličine objekta koji se prati, međutim, uočen je i niz nedostataka koji je otvorio prostor za dalja poboljšanja.

Jedan od nedostataka leži u prevelikom broju iteracija po frejmu tokom pretrage ciljnog objekta. Naime, predefiniran je maksimalni broj iteracija i bez obzira na to kolika je razlika u korelaciji između ciljnog objekta i kandidata, pretraga se nastavlja dok se jedan frejm ne obradi navedeni broj puta. Zbog toga su autori rada [37] došli na ideju uvođenja vrijednosti praga razlike između prethodnog i trenutnog kandidata, za čiji proračun koriste metriku izvedenu iz Bhattacharyya koeficijenta. Na taj način se pretraga završava prije dostizanja maksimalnog broja iteracija, ukoliko je izračunata razlika manja od vrijednosti praga. Pored toga, koristi se kvadratni ili pravougaoni model za predstavljanje modela ciljnog objekta, dok inicijalni KBOT algoritam koristi kružni ili eliptički model. Obje izmjene uzrokovale su skraćenje vremena izvršenja algoritma.

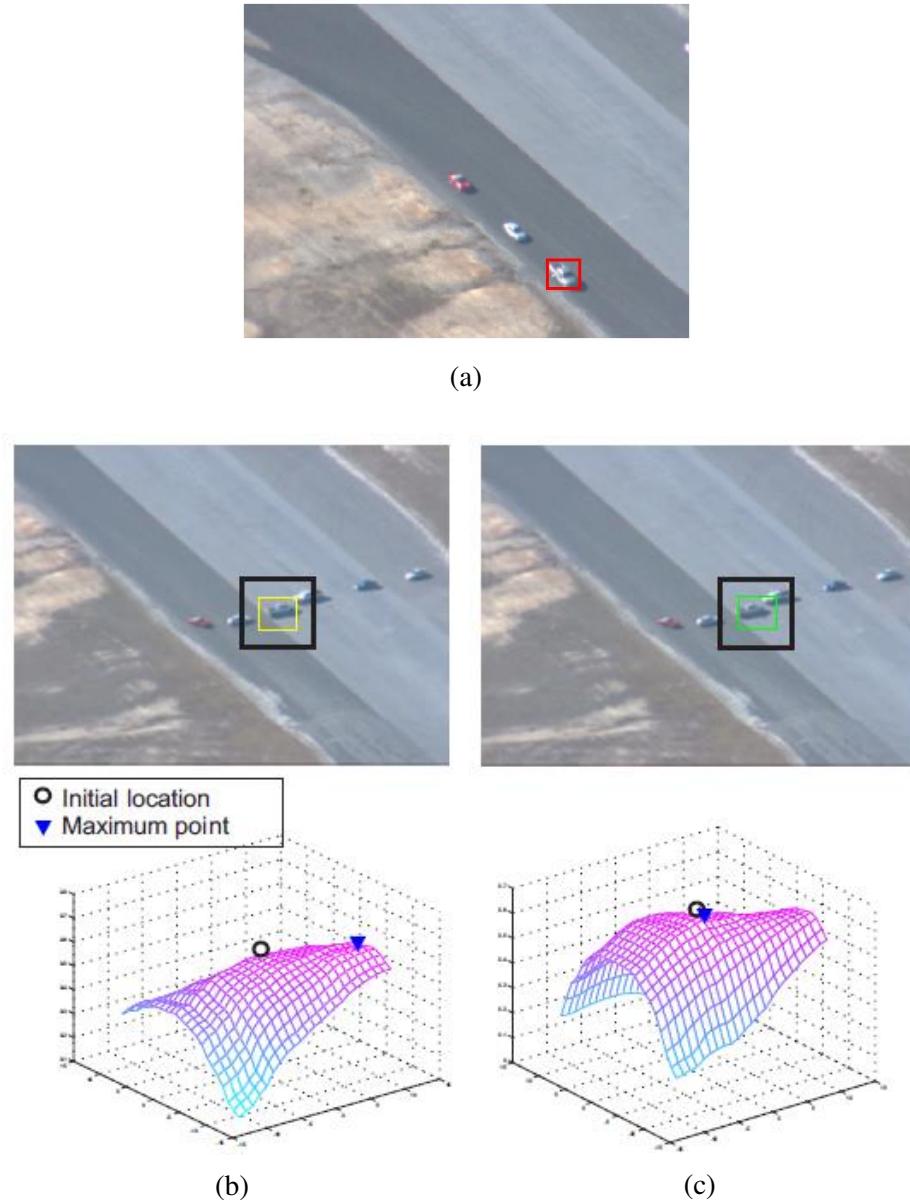


Slika 3.5: Primjer rezultata postignutog praćenjem objekta na video zapisu upotrebom asimetričnog jezgra [7].

Drugi uočeni nedostatak je nepouzdana adaptacija skaliranja i nepromjenjivost modela ciljnog objekta. Da bi se riješio ovaj problem, autori rada [8] su na postojeći algoritam implementirali mehanizam koji vrši predikciju centra objekta, nakon čega se vrši estimacija optimalnog skaliranja i podešavanje modela ciljnog objekta korištenjem strategije bazirane na estimaciji gustoće jezgra. Ovaj algoritam se pokazao robusnijim u odnosu na inicijalni KBOT, a pritom je zadržana mogućnost procesiranja u realnom vremenu. Usporedba rezultata koji se dobivaju primjenom spomenutih algoritama data je na slici 3.4. Plavim elipsama je označen pronađeni ciljni objekat upotrebom KBOT algoritma, dok je označavanje izvršeno crvenom bojom u slučaju korištenja dodatnog mehanizma koji uvodi otpornost na skaliranje i mogućnost adaptacije modela objekta čija se pretraga vrši. Sa navedene slike je uočljivo da KBOT algoritam gubi mogućnost pronalaženja ciljnog objekta ukoliko se njegov oblik znatno promijeni, te da se ne vrši prilagodba veličine oznake objekta ako dođe do potrebe za istom. Zahvaljujući adaptivnoj promjeni modela ciljnog objekta, algoritam predstavljen u radu [8] pokazuje otpornost na navedene probleme.

Kako je navedeno u okviru poglavlja 3.1.1, izotropska jezgra, koja se koriste u KBOT algoritmu i njegovim varijacijama, imaju nedostatak da u većini slučajeva ne prate dobro oblik objekta, što naročito predstavlja problem ukoliko dolazi do njegovog približavanja ili udaljavanja od kamere, čime se vizuelno mijenjaju njegove dimenzije. Kao prihvatljivo rješenje nekada se koriste anizotropska simetrična jezgra (eliptičnog oblika), jer se njihovom primjenom obično obuhvata samo mali dio podloge, koji nema veliki utjecaj na estimaciju pozicije objekta na narедnom frejmu. Međutim, na taj način se ne rješava inherentni problem izotropskih jezgara, već se on samo umanjuje. Kao prirodno rješenje se nameće upotreba asimetričnih jezgara koja u potpunosti prate oblik objekta. Korištenjem ovih jezgara u algoritmu pomjeranja srednje vrijednosti omogućava se adaptivna promjena veličine i orientacije jezgra u svakoj iteraciji. Navedeno se postiže proširenjem tradicionalnog *mean-shift* algoritma, tj. uključenjem skaliranja i orientacije kao dodatnih dimenzija prostora u kom se vrši analiza [7], a potom se simultano estimiraju sve nepoznate veličine u nekoliko iteracija algoritma. Dakle, navedena metoda podrazumijeva korištenje složenijeg jezgra, čiji se proračun vrši kroz više parametara. Podsjećanje radi, radikalno simetrična izotropska jezgra koja koristi inicijalni *mean-shift* algoritam zahtijevaju samo jedan parametar - veličinu jezgra. Ta činjenica automatski ukazuje na složenost pristupa korištenja asimetričnih jezgara za praćenje objekata na video zapisu, čime se produžava vrijeme izvršavanja algoritma. Ipak, robusnost je povećana na ovaj način, te se *mean-shift* metoda sa asimetričnim jezgom pokazala superiornom u odnosu na tradicionalni algoritam. Omogućena je bolja konzistentnost praćenja objekta kroz video, detekcija ciljnog objekta nije onemogućena ukoliko se njegova veličina ili orientacija promijene, te je poboljšana otpornost

detekcije objekta u slučaju kada se nalazi u okruženju koje je slične boje ili teksture kao i on sam. Primjer rezultata dobivenih primjenom opisanog algoritma dat je na slici 3.5.



Slika 3.6: Usporedba rezultata primjene tradicionalnog *mean-shift* algoritma i algoritma koji pored informacije o boji koristi i informacije o teksturi i obliku ciljnog objekta. [9]. (a) Prvi frejm na kom je ciljni objekat označen crvenim pravougaonikom; (b) Detekcija ciljnog objekta *mean-shift* algoritmom na 460. frejmu videa i površina sličnosti ciljnog i pronađenog objekta; (c) Detekcija ciljnog objekta algoritmom koji koristi informacije o boji, teksturi i obliku objekta na 460. frejmu videa i površina sličnosti ciljnog i pronađenog objekta.

Prethodno spomenuta rješenja koja kao osnovu imaju algoritam pomjeranja srednje vrijednosti oslanjaju se njegovu ideju praćenja objekata na osnovu boje. Povećanje vjerodostojnosti praćenja može se ostvariti upotrebom više od jedne značajke objekta. Ukoliko se uz informaciju o boji koriste i informacije o obliku i teksturi objekta, može se prevazići nemogućnost diferencijacije objekta od podloge u slučaju kada su oni slične boje. Autori rada [9] vrše ekstrakciju informacija o obliku i teksturi računanjem izvoda slike po x i y osi. Na osnovu dobivenih infor-

macija, formira se orientacioni histogram, nad kojim se primjenjuje Scharrova maska (jezgro). Ova maska omogućava pronalaženje ivica objekta, čime se postiže ekstrakcija informacija o njegovom obliku i teksturi. U računarskoj analizi slike uobičajeno se koristi Sobelova ili Canny metoda u ovu svrhu, međutim, pokazalo se da Scharrova maska daje vjerodostojnije rezultate [9]. Informacije o boji su u ovom algoritmu dobivene na sličan način kao i u prethodno navedenim. Na osnovu histograma boje i orientacionog histograma formira se združeni histogram kojim se predstavlja ciljni objekat. Nakon toga se vrši proračun vektora srednjeg pomjeranja, odnosno, izvršava se standardna *mean-shift* procedura. Zavisno od analiziranog video zapisa, algoritam je u nekim slučajevima pokazao bolje performanse u odnosu na KBOT, ali ipak nije u potpunosti riješio problem neprepoznavanja objekta u slučaju postojanja vrlo sličnog objekta ili podloge.

Za primjer prikazan na slici 3.6a, koja predstavlja prvi frejm analiziranog videa, iscrtane su površi sličnosti između ciljnog objekta i pronađenog kandidata na 460. frejmu videa. Na graficima površi je crnim kružnicama označena inicialna lokacija objekta, dok je plavim trokutom označena tačka u kojoj je postignuta najveća sličnost cilnjom objektu. Za dati primjer je uočljiva superiornost algoritma implementiranog u okviru rada [9] u odnosu na tradicionalni *mean-shift* algoritam.

Zbog uočenih poboljšanja pri ekstrakciji značajki teksture i oblika, u okviru ovog rada će se također evaluirati njihovo djelovanje u implementiranom algoritmu za praćenje objekata na video zapisu korištenjem *mean-shift* metode. Pored toga, radi smanjenja računarske kompleksnosti, koristit će se prag razlike između prethodnog i trenutnog kandidata za završetak obrade frejma. S druge strane, problemi skaliranja objekta ne postoje u industrijskom okruženju za koje se algoritam implementira, zbog čega se neće primjenjivati metode za njihovo rješavanje.

* * *

U okviru ovog poglavlja objašnjeno je prostorno filtriranje, s posebnim naglaskom na izotropske filtere. Uočene su prednosti i nedostaci u odnosu na druge vrste filtera, te je predstavljen princip rada algoritma pomjeranja srednje vrijednosti. U sklopu pregleda literature, datog u uvodnom dijelu, spomenut je širok spektar radova napisanih na temu korištenja *mean-shift* algoritma u svrhu praćenja objekata na video zapisu. Na ovom mjestu je izvršena komparativna analiza nekih od navedenih metoda, koje su najrelevantnije za ovaj rad. Svaka od njih ima svoje prednosti i nedostatke, koji se ogledaju u vremenu izvršavanja algoritma i njihovoj tačnosti. Zbog toga odabir algoritma zavisi od konkretne primjene i problema koji se javljaju na video zapisima od interesa.

Poglavlje 4

Korištenje Kalmanovog filtera za praćenje pokretnih objekata

Kalmanov filter se često koristi u aplikacijama za praćenje objekata na video zapisu, jer omogućava predikciju trajektorije na osnovu prethodnog kretanja objekta. Ovo je naročito pogodno u slučajevima kada dolazi do okluzije. Osim toga, ukoliko objekat ima veći pomak između susjednih okvira u odnosu na maksimalni predviđeni (praćenje objekta čije je kretanje vrlo brzo), ovaj filter može omogućiti poboljšanje performansi korištenog algoritma.

Zadatak ovog poglavlja je detaljan opis rada Kalmanovog filtera, te analiza fuzije *mean-shift* zasnovanih algoritama sa algoritmima koji koriste Kalmanovo filtriranje.

4.1 Kalmanov filter

Osnovne karakteristike Kalmanovog filtera su da se njegov matematički model može opisati u prostoru stanja, te da rješenje može biti izračunato rekurzivno. Pritom se pretpostavlja da je sistem linearan, da su njegove opservacije linearne funkcije stanja i da je šum (kako u sistemu, tako i u mjerjenjima) bijeli Gaussov šum [30]. U praksi, međutim, ne moraju biti strogo zadovoljeni navedeni uslovi da bi bio korišten Kalmanov filter.

Njegov matematski model opisan je relacijom 4.1, u kojoj je sa A_k označena matrica prelaza stanja, a sa w_k Gaussov bijeli šum, kovarijanse Q_k i nulte srednje vrijednosti. Ako se sa v_k označi druga varijabla bijelog Gaussovog šuma, kovarijanse R_k i srednje vrijednosti jednake nuli, te sa H_k matrica mjerjenja, model mjerjenja može biti predstavljen relacijom 4.2 [28].

$$x_{k+1} = A_k x_k + w_k \quad (4.1)$$

$$z_k = H_k x_k + v_k \quad (4.2)$$

Relacija 4.1 jasno ukazuje na rekurzivno djelovanje Kalmanovog filtera, pri čemu se u svakoj narednoj opservaciji očekuje bolja estimacija u odnosu na prethodno izračunatu. Stoga se Kalmanov filter sastoji iz dvije faze: faza predikcije i faza korekcije[15]. U prvoj fazi se na osnovu prethodnog predviđa naredno stanje, dok se u drugoj vrši proračun greške estimacije, radi njene korekcije u narednoj iteraciji. Greška se u odnosu na prethodnu opservaciju može izračunati na osnovu relacije 4.3, a u odnosu na narednu pomoću jednakosti 4.4. U navedenim relacijama znak " \wedge " ukazuje na to da se radi o estimaciji, " $-$ " da se odnosi na prethodnu

opservaciju, a "+" na narednu [30].

$$e_k^- = x_k - \hat{x}_k^- \quad (4.3)$$

$$e_k^+ = x_k - \hat{x}_k^+ \quad (4.4)$$

U idealnom slučaju, greška estimacije jednaka je nuli. Kako bi se ovo postiglo, u svakoj iteraciji se računa matrica Kalmanovih pojačanja (engl. *Kalman gain matrix*) K_k , koja omogućava ažuriranje estimacija u skladu sa relacijom 4.5 [30]. Član $z_k - K_k \hat{x}_k^-$ predstavlja rezidualnu grešku. Relacija ukazuje na to da se estimacija u narednoj opservaciji zasniva na estimaciji dobivenoj u prethodnoj iteraciji, pri čemu se ona koriguje kako bi se postigla ciljna nulta vrijednost greške procjene.

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \quad (4.5)$$

Neka su sa P_k^- i P_k^+ označene kovarijanse grešaka e_k^- i e_k^+ , respektivno i neka je R_k kovarijansa šuma v_k . Imajući u vidu prethodne relacije, činjenicu da je po definiciji kovarijansa greške jednaka srednjoj vrijednosti proizvoda greške i njene transponovane vrijednosti, te da su greške nezavisne i jednake nuli, može se izvesti relacija 4.6 [30], u kojoj je sa I označena jedinična matrica. Analogno se relacija može izvesti za kovarijansu P_k^- .

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (4.6)$$

Da bi se ostvarila nulta greška, potrebno je odabratи matricu Kalmanovih pojačanja tako da se izvrši minimizacija sume dijagonalnih elemenata matrice P_k^+ . Procedura se provodi na klasičan način pretrage ekstrema funkcije, traženjem prvog izvoda po K_k i njegovim izjednačavanjem sa nulom. Rješenje je prikazano relacijom 4.7, gdje je $P_k^- = A_k P_{k-1}^+ A_k^T + Q_{k-1}$. Kovarijansa P_k^+ se onda može iskazati relacijom 4.8.

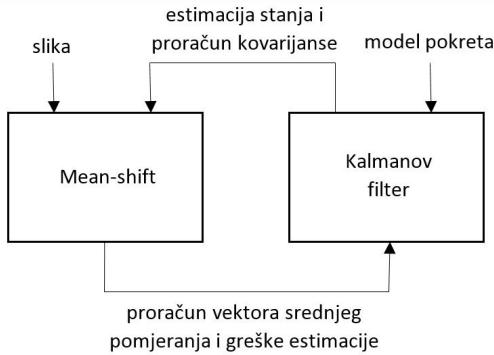
$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + E_k)^{-1} \quad (4.7)$$

$$P_k^+ = (I - K_k H_k) P_k^- \quad (4.8)$$

Uvrštavanjem prethodno navedenih relacija u jednakost 4.5, dobiva se optimalna estimacija za jedan korak postupka. Kroz njegovo iterativno ponavljanje dobivaju se sve bolje estimacije stanja. S obzirom na to, u algoritamskoj implementaciji Kalmanovog filtriranja, moguće je predefinirati prag maksimalne dopustive greške, kojim će se odrediti koliko iteracija će biti provedeno.

4.2 Prednosti i nedostaci fuzije Kalmanovog filtera sa metodom pomjeranja srednje vrijednosti

Analizom algoritama za praćenje objekata na video zapisu korištenjem Kalmanovog filtera i *mean-shift* procedure, u sklopu literature je izведен niz zaključaka o prednostima i nedostacima jedne metode u odnosu na drugu. S obzirom na to da se rad temelji na *mean-shift* zasnovanim metodama, prednosti Kalmanovog filtera su od naročitog značaja. Njihovim uočavanjem se mogu utvrditi slabosti postojećeg algoritma i izvesti zaključci o eventualnim poboljšanjima koja se mogu dobiti integracijom ovog tipa filtriranja u algoritam pomjeranja srednje vrijednosti. Uočeno je da Kalmanov filter dominira pri praćenju objekta čije kretanje nije podvrgnuto



Slika 4.1: Blok dijagram algoritma koji koristi fuziju *mean-shift* metode i Kalmanovog filtera [10].

isključivo translaciji, za šta dobre rezultate daje i sam *mean-shift* algoritam. Ipak, i u slučaju translacije je srednjekvadratna greška koja se postiže korištenjem Kalmanovog filtera mnogo manja u odnosu na onu dobivenu *mean-shift* metodom. Pored toga, on ima bolje performanse u slučaju analize video zapisa sa visokim nivoom šuma [39].

