

**UNIVERZA VLJUBLJANI**  
**FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO**

Borut Kumperščak

**UPORABA METOD RAČUNALNIŠKEGA VIDA ZA DELNO**  
**AVTOMATIZIRANO ANALIZO MEDSEBOJNIH RAZDALJ**  
**PTIC V MIROVANJU**

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Ljubljana, 2010



Št. naloge: 01677/2010

Datum: 14.06.2010

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

Kandidat: **BORUT KUMPERŠČAK**

Naslov: **UPORABA METOD RAČUNALNIŠKEGA VIDA ZA DELNO  
AVTOMATIZIRANO ANALIZO MEDSEBOJNIH RAZDALJ PTIC V  
MIROVANJU**  
**COMPUTER VISION METHODS APPLIED TO SEMI-AUTOMATED  
ANALYSIS OF INTER-INDIVIDUAL DISTANCES OF PERCHING BIRDS**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V sodobnem svetu prihaja do prepletanja in medsebojnega povezovanja različnih sistemov, ki avtonomno ali v sodelovanju zbirajo in procesirajo podatke, z namenom pridobivati oz. zagotavljati informacije za podporo uporabnikom v različnih okoljih. Na kratko temu pravimo vseprisotno računalništvo (angl. ubiquitous computing). Vir podatkov so tipično digitalne kamere, naj bodo ti v obliki statičnih digitalnih fotografij ali niza zajetega v obliki video posnetka. Procesiranje podatkov pa je zasnovano na tehnologiji računalniškega vida.

Preučite odprtokodno knjižnico OpenCV za podporo računalniškemu vidu. V diplomski nalogi kot prikaz uporabe razvijte aplikacijo, ki bo omogočala delno avtomatizirano analizo digitalnih fotografij ptic v mirovanju. Aplikacija naj bo dovolj splošna, da ni odvisna od okolja, v katerem je bil posnetek zajet, uporabniku naj dopušča kontrolo nad parametri aplikacije. Osredotočite se na detekcijo pozicije ptic, njihovo velikost in medsebojne oddaljenosti.

Mentor:

doc. dr. Iztok Lebar Bajec



Dekan:

prof. dr. Franc Solina

**UNIVERZA VLJUBLJANI**  
**FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO**

Borut Kumperščak

**UPORABA METOD RAČUNALNIŠKEGA VIDA ZA DELNO**  
**AVTOMATIZIRANO ANALIZO MEDSEBOJNIH RAZDALJ**  
**PTIC V MIROVANJU**

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor: doc. dr. Iztok Lebar Bajec

Ljubljana, 2010

## **POVZETEK**

Z uporabo tehnologije računalniškega vida in knjižnice OpenCV je realiziran sistem, ki omogoča analizo slik ptic v mirovanju. Vhodni podatki so visoko ločljive slike ptic, ki sedijo na žicah, z uniformnim ozadjem neba oz. so ptice v mirovanju na kakem drugem uniformnem ozadju. Z operacijami računalniškega vida se iz vsake slike izločijo objekti, za katere z razumno verjetnostjo sklepamo, da so ptice. Nad temi objekti se opravijo analitične meritve na način, kot so se prej izvajale ročno, in se zapišejo v podatkovno bazo. Dobljeni rezultati so v pomoč pri gradnji modelov digitalnih in avtonomnih agentov na področju upravljanja skupin objektov.

## **KLJUČNE BESEDE**

računalniški vid, OpenCV, digitalna obdelava slik, segmentacija, analiza objektov, ptice, avtonomni agenti

## **ABSTRACT**

A system for analysis of perching bird flocks based on computer vision technologies and OpenCV library. Input data are high resolution photos of perching birds with uniform background. Number of birds, their sizes, and positions is computed out of these images. First, the objects are found that are reasonably assumed to be birds and segmented out of the background by means of computer vision methods. Analytic measurements are made on these objects, and results are saved to a database. Gathered data supports research in the field of modelling flocks of autonomous agents.

## **KEYWORDS**

computer vision, OpenCV, digital image processing, segmentation, object analysis, birds, autonomous agents

## **ZAHVALA**

Zahvaljujem se mentorju doc. dr. Iztoku Lebar Bajcu za motivacijo, strokovno vodstvo in pomoč pri izvedbi diplomskega dela.

## KAZALO VSEBINE

|                                                                                                                  |    |
|------------------------------------------------------------------------------------------------------------------|----|
| KAZALO VSEBINE.....                                                                                              | I  |
| 1 UVOD .....                                                                                                     | 1  |
| 2 IZBRANA ORODJA.....                                                                                            | 4  |
| 2.1 Izbira programskega jezika.....                                                                              | 4  |
| 2.2 Knjižnica OpenCV.....                                                                                        | 4  |
| 2.3 Programsko ogrodje OpenFrameworks .....                                                                      | 6  |
| 3 OBNAŠANJE PTIC V JATAH.....                                                                                    | 8  |
| 4 PRISTOP K REŠITVI.....                                                                                         | 13 |
| 4.1 Vhodni podatki.....                                                                                          | 14 |
| 4.2 Obdelava slike.....                                                                                          | 16 |
| 4.3 Izločanje objektov – segmentacija.....                                                                       | 18 |
| 4.4 Razpoznavna.....                                                                                             | 21 |
| 4.4.1 Obdelava obrisov .....                                                                                     | 22 |
| 4.4.2 Obdelava barvnih slik objektov.....                                                                        | 23 |
| 4.4.3 Sestavljeno znanje.....                                                                                    | 24 |
| 4.5 Uporaba pridobljenih informacij.....                                                                         | 25 |
| 4.5.1 Število ptic na fotografiji.....                                                                           | 26 |
| 4.5.2 Velikost posamezne ptice v pikslih in skupna povprečna velikost ptic.....                                  | 26 |
| 4.5.3 Medsebojne oddaljenosti ptic in povprečna medsebojna oddaljenost, merjeno v povprečni velikosti ptice..... | 26 |
| 4.5.4 Razvrščanje ptic na odrasle in mladiče glede na obarvanost perja .....                                     | 27 |
| 5 REZULTATI.....                                                                                                 | 28 |
| 6 ZAKLJUČEK.....                                                                                                 | 34 |
| 6.1 Možne izboljšave.....                                                                                        | 35 |
| 6.1.1 Povečanje seta vhodnih podatkov.....                                                                       | 35 |
| 6.1.2 Izboljšava posameznih korakov.....                                                                         | 35 |
| 6.1.3 Delovanje v realnem času.....                                                                              | 36 |
| 6.1.4 Uporaba 3D kamere.....                                                                                     | 36 |
| 7 SEZNAM UPORABLJENIH VIROV.....                                                                                 | 37 |

# 1 UVOD

Računalniški vid je znanstvena disciplina, ki se ukvarja z razvojem sistemov za pridobivanje informacij iz (digitalnih) slik [1]. Pojem slika je v tem primeru mišljen široko: lahko je to samostojna digitalna fotografija, lahko je video posnetek, lahko je serija usklajenih posnetkov istega motiva (tako imenovan stereo posnetek), lahko je posnetek v umetnih barvah iz toplotne IR kamere ali posnetek iz kakega drugega dvo- ali več-dimenzionalnega senzorja. Posnetki se lahko zajemajo v realnem času. Iz navedenega sledi, da v tehnološkem smislu razvoj sistemov računalniškega vida pomeni razvoj senzorjev za zajem podatkov (optičnih in drugih »kamer« oz. podsistemov), razvoj učinkovitih algoritmov za različne načine obdelave posnetkov, ter čimbolj optimalno implementacijo teh algoritmov v programski in strojni opremi.

Za računalniški vid je značilno, da ne obstaja standardiziran način reševanja problemov. Namesto tega imamo množico algoritmov, ki rešujejo posamezne dobro definirane naloge, težko pa jih je posplošiti. Reševanje problemov je tako večinoma odvisno od inženirske intuicije, trenutne inspiracije in pristopa »poskusi in popravi« dokler ne pridemo do želenega rezultata. V splošnem je večina sistemov računalniškega vida namenjenih reševanju specifičnih problemov nad dobro definiranimi množicami podatkov, v zadnjem času pa postajajo vse pogostejši sistemi, ki z uporabo metod strojnega učenja dosegajo določeno mero sprotnega prilagajanja in dajejo smiselne rezultate tudi, če množica vhodnih podatkov ni vnaprej definirana.

Tradicionalna področja uporabe sistemov računalniškega vida so:

- kontrola procesov, primer so industrijski roboti in avtonomna vozila,
- detekcija dogodkov, primer je video nadzor, štetje ljudi,
- organiziranje slikovnih informacij ali drugih dvo- ali več-dimenzionalnih podatkov, npr. indeksiranje slikovnih baz,
- modeliranje objektov ali okolij, npr. industrijsko preverjanje, analiza slik v medicini, topografsko modeliranje,
- inovativni računalniški vmesniki [1].

Sodobna področja uporabe na katerih trenutno poteka intenziven razvoj pa so:

- vseprisotno računalništvo (angl. ubiquitous computing), ki pomeni korak naprej v razvoju računalniških vmesnikov – vmesnik ni več sestavljen iz tipkovnice, miške in ekrana, ampak je lahko kakršnekoli oblike, sama oblika pa je izrazito odvisna od funkcije [3],
- obogatena resničnost (angl. augmented reality), ki je zmes tehnologij računalniškega vida in navidezne resničnosti, in pomeni združevanje posnetka resničnega sveta z umetno ustvarjenimi objekti in pokrajinami v enoten pogled z namenom “obogatitve” izkušnje opazovalca [4][5],
- umetni spoznavni vid (angl. cognitive vision), povezovanje računalniškega vida in metod umetne inteligence z namenom omogočiti računalniškemu agentu razumevanje vsebine [6],
- prostorska inteligenca (angl. ambient intelligence), katere cilj je izdelava bivalnega okolja, ki se zna samodejno prilagajati uporabnikom in poskrbeti za njihove potrebe [7].

V pričujočem delu bomo metode računalniškega vida uporabili za analizo in organizacijo slikovnih podatkov na področju obravnave obnašanja ptic v jatah. Do zdaj se je klasifikacija in organizacija teh podatkov opravljala ročno, mi pa bomo z uporabo računalniškega vida izdelali orodje, ki bo omogočalo avtomatizirano in masovno obdelavo teh podatkov. S tem orodjem želimo omogočiti pridobitev večje zaloge podatkov, ki je potrebna za preverjanje hipotez o obnašanju ptic v jatah. Posledično želimo prispevati delež k razvoju področja umetne inteligence, ki obravnava obnašanje umetnih agentov v jatah in je del širšega področja upravljanja skupin objektov [10].

Najprej v drugem poglavju predstavimo uporabljen knjižnico metod računalniškega vida OpenCV in utemeljimo odločitev za programski jezik C++ v povezavi z odprtokodnim ogrodjem OpenFrameworks, ki omogoča hiter razvoj eksperimentalnih OpenGL grafičnih aplikacij. V tretjem poglavju orišemo področje umetnih agentov v jatah in izpostavimo hipotezo, katere rešitev zasledujemo. Pregledamo nabor vhodnih podatkov – slik ptic v jatah, opišemo trenutni ročni način analiziranja podatkov in v tem okviru podrobno zastavimo problem ter definiramo željene rezultate. Četrto poglavje opisuje pot do rešitve skozi več

stopenj operacij nad podatki, na koncu pa predlagamo tudi nekaj možnih izboljšav na nivoju operacij in na nivoju celotnega sistema. V petem poglavju nato ovrednotimo dobljene rezultate glede na primerjavo z ročno izvedeno analizo slik. Šesto in hkrati zadnje poglavje je ocena izvedbe sistema računalniškega vida in podanimi smernicami nadaljnega razvoja.

