

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Juvančič

**Zasnova večkamernega sistema za
priznavanje zadetka**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Danijel Skočaj

Ljubljana, 2018

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V zadnjih letih se na področju športa vse bolj uveljavljajo nove tehnologije, ki sodnikom pomagajo pri sprejemanju odločitev. Pri nogometu se tako uporablja sistem za priznavanje zadetka, ki preveri ali je žoga s celotnim obsegom prečkala golovo črto. V diplomski nalogi zasnujete poenostavljen sistem s to funkcionalnostjo. Sestavljen naj bo iz dveh kamer, ki snemata dogajanje ob golovi črti. Sistem naj s tehniko stereo vida oceni položaj žoge in izračuna oddaljenost žoge od golove črte ter s tem ugotovi ali je bil dosežen zadetek ali ne. Sistem tudi ustrezno ovrednotite.

Rad bi se zahvalil vsem, ki ste me na tak ali drugačen način podpirali in spodbujali med pisanjem.

Posebno bi se rad zahvalil mentorju prof. dr. Danijelu Skočaju za njegovo pomoč, predloge, odzivnost in potrpežljivost pri izdelavi diplomskega dela.

Hvala družini (mama, oče, sestra), ki mi zmeraj stoji ob strani in me vedno znova motivira.

Zahvalil bi se tudi Primožu, ki mi je omogočil pisanje diplomske naloge ob delu, Davorju in Nejcu, ki sta z laičnimi predlogi razblinjala dvome in ponostavljala probleme, ki sem si jih sam ustvaril med implementacijo sistema, in Jakobu, ki mi ni dal miru ter neprestano spraševal po napredku.

Hvala tudi Andreji, ki je v tako kratkem času popravila slovnične napake.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Obstoječi sistemi	2
1.3	Definicija problema	2
1.4	Naša rešitev	3
1.5	Struktura diplome	5
2	Obdelava slik	7
2.1	Barvni prostor	7
2.2	Modeliranje ozadja	9
2.3	Segmentacija	16
2.4	Določanje centra objekta	17
3	Stereo rekonstrukcija	19
3.1	Model kamere	19
3.2	Določanje parametrov kamere	24
3.3	Koordinatni sistem	26
3.4	Linearna triangulacija	26
4	Priznavanje zadetka	29
4.1	Ravnina gola	29

4.2	Oddaljenost od ravnine	32
4.3	Odločanje o zadetku	33
5	Rezultati	35
5.1	Postavitev sistema	35
5.2	Reprojekcijska napaka	36
5.3	Natančnost sistema	40
5.4	Kroženje žoge	44
6	Zaključek	47
	Literatura	50

Povzetek

Naslov: Zasnova večkamerne sistema za priznavanje zadetka

Avtor: Jure Juvančič

Pri nogometu prihaja do različnih situacij, v katerih sodniki brez dodatne pomoči težko pravilno raziščejo. V ta namen se čedalje pogosteje uporabljajo različne tehnologije.

V diplomski nalogi z uporabo spletnih kamer implementiramo zasnovo večkamerne sistema za priznavanje zadetka. V laboratoriju pri kontroliranih okoliščinah postavimo sistem s kamerami, ki jih s kalibracijo umestimo v skupen koordinatni sistem. Nato vhodne slike z modeliranjem ozadja segmentiramo in poiščemo središče objekta, ki je v našem primeru rumena žoga. S triangulacijo določimo položaj žoge in golove črte v svetovnem koordinatnem sistemu, vse skupaj izrišemo v 3D prostoru in ocenimo, ali je žoga prešla golovo črto ali ne. Ocenimo tudi natančnost delovanja sistema in napake, ki se pri tem pojavijo. Končna povprečna napaka pri ugotavljanju zadetka je manjša od 1,5 mm, kar je za sistem z neprofesionalnimi kamerami spodbudno.

Ključne besede: segmentacija, modeliranje ozadja, večkamerni sistem, kalibracija kamer, triangulacija, iskanje središča.

Abstract

Title: Design of multi-camera goal detection system

Author: Jure Juvančič

In football, various situations arise in which referees have difficulty making correct judgements without additional assistance. To that end, various technologies are being employed with increasing frequency.

The thesis presents the implementation of a design of a multi-camera goal detection system using online cameras. We set up a system in controlled laboratory circumstances using cameras which are calibrated in a single coordinate system. Then the input images are segmented using background modelling and we search for the centre of the object, in this case a yellow ball. The positions of the ball and the goal line are determined using triangulation in a global coordinate system, which is then converted to 3D space, making it possible to determine whether the ball has crossed the line. We can also assess the accuracy of the operation of the system and any errors that occur. The final average error in detecting goals is less than 1.5 mm, which is encouraging for a system that uses consumer-level cameras.

Keywords: segmentation, background modelling, multi-camera system, calibration of cameras, triangulation, detection of centre.

Poglavje 1

Uvod

1.1 Motivacija

Pri nogometu in vseh drugih športih je pomembno pravično sojenje, ki je v določenih primerih zaradi človeške napake lahko vprašljivo. V zadnjem času se pri nogometu za pomoč pri priznavanju zadetka in ugotavljanju, ali je žoga s celotnim obsegom prečkala golovo črto, čedalje bolj uporablja tehnologija. Mednarodna nogometna federacija (*fr. Fédération Internationale de Football Association*) trenutno dovoljuje uporabo dveh različnih tehnoloških pristopov. Prvi za ugotavljanje veljavnosti zadetka uporablja spremembe v magnetnem polju, pri drugem pa se za to uporabljajo visokokakovostne kamere in računalniški vid. Pri obeh je največja pomanjkljivost visoka cena postavitve sistema, zato se ju uporablja samo na največjih tekmovanjih.

Glavna motivacija za izdelavo diplomske naloge je uveljavljen sistem Hawk-Eye, ki ga je leta 2001 v Veliki Britaniji razvil dr. Paul Hawkings [3]. V splošnem sistem s pomočjo sedmih kamer zajame dogajanje in v realnem času vsako sličico obdela. S pridobljenimi podatki nato izračuna položaj objekta v tridimenzionalnem prostoru. V diplomski nalogi želimo z uporabo dveh cenovno ugodnih spletnih kamer zasnovati podoben sistem za priznavanje zadetka.

1.2 Obstoječi sistemi

Za priznavanje zadetka se trenutno uporabljata dve vrsti tehnologije. Na Danskem so razvili sistem, imenovan GoalRef, ki ga je leta 2011 pod svoje okrilje vzelo podjetje Fraunhofer. GoalRef uporablja posebne vratnice, okoli katerih so nameščene tuljave, ki ustvarjajo nizkofrekvenčno magnetno polje. Sistem predvideva uporabo prilagojene žoge, v kateri je vgrajeno elektronsko vezje, ki okoli žoge z magnetno indukcijo ustvari magnetno polje. S primerjavo in analizo sprememb v magnetnem polju okoli gola nato program lahko določi, ali je bil zadetek pravilno dosežen [14]. Omenjena rešitev ne predvideva uporabe kamer in računalniškega vida, zato se bomo v nadaljevanju posvetili že omenjeni tehnologiji Hawk-Eye [6].

Hawk-Eye se je sprva uveljavil pri kriketu, kjer so se spopadali s podobnim problemom pravičnega dosojanja kot pri nogometu. V osnovi ima sistem okoli igrišča postavljenih šest ali sedem kamer, ki spremljajo dogajanje pod različnimi koti. Posnete sličice program v realnem času obdelava in določi položaj objekta v prostoru, zbrane podatke pa uporabi za predvidevanje krivine leta objekta. Obdelava sličice se začne z identifikacijo slikovnih elementov, ki predstavljajo objekt zanimanja na sliki. Hawk-Eye z metodo za prepoznavanje okroglih regij, ki ustrezajo podani velikosti objekta na sliki, pridobi podatke o koordinatah x in y objekta na vsaki sliki. Algoritem nato podatke o koordinatah iz vsake slike triangulira in izračuna lego objekta v 3D prostoru glede na referenčne točke [6].

Omenjeni sistem se je z določenimi prilagoditvami v zadnjih petnajstih letih uveljavil tudi v nogometu, tenisu, biljardu in badmintonu. Pri nogometu se tako za spremljanje dogajanja okoli enih vrat uporablja sedem visokokakovostnih kamer.

1.3 Definicija problema

Profesionalni sistemi uporabljajo sedem visokokakovostnih kamer za spremljanje dogajanja na posameznem голу, saj morajo delovati v vseh možnih situ-

acijah. Naš sistem bo deloval z uporabo le dveh enostavnih spletnih kamer, zato problem na začetku poenostavimo in predpostavimo:

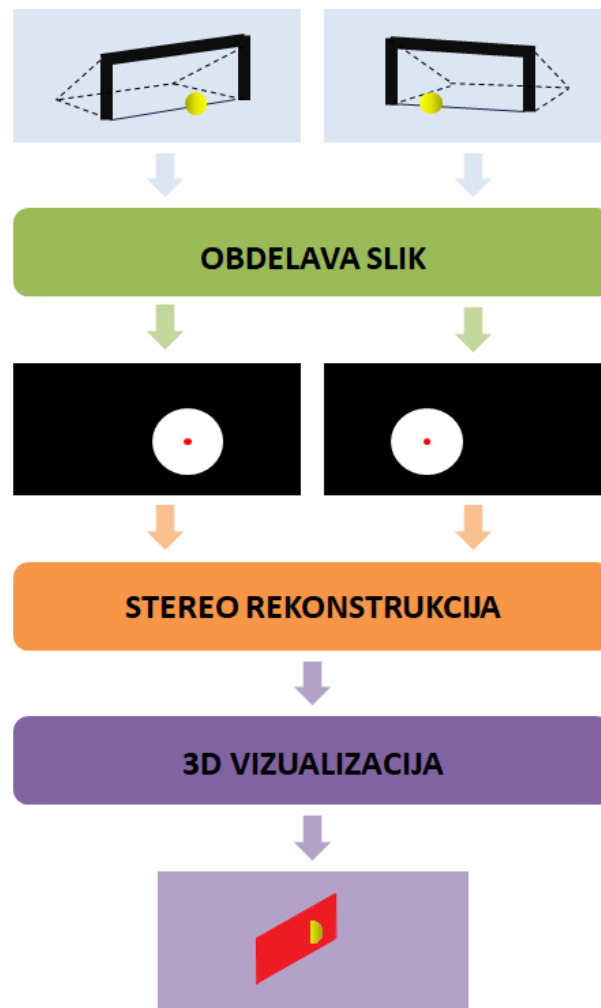
1. žoga je v okolici golove črte v celoti vidna obema kamerama,
2. edini objekt, ki se premika, je žoga,
3. obseg žoge je znan,
4. gibanje žoge je počasno, saj kameri zajemata le 30 sličic na sekundo in bi se ob večji hitrosti gibanja možnost napake povečala,
5. žoga in črta sta različnih barv.

Sistem postavimo v laboratoriju, kjer lahko zmanjšamo vpliv zunanjih dejavnikov in natančno izmerimo natančnost delovanja. Namesto nogometne žoge uporabimo rumeno žogo za tenis in oranžno žogo za skvoš, pri presojanju, ali je žoga prečkala črto, pa upoštevamo nogometna pravila, kar pomeni, da mora žoga črto prečkati s celotnim obsegom.

1.4 Naša rešitev

Za rešitev problema smo zasnovali sistem, ki je shematično predstavljen na sliki 1.1. Sistem kot vhod prejme sličici iz obeh kamer, na katerih poiščemo koordinate središča objekta. Vsako sličico zato najprej obdelamo, da dobimo črno-belo sliko, na kateri je objekt zanimanja predstavljen z belo, ozadje pa s črno barvo. Nato na obdelani sliki poiščemo koordinate središča regije, ki služijo kot vhodni podatek naslednji fazi. Točke iz 2D prostora v stereo rekonstrukciji trianguliramo in izračunamo njihov položaj v 3D prostoru glede na referenčne točke. Na koncu vse skupaj predstavimo v 3D prostoru in ocenimo natančnost delovanja sistema.

Na koncu diplomske naloge želimo imeti sistem, ki bo deloval po prej opisanih prilagojenih pogojih in bi ga z nekaterimi nadgradnjami lahko preizkusili v realnih situacijah pri nogometu. Temu primerno moramo minimizirati



Slika 1.1: Shema naše rešitve

napako in zagotoviti pravilno razsojanje v območju, ki ga dovoljuje mednarodna nogometna federacija, zato mora sistem zaznati zadetek najkasneje, ko je razdalja med žogo in črto 15 mm.