Fuzija Kalmanovog filtera sa metodom pomjeranja srednje vrijednosti prikazana je blok dijagramom 4.1. *Mean-shift* algoritam se koristi za proračun vektora srednjeg pomjeranja, na način kako je opisano u poglavlju 3, čime je omogućena brza lokalizacija objekta od interesa. Nakon toga, vrši se predikcija naredne lokacije objekta korištenjem Kalmanovog filtera. Za ovaj proces je neophodno obezbijediti mogućnost detekcije i lokalizacije objekta na inicijalnom frejmu, te periodičnu analizu objekta u cilju omogućavanja ažuriranja njegovog modela [10], koji se kroz vrijeme može mijenjati.

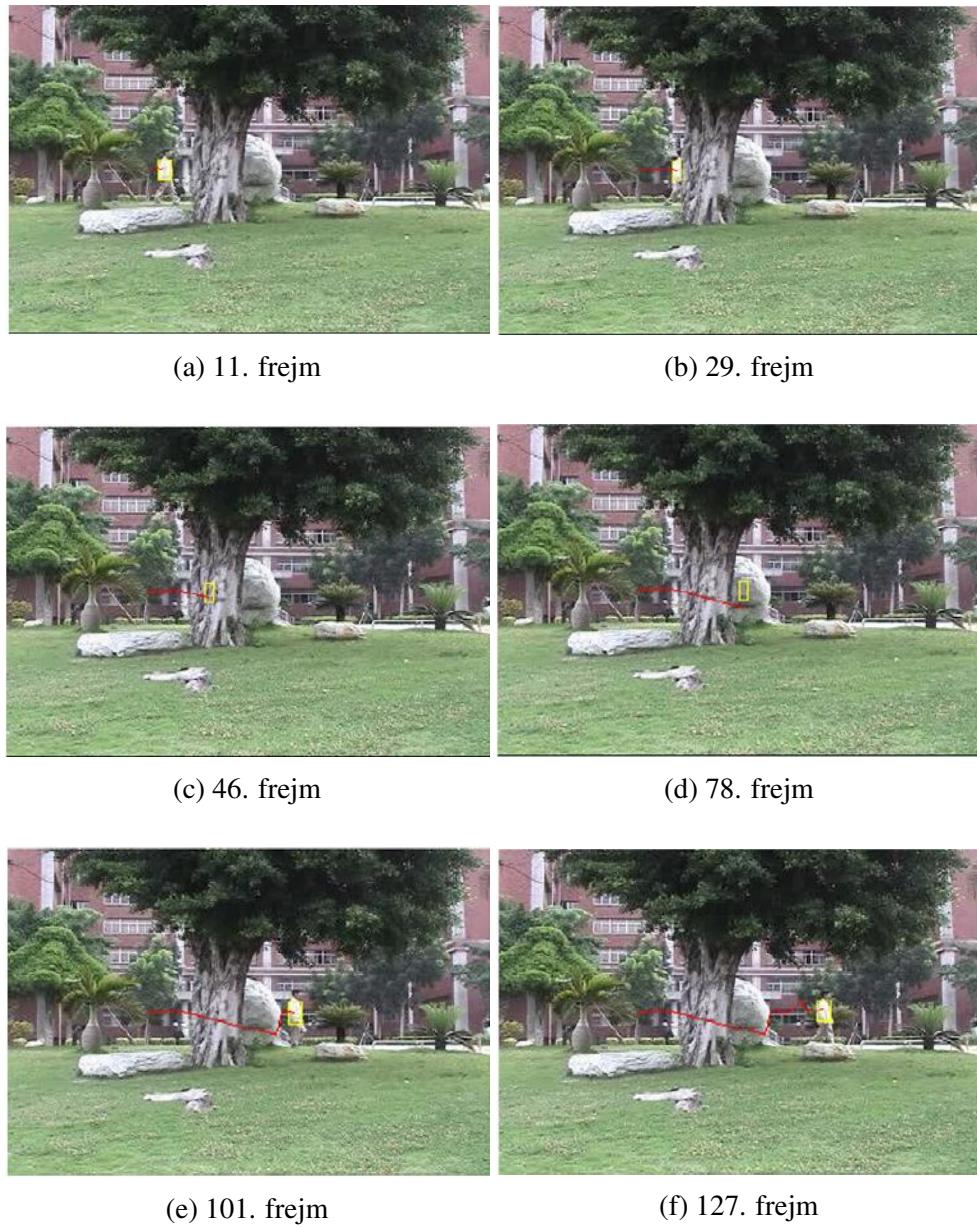
Prednost korištenja kombinacije dvije spomenute metode je u niskoj računarskoj kompleksnosti koja obezbeđuje rad u realnom vremenu, kao i u mogućnosti uspješnog praćenja objekata u slučaju potpune okluzije [10]. Primjer rezultata praćenja objekta u slučaju dugotrajne potpune okluzije dat je na slici 4.2. Sam algoritam pomjeranja srednje vrijednosti nije u mogućnosti izvršiti praćenje s tačnošću ekstrakcije trajektorije prikazane na slici. Umjesto toga, ispravno praćenje objekta bi bilo moguće samo onda kada je on barem djelimično vidljiv. Ako objekat pri kretanju ne mijenja naglo svoj smjer, Kalmanov filter dodatno potpomaže rezultate koje bi postigao sam *mean-shift* i osigurava bolje praćenje objekata u navedenim slučajevima.

Primjena Kalmanovog filtera kao dodatka na *mean-shift* algoritam može biti povoljna i u situacijama kada se prati objekat čija se boja ne razlikuje znatno od boje podloge po kojoj se on kreće [40]. Razlog tome je činjenica da se predikcijom koju vrši Kalmanov filter može smanjiti regija pretraživanja objekta od interesa, što dovodi do fokusiranja na željeni objekat i mogućnosti boljeg praćenja njegovog kretanja.

Nedostatak fuzije Kalmanovog filtera sa metodom pomjeranja srednje vrijednosti prvenstveno leži u inherentnim pretpostavkama Kalmanovog filtera. Pretpostavlja se da objekat koji se kreće prolazi kroz blage promjene, koje su modelirane bijelim Gaussovim šumom nulte srednje vrijednosti [10], što u praksi nije slučaj. Ukoliko su promjene kretanja objekta nagle, Kalmanov filter će zbog predikcije kretanja na osnovu prethodnog stanja objekta pogoršati rezultate u odnosu na one koje bi dao sam algoritam pomjeranja srednje vrijednosti. Pored toga, u većini aplikacija koje zahtijevaju rad u realnom vremenu nije jednostavno odrediti kriterij ili značajke za podjelu objekta na više regija za analizu koja se provodi fuzijom razmatrana dva algoritma [15].

Konačno, moguće je izvesti zaključak da fuzija *mean-shift* zasnovanih algoritama sa Kalmanovim filterom može dovesti kako do poboljšanja ekstrakcije trajektorije objekta koji se prati,

tako i do njenog pogoršanja, u zavisnosti od konkretnog slučaja koji se analizira. Ukoliko objekat nije podvrgnut okluziji i jasno je distinktivan u odnosu na podlogu, uloga Kalmanovog filtera u zajedničkom djelovanju sa *mean-shift* algoritmom nije velika. Način kretanja objekta i transformacije kroz koje on tokom vremena prolazi nisu unaprijed poznate, te se stoga ne može sa sigurnošću tvrditi da bi neka od metoda zasebno dala bolje rezultate ili da bi njihova fuzija imala najbolje performanse.



Slika 4.2: Primjer rezultata fuzije Kalmanovog filtera i *mean-shift* algoritma pri praćenju objekta koji prolazi kroz dugotrajnu potpunu okluziju [8].

* * *

U ovom poglavlju je opisan princip rada Kalmanovog filtera, te je izvršena analiza njegove fuzije sa algoritmom pomjeranja srednje vrijednosti u svrhu praćenja objekata na video zapisu.

Uočene su prednosti i nedostaci tog pristupa, koje su dovele do zaključka da poboljšanja ili pogoršanja koja uvodi kombinacija navedenih algoritama u odnosu na sam *mean-shift* zavise od karakteristika konkretnog video zapisa koji se analizira.

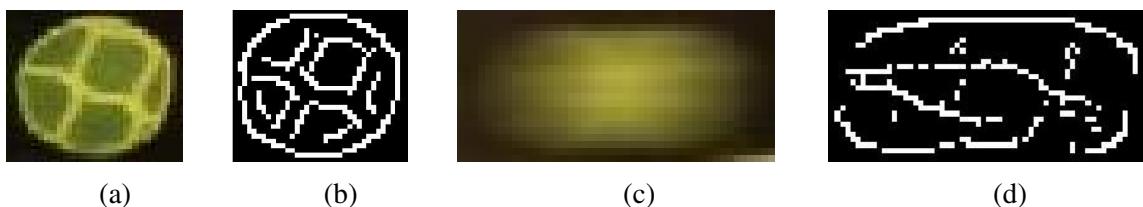
Poglavlje 5

Praktična realizacija algoritma za praćenje objekata na video zapisu

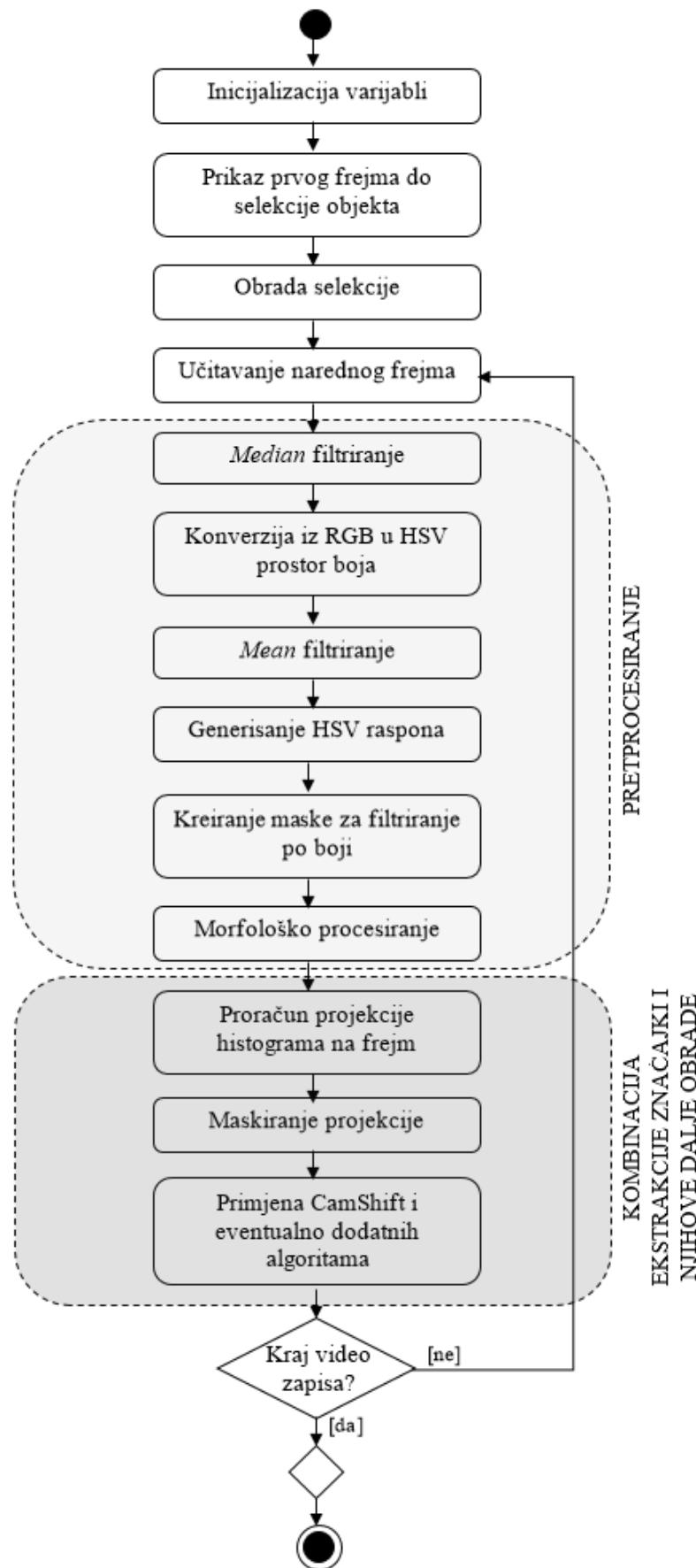
U okviru prethodnih poglavlja teorijski su objašnjeni postupci koje je potrebno izvršiti u cilju praćenja objekata na video zapisu korištenjem izotropskih jezgara i metode pomjeranja srednje vrijednosti. Ovo poglavlje ima za zadatak predstaviti način njihove implementacije u okviru OpenCV okruženja, korištenjem C++ programskog jezika.

Implementirani algoritam nudi mogućnost praćenja objekata na video zapisu korištenjem unaprijeđene CamShift metode, koja prati objekat bez obzira na broj boja koje on sadrži. U cilju utvrđivanja da li može doći do daljih poboljšanja ukoliko se izvrši fuzija sa drugim algoritmima, u programskom kodu se nudi mogućnost izbora uključenja ekstrakcije značajki teksture, SURF algoritma i Kalmanovog filtriranja. Ukoliko nijedna od dodatnih opcija nije uključena, izvršava se samo implementirana CamShift procedura. Treba napomenuti da se u toku formiranja algoritma testiralo i djelovanje ekstrakcije informacije o obliku objekta, međutim, nad analiziranim videima ono nije moglo uvesti nikakva poboljšanja. Razlog tome je obrada videa na kojima se objekti vrlo brzo kreću, što u toku vremena uzrokuje njihovo zamagljenje (engl. *blurring*) na frejmovima, te se informacija o obliku gubi. Konkretno, praćenjem objekta kružnog oblika se u slučaju stacionarnosti vrši ekstrakcija ivica na očekivan način, ali se pri ubrzanim kretanju kružnica koja ga opisuje postepeno razvlači u elipsu, čiji se krajevi pri ubrzavanju otvaraju i u konačnici se ekstraktuju linije koje prolaze kroz dvije tangentne tačke elipse. Primjer rezultata dobivenih primjenom Canny detekcije ivica prikazan je na slici 5.1.

Nakon pokretanja programskog koda, prikazuje se učitani video, na kom korisnik treba označiti objekat koji želi pratiti. Algoritam vrši dalju obradu frejmova, te u toku njihovog prikaza označava onaj dio frejma koji je prepoznao kao željeni objekat. Nakon završetka procedure nad svim analiziranim frejmovima, korisnik na uvid dobiva izdvojenu trajektoriju kretanja. Detaljan



Slika 5.1: Primjer rezultata ekstrakcije ivica radi dobivanja informacija o obliku objekta koji se prati. (a) Označeni objekat za praćenje. (b) Ekstrakcija ivica objekta u slučaju njegove stacionarnosti. (c) Prikaz objekta pri ubrzanim kretanjem. (d) Ekstrakcija ivica objekta kada se on ubrzano kreće.



Slika 5.2: Dijagram aktivnosti implementiranog algoritma za praćenje objekata na video zapisu.

opis postupka koji se izvršava dat je u okviru ovog poglavlja.

Mogu se razgraničiti dva nivoa obrade. Jedan nivo se odnosi na obradu slike objekta koji se prati (korisničke selekcije sa prvog frejma videa), a drugi na obradu svakog narednog frejma analiziranog video zapisa. Pritom se ova dva nivoa razlikuju za različite primijenjene metode. U oba dijela se provode postupci pretpresiranja, ekstrakcije značajki i obrade dobivenih informacija koje je potrebno izvršiti kako bi se moglo provesti *mean-shift* procedura i modifikacije koje su obuhvaćene ovim radom. Radi lakšeg shvatanja rada algoritma, priložen je dijagram aktivnosti 5.2 koji ugrubo prikazuje postupke koji se izvršavaju, dok će u nastavku poglavlja biti dati i dijagrami sa detaljnijim prikazom postupaka koji se vrše nad selekcijom objekta koji se želi pratiti, te nad frejmovima u slučaju da se odabere opcija uključenja dodatnih metoda.

5.1 Obrada selekcije objekta koji se prati

5.1.1 Pretpresiranje

Postupak pretpresiranja prvobitno se vrši nad označenim dijelom frejma koji predstavlja objekat koji se želi pratiti. Nakon toga, procedura se provodi nad cijelim frejmovima koji slijede u video zapisu, kako bi se omogućila lokalizacija objekta od interesa i ekstrakcija trajektorije njegovog kretanja. Koraci pretpresiranja zavise od značajke koju je cilj izdvojiti i razlikuju se u zavisnosti od toga koja se metoda primjenjuje uz pomjeranje srednje vrijednosti, kao što je prikazano na dijagramu 5.3.

Postupci koji se vrše, bez obzira na to da li je uključena neka dodatna metoda ili ne, su *median* filtriranje radi potiskivanja šuma i prelazak iz RGB u HSV model boja, jer on pruža bolju otpornost na promjene osvjetljenja u odnosu na RGB model i osnova je CamShift algoritma. U OpenCV okruženju je ovo moguće izvršiti primjenom funkcije *cvtColor*.