## 2 IZBRANA ORODJA

### 2.1 Izbira programskega jezika

Sistemi računalniškega vida so po naravi računsko intenzivni, zato so edina primerna izbira za razvoj prevajani jeziki. Želimo, da je rezultat dela prenosljiv in uporaben najbolj razširjenih sistemih za osebne računalnike, torej GNU/Linux, Mac OS X in MS Windows, zato izbiramo le med jeziki, ki je so podprti na vseh omenjenih sistemih. Pri izbiri smo pozorni tudi na jezik, v katerem je napisana knjižnica metod računalniškega vida OpenCV, ki jo bomo uporabili pri snovanju rešitve. Izbrani jezik je tako C++, delovno okolje pa je Mac OS X in nabor razvojnih orodij Xcode.

### 2.2 Knjižnica OpenCV

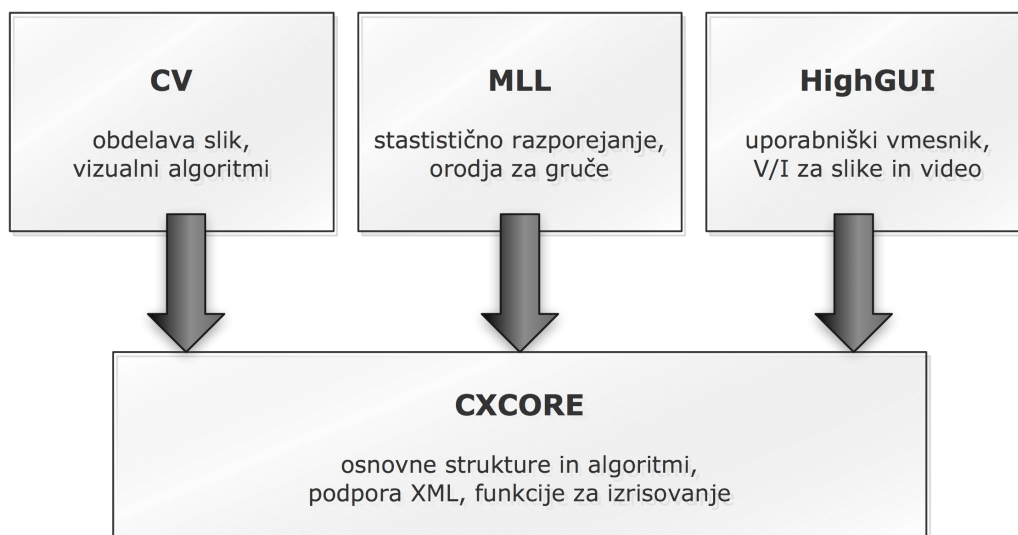
Kot glavno orodje za rešitev zastavljenega problema računalniškega vida bomo uporabili odprtokodno knjižnico OpenCV<sup>1</sup>. Knjižnica vsebuje več kot 500 funkcij iz različnih področij računalniškega vida, ki so uporabne za tako raznovrstne aplikacije kot so nadzor nadzor kvalitete izdelkov v tovarni, analiza slik v medicini, video nadzor in varovanje, izdelava uporabniških vmesnikov, kalibracija kamer, stereo vid, robotika. Vsebuje tudi podknjižnico za strojno učenje, saj se problemi računalniškega vida velikokrat rešujejo tudi z uporabo strojnega učenja. Izvedbe algoritmov so računsko optimizirane in prilagojene za izvajanje v realnem času. Dodatno so podprti sodobni večjedrni procesorji in Intel-ova komercialna knjižnica Integrated Performance Primitives, v kateri so zbrane nizkonivojske optimizirane operacije za uporabo v algoritmihi.

Osnovno strukturo OpenCV prikazuje slika 1. Metode v knjižnici so razdeljene v več modulov. Glavna modula sta CV, ki vsebuje nabor metod računalniškega vida, in MLL, v katerem so metode za strojno učenje. Knjižnico sestavljata še modula HighGUI, ki vsebuje

---

<sup>1</sup>Prosto dostopna na naslovu <http://sourceforge.net/projects/opencvlibrary/>.

gradnike za izdelavo enostavnega uporabniškega vmesnika, namenjenega testiranju različnih funkcionalnosti knjižnice, in CXCORE, ki vsebuje podatkovne strukture, določene splošne algoritemske operacije in podporo za XML ter izrisovanje grafike.



Slika 1: Osnovna struktura knjižnice OpenCV<sup>2</sup>.

Pri našem delu se bomo osredotočili izključno na modul CV, za ostale funkcionalnosti pa bomo uporabili druge knjižnice.

Glede na [2] in [8], lahko funkcije modula CV v grobem razdelimo v naslednje kategorije:

- obdelava slik: glajenje (angl. blur) in druge jedrne operacije (opomba: jedrne operacije so tiste, pri katerih se nova vrednost trenutnega piksla izračuna z upoštevanjem vrednosti sosednjih pikslov), poplavljanje (angl. flood fill), upragovanje (angl. threshold), detekcija robov, iskanje premic in krogov, afine preslikave, pretvorbe barvnih prostorov, pretvorbe koordinatnih sistemov;
- srednji nivo: iskanje obrisov krivulj (angl. contour), operacije nad krivuljami, izločanje ozadja, različne metode segmentacije;
- višji nivo: iskanje primernih točk za sledenje, različne metode sledenja in iskanja objektov;
- obdelava stereo slik: triangulacija, stereo kalibracija in ujemanje točk na slikah.

<sup>2</sup> Diagram prirejen po [8], stran 13.

Pri reševanju zastavljenega problema bomo črpali večinoma iz prve in druge kategorije, tretja in četrta kategorija pa predstavljata temelj za možne nadgradnje. Nekaj takih možnosti bo predstavljenih v šestem poglavju.

## 2.3 Programsko ogrodje OpenFrameworks

Z namenom hitrega razvoja in sprotnega eksperimentiranja smo se odločili, da programa ne bomo razvijali od začetka, ampak z uporabo ogrodja, ki bo omogočalo določen del razvoja v osnovnem programskem jeziku, hkrati pa bo dovolj fleksibilno, da bo v njem možno hitro preizkušanje različnih kombinacij algoritmov. Potrebujemo tudi dobro podporo standardni grafični knjižnici OpenGL, ter operacijam, ki niso neposredno povezane z reševanjem zastavljenega problema, kot je shranjevanje nastavitev, nalaganje podatkov iz diska, upravljanje uporabniškega vmesnika.

Kot že omenjeno, smo izbrali programski jezik C++ zaradi njegove zmogljivosti in zaradi ohranjanja kompatibilnosti med različnimi platformami, kar zahtevamo tudi od izbranega programskega ogrodja.

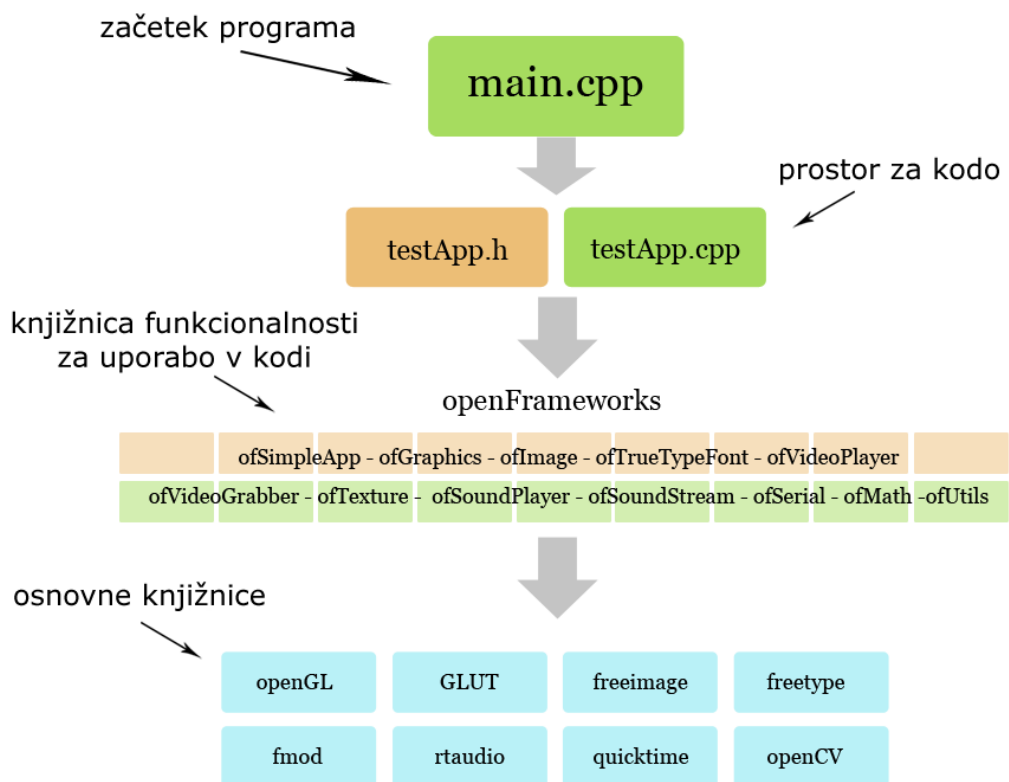
Po preverjanju licenčnih pogojev, potrebnega učenja in količine prilagajanja za posamezno ogrodje, smo se odločili za uporabo odprtokodnega ogrodja OpenFrameworks<sup>3</sup>.

Ogrodje nudi poenoten programski vmesnik (angl. API, application programming interface) do različnih funkcionalnosti, za katere so uporabljene obstoječe knjižnice, npr. grafični del je zasnovan na OpenGL knjižnici GLUT, vsebuje pa tudi delno podporo knjižnici OpenCV. Podprt je sicer le majhen izbor najpogostejših funkcij modula CV, ostale funkcije pa je možno uporabiti v izvorni obliki (angl. native) ali pa jih dodati v nabor kot nove izvedene funkcije s programsko vmesno kodo (angl. glue) obstoječih izvedenih funkcij [9].

Struktura ogrodja je prikazana na sliki 2. Razdeljena je v štiri nivoje, od katerih najnižji nivo sestavljajo osnovne knjižnice, nivo višje pa najdemo bistvene komponente ogrodja. Te poskrbijo za določen nivo abstrakcije funkcionalnosti osnovnih knjižnic, ob tem pa nudijo poenoten programski vmesnik za razvoj aplikacije. Še nivo višje je prostor za kodo, na vrhu pa je vstopna točka, ki skrbi za zagon in zaustavitev aplikacije.

---

<sup>3</sup> Prosto dostopno na naslovu <http://www.openframeworks.cc>.



