UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Anže Kovač

Zaznavanje in sledenje ljudem v sistemih z več kamerami

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Aleš Leonardis

Ljubljana, 2009

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01533/2008
Datum: 15.12.2008

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANŽE KOVAČ**

Naslov: **ZAZNAVANJE IN SLEDENJE LJUDEM V SISTEMIH Z VEČ KAMERAMI**
DETECTION AND TRACKING PEOPLE USING MULTIPLE CAMERAS

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Proučite problematiko samodejnega zaznavanja in sledenja ljudem v sistemih z več kamerami. Načrtajte in implementirajte ustrezne algoritme za detekcijo in sledenje ljudem ter za združevanje informacije pridobljene iz posameznih kamer. Analizirajte učinkovitost teh algoritmov ter jih eksperimentalno ovrednotite na primernih slikovnih video vzorcih.

Mentor:
prof. dr. Aleš Leonardis

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

Dekan:
prof. dr. Franc Solina

Zahvala

Za mentorstvo bi se rad zahvalil prof. dr. Alešu Leonardisu. Ravno tako bi se rad zahvalil vsem prijateljem, družini in še posebno moji mami za izkazano moralno podporo.

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Anže Kovač,

z vpisno številko 63020085,

sem avtor diplomskega dela z naslovom:

Zaznavanje in sledenje ljudem v sistemih z več kamerami

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
  prof. dr. Aleša Leonardisa
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 10.4.2009

Kazalo

Povzetek

Eno od zanimivih področij, s katerim se ukvarja računalniški vid, je zaznavanje in sledenje ljudem. Sistemi, s katerimi sledimo ljudem, vsebujejo eno ali več kamer. V tem diplomskem delu smo implementirali sledilnik, ki je sposoben zaznavanja in sledenja ljudem z več kamerami. Algoritem za vsak pogled posebej gradi model najpogostejših vrednosti posameznih slikovnih elementov ozadja v obliki mešanice gaussov. Slikovni elementi, ki močno odstopajo od svojega povprečja, so označeni kot nov objekt. Vse označene slikovne elemente povežemo v celoto in iz njih zgradimo barvne modele vseh novih ljudi. Preko teh modelov in napovedanega položaja sledimo ljudem v nekem časovnem obdobju. Informacijo iz posameznih kamer združimo v celoto s pomočjo tako imenovanih glavnih osi človeka. Te osi transformiramo z uporabo homografij na pogled od zgoraj, kjer določimo položaj posameznih ljudi na prizoru. Rezultati so pokazali, da tak sistem deluje učinkovito v primeru sledenja posameznemu človeku. Pogosto odpove v situacijah, kjer opazujemo več ljudi hkrati in se ti med seboj delno prekrivajo.

Ključne besede:

računalniški vid, mešanica gaussov, homografija, glavna os človeka

Abstract

One of the most interesting areas of research in computer vision is segmentation and tracking of people using monocular or multi-view systems. In this thesis we present and implement a tracker, which is capable to detect and track people using multiple cameras. Algorithm is incrementaly building a model called mixture of gaussians for each pixel independently. If the current observation does not match its model, then the appropriate pixel is marked as a foreground object (person). From those pixels we create a color representation for each foreground object. Considering color models and probable positions of the people, we track those people across the current scene. To precisely determine the ground location of a person, we map vertical axis of the person (principal axis) to a top-view plane by using homographies. The results show that this approach performs effectively when tracking individual person. However some problems are observed in situations where we monitor several occluded people in a cluttered scene.

Keywords:

computer vision, mixture of gaussians, homography, principal axis

Poglavje 1

Uvod

Kamere so postale del našega vsakdanjika. Nameščene so praktično vsepovsod: na ulicah, v avtomobilih, javnih prostorih in celo v naših domovih. Z razvojem kamer se je vzporedno razvijal tudi računalniški vid, veda ali veja računalništva, ki se ukvarja z analizo in interpretacijo slik. Sicer kot področje raziskovanja računalniški vid obstaja praktično od pojava računalnika, vendar je velik napredek doživel šele v zadnjih 20 letih. To je postalo mogoče s prihodom zmogljivejših osebnih računalnikov, ki so bili sposobni obdelati dovolj velike količine podatkov.

Eno pomembnih področij v računalniškem vidu se ukvarja z zaznavanjem in sledenjem ljudem ali drugim predmetom z eno ali več kamerami. Zgodnejše raziskave so v glavnem vsebovale sisteme z eno kamero. S prihodom splošno dostopnih, cenejših in kvalitetnih kamer so se razvili tudi sistemi, ki uporabljajo več kamer. Možnost uporabe takih sistemov se kaže na različnih področjih. Naj omenimo le nekatere izmed njih:

- Nadzor in varovanje. Tukaj gre predvsem za nudenje pomoči zaposlenemu/varnostniku, ki je zadolžen za varovanje določenega objekta ali stavbe. Namen takih aplikacij je opozarjanje na gibanje v vidnem polju kamere. Končno oceno situacije tako še vedno poda človek.

- Optimizacija delovnih procesov v skladiščih. S pomočjo rekonstruiranih poti zaposlenih ali vozil v nekem časovnem obdobju, ki so pridobljena s pomočjo takega sistema, lahko zmanjšamo število opravil ali izboljšamo učinkovitost le-teh. Podobno bi lahko tak sistem uporabili tudi v trgovinah, kjer bi analizirali nakupovalne navade potrošnikov. Na podlagi

te informacije bi prodajalec lahko ustrezneje razdelil izdelke na prodajne police.

- Naslednje zanimivo področje je analiza pretočnosti prometa, nadzor pešpoti in analiza uporabnosti določenih prostorov v mestih. Tako pridobljeni podatki bi lahko predstavljali temelje za boljšo ureditev mesta.

- Beleženje statistike na športnih prireditvah. Pridobljena informacija je uporabna za trenerje in ostale športne delavce ali pa samo predstavlja pomoč komentatorjem in režiserjem prenosov, ki gledalcem predstavijo različne podatke.

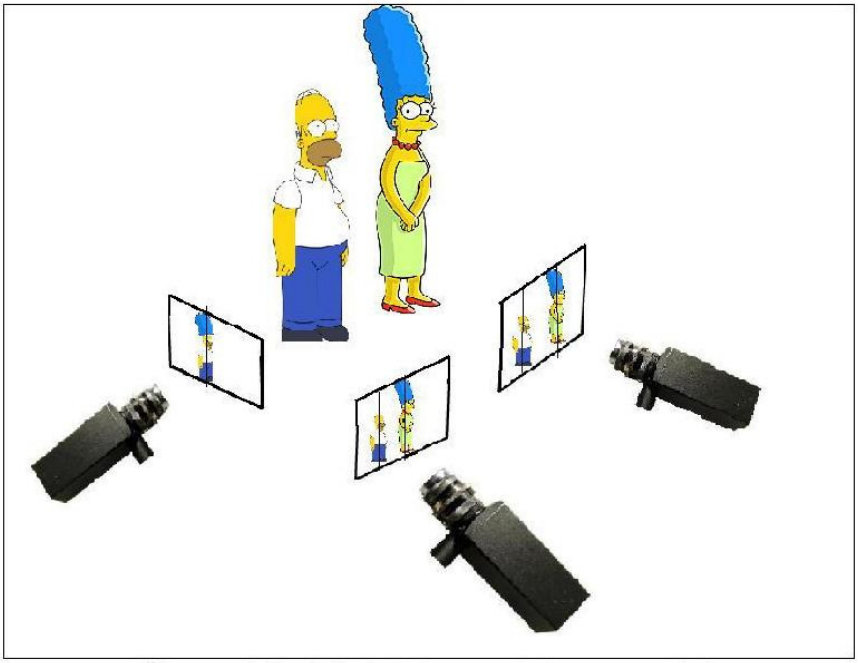
Ena izmed zelo zanimivih aplikacij, ki se danes s pridom uporablja v športih, kot so kriket in tenis, se imenuje Hawk Eye ali po slovensko Sokolje oko [20]. Sistem v vsakem delu igre sledi poti žogice in kasneje pokaže rezultat v grafični predstavitvi. Sistem pri tenisu lahko zelo natančno določi mesto padca žoge. V kolikor se igralec ne strinja z odločitvijo sodnika, lahko trikrat na niz zaprosi za pomoč omenjenega sistema. V primeru, da je imel igralec prav, mu še vedno ostanejo trije pozivi. V nasprotnem primeru zgubi eno možnost ponovnega vpogleda. Pri tem gre dejansko za enega izmed prvih uspešnih integracij merodajnega sistema v športu nasploh. Naloga te aplikacije je podobna kot v našem primeru, vendar ta sistem sledi teniški žogici, v naši aplikaciji pa bomo poskušali slediti ljudem.

Take aplikacije imajo v primerjavi s človekom nekaj prednosti. Delo, za katero so bile narejene, opravijo hitreje in predvsem bolj poceni. V glavnem predstavljajo za uporabnika enkratno denarno naložbo, kar se cenovno pozna na dolgi rok. Nadgradnja sistema z eno kamero je očitno sistem z več kamerami. Ti sistemi imajo pred sistemom z eno kamero očitno prednost. Le-ta se kaže v tem, da opazujemo prizor iz več pogledov in imamo v vsakem opazovanem trenutku celovito informacijo o dogajanju v prizoru. Pogosto se namreč dogaja, da so določena področja opazovanega prizora na enem pogledu zakrita za drugimi (ang. occluded regions), a hkrati vidna na drugem pogledu.

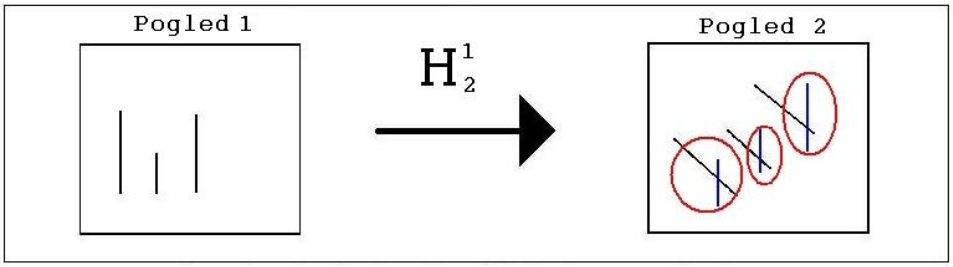
V tem diplomskem delu se bomo dotaknili dveh glavnih problemov. Prvi problem se nanaša na samo zaznavanje in sledenje ljudem z uporabo ene same kamere. Sama tematika ima zelo bogato zgodovino, saj je bilo narejeno mnogo raziskav in napisanih veliko člankov. Zadnje čase je v glavnem v rabi pristop, pri katerem vsakemu premikajočemu objektu priredimo tako imenovani barvni model videza posameznih predmetov oziroma ljudi, ki jim sledimo [2, 12]. Mi se

bomo osredotočili na metodo, ki so jo razvili Roth s sodelavci [14] in uporablja Bayesovo klasifikacijo vsakega dela slike na podlagi modelov videza in gibanja ljudi.

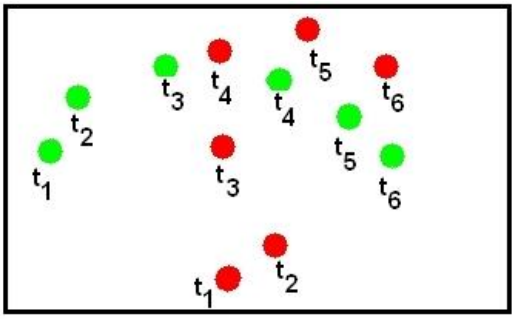
Drugi problem, s katerim se bomo soočili, pa bo predstavljal problem iskanja ujemanja oseb iz različnih pogledov. Z drugimi besedami: hočemo povezati osebe iz več pogledov in določiti ali gre dejansko za isto osebo. Problem spada v široko skupino problemov ujemanja (ang. correspondence problem). Tema velja za eno izmed novejših področij v raziskavah računalniškega vida. Obstajata dve glavni skupini pristopov za reševanje opisanega problema. Prva metoda išče podobnosti v barvnih shemah istih ljudi med različnimi pogledi in tako določi najverjetnejša ujemanja. Ena takih metod je uporabljena v [13]. Druga skupina metod, v katero bo spadala tudi naša metoda, se osredotoča na iskanje skladnosti nekaterih tipičnih točk ali telesnih značilnosti človeka [17]. Te točke ali značilnosti ponavadi predstavljajo ključne dele človeka (na primer vrh glave, najnižjo točko človeka in podobno). Mi se bomo lotili reševanja problema preko tako imenovane glavne osi človekovega telesa [6, 7]. Z uporabo določenih transformacij, ki preslikajo točke iz enega pogleda v drugega, bomo iskali najverjetnejša ujemanja ljudi, ki so opazovani iz več kamer. Kasneje bomo dva dela združili v celoto in določili položaje oseb, ki jih opazujemo. Rezultat bo predstavljen na pogledu od zgoraj (ang. top-view). Celotno idejo zasnove si lahko ogledamo na sliki 1.1.



(a) Zaznavanje in sledenje ljudem z vsako kamero posebej. Poglavje 2



(b) Transformiranje glavnih osi ljudi s homografijami in vzpostavitev ujemanja. Poglavje 3



(c) Položaj ljudi predstavljen na pogledu od zgoraj ob času t_1,...,t_n.

Slika 1.1: Zasnova delovanja našega algoritma.

Slika 1.1 nazorno predstavlja zgradbo našega pristopa. Prizor bomo opazovali z več kamerami, vsaka bo zajemala prizor iz drugega kota (slika 1.1 a). Tako bomo dobili informacijo o celotnem izgledu človeka. Z vsako kamero posebej bomo v vsakem trenutku določili nahajanja ljudi ob različnih časih t_1,...,t_n. Celoten algoritem zaznavanja in sledenja ljudem z eno kamero je podrobneje opisan v naslednjem poglavju. Za vzpostavitev ujemanja med osebami bomo osebam opisali glavne osi in jih preslikali iz enega pogleda na drugega(slika 1.1 b). Za določanje točnega položaja ljudi ob časih t_1,...,t_n, bomo njihove osi transformirali tudi na pogled od zgoraj, kjer bomo določili najverjetnejše položaje (slika 1.1 c). Tretje poglavje opisuje postopek za računanje homografij in njihova uporaba pri združevanje informacij posameznih enot v celoto. Četrto poglavje oceni kvaliteto delovanja algoritmov in prikazuje njihove rezultate, medtem ko zadnje poglavje predstavi smernice za nadaljnji razvoj in možne izboljšave programa.

Poglavje 2

Zaznavanje in sledenje z uporabo ene same kamere

V želji po visoki stopnji robustnosti bomo zasnovali naš sledilnik tako, da bo vsaka kamera predstavljala samostojno enoto. Informacija, ki prihaja iz ene kamere bo obdelana samostojno, ne ozirajoč se na druge kamere. Ta princip nam bo zagotovil določeno stabilnost sistema, saj bi tak sistem lahko nemoteno deloval tudi v primeru prenehanja delovanja ene izmed kamer. Sistem kot tak pa lahko uporabljamo na poljubni postavitvi kamer in s poljubnim številom le-teh.