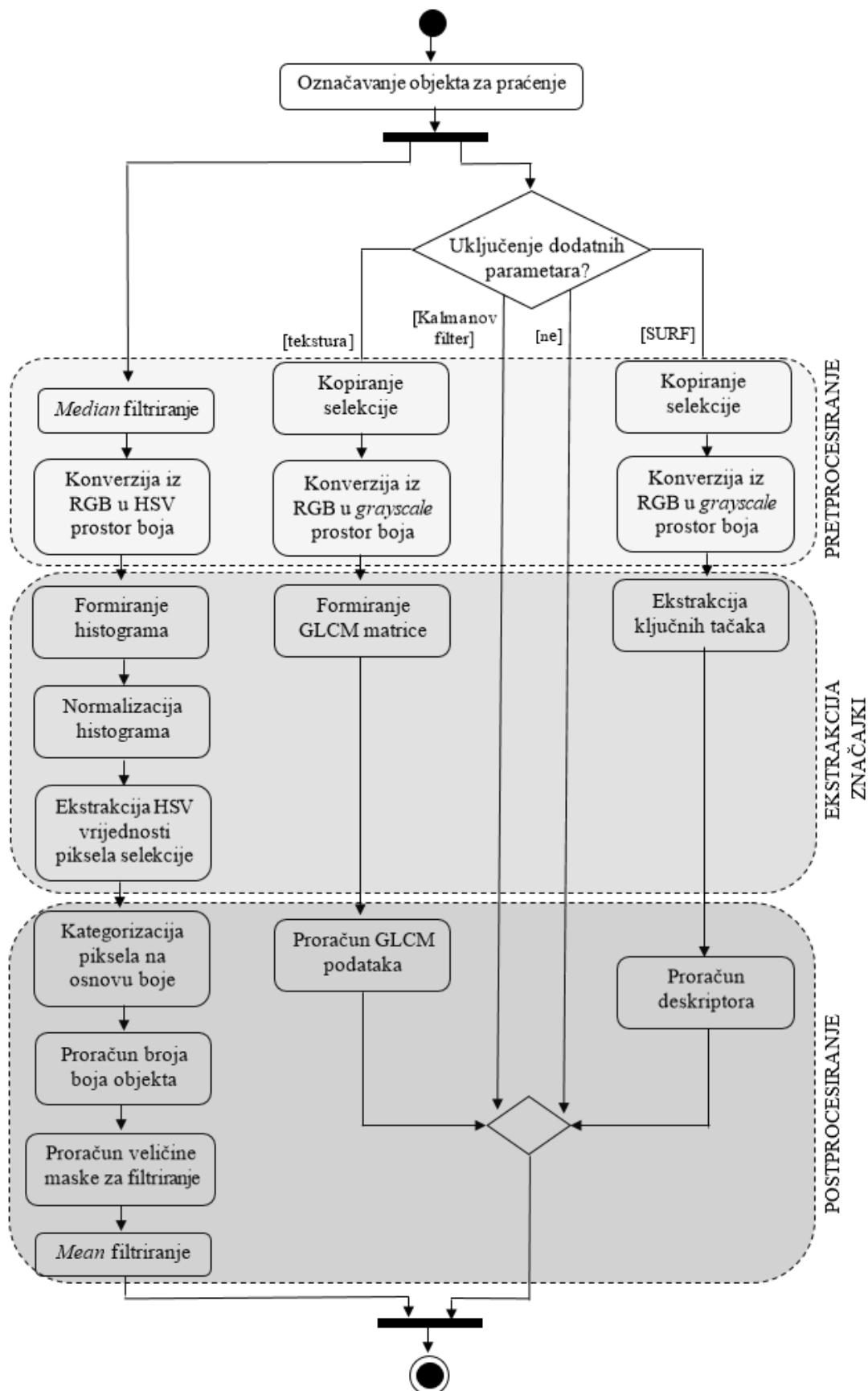
Ukoliko se primjenjuje i ekstrakcija teksture ili SURF algoritam, implementirano rješenje pravi kopiju selekcije, kako bi se nad njom moglo vršiti dodatne operacije koje zahtijevaju ove metode. U oba slučaja, prvi korak obrade kopije je prelazak iz RGB u *grayscale* model boja. Za Kalmanovo filtriranje nije potrebno vršiti dodatne operacije nad selekcijom objekta. Međutim, prije procesiranja frejmova podrazumijeva se inicijalizacija matrica čije korištenje zahtijeva Kalmanov filter.

5.1.2 Ekstrakcija značajki

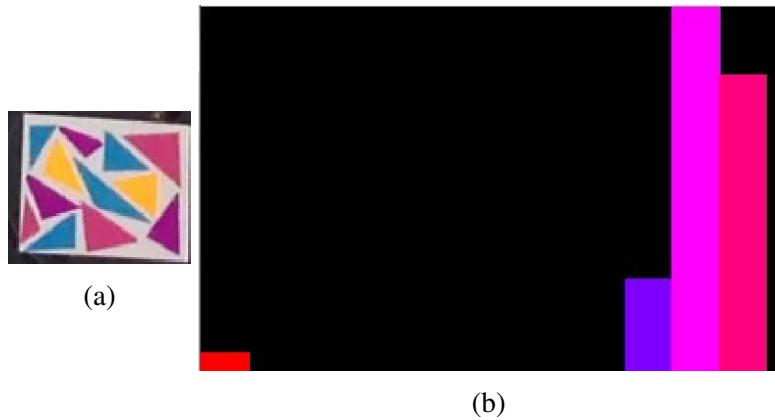
U okviru pretpresiranja, izvršeno je preslikavanje iz RGB u HSV model boja, a u ovom koraku se vrši formiranje histograma na osnovu pretpresirane selekcije objekta. Proračun histograma se u OpenCV C++ okruženju vrši korištenjem ugrađene funkcije *calcHist*, a nakon toga se on normalizira primjenom funkcije *normalize*. Primjer rezultata dat je na slici 5.4. Odabранo je da broj binova histograma bude jednak 12, jer se empirijskim putem pokazalo da se smanjenjem ili povećanjem ovog broja postižu lošiji rezultati.

U cilju izvršenja modifikacije tradicionalnog algoritma radi povećanja njegove robusnosti, potrebno je utvrditi da li objekat koji se prati sadrži samo jednu ili više boja. Zbog toga se ekstrahuju *hue* vrijednosti (tonovi) sastavnih piksela selekcije. Cilj je svaki piksel svrstati u jednu od konačnog broja skupina boja, radi prebrojavanja članova svake skupine. Ova informacija se u algoritmu koristi kako bi se na različit način tretirali jednobojni i višebojni objekti, odnosno, kako bi se parametri procesiranja adaptirali u zavisnosti od karakteristika objekta koji se prati.

Ukoliko se pri praćenju koristi i informacija o teksturi, u okviru ovog koraka se formira i GLCM matrica na osnovu vrijednosti piksela monohromatske kopije selekcije objekta.



Slika 5.3: Dijagram aktivnosti obrade korisničke selekcije objekta koji se želi pratiti.



Slika 5.4: Prikaz generiranog histograma boje označenog objekta za praćenje. (a) Označeni objekat. (b) Histogram boje.

Korištenje detektora i deskriptora značajki je olakšano u OpenCV okruženju, s obzirom na to da postoje ugrađene funkcije čijim se pozivanjem može relativno jednostavno doći do rezultata. Za SURF je potrebno formirati detektor pomoću funkcije *create*, kojoj se prosljeđuje minimalna vrijednost Hessijana. Ova vrijednost postavljena je na 700, a predstavlja prag na osnovu kog se vrši filtriranje potencijalnih ključnih tačaka u procesu njihove lokalizacije. Detekcija se potom vrši primjenom definiranog detektora, pozivom funkcije *detect* nad slikom označenog objekta za praćenje. Dobivene ključne tačke koristit će se u narednom koraku za računanje deskriptora.

5.1.3 Postprocesiranje

Kategorizacija piksela na osnovu boje vrši se provjerom pripadnosti njegove ekstraktovane vrijednosti nekoj od definiranih skupina. U tu svrhu je izvršena podjela ukupnog opsega koji definira OpenCV na 12 podopsega (jednak broj broju binova histograma), od kojih se svaki može pridružiti nekoj boji određene nijanse. Na taj način je izvršena svojevrsna kvantizacija boja sa 12 kvantnih nivoa i omogućeno je prebrojavanje piksela koji spadaju u svaku od generisanih skupina. Ukoliko su svi pikseli svrstani u istu skupinu, algoritam prepoznaće samo jednu boju i nastavlja proceduru sa različitim parametrima u odnosu na slučaj da je prepoznato više boja. Pored mogućnosti podjele opsega tonova na njih 12, algoritam omogućava promjenu postavki na kvantizaciju sa 6 nivoa, ali se u analiziranim video zapisima pokazalo da se postiže bolja detekcija boja korištenjem 12 nivoa.

Treba napomenuti da se u OpenCV okruženju vrši preslikavanje definiranog opsega tonova prema kom su oni predstavljeni na krugu, u opsegu od 0° do 360° na opseg od 0° do 180° . Prema tome, algoritam dodjeljuje po 15° svakoj skupini boja.

Uobičajeno je da se u sklopu pripreme slike za ekstrakciju značajki filtriranje po srednjoj vrijednosti koristi za potiskivanje šuma, kako je opisano u jednom od prethodnih poglavlja. Međutim, u slučaju implementiranog algoritma, ono je korišteno kako bi se poboljšale performanse standardnog CamShift-a onda kada se prati objekat sastavljen iz više boja. Kako nivo zamagljenja, tj. *blurring*-a objekta primjenom *mean* filtera zavisi od veličine jezgra kojim se filtriranje vrši, algoritam adaptira veličinu u skladu sa dimenzijama selekcije, prema relaciji 5.1. Sa w je označena širina selekcije izražena u pikselima, dok je h njena visina. Zamagljenjem objekta postiže se svojevrsno miješanje boja obuhvaćenih analiziranim dijelom slike. Ideja algoritma je da se na taj način naglasi razlika karakteristika objekta od interesa u odnosu na ostale, koje

pripadaju drugim objektima.

$$l = \frac{w \times h}{50} \quad (5.1)$$

Pored informacije o broju boja označenog objekta, njegovom obradom je potrebno doći i do saznanja o prosječnoj vrijednosti za svaku od HSV komponenata. To je u slučaju objekata koji sadrže više boja izvršeno jednostavnim sumiranjem odgovarajućih vrijednosti svih pripadajućih piksela zamagljene selekcije i normalizacijom sume njenim dimenzijama. Za jednobojsne objekte, izvršeno je izdvajanje okoline od po 5 piksela u vertikalnom i horizontalnom pravcu u odnosu na centralni, te je nad njima proveden isti postupak. Razlog uzimanja male okoline oko centralnog piksela je taj da se izbjegnu udaljenije tačke koje mogu biti podvrgnute efektu stapanja s podlogom. Procedura pronalaženja prosječne vrijednosti HSV komponenata se vrši kako bi se u samom algoritmu koji koristi *mean-shift* mogli postaviti pragovi za ton, nijansu i zasićenje, prema kojima će se vršiti prepoznavanje objekta.

Dalje procesiranje u slučaju obuhvatanja informacija o teksturi podrazumijeva proračun GLCM parametara (energija, entropija, homogenost i kontrast), dok se na osnovu detektovanih ključnih tačaka računaju deskriptori ukoliko je uključen SURF algoritam. U ovu svrhu se koristi ugrađena OpenCV funkcija *compute*, a rezultat koji ona daje će se koristiti za usporedbu sa deskriptorima detektovanog objekta na narednim frejmovima, što je standardni princip rada SURF algoritma.

5.2 Obrada frejmova pri pretrazi objekta koji se prati

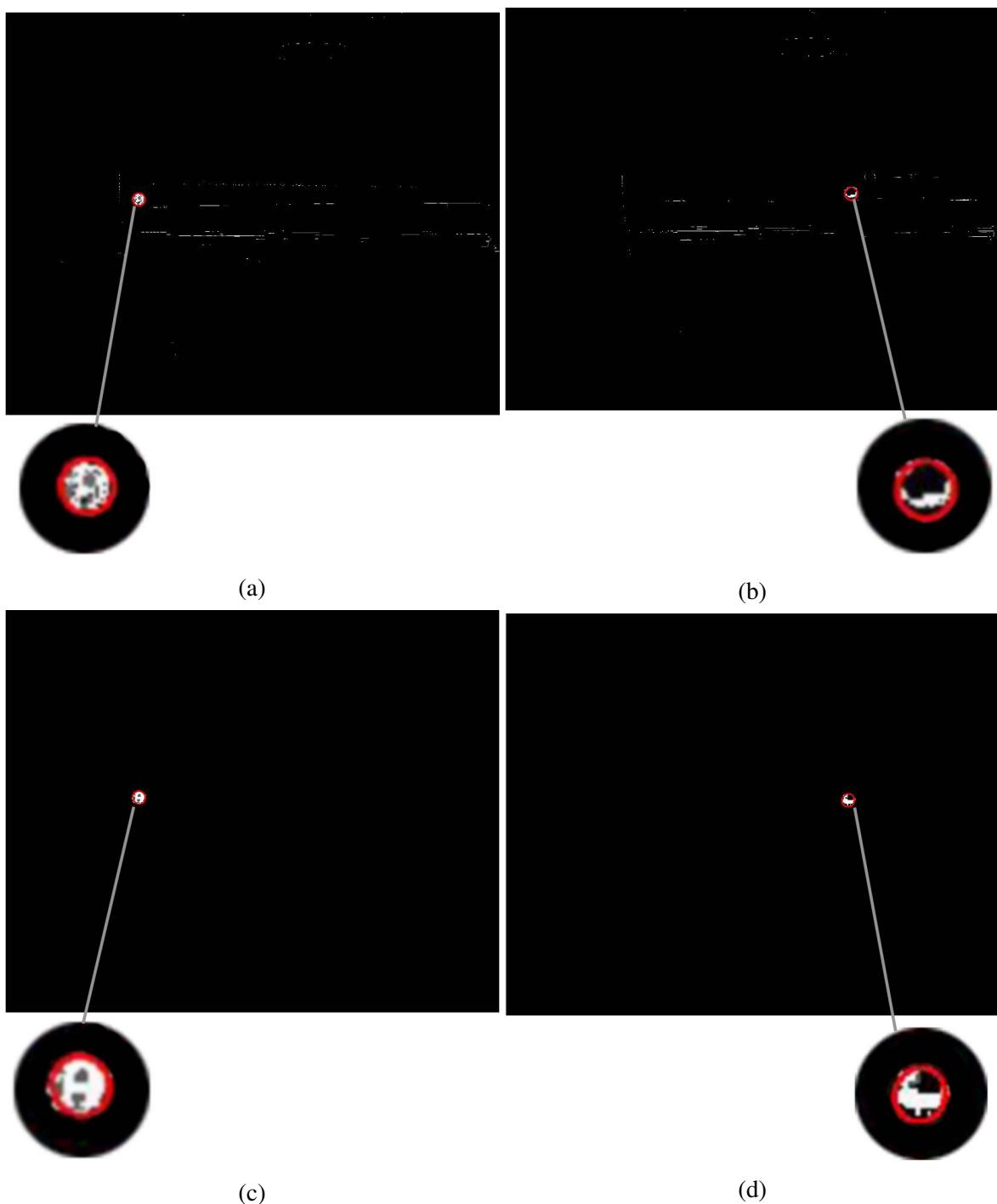
5.2.1 Preprocesiranje

Kao i u slučaju preprocesiranja selekcije objekta, svaki frejm se prvobitno filtrira po medijani radi potiskivanja šuma, a potom se primjenjuje konverzija iz RGB u HSV model boja. Nakon toga se vrši filtriranje po srednjoj vrijednosti, za koje se koristi prethodno izračunata veličina jezgra za zamagljenje.

Prije primjene metode pomjeranja srednje vrijednosti, poželjno je još dodatno obraditi dobivene informacije u cilju ostvarenja što boljih rezultata. Prije svega, ovo se odnosi na vrijednosti svake od HSV komponenata. Izdvajanje objekta od podloge i ostalih elemenata prikazanih na frejmu vrši se upravo na osnovu ovih vrijednosti, pa je potrebno postaviti pragove za iste. Najosjetljivija je vrijednost tona, zbog čega je za nju postavljena mala dozvoljena devijacija u odnosu na prosječnu vrijednost. Za slučaj praćenja jednobojsnih objekata, pokazalo se optimalnim koristiti maksimalnu devijaciju $\delta_H^{OCT} = 5$, a za objekte sa više boja $\delta_H^{MCT} = 10$ stepeni. Empirijskim putem došlo se i do pogodnih vrijednosti dozvoljene devijacije zasićenja i nijanse. Za jednobojsne objekte su te vrijednosti jednake $\delta_S^{OCT} = 40$ i $\delta_V^{OCT} = 50$, respektivno, dok su u suprotnom jednake $\delta_S^{MCT} = 30$ i $\delta_V^{MCT} = 40$. Na ovaj način je izvršena automatska adaptacija spomenutog raspona, koji u većini ostalih algoritama zahtijeva manuelno podešavanje radi mogućnosti praćenja.

Na osnovu prethodno navedenih vrijednosti, kreira se maska (jezgro) po principu izdvajanja samo onih piksela koji zadovoljavaju uslov iskazan relacijom 5.2, gdje su sa v_H , v_S i v_V označeni ton, zasićenje i nijansa piksela, respektivno, dok m_H , m_S i m_V označavaju izračunate srednje vrijednosti.

$$\begin{aligned} v_H &\in [m_H - \delta_H, m_H + \delta_H] \\ v_S &\in [m_S - \delta_S, m_S + \delta_S] \\ v_V &\in [m_V - \delta_V, m_V + \delta_V] \end{aligned} \quad (5.2)$$



Slika 5.5: Usporedba rezultata dobivenih sa i bez primjene morfoloških operacija. Prikazana je projekcija histograma boje na analizirani frejm za slučajeve: (a) praćenje statičnog objekta bez primjene morfoloških operacija; (b) praćenje objekta koji se brzo kreće, pri promjeni osvjetljenja na video zapisu, bez primjene morfoloških operacija; (c) praćenje statičnog objekta uz primjenu kombinacije erozije i dilatacije; (d) praćenje objekta koji se brzo kreće, pri promjeni osvjetljenja na video zapisu, uz primjenu kombinacije erozije i dilatacije.

Radi naglašavanja oblika objekta koji se prati, korištena je kombinacija morfoloških operacija erozije i dilatacije. U tu svrhu je odabran strukturalni element kružnog oblika, pri čemu je njegov radijus jednak 2 piksela za primjenu erozije i 4 piksela ukoliko se primjenjuje dilatacija. Prvobitnim provođenjem procesa erozije omogućava se odstranjivanje ili potiskivanje elemenata sličnih objektu koji se želi pratiti, dok se nakon toga dilatacijom omogućava naglašavanje preostalih elemenata slike. Na ovaj način objekat od interesa dolazi više do izražaja.

Ukoliko se posmatra pozadinska projekcija (engl. *backprojection*) histograma boje na analizirani frejm, jednostavno je uočiti razlike između slučaja kada se ne primjenjuju morfološke operacije u odnosu na slučaj kada su one primijenjene na prethodno opisani način. Treba napomenuti da projekcija ne spada u dio pretprecesiranja i bit će opisana u jednom od narednih potpoglavlja, ali je na ovom mjestu data kako bi se uočilo djelovanje ovog koraka u konačnici. Na slici 5.5 je prikazan primjer usporedbe rezultata (ne)korištenja morfoloških operacija na jednom od analiziranih video zapisa. Kada je objekat od interesa statičan, na samom početku videa, bez obzira na primjenu dodatnih operacija, njegov oblik je jasno uočljiv. Međutim, usporedbom slika 5.5a i 5.5c je jednostavno utvrditi da bez upotrebe morfoloških operacija algoritam pored objekta koji se prati detektuje i druge elemente sličnih karakteristika. Zahvaljujući eroziji, to na slici 5.5c nije slučaj. Kada se objekat brzo kreće ili se mijenja osvjetljenje u toku video zapisa, gubi se informacija o njegovom obliku. Dilatacijom se obezbjeđuje djelimično vraćanje informacije, kao što je prikazano na slici 5.5d. Usporedbom sa rezultatom dobivenim bez primjene morfoloških operacija 5.5b, vidljivo je poboljšanje u predstavljanju stvarnog oblika objekta.

Kada se u algoritam uključi i ekstrakcija informacija o teksturi, obuhvaćena je dodatna operacija koja omogućava brže provođenje postupka kreiranja i obrade GLCM matrice. Naime, zbog njenih velikih dimenzija uslijed broja nivoa sivih tonova, tj. broja kvantizacionih nivoa, vrijeme potrebno za izvršavanje algoritma se znatno produžava. U konkretnom slučaju, nad skupom analiziranih video zapisa u datom okruženju izvršenje algoritma uopće nije bilo moguće bez prethodnog smanjenja dimenzija frejmova videa, jer memorijski resursi računara nisu bili dovoljni za provođenje potrebne procedure. Kopiranje frejmova i smanjenje dimenzija kopije faktorom 5, primjenom funkcije *resize* izvršeno je isključivo za potrebe ekstrakcije tekture, dok su za ostale operacije korišteni frejmovi originalnih dimenzija.