Slika 2: Struktura programskega ogrodja OpenFrameworks (diagram prirejen po prosto dostopnem na spletnem naslovu [http://wiki.openframeworks.cc/index.php?title=OF\\_code\\_structure\\_%28image%29](http://wiki.openframeworks.cc/index.php?title=OF_code_structure_%28image%29)).

### 3 OBNAŠANJE PTIC V JATAH

Oris področja obnašanja ptic v jatah povzemamo po [10] avtorjev I. Lebar Bajec in F. Heppner. Znanstveniki različnih disciplin (ornitologi, biologi, fiziki, informatiki) že kakih 40 let poskušajo pojasniti vzroke in pravila, ki omogočajo pticam njihovo fascinantno gibanje v jatah. Opazovalcu tega gibanja se zdi, da se vse ptice gibajo sinhrono, da menjajo smer vse v istem trenutku.

Polje raziskav se je že zelo zgodaj razdelilo v dva pola. En pol obsega raziskave ptic, ki letijo v formacijah V, J ali diagonalni (angl. echelon), drugi pol pa raziskuje ptice, ki letijo v gručah. V prvi skupini so večinoma večje ptice kot so npr. gosi in race, v drugi pa manjše, npr. škorci.

V začetku je prevladovalo mnenje, da se ptice združujejo v jate oblike V, J ali diagonalne z namenom prihranka energije pri letenju na velike razdalje. Letenje v taki formaciji naj bi pticam omogočalo boljše aerodinamične izkoristke in zmanjšanje zračnega upora. Tako naj bi ptice po določenih kriterijih prihranile tudi do 70% enegije. Kasnejša opazovanja in simulacije so pripeljala do ugotovitve, da je resničen prihranek precej nižji in sicer okrog 10%. Izkazalo se je namreč, da ptice v jatah premalo časa letijo v idealni formaciji, da bi iz tega lahko sklepali, da je prihranek energije na račun aerodinamike edino oz. glavno vodilo njihovega nastanka. Druga verjetna hipoteza razlaga za letenje v formaciji pripisuje večji pomen medsebojnemu opazovanju ptic. Tako imamo vodilno ptico, ki je prva gledano v smeri letenja, ostale ptice pa se poravnajo tako, da imajo odprt pogled nanjo in hkrati dober pregled naprej v smeri letenja. To jim omogoča hitro reagiranje glede na spremembe leta vodilne ptice, brez "čakanja", da se informacija prenese preko sosedov. Razširjeno mnenje je, da ptice upoštevajo tako kriterij izkoriščanja aerodinamike, kot tudi kriterij opazovanja vodilnega, vendar razmerje med kriterijema še ni dobro raziskano, med drugim tudi zaradi pomanjkanja raziskav letenja ptic na dolge proge, npr. ob selitvah.

Pri manjših pticah, ki oblikujejo jate v obliki gruč, se kot glavna neznanka izpostavlja problem hipnega prenosa informacije o obratu skozi jato, da izgleda kot da se vse ptice obrnejo istočasno. Ker so bili izmerjeni reakcijski časi ptic pri obratu v jati za razred nižji od izmerjenih reakcijskih časov posameznih ptic v vetrovniku (14ms proti 100ms), je za nekaj časa obveljalo mnenje, da dobijo vse ptice hkrati ukaz (navodilo) od trenutno vodilne ptice v

jati. Tako so dolgo potekale raziskave potencialnih mehanizmov za hitro in učinkovito prenašanje takih ukazov med pticami, vključno z možnostjo prenosa misli. Sčasoma pa je, tudi ob pomoči računalniških simulacij katerim se bomo čez nekaj trenutkov posvetili, prevladala hipoteza, da ptice v jati uporabljajo preprostejše mehanizme obnašanja, ki temeljijo na opazovanju sosednjih ptic in neprestanemu prilagajanju smeri in hitrosti leta glede na določena pravila obnašanja.

Oba pola raziskav obnašanja ptic v jatah torej v neki točki pripeljeta do podobnega zaključka: ptice se gibljejo na podlagi določenih pravil, katerih temelj je opazovanje hitrosti, smeri in oddaljenosti sosednjih ptic v jati. Ta pravila jim omogočajo, da ostanejo del jate. Natančna biološka vsebina pravil ni dobro raziskana, niti ni raziskano ali so pravila znotraj jate homogena ali heterogena. Poleg tega je zelo verjetno, da se pri posameznih vrstah ptic določeni parametri razlikujejo oz. da so vsa ali del pravil obnašanja popolnoma drugačen.

S pojavom dovolj zmogljivih procesorjev in grafičnih sistemov se je za raziskovanje obnašanja ptic v jatah začelo intenzivno uporabljati simulacije, ki omogočajo prilagajanje parametrov simulacije in opazovanje dogajanja z grafičnim izrisovanjem v realnem času. V zadnjih letih je tako na tem področju nastala vrsta računalniških modelov, v katerih so ptice predstavljene kot *animati* (angl. animats, boids), ki se ravnaajo po pravilih, ki poskušajo posnemati obnašanje ptic v naravi. Kot že omenjeno, je bistvo takih pravil opazovanje sosedov do določene oddaljenosti in na podlagi tega sprejemanje odločitev. Kot v naravi, tudi tu odločitve poskušajo zasledovati kriterije kot so ohranjanje enakomerne razdalje do sosedov in ohranjanje čimbolj usklajene hitrosti gibanja s sosedi.

Ko so taki modeli prvič pojavili konec osemdesetih let, so takoj postali zanimivi za uporabo v filmski industriji, saj so pomenili ogromen skok naprej v primerjavi s tradicionalnimi poskusi realističnega animiranja obnašanja množic. Filmska industrija je sčasoma za lastne potrebe razvila zelo kompleksne modele, a zaradi njihove komercialne narave ostajajo ti modeli zaprti za znanstveno javnost.

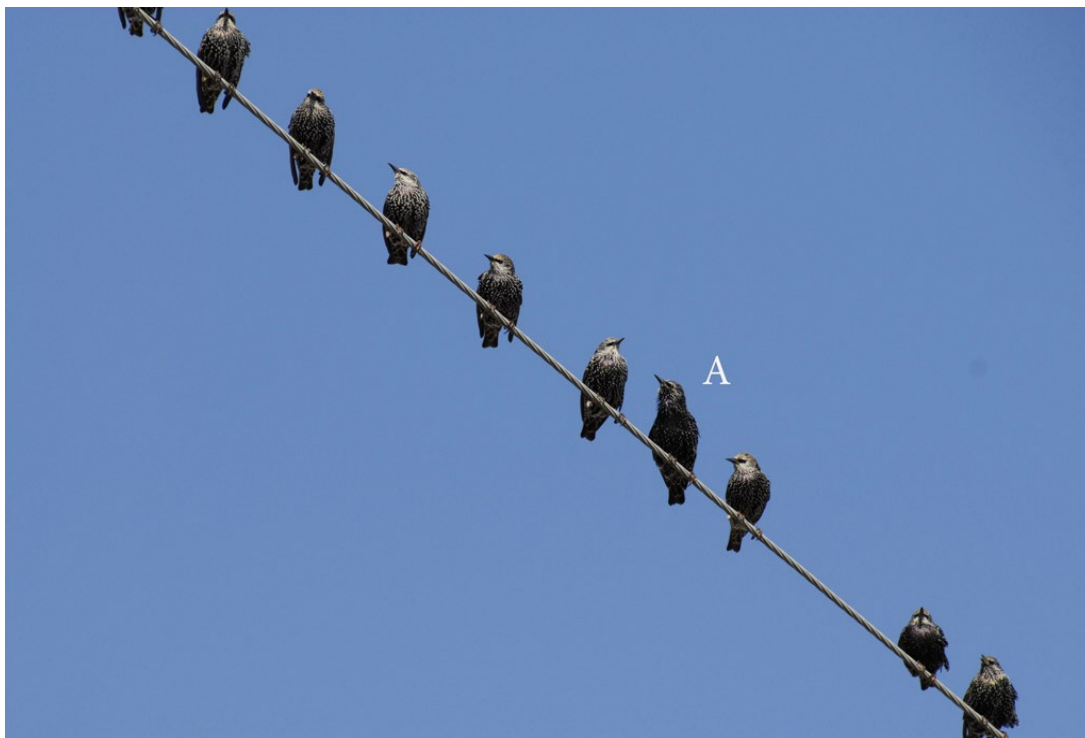
Začetni modeli so vsebovali preproste odločitvene mehanizme, včasih temelječe celo na uporabi enega samega pravila obnašanja. S kasnejšimi modeli so se razvile čedalje kompleksnejše metode obravnavanja pravil, npr. optimiziranje parametrov odločanja z genetskimi algoritmi, ali različne metode umetne inteligence. Poleg samih pravil se je povečala tudi kompleksnost obravnave zaznavanja okolice, npr. z uporabo različnih mejnih oddaljenosti opazovane sosednosti za različna pravila ali skupine pravil. Trenutna generacija

modelov poleg naštetih lastnosti dodaja uporabo mehke logike na odločitvenem nivoju.

Kljub temu, da obstoječi modeli v določenih primerih zelo dobro posnemajo obnašanje ptic v naravi, ostaja še veliko prostora za izboljšave. Ena glavnih pomanjkljivosti trenutnih modelov je, da je obnašanje vseh animatov v dani simulaciji homogeno, torej podrejeno istim pravilom obnašanja. V naravi je situacija seveda precej drugačna, saj se ptice v jati razlikujejo po starosti, spolu, zdravju in drugih faktorjih, ki vplivajo na njihovo kolektivno obnašanje v jati. V tej točki je pridobivanje terenskih podatkov ključnega pomena, saj je šele na podlagi teh podatkov možno predvidevati dejansko sestavo jat in pridobiti znanje za modeliranje nehomogenih jat ptic.

Pri pridobivanju terenskih podatkov obstajata dva ključna problema. Prvi problem je zajem podatkov (mišljen je tehnološki zajem, sam proces iskanja primerje jate je prepuščen držnim ornitologom), ki mora biti hiter, dovolj natančen in poceni. Drug problem pa je analiza podatkov, ki velikokrat poteka ročno (npr. štetje ptic na fotografijah ali ročno sledenje konkretni ptici na posnetku).

Primer slike ptic vidimo na sliki 1. S črko A je analitik označil ptico, za katero meni, da je odrasla, ostale pa so po njegovi oceni mladiči. Tabela 1 je primer analize slike 1. Najprej so izmerjene velikosti posameznih ptic v slikovnih točkah – pikslih (oznaka px), iz katerih je izračunana povprečna velikost (oznaka PV) za konkretni posnetek. Potem so izmerjene medsebojne razdalje med pticami najprej v pikslih, nato še izražene v povprečnih velikostih (PV), kar je ključno, da lahko med seboj primerjamo rezultate različnih posnetkov. Za vsako ptico je še podana verjetnost, da je mladič ( $P_{\text{mladič}}$ ).



Slika 3: Primer posnetka ptic, s črko A je analitik označil odraslo ptico.