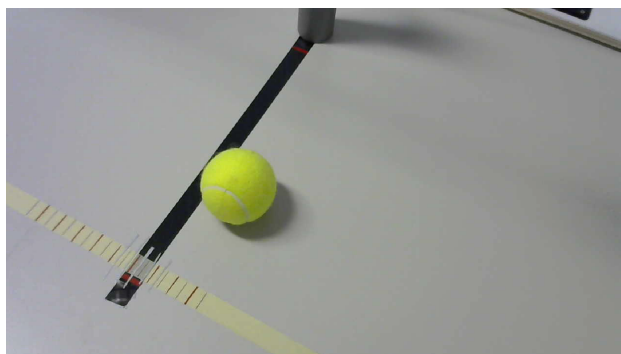
1.5 Struktura diplome

Diploma je razdeljena na šest poglavij v katerih želimo v metodološkem zaporedju opisati delovanje sistema za priznavanje zadetka. V uvodu predstavimo motivacijo za izdelavo naloge skupaj s pregledom obstoječih sistemov in opisom naše rešitve. Sledi poglavje 2, v katerem so teoretično predstavljene uporabljene metode za obdelavo slik. V tretjem poglavju predstavimo model kamere z luknjico, način ocenjevanja njenih parametrov in preslikavo koordinat iz slik v svetovni koordinatni sistem. Sledi poglavje 4, v katerem opišemo postopek priznavanja zadetka in uporabo teorije, opisane v prejšnjih poglavjih. Peto poglavje je namenjeno predstavitvi eksperimentalnih rezultatov, sledi pa mu zadnje poglavje z zaključkom, opisom problemov in predlaganimi izboljšavami.

Poglavje 2

Obdelava slik

Glavni cilj je določiti koordinate objekta v 3D prostoru, za kar moramo najprej na posamezni sliki poiskati objekt in določiti koordinati njegovega središča. Primer slike, na kateri nas zanima središče objekta, prikazuje slika 2.1.

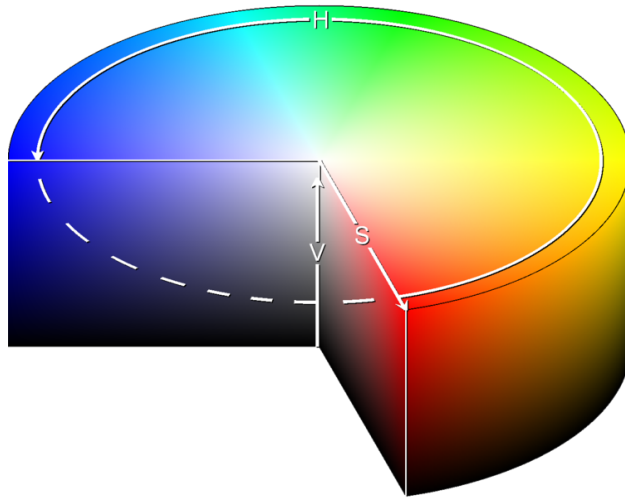


Slika 2.1: Primer slike, na kateri iščemo središče objekta.

2.1 Barvni prostor

Na vsakem paru videosličic nas zanimajo koordinate središča objekta, zato moramo slike, kot je npr. slika 2.1, ustrezno obdelati. Pred postopkom segmentacije originalno sliko, ki je predstavljena v barvnem modelu RGB

(*ang. red, green, blue color mode*), pretvoriti v primernejši model. Pri modelu RGB so barve predstavljene s kombinacijami rdeče, zelene in modre svetlobe, medtem ko pri barvnem prostoru HSL (*ang. Hue, Saturation, Lightness*) in HSV (*ang. Hue, Saturation, Value*) barvo določa komponenta tona (H), na katerega imajo spremembe v nasičenosti (S) in svetlobi manjši vpliv. Predstavitev barvnega modela HSV, ki ga uporabimo v diplomski nalogi,



Slika 2.2: Tridimenzionalna predstavitev barvnega modela HSV [1].

v tridimenzionalnem prostoru si lahko predstavljamo v obliki valja, kot je to prikazano na sliki 2.2. Pri tem navpična os predstavlja intenziteto (*ang. Value – V*) posameznih barv, ki se v krogu prelivajo ena v drugo. To določa komponenta tona (*ang. Hue – H*), ki je definirana kot kót v intervalu $[0, 2\pi]$, pri čemer se rdeča barva začne pri kótu 0 in se preliva v zeleno pri $\frac{2\pi}{3}$, tej pa od $\frac{4\pi}{3}$ naprej sledi modra. Krožno zaporedje zaključi ponovno rdeča barva pri 2π . Tretja komponenta modela HSV je nasičenost (*ang. Saturation – S*), ki jo merimo z radialno razdaljo od osrednje osi in v središču valja zavzame vrednost 0, na zunanji površini pa je njena vrednost 1. V kolikor je vrednost nasičenosti blizu 0, so vsi slikovni elementi, ne glede na izbrani ton, enaki in predstavljajo odtenek sivine. Ko povečujemo nasičenost in se njena vrednost približuje 1, se pričnejo slikovni elementi ločevati in jih tudi vizualno zaznamo

kot barve, ki jih določa njihov ton [13].

V diplomski nalogi smo uporabili barvni model HSV, saj imajo spremembe svetlobe manjši vpliv na komponento tona in barvo kot pri barvnem modelu RGB.

2.2 Modeliranje ozadja

Pred segmentacijo moramo vhodno sliko ustrezno pripraviti in s tem povečati možnost za boljše rezultate. Izberemo tehniko, ki se na področju računalniškega vida pogosto uporablja za zaznavanje premikajočih se objektov s statičnimi kamerami. Modeliranje ozadja (*ang. background modelling/subtraction*) oz. zaznava ospredja (*ang. foreground detection*) je metoda, pri kateri je osnovna ideja, da ospredje (*ang. foreground*) oz. premikajoči se objekt dobimo kot razliko med modelom ozadja (*ang. background model*) in trenutno videosličico [8]. Dober algoritem za posodabljanje ozadja mora biti robusten, kar pomeni, da pravilno loči ozadje od ospredja ne glede na spreminjaje svetlobe, sence objektov in gibanja v naravi (npr. listi). Sliko razlike, ki jo dobimo z odštevanjem trenutne videosličice od modela ozadja, nato v podpoglavju 2.3 segmentiramo. Obstaja več tehnik, s katerimi lahko ozadje modeliramo.

2.2.1 Razlika med zaporednima videosličicama

Najbolj osnovna metoda uporablja razlikovanje med dvema zaporednima sličicama videa (*ang. frame differencing*). Slika ozadja, označena z B , je sličica posneta v času $(t - 1)$, medtem ko je $I(t)$ trenutna sličica videa. Ospredje segmentiramo z enostavno operacijo odštevanja, in sicer vsakemu slikovnemu elementu v $I(t)$ odštejemo istoležeči slikovni element iz B :

$$F(x, y, t) = |I(x, y, t) - B(x, y, t - 1)|, \quad (2.1)$$

kjer je $F(x, y, t)$ nova vrednost slikovnega elementa, x in y spremenljivki lokacije slikovnega elementa in t časovna komponenta (npr. $F(4, 5, 6)$ predstavlja intenziteto slikovnega elementa na lokaciji $(4, 5)$ v času $t = 6$). Za

izboljšanje rezultata določimo prag (*ang. threshold*), po katerem se vrednost slikovnega elementa pretvori v ospredje (1) ali ozadje (0).

$$F(x, y, t) > Threshold \quad (2.2)$$

Rezultat je segmentirana slika s črnim ozadjem in belim objektom, ki smo ga iskali. Opisana tehnika ni najbolj odporna proti spremembam svetlobe, saj predpostavlja, da se vrednosti slikovnih elementov ozadja ne spreminjajo. Poleg tega se objekti z enakomerno porazdeljenimi vrednostmi slikovnih pik napačno interpretirajo kot ozadje.

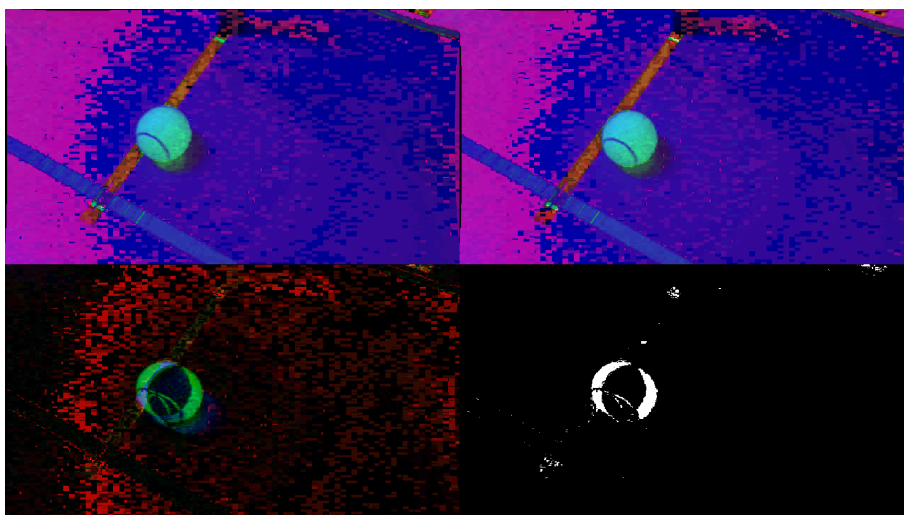
V zgoraj opisanem postopku za model ozadja vzamemo videosličico, posneto pred trenutno, na kateri iščemo objekt. Na sliki 2.3 lahko opazimo, da se ob počasnem premikanju predmeta del tega prekriva s položajem v času $t - 1$, zaradi česar se slikovni elementi v prekrivanju napačno segmentirajo v ozadje. Podobno se ostanki objekta iz modela ozadja, ki se ne prekrivajo, napačno segmentirajo v ospredje. S takim rezultatom segmentacije bomo težko natančno določili center žoge, zato moramo razmisliti, kako bolje modelirati ozadje.

2.2.2 Povprečje vrednosti

Možnost, ki se nam ponuja je uporaba povprečja vrednosti preteklih istoletskih slikovnih elementov (*ang. mean filter*). Model ozadja B predstavljajo srednje vrednosti slikovnih elementov preteklih N slik. B izračunamo kot

$$B(t) = \frac{1}{N} \sum_{i=1}^N I(t - i), \quad (2.3)$$

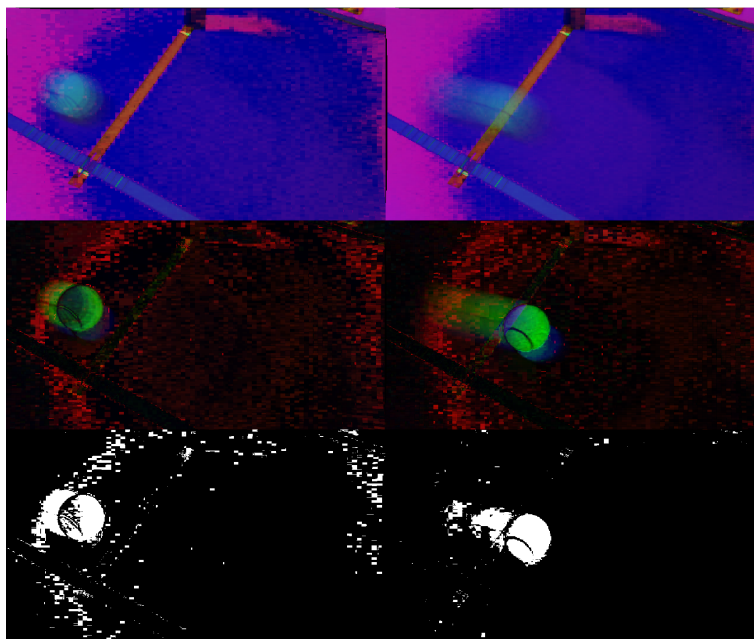
kjer je N odvisen od števila videosličic na sekundo in intenzivnosti gibanja na posnetku, kar je prikazano na sliki 2.4. S posodobljenim modelom ozadja B lahko sedaj izračunamo sliko razlike (*ang. difference image*) tako kot v enačbi (2.1) in apliciramo prag (2.2).