5.3 Ekstrakcija i obrada značajki

Rezultat projekcije histograma, dobivene pozivom funkcije *calcBackProject* i maskirane odgovarajućim jezgrom, prikazan je na slici 5.7b. Standardni *mean-shift* i CamShift algoritmi ne omogućavaju praćenje ukoliko objekat ima više boja, ili je praćenje jako degradirano. Na slici 5.7c se uočava da se primjenom tradicionalnog algoritma ne pronalazi niti jedan sličan objekat, a crvena linija u gornjem lijevom uglu ukazuje na to da je praćenje onemogućeno (oznaka objekta se vraća u koordinatni početak ukoliko on nije detektovan na frejmu). S druge strane, ako se označi vrlo mali dio slike koji sadrži samo jednu boju, potvrđuje se ispravnost rada implementiranog algoritma. Na slikama 5.7d i 5.7e je prikazana projekcija u slučajevima označavanja dijela markera koji sadrži žutu i plavu boju, respektivno. Može se uočiti da su pored detekcije dijelova markera odabrane boje ujedno detektovani i pozadinski markeri odgovarajućih boja.

Kao prozor praćenja se inicijalno uzima selekcija koju korisnik napravi pri odabiru ciljnog objekta, a potom se naredni prozori računaju na temelju projekcije histograma i kriterija pretrage kandidata, pozivom ugrađenog CamShift algoritma. Kriterij je podešen tako da je predefinirani maksimalni broj iteracija za pretragu objekta unutar jednog frejma postavljen na 20, dok je najveća dozvoljena razlika između prethodnog i novog kandidata podešena na 0,03. Ukoliko se ispunji bilo koji od postavljenih uslova, obrada frejma se završava i učitava se novi,

nad kojim se ponavlja cijela procedura, počevši od pretpresiranja.

Ekstrakcija informacija o teksturi vrši se po principu opisanom u okviru poglavlja 2.2.3, ali ne na nivou čitavog frejma, već na nivou jednog njegovog dijela. Nakon početnog učitavanja, izvršava se pretvorba frejma u njegov monohromatski ekvivalent i izdvaja dio za koji će se računati GLCM vrijednosti. Pod tim dijelom se u prvoj iteraciji podrazumijeva korisnička selekcija, a u svakoj narednoj niz prozora dobivenih kao rezultat pomjeranja inicijalnog prozora izračunatog CamShift algoritmom. Prozor se pomjera u 8 različitim pravaca na udaljenosti do maksimalno 4 piksela u odnosu na početnu poziciju. Da bi se smanjilo vrijeme izvršavanja, korišten je korak od po 2 piksela, što znači da se za svaki pravac računa GLCM matrica prozora praćenja pomjerenog za 2 i 4 piksela. Pravci se nalaze pod uglovima od 0° , 45° , 90° , 135° , 180° , 225° , 270° , i 315° , što odgovara horizontalnom, vertikalnom i dijagonalnom pomjeranju početnog prozora.

Dalja analiza izdvojenog dijela frejma podrazumijeva uvid u vrijednosti svih piksela i prebrojavanje slučajeva u kojima je zadovoljen uslov da piksel na poziciji (x, y) ima nivo sive boje jednak i , a da je pritom taj nivo za neki od piksela iz njegove okoline, udaljen za $(\Delta x, \Delta y)$ jednak j . Na taj način se formiraju koeficijenti $c_{i,j}$, u skladu sa relacijom 2.1, a potom se vrši njihova normalizacija. Na osnovu dobijenih koeficijenata, pronalaze se srednja vrijednost, standardna devijacija i varijansa za analizirani frejm, te se ove vrijednosti koriste za dalje proračune.

Usporedbom parametara (homogenost, entropija, energija, kontrast) dobivenih na osnovu GLCM matrica početne selekcije i prozora pretrage objekta, utvrđuje se da li se pomjeranjem prozora u odnosu na inicijalnu poziciju ostvaruje veća sličnost tekture traženog objekta u odnosu na onu koja je postignuta algoritmom koji za praćenje koristi samo informaciju o boji. Ovako definirana pretraga podudarnosti po teksturi se oslanja na mogućnost CamShift algoritma da izvrši detekciju barem približne pozicije objekta, a nakon toga radi njeno finije podešavanje, utvrđivanjem GLCM sličnosti.

Ispitivanjem je utvrđeno da se proračunom razlike samo jednog od izračunatih GLCM parametara za selekciju i (pomjereni) prozor praćenja ostvaruju približno isti rezultati, bez obzira na odabrani parametar. To jasno ukazuje da nijedan od njih nije od dominantnog značaja u odnosu na druge, pa je za pretragu najveće podudarnosti po teksturi izvršeno pronalaženje minimuma sume apsolutnih razlika svih parametara za regije od interesa. Ovo je urađeno u skladu sa relacijom 5.3, u kojoj su oznakama E , H , K i U označeni entropija, homogenost, kontrast i uniformnost (energija), respektivno. Indeks f ukazuje na to da se radi o vrijednostima dobivenim nad prozorom na nekom od narednih frejmova, dok indeks s odgovara početnoj selekciji objekta.

$$T = \min\{|E_f - E_s| + |H_f - H_s| + |K_f - K_s| + |U_f - U_s|\} \quad (5.3)$$

Kada se u algoritam uključi ekstrakcija značajki tekture, korisniku se tokom reprodukcije videa istovremeno prikazuju rezultati praćenja postignuti samim CamShift algoritmom (što je označeno crvenom elipsom) i njegovom kombinacijom sa informacijama o teksturi (označeno zelenim pravougaonikom). Primjer oznaka je prikazan na slici 5.8a.

Slično kao što je urađeno za teksturu, pri uključenju praćenja objekata SURF algoritmom, oslanja se na relativno ispravnu lokalizaciju objekta primjenom CamShift-a. Prozor pretrage se u odnosu na izračunati povećava za maksimalno ± 30 piksela i unutar njega se traži objekat. Ukoliko bi se proširenjem prozora u bilo kom smjeru za navedeni broj piksela izašlo izvan granica frejma, vrši se adaptacija proširenja i u tom smjeru podešava na maksimalni mogući broj preostalih piksela koji pripadaju slici. Dalja procedura provodi se u skladu sa teorijskim objašnjenjem datim u okviru poglavlja 2, odnosno, vrši se ista procedura kao nad selekcijom objekta, a potom se uparuju deskriptorski vektori korištenjem FLANN uparivača (engl. *(Fast*

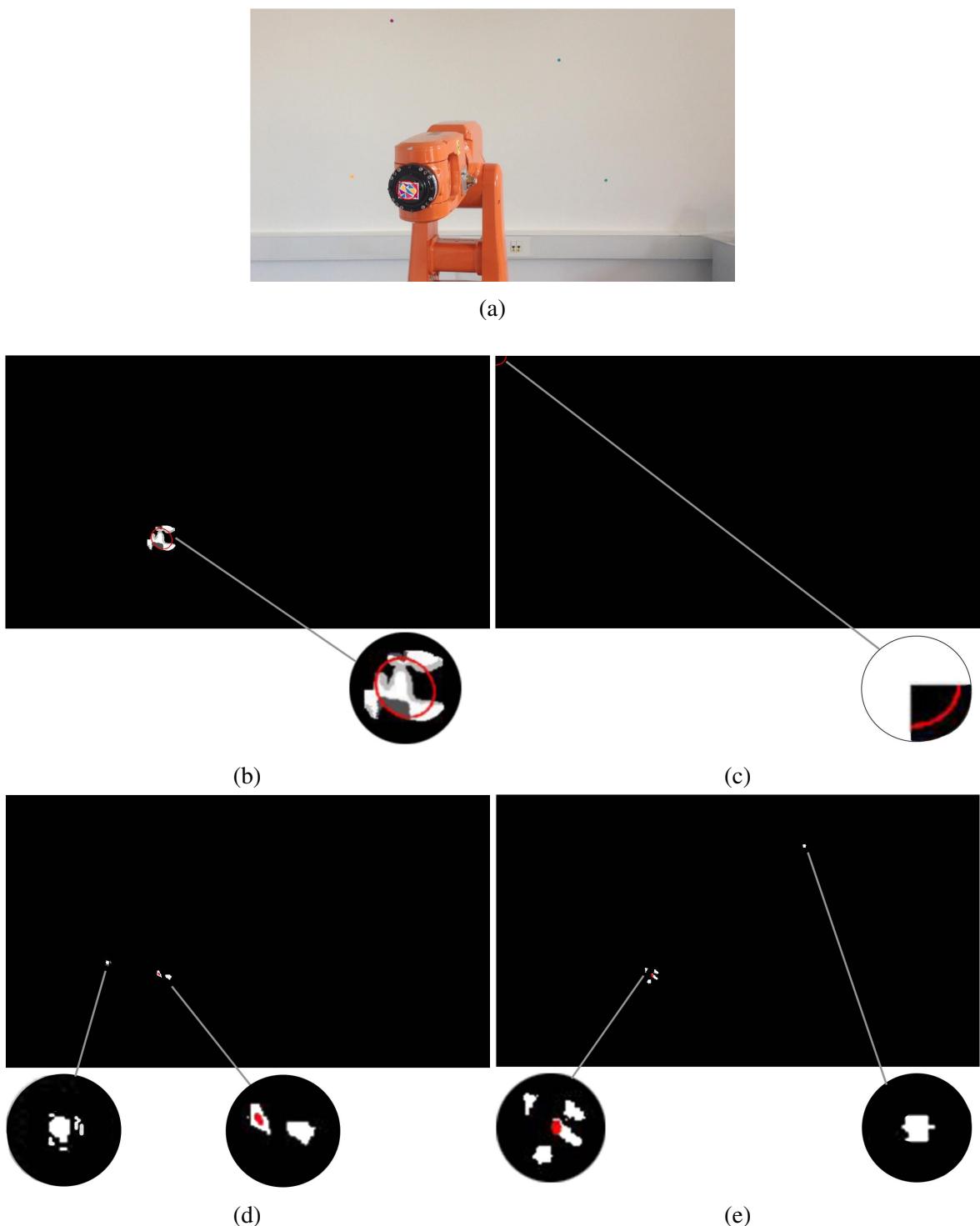


(a)

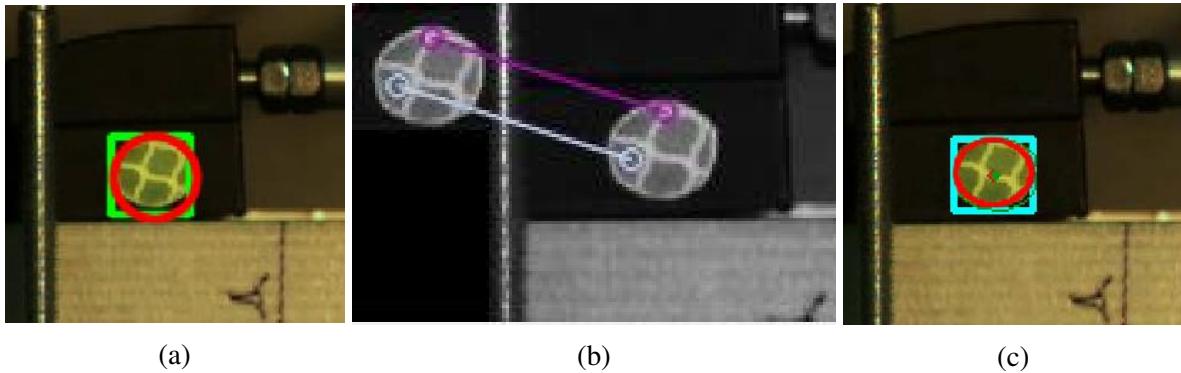


(b)

Slika 5.6: Primjer rezultata filtriranja po srednjoj vrijednosti u cilju detekcije objekta koji sadrži više od jedne boje korištenjem CamShift algoritma. (a) Frejm sa selektovanim višebojnim objektom označenim crvenom elipsom. (b) Zamagljenje frejma na osnovu dimenzija označenog objekta za praćenje.



Slika 5.7: Usporedba praćenja višebojnog objekta korištenjem tradicionalnog CamShift algoritma i njegove modifikacije. (a) Originalni frejm sa označenim objektom za praćenje. (b) Detekcija objekta uz pomoć prethodnog filtriranja po srednjoj vrijednosti. (c) Nemogućnost detekcije višebojnog objekta bez filtriranja. (d) Detekcija žutih objekata korištenjem modificiranog CamShift algoritma. (e) Detekcija plavih objekata korištenjem modificiranog CamShift algoritma.



Slika 5.8: Prikaz praćenja objekta na video zapisu uključenjem dodatnih metoda u CamShift algoritam. (a) Uvećani prikaz oznaka detektovanog objekta CamShift algoritmom (crvena elipsa) i njegovom kombinacijom sa ekstrakcijom značajki teksture (zeleni pravougaonik). (b) Detektovane i uparene karakteristične tačke primjenom SURF algoritma nad prozorom pretrage određenim CamShift algoritmom. (c) Uvećani prikaz oznaka detektovanog objekta CamShift algoritmom (crvena elipsa) i njegovom fuzijom s Kalmanovim filterom (tirkizni pravougaonik).

Library for Approximate Nearest Neighbors matcher) pomoću funkcije *match*. Da bi se utvrdilo koja su uparivanja dobra, a koja ne, računaju se minimalna i maksimalna udaljenost između uparenih ključnih tačaka, te se na slici povezuju samo one čija je udaljenost manja od predefiniranog praga. Taj prag je u konkretnom slučaju postavljen na trostruku vrijednost minimalne udaljenosti. Lokalizacija objekta od interesa vrši se na osnovu povezanih ključnih tačaka, a sam objekat se na frejmu označava zelenim pravougaonikom, kako je to prikazano na slici 2.13 u okviru poglavlja 2.

Slika 5.8b prikazuje način na koji se u toku reprodukcije videa korisniku daje povratna informacija o radu algoritma. S lijeve strane slike nalazi se objekat koji se prati (početna selekcija), a na desnoj strani je prikazan dio frejma koji se trenutno obrađuje. Uparene karakteristične tačke označene su istim bojama i povezane odgovarajućim linijama, radi lakšeg uočavanja. Kada se SURF algoritmom detektuje objekat, on se označava zelenim pravougaonikom, što ovdje nije slučaj. Dakle, može se zaključiti da u razmatranom primjeru SURF algoritam nije uspio izvršiti praćenje objekta, bez obzira na smanjenje prostora pretrage koji je omogućio CamShift. Zbog toga, umjesto oslanjanja na detekciju samog SURF algoritma, pri integraciji sa metodom srednjeg pomjeranja iskorištena su njegova pozitivna svojstva da bi se detekcija provela na nešto drugačiji način. Pozicije uparenih karakterističnih tačaka su uzete u obzir pri proračunu najvjerojatnije pozicije objekta u skladu sa relacijom 5.4, u kojoj je pretpostavljeno postojanje n uparenih tačaka. Sa $x_{k,i}$ i $y_{k,i}$ su označene pozicije i -te ($i = 1, \dots, n$) karakteristične tačke na frejmu po x i y osi. Analogno, oznake x_p i y_p su korištene za definiranje pozicije na originalnom video zapisu. Veličina w predstavlja širinu prozora na koji je svedena pretraga, dok je h visina istog.

$$\begin{aligned} x_e &= \frac{1}{n} \sum_{i=1}^n x_{k,i} + x_p - \frac{w}{2} \\ y_e &= \frac{1}{n} \sum_{i=1}^n y_{k,i} + y_p - \frac{h}{2} \end{aligned} \quad (5.4)$$

Da bi se provelo Kalmanovo filtriranje, koriste se dva parametra mjerena: pozicija objekta po x i po y osi. U zavisnosti od tipa kretanja, odnosno, da li postoji ubrzanje objekta ili ne, prilikom pokretanja koda se ostavlja izbor korištenja 4 ili 6 promjenjivih parametara. Četiri se

koriste u slučaju da se analizira kretanje objekta konstantnom brzinom i to su pozicija objekta u dvodimenzionalnom prostoru (koordinate x i y), te brzina kretanja po svakoj od osa. Pored ovih parametara, pri modeliranju Kalmanovog filtera za slučaj postojanja ubrzanja pri kretanju, koriste se i parametri ubrzanja po x i y osi. Broj navedenih varijabli definira dimenzije vektora stanja, a samim time i matrica koje se koriste za provođenje postupka Kalmanovog filtriranja. Vektor stanja za slučaj 6 promjenjivih parametara se definira kao $[x \ y \ v_x \ v_y \ a_x \ a_y]$, odnosno, bez posljednje dvije komponente za 4 stanja.

Za definiranje matrica prelaza stanja, inicijalne matrice mjerena, te kovarijansne matrice bijelog Gaussovog šuma, bilo bi idealno posjedovati potpune informacije o sistemu koji se analizira, kako bi se one mogle popuniti na odgovarajući način. S obzirom na to da ove informacije nisu bile dostupne, izvršeno je testiranje algoritma za više različitih vrijednosti parametara. Na taj način je utvrđeno da se, iz skupa testiranih vrijednosti, najbolji rezultati pri analizi kretanja sa ubrzanjem dobivaju korištenjem spomenutih matrica popunjениh na način prikazan relacijama 5.5-5.7, respektivno. Sa x_s i y_s su označene koordinate centra korisničke selekcije objekta na prvom fejmu videa, dok je sa \mathbf{I} u relaciji 5.7 označena jedinična identitet matrica. Za slučaj analize kretanja bez ubrzanja, koristi se analogne matrice A_k , H_k i Q_k , s tim da se dimenzije matrica mijenjaju, a vrijednosti na odgovarajućim pozicijama ostaju iste. Popunjavanje matrice prelaza stanja na način prikazan relacijom 5.5 odgovara primjeni seta jednakosti 5.8, gdje je sa v označena brzina, sa a ubrzanje, sa t vrijeme, a indeksi x i y ukazuju na njihove komponente po prostornim koordinatama.

$$A_k = \begin{bmatrix} 1 & 0 & dT & 0 & \frac{dT}{2} & 0 \\ 0 & 1 & 0 & dT & 0 & \frac{dT}{2} \\ 0 & 0 & 1 & 0 & dT & 0 \\ 0 & 0 & 0 & 1 & 0 & dT \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

$$H_k = [x_s \ y_s \ 0 \ 0 \ 0 \ 0] \quad (5.6)$$

$$Q_k = 10^{-4} \times \mathbf{I} \quad (5.7)$$

$$\begin{aligned} x(t+1) &= x(t) + dT \times v_x(t) + \frac{dT}{2} \times a_x(t) \\ y(t+1) &= y(t) + dT \times v_y(t) + \frac{dT}{2} \times a_y(t) \\ v_x(t+1) &= v_x(t) + dT \times a_x(t) \\ v_y(t+1) &= v_y(t) + dT \times a_y(t) \\ a_x(t+1) &= a_x(t) \\ a_y(t+1) &= a_y(t) \end{aligned} \quad (5.8)$$

Radi pojednostavljenja implementacije filtera, vrijednost dT iz relacije 5.5 se računa u svakoj iteraciji primjenom jednakosti 5.9, u kojoj je sa t označeno trenutno vrijeme (trenutna vrijednost *clock-a*), a sa t_p ista veličina iz prethodne iteracije algoritma. Normalizacija je izvršena frekvencijom *clock-a* f_c .

$$dT = \frac{t - t_p}{f_c} \quad (5.9)$$

Dalji postupak Kalmanovog filtriranja implementiran je na način opisan u poglavljju 4. Vrši se predikcija narednog stanja i korekcija vrijednosti u matricama u skladu sa predikcijom. Označavanje objekta koji je detektovan fuzijom ovog algoritma sa metodom pomjeranja srednje

vrijednosti prikazan je na slici 5.8c. Rezultati primjene ovog, kao i ostalih algoritama čija je implementacija prethodno predstavljena, bit će obrađeni u narednom poglavlju.