Tabela 1: Podrobna analiza posnetka s slike 1.

|               |      |      |      |      |      |      |      |      |      |       |
|---------------|------|------|------|------|------|------|------|------|------|-------|
| birds6.jpg    |      |      |      |      |      |      |      |      |      |       |
| ptica št.     | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | PV    |
| dolžina v px  | 95   | 96   | 89   | 89   | 97   | 112  | 94   |      |      | 96,00 |
| razdalja v px | 110  | 117  | 118  | 131  | 88   | 90   | 219  | 78   |      |       |
| razdalja v PV | 1,15 | 1,22 | 1,23 | 1,36 | 0,92 | 0,94 | 2,28 | 0,81 |      |       |
| Pmladič       | 0,60 | 0,60 | 0,75 | 0,60 | 0,70 | 0,00 | 0,20 | 0,20 | 0,20 |       |

Pričujoče delo je poskus reševanja problema analize terenskih podatkov z uporabo tehnologij računalniškega vida. Predpostavke, ki so bile vodilo pri nastanku dela, so:

- ptice se gibljejo v zraku, kar pomeni precej uniformno ozadje neba, ki izrazito olajša uporabo ločevanja objektov od ozadja (segmentacije), ki je ena od bistvenih operacij računalniškega vida, na kateri lahko temeljijo ostale analitične operacije;
- ptice se gibljejo podnevi, kar pomeni dobro osvetljenost in s tem dobro izraženost podrobnosti na pticah, kar je drug steber za doseg kvaliteten računalniškemu vidu naklonjene baze posnetkov;
- sodobni digitalni fotoaparati omogočajo pridobivanje visoko ločljivih in barvno bogatih posnetkov ptic na enostaven način ob majhnem trudu fotografa.

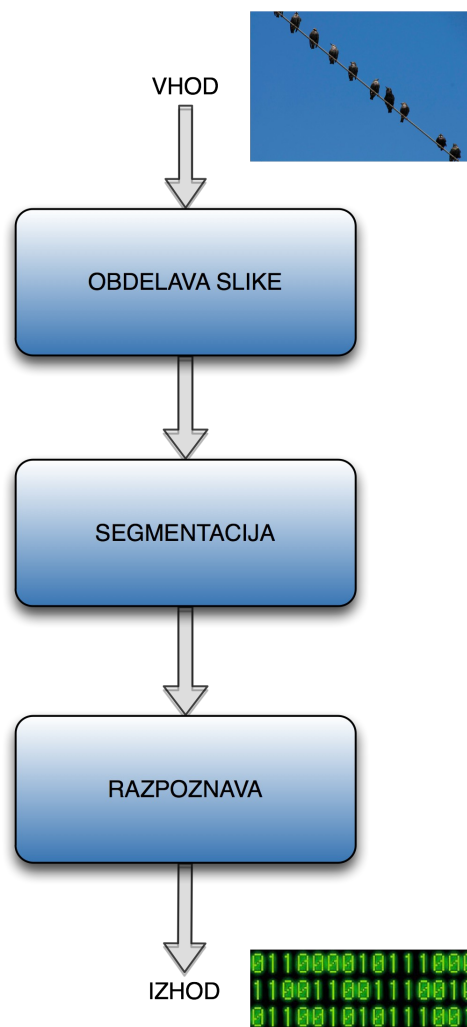
Z namenom olajšanja začetnega razvoja sistema smo dodali omejitve pri slik, primernih za obdelavo, katere načrtujemo v prihodnosti opustiti. Te omejitve so:

- ptice naj bodo v mirovanju, kar pomeni da lahko ocenimo njihove medsebojne oddaljenosti;
- ptice se med sabo ne smejo prekrivati, da je možna dobra segmentacija na podlagi razlikovanja med nebom in pticami;
- na posnetkih naj ne bo objektov, ki bi bili po velikosti in obliki podobni pticam.

Omenjene omejitve vpeljujemo z namenom, da čim hitreje pridemo do uporabnega segmentacijskega algoritma, s katerim bomo v kratkem času zmožni zbrati veliko vzorcev informacij. Z vzorčenjem teh informacij bo kasneje možno izdelati robusten algoritem, ki bo znal dobro razločevati ptice tudi na manj idealnih slikah.

## 4 PRISTOP K REŠITVI

Kot je bilo omenjeno že v uvodu, je bistvena lastnost današnjih sistemov računalniškega vida, da so izrazito prilagojeni problemu, ki ga rešujejo. Tudi v našem primeru bo tako, saj bomo poskušali doseči tako kombinacijo operacij, ki bo iz danih slikovnih podatkov čimbolj učinkovito izločila za nas pomemben nabor informacij. Kljub temu, da bo rešitev zastavljena kar se da široko v okviru danih omejitev, se zavedamo, da bo potrebno dodatno prilagajanje sistema v primeru izrazite spremembe vsebine vhodnih podatkov.



Slika 4: Splošni koraki algoritma računalniškega vida.

Slika 4 kaže splošne korake algoritma računalniškega vida (povzeto po [11]). Vhodni podatek je digitalna slika, ta se obdelava čez tri glavne stopnje operacij, na koncu pa dobimo informacijo, ki jo je algoritem uspel izluščiti. Prva stopnja operacij je obdelava slike, kjer se izvajajo osnovne operacije obdelave enostavnih slik (glej razdelek 2.2). Namen teh operacij je, da izpostavijo značilnosti slike, pomembne za učinkovito delovanje operacij na naslednji stopnji obdelave. Druga stopnja je segmentacija, kjer se slika loči na dele, ki nas zanimajo, ti tvorijo tako imenovano ospredje, in dele, katere lahko v kasnejši obdelavi zavržemo, to je ozadje. Zadnja stopnja obdelave je razpoznavanje, v kateri poskušamo izluščiti čim več uporabnih informacij iz delov, ki smo jih skozi prejšnji dve stopnji označili kot zanimive. To lahko vključuje iskanje oblik, prepoznavanje vzorcev, primerjanje z predlogami (angl. template matching) in še vrsto drugih zahtevnih operacij. Stopnja kompleksnosti operacij v splošnem narašča v smeri poteka algoritma [11].

V naslednjih razdelkih bomo posamezne korake splošnega diagrama s slike 4 postopno popolnili z izbranimi metodami, ki nas bodo pripeljale do rešitve zadanega problema računalniškega vida. Sprehodili se bomo torej od vhoda do izhoda in sproti gradili algoritem.

## 4.1 Vhodni podatki

Gradnja vsakega sistema računalniškega vida se začne s pregledom vhodnih podatkov. Vhodni slikovni podatki morajo v grobem zadostiti dvema kriterijema. Najprej morajo biti take vrste, da so primerni obdelavi z metodami računalniškega vida na način, da lahko izločimo informacije, ki nas v konkretnem primeru zanimajo. V splošnem to pomeni, da v kolikor je le možno želimo podatke s čim manj šuma in čim manj motečimi elementi, s čim večjim dinamičnim razponom, in v čim višji ločljivosti ter barvni globini. Drug kriterij je konsistentnost podatkov med seboj, ki je pomemben zaradi avtomatizacije obdelave. Želimo podatke, ki so dovolj podobni med seboj, da se uporabljeni algoritmi lahko (do določene mere samodejno) prilagajajo podatkom v okvirih svojih zmožnosti, in ni potrebe po večjem popravljanju parametrov »od zunaj« za posamezen primerek ali za različne podmnožice podatkov.

Ob tem določimo še dodatne kriterije, ki morajo veljati za vse podatke, ki vstopajo v

konkreten sistem, torej tudi za množico podatkov, katerih trenutno še nimamo na razpolago. V našem primeru so vhodni podatki barvne RGB fotografije ptic, sedečih na žicah daljnovoda, v ozadju pa je nebo. Ta predpostavka je zelo pomembna, saj uniformno ozadje, kot je nebo, omogoča učinkovit proces segmentacije. Ker so na nebu možni oblaki, bomo poskušali parametre algoritma dodatno prilagoditi tako, da bo oblake do določene mere kontrastna obravnaval kot da so nebo. Glede žic na slikah velja predpostavka, da so dovolj različen element od ptic, da jih bomo znali skozi obdelavo učinkovito izločiti. Pomembno je še, da so vse ptice v približno isti poziciji (stoječe, s spravljenimi krili) in približno isto oddaljene od kamere. To nam omogoča boljše izhodišče pri zajemanju informacij o njihovih dimenzijah in razmerjih. Za optimalne rezultate bi bilo idealno, da se ptice med seboj ne bi prekrivale oz. niti dotikale.

Za obdelavo imamo pripravljeno množico, ki jo sestavljajo trije posnetki, primer je na sliki 5. Te bodo hkrati učna in testna množica.



**Slika 5: Primer slike ptic.**

Na vsaki sliki je med 7 in 10 ptic, za katere želimo izločiti parametre:

- število ptic na fotografiji,
- velikost posamezne ptice v pikslih in skupna povprečna velikost ptic na dani sliki,
- medsebojne oddaljenosti ptic in povprečna medsebojna oddaljenost, merjena v

povprečni velikosti ptice,

- razvrstitev ptic v dva razreda glede na obarvanost perja, odrasle in mladiče.

Cilj je sestaviti tak algoritem, da bodo parametri nastavitev dovolj splošni, da bodo brez sprotnega prilagajanja pravilno obdelali vse tri slike vhodne množice. Za prilagajanje parametrov ob pojavu novih slik bomo izdelali grafični uporabniški vmesnik, ki bo omogočal spreminjanje parametrov, spremljanje rezultatov obdelave v realnem času in shranjevanje parametrov za kasnejšo uporabo.

## 4.2 Obdelava slike

Bistvo obdelave slike je, da izpostavi določene značilnosti vhodne slike na način, da postanejo osnova za učinkovito delovanje operacij segmentacije. Z drugimi besedami to pomeni, da pripravimo slikovne podatke tako, da postane jasna razmejitev med *ospredjem*, ki predstavlja pomemben del podatkov, in *ozadjem*, ki označuje za našo obdelavo nepomemben del podatkov. Ta razmejitev se lahko izrazi na zelo različne načine.