Slika 2.3: Modeliranje z uporabo razlike med zaporednima videosličicama. Levo zgoraj je model ozadja, ki ga predstavlja prejšnja videosličica, desno pa je trenutna videosličica. Levo spodaj je rezultat razlike zgornjih slik. Desno spodaj je segmentirana slika, na kateri se vidi, da se na mestih, kjer se podobne vrednosti intenzitete slikovnih elementov prekrivajo, deli slike napačno pretvorijo v ozadje.

2.2.3 Mediana in približna mediana

Podobno lahko namesto povprečja za računanje B uporabimo mediano (*ang. median filter*), pri tem pa moramo v pomnilniku hraniti N preteklih slik, kjer je N odvisen od hitrosti gibanja objekta na posnetku. Počasnejše kot je gibanje na posnetku, več videosličic moramo hraniti, kar lahko zavzame veliko prostora v pomnilniku. Alternativno uporabi mediane predstavlja tehnika približne mediane (*ang. approximate median*), ki sta jo med raziskavo sledenja prašičem zasnovala N. J. B. McFarlane in C. P. Schofield [10]. Model ozadja izračunamo tako, da vrednosti istoležečih slikovnih elementov $B(x, y)$ in $I(x, y)$ med seboj primerjamo. V kolikor je vrednost slikovnega elementa na trenutni videosličici večja od slikovnega elementa ozadja, vrednost slikovnega elementa ozadja povečamo za 1. Če je vrednost manjša, pa ozadje



Slika 2.4: Levi stolpec prikazuje obdelavo slik pri $N = 5$ in desni pri $N = 19$. Prvi dve sliki prikazujeta model ozadja, dobljenega z računanjem povprečnih vrednosti, drugi pa razliko med modelom ozadja in trenutno videosličico. Spodaj sta prikazani segmentirani sliki, kjer se vidi, da je segmentacija pri $N = 19$ v primerjavi s segmentacijo pri $N = 5$ boljša, vendar še vedno ni zadovoljiva.

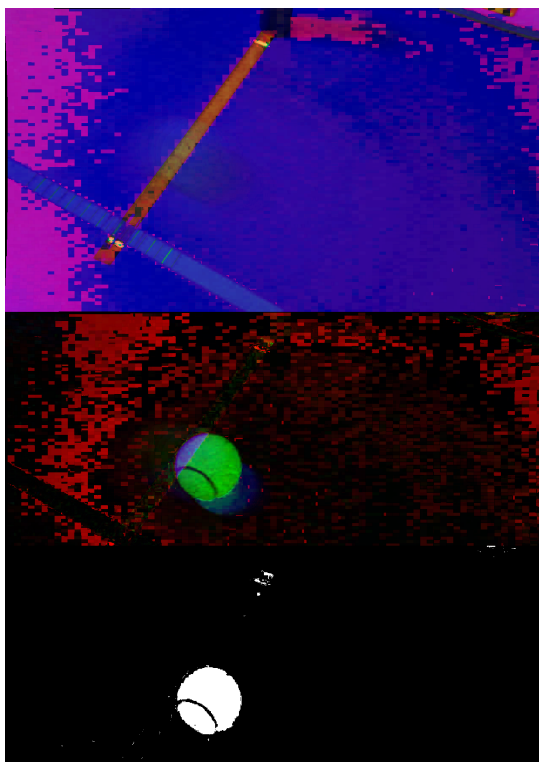
zmanjšamo za 1. Slika 2.5 opisno prikazuje omenjeni postopek računanja približne mediane, slika 2.6 pa dobljen rezultat. Vsak slikovni element v tako pridobljenem modelu ozadja konvergira k neki vrednosti m , od katere je bila polovica posodobljenih slikovnih elementov večja, polovica pa manjša [10].

2.2.4 Uteženo tekoče povprečje

Želimo oblikovati ozadje, na katerega imajo spremembe svetlobe v prostoru čim manjši vpliv, prav tako pa moramo pri posodabljanju ozadja omejiti vpliv slikovnih elementov, ki so bili v bližnji preteklosti prepoznani kot ospredje. Izboljšana metoda povprečja je uteženo tekoče povprečje (*ang. weighted*

za vsak $x = 1$ do dolžina(B)
 za vsak $y = 1$ do dolžina(I)
 če $I(x, y) > B(x, y)$
 potem $B(x, y)++$
 drugače $B(x, y)--$

Slika 2.5: Psevdo koda računanja približne mediane.



Slika 2.6: Rezultat modeliranja ozadja z računanjem približne mediane. Od zgoraj navzdol si sledijo: model ozadja, slika razlike in rezultat segmentacije.

moving mean), ki združuje dve posamezni tehniki. Matematična enačba uteženega povprečja je

$$\mu_t = \frac{\sum_{i=1}^N \omega_i x_i}{\sum_{i=1}^N \omega_i}, \quad (2.4)$$

kjer so x_i podatki in ω_i njihove uteži. Podatki z višjimi utežmi torej bolj vplivajo na povprečje od podatkov z nižjimi utežmi. Če bi za posodabljanje

ozadja uporabili enačbo (2.4), bi povprečje v času t računali z vrednostmi slikovnih elementov vseh prebranih slik do tega trenutka. Ker pa želimo, da po določenem času model začne pozabljati stare vrednosti in s tem še povečati vpliv novih vrednosti, določimo razpon n slik, na katerih izvajamo računanje. Povprečje namesto iz vseh izračunamo samo iz preteklih n slik, pri tem pa nove vrednosti prištejemo vsoti, stare pa odštejemo. S tem se izognemo nepotrebnemu računanju celotne vsote vseh n slik.

$$\mu_t = \mu_{t-1} + \frac{I_t}{n} - \frac{I_{N-(n-1)}}{n}, \quad (2.5)$$

kjer je N število prebranih videosličic v času t in n velikost razpona oz. število zaporednih slik, za katere želimo, da vplivajo na model ozadja.

Iz poznavanja enačb (2.4) in (2.5) želimo dobiti rešitev za naš problem, vendar se pojavi problem določanja uteži. Odločimo se, da uteži slikovnih elementov sproti spreminjamo, in sicer za vsak slikovni element vodimo podatke o tem, kolikokrat se je pretvoril v ozadje. Določimo matriko N , v kateri hranimo števce.

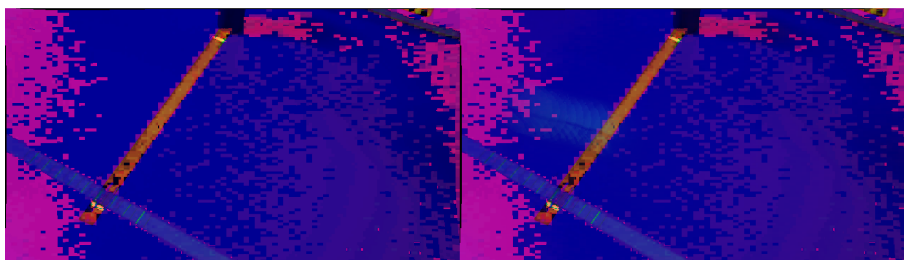
$$N_t = \begin{bmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1i} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2i} \\ \dots & \dots & \dots & \dots & \dots \\ k_{j1} & k_{j2} & k_{j3} & \dots & k_{di} \end{bmatrix}, \quad (2.6)$$

kjer je k_{ij} števec za slikovni element v $B(i, j)$, matrika N_t pa je po velikosti enaka matriki B .

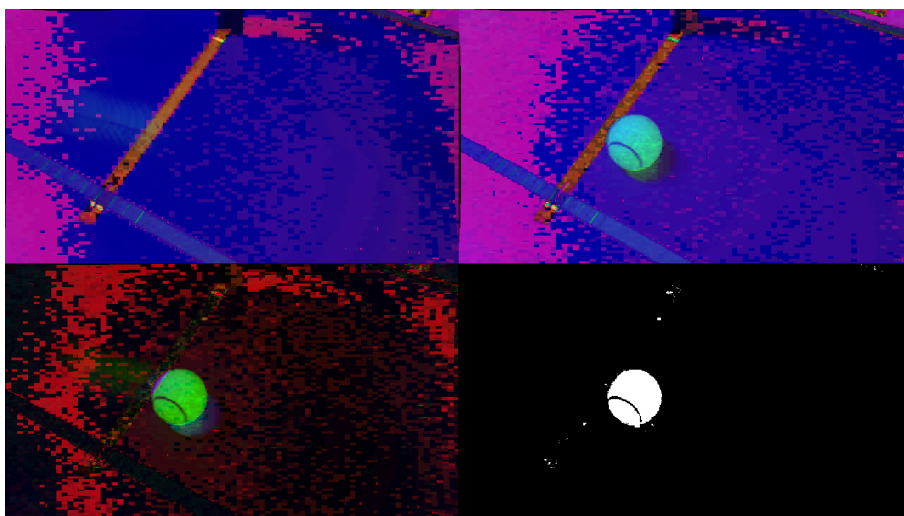
V primeru, ko je slikovni element p_{ij} del ozadja, k_{ij} povečamo in tako slikovni element pridobi pomembnost. Slikovnim elementom, ki so del ospredja, števce zmanjšamo in s tem poskrbimo, da imajo manjši vpliv na referenčno sliko. Prav tako utežimo slikovne elemente trenutne slike s parametrom α , ki ga določimo glede na razpon n , in sicer $\alpha = \frac{1}{n}$. Model ozadja v diplomski nalogi tako posodabljam po enačbi

$$B(t) = \frac{N_t * B_{t-1} + \alpha * I_t}{N_t + \alpha}. \quad (2.7)$$

Primer modela ozadja, dobljenega po enačbi (2.7), je upodobljen na sliki 2.7, medtem ko slika 2.8 prikazuje končni rezultat uporabe metode modeliranja ozadja.



Slika 2.7: Primer začetnega modela ozadja (levo) in model ozadja v času $t = 20$ pri vrednosti parametra $\alpha = \frac{1}{20}$ v barvnem prostoru HSV.



Slika 2.8: Končni rezultat modeliranja ozadja z uporabo tekočega povprečja. Slika levo zgoraj prikazuje model ozadja, desno od nje je prikazana trenutna videoslička, spodaj pa si od leve proti desni sledita slika razlike in rezultat po segmentaciji slike razlike.

2.3 Segmentacija

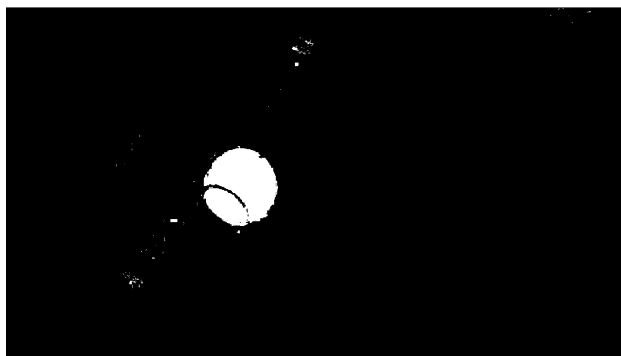
Segmentacija slike (*ang. image segmentation*) je postopek razčlenitve slike v skupine slikovnih elementov s podobnimi lastnostmi (npr. barva, intenziteta ...). Cilj procesa je ločiti ozadje od objektov, da jih lahko potem lažje analiziramo.

Poznamo več pristopov, s katerimi lahko pridemo do segmentirane slike, na kateri je ozadje predstavljeno z vrednostjo 0, objekt oz. regija zanimanja pa z 1. Zelo razširjena tehnika za segmentacijo sivinskih slik je določanje vrednosti praga (*ang. threshold*) na podlagi histograma slike (*ang. histogram thresholding method*), v katerem je vsak slikovni element razvrščen v ustrezno tonsko vrednost [5]. Omenjena tehnika razdeli sliko na dva razreda, svetlejšo ospredje in temnejše ozadje, kot je to prikazano na sliki 2.9. Prag lahko določimo tudi na podlagi večjih skupin slikovnih elementov. Taki tehniki pravimo segmentacija na podlagi regije (*ang. region based segmentation*) [5], ki slikovne elemente razdeli v različne skupine po izbrani metodi. Tretji možen pristop je z zaznavanjem robov (*ang. edge based segmentation*) [5], pri katerem je v prvem koraku potrebno zaznati robove in jih nato v drugem pravilno povezati in ustvariti meje regij.

