

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Iztok Žužek

**Vizualno prepoznavanje kovancev  
z računalniškim vidom**

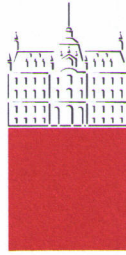
DIPLOMSKO DELO  
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Aleš Leonardis

Ljubljana, 2012

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*



Št. naloge: 01799/2012

Datum: 01.02.2012

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

Kandidat: **IZTOK ŽUŽEK**

Naslov: **VIZUALNO PREPOZNAVANJE KOVANCEV Z RAČUNALNIŠKIM  
VIDOM**

**VISUAL RECOGNITION OF COINS BY COMPUTER VISION**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Proučite uporabo računalniškega vida za prepoznavanje kovancev. Analizirajte obstoječe pristope in na tej podlagi predlagajte primerno metodo, ki jo implementirajte in eksperimentalno ovrednotite na različnih zbirkah kovancev.

Mentor: 

prof. dr. Aleš Leonardis

Dekan:



prof. dr. Nikolaj Zimic



# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani Iztok Žužek,

z vpisno številko 63050137,

sem avtor diplomskega dela z naslovom:

Naslov diplomskega dela

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 14.05.2012

Podpis avtorja:



# Zahvala

Zahvalil bi se mentorju prof. dr. Alešu Leonardisu za vodenje in pomoč pri izdelavi diplomske naloge. Prav tako bi se zahvalil še Domnu Taberniku za tehnično pomoč pri uporabi digitalne kamere.

Posebna zahvala gre tudi mojim staršem, ki so me tekom študija podpirali tako finančno kot moralno.

# Kazalo

<b>Povzetek</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>1 Uvod</b>	<b>3</b>
1.1 Organizacija diplomske naloge . . . . .	5
<b>2 Pregled metod za detekcijo kovancev</b>	<b>6</b>
2.1 Metode za detekcijo kovancev . . . . .	6
2.2 Obdelava slike . . . . .	8
2.2.1 Cannyjev filter . . . . .	9
2.2.2 Houghova transformacija . . . . .	9
2.2.3 Segmentacija slike . . . . .	10
2.2.4 SIFT . . . . .	10
2.2.5 Interpolacija po metodi najbližjega sosedu . . . . .	11
2.2.6 Spiralna dekompozicija . . . . .	12
2.2.7 Fourierjevi koeficienti v polarni obliki . . . . .	12
2.3 Klasifikacija kovancev . . . . .	12
2.3.1 Evklidska razdalja . . . . .	13
2.3.2 City-block razdalja . . . . .	13
2.3.3 Križna korelacija . . . . .	14
2.3.4 Odštevanje slike z invariantnostjo rotacije . . . . .	14
<b>3 Izbira ustrezne metode</b>	<b>15</b>
3.1 Problemi pri detekciji . . . . .	15
3.2 Kriteriji izbire . . . . .	15
3.3 Struktura . . . . .	17
3.4 Opis metode . . . . .	18
3.4.1 Pretvorba v sivinsko sliko . . . . .	18
3.4.2 Neobčutljivost na šum . . . . .	19

3.4.3	Lokalizacija . . . . .	20
3.4.4	Invariantnost skaliranja . . . . .	21
3.4.5	Invariantnost rotacije . . . . .	22
3.4.6	Klasifikacija . . . . .	23
<b>4</b>	<b>Rezultati in analiza</b>	<b>24</b>
4.1	Strojna in programska oprema . . . . .	24
4.2	Implementacija . . . . .	25
4.2.1	Pretvorba v sivinsko sliko . . . . .	25
4.2.2	Iskanje robov . . . . .	26
4.2.3	Iskanje kovancev v sliki . . . . .	27
4.2.4	Izločanje kovanca iz slike . . . . .	27
4.2.5	Normalizacija slike . . . . .	28
4.2.6	Spiralna dekompozicija . . . . .	28
4.2.7	Iskanje podobnosti . . . . .	29
4.3	Zajem testnih podatkov . . . . .	29
4.4	Rezultati testiranja . . . . .	30
4.4.1	Postopek testiranja . . . . .	30
4.4.2	Grafi . . . . .	30
<b>5</b>	<b>Sklepne ugotovitve</b>	<b>34</b>
	<b>Dodatki</b>	<b>35</b>
<b>A</b>	<b>Dodatek</b>	<b>35</b>
	<b>Seznam slik</b>	<b>37</b>
	<b>Seznam tabel</b>	<b>39</b>
	<b>Literatura</b>	<b>40</b>