* * *

Ovo poglavlje opisuje način na koji je izvršena praktična realizacija algoritma za praćenje objekata na video zapisu. Objasnjen je postupak implementacije ranije opisanih teorijskih koncepta, te su predstavljene modifikacije koje su izvršene u cilju poboljšanja performansi algoritma.

Poglavlje 6

Eksperimentalna analiza

Evaluacija performansi algoritma izvršena je nad 17 video zapisa, na kojima su prikazani model za simulaciju kretanja mehaničkih dijelova VN prekidača, model samog VN prekidača, te kretanje robotske ruke. Dimenzije analiziranih video zapisa robotske ruke su 1920×1080 piksela, dok je frekvencija prikazivanja 29fps . Preostali snimci su različitih dimenzija, koje se kreću od 480×516 do 1280×1044 piksela, sa frekvencijom od 10fps . Testiranje algoritma izvršeno je korištenjem Dell računara sa Intel(R) Core i5 @ 1.70 GHz procesorskom jedinicom, 4 GB RAM memorije i Windows 7 Ultimate operativnim sistemom.

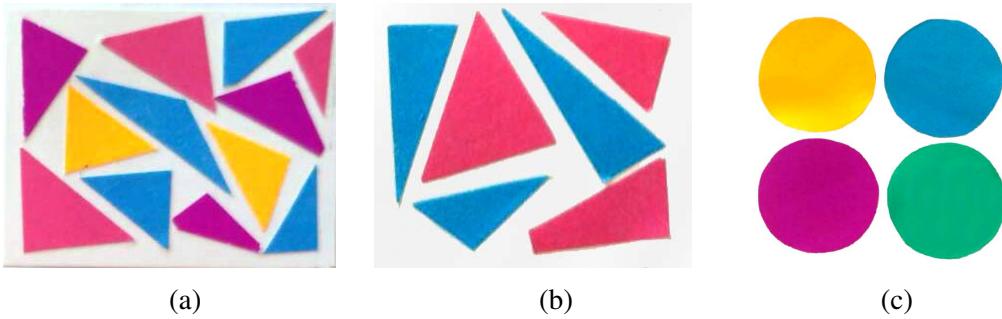
Snimci robotske ruke su namjenski napravljeni za potrebe ovog rada i stoga su generisani različiti tipovi trajektorija koji bi omogućili što bolje utvrđivanje performansi algoritma, dok su za modele VN prekidača i njegovog kretanja korišteni već postojeći video zapisi. Trajektorije kretanja robotske ruke zadane su korištenjem ROS (engl. *Robot Operating System*) softvera, kroz definiranje uglova za svaki od 6 stepeni slobode robota. Uglovi su izračunati primjenom inverzne kinematike nad skupom trodimenzionalnih prostornih koordinata, na osnovu kojih je omogućena usporedba stvarne sa ekstraktovanom trajektorijom.

Provedena je analiza tačnosti praćenja i osjetljivosti implementiranih algoritama, kao i analiza brzine njihovog izvršavanja. Ovi postupci su detaljnije predstavljeni u nastavku poglavlja.

6.1 Analiza tačnosti praćenja i osjetljivosti implementiranih algoritama

Da bi se omogućio proračun devijacije ekstraktovane trajektorije kretanja objekta od očekivane, formirane su datoteke u koje se tokom obrade frejmova upisuju koordinate centralnih tačaka detektovanih prozora praćenja. Pored toga, pri analizi kretanja robotske ruke se preuzimaju informacije o početnim položajima obojenih pozadinskih tačaka (markera), kako bi se mogla izvršiti korekcija slučajnih pomaka kamere. U slučaju praćenja modela za simulaciju kretanja mehaničkih dijelova visokonaponskog prekidača nema potrebe za izdvajanjem dodatnih informacija o pozicijama pojedinih tačaka, jer je izvršena kalibracija kamere. Informacije o tačnim pozicijama sastavnih tačaka trajektorije ovih modela nažalost nisu dostupne, te je stoga analiza rezultata algoritama u ovom eksperimentu provedena korištenjem poznatih informacija o kretanju objekata koji se prate.

Za ostvarenje mogućnosti analize tačnosti algoritma potrebno je riješiti i problem nepodudarnosti zadanih tačaka trajektorije sa ekstraktovanim. Problem se javlja ako korisnik može u bilo kom trenutku gledanja video zapisa označiti objekat koji želi pratiti. Od tog trenutka se vrši zapisivanje informacija o centru pronađenog prozora praćenja, dok su informacije o stvar-



Slika 6.1: Prikaz markera korištenih za praćenje kretanja robotske ruke. (a), (b) Marker sa više boja. (c) Jednobojni markeri, koji su osim za praćenje korišteni i za korekciju eventualnog pomjeranja kamere.

noj trajektoriji zadane od trenutka kad robotska ruka dođe u početnu poziciju za "iscrtavanje" zadane trajektorije. U implementiranom algoritmu je izvršeno zaustavljanje reprodukcije videa na prvom frejmu, dok korisnik ne označi objekat za praćenje. Na taj način se informacije o njegovoj poziciji zapisuju u datoteku od početka do kraja reprodukcije videa i sprječava se selekcija objekta u toku njegovog kretanja po zadanoj putanji. U sklopu obrade ekstraktovanih podataka, izdvaja se samo onaj dio trajektorije koji se odnosi na zadano kretanje, radi utvrđivanja tačnosti detekcije algoritma. Podaci koji su zapisani u datoteke tokom izvršavanja programskog koda su naknadno obrađeni pomoću skripte u MATLAB okruženju i na osnovu njih su formirani odgovarajući grafici upotrebljem TikZ alata.

6.1.1 Eksperiment 1: Kretanje robotske ruke

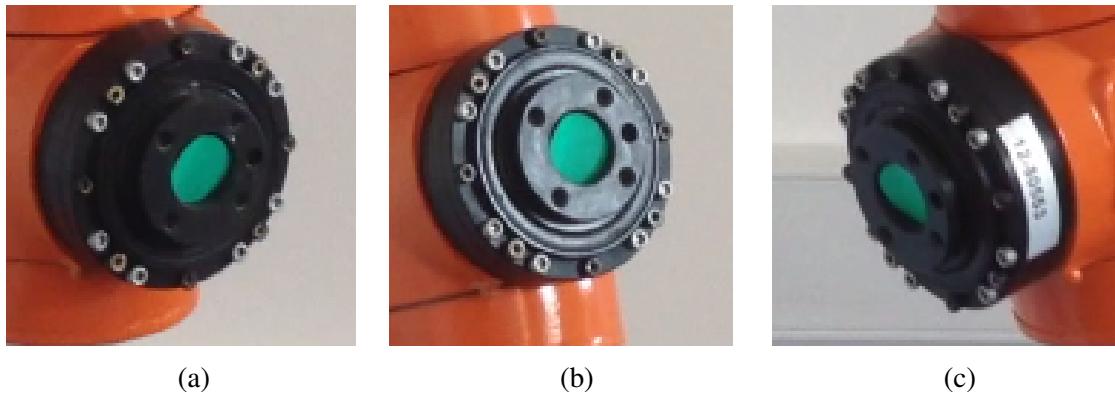
U okviru ovog potoglavlja izdvojiti će se karakteristični slučajevi provedene analize nad 10 video zapisa kretanja robotske ruke, kroz koje je moguće uočiti prednosti i nedostatke svake od implementiranih metoda za praćenje objekta na video zapisu.

Markeri koji su korišteni u svrhu praćenja kretanja prikazani su na slici 6.1. Ispitano je kretanje pri različitim brzinama (sa i bez ubrzanja), te zadavanjem različitih trajektorija (lemniskata, kružnica, kvadrat), pri čemu dolazi do promjene orientacije objekta koji se prati, promjene osvjetljenja tokom video zapisa, pojave ubrzanog kretanja i nagle promjene smjera kretanja, te do djelimične okluzije. Analizirani su samo dijelovi trajektorija, s obzirom na to da se na procesiranim video zapisima robotske ruke trajektorija ponavlja.

Radi utvrđivanja tačnosti i osjetljivosti algoritma izvršena je usporedba ekstraktovane trajektorije kretanja po x i y osi sa zadanim trajektorijom. Utvrđivanje prosječnog odstupanja od očekivane trajektorije izvršeno je primjenom relacije 6.1, gdje je sa N označen ukupan broj ekstraktovanih parova prostornih koordinata, tj. ukupan broj analiziranih frejmova. $x_{z,N}$ i $y_{z,N}$ su koordinate zadane trajektorije u N -tom frejmu, dok su $x_{e,N}$ i $y_{e,N}$ ekstraktovane koordinate za isti.

$$d = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_{z,N} - x_{e,N})^2 + (y_{z,N} - y_{e,N})^2} \quad (6.1)$$

Na prvom razmatranom video zapisu praćena je selekcija objekta (zeleni marker na crnoj podlozi) na kojoj su prepoznate dvije boje. Objekat se kreće bez ubrzanja u uslovima promjenjivog osvjetljenja, a pri kretanju se javlja djelimična okluzija, koju algoritam uspijeva prevazići i nastaviti praćenje objekta. Primjer problema s kojima se suočava algoritam u ovom slučaju prikazan je na slici 6.2.



Slika 6.2: Primjer problema koje je potrebno riješiti za uspješno praćenje objekata na video zapisu robotske ruke koja se kreće po trajektoriji kružnice. (a) Marker na početnom frejmu. (b) Promjena osvjetljenja. (c) Djelimična okluzija.

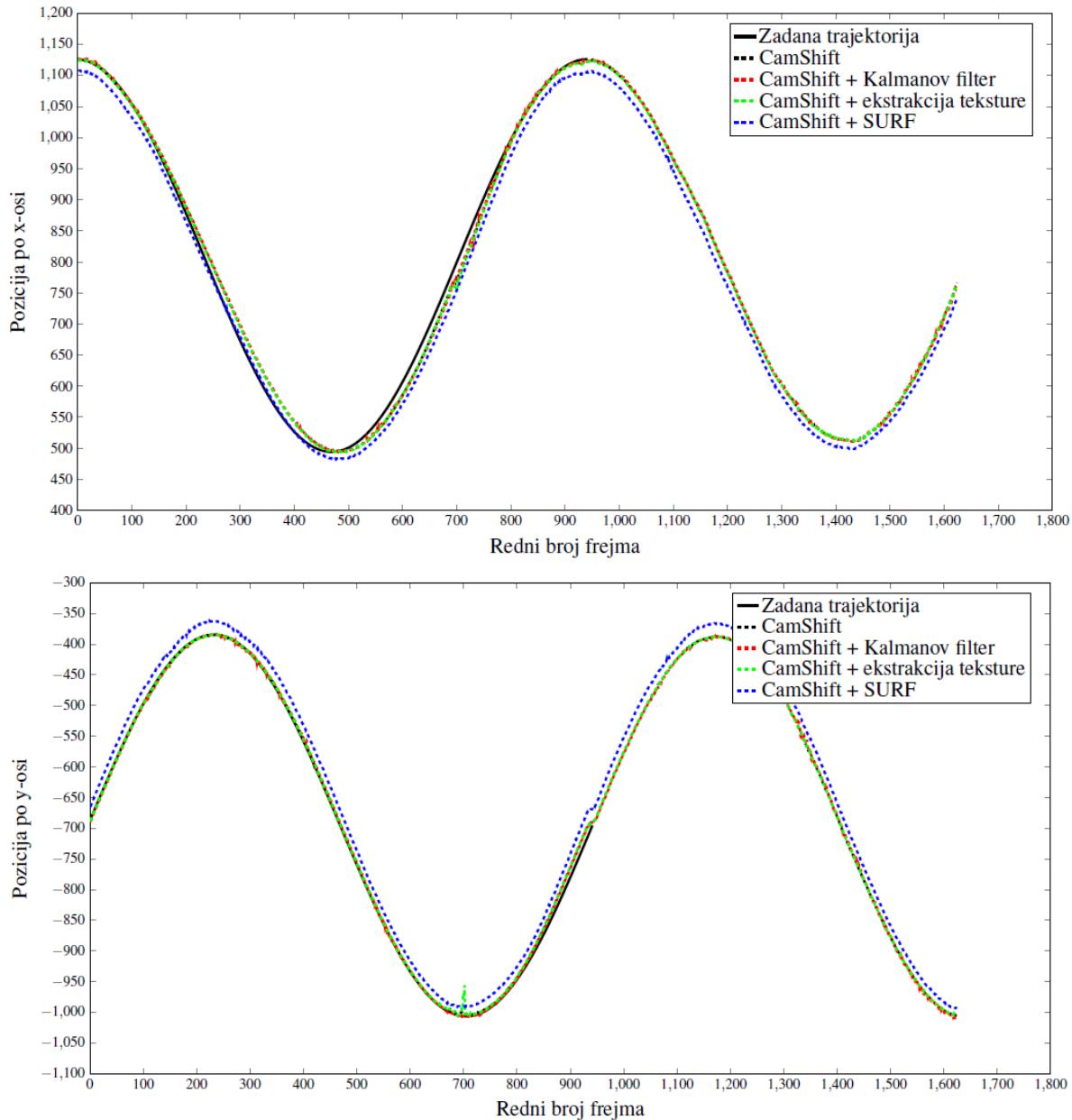
Rezultati koji su dobiveni primjenom modificiranog CamShift algoritma nad konkretnim video zapisom dati su na grafiku 6.3, s kog je uočljivo da nema velikih odstupanja u odnosu na zadanu trajektoriju. Prosječna greška koja se pravi je za razmatrani slučaj jednaka 15.48 piksela za analizirana 1624 frejma.

Ukoliko se uključi i pretraga po teksturi, prosječna greška se neznatno smanjuje i iznosi 13.42 piksela, ali vizuelno razlike nisu uočljive, jer se vrše samo male korekcije pomaka prozora praćenja. Empirijskim putem je utvrđeno da se korištenjem većih pomaka prozora za pretragu po teksturi greška povećava, zbog čega je zadržana mogućnost samo malih pomjerenja. U konkretnom slučaju je pri pomaku do 8 piksela u 8 definiranih pravaca, kako je to opisano u prethodnom poglavljju, ostvarena greška od 17.32 piksela. Razlog povećanja greške je mali broj značajki teksture koji se može ekstraktovati.

Kako se radi o praćenju markera bez mnogo detalja, uključenjem SURF pretrage se ne postiže pronalaženje mnogo karakterističnih tačaka na selekciji objekta, kao ni unutar prozora praćenja. Stoga se vrlo rijetko vrši uparivanje ključnih tačaka i sam SURF algoritam u ovom slučaju ne daje dobre rezultate i ne uspijeva prepoznati objekat koji je potrebno pratiti. Kako su u sklopu implementiranog algoritma korištene samo informacije o pozicijama uparenih tačaka radi pomjerenja prozora praćenja dobivenog CamShift algoritmom, ostvareni su znatno bolji rezultati nego što bi bili sa samom SURF metodom. Ipak, karakteristične tačke koje pronalazi SURF u proširenom prozoru praćenja većinom se nalaze na granici markera sa pozadinom, pa u ovom slučaju dodatna informacija uvodi pogoršanje rezultata ekstrakcije trajektorije. Izračunata je srednja greška od 26.31 piksela.

Fuzijom CamShift algoritma sa Kalmanovim filterom se u analiziranom slučaju postižu slični rezultati kao pri praćenju samim CamShift-om, jer ne dolazi do pojave potpune okluzije i samo praćenje po boji obezbjeđuje dosta dobre rezultate, pa se ne javlja potreba za predikcijom pozicije objekta. Greška koja se postiže jednaka je 15.38 piksela za analizirani broj frejmova.

Drugi karakteristični slučaj je video zapis robotske ruke koja se kreće bez ubrzanja formirajući lemniskatu, pri čemu je kao objekat koji se prati označen marker predstavljen na slici 6.1a. Tokom video zapisa dolazi do promjene orientacije objekta i do djelimične okluzije uslijed zakretanja robotske ruke pri formiraju bočnih rubova lemniskate. Cilj je ispitati djelovanje algoritama pri praćenju objekta sa više boja, s obzirom na to da je poznati nedostatak CamShift algoritma nemogućnost praćenja takvih objekata.

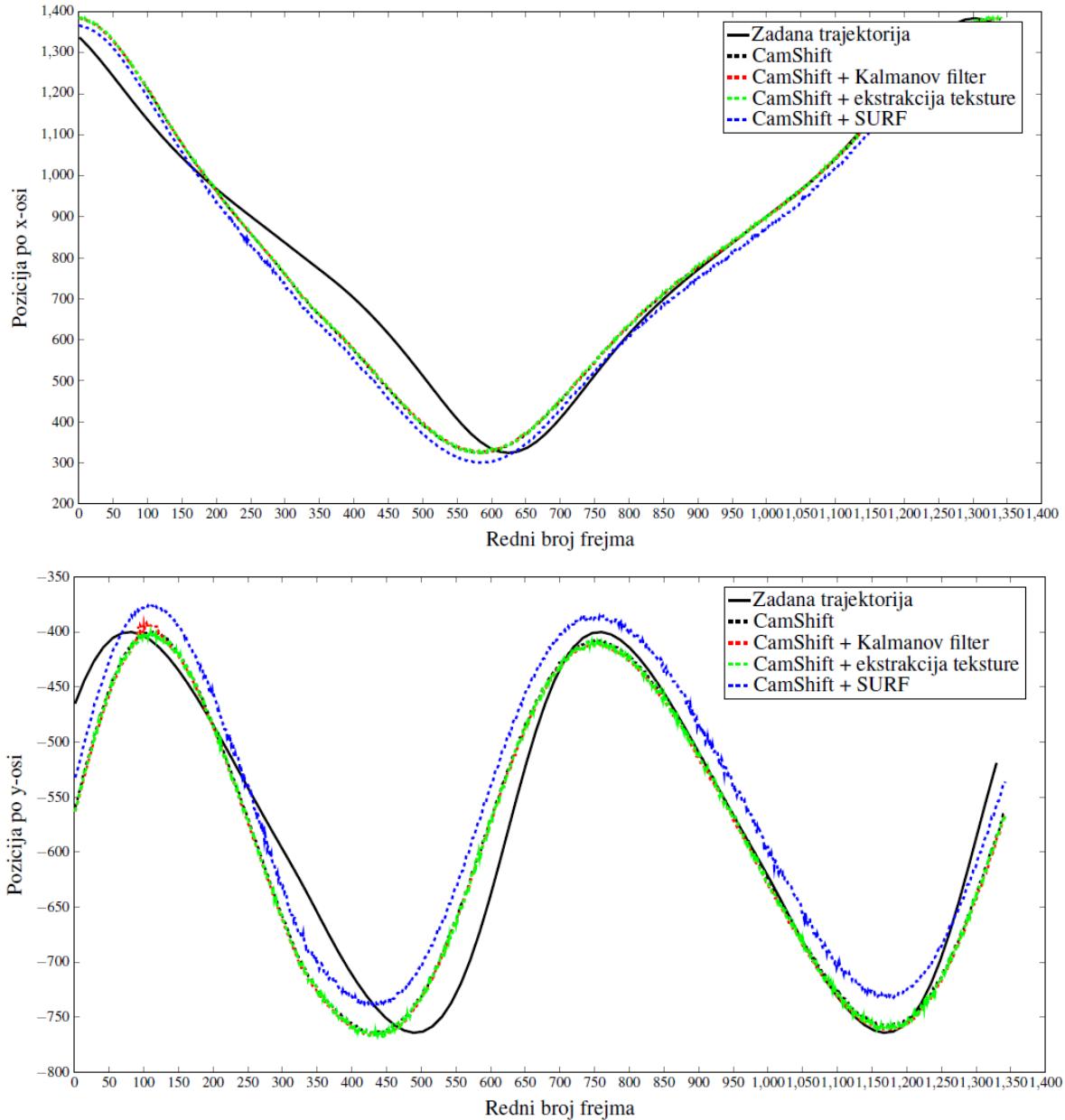


Slika 6.3: Usporedba zadane i ekstraktovane trajektorije kretanja (kružnica) po x i y osi, korištenjem modificiranog CamShift algoritma, te njegove kombinacije sa ekstrakcijom teksture, Kalmanovim filterom i SURF algoritmom.