Za primere v diplomski nalogi uporabimo segmentacijsko tehniko, ki temelji na histogramu slike. Na to, kako slikovne elemente razvrstimo, torej vpliva vrednost praga po enačbi (2.2). Prag (*v nadaljevanju Th*) je številka vrednost, s katero primerjamo intenziteto vsakega slikovnega elementa. V kolikor je ta nižja od Th , se slikovnemu elementu dodeli vrednost 0, v nasprotnem primeru dobi slikovni element vrednost 1. Vprašanje, ki se pri tem načinu segmentacije pojavlja, je, kako izbrati ustrezen prag? To lahko storimo globalno, kjer vsem slikovnim elementom na sliki določimo enako vrednost Th , ali lokalno, pri čemer vsakemu slikovnemu elementu določimo svojo vrednost.

Uporabimo globalno metodo, ki jo je opisal Nobuyuki Otsu [11]. Ta predvideva, da je slika bimodalna, kar pomeni, da jo sestavljata dva razreda slikovnih elementov, in sicer slikovni elementi ospredja ter ozadja. Histo-

gram take slike ima slikovne elemente razporejene okoli dveh vrhov ločenih z globoko dolino. Metoda določi ustrezen prag tako, da minimizira uteženo varianco znotraj razreda oz. maksimizira varianco med razredi.



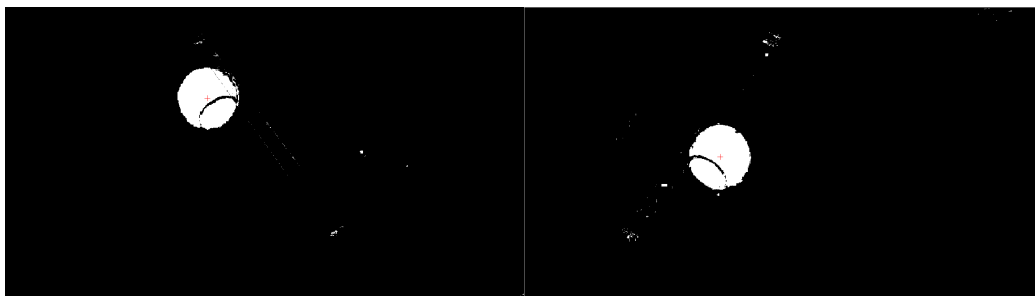
Slika 2.9: Slika 2.1 po segmentaciji.

2.4 Določanje centra objekta

Po segmentaciji razpolagamo z binarno sliko, na kateri moramo poiskati koordinate središča bele regije. Ker poznamo premer objekta, se nam ni treba osredotočiti na iskanje centra kroga. Poiščemo lahko centre vseh regij, ki so podobne krogu in ustrezajo podanemu premeru objekta. Za vsako regijo izračunamo njeno širino, dolžino in površino ter s pomočjo teh lastnosti izberemo središče najustreznejše. Ker je segmentacija dovolj dobra, lahko iščemo središče bele regije z največjo površino. Na sliki 2.10 označimo zaznano središče objekta na levem in desnem pogledu.

Iskanja bi se lahko lotili tudi s Houghovo transformacijo (*ang. Circular Hough Transform (CHT)*) [2], vendar je pri uporabi prihajalo do prevelikih odstopanj. Regija ima v nekaterih primerih obliko elipse, zaradi česar metoda *CHT* od primera do primera različno določa središče. Na videosličici v času t_0 metoda krožnico prilagodi zgornjemu robu elipse in tako zaznано središče zamakne v smeri proti zgornjemu robu. V času t_1 lahko ista metoda krožnico prilagodi spodnjemu robu in zaznано središče premakne nižje. Zaradi teh

napak ne razpolagamo z ustreznimi korespondenčnimi pari in posledično je center v 3D svetu preveč odstopal.



(a)

(b)

Slika 2.10: Po segmentaciji na sliki 2.9 in korespondenčni sliki iz desnega pogleda, poiščemo središče (označeno z rdečo barvo) največje bele regije.

Poglavje 3

Stereo rekonstrukcija

Stereo vid (*ang. stereo vision*) je proces, s katerim iz dveh ali več 2D pogledov pridobimo 3D informacijo. Standardni stereo pogled tvorita dve kameri, ki sta vodoravno ločeni in zajemata dogajanje iz različnih zornih kotov. Iz para slik, ki ga imenujemo tudi stereo par, lahko nato z ujemanjem korespondenčnih točk ocenimo relativno globino prostora in jo predstavimo z disparitetno sliko (*ang. disparity map*), iz katere lahko izračunamo absolutne koordinate točk v prostoru.

Pri stereo rekonstrukciji želimo torej za center objekta, ki smo ga v razdelku 2.4 detektirali, poiskati svetovne koordinate.

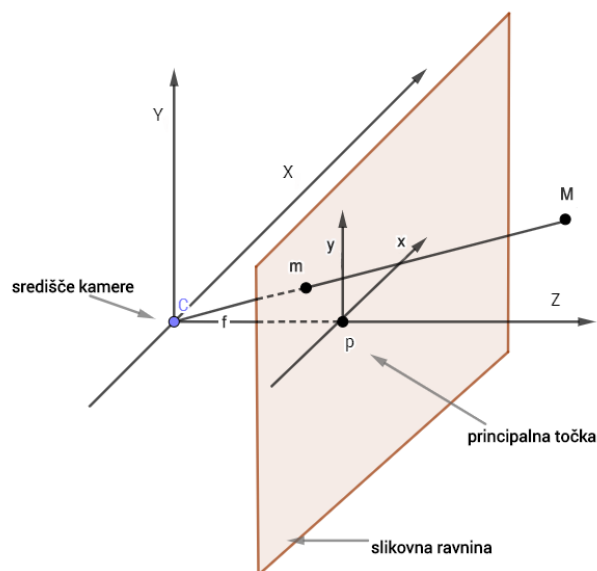
3.1 Model kamere

Kamera je optična naprava, ki 3D točko iz sveta preslika na 2D sliko. Opisali bomo model kamere z luknjico (*ang. pinhole camera*), ki opiše matematično relacijo med točko v svetu in njeno preslikavo na slikovno ravnino (*ang. image plane*). Kamere se med seboj razlikujejo, vendar lahko vsako predstavimo z določenimi lastnostmi, kot sta npr. goriščna razdalja in optični center kamere. Tem lastnostim kamere pravimo notranji parametri (*ang. intrinsic parameters*). Poleg notranjih parametrov potrebujemo tudi podatke o legi kamere v svetu in to opisujejo zunanji parametri (*ang. extrinsic paramete-*

ters) [16].

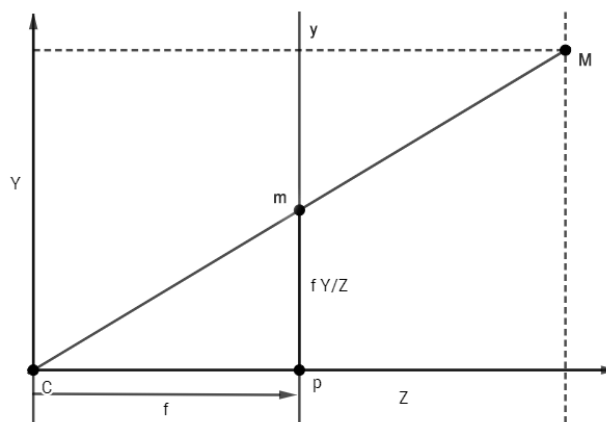
3.1.1 Notranji parametri

Z notranjimi parametri torej opišemo interne lastnosti kamere in projekcijo točke iz sveta na slikovno oz. goriščno ravnino (*ang. focal plane*). V center kamere postavimo izhodišče koordinatnega sistema, od katerega os Z poteka v smeri odprtine kamere. Na omenjeni slikovni ravnini imajo tako točke koordinato Z enako goriščni razdalji (glej sliko 3.1).



Slika 3.1: Izhodišče koordinatnega sistema je v središču kamere C , os Z poteka skozi principalno točko p , ki je od centra oddaljena za goriščno razdaljo f . Točka M se iz sveta preslika v točko m na slikovni ravnini, ki je postavljena pred kamero in je od središča kamere oddaljena za f [7].

Točka v prostoru $M = (X, Y, Z)$ se preslika v točko m na presečišču črte, ki poteka od M do izhodišča koordinatnega sistema C , in goriščne ravnine [7]. Na sliki 3.2 vidimo dva podobna trikotnika in z uporabo izrekov o podobnosti



Slika 3.2: Podobna pravokotna trikotnika [7].

trikotnikov lahko izrazimo kartezični koordinati točke m

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{fX}{Z} \\ \frac{fY}{Z} \end{bmatrix}. \quad (3.1)$$

Zavedati se moramo, da izraz (3.1) predvideva, da je koordinatno središče slikovne ravnine v principalni točki $p(p_x, p_y)$, torej v sredini. Koordinatno izhodišče slik v računalništvu se ponavadi nahaja v zgornjem levem kotu, zato moramo s premikom za p_x in p_y prestaviti izhodišče. Tako dobimo koordinati $(\frac{fX}{Z} + p_x, \frac{fY}{Z} + p_y)$ [7]. Premik za nekaj časa pustimo ob strani, k njemu se bomo ponovno vrnili ob končnem modelu kamere.

Za lažjo predstavitev lahko točko m zapišemo s homogenimi koordinatami, s katerimi lahko točke v neskončnosti opišemo s končnimi koordinatami. Za homogene koordinate velja, da niso fiksno določene s številskimi vrednostmi in jih lahko pomnožimo z neničelnim faktorjem, dobljeni rezultat pa predstavlja isto točko. Točki m torej dodelimo homogene koordinate

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix}. \quad (3.2)$$

Projekcijo točke v prostoru, predstavljeno s homogenimi koordinatami, lahko zapišemo kot matrično množenje s 3×3 diagonalno matriko $(f, f, 1)$

in 3×4 matriko $[I|0]$. Točka M je sedaj zapisana kot homogen vektor $\begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ in tako izrazimo točko m kot

$$m = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (3.3)$$

kjer leva matrika predstavlja osnovno kalibracijsko matriko kamere K . Zapis lahko poenostavimo $m = K\vec{M}$ [7, 15].

Model predpostavlja, da so koordinate slikovnih elementov na sliki v evklidskem prostoru z enakim merilom na obeh oseh koordinatnega sistema. Pri nekaterih kamerah ta predpostavka ne drži, zato moramo matriko K z leve pomnožiti z diagonalno matriko $(m_x, m_y, 1)$ in tako dobimo

$$K = \begin{bmatrix} fm_x & 0 & p_x m_x & 0 \\ 0 & fm_y & p_y m_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (3.4)$$

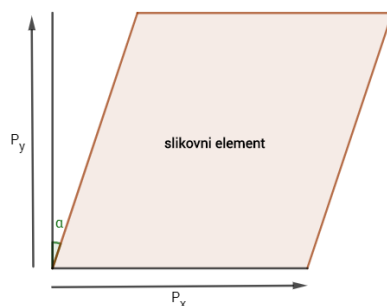
kjer sta m_x in m_y velikost slike, p_x in p_y pa prej omenjeni premik [7].

Poleg omenjene kalibracijske matrike pa v skupino notranjih parametrov spadajo tudi faktorji, ki vplivajo na popačenost slike oz. jo odpravljajo. Prvi izmed teh je poševnost slikovnega elementa s (*ang. skew*), ki slikovni element spremeni v pravilen kvadrat. Primer poševnosti slikovnega elementa prikazuje slika 3.3. Vpeljemo ga v matriko K , tako da je prva vrstica matrike $\begin{bmatrix} fm_x & s & p_x m_x & 0 \end{bmatrix}$.

Kalibracijska matrika, dobljena po modelu kamere z luknjico, ne upošteva popačenosti slike, ki nastane zaradi leče kamere, saj popolna taka kamera nima leče. Zato moramo v model vključiti tudi tangencialno in radialno popačenost.

Do tangencialne popačenosti pride zaradi nepravilnosti sensorja z lečo. Če senzor kamere ni vzporeden z lečo, moramo to popačenost upoštevati: Upoštevamo jo

$$\begin{aligned} x_{popacen} &= x + (2t_1xy + t_2(r^2 + 2x^2)) \\ y_{popacen} &= y + (t_1(r^2 + 2y^2) + 2t_2xy), \end{aligned} \quad (3.5)$$



Slika 3.3: Faktor poševnosti poravnava x in y os slikovnega elementa, tako da sta pravokotni ena na drugo, $s = f_y \tan \alpha$.

kjer sta x in y normalizirani koordinati slikovnega elementa, t_1 in t_2 koeficienta tangencialne popačenosti leče in $r^2 = x^2 + y^2$ [15, 9].