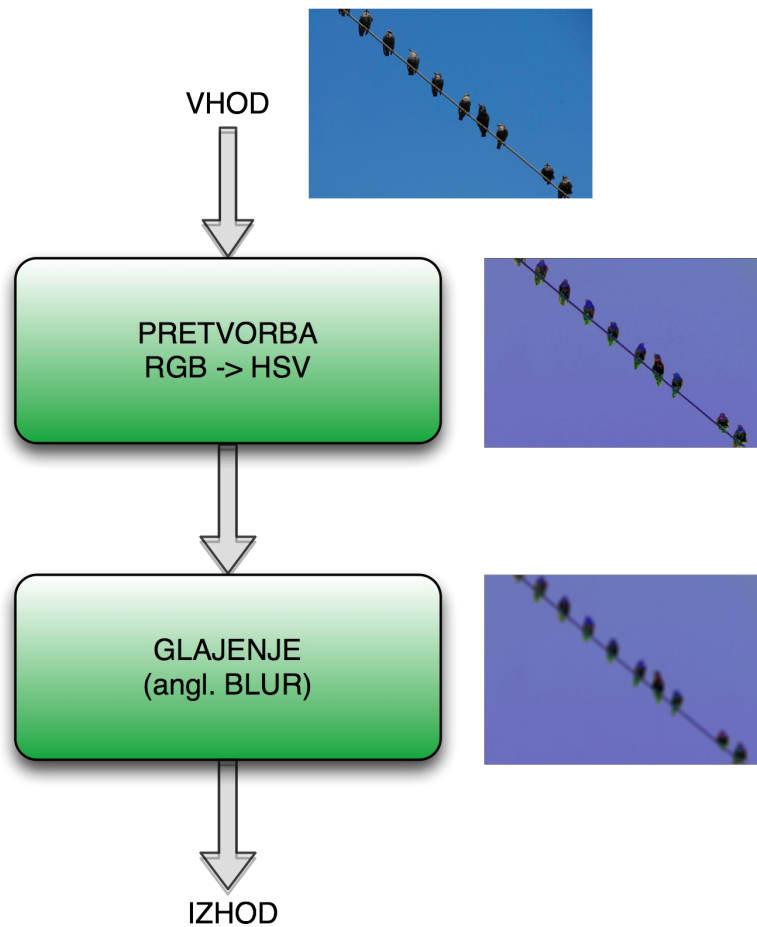
Na tej stopnji obdelave se operacije izvajajo istočasno in v enaki meri na vseh pikslih vhodne slike. Operacije so lahko matrične, ki za izračun vsakega piksla slike upoštevajo prispevek vrednosti okoliških pikslov, te so npr. glajenje (angl. blur), Gaussovo glajenje, erozija (angl. erosion), raztegovanje (angl. dilate). Druga vrsta operacij se izvaja nad barvnimi komponentami pikslov, kot npr. sprememba svetlosti (angl. brightness), kontrasta (angl. contrast), ali sprememba barvnega prostora (npr. iz RGB v HSV, podrobna razlaga sledi). Tretja vrsta obdelav so odločitvene (angl. threshold) operacije, ki vsako točko glede na barvno vrednost postavijo v eno od dveh možnih stanj, ozadje ali ospredje oz. skrajno črno ali skrajno belo. Ta vrsta obdelav v bistvu že spada v domeno segmentacije in jo bomo posebej obravnavali v naslednjem razdelku.

Pri določanju kriterijev za dovoljene vrste vhodnih slik za naš algoritem, je bila ena od osnovnih predpostavk ta, da so ptice fotografirane na uniformnem ozadju neba. Glede na to predpostavko sprejmemo odločitev, da bomo pri segmentaciji za ločevanje med ozadjem in ospredjem uporabili točno določen barvni odtенок. To pomeni, da bomo vse piksele, ki bodo

ustrezali tej valovni dolžini (z možnostjo določenega odstopanja), označili kot nepomembne, torej ozadje, vse ostale pa kot pomembne - ospredje. Da bo ta operacija v segmentaciji možna, moramo pretvoriti barvni prostor vhodne slike iz RGB, ki je izvorna oblika, v HSV, ki je primernejša oblika za segmentacijo na osnovi barve. Vzrok za to je v tem, na kakšen način so predstavljeni različni barvni odtenki v različnih barvnih prostorih. V prostoru RGB je vsak odtenek predstavljen kot kombinacija treh osnovnih barv, rdeče (R, angl. red), zelene (G, angl. green) in modre (B, angl. blue). To pomeni, da moramo pri ločevanju na osnovi barv za vsako barvo primerjati tri vrednosti. V prostoru HSV je vsak barvni odtenek prav tako predstavljen s tremi vrednostmi, ki so odtenek (H, angl. hue), nasičenost (S, angl. saturation) in vrednost (V, angl. Value). Ključno je, da je informacija o barvnem odtenku je zapisana samo v vrednosti H. To pomeni, da je izvedba ločevanja na osnovi odtenka v tem barvnem prostoru precej bolj učinkovita. Potrebujemo še način, kako bomo določili pravi odtenek modre barve. Dobili jo bomo tako, da bomo v uporabniški vmesnik vgradili funkcijo izbire barve kar na vhodni sliki. Poleg tega bomo v uporabniškem vmesniku omogočili še nastavitve dovoljenega odstopanja, da bodo zajete določene variacije med vhodnimi slikami, morebitni oblaki in malenkostne spremembe v osvetlitvi na različnih delih slike (posledica leče kamere, nagnjenosti kamere glede na horizont in podobno).

Poleg neba so odvečni element na vhodnih slikah še žice, na katerih sedijo ptice. Ker so žice zelo podobne barve kot ptice, zanje ne moremo uporabiti podobnega prijema kot za nebo. Izkaže pa se, da so žice v eni dimenziji bistveno manjše (ožje) od ptic, to pa pomeni, da jih lahko zabrišemo z operacijo glajenja (angl. blur), ob tem pa ohranimo ptice skoraj nespremenjene. Na ta način naredimo žice dovolj podobne ozadju, da jih bo algoritem za segmentacijo tudi označil kot ozadje. Operacija glajenja ne sme biti premočna, ampak je pomembno, da je glajenje ravno še dovolj intenzivno, da izloči žice, ptice pa pusti čimbolj nedotaknjene. V ta namen podobno kot pri določanju barve neba tudi tu vgradimo možnost izbire intenzivnosti operacije glajenja v uporabniški vmesnik.

Obdelavo slike za naš algoritem smo tako sestavili iz dveh operacij. Prva je pretvorba barvnega prostora iz RGB v HSV, ki omogoča segmentacijo na osnovi izbrane barvnega odtenka, druga pa je operacija glajenja, ki zabriše neželene elemente - žice tako, da postanejo podobni ozadju. Na sliki 6 so operacije obdelave slike.



Slika 6: Operacije obdelave slike.

### 4.3 Izločanje objektov – segmentacija

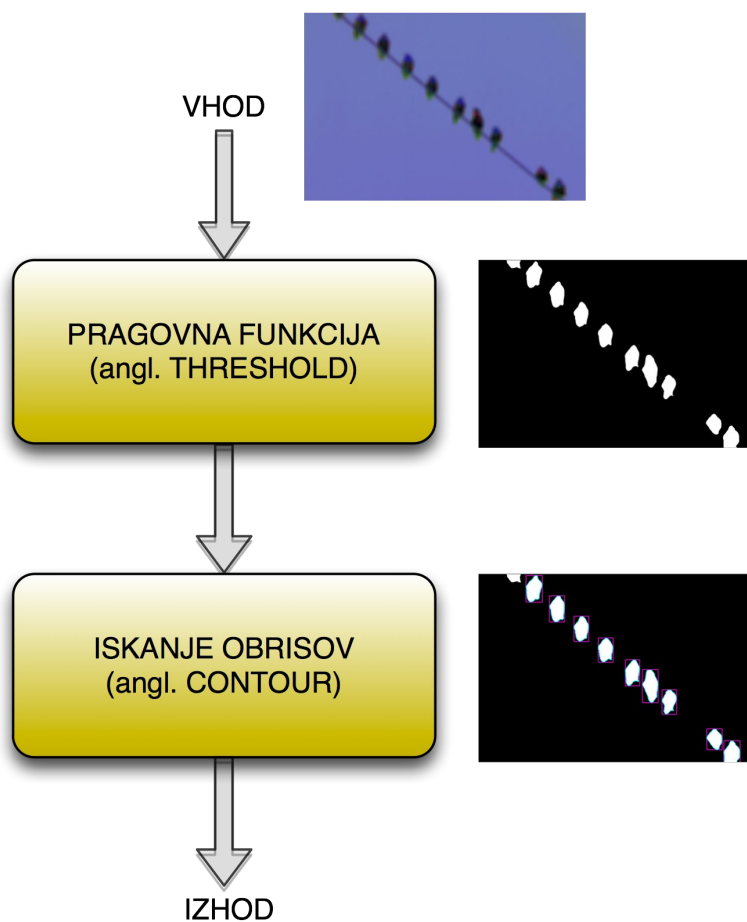
Raziskovanje algoritmov za segmentacijo je eno najpomembnejših področij računalniškega vida. Rezultati segmentacije so ponavadi prvi korak na poti izločevanja uporabnih informacij iz množice vhodnih podatkov, zato je zelo zaželeno, da je segmentacija učinkovita, predvidljiva in ponovljiva operacija [8][11]. Za enostavne slike je lahko operacija zelo učinkovita, pri kompleksnih slikah pa se lahko zadeve izredno zapletejo in postanejo za današnjo stopnjo tehnologije celo nerešljive. Tudi (ali predvsem) zato obstajajo precejšnje število algoritmov za izvajanje segmentacije, od preprostih, kot sta iskanje obrisov na binarnih (črno-belih) slikah ali poplavljanje označenih regij (angl. watershedding), do

visokonivojskih, kot je iskanje objektov s Haar značilkami. Ta sicer meji že na področje razpoznavanja. Rezultat segmentacije so običajno informacije o številu najdenih objektov ali področij zanimanja, velikosti in pozicije teh objektov in področij, večkrat pa so prisotne tudi druge informacije, npr. o obliki najdenih objektov. Primer preproste uporabe je v video nadzoru, ko kamera večinoma snema nepremično ozadje, zanimivi so le posnetki, ko se na posnetku nekaj bistvenega spremeni. To spremembo segmentacija zelo jasno opredeli in tako pomaga pri iskanju zanimivih delov posnetkov. Z naprednejšimi operacijami lahko segmentacija omogoča veliko več kot le ločevanje ozadja in ospredja. Možna je večstopenjska segmentacija kot je izolacija posameznih objektov ali struktur, npr. ločitev obraza ali dlani iz silhuete telesa. Dodatno je tako ločene segmente z uporabo naprednih podatkovnih struktur možno spremljati skozi sekvenco posnetkov in tako skozi čas slediti dogajanju, ki nas zanima [8].

Vhodna slika je predhodno obdelana z določenimi operacijami, ki poudarijo značilnosti podatkov, ki so pomembne za učinkovito izvedbo segmentacije. Tako imamo pripravljeno sliko v barvnem prostoru HSV in obdelano z glajenjem. Kot že omenjeno v prejšnjem razdelku, bo operacija segmentacije temeljila na ločevanju na podlagi barvnega odtenka. To bomo izvedli z uporabo pragovne funkcije (angl. threshold), pri kateri bodo vsi piksli, ki ustrezajo določeni valovni dolžini, označeni kot ozadje, ostali pa kot ospredje. Poleg barve je parameter pragovne funkcije še širina dovoljenega odstopanja od dane barve, za katerega omogočimo nastavitve v uporabniškem vmesniku na istem mestu, kot je omogočena izbira barve. Rezultat pragovne funkcije je črno-bela slika, kjer so z belimi lisami označena področja, ki nas zanimajo za nadaljno obdelavo.

Naslednja operacija segmentacije je iskanje obrisov (angl. contour finding), ki iz črno-bele slike izloči krivulje obrisov področij bele barve (in, če bi tako želeli, tudi krivulje obrisov področij črne barve, če se ta nahajajo znotraj področij bele barve – to so tako imenovane luknje). Najdene krivulje obravnava kot objekte in za vsakega izračuna lokacijo na sliki, širino in višino v pikslih, ter vektorski opis krivulje. Pri izvajanju operacije so parametri najmanjša in največja velikost belega področja, ki naj se še obravnava kot ločen objekt, s čimer pomagamo pri izločanju potencialnih nepravilnosti, ter že omenjena izbira ali želimo obravnavo lukenj. Rezultat iskanja obrisov je seznam najdenih objektov, za katere smo kot razvijalci algoritma prepričani, da so ptice. V tem seznamu najdemo za vsak objekt podatke o poziciji zgornjega levega vogala objektu očitane pravokotnika, širini in višini tega pravokotnika, poziciji centralne točke objekta in vektorsko zapisano krivuljo obrisa.