2.1 Terminologija

Slikovni element je najmanjši del slike, ki nosi neko informacijo (ang. pixel). Za predstavitev slikovnega elementa obstajajo različni modeli, kot sta na primer modela CMYK ali HSL. Za naše potrebe pa bomo uporabili najbolj razširjen model RGB. V tem primeru je vsak slikovni element sestavljen iz treh komponent - barv, vsaka od teh barv pa se lahko pojavi v 256 odtenkih.

Regija predstavlja skupek slikovnih elementov, ki so med seboj povezani in naj bi pripadali istemu predmetu (ang. blob).

Območje je del slike. Lahko predstavlja množico regij, ki imajo visoko verjetnost pripadanja istemu predmetu. Tipično ponazorimo območje v obliki pravokotnika ali elipse.

Predmet sledenja je predmet, ki se nahaja v realnem svetu in kateremu poskušamo slediti s pomočjo sledilnika. V našem primeru naj bi ta predstavljal človeka.

Za boljšo predstavo zgoraj opisanih pojmov si oglejmo sliko 2.1.

Slika 2.1: Prikaz omenjenih pojmov.

2.2 Metode zaznavanja gibanja

V preteklosti so se razvili različni pristopi za zaznavanje gibajočih ljudi na zaporedju slik v sistemih z eno kamero, ki se ne premika. Ena prvih metod temelji na izračunu razlik dveh zaporednih posnetkov na določenem videu (ang. frame differencing) [11]. Formula, ki predstavlja idejo je naslednja:

$$\|I_t - I_{t-1}\| > T. \tag{2.1}$$

I_t predstavlja posnetek videa ob času t, I_{t-1} je predhodni posnetek videa in T prag razlike med obema posnetkoma. V primeru, da je razlika večja kot T, označimo regijo kot ospredje ali gibajoč predmet. Metoda ima več pomanjkljivosti. Zaznavamo lahko le regije, ki so tisti hip spremenile vrednosti. To so prednji del gibajočega se objekta in del ozadja takoj za objektom. Dejansko sploh ne dobimo celotne informacije o predmetu, ki se premika. Velik vpliv

na delovanje metode ima vrednost T. Prenizko nastavljen T vpliva na to, da je sistem zelo občutljiv na majhne spremembe, ki so posledica šuma ali rahlo spreminjajoče se osvetlitve. Previsoko nastavljen T pa ne bo dovolj uspešno ocenil objekta, ki se premika. Ta parameter se zato ponavadi določi s pomočjo poskušanja in testiranja sistema. V primeru, da se opazovani objekt ustavi, ga metoda nemudoma označi kot del ozadja in sledenje je prekinjeno. Potrebujemo nekaj, kar bo delovalo tudi v primeru, ko sledeni predmet miruje. Tako preidemo k naslednji metodi.

Računanje razlike med trenutnih posnetkom in nekim referenčnim posnetkom, ki predstavlja ozadje (ang. background substraction) [9], po naslednji formuli:

$$\|I_t - B\| > T. \tag{2.2}$$

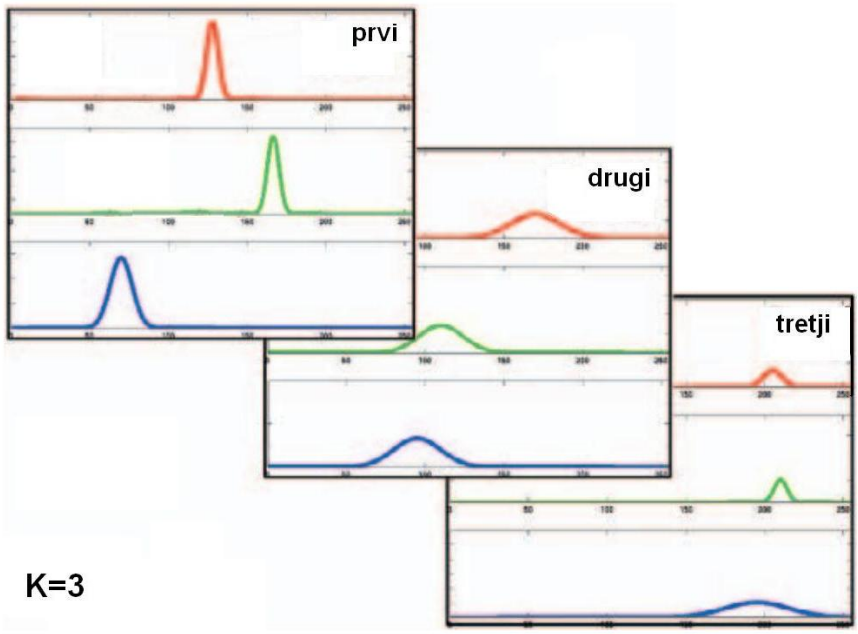
I_t predstavlja posnetek videa ob času t, T prag razlike med obema posnetkoma in B je slika ozadja. Vsak posnetek je primerjan s prednastavljeno vrednostjo ozadja. Vrednost ozadja je pridobljena na različne načine. Lahko preprosto označimo ozadje kot prvi posnetek, za katerega vemo, da ne vsebuje nobenih gibajočih objektov. Lahko pa tudi vzamemo nekaj posnetkov in ocenimo njihovo povprečno vrednost ali mediano. Glede T imamo podobne težave kot pri prvi metodi. Metoda sicer kaže zadovoljive rezultate v "nadzorovanih okoljih"(zaprti prostori). Pod pojmom nadzorovano okolje imamo v mislih okolja oziroma prizore, kjer se osvetlitev ne spreminja veliko. V primeru, da imamo okolje, kjer se osvetlitev neprestano spreminja, potrebujemo nek mehanizem, ki se bo uspel prilagoditi tem spremembam in jih upošteval kot ozadje — prilagodljivo ozadje (ang. adaptive background) [9].

$$B_t = (1 - \alpha)B_{t-1} + \alpha I_t, \alpha \in < 0, 1 > . \tag{2.3}$$

Ozadje se spreminja z α (ang. learning rate). I_t predstavlja posnetek videa ob času t. Večji α pomeni, da je trenutni posnetek integriran v ozadje hitreje. Z drugimi besedami: mirujoči objekti so hitreje označeni kot ozadje. V nasprotnem primeru pa bo hitro spreminjajoče ozadje označeno kot premikajoč se objekt. Vrednost α ponovno pridobimo s poskušanjem in testiranjem sistema.

2.3 Vzorčenje barvnih modelov z mešanico gaussov (ang. mixture of gaussians)

Ogrodje našega algoritma za zaznavanje in sledenje ljudem predstavlja nekoliko izboljšana različica prilagodljivega ozadja. Trenutno zelo priljubljena metoda se imenuje mešanica gaussov [14, 16]. Zanjo bi lahko rekli, da odpravi večino pomanjkljivosti metod iz prejšnjega podpoglavja in daje zadovoljive rezultate za naše potrebe. Sam princip metode bomo razložili na podlagi modeliranja enega slikovnega elementa preko določenega časovnega obdobja. Trenutno vrednost slikovnega elementa ob času t lahko zapišemo kot Y_t=$[R_t,G_t,B_t]^T$. Predhodno zgodovino slikovnega elementa $[Y_1,Y_2\ldots,Y_{t-1}]$ ocenjujejo kot statistični proces, ki je neodvisen od sosednjih slikovnih elementov. Tak proces je modeliran kot mešanica K gaussovih porazdelitev. K oziroma število gaussovih porazdelitev je fiksno in se giblje od 1 do 5. Tipično ga določa razpoložljiva moč računanja sistema. Večje število gaussov je ponavadi uporabljeno za modeliranje zahtevnejših prizorov na račun počasnejšega delovanja. Za boljšo predstavo mešanice gaussov s $K = 3$ si oglejmo sliko 2.2.



Slika 2.2: Mešanica gaussov s $K = 3$. Slika povzeta po [14].

I-ta gaussova porazdelitev v mešanici je natančno določena s tremi parametri, povzeto po [16]:

- $\mu_{t,k}$ - je vektor aritmetičnih sredin (ang. mean value) za vse tri RGB kanale.

$$\mu_{t,k} = \begin{bmatrix} \mu_{R,t,k} \\ \mu_{G,t,k} \\ \mu_{B,t,k} \end{bmatrix}$$

  Indeksa t in k določata stanje aritmetične sredine k gaussove porazdelitve ob času t.

- Kovariančna matrika $\Sigma_{i,t}$ ima po diagonali disperzije oziroma variance. Predpostavljamo, da so komponente RGB med seboj neodvisne, zato so ostali elementi v matriki 0. S tem prihranimo izračun kovarianc med posameznimi kanali in omogočimo hitrejši izračun inverza matrike na račun pravilnosti predpostavke.

$$\Sigma_{t,k} = \begin{bmatrix} \sigma^2_{R,t,k} & 0 & 0 \\ 0 & \sigma^2_{G,t,k} & 0 \\ 0 & 0 & \sigma^2_{B,t,k} \end{bmatrix}$$

- $w_{t,k}$ - utež k gaussove porazdelitve ob času t nam pove, kolikšen del mešanice zavzema določena gaussova porazdelitev. Velja tudi enačba 2.4:

$$\sum_{i=k}^{K} w_k = 1. \tag{2.4}$$

  Formula nam pove, da je seštevek vseh uteži določene mešanice gaussov vedno 1. Tako v vsakem trenutku vemo, kolikšen del zastopa določena gaussova porazdelitev v mešanici.

Verjetnost opazovanja določene vrednosti glede na mešanico gaussov iz predhodnih vrednosti je:

$$P(Y_t|Y_1, Y_2, ..., Y_{t-1}) = \sum_{k=1}^{K} w_{t-1,k}\, \eta(Y_t, \mu_{t-1,k}, \Sigma_{t-1,k}). \tag{2.5}$$

η predstavlja funkcijo gostote verjetnosti in jo izračunamo na naslednji način:

$$\eta(Y_t, \mu_{t-1,k}, \Sigma_{t-1,k}) = \frac{1}{\sqrt{(2\pi)^d \ |\Sigma_{t-1,k}|}} e^{-\frac{1}{2}(Y_t-\mu_{t-1,k})^T \Sigma_{t-1,k}^{-1}(Y_t-\mu_{t-1,k})}. \quad (2.6)$$

V zgornji enačbi predstavlja d dimenzionalnost. V našem primeru je nastavljena na tri (trije kanali — RGB). Z vsako iteracijo algoritma je potrebno posodobiti tudi model. Najustreznejša metoda za to se imenuje EM algoritem (ang. Expectation - Maximization algorithm)[4]. To si lahko predstavljamo kot rojenje (ang. clustering) na trenutni vrednosti slikovnega elementa in na vseh prejšnjih. Poganjanje EM algoritma na vseh slikovnih elementih je preveč časovno potratno, zato uporabimo drugo vrsto rojenja, ki je precej manj računsko zahtevno (ang. K-means approximization). Pri tej metodi je vsaka trenutna vrednost slikovnega elementa Y_t primerjana z vsemi normalnimi porazdelitvami v mešanici. V primeru, da je nova vrednost v območju 2.5 standardnega odklona gaussove porazdelitve, jo ustrezno posodobimo na naslednji način:

$$\mu_{t,k} = (1-\alpha)\mu_{t-1,k} + \alpha Y_t, \quad (2.7)$$

$$\sigma^2_{R,t,k} = (1-\alpha)\sigma^2_{R,t-1,k} + \alpha(Y_{R,t} - \mu_{R,t-1,k})^2, \quad (2.8)$$

$$\sigma^2_{G,t,k} = (1-\alpha)\sigma^2_{G,t-1,k} + \alpha(Y_{G,t} - \mu_{G,t-1,k})^2, \quad (2.9)$$

$$\sigma^2_{B,t,k} = (1-\alpha)\sigma^2_{B,t-1,k} + \alpha(Y_{B,t} - \mu_{B,t-1,k})^2, \quad (2.10)$$

$\alpha \in$<0,1> nadzoruje hitrost prilagajanja gaussove porazdelitve trenutni vrednosti in gre dejansko za isti α kot pri osnovni metodi prilagodljivega ozadja. Želimo doseči, da bi bil človek v primeru ustavitve označen kot del ozadja v približno 1 sekundi videa. Tako je ponavadi α nastavljena od 0.05 do 0.15, odvisno od števila slik na sekundo v videu (ang. frame rate). Z višjim številom slik na sekundo pričakujemo znižanje vrednosti α.
Ravno tako je potrebno posodobiti uteži gaussovih porazdelitev. Utež porazdelitve, kateri je vrednost pripadanja trenutne vrednosti največja, izračunamo po formuli:

$$w_{t,k} = (1-\alpha)w_{t-1,k} + \alpha, \quad (2.11)$$

medtem ko je za vse ostale:

$$w_{t,k} = (1 - \alpha)w_{t-1,k}. \tag{2.12}$$

Utež ustrezne porazdelitve se tako poveča, vsem ostalim porazdelitvam pa pade. Na koncu je potrebno izvesti tudi normalizacijo, da ponovno zadostimo dejstvu, da je seštevek vseh uteži 1. V primeru, da trenutna vrednost ne ustreza nobeni gaussovi porazdelitvi, nova gaussova porazdelitev nadomesti tisto z najmanjšo utežjo. Nova porazdelitev ima aritmetično sredino nastavljeno na trenutno vrednost Y_t, utež na α in varianco pa na neko inicializacijsko vrednost, ki je tipično nastavljena visoko.