Radialno popačenost povzroči leča kamere in je odvisna od goriščne razdalje. Obstajajo tri vrste popačenosti, in sicer sodčasta, blazinasta ter brkasta. Upoštevamo jo

$$\begin{aligned} x_{popacen} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{popacen} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \end{aligned} \quad (3.6)$$

kjer sta x in y normalizirani koordinati slikovnega elementa, k_1 in k_2 koeficienta radialne popačenosti leče in $r^2 = x^2 + y^2$ [15, 9].

Če združimo vse zgoraj zapisano, lahko sedaj zapišemo kalibracijsko matriko kamere K in koeficiente radialne ter tangencialne popačenosti.

$$\begin{aligned} K &= \begin{bmatrix} fm_x & s & p_x m_x \\ 0 & fm_y & p_y m_y \\ 0 & 0 & 1 \end{bmatrix} \\ \vec{t} &= \begin{bmatrix} t_1 & t_2 \end{bmatrix} \\ \vec{r} &= \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \end{aligned} \quad (3.7)$$

3.1.2 Zunanji parametri

Če notranji parametri predstavljajo lastnosti kamere, potem zunanji opisujejo položaj in orientacijo sveta glede na kamero. Na sliki 3.1 smo pokazali,

da os z v koordinatnem sistemu kamere poteka naravnost skozi luknjico v svet. V svetovnem koordinatnem sistemu pa z ne poteka v isti smeri, zato je potrebna rotacija s 3×3 rotacijsko matriko R . Množenje z matriko R , ki predstavlja orientacijo koordinatnega sistema kamere, poravnava osi obeh sistemov.

Poleg rotacije je treba poravnati tudi obe izhodišči, premik določa translacijski vektor $\vec{T} = (T_1, T_2, T_3)^T$. S premikom in rotacijo lahko dobimo poljuben koordinatni sistem.

Premik točke v svetu M , zapisane kot homogen vektor $(X, Y, Z, 1)^T$, je torej

$$\vec{M}_p = \begin{bmatrix} R & \vec{T} \end{bmatrix} \vec{M}. \quad (3.8)$$

Položaj kamere lahko izrazimo s svetovnimi koordinatami $K_n = -R^{-1}\vec{T} = -R^T\vec{T}$.

3.1.3 Končni model

Sedaj lahko združimo notranje in zunanje parametre, da dobimo končni model kamere in preslikavo točke iz sveta

$$\begin{aligned} m &= K \begin{bmatrix} R & \vec{T} \end{bmatrix} \vec{M} \\ \vec{t} &= \begin{bmatrix} t_1 & t_2 \end{bmatrix} \\ \vec{r} &= \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}. \end{aligned} \quad (3.9)$$

3.2 Določanje parametrov kamere

Obstaja več različnih tehnik za določanje notranjih in zunanjih parametrov, najbolj znane pa so Zhangova fleksibilna metoda za kalibracijo kamer [15], Tsaijev kalibracijski algoritem [9] in Selbyjeva metoda [12], ki se uporablja predvsem za samodejno kalibriranje rentgenskih kamer. V diplomski nalogi parametre določimo z orodjem za kalibracijo stereo sistema v MATLAB aplikaciji, ki uporablja Zhangovo fleksibilno tehniko [15].

Iz dveh pogledov želimo torej natančno določiti položaj objekta, zato moramo oceniti notranje in zunanje parametre vsake kamere ter njuno lego v prostoru. V orodje za kalibracijo naložimo približno dvajset stereo parov slik, podobnih kot na sliki 3.4, na katerih je jasno vidna kalibracijska šahovnica. Pred kalibracijo izberemo izračun poševnosti slikovnega elementa (s) in koeficente radialne ter tangencialne popačenosti. Po končani kalibraciji nam orodje za kalibracijo vrne parametre obeh kamer z izhodiščem koordinatnega sistema v središču kamere 1 [4].



Slika 3.4: Primer para slik, ki ga uvozimo v MATLAB.

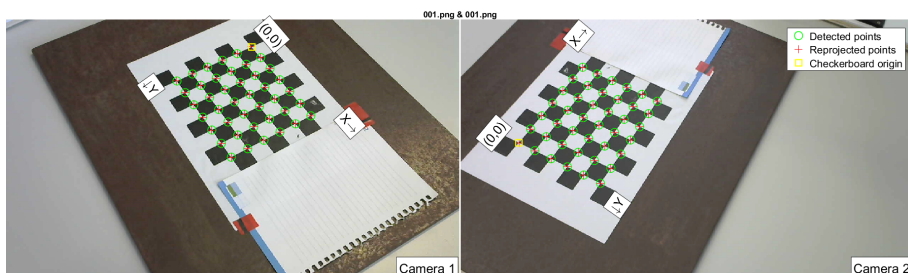
V kolikor je napaka pri kalibraciji prevelika, lahko posnamemo nove ali pa izmed obstoječih odstranimo neustrezne pare slik. Izračunane notranje parametre kamere 1 prikazuje enačba (3.10).

$$\begin{aligned} K_1 &= \begin{bmatrix} 1388,7160 & 0,8848 & 652,3279 \\ 0 & 1400,8167 & 398,5548 \\ 0 & 0 & 1 \end{bmatrix} \\ \vec{r}_1 &= [0,0088 \quad 0,8063 \quad -3,6933] \\ \vec{t}_1 &= [-0,0013 \quad -0,0028] \end{aligned} \quad (3.10)$$

Orodje za kalibracijo nam za vsak par slik vrne pripadajoče zunanje parametre, zato moramo ročno izbrati kalibracijski par, s katerim določimo želeni koordinatni sistem.

3.3 Koordinatni sistem

Ker želimo imeti opravka s svetovnimi koordinatami, moramo koordinatni sistem ustrezno obrniti in premakniti. Za vsak par kalibracijskih slik razpolagamo z zunanjimi parametri obeh kamer, zato izberemo sliko šahovnice 3.5, ki določa ustrezen koordinatni sistem.



Slika 3.5: Z rumeno barvo označeno izhodišče izbranega koordinatnega sistema.

Naj bosta R_1 in R_2 rotacijski matriki, \vec{T}_1 in \vec{T}_2 pa translacijska vektorja, izračunana glede na kalibracijski par iz slike 3.4. Notranji parametri kamer so predstavljeni z matrikama K_1 in K_2 . Vsaki kameri določimo projekcijsko matriko P , ki ustrezno poenoti koordinatna sistema in ju postavi v prostor.

$$\begin{aligned} P_1 &= K_1 \begin{bmatrix} R_1 | \vec{T}_1^T \end{bmatrix} \\ P_2 &= K_2 \begin{bmatrix} R_2 | \vec{T}_2^T \end{bmatrix} \end{aligned} \quad (3.11)$$

3.4 Linearna triangulacija

Triangulacija je funkcija, ki izmerjenim točkam m_1 in m_2 na obeh slikah izračuna položaj v 3D prostoru. Če torej poznamo matriki obeh kamer in korespondenčni točki na slikah, lahko funkcijo triangulacije zapišemo kot

$$M = \tau(m_1, m_2, P_1, P_2). \quad (3.12)$$

V podpoglavju 3.1 smo omenili več vrsti popačenosti, zaradi česar pri projekciji točke na obe slikovni ravnini pride do napak. Popačenost povzroči,

da korespondenčni točki ne ležita točno na epipolarnih črtah, torej v svetu ne bo obstajala točka M , ki bi popolnoma ustrezala enačbama [7]

$$\begin{aligned} m_1 &= P_1 M \\ m_2 &= P_2 M. \end{aligned} \tag{3.13}$$

Predpostavimo, da poznamo meritve iz enačbe (3.13). Ker imamo opravka s homogenimi koordinatami, se moramo najprej znebiti neničelnega faktorja λ , kar dosežemo z vektorskim produktom [7]. S tem dobimo tri linearne enačbe, od katerih sta dve linearno neodvisni. Splošen primer za $m \times PM = 0$

$$\begin{aligned} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \times \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} &= 0 \\ \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \times \begin{bmatrix} p_{11}X + p_{12}Y + p_{13}Z + p_{14} \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} \\ p_{31}X + p_{32}Y + p_{33}Z + p_{34} \end{bmatrix} &= 0. \end{aligned} \tag{3.14}$$

Matriko na desni strani produkta lahko krajše zapišemo kot

$$\begin{bmatrix} P_1 M \\ P_2 M \\ P_3 M \end{bmatrix}, \tag{3.15}$$

kjer P_k predstavlja k -to vrstico v matriki P [7]. Z izračunom vektorskega produkta dobimo tri linearne enačbe, od tega sta dve med seboj linearno neodvisni [7]

$$\begin{aligned} yP_3M - P_2M &= 0 \\ P_1M - xP_3M &= 0 \\ xP_2M - xP_1M &= 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} xP_3M - P_1M &= 0 \\ yP_3M - P_2M &= 0. \end{aligned} \tag{3.16}$$

Enako naredimo tudi s podatki na drugem pogledu in tako dobimo dva para linearno neodvisnih enačb. Zapišemo jih lahko v obliko enačbe $AM =$

0 [7], pri čemer izrazimo matriko A kot

$$A = \begin{bmatrix} xP_3 - P_1 \\ yP_3 - P_2 \\ x'P'_3 - P'_1 \\ y'P'_3 - P'_2 \end{bmatrix}, \quad (3.17)$$

kjer P_k predstavlja k -to vrstico projekcijske matrike kamere 1, P'_k k -to vrstico projekcijske matrike kamere 2, x in y koordinati točke m_1 ter x' in y' koordinati točke m_2 .

Sistem enačb je rešljiv z direktno linearno transformacijo, ki rešitev poišče z razcepom matrike na singularne vrednosti. Rešitev ustreza najmanjši singularni vrednosti matrike A , ki je dobljena po metodi najmanjših kvadratov [7]. Na ta način dobimo vektor \vec{M} oziroma 3D koordinate točke v svetovnem koordinatnem sistemu.

Poglavje 4

Priznavanje zadetka

V tem poglavju opišemo način, s katerim ugotavljamo, ali je žoga prečkala golovo črto s celotnim obsegom. Ker bomo računali oddaljenost centra žoge od ravnine gola, moramo najprej poiskati enačbo ustrezne ravnine.

4.1 Ravnina gola

Pri ugotavljanju, ali je objekt prečkal črto, bomo računali oddaljenost centra objekta od ravnine z upoštevanjem polmera. Potrebujemo torej enačbo ravnine v 3D prostoru, ki jo določa notranja stran golove črte.

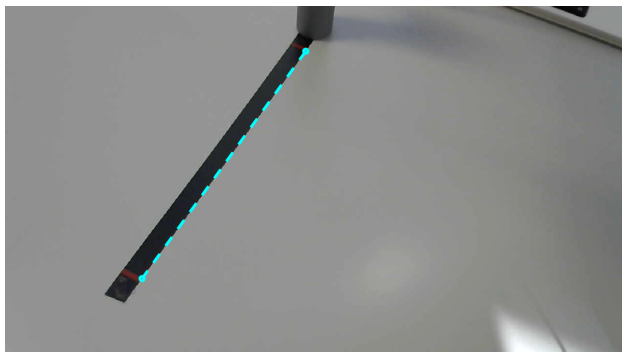
4.1.1 Določitev črte v 2D prostoru

Golova črta je na sliki predstavljena z daljico, ki jo na obeh straneh omejujeta vratnici. Obe stičišči vratnice in črte v splošnem označimo s točkama A in B . Položaj daljice na obeh pogledih nam je poznan in se zaradi statične postavitve kamer ne spreminja, zato lahko v fazi priprave sistema ročno določimo črti L_1 L_2

$$\begin{aligned} L_1(A_1, B_1) \\ L_2(A_2, B_2), \end{aligned} \tag{4.1}$$

kjer je A_1 stičišče na sliki kamere 1 in A_2 stičišče na sliki kamere 2. Enako velja za točki B_1 in B_2 .

Pri določanju stičišč moramo biti natančni in zagotoviti, da točki, ki predstavljata isto stičišče na različnih pogledih, predstavljata isto točko v realnem svetu. Zato pri postavljanju sistema na lepilnem traku z markerjem označimo točki A in B . Izbrana daljica je prikazana na sliki 4.1.



Slika 4.1: Označena črta, za katero nas zanima enačba ravnine.