Segmentacijo smo, tako kot obdelavo slike, sestavili iz dveh zaporednih operacij. Prva operacija je vhodno sliko z uporabo pragovne funkcije pretvorila v črno-belo oz. binarno sliko, na kateri so z belo izraženi objekti, katere želimo obravnavati. Druga operacija je tem objektom določila lokacijo, izmerila njihovo velikost in vektorizirala njihove obrise. Potek operacij je prikazan na sliki 7.



**Slika 7: Operacije segmentacije.**

Del dobljenih podatkov je v tem trenutku že del rešitve zastavljenega problema. Ti podatki so število, velikost in pozicija objektov - ptic, iz katerih je že možno grobo izračunati pozicijo in določene parametre medsebojnih razmerij. Grobo pravimo zato, ker so vrednosti velikosti merjene na pticam očitanih pravokotnikih in tako ne upoštevajo nagnjenosti ptic glede na pogled kamere. Da bi to kompenzirali in dobili natančnejše podatke, so potrebne dodatne operacije, ki jih bomo izvedli nad dobljenim seznamom ptic v postopku razpoznavne.

## 4.4 Razpoznavna

Če je segmentacija najpomembnejše področje računalniškega vida, je razpoznavna daleč najzahtevnejše. Razpoznavna pomeni razumevanje vsebine prizora, ki je na sliki. Kljub temu, da so zelo učinkoviti pri reševanju določenih zelo zahtevnih problemov, kot je npr. rekonstrukcija 3D pogleda iz dveh 2D slik, posnetih pod različnim kotom, je razumevanje celo dokaj enostavnih prizorov za računalnike še vedno nemogoča naloga, katero bi na drugi strani zmogel že dveletni otrok [11]. Za nameček niti raziskovalci sami ne znajo napovedati, kdaj bodo računalniki tega zmožni. Omenjeno je posledica obsega problema razumevanja splošnih prizorov, ki pa se da z določenimi zožitvami zmanjšati na obvladljivo raven vsaj za določene skupine problemov. Zelo pogosta zožitev je iskanje dobro definiranega objekta (angl. object detection), za katero obstajajo hitre in učinkovite metode. Druga možnost je iskanje togih objektov, za katere poznamo medsebojna razmerja njihovih sestavnih delov (angl. instance recognition), s pomočjo iskanja značilnih točk. Slednji pristop se npr. uporablja na področju obogatene resničnosti (glej razdelek 2.3). Pomembno podpodročje razpoznavne je seveda strojno učenje, saj brez uporabe metod strojnega učenja praktično ni možno predelati (dovolj) velikih baz, ki so osnova za "treniranje" sistemov, ki opravljajo kompleksne razpoznavne različnih kategorij objektov.

V našem primeru je problem razpoznavne razmeroma preproste narave. Zaradi omejitev vhodnih podatkov smo bistven korak iskanja ptic glede na ozadje dosegli že s segmentacijo. Ostane nam le še poskus, da iz do sedaj zbranega znanja o pticah pridobimo še dodatne informacije. Glede na v segmentaciji pridobljeno znanje o pticah, je možna nadaljna obdelava v dveh smereh. Prva možnost je dodatna obdelava obrisa ptice, ki lahko pomeni osnovo za dokončno razpoznavo objekta, ali pridobitev dodatnih informacij o že poznanem objektu. Druga možnost je vrnitev na obdelavo začetne barvne slike objekta, vendar tokrat z dodatno informacijo o lokaciji, velikosti in obliki obravnavanega objekta.

Odločili smo se uporabiti obe poti in ju kasneje združiti v kompleksno znanje o obravnavanih pticah. Obdelavo obrisa bomo uporabili za določitev položaja posamezne ptice glede na kamero. Ta položaj bomo ugotovili iz nagiba ptice glede na njej očrtani pravokotnik in nam bo v pomoč pri natančnejšem določanju velikosti ptice in njene relativne pozicije. Obdelavo dela barvne slike skupaj z že pridobljenimi informacijami o ptici, pa bomo uporabili za ločevanje ptic glede na starost. V ta namem bomo izdelali barvni model, na podlagi katerega

bomo ptice primerjali med seboj in jih razvrščali v dva razreda, mladiče in odrasle.

#### 4.4.1 Obdelava obrisov

Kot že omenjeno, želimo iz krivulje obrisa ptice določiti njen nagib glede na očrtani pravokotnik z namenom pridobitve natančnejše informacije o velikosti ptice. Ker je krivulja obrisa telesa ptice izrazito podobna elipsi, in ker knjižnica OpenCV vsebuje učinkovito funkcijo za izračun krivulji prilegajoče se elipse, smo se odločili da ti dve dejstvi koristno povežemo. Na krivulji ptice tako izvedemo operacijo izračuna prilegajoče se elipse in s tem pridobimo dolžini krajše in daljše osi elipse ter nagnjenost ptice glede na očrtani pravokotnik. Na sliki 8 je prikazana ptica z očrtanim pravokotnikom, krivuljo obrisa in prilegajočo se elipso.



**Slika 8: Ptica in geometrijske informacije. Vijolična – očrtan pravokotnik, svetlo modra – krivulja obrisa, rumeno – krivulji obrisa prilegajoča se elipsa.**

Dodatna prednost uporabe prilegajoče se elipse je, da učinkovito deluje tudi v primeru, ko je del ptice izven vidnega polja. Tak primer vidimo na sliki 9, ko se velikost elipse zelo dobro ujema z vidnim delom ptice, nakazuje pa nam celotno telo ptice, tudi zakriti del. In to kljub temu, da orisana krivulja in očrtani pravokotnik obkrožata samo vidni del telesa ptice.



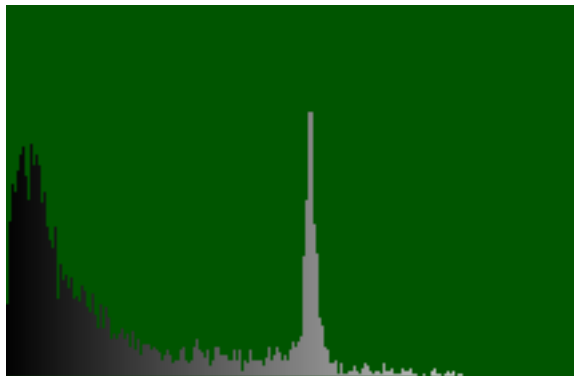
**Slika 9:** Krivulji prilagajóča se elipsa dobro oceni velikost ptice, tudi ko je ta delno zakrita.

#### 4.4.2 Obdelava barvnih slik objektov

Z uporabo začetne barvne slike ptic v kombinaciji z že pridobljenimi informacijami o pticah želimo izluščiti informacijo, s katero bo možno razvrščati ptice glede na oceno njihove starosti. Ptice želimo razvrstiti v dva razreda, mladiče in odrasle. Vemo, da se odrasle ptice razlikujejo od mladičev po temnejši obarvanosti perja (osebna korespondenca s prof. F.H. Heppnerjem z oddelka za biologijo Univerze v Rhode Islandu, ZDA). Natančneje, obarvanost perja mladičev je tem svetlejša bolj kot gremo od trupa proti glavi, pri odraslih pticah pa ta prehod ni zelo izrazit. Ta postulat nam bo služil za osnovo pri snovanju postopka za klasifikacijo. Cilj postopka je iz slike konkretne ptice pridobiti podatek, ki bo omogočal njeno razvrstitev glede na postulat o različni obarvanosti perja med mladiči in odraslimi. Osnovna operacija nad barvnimi vrednostmi slike ali dela slike je izdelava njihovega histograma. Histogram je množica zbranih seštevnikov količin iste vrste, razporejenih v določene razrede. Uporablja se za prikaz statistične porazdelitve vrednosti merjenih količin po razredih in ponavadi zmanjša število dimenzij opazovanih podatkov. Zaradi tega je zelo uporabno orodje pri iskanju vzorcev in izvajanju primerjav množic podatkov. Histogram dela slike je množica seštevnikov pikslov glede na barvo za obravnavano področje. Ker sta funkciji izdelave in primerjave histogramov dobro podprti znotraj OpenCV, izberemo primerjanje histogramov kot merilo za ravrščanje ptic v razrede. Primer histograma slike ptice je na sliki 10.

Na histogramu opazimo izrazit vrh približno na sredini, ki je skoraj identičen pri vseh pticah. Ta vrh je *odsev žice*, ki na slikah prečka vse ptice. Z nekaj dodatnega dela bi bilo možno doseči, da se pri računanju histograma ta del izloči, vendar smo ocenili, da njegov prispevek

med primerjanjem histogramov ni bistven. To velja ravno zato, ker je pri vseh pticah praktično isti.



**Slika 10: Histogram ptice iz slike 8.**

Na tem mestu velja opomniti na dvoje. Prvič, z izbiro histogramov kot orodja za razvrščanje ptic smo upoštevali le prvi del postulata o razlikovanju odraslih ptic in mladičev. To je del, ki govori, da imajo odrasli v splošnem temneje obarvano perje. Drugi del, ki govori o izraziti smeri spreminjanja obarvanosti perja pri mladičih, pa smo izpustili. Kljub temu, da je klasifikacija z uporabo histogramov dokaj zanesljiva metoda, velja kot možno nadgradnjo algoritma razmisliti o izdelavi boljše reprezentacije podatkov o barvni razporeditvi, ki bi upoštevala poleg količinske tudi prostorsko komponento razporeditve barv. Drug primer nadgradnje bi bila razširitev uporabe pridobljenih histogramov tako, da se jih uporabi kot pripomoček pri segmentaciji slik. S tem bi lahko delno zrahljali začetne pogoje za vhodne slike.

#### **4.4.3 Sestavljeno znanje**

V razpoznavi smo združili dva sklopa operacij, ki se lahko izvajata vzporedno. Prvi sklop sestavlja operacija izračuna ptici prilegajoče se elipse, s katero smo pridobili natančnejše podatke o velikosti ptice. Drugi sklop sestavljata operaciji izdelave histograma za barvno sliko ptice in njegova uporaba za razvrstitev v razreda mladičev ali odraslih. Potek operacij je



#### 4.5.1 Število ptic na fotografiji

Kot že omenjeno v razdelku 4.3, dobimo število najdenih objektov neposredno iz postopka segmentacije. Ker smo zaradi predpostavk pri razvoju algoritma in podanih omejitev vhodnih podatkov prepričani, da so vsi najdeni objekti tudi ptice, lahko to število obravnavamo kar kot najdeno število ptic. Med izvajanjem programa lahko s pogledom na sliko ptic z označenimi najdenimi objekti ta podatek takoj tudi preverimo, kot avtomatizacijo preverjanja pa bi na tem mestu lahko uporabili v prejšnjem razdelku omenjeno možnost uporabe histogramov pri določanju objektov.