2.4 Bayesova klasifikacija slikovnih elementov

Za našo metodo zaznavanja in sledenja različnim predmetov bomo uporabili tako imenovano Bayesovo klasifikacijo slikovnih elementov. Princip metode je segmentacija oziroma razdelitev slike na tako imenovane objekte ospredja in ozadje. Delitev je narejena na podlagi verjetnosti pripadanja slikovnih elementov tem objektom. Sama verjetnost predstavlja oceno za pripadnost določenega slikovnega elementa tem predmetom, upoštevajoč trenutno vrednost slikovnega elementa, barvne modele posameznih predmetov in njihov pričakovani položaj. Z drugimi besedami: Bayesova klasifikacija na vsaki iteraciji algoritma vsak slikovni element dodeli enemu izmed predmetov ospredja O_i (i $\in \{1..n\}$) ali ozadju $B = O_0$. Matematično bi to zapisali z naslednjima enačbama, ki sta povzeti po [14]:

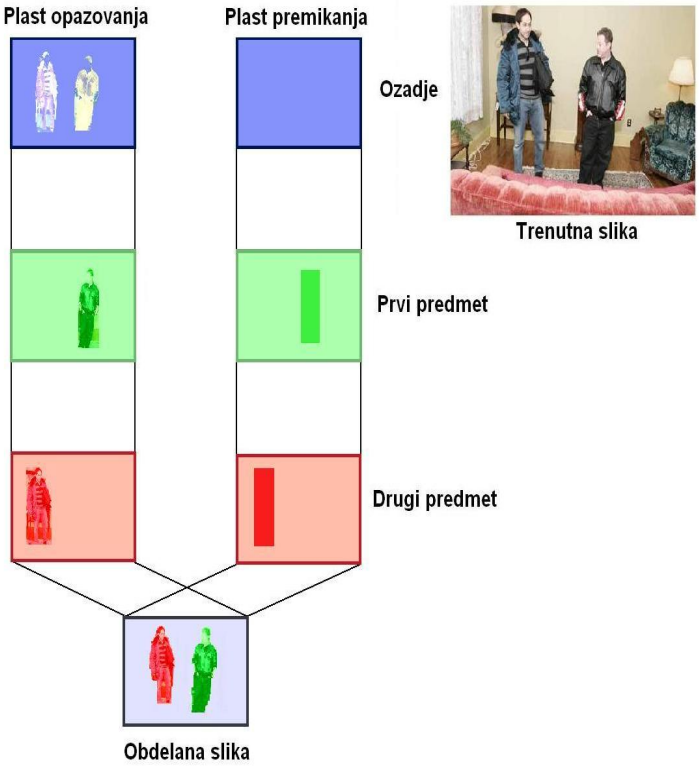
$$P_{posterior}(O_i|pixel) \ \propto \ P(pixel|O_i)P_{prior}(O_i), \tag{2.13}$$

$$class(pixel) = \arg\max_{i=0..n} P_{posterior}(O_i|pixel). \tag{2.14}$$

Z besedami to pomeni: pogojna verjetnost predmeta O_i pri dejstvu, da poznamo trenutne vrednosti slikovnega elementa (RGB in koordinate) je sorazmerna zmnožku verjetnosti trenutne vrednosti slikovnega elementa pri pogoju, da poznamo "vrednosti"(barvne modele) predmeta, in predhodne verjetnosti nahajanja predmeta. Slikovni element tako označimo kot del objekta, kateremu najverjetneje pripada.

Kot nazorno vidimo, je končna verjetnost sestavljena iz dveh delov. Ideja Bayesove klasifikacije se kaže tudi v ločevanju barvnih modelov od dela, ki

se ukvarja s položaji in gibanjem posameznih predmetov. To nam kasneje pride še kako prav, saj tako v bistvu rešujemo dva problema, ki med seboj nista strogo povezana. Plast, ki se ukvarja s pogojno verjetnostjo $P(pixel|O_i)$, imenujemo plast opazovanja (ang. observation layer). Sama plast se ukvarja z barvnimi modeli predmetov in ozadja. Pod pojmom modeli tukaj razumemo barvni izgled posameznega dela predmeta ali ozadja. Za naš algoritem bomo uporabili predhodno opisano mešanico gaussov. Druga verjetnost $P_{prior}(O_i)$, ki sledi iz plasti gibanja (ang. motion layer), pa ima v domeni gibanje in položaje predmetov. Zasnovo plasti in objektov prikazuje slika 2.3, kjer se jasno vidijo plasti vsakega predmeta, ki mu sledimo. Navpično imamo delitev med plastjo opazovanja in plastjo gibanja. Vodoravno pa je delitev med posameznimi predmeti (predmeti, ki se gibajo in ozadje). Z vsakim novim predmetom, ki mu sledimo, se stolpca na sliki 2.3 povečata za ena. Teoretično tako lahko sledimo poljubnemu številu oseb.



Slika 2.3: Koncept sledenja več predmetom: vsak predmet je sestavljen iz plasti opazovanja in plasti gibanja.

2.5 Plast opazovanja

Plast, ki deluje neposredno na informaciji pridobljeni iz trenutnega posnetka, imenujemo plast opazovanja. Ukvarja se z barvnimi shemami oziroma barvnimi modeli različnih premikajočih predmetov in ozadja. Skrbi za detekcijo (zaznavanje) premikajočih predmetov, sledenje, inicializacijo njihovih modelov in vzdrževanje le-teh. Glavna ideja vpeljave barvnih modelov izgleda predmetov je, da s pomočjo njih izračunamo verjetnostne slike posameznih objektov, kjer imajo tisti slikovni elementi, ki se ujemajo z modelom, visoko verjetnost in tisti, ki se ne, nizko verjetnost. Sama plast je razdeljena na dva dela, saj razlikujemo med modelom premikajočih predmetov in modelom ozadja. Čeprav modela v osnovi uporabljata mešanico gaussovih porazdelitev, ki smo jo predhodno opisali, so razlike med njima kar precejšnje. Posamično jih bomo razložili v naslednjih dveh podpoglavjih.

2.5.1 Barvni model ozadja (ang. background appearance models)

Z dejstvom, da imamo v našem sistemu le kamere, ki se ne premikajo, lahko vsak slikovni element modeliramo z mešanico gaussovih porazdelitev. Tak pristop se imenuje TAPPMOG (ang. time adaptive per-pixel mixture of gaussians). Eden prvih takih algoritmov je opisan v [16]. Pred tem je bila v uporabi metoda, ki je uporabljala le eno gaussovo porazdelitev. S tako metodo bi na primer lahko modelirali okolje, kjer se osvetlitev spreminja zelo počasi ali sploh ne. Kot primer lahko podamo zaprto sobo, kjer je edini vir svetlobe luč. Slabost modeliranja ozadja z eno gaussovo porazdelitvijo se pokaže v primeru, ko opazujemo "zunanji svet". Gibanje trave, dreves ali oblakov želimo označiti kot del ozadja. Za kompenzacijo takih gibanj, ki se človeku zdijo za nekaj samoumevnega, je potrebno uporabiti več gaussovih porazdelitev. Vsaka porazdelitev tako predstavlja odtenke barv, ki se največ pojavljajo v nekem časovnem obdobju opazovanja.

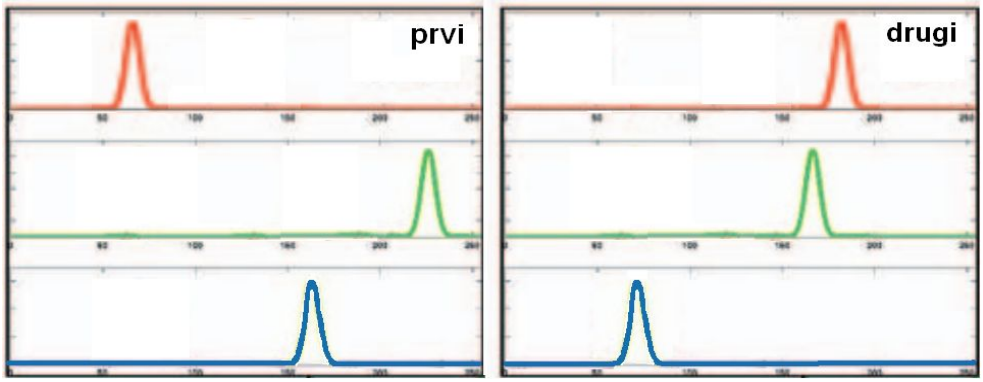
Vsak slikovni element na sliki je modeliran na podlagi predhodnih opazovanj kot mešanica K gaussovih porazdelitev. Pri tem se pojavi vprašanje, katere porazdelitve predstavljajo ozadje. Z drugimi besedami, katere porazdelitve imajo največjo verjetnost, da so pridobljene iz ozadja. Zanimajo nas le tiste, ki imajo največjo utež in najnižjo varianco. Da bi to bolje razumeli, si oglejmo naslednje dogajanje. Večino časa opazujemo neko statično okolje — ozadje. Trenutne vrednosti slikovnih elementov ostajajo večinoma iste. Tako bo posodobljena

vedno ista porazdelitev. Tej porazdelitvi se bo povečala utež in zmanjšala varianca. V nasprotnem primeru, ko slikovni element izrazito spremeni svojo vrednost, oziroma se na prizoru pojavi nov predmet, pa ustvarimo novo porazdelitev z nizko utežjo in visoko varianco. Za ustrezno ločitev ozadja od premikajočih predmetov potrebujemo nek mehanizem, ki bo določil katere porazdelitve pripadajo premikajočim predmetov in katere ozadju. Rešitev za ta problem je razmeroma preprosta. Najprej sortiramo porazdelitve po kvocientu w/σ od najvišjega do najnižjega. Nato se ravnamo po formuli 2.15, ki je povzeta po [16]:

$$B = \arg\min_{b} \left(\sum_{k=1}^{b} \frac{w_k}{\sigma_k} > T \right). \tag{2.15}$$

Prvih B porazdelitev bo označenih kot ozadje. T je prag, ki določa kolikšen del vseh porazdelitev predstavlja ozadje. V primeru, da je T nastavljen na nizko vrednost, je ponavadi izbrana le prva porazdelitev. V nasprotnem primeru, za kompleksnejše ozadje, je le-to sestavljeno iz več porazdelitev. Izbira je pogojena z izbiro okolja, ki ga opazujemo.

Za našo implementacijo bomo uporabili nekoliko optimizirano verzijo zgornjega algoritma uporabljeno v [14]. Imenuje se (ang. optimized adaptive per-pixel model). Razlogov za tako izbiro je več. Pogosto v mešanici prevladuje le ena gaussova porazdelitev z utežjo 0.80 ali več. Tako ne potrebujemo visokega K za predstavitve zahtevnejših oziroma kompleksnejših prizorov. Uporabljali bomo maksimalno K=3. Poganjamo namreč celoten algoritem za vsak slikovni element posebej, kar predstavlja enega izmed ozkih grl našega celotnega programa. V algoritem lahko vpeljemo še dodatke poenostavitve. Za vse gaussove porazdelitve lahko privzamemo fiksno varianco. Nastavimo jo lahko na neko primerno vrednost glede na prisotnost šuma na videu, dobljena pa je ponavadi s testiranjem sistema in je tipično na intervalu od 10 do 20. Model za ozadje enega slikovnega elementa s to metodo si lahko ogledamo na sliki 2.4, ki predstavlja mešanico dveh gaussovih porazdelitev s fiksno varianco. Fiksna varianca se odraža v enaki višini in širini gaussove krivulje vseh barvnih kanalov v gaussovih porazdelitvah v mešanici.

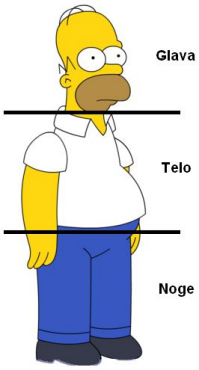


Slika 2.4: Model ozadja za posamezen slikovni element z dvema porazdelitvama in s fiksno varianco.

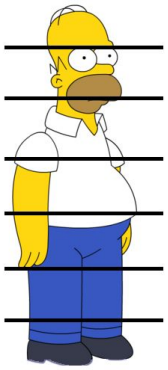
2.5.2 Model premikajočih predmetov (ang. foreground appearance models)

Ko je verjetnost pripadanja slikovnega elementa določenemu objektu nizka za vse obstoječe objekte $\{B_0,O_1,...,O_n\}$, je opazovani slikovni element kandidat za obstoj novega objekta. Imenujmo verjetnost za pripadanje novemu predmetu P_n. Ta verjetnost je uniformno nastavljena na zelo nizko vrednost in je za vse slikovne elemente enaka. Vpeljava te verjetnosti v algoritem prinaša nekatere prednosti. Verjetnost lahko enakovredno tekmuje z ostalimi modeli pri klasifikaciji slikovnih elementov. Poleg tega predstavlja učinkovit mehanizem za čiščenje različnega šuma na sliki. V primeru, da je bil slikovni element večkrat označen kot kandidat za nov objekt, pri tem pa nikoli ne pride do same inicializacije novega objekta, lahko naknadno še bolj zmanjšamo P_n in tako onemogočimo pripadnost tej vrednosti. Nov objekt se inicializira takoj, ko ima regija določeno velikost. Inicializacijska velikost predmeta je odvisna od opazovanega prizora. Kadar kamera opazuje okolje od blizu, pričakujemo da bo tudi sam predmet zavzemal večje področje na sliki. Velikost bomo v takem primeru nastavili na višjo vrednost. V obratnem primeru pa, ko kamera opazuje okolje od daleč, se velikost predmeta ustrezno zmanjša. Vsak tak predmet dobi svoj model izgleda. Ker za celotnega človeka model ene mešanice gaussovih porazdelitev ni dovolj, se bomo odločili za tako imenovani model z rezinami (ang. sliced object model). Pri tej metodi predvidevamo, da je premikajoč človek ponavadi v stoječem položaju in ga tako razdelimo na več delov [12].

Pri tem se nam ponujata dve možnosti. Lahko razdelimo človeka na tri različne dele: glavo, telo in noge (Slika 2.5). V drugem primeru pa je človek razdeljen na poljubno n število enako velikih rezin, vsaka zavzema $1/n$ človeka (Slika 2.6). Tipično je vrednost n med 1 in 10.



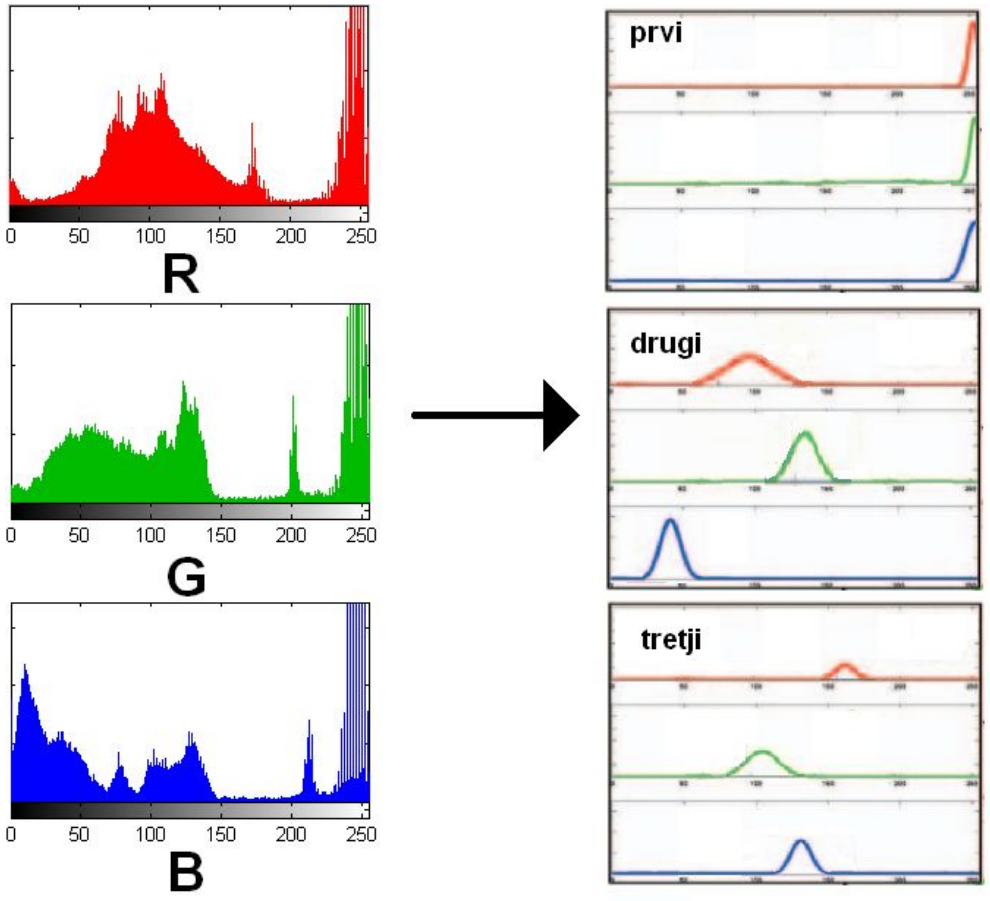
Slika 2.5: 3 delni model izgleda človeka.