4.1.2 Od črte do ravnine v 3D prostoru

Ravnina je dvodimenzionalna ploskev v trirazsežnem prostoru in jo med drugim natanko določajo tri nekolinearne točke. Zapišemo jo z enačbo

$$n_x x + n_y y + n_z z = d, \quad (4.2)$$

kjer je $\vec{n} = (n_x, n_y, n_z)$ neničelni vektor pravokoten na ravnino, ki ga imenujemo normala ravnine.

Definirana imamo dva para točk, ki ju z uporabo triangulacije preslikamo v svetovne koordinate. Naj velja, da sta $A(A_x, A_y, A_z)$ in $B(B_x, B_y, B_z)$ točki, ki omejujeta črto v svetovnem koordinatnem sistemu. Za določitev ravnine potrebujemo še tretjo točko C , ki ne leži na daljici AB . Točki A in B sta opisani s svetovnimi koordinatami v koordinatnem sistemu, ki smo ga določili v razdelku 3.3. Ker smo si želeni koordinatni sistem izbrali, poznamo orientacijo koordinatne osi z , ki poteka pravokotno na mizo. S pomočjo te

predpostavke lahko točko C izračunamo

$$\begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \lambda \end{bmatrix}, \quad (4.3)$$

kjer je $\lambda \neq 0$.

Iz treh točk določimo vektorja \vec{AB} in \vec{CB}

$$\begin{aligned} \vec{AB} &= \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} - \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \\ \vec{CB} &= \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} - \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}, \end{aligned} \quad (4.4)$$

v nadaljevanju $\vec{AB} = \vec{u}$ in $\vec{CB} = \vec{v}$.

Potrebujemo enačbo ravnine, katere normalni vektor je pravokoten na oba vektorja \vec{u} in \vec{v} . Izkoristimo lastnost vektorskega produkta, za katerega velja, da je rezultat produkta med dvema linearno neodvisnima vektorjema $a \times b$ vektor, ki je pravokoten na oba.

Dva vektorja sta linearno neodvisna, če ju med sabo ne moremo izraziti z drugim, npr. $\vec{a} = k\vec{b}$, $k \in \mathbb{R}$, kar je natanko takrat, ko sta nekolinearna. Normalno ravnine n zapišemo kot

$$\vec{n} = \vec{u} \times \vec{v} \Rightarrow \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix}. \quad (4.5)$$

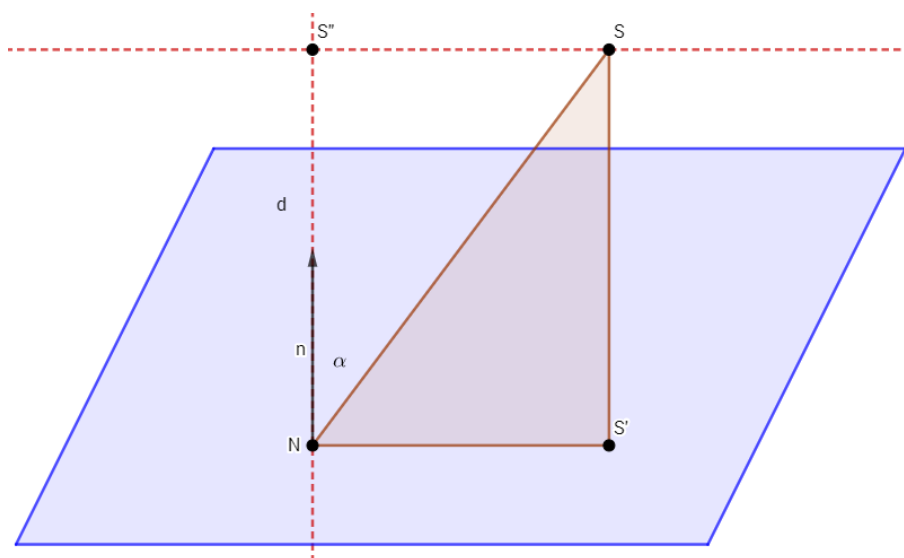
Poznamo torej tri točke na ravnini in normalni vektor, potrebujemo še parameter d , ki ga enostavno izračunamo, če vstavimo izračunane vrednosti

$$n_x A_x + n_y A_y + n_z A_z = d. \quad (4.6)$$

Enačbo ravnine bomo potrebovali pri ugotavljanja položaja žoge glede na golovo črto.

4.2 Oddaljenost od ravnine

Če je bila žoga na slikah detektirana, je treba ugotoviti njen položaj v svetu glede na golovo črto. V poglavju 3 smo opisali enačbo ravnine, iz segmen-tiranih slik ocenili središče bele regije na obeh pogledih in s triangulacijo projicirali položaj središča žoge v 3D svetovne koordinate. Iz geometrije vemo, da je razdalja nekoplanarne točke S do ravnine π enaka razdalji d med točko S in njeno pravokotno projekcijo S' na ravnino (glej sliko 4.2). d je enaka projekciji vektorja od poljubne točke na ravnini do točke S na smer normale \vec{n} .



Slika 4.2: Pravokotni trikotnik, ki ga tvorijo \vec{NS} , projekcija vektorja \vec{NS} na smer normale d in navidezni vektor $\vec{SS'}$.

Naj bo N poljubna točka na ravnini, \vec{NS} in $\vec{NS'}$ vektorja med točkami ter α kot med obema vektorjema. Razdaljo $d = (S, S')$ označimo s $pr_{\vec{n}}\vec{NS}$, kar označuje projekcijo vektorja \vec{NS} na smer normale \vec{n} . Vektorji skupaj tvorijo pravokotni trikotnik, za katerega lahko zapišemo

$$\cos(\alpha) = \frac{pr_{\vec{n}}\vec{NS}}{\|\vec{NS}\|} \Rightarrow pr_{\vec{n}}\vec{NS} = \|\vec{NS}\| \cdot \cos(\alpha). \quad (4.7)$$

Zgornjo zvezo lahko vstavimo v enačbo za skalarni produkt

$$\vec{n} \cdot \vec{NS} = \|\vec{n}\| \cdot \|\vec{NS}\| \cdot \cos(\alpha) \iff \vec{n} \cdot \vec{NS} = \|\vec{n}\| \cdot pr_{\vec{n}}\vec{NS} \quad (4.8)$$

in izrazimo projekcijo vektorja \vec{NS} na smer normale

$$pr_{\vec{n}}\vec{NS} = \frac{\vec{n} \cdot \vec{NS}}{\|\vec{n}\|}. \quad (4.9)$$

Skalarni produkt $\vec{n} \cdot \vec{NS}$ lahko zapišemo po komponentah vektorjev

$$\begin{aligned} (S_x - N_x, S_y - N_y, S_z - N_z) \cdot (n_x, n_y, n_z) &= \\ = n_x S_x - n_x N_x + n_y S_y - n_y N_y + n_z S_z - n_z N_z &= \\ = n_x S_x + n_y S_y + n_z S_z - (n_x N_x + n_y N_y + n_z N_z). \end{aligned} \quad (4.10)$$

Ker točka N leži na ravnini, velja $n_x N_x + n_y N_y + n_z N_z = e$, zato je

$$\vec{n} \cdot \vec{NS} = n_x S_x + n_y S_y + n_z S_z - e. \quad (4.11)$$

Zapis v komponentah vnesemo v enačbo (4.9) in dobimo končno enačbo za izračun razdalje med točko S in ravnino π

$$d = \frac{n_x S_x + n_y S_y + n_z S_z - e}{\|\vec{n}\|}, \quad (4.12)$$

kjer je $\|\vec{n}\|$ velikost normale definirana kot $\sqrt{n_x^2 + n_y^2 + n_z^2}$.

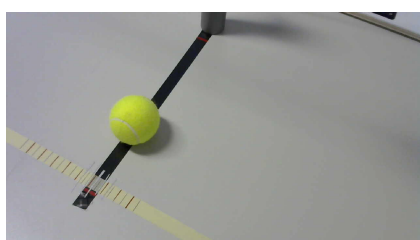
4.3 Odločanje o zadetku

Sedaj ko poznamo enačbo (4.12), izračunamo razdaljo d od središča žoge v 3D prostoru do ravnine. Sama razdalja nam ne da popolne informacije, vendar ker poznamo polmer žoge r , lahko z enostavno primerjavo, kot je prikazano na sliki 4.3, ugotovimo, ali je žoga prečkala golovo črto ali ne.

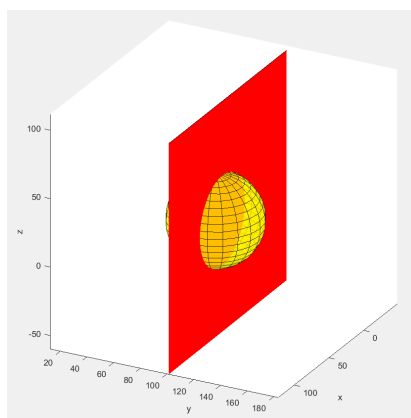
Na koncu nam preostane, da vse skupaj izrišemo s pomočjo MATLAB funkcij v 3D prostoru. Ravnina je v primeru, ko žoga ni s celotnim obsegom prečkala črte, obarvana z rdečo. V trenutku, ko se to zgodi, ravnina spremeni barvo in s tem sistem pokaže, da je gol dosežen. Na sliki 4.4 je žoga na črti, medtem ko je na sliki 4.5 s celotnim obsegom prečkala črto.

če $d \geq r$ potem
gol
drugače
ni gola

Slika 4.3: Odločanje o zadetku s primerjavo razdalje d in radija r .

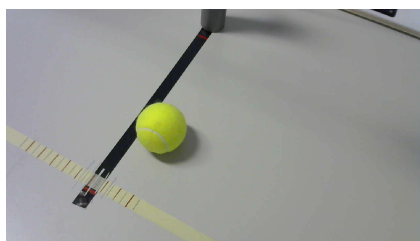


(a)

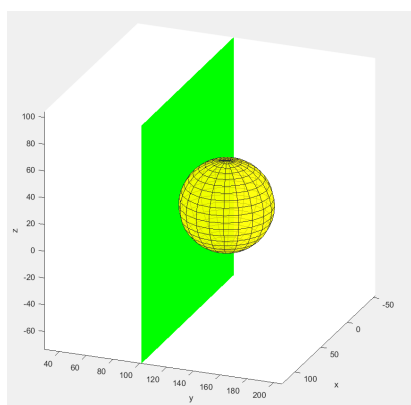


(b)

Slika 4.4: a) Pogled na žogo, ko se prekriva z golovo črto. b) 3D vizualizacija primera a).



(a)



(b)

Slika 4.5: a) Pogled na žogo v trenutku, ko je s celotnim obsegom čez črto. Zaradi lege kamere imamo občutek, kot da ni tako. b) 3D vizualizacija primera a).

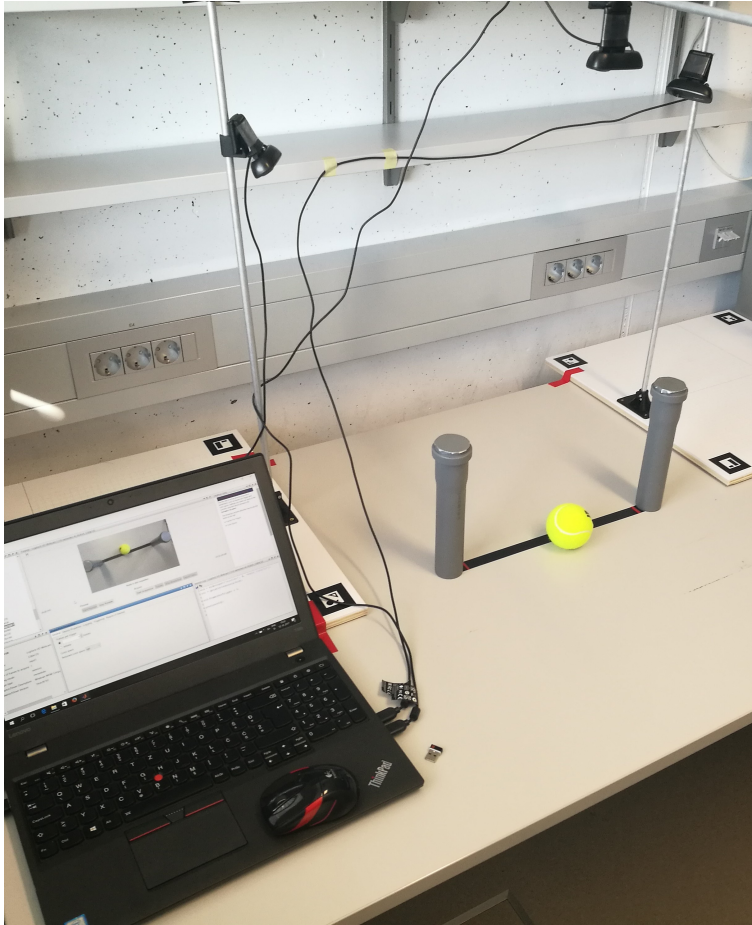
Poglavje 5

Rezultati

5.1 Postavitev sistema

V laboratoriju postavimo poenostavljen model gola. Dve kameri Logitech C310 pritrdimo na kovinski palici in ju pomaknemo v zgornja notranja kota gola, na mizi pa z barvnim lepilnim trakom označimo golovo črto. Tretjo kamero, prav tako Logitech C310, postavimo z optičnim centrom pravokotno nad golovo črto in jo bomo uporabili kot kontrolno kamero pri ocenjevanju natančnosti delovanja sistema. Kameri povežemo na prenosni računalnik, na katerem obdelamo vse podatke. Slika 5.1 prikazuje postavitev sistema v laboratoriju.