# Seznam uporabljenih kratic in simbolov

**SIFT** —vizualni opisnik (ang. Scale Invariant Feature Transform)

**RGB** —barvni prostor (ang. R - red, G - green, B - blue)

# Povzetek

V diplomski nalogi smo se osredotočili na problem računalniškega vizualnega prepoznavanja kovancev v avtomatih. To prepoznavanje sodi na področje računalniškega vida. V uvodu smo povzeli zgodovino razvoja avtomatov in njihovih mehanizmov preverjanja kovancev. Prišli smo do spoznanja, da je potrebno zaradi velikega števila podobnih kovancev, ki izhajajo iz različnih držav, obstoječim mehanizmom dodati računalniško vizualno preverjanje. Ugotovili smo, da so se s tem problemom začeli ukvarjati Fukuni in ostali že v začetku 90ih let prejšnjega stoletja. V glavnem delu smo povzeli pregled metod za detekcijo kovancev, ki so jih v zadnjem času objavili Chalechale in ostali. Med njimi smo poiskal najbolj primerno metodo, ki je ustrezala našim kriterijem. Izbrali smo Chalechalejevo metodo in teoretično opisal njeno delovanje, ki smo jo kasneje uporabil za implementacijo te metode. Metodo smo tudi testiral na dveh bazah slik, in sicer smo testiral njeno uspešnost pri pravilni detekciji. Dobil sem podobne rezultate kot Chalechale in Shen. Kasneje smo metodo testiral še glede na rotacijo vhodnih slik in število kategorij v učni množici, pri čemer sem ugotovil, da ima rotacija slik zanemarljiv vpliv. To potrjuje, da je metoda invariantna na rotacijo, druga ugotovitev pa je, da ob povečevanju kriterijev ni tako visok padec uspešnosti pravilne detekcije. V zadnjem testu smo preverjali, če lahko izboljšamo rezultate tako, da izpustimo zunanji kolar kovanca, kjer je ponavadi besedilo, vendar smo ugotovili, da s tem samo za malenkost izboljšamo rezultate. Metodo je kljub solidnim rezultatom potrebno dodatno izboljšati, kar smo predlagali v zaključku.

## Ključne besede:

prepoznavanje kovancev, računalniški vid, detekcija kovancev

# Abstract

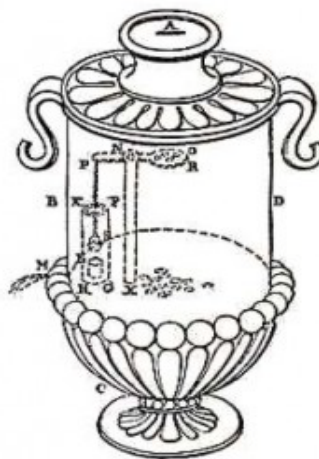
The diploma paper focuses on the issue of computer visual recognition of coins in automatic machines. The recognition falls within the scope of computer vision. The introduction summarizes the history of development of automatic machines and their mechanisms of coins recognition. We came to the conclusion that due to a number of similar coins from different countries, the existing mechanisms should be upgraded by computer visual recognition. We found out that this issue had already been addressed by Fukuni and others in the beginning of nineties of the 20th century. The central part of the thesis summarizes the overview of coin detection methods that were recently published by Chalechale and others. Among these we selected the most appropriate method which met our criteria. We chose the Chalechale's method and theoretically described its functioning, which was subsequently used for the implementation of this method. The method was also tested on two groups of pictures. We tested its effectiveness with respect to correct detection. The results obtained were fairly similar to those of the Chalechale's and Shen's. Later on, we tested the method against the rotation of input images and the number of categories within a data set. We established that, firstly, the rotation of images has an insignificant effect, which proves that the method is invariable to rotation, and secondly that the increase of categories does not cause significant decrease in correct detection effectiveness. The last test verified whether the results could be improved by leaving out the outer coin ring, which usually contains text, however we found out that by doing so the results were insignificantly improved. Despite good results, the method needs many further improvements, some of which we suggested in the conclusion.