Slika 2.6: 7 delni model človeka.

Za naše potrebe bomo uporabili model z n rezinami. Število rezin je odvisno predvsem od postavitve in oddaljenosti kamer (velikosti opazovanega premeta). V primeru, da kamera opazuje človeka od blizu in iz strani, uporabimo večje število rezin. V nasprotnem primeru to število zmanjšamo. Pri določanju h kateri rezini slikovni element spada, si pomagamo z domnevnim položajem celotne osebe in koordinatami določenega slikovnega elementa.

Vsak del je modeliran z mešanico gaussovih porazdelitev pridobljeno iz histograma določene regije. Na sliki 2.7, ki prikazuje mešanico treh gaussovih porazdelitev pridobljeno iz histograma. Jasno se vidi, da je prva gaussova porazdelitev iz najpogostejših vrednosti določene barve. Najvišji vrhovi posamezne barve sovpadajo z vrhovi gaussovih krivulj prve gaussove porazdelitev. Druga in tretja gaussova porazdelitev pa sta tipično narejeni iz ostalih vrhov histograma, kar se odraža v porazdelitvah z visoko oziroma višjo varianco in nižjimi vrhovi.



Slika 2.7: Mešanica gaussov pridobljena iz ustreznega histograma.

2.6 Plast gibanja

Vsak predmet ima poleg barvnega modela izgleda tudi informacijo, kje naj bi se določen predmet nahajal. Plast gibanja je neposredno odgovorna za delo z modelom gibanja oziroma nahajanja predmeta. V Bayesovi klasifikaciji slikovnih elementov iz te plasti dobimo verjetnost predmetov $P_{prior}(O_i)$ povzeto po enačbi 2.13. Po vzoru prejšnje plasti tudi tukaj ločimo dva modela, statičnega in model gibajočih predmetov. Ozadju B in novim objektom N_i privzamemo uniformno verjetnost nahajanja. Tako dodelimo enako verjetnost P_u celotni sliki. Statični model se uporablja predvsem v primerih, ko imamo opraviti z predmeti, katerih položaja ne moremo natančno napovedati. Sicer se osebe oziroma gibajoči predmeti ponavadi pojavijo v točno določenem delu

prizora, ki jo opazujemo. Če bi dovolj dolgo opazovali prizor, bi lahko take dele lahko označili z višjo verjetnostjo nahajanja novih predmetov. Brez posebnih težav pa lahko uporabljamo za vse slikovne elemente isto verjetnost P_u. Predmetom, ki se gibajo, je dodeljena druga vrsta modela. Vsak predmet je opisan z mejnim okvirjem (ang. bounding box), za katerega poznamo center C=[x,y]^T in velikost V=[V_x,V_y]^T predmeta (slika 2.10). Napoved, kje naj bi se predmet nahajal na naslednjem posnetku, je pridobljena iz hitrosti gibanja centra predmeta. Hitrost centra in velikost mejnega okvirja se spreminja po podobnih enačbah 2.16, 2.17. α predstavlja hitrost učenja, C_t - C_{t-1} razliko centrov na zadnjih posnetkih in H je okvir predmeta na trenutnem posnetku:

$$dC_t = (1 - \alpha)dC_{t-1} + \alpha(C_t - C_{t-1}), \tag{2.16}$$

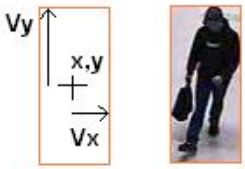
$$V_t = (1 - \alpha)V_{t-1} + \alpha H. \tag{2.17}$$

Predhodno verjetnost nahajanja predmeta tako dobimo iz napovedi položaja centra in mejnega okvirja. Verjetnost je določena uniformno visoko znotraj tega območja ter se nato linearno zmanjšuje z razdaljo. Vzrok za vpeljavo območja, kjer verjetnost linearno pada, leži predvsem v kompenzaciji za možne napake pri izračunih domnevnih položajev. V tem območju ima predmet nižjo verjetnost nahajanja, vendar ne nič. Širina tega območja je tipično nastavljena od 10 do 20 slikovnih elementov.

Kot vidimo iz slike 2.9, je predhodna verjetnost večji del slike enaka nič. To pomeni, da se izognemo računanju verjetnosti pripadanja barvnih shemam predmeta za slikovne elemente, kjer je verjetnost nahajanja predmeta nič. Še posebno pride to dejstvo do izraza, ko sledimo več objektov, saj je matrični izračun verjetnosti ena izmed računsko najbolj potratnih operacij v našem programu.

Slika 2.8: Nahajanje osebe na prizoru.

Slika 2.9: Predhodna verjetnost nahajanja človeka.

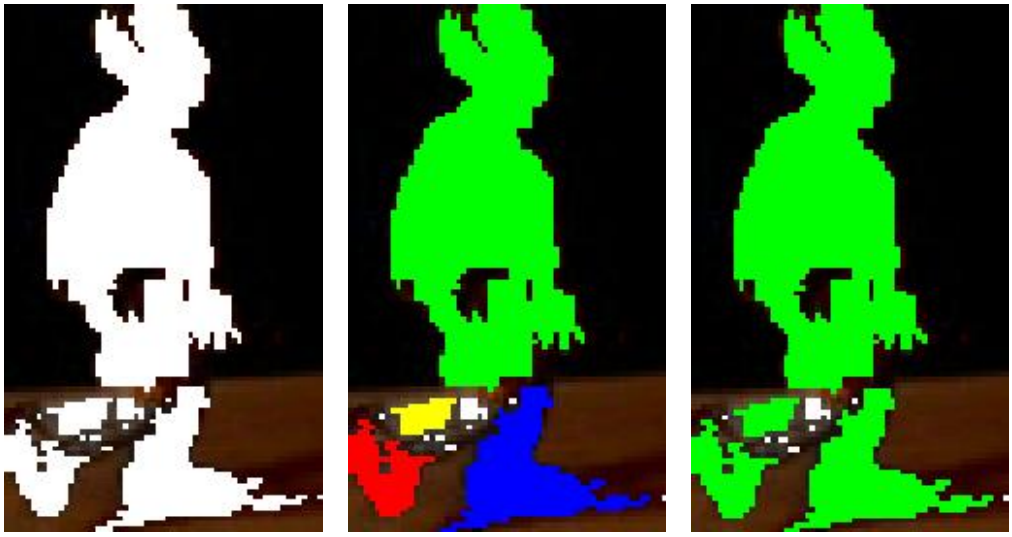


Slika 2.10: Mejni okvir sledenega človeka.

2.7 Povezava slikovnih elementov v območja

Po fazi klasificiranja moramo slikovne elemente povezati med seboj v regije in nato regije v območja (ang. blobing). Povezovanje slikovnih elementov v celoto je nujni korak, saj želimo imeti celovito informacijo o predmetu, ki mu sledimo. Zaradi šuma na sliki, prevelike podobnosti med barvnimi modeli predmetov, se pogosto zgodi, da je nekaj slikovnih elementov narobe klasificiranih. Rezultat tega je, da ne obstajajo direktne povezave med vsemi slikovnimi elementi, ki naj bi pripadali istemu predmetu.

Vse med seboj povezane slikovne elemente označimo kot isto regijo. Obstaja mnogo pristopov kako povezati slikovne elemente v ustrezne skupine (ang. connected component labeling). Algoritmi tipično naredijo dva sprehoda čez sliko. Obstajajo tudi algoritmi, ki združijo pomembne korake v združevanju, kar se odraža v enem sprehodu čez celo sliko. Mi bomo za naš namen uporabili preprosto metodo, ki deluje po principu poplave (ang. flood-fill). Sprehajamo se po slikovnih elementih, dokler ne naletimo na slikovni element z ustrezno vrednostjo. Iz tega slikovnega elementa nato naredimo "poplavo". V kolikor imajo



(a) (b) (c)

Slika 2.11: Princip povezave slikovnih elementov v območja.

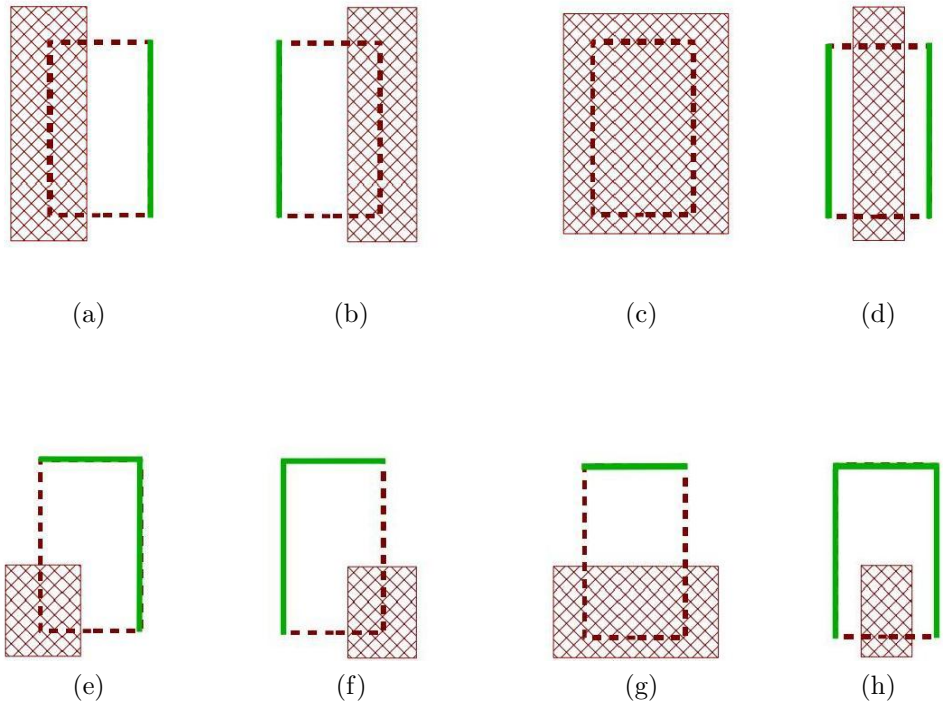
sosednji slikovni elementi enako vrednost, jih označimo z isto številko regije (slika 2.11b). Vsi slikovni elementi, ki jih obiščemo, pa so označeni kot obiskani. Tako jih ob glavnem sprehodu od začetka do konca preskočimo. Regije z enako vrednostjo nato povežemo v skupno območje (slika 2.11c). Kadar je razdalja med mejnimi okvirji regij manjša od praga, označimo regije kot eno območje.

2.8 Popolno in delno zakritje predmetov (ang. occlusion handling)

Eden najpomembnejših delov našega algoritma predstavlja detekcija zakrivanja predmetov med seboj. Potrebujemo nek mehanizem, ki bo zmožen odkrivanja takih situacij in pravilnega interpretiranja le-teh. Pogosto zgodi, da pride pri premikanju do križanja objektov med seboj (ang. overlapping). To pomeni, da en objekt zakrije drugega in drugi se znajde izven vidnega polja kamere (ang. occluded object). Informacija o teh objektih postane nezanesljiva. V preteklosti so razvili različne pristope, ki ta problem obravnavajo v sistemih z eno kamero [15, 18]. Pri nas bomo detektirali zakritje z vsako kamero posebej, kot je opisano v [14].
Za detekcijo zakrivanja predmetov moramo le izračunati ali se mejna okvirja dveh predmetov med seboj prekrivata. Problem pa nastane takoj, ko želimo

doreči, kateri izmed predmetov je bližje kameri. Pri tem preprosto predpostavimo, da je predmet, ki je bližje kameri, tudi bližje dnu slike. Slednja trditev velja večinoma v primeru, ko opazuje predmete s strani. Na pogledu od zgoraj to ne drži, vendar pa na tem pogledu zakrivanja sploh ne pričakujemo.

Položaj vsakega predmeta $O_i \in \{O_1,...,O_n\}$, ki ga predstavlja mejni okvir (*xMax, xMin, yMax, yMin*), preverimo, ali se prekriva s položajem drugih objektov $O_j \in \{O_1,...,O_n\} \setminus O_i$. Če najdemo takšen primer zakritja, določimo, za katero vrsto zakritja gre. Imamo namreč 8 različnih možnosti, da se to zgodi. Pri vsaki vrsti se zakriva drug del predmeta. Del, kjer se zgodi zakrivanje, je označen kot "netočen"oziroma nezanesljiv.



Slika 2.12: 8 vrst zakrivanja dveh predmetov. Slika je povzeta po [14].

Vse možne vrste zakrivanja si lahko ogledamo na sliki 2.12. Mrežast mejni okvir predstavlja predmet v ospredju. Glede na vrsto zakrivanja je ena ali več mejnih koordinat (*xMax, xMin, yMax, yMin*) označenih kot nezanesljive. Na sliki se to vidi kot črtkasta črta. Polna črta predstavlja koordinate, ki jim verjamemo. Položaj in velikost predmeta je kasneje rekonstruirana glede na veljavne koordinate. Ostalo je pridobljeno iz informacije, ki nam je bila posredovana iz prejšnjih posnetkov. Takrat naj bi bil predmet viden v celoti. Za jasnejšo predstavo si oglejmo naslednji primer g iz slike 2.12. Imamo 3 neveljavne koordinate. Kot natančno vzamemo samo zgornjo koordinato okvirja. Ostale koordinate okvirja so pridobljene iz predhodnega stanja. V tem primeru tako posodobimo le položaj centra v y smeri in ne v x smeri. Sama velikost predmeta se ne spremeni. Center se premakne le v y smeri glede na spremembo zgornje y koordinate ($yMax$). V primeru c iz slike 2.12, ko je predmet popolnoma zakrit, je njegov položaj in velikost domena izključno predhodne informacije.

V tem poglavju smo spoznali princip za zaznavanje in sledenje ljudem z eno samo kamero. Opisali smo glavne korake pri sami zasnovi metode, ki smo jo uporabili za naš algoritem. Pogosto se pri omenjenih pristopih pojavljajo težave, ki smo jih poskušali identificirati in kasneje odpraviti. V naslednjem poglavju skušamo združiti informacijo iz posameznih pogledov v celoto in zasnovati sistem več kamer, ki bo zmožen sledenja ljudem in bolj natančnega določanja njihovega položaja.