Kada se praćenje vrši isključivo na osnovu boje, ekstrahuje se trajektorija prikazana crvenom linijom na grafiku 6.4. Uočava se veće odstupanje od zadane trajektorije u odnosu na prethodno analizirani slučaj, a prosječna greška koja se pravi jednaka je 54.13 piksela na 1343 frejma. Razlog tome je činjenica da se vrši zamagljenje višebojnog objekta koji se prati, čime se značajke boje proširuju po njegovoj okolini, pa dolazi do pomjeranja centra ekstraktovane pozicije praćenog objekta. Oblik trajektorije je vrlo sličan zadanoj, a radi usporedbe očekivanih rezultata sa dobivenim, prikaz ekstraktovane putanje dat je na slici 6.5. Može se zaključiti da se objekat uspješno prati, ali da je smanjena preciznost praćenja centralne tačke njegove pozicije.

Ekstrakcija teksture ni u ovom slučaju nije doprinijela znatnom poboljšanju rezultata praćenja. Ostvarena greška u prosjeku iznosi 53.15 piksela, a prikaz rezultata je na slici 6.4 označen

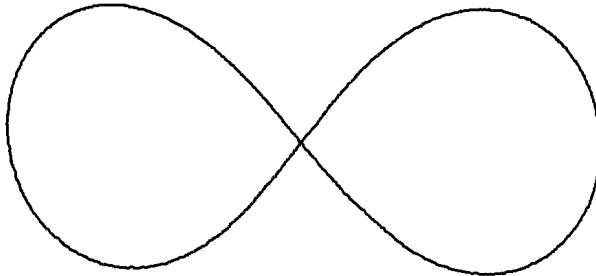
zelenom bojom.



Slika 6.4: Usporedba zadane i ekstraktovane trajektorije kretanja (lemniskata) po x i y osi, korištenjem modificiranog CamShift algoritma, te njegove kombinacije sa ekstrakcijom teksture, Kalmanovim filterom i SURF algoritmom.

Za razliku od prethodno analiziranog slučaja, SURF sada ima mogućnost detekcije velikog broja ključnih tačaka, s obzirom na to da se prati marker kog opisuje veliki broj značajki. Ključne tačke selekcije i detektovanog objekta se uspješno povezuju i na nekim frejmovima je moguća lokalizacija objekta primjenom standardne SURF procedure. Međutim, većina ključnih tačaka se u analiziranom slučaju pronađe na jednom dijelu markera, zbog čega se ponovo prozor praćenja pomjera ka centru raspodjele tih tačaka. Prosječna greška pritom iznosi 68.98 piksela.

Primjenom Kalmanovog filtera se ponovo ostvaruju slični rezultati kao u slučaju korištenja



Slika 6.5: Ekstraktovana putanja kretanja robotske ruke koja formira lemniskatu.

same informacije o boji, jer je on implementiran tako da prati objekat na osnovu projekcije njegovog histograma na frejm. Samim time, nije u mogućnosti korigovati *blurring* piksela u okolini objekta, koji čine da se ekstraktovana centralna pozicija pomjera u odnosu na stvarnu. Priloženim rezultatima sa grafika 6.4 odgovara srednja greška od 54.31 piksela.

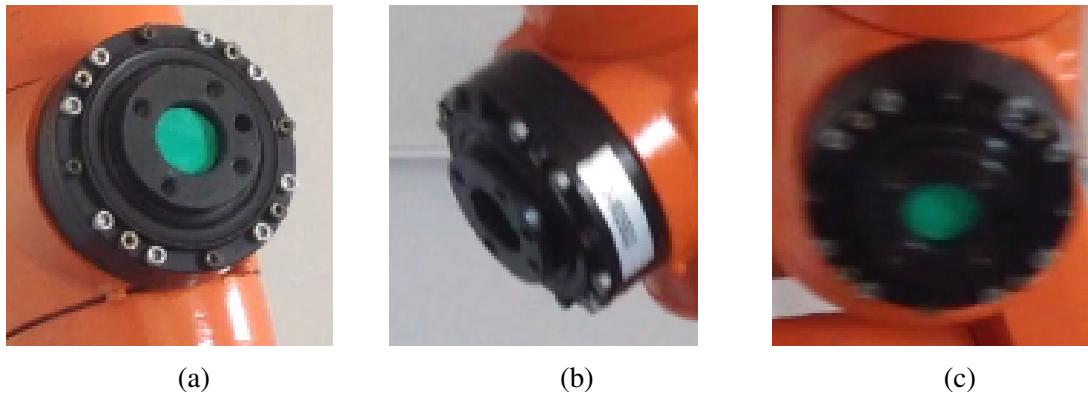
U dva prethodno analizirana slučaja nema naglih promjena kretanja. Međutim, za zadalu trajektoriju kvadrata, robotska ruka pravi nagle promjene smjera kretanja. Kretanje je linearne tokom "iscrtavanja" stranica kvadrata, ali se u njihovim krajnjim tačkama mora promijeniti ugao kretanja za 90° . Djelimična okluzija je prisutna u većini frejmova videa, dok potpuna nastaje pri formiranju ugla između donje i lijeve stranice kvadrata. Pritom je kretanje vrlo brzo, a prati se zeleni marker prikazan na slici 6.1b, kao i u prvom analiziranom slučaju. Razlika je, međutim, u tome da je ovaj put selekcija objekta izvršena tako da nije obuhvaćen veliki dio podloge markera, pa je detektovana jedna boja. Problemi koji se javljaju u razmatranom video zapisu prikazani su na slici 6.6.

Robotska ruka je na početnom frejmu pozicionirana tako da je marker jasno vidljiv, kao što se primjećuje sa slike 6.6a. Nakon toga, robotska ruka dolazi u početnu poziciju zadane trajektorije (donji lijevi ugao kvadrata) i od tog trenutka se ekstrahuju pozicije pronađenih prozora praćenja. Međutim, upravo je početna pozicija robotske ruke kritični dio trajektorije, na kom dolazi i do potpune okluzije i do nagle promjene smjera kretanja.

Modificirani CamShift algoritam nema mogućnost oporavka od potpune okluzije, zbog čega se privremeno gubi informacija o poziciji objekta. U tom slučaju se prozor praćenja vraća u koordinatni početak. Kada se na projekciji histograma na frejm ponovo detektuje objekat, praćenje se nastavlja. Prikaz ekstraktovane trajektorije kvadrata dat je na grafiku 6.7, gdje je crnom isprekidanim linijom označena trajektorija izvedena korištenjem same ekstrakcije boje. Međutim, u većini tačaka se ova trajektorija preklapa sa onom dobivenom uključenjem i informacija o teksturi, pa na prikazanom grafiku nije jasno uočljiva. Za ovaj slučaj je ostvarena prosječna greška od 408,9 piksela. Ovako velika greška je dobivena iz razloga što se pri gubitku informacije o stvarnoj poziciji objekta proračun greške vrši korištenjem ekstraktovane pozicije (0,0).

Sa grafika se može primijetiti i da sve metode imaju vidljivo odstupanje amplitude po y osi. Razlog tome je ranije spomenuta okluzija, koja naročito dolazi do izražaja kada se robotska ruka kreće po trajektoriji bočnih stranica kvadrata. U tom slučaju se centar prozora praćenja pomjera prema unutrašnjosti frejma, a taj pomak je još više naglašen uslijed brzog kretanja objekta na video zapisu. Time je uvedeno dodatno zamagljenje, a broj frejmova u sekundi je nedovoljan za precizno praćenje. Obrada snimaka brze kamere bi u ovom slučaju rezultirala tačnjom ekstrakcijom trajektorije.

Ukoliko se pored informacije o boji uključi i informacija o teksturi, ne obezbeđuje se oporavak od okluzije i rezultati stoga nisu poboljšani u odnosu na praćenje samim CamShift algo-



Slika 6.6: Primjer problema koje je potrebno riješiti za uspješno praćenje objekata na video zapisu robotske ruke koja se kreće po trajektoriji kvadrata. (a) Marker na početnom frejmu. (b) Potpuna okluzija. (c) Zamagljenje (*blurring*) uslijed brzog kretanja.

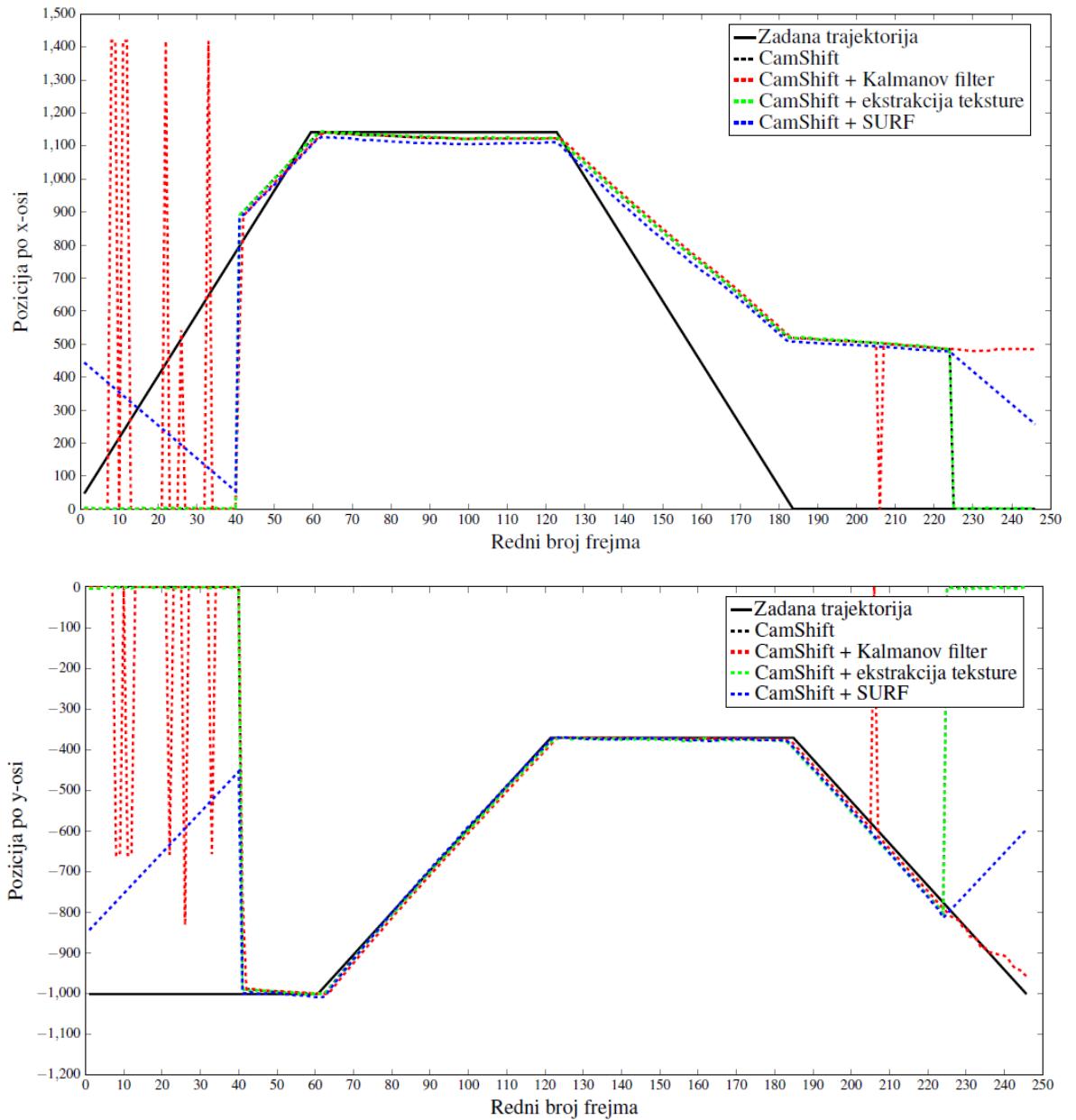
ritmom. Prosječno odstupanje od zadane trajektorije jednako je 408,84 piksela.

Kako se, kao i u slučaju praćenja robotske ruke koja se kreće po kružnoj putanji, prati marker koji nema mnogo ključnih tačaka koje SURF algoritam može detektovati, informacije koje on obezbjeđuje u ovom slučaju ne pospješuju rezultate. Međutim, primjenom definirane metrike za određivanje prosječne greške, dobiva se znatno manja vrijednost, zato što se pri gubitku mogućnosti praćenja objekta u slučaju SURF metode samo proširuje prozor pretrage, dok se u ostalim metodama vrši povratak u koordinatni početak. Greška je u ovom slučaju u prosjeku jednaka 265,39 piksela na 650 frejmova.

Kalmanov filter se pokazao djelimično uspješnim u borbi protiv okluzije, s obzirom na to da trajektorija kretanja nije izgubljena pri prolasku robotske ruke putanjom desne bočne stranice kvadrata, kao u svim prethodnim slučajevima (grafik 6.7). Postoji određeno odstupanje od zadane trajektorije, koje je u prvom redu određeno pomjeranjem prozora praćenja pri pojavi parcijalne okluzije koja nastupa prilikom iscrtavanja bočnih strana kvadrata. Predviđena trajektorija kretanja se onda oslanja na prethodno izvedenu, pa se greška stoga akumulira u ovom slučaju. Razlog zbog kog Kalmanovo filtriranje nije riješilo problem pomaka prozora praćenja pri parcijalnoj okluziji je taj da se ona ne pojavljuje naglo, već postepeno, te da se boja objekta ipak detektuje CamShift algoritmom, koji je osnova programskog koda. Time se objekat smatra pronađenim, a Kalmanov filter adaptira vrijednosti matrica na osnovu kojih vrši predikciju u skladu sa takvim kretanjem.

Kako je na samom početku kretanja po zadanoj trajektoriji nastupila potpuna okluzija na mjestu gdje se vrši nagla promjena kretanja, Kalmanov filter nije mogao izvršiti takvu predikciju i stoga se na početnim fremovima objekat ne detektuje, kao ni primjenom ostalih metoda. Prosječna greška primjene ove metode u analiziranom slučaju jednaka je 372,25 piksela.

Preostali analizirani video zapisi sadrže kombinaciju problema koji su predstavljeni u okviru ovog poglavlja i daju rezultate u skladu sa zaključcima izvedenim na osnovu predstavljena tri slučaja analize.

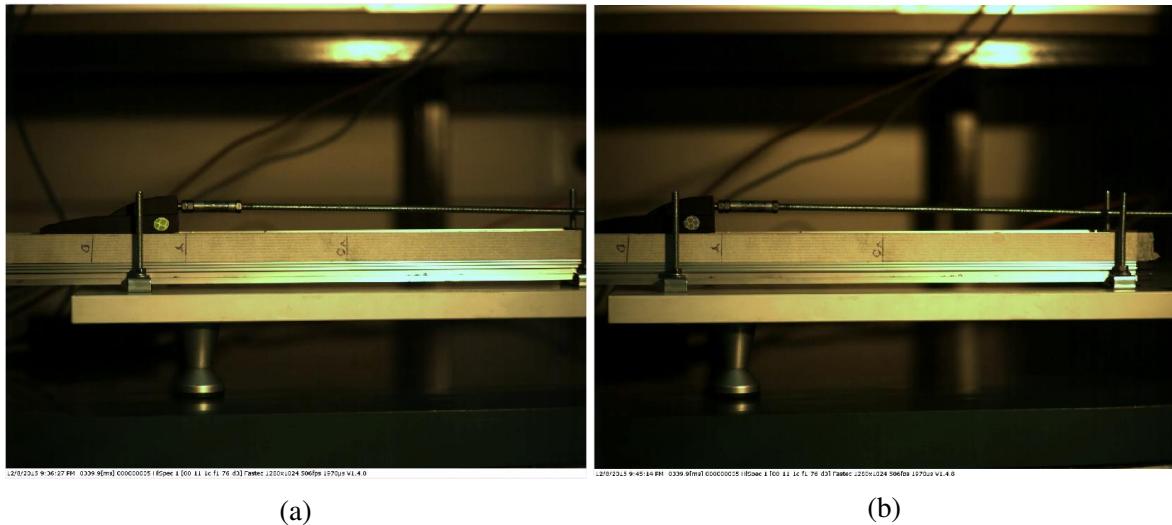


Slika 6.7: Usporedba zadane i ekstraktovane trajektorije kretanja (kvadrat) po x i y osi, korištenjem modificiranog CamShift algoritma, te njegove kombinacije sa ekstrakcijom teksture, Kalmanovim filterom i SURF algoritmom.

6.1.2 Eksperiment 2: Kretanje modela za simulaciju uklopnog mehanizma VN prekidača

Drugi eksperiment je izvršen nad dva snimka modela za simulaciju uklopnog mehanizma visokonaponskog prekidača ((6.8)), pri čemu je nivo osvjetljenja na analiziranim snimcima različit. U okviru ovog potpoglavlja priloženi su rezultati djelovanja algoritama pri analizi snimka sa slabijim osvjetljenjem (6.8b), radi utvrđivanja njegove robusnosti. Ekstrakcija trajektorije kretanja vrši se na osnovu zelenog markera kružnog oblika.

Uzimajući u obzir da se radi o linearном kretanju, koje je prikazano na slici 1.1 u okviru poglavlja 2, u idealnom slučaju ne bi trebalo postojati kretanje po y osi. Međutim, usljud brzine



Slika 6.8: Analizirani snimci modela za simulaciju uklopnog mehanizma VN prekidača sa različitim nivoima osvjetljenja: (a) izraženije osvjetljenje; (b) slabije osvjetljenje.