**Key words:** coin recognition, computer vision, coin detection

# Poglavje 1

## Uvod

Prvi prodajni avtomat [19] je bil izumljen že v 1. stoletju v Grčiji. Izdelal ga je grški matematik Hero Aleksandrije. Ta prodajni avtomat, ki ga vidimo na sliki 1.1, je za denar ponujal sveto vodo. Mehanizem je deloval tako, da je ob prejemu kovanca skozi zgornjo režo posode začela za krajši čas teči sveta voda. To pa je bilo izvedeno tako, da je vhodni kovanec padel na skrito notranjo ročico, s čimer je za kratek čas odprl dovod vode, dokler ni padel z ročice na dno posode.



Slika 1.1: Prvi prodajni avtomat v 1. stoletju, ki je prodajal sveto vodo [18].

Šele v drugi polovici 19. stoletja so se začeli pojavljati različni komercialni avtomati, namenjeni razglednicam, znamkam in žvečilkam. Pri teh so izdelali

prvi mehanizem, ki je vseboval preverjanje velikosti kovancev. Če je bil kovanec premajhen, ga je mehanizem sprejel, vendar se ob vrtenju ni zgodilo nič, v nasprotnem primeru pa ni bilo možno spraviti kovanca skozi režo. Na začetku je mehanizem sprejemal zgolj eno velikost kovancev, kasneje so dodali možnost sprejemanja različnih velikosti. Takrat se je pojavil prvi problem, kako kovanke ločiti med seboj. V ta namen so začeli postopoma dodajati mehanične metode, ki so preverjale premer, debelino in težo kovanca, ter v manjši meri tudi magnetizem. Ta metoda pa ni razlikovala različnih materialov. Kasneje so to izboljšali z uvedbo elektromagnetne metode, pri kateri kovanec pri določeni frekvenci potuje skozi oscilacijsko tuljavo. Različni materiali so imeli različen odziv, ki se je odražal na spremembo amplitude in smeri frekvence.



Slika 1.2: Velika podobnost med kovancema: poljski zlot in madžarski forint.

S toliko različnimi pregledovalnimi mehanizmi bi pričakovali, da bodo zadovoljivo učinkoviti, vendar je okoli nas veliko držav in skoraj prav toliko različnih valut z različnimi kovanci, med katerimi se vedno najdejo tudi takšni, ki jih po določenih atributih ne bi prepoznali, saj bi si bili enostavno preveč podobni po teži, velikosti, debelini ter podobnosti materiala. Kot na primer vidimo na sliki 1.2, sta si kovanca po teh atributih preveč podobna, da bi ju bilo mogoče ločiti med seboj. Tako moramo uvesti še dodatno preverjanje, ki bo preverjalo, kar je vkovano na kovancu. To ni trivialen problem, saj ima že človeški vid veliko težav pri prepoznavanju kovancev, kar najbolj opazimo v trgovini, ko nam trgovec vrne denar napačne valute ali celo ponarejen denar. Tega po vsej



verjetnosti ne bomo prepoznali kot ponarejenega. Iskanje rešitve za pričujoči problem se prične šele v začetku 90. let prejšnjega stoletja, ko so akademiki ob pomoči računalnikov, ki so se v tistem času dodobra razširili, začeli raziskovati področje vizualnega računalniškega zaznavanja tipov objektov, kot so kovanci. Eden prvih člankov, ki so se pojavili na temo detekcije kovancev, je Fukumijev [5], v katerem je predlagal, da bi za vizualno detekcijo kovancev uporabljali računalniški algoritem na osnovi nevronske mreže, pri čemer bi nevronske mreže vzorcev zgradili s pomočjo velike baze slik kovancev, ki so rotirani pod vsemi koti. Vendar ta rešitev ni bila zrela za uporabo na avtomatih, saj je bila preveč kompleksna in prepočasna, pa tudi ne dovolj učinkovita. Težave, s katerimi se srečujejo te metode, so velika občutljivost na motnje, kot so svetloba, umazanost kovanca, poškodovanost ter zabrisanost podobe na njem, ki so posledica pogoste uporabe. Zato moramo uporabiti različne metode vizualnega računalniškega zaznavanja, ki bodo odpravile probleme, s katerimi se srečujemo pri obdelavi. Pri tem moramo biti učinkoviti, saj preverjamo denar, kar bi ob napačni napovedi prineslo izgubo. Poleg tega mora biti avtomatski postopek kar se da hiter, saj uporabnik prodajnega avtomata ne želi dolgo čakati na njegov odziv.