Poglavje 3

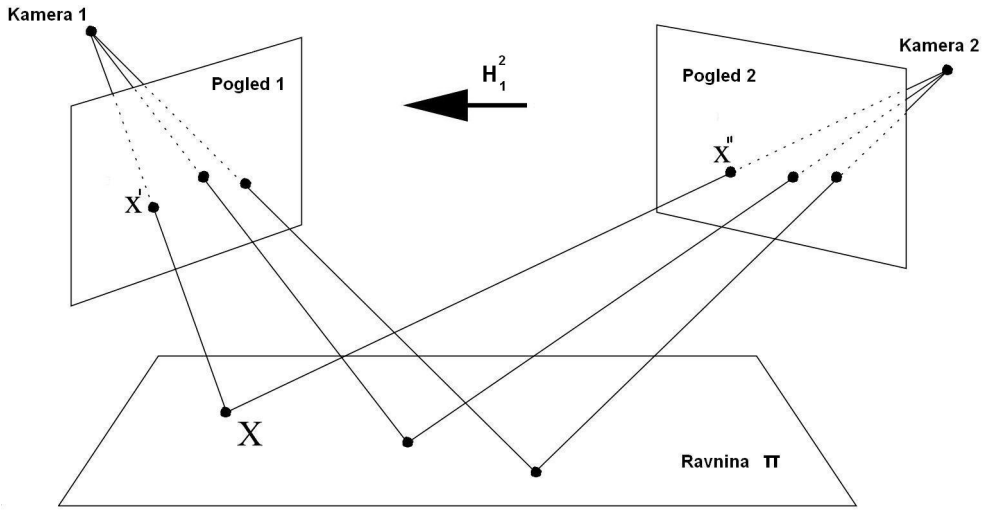
Sledenje ljudem z več kamerami

V prejšnjem poglavju smo razložili princip, kako identificirati posamezne ljudi v prizoru, ki ga opazujemo. Tako določimo, kateri deli slike pripadajo osebam in kateri ozadju. Teoretično je za določitev mesta nahajanja posameznih ljudi v prizoru in predstavitev na pogledu od zgoraj dovolj že ena sama kamera. Vendar kot se bo kasneje izkazalo, je informacija iz ene same kamere razmeroma nezanesljiva. Vzrok za to tiči v slabši klasifikaciji slikovnih elementov. Da bi točneje določili položaj oseb v prizoru, bomo sledenje razširili na več kamer. V tem poglavju bomo poskušali opisati metodo, kako kar najbolj učinkovito združiti informacijo iz posameznih kamer v celoto.

Predpostavljamo, da vse kamere opazujejo isti prizor iz več različnih zornih kotov. Vsaki osebi na enem pogledu mora tako pripadati ustrezna oseba na ostalih pogledih. Določili bomo ujemanja na podlagi njihovih položajev. Kot položaj tukaj razumemo mesto nahajanja človeka. Za mesto nahajanja človeka bomo uporabili najnižjo točko človeka ali glavno os človeka (več o tem v podpoglavju o glavni osi in talni točki). Da bi ustrezno povezali osebe iz različnih pogledov, moramo primerjati med seboj položaje teh oseb in tako določiti, ali gre za isto osebo. Obstajajo tako imenovane obojestranske transformacije - projekcijske transformacije (ang. projectivities), ki vsako točko iz enega pogleda slikajo v točno določeno točko na drugem pogledu. Imenujemo jih tudi homografije (ang. homographies, collineations). Razložili bomo izračun homografije in njihovo uporabo pri ugotavljanju mere ujemanja med osebami iz različnih pogledov. Za konec pa bomo predstavili izboljšano določanje položajev oseb, ko uporabimo več kamer, v primerjavi z eno samo.

3.1 Izračun homografije

Opazujemo predmet ali točke na ravnini π iz več pogledov. Vsaki realni točki na ravnini tako ustreza točka na vsakem izmed pogledov. Zanima nas, pri katerih točkah iz različnih pogledov gre dejansko za iste realne točke na ravnini π. Za lažjo predstavo si oglejmo sliko 3.1, kjer se jasno vidijo ujemanja med realno točko X in točkami X' in X'' na obeh pogledih.



Slika 3.1: Realni točki X na ravnini π ustreza točka X'=[x',y'] na pogledu 1 in točka X"=[x'',y''] na pogledu 2. H_1^2 pomeni transformacijo iz ravnine 2 na ravnino 1.

Želimo najti "ujemanje"koordinatnega sistema x' (iz pogleda 1) s koordinatnim sistemom x" (iz pogleda 2). Povezavo med ravninama opisujeta formuli 3.1 in 3.2. Isti indeksi pri komponentah h označujejo iste elemente, hkrati pa tudi predstavljajo mesto nahajanja v matriki, ki jo bomo uporabili v formuli 3.3.

$$x' = \frac{h_{11}x'' + h_{12}y'' + h_{13}}{h_{31}x'' + h_{32}y'' + h_{33}}, \tag{3.1}$$

$$x_2' = \frac{h_{21}x'' + h_{22}y'' + h_{23}}{h_{31}x'' + h_{32}y'' + h_{33}}. \tag{3.2}$$

Za lažjo predstavo si tako homografijo H_1^2 lahko predstavljamo kot matriko 3x3, ki preslika vektor X''=[x",y",1]^T v vektor X'=[x',y',1]^T. Vsaka točka je predstavljana v homogenih koordinatah. Matrični zapis tako sledi:

$$\begin{bmatrix} x_1' \\ x_2' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1'' \\ x_2'' \\ 1 \end{bmatrix} \tag{3.3}$$

ali krajše

$$X' = H * X''.$$

Za izračun matrike H bomo uporabili tako imenovano homogeno metodo (ang. homogeneous estimation method) [19], pri kateri predpostavimo, da poznamo n parov (X_i iz prvega koordinatnega sistema in U_i iz drugega koordinatnega sistema), za katere vemo, da predstavljajo iste točke. Za vsak par morata tako veljati enačbi 3.1 in 3.2. Vsak par (X_i, U_i) nam da dve homogeni enačbi. Z enostavnim izračunom lahko zgornje enačbe zapišemo na naslednji način:

$$\begin{bmatrix} h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_1^i \\ u_2^i \\ 1 \end{bmatrix} x_1^i - \begin{bmatrix} h_{11} & h_{12} & h_{13} \end{bmatrix} \begin{bmatrix} u_1^i \\ u_2^i \\ 1 \end{bmatrix} = 0, \tag{3.4}$$

$$\begin{bmatrix} h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} u_1^i \\ u_2^i \\ 1 \end{bmatrix} x_2^i - \begin{bmatrix} h_{21} & h_{22} & h_{23} \end{bmatrix} \begin{bmatrix} u_1^i \\ u_2^i \\ 1 \end{bmatrix} = 0. \tag{3.5}$$

Za pomoč pri omenjeni metodi računanja homografije, lahko elemente matrike H zapišemo kot vektor h na naslednji način:

$$\mathrm{h} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_3 \end{bmatrix}^T.$$

Sedaj lahko iz n predhodnih parov ustvarimo matriko A in jo uporabimo za tvorbo formule 3.6:

$$\begin{bmatrix} u_1^1 & u_2^1 & 1 & 0 & 0 & 0 & -x_1^1 u_1^1 & -x_1^1 u_2^1 & -x_1^1 \\ 0 & 0 & 0 & u_1^1 & u_2^1 & 1 & -x_2^1 u_1^1 & -x_2^1 u_2^1 & -x_2^1 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ u_1^n & u_2^n & 1 & 0 & 0 & 0 & -x_1^n u_1^n & -x_1^n u_2^n & -x_1^n \\ 0 & 0 & 0 & u_1^n & u_2^n & 1 & -x_2^n u_1^n & -x_2^n u_2^n & -x_2^n \end{bmatrix} h = Ah = 0. \tag{3.6}$$

Matriko H lahko tako izračunamo iz enačbe $A^*h{=}0$. Očitno je, da lahko takoj izluščimo trivialno rešitev $h{=}0$, ki pa nas ne zanima. Tako lahko postavimo

naslednjo omejitev $\|h\|$ =1. Zanima nas taka rešitev, ki minimizira $\|Ah\|$. Opisani problem velja za standardnega v linearni algebri. Rešitev h je lastni vektor, ki pripada najmanjši lastni vrednosti A^TA.

Matrika H ima 8 neodvisnih parametrov (ang. degrees of freedom). V matriki z 9 elementi zadnji parameter predstavlja razmerja med točkami, ki pa se v projekcijskih transformacijah ohranjajo. Matriko H lahko torej pomnožimo s poljubnim številom in to ne bo imelo vpliva na samo transformacijo. Pomembna so tako le razmerja med elementi v matriki. Glede na to, da ima enačba (3.6) 8 neznank, mora imeti matrika A v enačbi najmanj 8 vrstic oziroma naj bi bila izpeljana iz 4 parov ustreznih točk. Za izračun matrike homografije tako potrebujemo najmanj 4 pare točk, ki pa imajo še dodatno omejitev. Ne moremo izbrati poljubno postavitev točk iz dveh ravnin. V primeru, da katerekoli 3 točke ležijo na premici (ang. collinear points), ne bomo dobili ustrezne rešitve. Z potrebe našega algoritma bomo v začetni fazi opazovanja ročno določili vsaj 4 točke na vsakem izmed pogledov. Te točke naj bi sovpadale med seboj med različnimi pogledi. Na podlagi teh točk nato izračunamo ustrezne homografije med vsemi pogledi, kot tudi med pogledom od zgoraj.

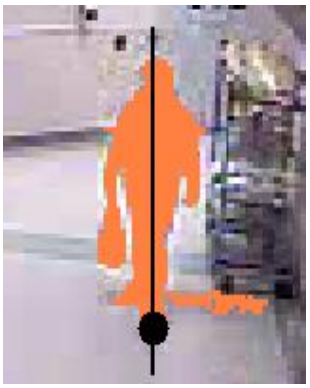
3.2 Glavna os in talna točka človeka (ang. principal axis and ground point)

Pri določanju ujemanja med osebami z različnih pogledov, bomo vsaki osebi opisali dva nova pojma:

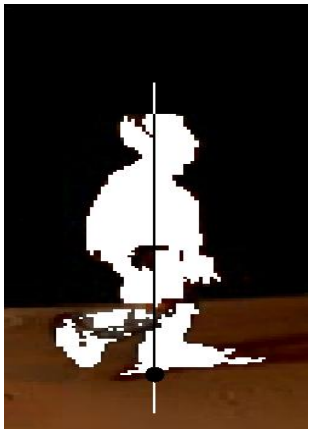
- Talna točka človeka predstavlja najnižjo točko osebe, oziroma gre za točko, kjer naj bi se oseba stikala s tlemi.
- Ob predpostavki, da so gibajoči ljudje v stoječem položaju, lahko trdimo, da je človek simetričen glede na levo in desno stran. Kot takemu mu lahko opišemo glavno os telesa.

Za jasnejšo sliko si lahko ogledamo sliki 3.2 in 3.3.

Pomen glavne osi je, da ima na levi in desni strani enako število slikovnih elementov. Tako jo najlažje izračunamo, če minimiziramo razdaljo vsakega slikovnega elementa do glavne osi (formula 3.7). Glede na to, da je glavna os človeka vedno navpična premica, nas zanima le x koordinata osi. Izračunamo



Slika 3.2: Kljub nekaj napačno klasificiranim slikovnim elementom je glavna os detektirana pravilno.



Slika 3.3: Drugi primer glavne osi in talne točke človeka.

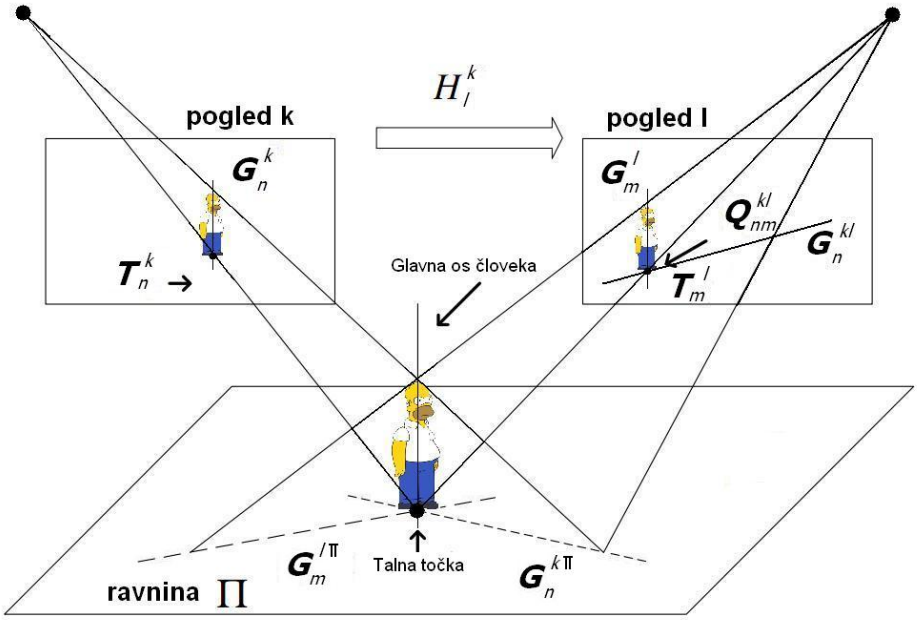
jo kot seštevek vseh x koordinat slikovnih elementov deljeno z njihovim številom.

$$x_G = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{3.7}$$

Glavna os zakritih ljudi se izračuna glede na stopnjo zakritosti. V primeru, da imamo dovolj pravilno klasificiranih slikovnih elementov, se glavna os detektira na trenutnem stanju. Nasprotno jo izračunamo na podlagi pričakovanega položaja iz predhodne informacije iz modela gibanja človeka. Talno točko T bomo preprosto izbrali kot najnižji del človekove glavne osi (slika 3.2). Iz tega sledi, da je x koordinata točke T kar x koordinata glavne osi. Y koordinato pa bomo dobili kot najmanjši y med vsemi pripadajočimi slikovnimi elementi.