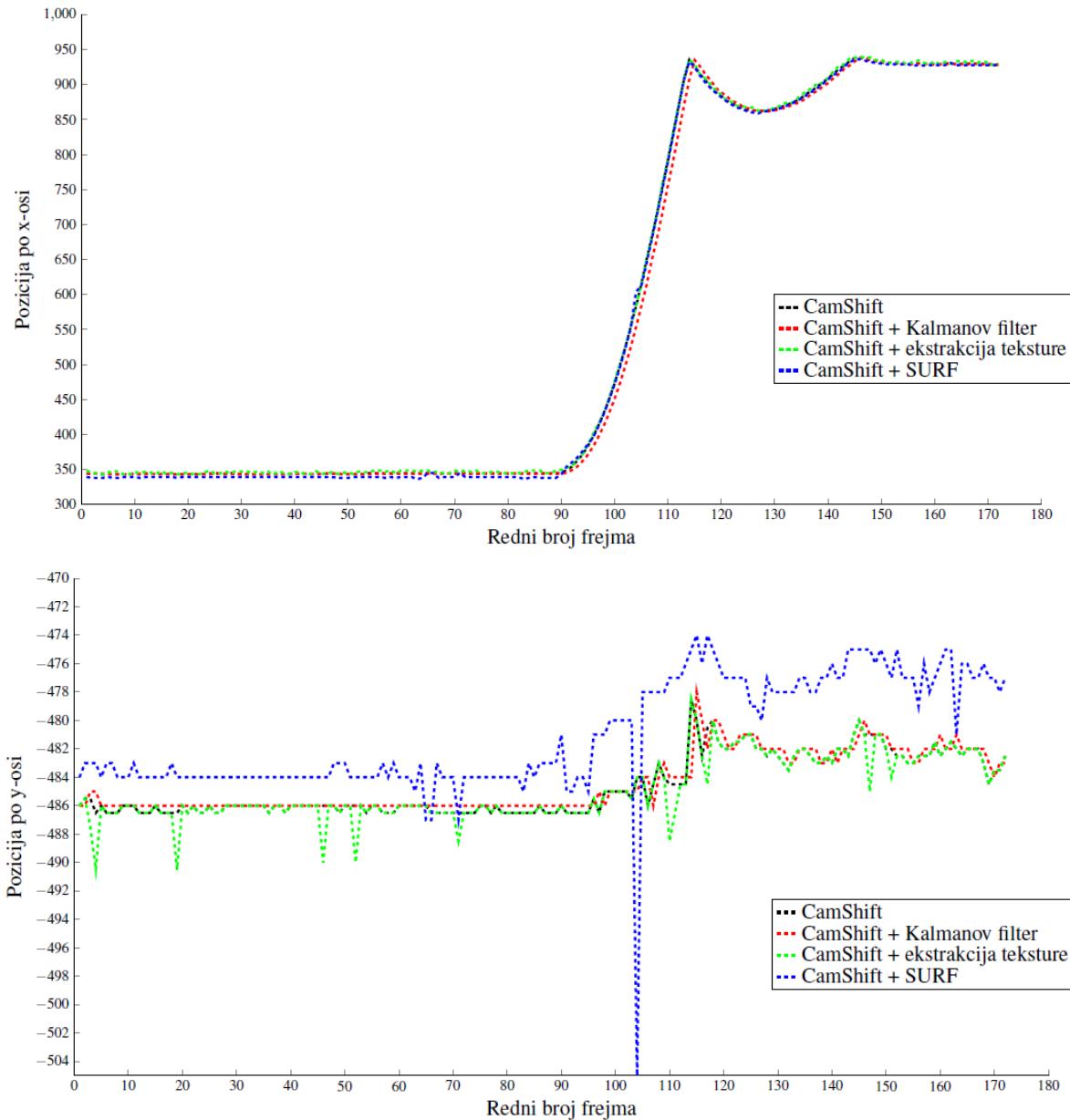
Slika 6.9: Prikaz modela VN prekidača čije se kretanje prati. (a) Isječak sa video zapisa na kom je prikazan isklop. (b) Isječak sa video zapisa na kom je prikazan uklop.

zaustavljanja objekta i njegovog odbijanja u suprotnom smjeru, i na samom snimku dolazi do blagog kretanja (trzaja) po y-osi. Sa grafika 6.10 dobivenih rezultata se može uočiti da svaka od primijenjenih metoda kao rezultat daje određena odstupanja od konstantne vrijednosti, pri čemu su ona izraženija na posljednjim frejmovima analiziranog videa. Taj dio odgovara stvarnom postojanju kretanja po apscisnoj osi. Sa grafika je uočljivo da sama CamShift metoda, te njeni kombinaciji sa Kalmanovim filterom imaju najmanje odstupanje od konstante, što ukazuje na njihove bolje performanse u odnosu na druge metode u razmatranom slučaju, s obzirom na to da je kretanje po x osi približno jednako za sve primjenjene algoritme.

Pored ovoga, izvršeno je praćenje kretanja modela VN prekidača u uslovima vrlo slabe osvjetljenosti. Radi uočljivosti markera, na slici 6.9 su dati isječci iz video sekvenci sa nešto boljim osvjetljenjem.

Na snimku čiji je isječak prikazan na slici 6.9a je izvršeno praćenje crvenkaste tačke koja prilikom isklopa pravi rotaciono kretanje. Rezultati primijenjenih algoritama prikazani su na slici 6.11, sa koje je uočljivo da ne postoji značajna odstupanja u ekstraktovanoj trajektoriji, bez obzira na primijenjenu metodu. Razlog tome je prvenstveno implementacija CamShift

algoritma, koja omogućava dobru detekciju objekta na osnovu njegove boje i time ne ostavlja mnogo prostora za odstupanja od trajektorije koja je ekstraktovana njim samim.

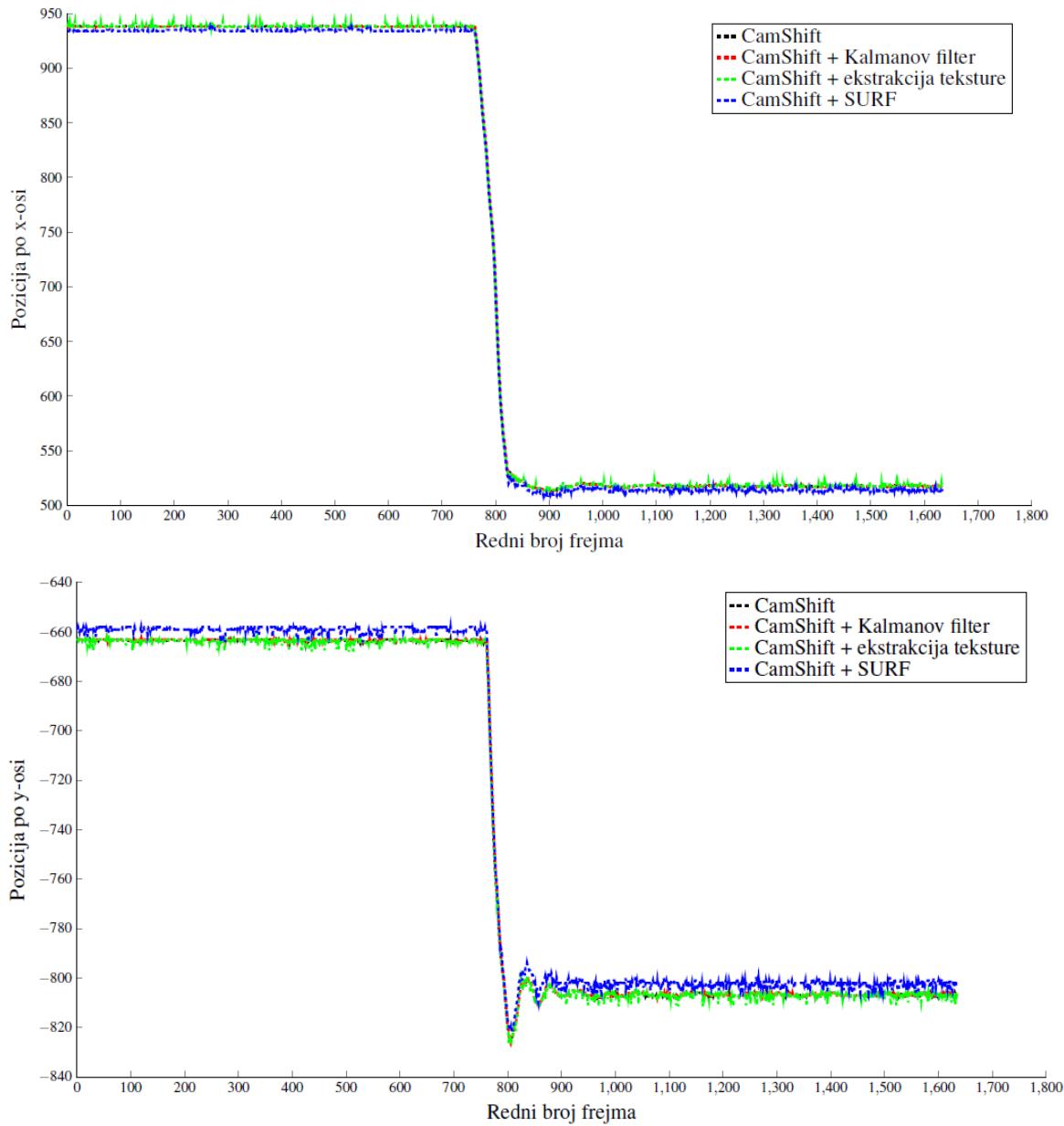


Slika 6.10: Ekstraktovana trajektorija kretanja modela za simulaciju uklopnog mehanizma VN prekidača.

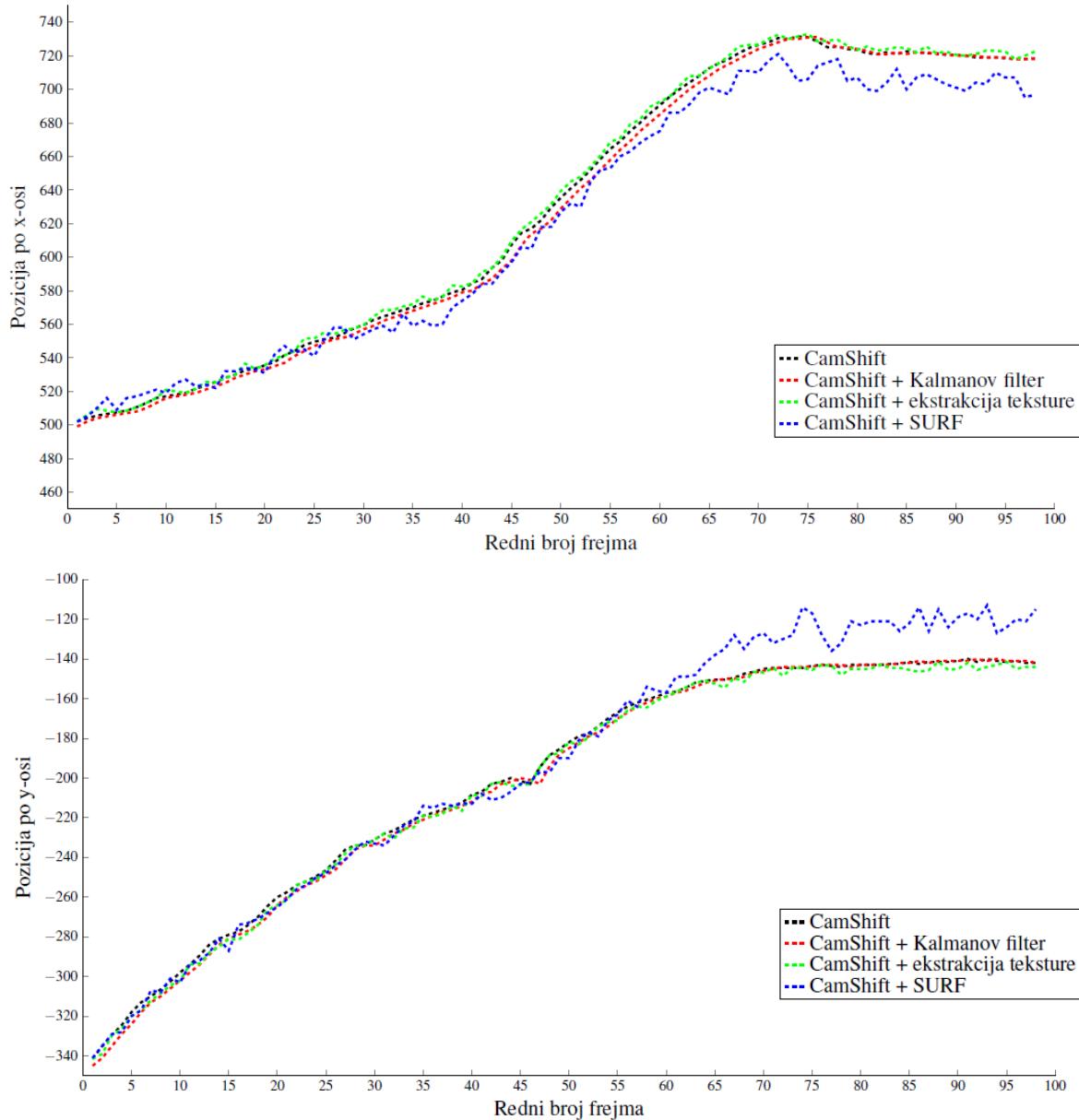
Analizom kretanja modela VN prekidača pri uklopu, praćenjem markera sa više boja, prikazanog na slici 6.9b, dobiveni su rezultati predstavljeni grafikom 6.12. Treba napomenuti da je skala po y osi na spomenutom grafiku značajno uža u odnosu na onu po x osi. I u ovom slučaju je riječ o rotacionom kretanju, a testira se mogućnost praćenja više boja u uslovima industrijskog okruženja. Iz dobivenih rezultata se intuitivno može zaključiti da je praćenje označenog objekta omogućeno, ali da je u analiziranom slučaju preciznost smanjena. Ovaj zaključak se izvodi iz postojanja oscilacija ekstraktovane putanje. Pritom rezultati SURF metode nešto više odstupaju od ostalih, naročito na posljednjim frejmovima videa. Razlog tome je činjenica da

SURF algoritam može detektovati veliki broj ključnih tačaka objekta i na osnovu njihove lokalizacije izvršiti pomjeranje prozora praćenja ka centru njihove najveće koncentracije. Dakle, mogu se potvrditi zaključci izvedeni u okviru prethodnog eksperimenta.

Analiza je provedena i na preostalim dostupnim video zapisima, te su dobiveni slični rezultati ekstraktovanih trajektorija. Pri analizi snimaka niže rezolucije i jako slabog osvjetljenja, pogodno je podesiti zadane pragove tona, nijanse i zasićenja na način da se dozvoli manje odstupanje u odnosu na ekstraktovane HSV značajke objekta koji se prati. Na taj način se ostvaruje bolje maskiranje projekcije i omogućava ekstrakciju trajektorije sa manje oscilacija.



Slika 6.11: Ekstraktovana trajektorija kretanja modela VN prekidača pri isklopu.

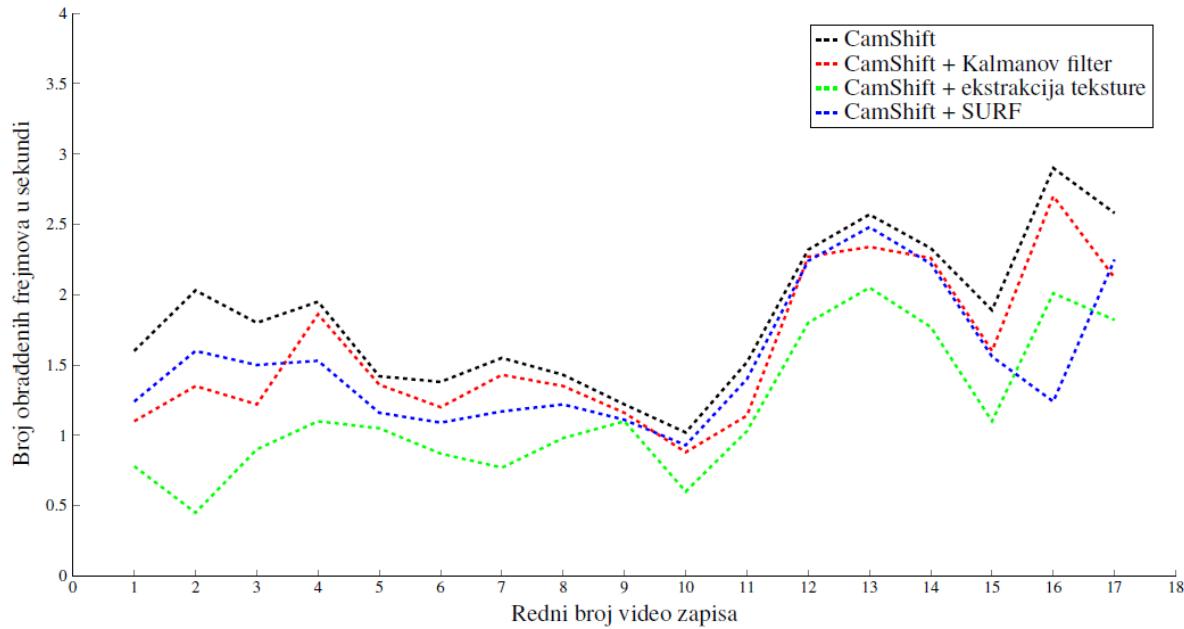


Slika 6.12: Ekstraktovana trajektorija kretanja modela VN prekidača pri uklopu.

6.2 Analiza brzine izvršavanja algoritma za praćenje objekata na video zapisu i diskusija rezultata

Prilikom izvršavanja algoritma bilježi se početno i krajnje vrijeme obrade, radi mogućnosti analize vremena procesiranja video zapisa. Na grafiku 6.13 je prikazan broj frejmova koji se mogu obraditi u jedinici vremena pri izvršavanju svake od primijenjenih metoda za analizirane video sekvence. Podaci za prvih 10 video zapisa na spomenutom grafiku odgovaraju snimcima robotske ruke, dok se preostali odnose na one obuhvaćene eksperimentom 2. Na osnovu same ove informacije i usporedbe vremena izvršavanja, može se potvrditi intuitivna pretpostavka da dimenzije, tj. rezolucija video zapisa utječe na vrijeme njegove obrade.

U skladu sa očekivanjima, sam CamShift algoritam zahtijeva najmanje vremena za izvrša-



Slika 6.13: Dijagram broja procesiranih frejmova u jedinici vremena za analizirane video zapise.

vanje, odnosno, može obraditi najveći broj frejmova u sekundi. Vrijeme potrebno za obradu se produžava uvođenjem dodatnih metoda i informacija na osnovu kojih će se pratiti objekat na video zapisu.

Ekstrakcija značajki teksture, prema dobivenim rezultatima, zahtijeva najviše vremena za izvršenje. S obzirom na to da se za svaki frejm kreiraju GLCM matrice dimenzija jednakih dimenzijama selekcije objekta na prvom frejmu, vrijeme izvršavanja prvenstveno zavisi od veličine selekcije. Sam rad sa GLCM matricama je složen i zahtijeva mnogo vremena za obradu parametara na osnovu kojih će se ustanoviti eventualna teksturalna sličnost objekata.

Dodatak SURF metode ili Kalmanovog filtriranja ima približno jednak utjecaj na produženje vremena izvršavanja algoritma. Međutim, treba napomenuti da je za primjenu SURF metode korišten mali prozor pretrage, koji je omogućio brži rad ove metode. Za sve navedene nadogradnje CamShift algoritma, potrebno je utvrditi kada njihovo korištenje zaista uvodi poboljšanje rezultata.

Smanjenje vremena izvršavanja algoritma može se ostvariti njegovom optimizacijom, dok se za potrebe praćenja objekata koji se sporo kreću može koristiti obrada svakog n -toga frejma videa, bez umanjenja tačnosti praćenja. Naravno, podrazumijeva se da n ima dovoljno malu vrijednost, prilagođenu brzini kretanja objekta. Na taj način se algoritam može dalje unaprijediti za rad u (približno) realnom vremenu.