#### 4.5.2 Velikost posamezne ptice v pikslih in skupna povprečna velikost ptic

Zelo kvaliteten podatek o velikosti posamezne ptice dobimo iz izračuna krivulji obroisa ptice prilegajoče se elipse. Ta operacija za vsako posamezno ptico izračuna nagnjenost in ovalnost telesa glede na pogled kamere, ter iz tega prilagojeno širino in višino (velikost) ptice. Za izračun skupne povprečne velikosti ptic (PV) uporabimo formulo (1), pri čemer so  $v_i$  višine vseh ptic na dani sliki.

$$PV = \frac{1}{n} \sum_{i=1}^n v_i \quad (1)$$

#### 4.5.3 Medsebojne oddaljenosti ptic in povprečna medsebojna oddaljenost, merjeno v povprečni velikosti ptice

Pri izračunu medsebojnih oddaljenosti ptic začnemo v levem zgornjem kotu in napredujemo od leve proti desni ter od vrha navzdol. Razdalje računamo iz pozicij (x,y) centrov teles ptic po formuli (2).

$$d(A, B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \quad (2)$$

Razdalje, dobljene po (2) so v piksljih. Za preračun v povprečne velikosti ptice uporabimo vrednost PV.

#### 4.5.4 Razvrščanje ptic na odrasle in mladiče glede na obarvanost perja

Ko imamo histograme ptic, je za potrebe razvrščanja potrebno določiti še referenčne histograme, s katerimi se bodo izvajale primerjave. Za potrebe testiranja ročno razvrstimo po enega tipičnega predstavnika ptic v razred, z njim primerjamo ostale ptice in jih razvrstimo glede na ujemanje.

Ujemanje histogramov preverjamo s statistično metodo  $\chi^2$ , ki je vgrajena v OpenCV. Mera za podobnost histogramov je razdalja Hi-kvadrat, ki je definirana z:

$$d_{\chi^2}(H_1, H_2) = \sum_i \left( \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)} \right) \quad (3)$$

Večja kot je vrednost razdalje (3), manjše je ujemanje primerjanih histogramov. Razdalja nič pomeni popolno ujemanje.

## 5 REZULTATI

Realiziran program smo preizkusili tako, da smo testne slike ptic najprej dali v analizo analitiku, potem pa smo iste slike analizirali še s pomočjo našega sistema. Analiza s sistemom je potekala tako, da smo posamezno sliko naložili, po potrebi prilagodili parametre, in sprožili izračun analize. Parametri programa so razdeljeni v tri stopnje, ki ustrezajo stopnjam algoritma (glej prejšnje poglavje).

Parametri obdelave slike so:

- izbira barvnega odtenka za segmentacijo;
- izbira širine dovoljenega odstopanja od izbranega barvnega odtenka za segmentacijo;
- moč oz. stopnja glajenja.

Parameter segmentacije je:

- najmanjša in največja velikost obrisa, ki ga še obravnavamo.

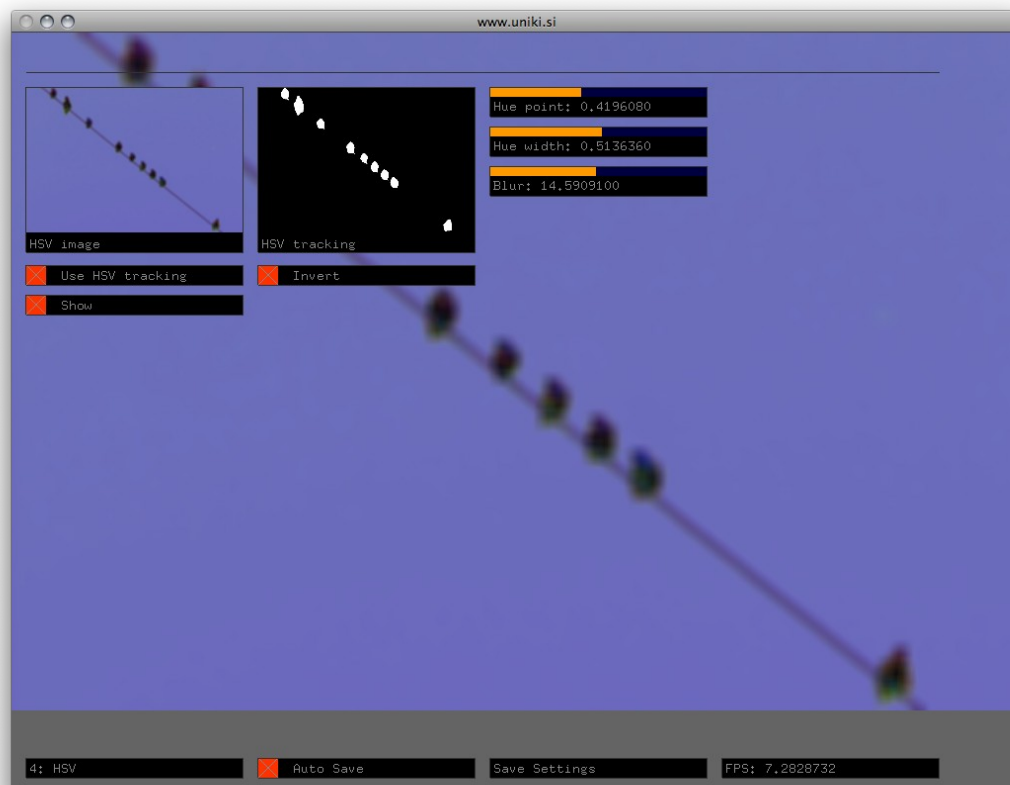
Parameter razpoznavne je:

- določitev referenčnih histogramov za mladiča in za odraslo ptico.

Vsi parametri se shranijo v datoteko XML in se ohranijo pri ponovnem zagonu programa. Poleg nastavitve parametrov lahko v uporabniškem vmesniku nastavljamo še različne možnosti pri prikazu poteka obdelave z namenom omogočiti lažje eksperimentiranje s parametri. Za segmentacijo npr. lahko izbiramo katere informacije o najdenih objektih so prikazane, pri razpoznavi pa lahko izbiramo histogram katere ptice želimo gledati.

Na sliki 12 vidimo uporabniški vmesnik za nastavitve parametrov obdelave slike. S klikom na katerokoli točko v pomanjšani sliki v HSV barvnem prostoru se vrednost te točke shrani in

uporabi za pragovno funkcijo v segmentaciji. Predogled rezultata pragovne funkcije lahko vidimo v drugi pomanjšani sliki. Nastavljamo lahko tudi moč oz. stopnjo glajenja, katerega rezultat vidimo na glavni sliki, ki zaseda celotno vidno polje programa. Sprememba glajenja vpliva tudi na pragovno funkcijo, zato je tudi rezultat prilagajanja glajenja moč videti v pomanjšanem oknu predogleda pragovne funkcije.



Slika 12: Uporabniški vmesnik z nastavitvami za obdelavo slike.

Ker je zaradi zelo omejenega števila slik učna množica hkrati tudi testna množica, so ugotovljeni rezultati obdelave lahko precej zavajajoči. Zaradi velike medsebojne podobnosti nabora slik smo namreč lahko ključne parametre zelo dobro izbrali in tako dosegli izvrsten rezultat razpoznavanja ptic in vseh zelenih lastnosti. Na drugi strani pa se pojavi vprašanje ali bi bilo možno parametre podobno dobro izbrati tudi ob večji raznovrstnosti vhodnih slik, torej ali bodo obstoječi parametri podobno učinkoviti tudi na novih testnih slikah brez dodatnega prilagajanja.

Pri delu na testni množici slik smo ugotovili, da je parametre obdelave slike in segmentacije možno nastaviti tako, da so učinkoviti za vse tri slike brez vmesnega prilagajanja. To pomeni, da je na tej točki z izdelanim sistemom možno doseči določeno stopnjo avtomatizacije. Pri operaciji razpoznavne pa se je izkazalo, da so histogrami, kljub na oko podobni obliki, slabo prenosljivi med slikami. Zato smo izbirali referenčna histograma mladiča in odraslega na vsaki sliki posebej. Ugotovimo torej lahko, da je potreben dodatni vložek v razvoj metrike, ki bi bila uporabna znotraj večje množice vhodnih slik.

V nadaljevanju so predstavljene analize treh slik ptic z rezultati ročnega preštevanja in merjenja ter z rezultati obdelave z izdelanim sistemom računalniškega vida. Kategorije v rezultatih so sledeče:

- PV, povprečna velikost ptice;
- dolžina v px, dolžina oz. velikost ptice v pikslih;
- razdalja v px, razdalja od trenutne do desne sosednje ptice v pikslih (zadnja ptica nima zaznane desne sosede, zato ima to vrednost prazno);
- razdalja v PV, preračunana iz razdalja v px in povprečne velikosti ptice PV;
- $P_{\text{mladič}}$ , verjetnost, da je trenutna ptica mladič; ročna analiza je tu natančnejša, naš algoritem ima samo dva pola: mladič in odrasel;
- ptice štejemo od leve proti desni in od zgoraj navzdol.



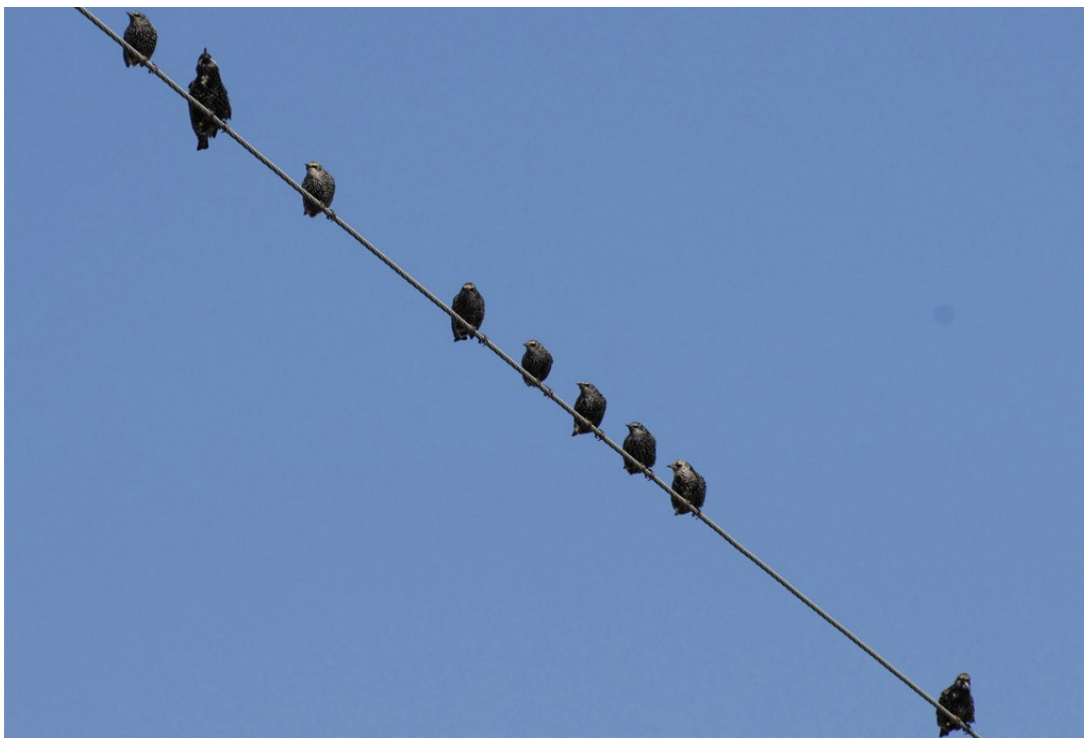
Slika 13: Vhodna slika birds2.jpg.