3.3 Maksimalna verjetnost ujemanja

Naj kamera k v nekem trenutku t opazuje N oseb. Vsaka izmed oseb ima pripadajočo glavno os G_1^k,G_2^k,...,G_N^k. Kamera l naj v istem trenutku t opazuje M oseb z glavnimi osmi G_1^l,G_2^l,...,G_M^l. Želimo določiti, pri katerih osebah iz različnih kamer gre dejansko za isto osebo. Naš problem je tako poiskati take pare $\{n, m\}$, da bomo maksimizirali verjetnost ujemanja med temi osebami. Problem formuliramo [6]:



Slika 3.4: Slika prikazuje povezavo med različnimi pogledi. Glavna os človeka iz pogleda k je transformirana na pogled l, kjer je ugotovljena povezava med talno točko in sečiščem obeh glavnih osi.

$$\{n, m\} = \arg\max_{n,m} \left\{ \mathrm{L}(G_n^k, G_m^l) \right\}, n \in [1, N], m \in [1, M]. \tag{3.8}$$

L(G_n^k,G_m^l) bomo izračunali po postopku, ki ga bomo opisali sedaj. Za lažjo predstavo glejmo sliko 3.4. Kamera k opazuje osebo n, ki ima glavno os G_n^k. Istočasno določimo tudi najnižjo točko glavne osi T_n^k. Ravno tako so definirani oseba m na kameri l z G_m^l in T_m^l. Glavno os osebe n G_n^k lahko transformiramo s homografijo H_l^k, ki predstavlja transformacijo med obema pogledoma. Tako dobimo transformirano glavno os G_n^{kl} na pogledu iz kamere l. Osi G_n^{kl} in G_m^l se bosta sekali v eni točki. To točko bomo poimenovali točka Q_{nm}^{kl}. V primeru, da gre dejansko za isto osebo, bi morali točka Q_{nm}^{kl} in točka T_m^l sovpadati oziroma biti zelo blizu ena drugi. Razdalja med točkama Q_{nm}^{kl} in točka T_m^l bo predstavljala mero za ujemanje osebe n iz pogleda k in osebe m iz pogleda l. Ob predpostavki, da sta pogleda neodvisna med seboj, lahko z inverzno homografijo (H^{-1}) slikamo tudi os G_m^l na pogled k. Tako dobimo točko Q_{mn}^{lk}. Razdalja med Q_{mn}^{lk} in T_n^k tako predstavlja enako mero za ujemanje oseb med

seboj. Končno mero za verjetnost D_{nm}^{kl} (D_{nm}^{kl} kot razdalja med osebo n iz kamere k in osebo m iz kamere l) izračunamo kot seštevek obeh prej omenjenih razdalj. Idejo opisujeta opisujeta enačbi 3.9 in 3.10.

$$\arg\max_{n,m} \mathrm{L}(G_n^k, G_m^l) \Longleftrightarrow, \arg\min_{n,m} D_{nm}^{kl}, \tag{3.9}$$

$$D_{nm}^{kl} = (T_n^k - Q_{mn}^{lk})\,(T_n^k - Q_{mn}^{lk})^T + (T_m^l - Q_{nm}^{kl})\,(T_m^l - Q_{nm}^{kl})^T. \tag{3.10}$$

3.4 Določanje položaja osebe na pogledu od zgoraj

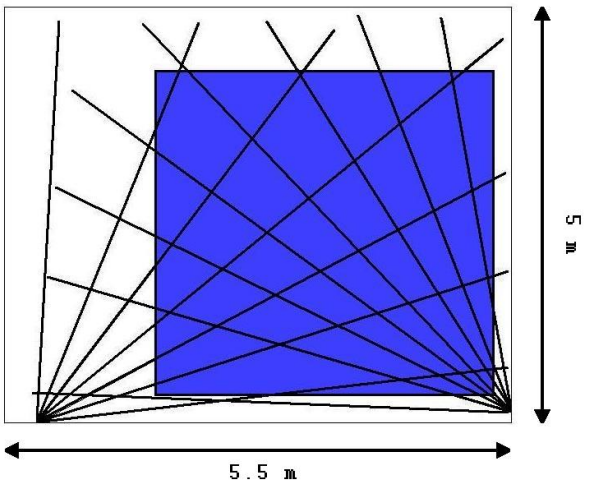
Z informacijo o nahajanju določene osebe na različnih pogledih lahko določimo njen položaj na pogledu od zgoraj. Glavne osi osebe transformirano z ustrezno homografijo med vsemi pogledi in pogledom od zgoraj Π. Na sliki 3.4 vidimo transformirane glavne osi kot črtkane črte na ravnini Π. Sečišče med njimi naj bi predstavljalo položaj nahajanja osebe. Pogosto se zgodi, da se glavne osi ne stikajo v eni točki. V tem primeru položaj lahko določimo kot točko, ki ima najmanjšo vsoto razdalj do vseh sečišč transformiranih osi.

V tem poglavju smo razširili sistem z eno kamero na sistem z več kamerami. Opisali smo, na kakšen način je možno združiti informacijo iz več kamer v celoto. Zmožni naj bi bili natančneje določiti nahajanja posameznih oseb na prizoru, ki ga opazujemo. V naslednjem poglavju bomo prikazali rezultate in ovrednotili delovanje sistema.

Poglavje 4

Rezultati

Za testiranje sistema smo uporabili dve kameri. Opazovali smo notranjost sobe v velikosti okoli 5.5m x 5m. Osvetljenost prostora se je spreminjala malo oziroma nič. Kameri sta bili pozicionirani približno dva metra nad tlemi. Postavitev kamer v prostoru ponazarja slika 4.1. Za jasnejšo predstavo smo obarvali del slike modro, kar ponazarja modro preprogo v prostoru. Čas sledenja je bil okoli 30 sekund, oziroma dokler sledenje ni odpovedalo.



Slika 4.1: Postavitev kamer na testni sceni.

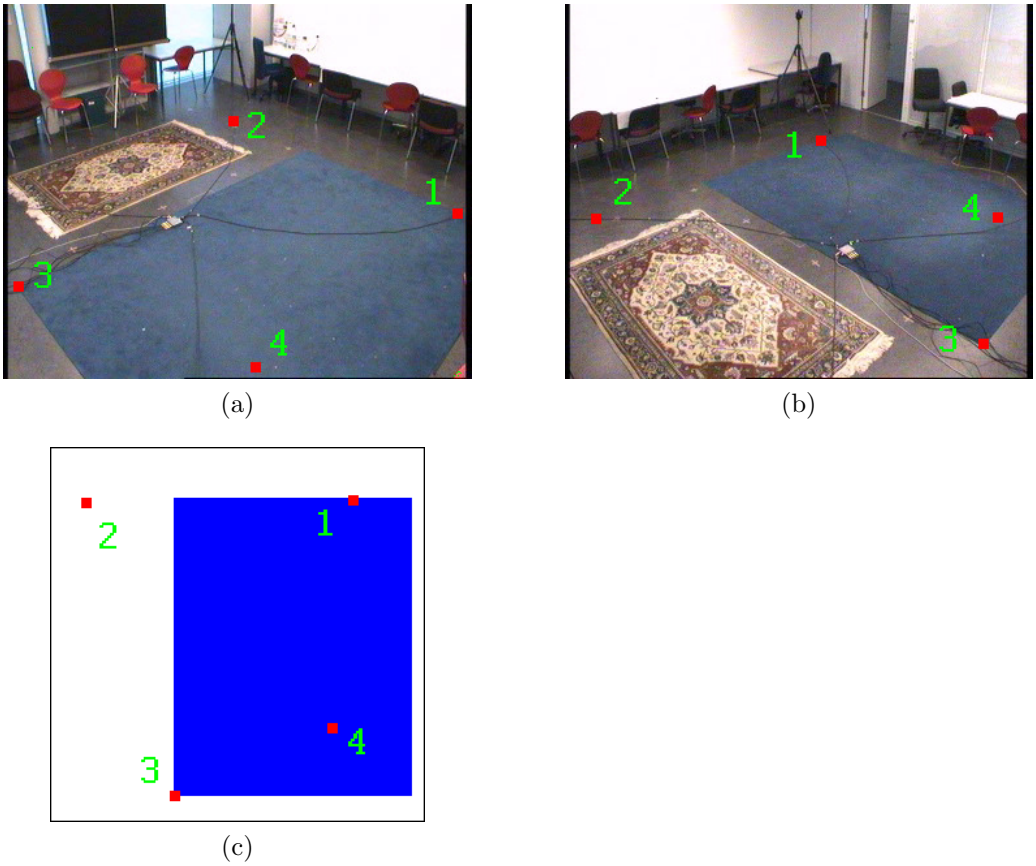
Velikost posnetkov je 360x288 slikovnih elementov. Testi so bili izvedeni na računalniku z 2.0 gigaherčnim procesorjem in 2.0 gigabajtnim pomnilnikom. Obdelava enega posnetka iz ene kamere je trajala okoli 2.3 sekunde. Algoritem

je deloval s parametri, ki so navedeni v tabeli 4.1. Vpliv parametrov na sistem je natančno opisan v predhodnih poglavjih, kratkek opis pa je podan tudi v spodnji tabeli.

Parameter	Formula	Opis parametra	Vrednost parametra
K_1		število gaussovih porazdelitev v mešanici slikovnega elementa	3
α_1	2.7, 2.8, 2.9, 2.10, 2.11, 2.12	hitrost integracije trenutne vrednosti slikovnega elementa v mešanici gaussov ozadja	0.01
		največje možno odstopanje za pripadanje določeni gaussovi porazdelitvi	$2.5\sigma^2$
β	2.15	teža gaussovih porazdelitev v mešanici slikovnega elementa, ki so označeni kot ozadje	0.75
σ^2		fiksno nastavljena varianca gaussove porazdelitve	20
		širina območja okoli mejnega okvirja, kjer verjetnost linearno pada	10
		minimalna velikost regij	15 slikovnih elementov
		minimalna velikost območja pri inicializaciji objekta	1500 slikovnih elementov
		največja razdalja med regijami	20 slikovnih elementov
		število rezin osebe	7
K_2		število gaussovih porazdelitev v mešanici ene rezine človeka	3
		največja razdalja med regijami	20 slikovnih elementov
α_2	2.16, 2.17	hitrost učenja in spreminjanje mejnega okvirja	0.5
		število točk za izračun homografije	4
D_{nm}^{kl}	3.10	prag za ujemanje oseb z različnih pogledov med seboj	60 slikovnih elementov

Tabela 4.1: Vsi parametri uporabljeni v sistemu in njihove vrednosti.

Izračun homografij je bil narejen z minimalnim številom (4) točk na vsakem pogledu (vključno s pogledom od zgoraj). Izbor točk na vsake pogledu predstavlja slika 4.2.



(a) (b)

(c)

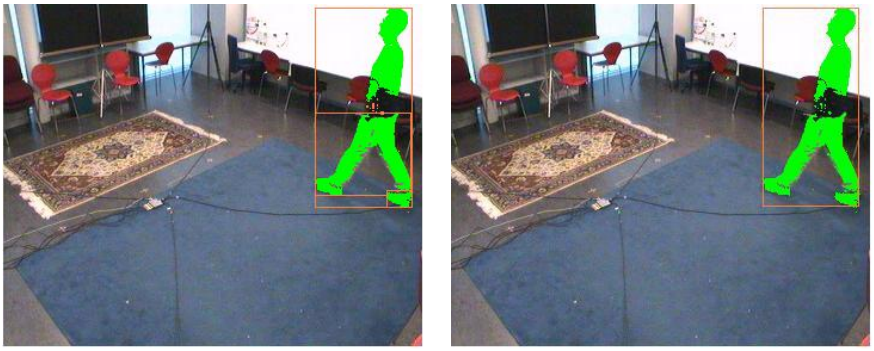
Slika 4.2: Točke za izračun homografije.

Cilj eksperimenta je bil:

- Oceniti zmožnosti implementiranega algoritma.
- Določiti maksimalno število oseb, ki jim lahko sledimo.
- Točneje ovrednoditi delovanje posameznih delov algoritma.
- Identificirati probleme, ki povzročajo zaustavitem sistema in izgubo sledenja.

Vsak slikovni element (ozadje) je modeliran s parametri K_1, α_1 in β iz tabele 4.1. Za lažje razumevanje nastavljenih parametrov lahko povemo, da bi bil nov objekt v primeru ustavitve na določenem mestu označen kot del ozadja v približno 30 posnetkih. Razločno se pokaže delovanje zaznavanja novih objektov na prizoru. Slikovni elementi, ki se dovolj razlikujejo od ozadja, so točno označeni kot del novega objekta. Napaka se je pojavila (kot je pričakovati) le pri slikovnih elementih, kjer je imel nov objekt približno iste vrednosti kot ozadje. Na sliki 4.3 se to nazorno vidi kot del človeka, ki je označen kot črno ozadje. Celoten interval za pripadanje gaussovi porazdelitvi je kar 50. Kljub temu, da bi človeško oko določilo ozadje za razmeroma nespreminjajoče, se posamezne vrednosti na opazujočem slikovnem elementu drastično spreminjajo. Od tod sledi razmeroma visoka toleranca za pripadanje slikovnega elementa ozadju.

Na sliki 4.3 predstavimo tudi združevanje slikovnih elementov v območja. Metoda je močno odvisna od klasifikacije slikovnih elementov, saj v primeru slabe oziroma nezadostne klasifikacije tudi združevanje povsem odpove. Parametri za združevanju slikovnih elementov so bili nastavljeni primerno prizoru, ki smo ga opazovali. Pomemben parameter je bil najmanjša velikost območja pri inicializaciji osebe. Nastavljen je bil na 1500 slikovnih elementov, saj smo pričakovali, da bodo osebe zavzemale precejšen del slike. Pri neprimerno nastavljenem parametru je sledenje delovalo negotovo, saj se je oseba inicializirala, ko smo videli le del njenega telesa.

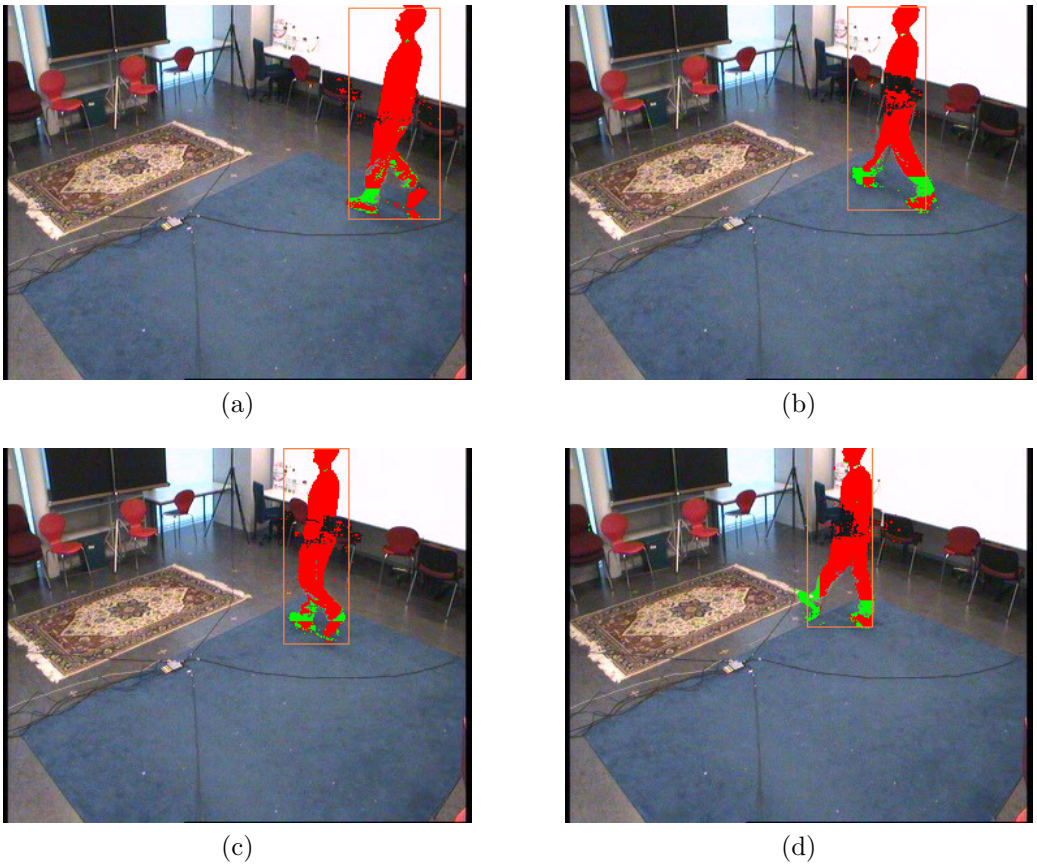


(a) (b)

Slika 4.3: Detekcija novih objektov in združevanje slikovnih elementov v območja. Vsaka regija ima svoj mejni okvir (slika 4.3a). Po združevanju regij dobimo mejni okvir celotnega območja (slika 4.3b).