Za preizkus delovanja sistema uporabimo prenosni računalnik Lenovo s procesorjem Intel i7-6500U @ 2,50 GHz in 16 GB razpoložljivega delovnega pomnilnika. S spletnimi kamerami Logitech C310 zajamemo videoposnetke v formatu avi in slike z resolucijo 1280x720. Pri ocenjevanju natančnosti sistema dodamo še četrto kamero, in sicer sprednjo kamero telefona Huawei P9. Povprečen čas obdelave korespondenčnega para videosličic je 0,5 sekunde, zaradi česar sistem ne deluje v realnem času. Poiskuse zato najprej posnamemo in kasneje obdelamo.



Slika 5.1: Ena kamera služi kot kontrolna, preostali dve pa se uporabljata za določanje položaja žoge.

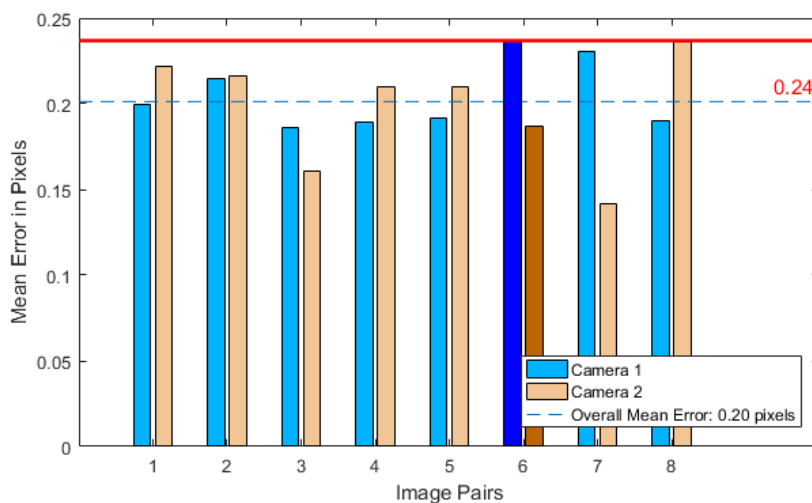
5.2 Reprojekcijska napaka

Reprojekcijska napaka (*angl. reprojection error*) nam pove, s kakšno natančnostjo se točka v svetovnem koordinatnem sistemu reprojicira nazaj na sliko. Izmerimo jo tako, da na slikah določimo korespondenčni točki x in x' , ki ju s triangulacijo preslikamo v svetovni koordinatni sistem, da dobimo točko X . Točko X nato preslikamo nazaj na obe sliki in dobimo dve novi točki x_p in x'_p , ki zaradi napak pri kalibraciji kamer ne bosta povsem ustrezali originalnima točkama x in x' . Reprojekcijska napaka je tako razdalja med

originalno in reprojicirano točko $RE = d(x, x_p)$.

Cilj je zmanjšati povprečno reprojekcijsko napako, kar lahko pri kalibraciji dosežemo tako, da odstranimo pare slik, kjer so napake zaradi zunanjih dejavnikov prevelike. Slika 5.2 prikazuje reprojekcijsko napako pri kalibraciji, katere vrednost je 0,2 slikovnega elementa.

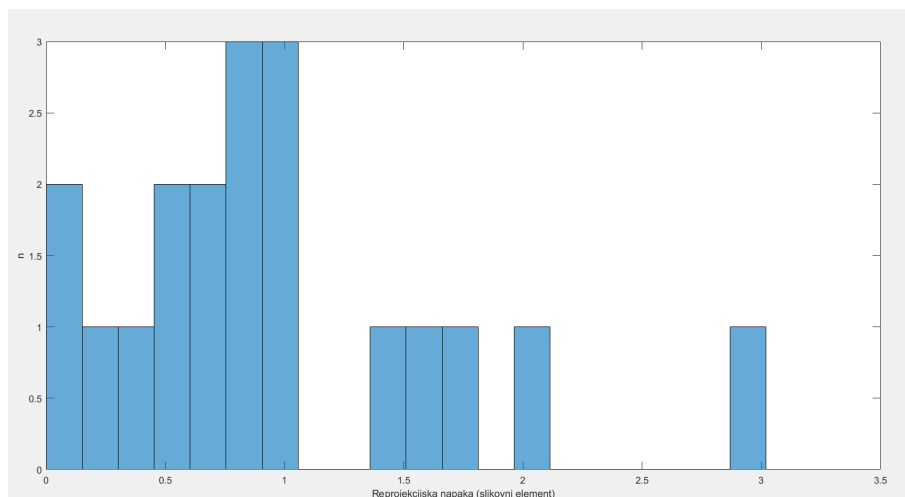
Povprečna reprojekcijska napaka pri preslikavi 3D koordinat centra žoge nazaj na sliko je bila 0,96 slikovnega elementa. Center, ki ga je na sliki določil sistem, in pa reprojicirani center sta prikazana na sliki 5.5. Posamezne meritve so predstavljene v tabeli 5.1, histogram napake za levo in desno kamero pa na slikah 5.3 in 5.4. Iz podatkov lahko vidimo, da je napaka na levem in desnem pogledu podobna in da pri tem ni večjih odstopanj.



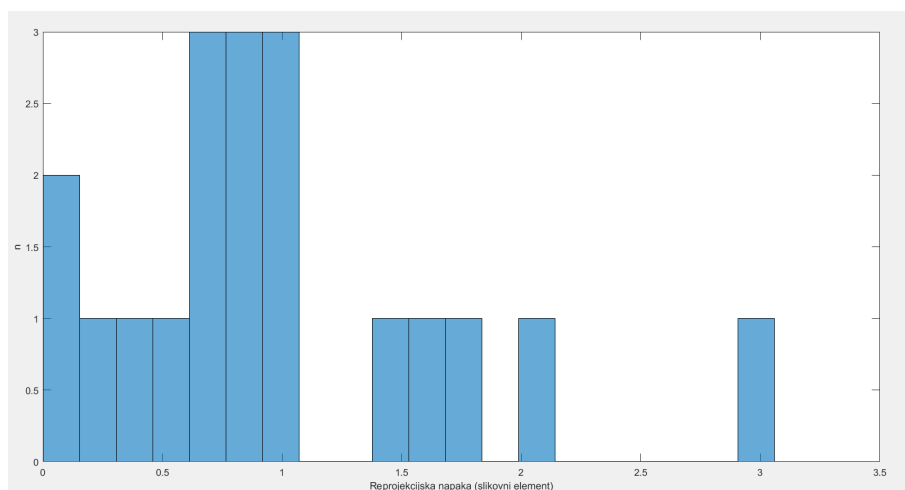
Slika 5.2: Reprojekcijska napaka posameznih slik pri kalibraciji, njena povprečna vrednost je 0,2 slikovnega elementa.

Tabela 5.1: Reprojekcijska napaka 3D koordinat centra žoge nazaj na levo in desno sliko.

RE leva slika	RE desna slika
0,7	0,7
0,0	0,0
0,2	0,3
0,0	0,0
0,7	0,9
0,9	0,9
1,0	1,0
3,0	3,0
1,0	1,0
2,0	2,0
1,6	1,6
0,6	0,6
0,3	0,3
0,5	0,5
0,6	0,6
0,8	0,8
1,8	1,8
1,4	1,4
0,9	0,9



Slika 5.3: Histogram reprojekcijske napake pri preslikavi središča krogle nazaj na sliko leve kamere.



Slika 5.4: Histogram reprojekcijske napake pri preslikavi središča krogle nazaj na sliko desno kamere.



Slika 5.5: Z rdečo označen zaznan center in z modro reprojiciran.

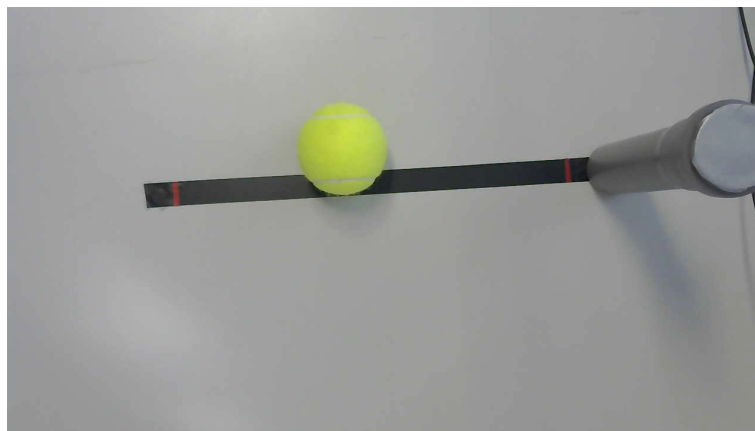
5.3 Natančnost sistema

Za ocenjevanje natančnosti obdelujemo množico zaporednih slik, na katerih poznamo točen položaj žoge glede na črto. Spletne kamere namreč ne omogočajo sočasnega zajema videosličic, zato lahko na posnetkih pride do zakasnitev oz. časovne neusklajenosti med kamerama.

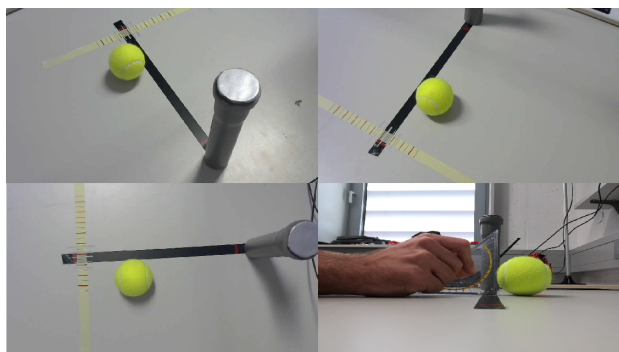
Pravilnost delovanja sistema ocenjujemo s pomočjo dveh dodatnih kamer, ki posnameta žogo iz ptičje in stranske perspektive. Obe kameri postavimo tako, da ravnina, ki predstavlja golovo črto, seka optični center kamere. Presečišče določimo ročno, vendar, kot bomo videli kasneje, to za naš primer zadostuje. Prvo kontrolno kamero postavimo torej navpično nad črto, drugo pa vodoravno ob vratnico. Primer pogleda iz ptičje perspektive prikazuje slika 5.6.

Zajamemo 20 slik, pri čemer s pomočjo geotrikotnika določimo položaj žoge. Premiki med posameznimi slikami okoli črte so na intervalu $(0, 10]$ milimetrov. Primer določanja položaja žoge v okolici črte prikazuje slika 5.7. Natančnost sistema merimo v okolici črte, in sicer:

1. od prvega stika žoge z lepilnim trakom, ki označuje golovo črto,
2. do trenutka, ko je žoga na črti, vendar s celotnim obsegom v голу.



Slika 5.6: Kontrolna kamera iz ptičje perspektive v trenutku, ko je žoga s celotnim obsegom še zmeraj v igralnem polju.



Slika 5.7: Položaj žoge smo določili s pomočjo geotrikotnika tako, da je s celotnim obsegom preko golove črte. Zgoraj pogled iz glavnih kamer, spodaj pogled iz kontrolnih kamer.

Začetek in konec merjenja prikazuje slika 5.8, vmes pa meritve opravimo na 11 slikah. Merimo razdaljo med ravnino in središčem krogle. Pričakujemo, da bo razdalja pri zadnjem merjenju enaka radiju žoge, ki je v tem primeru 33,5 mm. Razdaljo (d') izračunamo po enačbi (4.12).