Tabela 2: Ročna analiza slike birds2.jpg.

| birds2.jpg    |      |      |      |      |      |      |      |   |       |
|---------------|------|------|------|------|------|------|------|---|-------|
| ptica št.     | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8 | PV    |
| dolžina v px  | 107  | 104  | 94   | 94   | 89   | 93   | 95   |   | 96,57 |
| razdalja v px | 313  | 98   | 111  | 111  | 118  | 130  | 85   |   |       |
| razdalja v PV | 3,24 | 1,01 | 1,15 | 1,15 | 1,22 | 1,35 | 0,88 |   |       |
| Pmladič       | 0,5  | 0,75 | 0,75 | 0,75 | 1    | 0,75 | 0,75 | 0 |       |

Tabela 3: Avtomatizirana analiza slike birds2.jpg.

| birds2.jpg    |        |       |        |        |        |        |       |        |       |
|---------------|--------|-------|--------|--------|--------|--------|-------|--------|-------|
| ptica št.     | 1      | 2     | 3      | 4      | 5      | 6      | 7     | 8      | PV    |
| dolžina v px  | 99,22  | 92,08 | 83,01  | 83,36  | 80,56  | 79,4   | 82,26 | 115,68 | 89,45 |
| razdalja v px | 310,88 | 97,68 | 110,42 | 112,11 | 119,22 | 129,52 | 79,19 |        |       |
| razdalja v PV | 3,48   | 1,09  | 1,23   | 1,25   | 1,33   | 1,45   | 0,89  |        |       |
| Pmladič       | 1      | 1     | 1      | 1      | ref 1  | 1      | 1     | ref 0  |       |



Slika 14: Vhodna slika birds3.jpg.

Tabela 4: Ročna analiza slike birds3.jpg.

| birds3.jpg    |      |      |      |      |      |      |      |      |    |       |
|---------------|------|------|------|------|------|------|------|------|----|-------|
| ptica št.     | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9  | PV    |
| dolžina v px  | 51   | 90   | 50   | 55   | 44   | 49   | 50   | 52   | 65 | 56,22 |
| razdalja v px | 86   | 133  | 182  | 80   | 66   | 60   | 57   | 325  |    |       |
| razdalja v PV | 1,53 | 2,37 | 3,24 | 1,42 | 1,17 | 1,07 | 1,01 | 5,78 |    |       |
| Pmladič       | 0,2  | 0    | 0,2  | 0    | 0    | 0    | 0    | 0    | 0  |       |

Tabela 5: Avtomatizirana analiza slike birds3.jpg.

| birds3.jpg    |       |        |        |       |       |       |       |        |       |       |
|---------------|-------|--------|--------|-------|-------|-------|-------|--------|-------|-------|
| ptica št.     | 1     | 2      | 3      | 4     | 5     | 6     | 7     | 8      | 9     | PV    |
| dolžina v px  | 58    | 85,84  | 55,72  | 59,27 | 46,17 | 54,49 | 52,15 | 52,44  | 58,76 | 58,09 |
| razdalja v px | 85,03 | 133,81 | 180,58 | 80,13 | 65,4  | 60,44 | 58,33 | 323,25 |       |       |
| razdalja v PV | 1,46  | 2,3    | 3,11   | 1,38  | 1,13  | 1,04  | 1     | 5,56   |       |       |
| Pmladič       | 1     | 0      | ref 1  | ref 0 | 0     | 0     | 0     | 0      | 0     |       |



Slika 15: Vhodna slika birds6.jpg.

Tabela 6: Ročna analiza slike birds6.jpg.

| birds6.jpg    |      |      |      |      |      |      |      |      |      |       |
|---------------|------|------|------|------|------|------|------|------|------|-------|
| ptica št.     | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | PV    |
| dolžina v px  | 95   | 96   | 89   | 89   | 97   | 112  | 94   |      |      | 96,00 |
| razdalja v px | 110  | 117  | 118  | 131  | 88   | 90   | 219  | 78   |      |       |
| razdalja v PV | 1,15 | 1,22 | 1,23 | 1,36 | 0,92 | 0,94 | 2,28 | 0,81 |      |       |
| Pmladič       | 0,60 | 0,60 | 0,75 | 0,60 | 0,70 | 0,00 | 0,20 | 0,20 | 0,20 |       |

Tabela 7: Avtomatizirana analiza slike birds6.jpg.

| birds6.jpg    |        |        |        |        |       |        |        |       |       |       |
|---------------|--------|--------|--------|--------|-------|--------|--------|-------|-------|-------|
| ptica št.     | 1      | 2      | 3      | 4      | 5     | 6      | 7      | 8     | 9     | PV    |
| dolžina v px  | 91,87  | 92,09  | 85,74  | 84,11  | 89,54 | 111,86 | 84,29  | 69,25 | 88,81 | 88,62 |
| razdalja v px | 112,28 | 113,55 | 116,65 | 127,47 | 82,46 | 87,59  | 216,02 | 79,96 |       |       |
| razdalja v PV | 1,27   | 1,28   | 1,32   | 1,44   | 0,93  | 0,99   | 2,44   | 0,9   |       |       |
| Pmladič       | 1      | 1      | 1      | 1      | ref 1 | ref 0  | 1      | 0     | 1     |       |

## 6 ZAKLJUČEK

Z uporabo tehnologije računalniškega vida in knjižnice OpenCV smo realizirali sistem, ki omogoča analizo slik ptic v mirovanju. Na primerih smo prikazali, da je kvaliteta podatkov iz tako dobljene analize primerljiva s kvaliteto podatkov, ki jih pridobi analitik, ki opravlja ročno analizo.

Uporabili smo tri temeljne sklope operacij za izgradnjo računalniškega vida, ki so obdelava slike, segmentacija in razpoznavanje. Znotraj teh sklopov smo z zlaganjem enostavnih operacij dobili učinkovite metode, ki so skupaj dale želen rezultat v obliki algoritma, ki rešuje zastavljen problem.

Določene operacije algoritma so odvisne od parametrov, zato smo v uporabniški vmesnik programa vgradili prilagajanje teh parametrov in sprotno opazovanje učinkov. Ob preizkušanju algoritma na množici vhodnih slik smo ugotovili, da so nekateri parametri dovolj splošni, da s pravo vrednostjo lahko obdelamo več slik, nekateri pa so se izkazali za manj učinkovite in je potrebno nastavljanje za vsako sliko posebej. Na tem mestu čim prej pričakujemo izboljšave, nekaj možnosti je omenjenih v nadaljevanju.

Kot vidimo, so za rešitev danega problema uporabljene operacije računalniškega vida dovolj učinkovite, v splošnem pa uporabljeni pristopi spadajo pod dokaj enostavne prijeme, vsaj za današnji čas ko procesorska moč skokovito narašča. Zato ni nič čudnega, da se je tekom izdelave porodilo precej idej za razširitve, ki stremijo predvsem za tem, da se čimbolj razrahljajo omejitve pri vhodnih podatkih. Nekaj teh idej povzemamo v nadaljevanju tega poglavja.

## **6.1 Možne izboljšave**

### **6.1.1 Povečanje množice vhodnih podatkov**

Kot že nekajkrat omenjeno, je glavna izboljšava možna s povečanjem množice vhodnih podatkov. Le na ta način bi lahko podali končno oceno realiziranega sistema računalniškega vida in ugotovili potencialne šibke točke začetnih predpostavk ter omejitve uporabljenih metod.

### **6.1.2 Izboljšava posameznih korakov**

V zadnjem odstavku razdelka 4.4.2 smo že omenili, da uporaba histogramov za ugotavljanje barvnega ujemanja in s tem klasifikacije le delno upošteva postulat o razliki obarvanosti ptic. Menimo, da je tu še precej prostora za izboljšave z uporabo dodatnih prijemov, ki bi upoštevali prostorsko komponento razporeditve barv.

V istem odstavku je tudi omemba uporabe pridobljenega histograma obarvanosti ptic kot pripomočka pri segmentaciji. Menimo, da je ob povečanju množice slik zelo verjetno, da enostavna segmentacija na podlagi barve neba ne bo več dala zanesljivih rezultatov. Zato je souporaba segmentiranja na podlagi histogramov ena od možnosti kako povečati to zanesljivost.

Obe omenjeni izboljšavi je možno še dodatno oplemenititi z uvedbo strojnega učenja. Ob povečanju raznolikosti vhodnih podatkov želimo, da bi se, skozi ročno označevanje referenčnih ptic kot poteka sedaj, sistem sčasoma sam naučil, katere ptice so primerne reference za razločevanje.

### **6.1.3 Delovanje v realnem času**

Trenutno je izdelana programska oprema namenjena analizi posamezne vhodne slike naenkrat. Ker so uporabljeni algoritmi računsko obvladljivi, ni za takšno omejitev nobenega resnega razloga, saj se iz uporabniškega stališča vsi koraki algoritma izvedejo praktično v trenutku. Zelo koristna izboljšava bi bila torej ta, da priredimo sistem tako, da deluje v realnem času. To pomeni, da bi lahko analitik na terenu z zoom kamero snemal jato ptic, v prenosnem računalniku pa bi se mu sproti gradil model opazovane jate.

### **6.1.4 Uporaba 3D kamere**

Kot najbolj drzno razširitev sistema predlagamo možnost uporabe 3D kamere in z njo možnost spremljanja medsebojnih odnosov ptic v letu. Kljub dodatni dimenziji pričakujemo, da se pri dovolj majhnih jatah kompleksnost algoritma ne bi bistveno povečala. Postopka filtriranja in segmentacije bi se verjetno celo nekoliko poenostavila, posotpek opdelave objektov, pa bi moral pri izračunavanju razdalj upoštevati še tretjo dimenzijo.

## 7 SEZNAM UPORABLJENIH VIROV

- [1] (junij 2010) [http://en.wikipedia.org/wiki/Computer\\_vision](http://en.wikipedia.org/wiki/Computer_vision)
- [2] D. Forsyth, Ponce, Computer Vision A Modern Approach, Prentice Hall, 2003
- [3] (junij 2010) [http://en.wikipedia.org/wiki/Ubiquitous\\_computing](http://en.wikipedia.org/wiki/Ubiquitous_computing)
- [4] (junij 2010) <http://sourceforge.net/projects/artoolkit/>
- [5] (junij 2010) [http://en.wikipedia.org/wiki/Augmented\\_reality](http://en.wikipedia.org/wiki/Augmented_reality)
- [6] D. Skočaj, D. Omerčevć, A. Leonardis, Uporaba umetnega spoznavnega vida, Ljubljana, 2006.
- [7] P. Remagnino, G. L. Foresti, T. Ellis, Ambient intelligence: a novel paradigm, Springer, 2005.
- [8] G. Bradski, A. Kaehler, Learning OpenCV, O'Reilly, 2009.
- [9] (junij 2010) <http://www.openframeworks.cc/>
- [10] I. Lebar Bajec, F. H. Heppner, 2009, Organized flight in birds, Animal Behaviour, 78, str. 777.789.
- [11] (junij 2010) R. Szeliski, Computer Vision: Algorithms and Applications (osnutek prosto dostopen na naslovu <http://szeliski.org/Book/>)