Za pričujoč prizor, kjer ljudje zavzemajo dobršen del prostora/slike, smo se

odločili za 7 rezin. Telo osebe, ki ji sledimo, smo želeli razdeliti, tako da bi posamezna rezina zavzemala en del človeka. Ta del pa bi bil v glavnem sestavljen le iz ene ali največ treh barv. Ena rezina je tako zavzemala glavo, ena stopala, ostale pa so razdeljene na isto velike dele znotraj telesa.



(a) (b)

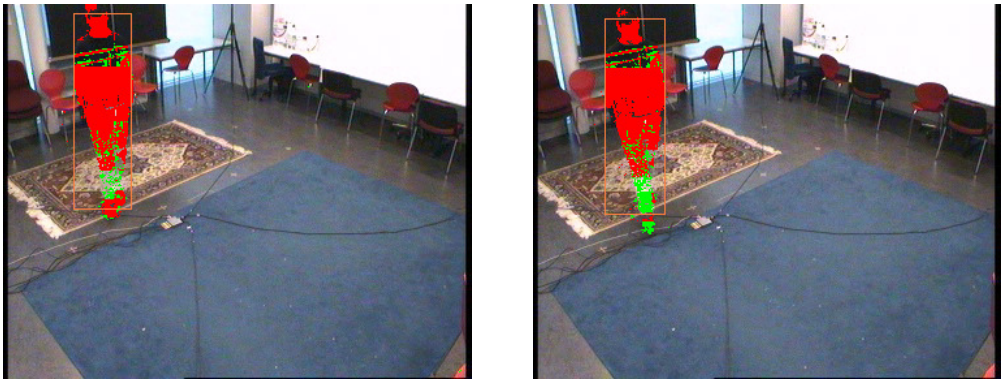
(c) (d)

Slika 4.4: Prikaz klasifikacije slikovnih elementov 7-delnega modela človeka. Zelena barva predstavlja slikovne elementev, ki pripadajo novemu predmetu, in rdeča barva predstavlja sledeno osebo. Na sliki so nekateri deli označeni zeleno, vendar so kasneje dodani sledeni osebi.

Na sliki 4.4 je prikaz klasifikacije slikovnih elementov oseb s 7 rezinami. Na prizoru je bila prisotna ena oseba. Slikovni elementi pripadajoči osebi so označeni rdeče, slikovni elementi, ki naj bi pripadali novemu objektu pa zeleno. 7-delni model človeka deluje zadovoljivo. Vendar se pri tem pojavita dva problema. Ko se človek giblje, se položaj njegovih delov spreminja. Tako ni več nujno, da se določen model rezine prilega delu telesa, iz katerega je bil ustvarjen. To se

vidi pri napačni klasifikaciji spodnjega dela noge človeka iz slike 4.4. Napačno klasificirani slikovni elementi so označeni zeleno, torej kot nov objekt, saj naj ne bi pripadali nobenemu obstoječemu modelu. Nekje je napačen model na sedmi, zadnji rezini (slika 4.4a, b in d) človeka. Na sliki 4.4c pa je neustrezen model na šesti rezini. Napačna klasifikacija se jasno vidi po (vodoravni) meji med dvema rezina. Drugi problem pa nastopi zaradi napačnega predvidevanja položaja človeka. Domnevni položaj človeka na naslednjem posnetku je odvisen od hitrosti in velikosti (mejnega okvirja) človeka. Tako se pogosto zgodi, da se deli človeka pri gibanju (predvsem roke in noge) ne nahajajo v predvidenem okvirju položaja osebe. Sicer okoli okvirja človeka obstaja mejno območje, kjer verjetnost pripadanja človeku linearno pada. Vendar nam to v takih primerih ne pomaga. V kolikor sledimo le izoliranemu človeku, oziroma ta človek ni v zakritju z ostalimi osebami lahko del, ki izstopa, preprosto integriramo najbližjemu človeku. Pri zakritju pride do težav, saj preprosto ne morem določiti, kateremu človeku omenjeni del pripada. Za ta namen smo privzeli, da ta del dodamo k prvi osebi, s katero se mejna okvirja stikata. Na sliki 4.4 so tudi jasno vidni nekateri slikovni elementi, ki so napačno klasificirani. Vendar so le-ti v manjšini. Regije, katere tvorijo, pa zaradi minimalne velikosti regij niso upoštevane.

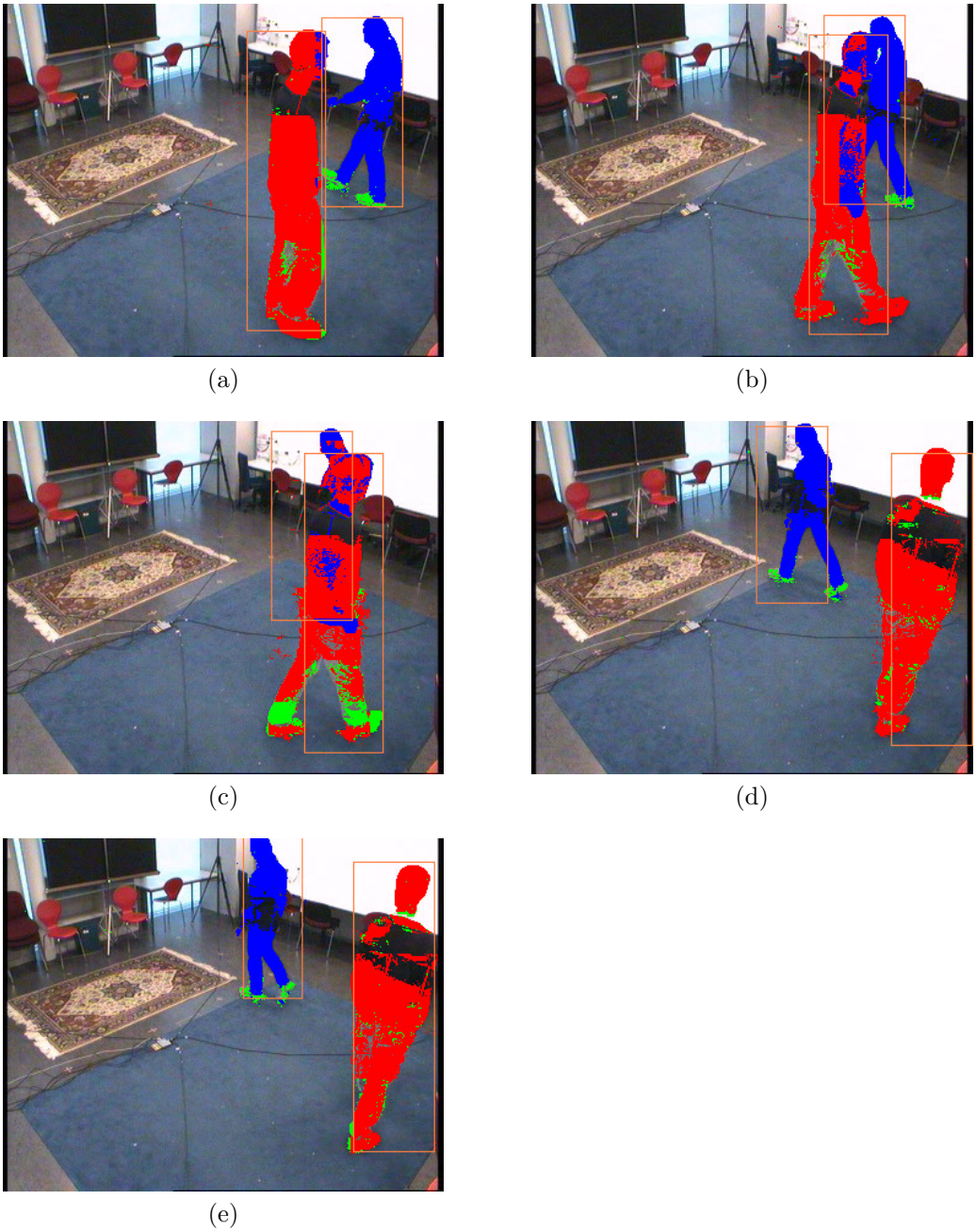
V primeru, da imata ozadje in premikajoč predmet razmeroma isti barvni model, pride do občutnega poslabšanja klasifikacije slikovnih elementov. Na sliki 4.5 se razločno vidi, da so slikovni elementi napačno klasificirani v predelu ramen in golena človeka. Mešanice gaussov v posameznih rezinah zavzemajo iste vrednosti kot posamezni slikovni elementi ozadja. Zadostna večina slikovnih elementov je pravilno klasificirana, ki tvorijo blobe s primerno velikostjo. V tej situaciji tako ne zgubimo sledenja človeku in le-to se lahko nadaljuje neovirano.



(a) (b)

Slika 4.5: Primer slabše klasifikacije slikovnih elementov.

Za delovanje v primeru zakritju oseb smo implementirali metodo predlagano v članku [14]. Problem pri tej metodi nastopi pri nezadovoljivem določanju položaja osebe, ki je bila zakrita. Pri testiranju našega prizora se je zgodilo, da je sledilnik izgubil sled za zakrito osebo. Vzrok teh težav tiči v dejstvu, da opazujemo prizor, kjer osebe zavzemajo precejšen del slike. Tako pogosto pride do popolnega zakritja. Kot smo opisali v podpoglavju o zakritjih, je takrat sledilnik popolnoma odvisen od domnevnega položaja, ki ga izračunamo s pomočjo smeri in hitrosti gibanja človeka pred zakritjem. Na sliki 4.6 se kaže delovanje sledilnika, ko je algoritem pravilno predvideval gibanje zakritega človeka. Sama klasifikacija pri zakritju je odvisna le od barvnih modelov oseb, ki se zakrivajo. Če sta modela med dvema človekova preveč podobna, je verjetnost, da bi bila sama klasifikacija na podlagi mešanice gaussov zadovoljiva, izredno majhna. V takih primerih sledenje brez predvidevanja položaja sledenje odpove. V kolikor oseba drastično spremeni smer gibanja v času prekrivanja pa sledenje odpove v vseh primerih.

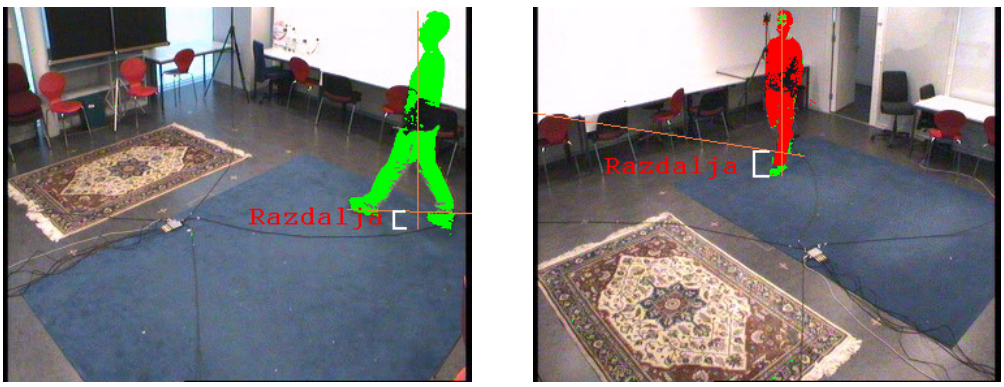


(a) (b) (c) (d) (e)

Slika 4.6: Delovanja algoritma v primeru zakrivanja.

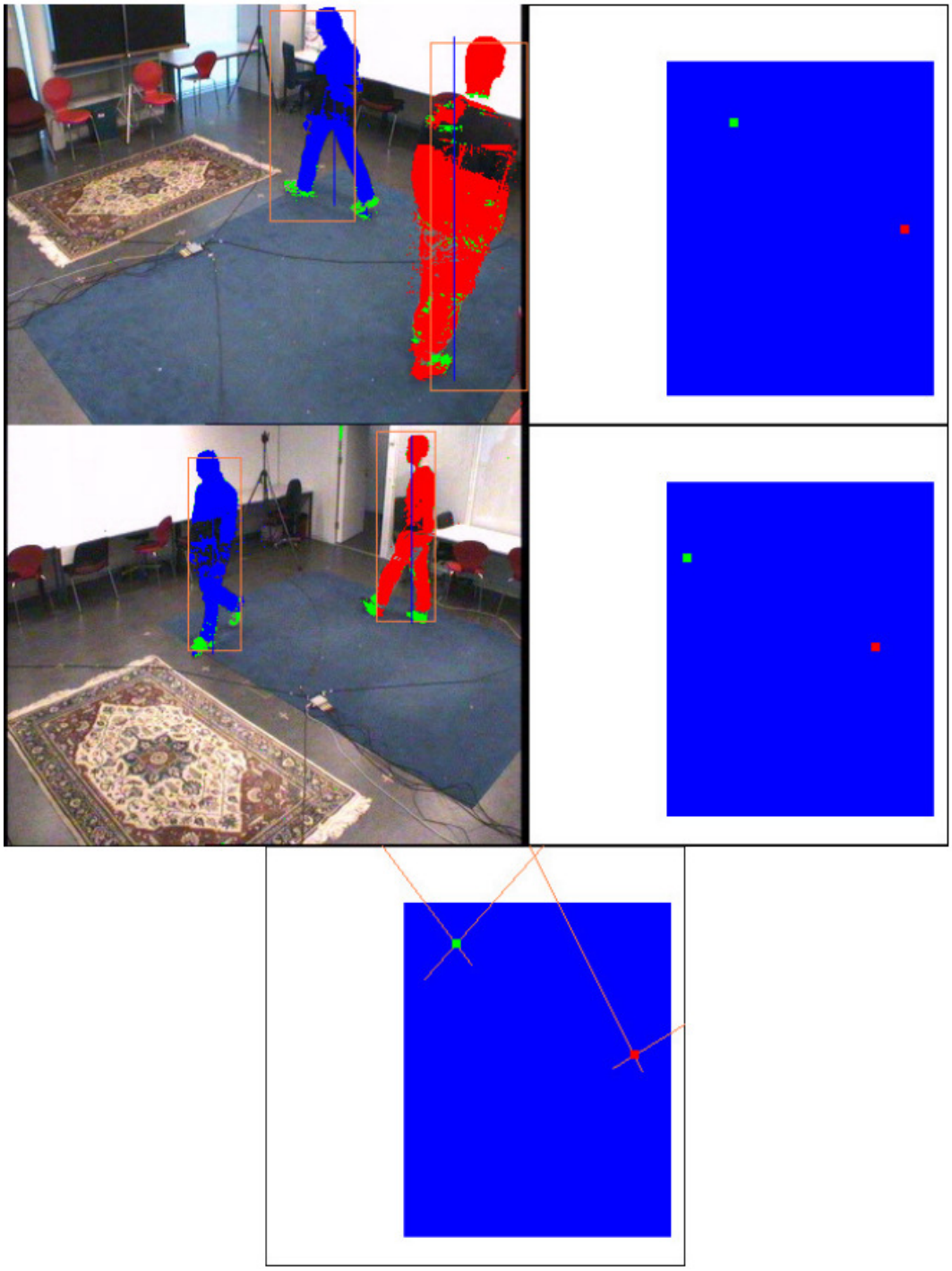
Ne glede na gibanje osebe ali smer, iz katere gledamo, smo uspešno določili glavno os osebe (slika 4.7). Tudi ob zelo skromni klasifikaciji slikovnih elementov smo bili zmožni ustrezne določitve glavne osi. Sicer bi taka klasifikacija vplivala na kasnejšo izgubo sledenja, vendar bi bila do takrat glavna os venomer pravilno določena. Talna točka človeka je bila preprosto označena kot najnižja točka glavne osi. Problem se je pojavil pri določitvi talne točke. Vzroka za nastal problem sta dva: prvi je, da se ponavadi na tleh pojavljajo sence samega človeka, kar pripomore k napačni klasifikaciji slikovnih elementov. Ponavadi so ti slikovni elementi dodani k osebi, kar jo v bistvu naredi večjo kot je v resnici. Drugi vzrok za napačno zaznavanje talne točke predstavlja sama postavitev kamer. Kamere so tipično postavljene nekaj metrov nad tlemi. Talna točka je tako bližje kameri kot je v resnici.