6.3 Diskusija rezultata

Na osnovu provedenih eksperimenata može se zaključiti da implementirana modifikacija CamShift algoritma ima dosta dobre performanse. Prevaziđen je problem nemogućnosti praćenja objekata sa više boja, koji ima tradicionalni algoritam. Ekstraktovana trajektorija pritom odgovara zadanoj, s tim da je translirana za određeni broj piksela, u zavisnosti od veličine elementa kojim je izvršeno zamagljenje objekta radi mogućnosti praćenja. Algoritam je također pokazao dosta veliku otpornost na parcijalnu okluziju, čijim nastupanjem objekat u većini slučajeva nije izgubljen u procesu praćenja. Međutim, slično kao pri praćenju objekta s više boja, okluzija uzrokuje pomjeranje prozora praćenja. Ovaj nedostatak ima i tradicionalni algoritam. Nemogućnost ekstrakcije trajektorije se javlja onda kada se u kombinaciji sa parcijalnom okluzijom objekat brzo kreće ili naglo mijenja svoj smjer kretanja. Korištenje brze kamere za snimanje video zapisa može riješiti ovaj problem.

Dakle, kroz provedenu analizu je utvrđeno da preciznost ekstraktovane trajektorije zavisi od uslova u kojima je snimak napravljen (nivo i promjenjivost osvjetljenja, šum, okluzija i sl.), karakteristika kamere (rezolucija, broj frejmova u jedinici vremena), te od samog kretanja objekta (translacija, rotacija, nagla promjena ugla kretanja, brzina, ubrzanje). Ukoliko se radi o pravilnom kretanju, Kalmanov filter može uvesti poboljšanje preciznosti praćenja. Producenje vremena izvršavanja algoritma njegovim uključenjem nije veliko, zbog čega bi implementacija fuzije CamShift algoritma i Kalmanovog filtriranja bila opravdana u slučaju praćenja objekata koji nemaju nagle promjene smjera kretanja ili onda kada se javlja potpuna okluzija.

Dodatna poboljšanja predikcije trajektorije korištenjem Kalmanovog filtera mogu biti ostvarena daljim testiranjem rada filtera zadavanjem različitih parametara, za čiju bi inicijalizaciju bilo optimalno poznavati u potpunosti karakteristike sistema koji se analizira.

U prethodnim potpoglavlјima je utvrđeno da doprinos poboljšanju rezultata uvođenjem informacija o teksturi u analiziranim slučajevima nije velik, dok se zahtijeva veliko produženje vremena izvršavanja algoritma. Stoga se može izvesti zaključak da za analizirani skup video zapisa nije isplativo koristiti ekstrakciju tekture kao dodatnu informaciju za praćenje.

SURF algoritam ne zahtijeva veliko dodatno vrijeme izvršavanja, ali nije pogodan za korištenje u slučaju kada se prate objekti sa malo karakterističnih tačaka. Tada fuzija ovog algoritma sa CamShift-om dovodi do pomjeranja prozora praćenja, najčešće u jednu od rubnih tačaka objekta koji se prati. Ukoliko postoji veći broj značajki, upotreba SURF-a može poboljšati rezultate. Ipak, s obzirom na to da se implementirani algoritam ne oslanja na lokalizaciju objekta putem SURF-a (jer se pokazalo da se ona uglavnom ne može izvršiti na analiziranim video zapisima), i u slučaju praćenja objekata sa više karakterističnih tačaka može doći do pogoršanja rezultata u odnosu na sam CamShift. Ovo se dešava onda kada su karakteristične tačke objekta uglavnom raspoređene na jednom njegovom dijelu.

Konačno, može se izvesti zaključak da svaka od primijenjenih metoda ima svoje prednosti i nedostatke, te da opravdanost njihove upotrebe zavisi od konkretne primjene. Provedeni eksperimenti sa robotskom rukom su izvršeni samo kako bi se utvrdila robušnost algoritama na različite pojave, ali se sve one ne očekuju u industrijskom okruženju u kom je u konačnici cilj izvršiti praćenje kretanja. Okluzija se u tim slučajevima najčešće ne dešava, te je, s obzirom na dobivene rezultate, najpogodnije praćenje vršiti samom modifikacijom CamShift algoritma ili njegovom fuzijom sa Kalmanovim filterom.

* * *

Analiza rada implementiranih algoritama, u vidu testiranja njihove preciznosti praćenja i vremena izvršavanja, izvršena je u okviru ovog poglavlja. Algoritmi su pokazali dosta dobru robusnost pri različitim uslovima u kojim se prati objekat na video zapisu, a pritom se vremena izvršavanja na nivou jednog frejma ne razlikuju mnogo za SURF algoritam i Kalmanovo filtriranje, dok ekstrakcija teksture zahtijeva nešto duže vrijeme.

Poglavlje 7

Zaključak i smjernice za dalje istraživanje

Digitalna obrada slike i videa ima sve češću primjenu u aplikacijama današnjice, jer omogućava ekstrakciju korisnih informacija i njihovu dalju obradu. U industrijskom okruženju je ovo naročito pogodno, jer omogućava konstantno praćenje rada pojedinih dijelova sistema bez korištenja ljudskih resursa, te obezbeđuje pravovremenu detekciju eventualnih nepravilnosti.

U okviru ovog rada implementiran je algoritam za praćenje objekata na video zapisu korištenjem izotropskih jezgara i metode pomjeranja srednje vrijednosti. Praćenje je izvršeno na temelju informacije o boji objekta. Za razliku od tradicionalnog algoritma, ova implementacija uspješno omogućava ekstrakciju trajektorije kretanja objekata koji sadrže više boja, te adaptivno podešava parametre tona, nijanse i zasićenja na temelju značajki boje objekta, čime se prilagođava različitim tipovima snimaka.

Međutim, praćenje objekta korištenjem samo informacije o njegovoj boji ne omogućava ekstrakciju trajektorije u slučaju pojave okluzije, te je preciznost smanjena ukoliko dolazi do zamagljenja objekta na video zapisu uslijed njegovog brzog kretanja. Zbog toga je implementirani algoritam nadograđen korištenjem tri različite metode: Kalmanovog filtriranja, SURF algoritma i ekstrakcije teksture. Izvršeno je ispitivanje performansi algoritama po pitanju njihove preciznosti i vremena izvršavanja.

Pokazano je da fuzija sa Kalmanovim filterom omogućava poboljšanje performansi u slučaju pravilnih kretanja, kada je ekstrakcija trajektorije na osnovu boje otežana brzim pokretima objekta koji se prati. Kombinacija sa značajkama ekstraktovanim putem SURF algoritma ne daje dobre performanse ukoliko se prati objekat koji nema mnogo karakterističnih tačaka, dok u suprotnom može ostvariti i poboljšanje i pogoršanje performansi, zavisno od distribucije detektovanih ključnih tačaka. Informacija o teksturi uvodi blaga poboljšanja preciznosti ekstrakcije trajektorije, ali na račun velikog vremena izvršavanja algoritma. Kalmanovo filtriranje i SURF metoda u razmatranoj fuziji zahtijevaju približno jednako vrijeme obrade video zapisa.

Povećanje brzine obrade može se izvršiti optimizacijom programskog koda implementiranih metoda. Radi poboljšanja preciznosti ekstrakcije trajektorije, moguće je pratiti nivo skaliranja dimenzija objekta u odnosu na početni. Na taj način se može detektovati parcijalna okluzija i spriječiti translacija stvarne trajektorije. Pored toga, moguće je implementirati Kalmanov filter na složeniji način, gdje će se mijenjati parametri korištenih matrica.

Primjena bilo koje metode pored praćenja zasnovanog na izotropskim jezgrima i *mean-shift* algoritmu ima različito djelovanje nad različitim video zapisima, te je stoga najbolje koristiti implementirane kombinacije metoda u zavisnosti od poznatih informacija o kretanju objekta koji se prati.

Literatura

- [1] da Fona Costa, L., Cesar Jr, R. M., Shape classification and analysis: theory and practice. CRC Press, 2009.
- [2] Lintern, G., Sills, G., “Techniques for automated measurement of floc properties”, Journal of Sedimentary Research, Vol. 76, No. 10, 2006, str. 1183–1195.
- [3] Wallet, B. C., Davogustto, O., “Integrating phase into the visualization of spectral decomposition attributes”, Interpretation, Vol. 3, No. 3, 2015, str. SS73–SS86.
- [4] Tou, J. Y., Lau, P. Y., Tay, Y. H., “Computer vision-based wood recognition system”, in Proceedings of International workshop on advanced image technology, 2007.
- [5] Sharma, K., Kim, S. G., Singh, M. P., “An improved feature matching technique for stereo vision applications with the use of self-organizing map”, International Journal of Precision Engineering and Manufacturing, Vol. 13, No. 8, 2012, str. 1359–1368.
- [6] Haar Romeny, B. M., “The gaussian kernel”, Front-End Vision and Multi-Scale Image Analysis: Multi-Scale Computer Vision Theory and Applications, written in Mathematics, 2003, str. 37–51.
- [7] Yilmaz, A., “Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection”, in Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on. IEEE, 2007, str. 1–6.
- [8] Yao, A., Wang, G., Lin, X., Wang, H., “Kernel based articulated object tracking with scale adaptation and model update”, in Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. IEEE, 2008, str. 945–948.
- [9] Wang, J., Yagi, Y., “Integrating color and shape-texture features for adaptive real-time object tracking”, IEEE Transactions on Image Processing, Vol. 17, No. 2, 2008, str. 235–240.
- [10] Comaniciu, D., Ramesh, V., “Mean shift and optimal prediction for efficient object tracking”, in Image processing, 2000. proceedings. 2000 international conference on, Vol. 3. IEEE, 2000, str. 70–73.
- [11] Maggio, E., Cavallaro, A., Video tracking: theory and practice. John Wiley & Sons, 2011.
- [12] Grzegorz, B., Tomasz, S., Henryk, M., “Improved sum-of-squared-differences tracking algorithm for thermal vision systems”, in Proceedings of SPIE, the International Society for Optical Engineering. Society of Photo-Optical Instrumentation Engineers, 2011.

- [13] Comaniciu, D., Ramesh, V., Meer, P., “Kernel-based object tracking”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 25, No. 5, 2003, str. 564–577.
- [14] Zhang, H., Huang, W., Huang, Z., Li, L., “Affine object tracking with kernel-based spatial-color representation”, in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1. IEEE, 2005, str. 293–300.
- [15] Weng, S.-K., Kuo, C.-M., Tu, S.-K., “Video object tracking using adaptive kalman filter”, Journal of Visual Communication and Image Representation, Vol. 17, No. 6, 2006, str. 1190–1208.
- [16] Babu, R. V., Pérez, P., Bouthemy, P., “Robust tracking with motion estimation and local kernel-based color modeling”, Image and Vision Computing, Vol. 25, No. 8, 2007, str. 1205–1216.
- [17] Yilmaz, A., Li, X., Shah, M., “Contour-based object tracking with occlusion handling in video acquired using mobile cameras”, IEEE transactions on pattern analysis and machine intelligence, Vol. 26, No. 11, 2004, str. 1531–1536.
- [18] Han, B., Comaniciu, D., Zhu, Y., Davis, L. S., “Sequential kernel density approximation and its application to real-time visual tracking”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 30, No. 7, 2008, str. 1186–1197.
- [19] Kim, C., Hwang, J.-N., “Fast and automatic video object segmentation and tracking for content-based applications”, IEEE transactions on circuits and systems for video technology, Vol. 12, No. 2, 2002, str. 122–129.
- [20] Elgammal, A., Harwood, D., Davis, L., “Non-parametric model for background subtraction”, in European conference on computer vision. Springer, 2000, str. 751–767.
- [21] Su, S.-T., Chen, Y.-Y., “Moving object segmentation using improved running gaussian average background model”, in Computing: Techniques and Applications, 2008. DICTA’08. Digital Image. IEEE, 2008, str. 24–31.
- [22] Comaniciu, D., Meer, P., “Mean shift: A robust approach toward feature space analysis”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 24, No. 5, 2002, str. 603–619.
- [23] Lerdsudwichai, C., Abdel-Mottaleb, M., Ansari, A.-N., “Tracking multiple people with recovery from partial and total occlusion”, Pattern Recognition, Vol. 38, No. 7, 2005, str. 1059–1070.
- [24] Zhang, C., Qiao, Y., Fallon, E., Xu, C., “An improved camshift algorithm for target tracking in video surveillance”, 2009.
- [25] Isard, M., Blake, A., “Condensation—conditional density propagation for visual tracking”, International journal of computer vision, Vol. 29, No. 1, 1998, str. 5–28.
- [26] Porikli, F., “Achieving real-time object detection and tracking under extreme conditions”, Journal of Real-Time Image Processing, Vol. 1, No. 1, 2006, str. 33–40.
- [27] Chang, C., Ansari, R., Khokhar, A., “Multiple object tracking with kernel particle filter”, in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1. IEEE, 2005, str. 566–573.

- [28] Ristic, B., Arulampalam, S., Gordon, N., "Beyond the kalman filter", IEEE Aerospace and Electronic Systems Magazine, Vol. 19, No. 7, 2004, str. 37–38.
- [29] Trucco, E., Plakas, K., "Video tracking: a concise survey", IEEE Journal of Oceanic Engineering, Vol. 31, No. 2, 2006, str. 520–529.
- [30] Sonka, M., Hlavac, V., Boyle, R., Image processing, analysis, and machine vision. Cengage Learning, 2014.
- [31] Han, B., Comaniciu, D., Zhu, Y., Davis, L., "Incremental density approximation and kernel-based bayesian filtering for object tracking", in CVPR (1), 2004, str. 638–644.
- [32] Han, B., Zhu, Y., Comaniciu, D., Davis, L., "Kernel-based bayesian filtering for object tracking", in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1. IEEE, 2005, str. 227–234.
- [33] Ali, Z., Hussain, S., Taj, I. A., "Kernel based robust object tracking using model updates and gaussian pyramids", in Emerging Technologies, 2005. Proceedings of the IEEE Symposium on. IEEE, 2005, str. 144–150.
- [34] Yilmaz, A., "Kernel-based object tracking using asymmetric kernels with adaptive scale and orientation selection", Machine Vision and Applications, Vol. 22, No. 2, 2011, str. 255–268.
- [35] Mishra, R., Chouhan, M. K., Nitnawwre, D. D., "Multiple object tracking by kernel based centroid method for improve localization", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 137, 2012.
- [36] Wang, J., Thiesson, B., Xu, Y., Cohen, M., "Image and video segmentation by anisotropic kernel mean shift", in European conference on computer vision. Springer, 2004, str. 238–249.
- [37] Muyun, W., Mingyi, H., Yifan, Z., "An adaptive implementation of the kernel-based object tracking method", in Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on, Vol. 2. IEEE, 2006, str. 354–357.
- [38] Allen, J. G., Xu, R. Y., Jin, J. S., "Object tracking using camshift algorithm and multiple quantized feature spaces", in Proceedings of the Pan-Sydney area workshop on Visual information processing. Australian Computer Society, Inc., 2004, str. 3–7.
- [39] Jatoh, R. K., Shubhra, S., Ejaz, A., "Performance comparison of kalman filter and mean shift algorithm for object tracking", International Journal of Information Engineering and Electronic Business, Vol. 5, No. 5, 2013, str. 17.
- [40] Zhao, J., Liu, W.-J., Sun, H.-J., "A motion tracking method based on kalman filter combined with mean-shift", in Wavelet Analysis and Pattern Recognition, 2008. ICWAPR'08. International Conference on, Vol. 1. IEEE, 2008, str. 91–95.
- [41] Wang, Z., Yang, X., Xu, Y., Yu, S., "Camshift guided particle filter for visual tracking", Pattern Recognition Letters, Vol. 30, No. 4, 2009, str. 407–413.
- [42] Shen, C., Kim, J., Wang, H., "Generalized kernel-based visual tracking", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 20, No. 1, 2010, str. 119–130.

- [43] Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.-M., Hicks, S. L., Torr, P. H., “Struck: Structured output tracking with kernels”, IEEE transactions on pattern analysis and machine intelligence, Vol. 38, No. 10, 2016, str. 2096–2109.
- [44] Sahane, M., Salve, H., Dhawade, N., Bajpai, S., “Visual interpretation of hand gestures for human computer interaction”, Int. Journal of Engineering Research and Applications, Vol. 4, 2014, str. 315–317.
- [45] Kalcounis-Rueppell, M., Parrish, T., Pauli, S., “Application of object tracking in video recordings to the observation of mice in the wild”, in Topics from the 8th Annual UNCG Regional Mathematics and Statistics Conference. Springer, 2013, str. 105–115.
- [46] DeSouza, G. N., Kak, A. C., “A subsumptive, hierarchical, and distributed vision-based architecture for smart robotics”, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 34, No. 5, 2004, str. 1988–2002.
- [47] Dropulić, T., Gongola, Ž., Maras, I., Meštrović, K., Petrić, I., Vidović, M., “Real-time diagnostics of high-voltage circuit-breakers”, in 8. savjetovanje HRO CIGRE, 4.-8. studenoga 2007, Cavtat, 2007.
- [48] Bečirović, V., Bešlija, D., Smajkić, A., Sokić, E., Kapetanović, M., “Sistem za istovremeno mjerjenje više hodograma u kinematskom lancu vn prekidača”, in 12. savjetovanje BH K CIGRE, 4.-8. oktobar 2015, Neum, 2015.
- [49] Benhimane, S., Najafi, H., Grundmann, M., Genc, Y., Navab, N., Malis, E., “Real-time object detection and tracking for industrial applications.”, in VISAPP (2), 2008, str. 337–345.
- [50] Yoon, Y., DeSouza, G. N., Kak, A. C., “Real-time tracking and pose estimation for industrial objects using geometric features”, in Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on, Vol. 3. IEEE, 2003, str. 3473–3478.
- [51] Crivellaro, A., Lepetit, V., “Robust 3d tracking with descriptor fields”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, str. 3414–3421.
- [52] Crivellaro, A., “Robust 3d object pose estimation and tracking from monocular images in industrial environment”, Doktorski rad, Ecole Polytechnique Fédérale de Lausanne, 2016.
- [53] Brun, L., Vento, M., Graph-Based Representations in Pattern Recognition: 5th IAPR International Workshop, GbRPR 2005, Poitiers, France, April 11-13, 2005, Proceedings. Springer Science & Business Media, 2005, Vol. 3434.
- [54] Russ, J. C., The Image processing handbook, seventh edition ed. CRC Press LLC, 2016.
- [55] Gonzalez, W., Woods, R. E., “Eddins, digital image processing using matlab”, Third New Jersey: Prentice Hall, 2004.
- [56] Rafael C. Gonzalez, R. E. W., Digital Image Processing, 3rd ed. Prentice Hall, 2007.
- [57] Lee, E. A., Structure and interpretation of signals and systems. Lee & Seshia, 2011.
- [58] Ibraheem, N. A., Hasan, M. M., Khan, R. Z., Mishra, P. K., “Understanding color models: a review”, ARPN Journal of Science and Technology, Vol. 2, No. 3, 2012, str. 265–275.

- [59] Bunks, C., Kylander, K., “Grokking the gimp”, 2000.
- [60] Vijaya Lakshmi, R., Palanivel, S., “Video based emotion recognition”.
- [61] Emmanuel, W. S., “Diagnose the thyroid using texture characterization and classification”, International Journal, Vol. 4, No. 8, 2014.
- [62] Khan, N. Y., McCane, B., Wyvill, G., “Sift and surf performance evaluation against various image deformations on benchmark dataset”, in Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on. IEEE, 2011, str. 501–506.
- [63] Panchal, P., Panchal, S., Shah, S., “A comparison of sift and surf”, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, No. 2, 2013, str. 323–327.