## 1.1 Organizacija diplomske naloge

Osrednji del diplomske naloge je razdeljen na tri poglavja. V prvem poglavju bomo obširneje pregledali robustnejše metode, ki so se pojavile v zadnjem času in se tudi uporabljajo. V drugem poglavju se bomo osredotočili na prednosti in slabosti metod, predstavljenih v prvem poglavju. Z njihovo pomočjo bomo našli najustreznejšo metodo in jo kasneje implementirali. V tretjem poglavju bomo nazorneje prikazali lastno implementacijo metode in njeno delovanje na praktičnih primerih, nato pa z različnimi testi prikazali, kako se naša metoda obnese na določenih bazah slik in kakšen vpliv imajo na natančnost prepoznavanja različni parametri.

## Poglavje 2

# Pregled metod za detekcijo kovancev

### 2.1 Metode za detekcijo kovancev

V svetu so se s problemom detekcije kovancev že marsikje srečali, kar potrjujejo članki [1, 2, 6, 8, 11], z mnogih koncev sveta. Vsem člankom je skupno izhodišče, in sicer poiskati najučinkovitejšo metodo, ki bo sposobna kar se da dobro prepoznati prave kovance.

V članku [1] Chalechale razdeli svojo metodo na dva dela. Prvi del je poimenovan abstrakcija slike, s katero obdeluje sliko po korakih. V prvem koraku vhodno sliko kovanca spremeni v sivinsko, nato jo obdela s Cannyjevim filtrom (glej 2.2.1), s čimer dobi vidni kovanec samo z robnimi točkami. Sledi normalizacija slike s pomočjo interpolacije po metodi najbližjega sosedu (glej 2.2.5), pri čemer sliki spremeni velikost na  $J \times J$ , v članku je uporabljena velikost  $129 \times 129$ . V drugem delu, ki je poimenovan spiralna dekompozicija, naprej obdela normalizirano sliko, kateri s pomočjo spiralne dekompozicije (glej 2.2.6) razdeli sliko kovanca na enako široke kolobarje in kreira spiralni vektor, v katerem so vsote slikovnih elementov na posamičnem kolobarju. V zadnjem koraku drugega dela izračuna city-block razdaljo (glej 2.3.2), v kateri kreirani spiralni vektor primerja z drugimi spiralnimi vektorji iz učne množice.