Prag med talno točko in sečiščem glavne osi človeka s transformirano glavno osjo z drugega pogleda je bil nastavljen na 60 slikovnih elementov. Morda se to sprva sliši veliko, vendar je potrebno vedeti, da je to seštevek dveh razdalj, saj upoštevamo tudi razdaljo pri obratnem transformiranju. Za testni prizor je bila povprečna razdalja med obema točkama 30 slikovnih elementov, kar znaša le 15 slikovnih elementov na enem pogledu. To se tudi nazorno vidi na sliki 4.7. Iskanje ujemanja oseb smo izvajali vedno, ko je na poljubnem pogledu obstajala oseba, ki še ni imela ujemanja z osebami na drugih pogledih. Ko je bilo ujemanje vzpostavljeno, korak iskanja ujemanja ni bil več potreben.



(a) (b)

Slika 4.7: Izračun verjetnosti ujemanja ljudi med pogledi. Na obeh slikah se jasno vidita transfomirana in obstoječa glavna os človeka.



Slika 4.8: Končni rezultat sledenja ljudem.

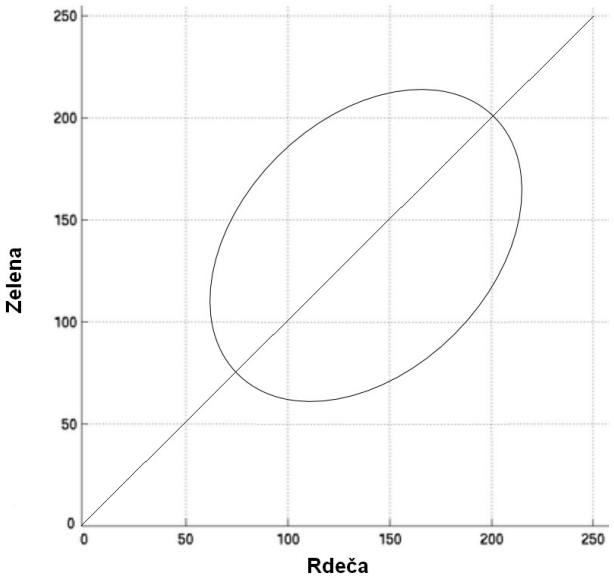
Na sliki 4.8 prikazuje končno preslikanje položajev ljudi na pogled od zgoraj. Na levi strani je prikazano sledenje ljudem vključno s klasifikacijo slikovnih elementov in mejnega okvirja, medtem ko desna stran predstavlja položaj oseb na podlagi informacije z ene kamere. Spodnji del slike pa je dobljen z uporabo dveh kamer. Ko so ugotovljene povezave med posameznimi osebami z drugega pogleda, smo uporabili obe glavni osi za določitev točnega položaja. Na sliki 4.8 se kaže boljše določanje položaja osebe z informacijo obeh kamer v primerjavi z eno, kjer transformiramo le talno točko. Vzrok tiči v dejstvu, da je končna lokacija osebe odvisna izključno od glavnih osi iste osebe na obeh pogledih. Z uporabo križanja transformiranih glavnih osi se težavam glede talne točke ognemo v celoti.

Poglavje 5

Sklepne ugotovitve in smernice za izboljšave

Na prizoru, ki smo ga uporabili za testiranje v prejšnjem poglavju, je bil sistem zmožen slediti dvema osebama (tudi v času zakritja). V primeru, da smo sledili več osebam, je sistem povsem odpovedal, predvsem zaradi dejstva, da so osebe zavzemale večji del slike. Onemogočena je bila inicializacija barvnega modela osebe. Predpostavljamo namreč, da za inicializijo osebe potebujemo vsaj dva posnetka (enega za inicializacijo barvnega modela in enega za inicializacijo gibanja) osebe, ki stoji sama. Le v tem primeru smo sposobni dobiti celotno informacijo o osebi. V našem prizoru pa je bilo temu dejstvu težko ugoditi. Ustreznejši bi bili prizori, kjer opazujemo ljudi od daleč. V tem prizorih obstaja večja možnost, da novo osebo opazimo izolirano od drugih oseb.

Največja slabost je delovanje algoritma v primeru zakritja več oseb. Predlagana metoda uspešno deluje le pri prekrivanju med dvema osebama. Ko se na prizoru pojavi več ljudi, se število možnih stanj obravnavanja zakritja povečuje eksponentno. Takrat je sledilnik odvisen le od napovedanega stanja osebe. Napovedano oziroma pričakovano stanje pa se izračuna glede na gibanje osebe pred tem. V kolikor oseba v zakritju spremeni smer, se sledenje ustavi in slednilnik odpove. Velik problem povzroča tudi prevelika podobnost med barvnimi modeli oseb. Glede na dejanske vrednosti RGB se ti ne razlikujejo veliko eden od drugega. Realni svet je v glavnem sestavljen iz sive barve in njenih odtenkov. Prevladujoči parameter pa je intenzivnost osvetlitve. To dejstvo nazorno prikazuje slika 5.1.



Slika 5.1: Razporeditev barv v realnem svetu. Barvni odtenki v naravi se v glavnem nahajajo blizu premice s kotom 45° in znotraj elipse. Tudi ko se človeku zdi, da gre za na primer rdečo barvo, se bo ta na tej sliki odražala na robu elipse. Za večjo stopnjo razlikovanja med modeli pa bi potrebovali vrednosti na samem robu grafa (okoli [255,0,0] ali [0,255,0]).

Za rešitev tega problema bi lahko uporabili drugačen model predstavitve posameznih slikovnih elementov kot na primer Hue-Saturation-Value ali HSV barvni model (barvni odtenek-intenzivnost-vrednost). Druga možnost pa je normalizacija barv $\frac{[R,G,B]}{\sqrt{R^2+G^2+B^2}}$ ali drugi način $r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}$, $s = \frac{R+G+B}{3}$. Res je, da bi program bolje deloval v primeru, ko bi opazovali ljudi, ki so med seboj barvno različni. To bi nam omogočilo boljše razlikovanje v času zakritja. Najboljša rešitev za izboljšanje delovanja v času zakritja pa je izračun najverjetnejšega položaja zakrite osebe iz ostalih kamer, kjer je oseba vidna v celoti. Za omogočenje te ideje bi morali vzpostaviti višjo stopnjo integracije med kamerami, kar pa ni bila naša izhodna predpostavka, saj smo oblikovali sistem v katerem vsaka kamera deluje kot samostojna enota. Poskusili bi z uporabo metod za obravnavanje zakritja ([1, 3, 7]), vendar tudi pri teh idejah kaj kmalu pridemo do primerov, kjer odpovedo. V prizorih, kot je bil naš, bi teoretično lahko sledili štirim do petim ljudem. Pri več ljudeh pogosto pridemo do situacije, ko ne vidimo določene osebe z nobeno kamero.

Manjše napake se pojavljajo tudi pri samih transformacijah glavnih osi med pogledi. Sam izračun homografije minimizira napako vseh preslikanih točk, katere smo uporabili pri njenem izračunu. Z večjim številom točk se zmanjšuje napaka homografije. Samo izbiro točk bi lahko povsem avtomatizirali z algoritmom za razpoznavo posebnih točk, ki imajo določene edinstvene poteze (ang. local points of interest), med različnimi pogledi. Eden takih pristopov je SIFT (ang. Scale-invariant feature transform) predstavljen v [8]. Algoritem sicer ne deluje stoodstotno, saj niso vsi pari med poljubnima pogledoma identificirani pravilno. Z namenom odstranitve takih parov lahko uporabimo Random Sample Consensus ali krajše RANSAC [5]. RANSAC je zmožen izbrati ustrezne pare iz množice, tudi ko le-ta vsebuje večino „slabih“ parov. Višjo natančnost določanja položaja oseb pa bi lahko dosegli tudi z višjim številom kamer.

Sicer sama hitrost delovanja aplikacije za nas ni bila v glavnem planu, vendar lahko na tem mestu spregovorimo nekaj besed o možnosti izboljšav na tem področju. Program je potreboval okoli 2.3 sekunde za obdelavo ene slike iz ene kamere, kar nam je povzročalo velike preglavice pri odpravljanju napak v programu, saj so se le-te pojavile po več sto obdelanih slikah. Računsko najzahtevnejša dela sta vzdrževanje mešanice gaussov za vsak slikovni element posebej in matrično računanje verjetnosti pripadanja slikovnega elementa glede na modele oseb in trenutno vrednost slikovnega elementa. Za zmanjšanje časa računanja bi lahko zmanjšali število gaussov v mešanici gaussov vsakega slikovnega elementa. Ravno tako bi namesto uporabe knjižnice za računanje z matrikami uporabili preprosto seštevanje in množenje 3x3 matrike. Še največjo pohitritev pa bi dosegli bržkone v primeru, da ne bi računali pripadanja vsakega slikovnega elementa posebej. Namesto tega bi to lahko storili le na vsakem drugem slikovnem elementu, vmesni element pa bi dodali najverjetnejšemu sosedu glede na izgled. Ravno tako bi lahko izpustili vsako drugo sliko v seriji brez pretirane spremembe v rezultatih. Z razširitvijo na več, med seboj povezanih računalnikov, pa bi mogoče sistem deloval tudi v realnem času.

Zanimiv pristop bi bil tudi iskanje regij, katerih deli imajo isto barvno strukturo. Označili bi torej skupke slikovnih elementov, ki so si med seboj podobni. Celotna regija bi bila lahko klasificirana na podlagi povprečne vrednosti. Eden takih algoritmov je MSER, opisan v [10], kjer dokaj uspešno najdemo take elemente slike. Celotna informacija o človeku bi bila zgrajena na podlagi teh regij.

Celotna zasnova sistema kaže obetavne rezultate. Po našem mnenju najzahtevnejše težave tičijo v zakrivanju ljudi med seboj in v sledenju velikega števila ljudi. Tovrstni algoritmi so usmerjeni zelo ozko in delujejo le v točno določenih situacijah. Čeprav je bilo na to temo izdanih že mnogo člankov, bi lahko rekli, da smo še vedno v začetni stopnji odkrivanja vseh možnosti uporabe sledenja ljudem in predmetom. Morda pa bo to delo dodalo majhen kamenček v mozaik tega področja raziskovanja.

Literatura

[1] J. Black, T. Ellis in P. Rosin, "Multi View Image Surveillance and Tracking", v zborniku *Workshop on Motion and Video Computing*, dec. 2002, str. 169-174.

[2] M. B. Capellades, D. Doermann, D. DeMenthon in R. Chellappa, "An appearance based approach for human and object tracking", *International Conference on Image Processing*, sept. 2003, str. zv.2 85 - zv.3 8.

[3] T. Chang, S. Gong in E. Ong, "Tracking Multiple People under Occlusion Using Multiple Cameras", *British Machine Vision Conference*, sept. 2000, str. 11-14 .

[4] A.P. Dempster,N.M. Laird in D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society*, Series B, nov. 1977, str. 1–38.

[5] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Comm. of the ACM 24*, jun. 1981, str. 381–395.

[6] M. Hu, J. Lou, Weiming Hu, and Tieniu Tan, "Multi-camera correspodence based on principal axis of human body", *International Conference on Image Processing*, okt. 2004, str. zv.2 1057-1060.

[7] K. Kim in Larry S. Davis, "Multi-camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering", *European Conference on Computer Vision*, 2006, str. vol.3 98-109.

[8] D. G. Lowe, "Object recognition from local scale-invariant features", v zborniku *International Conference on Computer Vision 2*, 1999, str. 1150–1157.

[9] A.M. McIvor, "Background Subtraction Techniques", v zborniku *Image and Vision Computing*, Auckland, Nova Zelandija, 2000, str. 147–153.

[10] J. Matas, O. Chum, M. Urba in T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions", v zborniku *British Machine Vision Conference*, 2002, str. 384-396.

[11] D. Migliore, M. Matteucci, M. Naccari in A. Bonarini, "A revaluation of frame difference in fast and robust motion detection", v zborniku *4th ACM international workshop on Video surveillance and sensor networks*, 2006, str. 215-218.

[12] A.Mittal in L. S. Davis, "M2tracker: Amulti-view approach to segmenting and tracking people in a cluttered scene using region-based stereo", *European Conference on Computer Vision*, 2002, str. 18-33.

[13] Orwell in P. Remagnino, G. A. Jones, "Multi-Camera Color Tracking", v zborniku *2ed IEEE Workshop on Visible Surveillance*, 1999, str. 14-24.

[14] D. Roth, P. Doubek in L. V. Gool, "Bayesian Pixel Classification for Human Tracking", *IEEE Workshop on Motion and Video Computing*, jan. 2005, str. 78-83.

[15] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti in R. Bolle, "Appearance Models for Occlusion Handling", *2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, dec. 2001.

[16] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking", *Computer Vision and Pattern Recognition*, 1999, str. vol. 252.

[17] A. Utsumi, H. Mori, J. Ohya in M. Yachida, "Multiple-Human Tracking Using Multiple Cameras", *3rd IEEE International Conference on FG*, 1998, str. 498-503.

[18] Y. Wu, T. Yu in G. Hua, "Tracking Appearances with Occlusions", *Computer Vision and Pattern Recognition*, jun. 2003, str. 789-795.

[19] A. Zisserman in R. Hartley, *Multiple view geometry in computer vision*, Cambridge University, Cambridge, 2nd edition, 2003.

[20] Spletna stran Hawk Eye, Dostopno na:
http://www.hawkeyeinnovations.co.uk/