Zanima nas absolutna razlika (E) med izmerjeno (d) in izračunano (d') razdaljo. Pri sistemu s popolno natančnostjo je razlika 0, vendar zaradi človeške napake pri določanju položaja žoge tega pri naših meritvah ne pričakujemo.



Slika 5.8: Začetek (a) in konec (b) merjenja natančnosti.

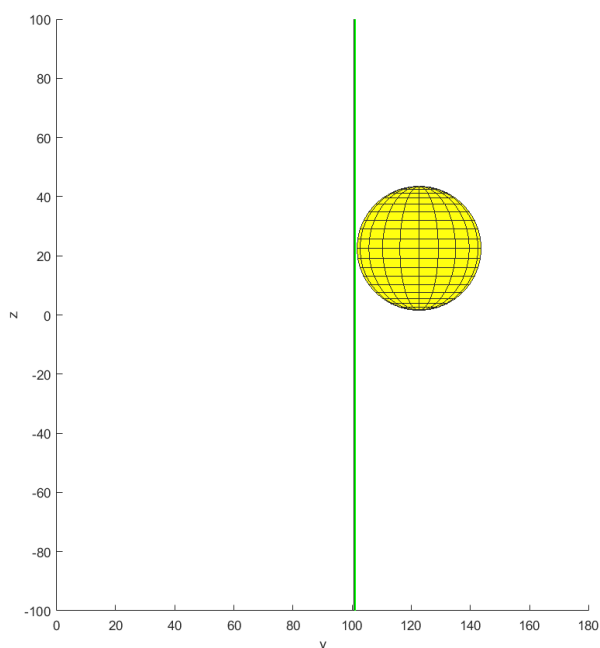
Posamezne meritve so predstavljene v tabeli 5.2.

Tabela 5.2: Tabela prikazuje izmerjeno in izračunano razdaljo od središča žoge do črte. Tretji stolpec prikazuje absolutno razliko in tako napako sistema, katere povprečna vrednost je 1,4 mm.

t	d (mm)	d' (mm)	E (mm)
1	52,5	53,5	1,0
2	41,5	43,3	1,8
3	39,5	40,5	1,0
4	33,5	33,7	0,2
5	27,5	27,3	0,2
6	25,5	-23,9	1,5
7	14,5	-12,4	2,0
8	-14,5	-18,5	4,0
9	-25,5	-27,7	2,2
10	-31,5	-32,3	0,8
11	-33,5	-34,7	1,1
avg	-	-	1,4

Razlika E pri obdelavi posameznih slik opisuje velikost napake v trenutku, ko poznamo točen položaj žoge. Središče žoge je v dveh trenutkih oddaljeno od črte natanko za radij. V času $t = 4$ je žoga s celotnim obsegom v igralnem polju, medtem ko je v času $t = 11$ s celotnim obsegom že prečkala golovo črto.

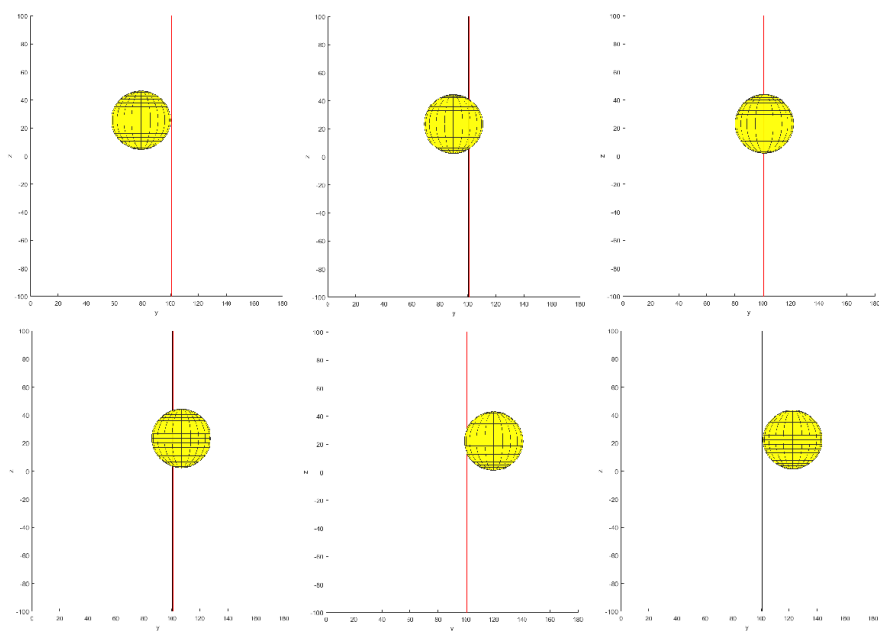
V drugem primeru je napaka večja, pri čemer je izračunana razdalja večja od izmerjene. To pomeni, da bi sistem pravilno priznal zadetek z napako 1,1 mm. Največja napaka sicer znaša 4,0 mm v času $t = 8$. Na sliki 5.9 vidimo žogo v trenutku, ko je s celotnim obsegom prečkala golovo črto, slika 5.10 pa prikazuje prečenje črte do trenutka, ko je zadetek veljaven.



Slika 5.9: Žoga je s celotnim obsegom prečkala črto. Središče žoge, s polmerom 33,5 mm, je od črte oddaljeno za $d' = 34,7$ mm.

Povprečna napaka je torej 1,4 mm. Do nje lahko pride zaradi slabše segmentacije, saj uporabljamo metodo, ki vsem slikovnim elementom dodeli enako vrednost praga. Zaradi tega se lahko v določenih primerih nekatere robne slikovne pike nepravilno klasificirajo v ozadje ali ospredje. Regija, katere center iščemo, je zato popačena in predstavlja nepravilen krog, njegovo središče pa se bo od pravega središča žoge razlikovalo za nekaj slikovnih pik. Po triangulaciji obeh primerov in izračunu razdalje od središča krogle do ravnine ugotovimo, da je razlika med primeri 0,4 mm.

Zavedati se moramo tudi, da lahko manjšo napako vnesemo pri ročnem

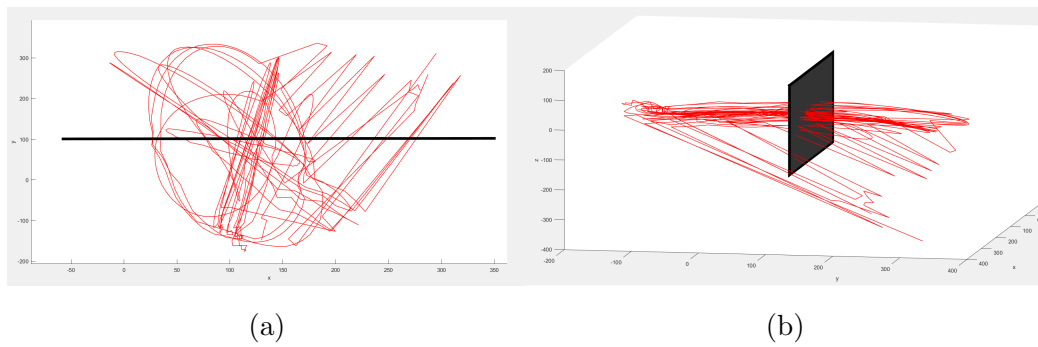


Slika 5.10: Prečenje žoge čez črto, prikazano v 3D prostoru.

določanju točk, ki omejujejo golovo črto. Kvaliteta zajetih slik je omejena in izbira korespondenčnega para točk je lahko nenatančna, zaradi česar se pojavi napaka v velikosti nekaj slikovnih pik.

5.4 Kroženje žoge

Sistem preizkusimo tudi v primeru, ko je žoga v zraku in se nenadzorovano giblje okoli črte. Na manjšo žogo smo navezali nit in jo spremenili v nitno nihalo s prijemaščem v roki nad celotnim sistemom. Roko prosto premikamo in s tem spreminjamo hitrost, smer ter višino žoge. Trajektorija posnetka je prikazana iz dveh pogledov na sliki 5.11. Velja omeniti, da pravih referenčnih vrednosti položaja žoge ne poznamo, zaradi česar ne moremo kvantitativno oceniti natančnosti. Iz trajektorije na sliki 5.11 lahko sklepamo, da je delovanje sistema gladko in brez večjih odstopanj.



Slika 5.11: Trajektorije, po katerih se je gibala žoga na niti, predstavljene iz dveh pogledov.

Poglavje 6

Zaključek

V diplomski nalogi smo zasnovali sistem za priznavanje zadetka in pri tem uporabili dve spletni kameri, ki smo ju skalibrirali in umestili v prostor. Z metodo zaznavanja ospredja smo poiskali središče žoge na korespondenčnih parih slik in ga preslikali v 3D prostor, nato pa ugotavljali položaj žoge glede na golovo črto. Ocenili smo natančnost delovanja poenostavljenega sistema in pokazali, da je mogoče z neprofesionalno opremo v kontroliranih okoliščinah doseči dobre rezultate.

Pri kalibraciji kamer nam reprojekcijske napake ni uspelo zmanjšati, na kar je delno vplivala kvaliteta uporabljenih kamer, delno pa uporabljena kalibracijska plošča, ki je bila na nekaterih mestih poškodovana. Z uporabo drugačnega vzorca kalibracijske plošče (krogi) bi morda lahko reprojekcijsko napako zmanjšali in s tem povečali natančnost sistema.

Pri videoposnetkih smo naleteli na problem nesinhroniziranosti kamer, saj te ne podpirajo zajema videoslčice v popolnoma enakem času. Potrebovali bi kvalitetnejše kamere s stikalo, preko katerega je možno sočasno zajeti slike. Trianguliramo torej središča v različnih časih, kar bi pri visokih hitrosih žoge lahko povečalo napako.

Največja omejitev sistema so naše predpostavke, saj pri zaznavanju žoge med nogometno tekmo prihaja do različnih prekrivanj, žoga ni edini premikajoči se objekt in lahko doseže visoke hitrosti. Sistem je možno nadgra-

diti z večjim številom zmogljivejših kamer, kot jih na primer uporabljajo profesionalni sistemi. Z izbiro robustnejše metode za segmentacijo in implementiranim sledenjem žogi bi sistem lahko uspešno preizkusili tudi zunaj laboratorijskega okolja.

Zasnovali smo torej poenostavljen sistem, ki v kontroliranih pogojih deluje z dobro natančnostjo. Z ustreznimi nadgradnjami bi sistem dobil tudi praktično uporabo, v trenutni obliki pa lahko služi za demonstracijo delovanja sistemov za priznavanje zadetka.

Literatura

- [1] Slika modela HSV– valj. <http://isda.ncsa.uiuc.edu/Im2Learn/doc/images/hsv.png>. Online; Dostopano: 4. 2. 2018.
- [2] Tim J Atherton and Darren J Kerbyson. Size invariant circle detection. *Image and Vision computing*, 17(11):795–803, 1999.
- [3] Baljinder Bal and Gaurav Dureja. Hawk eye: a logical innovative technology use in sports for effective decision making. *Sport Science Review*, 21(1-2):107–119, 2012.
- [4] Jean-Yves Bouguet. Matlab camera calibration toolbox. *Caltech Technical Report*, 2000.
- [5] Heng-Da Cheng, X. H. Jiang, Ying Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern recognition*, 34(12):2259–2281, 2001.
- [6] Shantanu Gangal and Sangram Raje. The hawkeye technology. *Computer Science and Engineering (CSE) Department, Indian Institute of Technology, Bombay*, 2007.
- [7] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [8] Marko Heikkila and Matti Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):657–662, 2006.

-
- [9] Berthold KP Horn. Tsai's camera calibration method revisited. *Online: http://people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf*, 2000.
- [10] Nigel JB McFarlane and C Paddy Schofield. Segmentation and tracking of piglets in images. *Machine vision and applications*, 8(3):187–193, 1995.
- [11] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [12] Boris Peter Selby, Georgios Sakas, Wolfgang-Dieter Groch, and Uwe Stilla. Patient positioning with x-ray detector self-calibration for image guided therapy. *Australasian physical & engineering sciences in medicine*, 34(3):391, 2011.
- [13] Shamik Sural, Gang Qian, and Sakti Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–II. IEEE, 2002.
- [14] Paul Swales, Joshua Cooper, and Adam Catlow. Technically... it was a GoalRef. 2014.
- [15] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.
- [16] Zhengyou Zhang. Camera calibration. In *Computer vision*, pages 76–77. Springer, 2014.