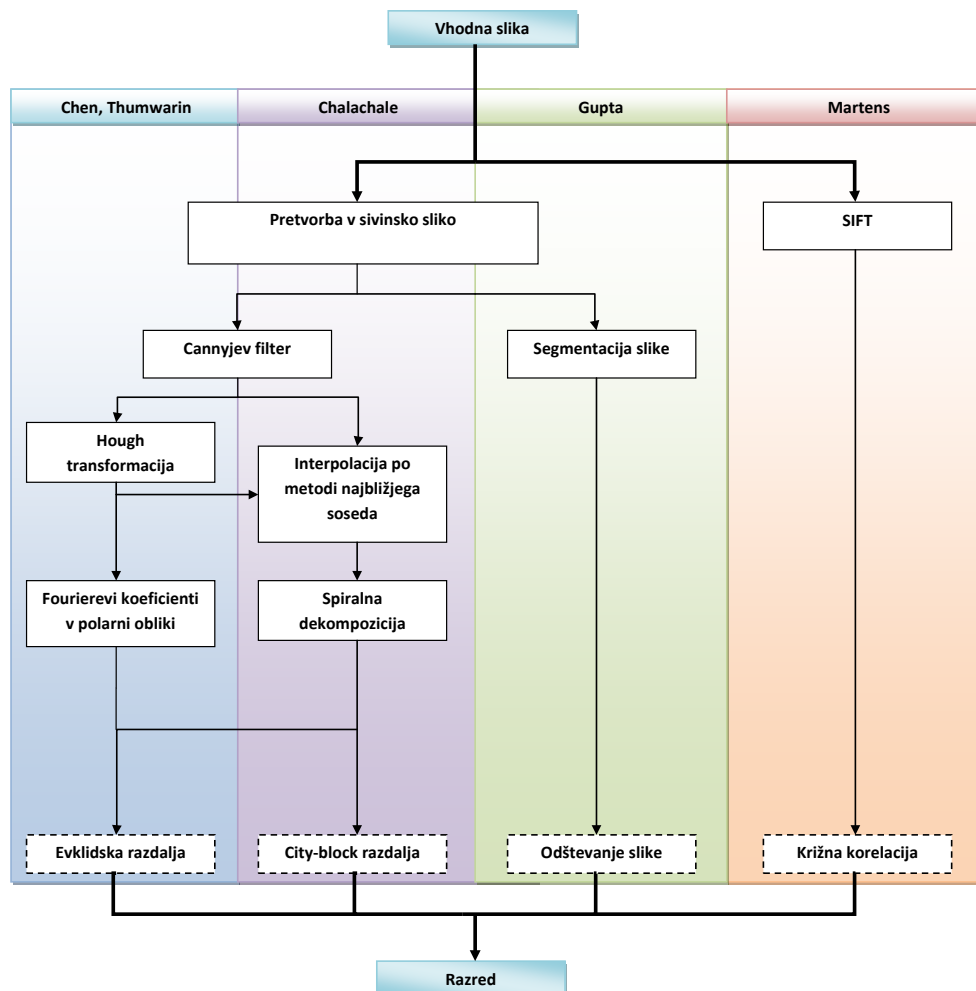
V članku [6] Gupta s sodelavci razdeli metodo na več korakov. V prvem koraku prav tako kot v prejšnjem članku [1] prvo vhodno sliko spremenijo v sivinsko. Vendar se metoda nadaljuje z uporabo segmentacije slike (glej 2.2.3), ki sliki ojači kontrast, zaradi česar se bolje vidijo razlike med kovanecem in njegovim ozadjem. Kontrastno sliko nato spremeni v binarno, v kateri je kovanec označen z vrednostjo slikovnega elementa 1. Krogu kovanca poišče njegov radij,

s katerim kovanec izreže iz sivinske slike. Pri tej novo pridobljeni izrezani sivinski sliki uporabi odštevanje slik (glej 2.3.4), s katerim najprej sešteje vrednosti vseh sivinskih slikovnih elementov. To vsoto sivinskih vrednosti pa odšteje od vseh razredov sivinskih slik kovancev iz učne množice, v kateri imamo za vsak razred shranjeno zbirko zaporedja sivinskih slik, ki so rotirane z vsemi koti. To izvede v dveh fazah. V prvi fazi pregleda razlike vsot sivin učnim kovancem pri kotih v razmiku  $10^\circ$  in za vsak razred izdela histogram razlik oziroma razdalj med slikama pod vsakim kotom, pri čemer dobi minimum razdalje. Če je vrednost minimuma manjša od pragovne vrednosti, preide v drugo fazo, sicer zavrne razred. Druga faza je enaka prvi s to razliko, da računa samo razdalje blizu kota minimuma, vendar z učnimi slikami v razmiku kota  $1^\circ$ . Če tudi pri tem vrednost minimuma ne preseže pragovne vrednosti, kovanec označi kot pripadnika tega razreda.

V naslednjem članku [8] uporabi zgolj dva koraka. V prvem koraku obdela sliko s pomočjo metode SIFT (glej 2.2.4), pri kateri vhodno sliko transformira v veliko zbirko lokalnih vektorskih značilnic. V drugem koraku s pomočjo križne korelacije (glej 2.3.3), izračuna razdaljo med testno in učno sliko. Vendar to ne naredi na celi sliki, temveč samo nad bloki slike z velikostjo  $20 \times 20$  slikovnih elementov s centrom lokacije značilnic kovanca, ki so bile izračunane prej.

V zadnjih dveh člankih [2, 11] so avtorji ponovno razdelili metodo na več korakov. V prvem koraku se, v primeru da je vhodna slika barvna, po navadi ta pretvori v sivinsko sliko. V naslednjem pa s pomočjo Cannyjevega filtra (glej 2.2.1) spremenijo sivinsko sliko v binarno sliko robov, ki jo potem uporabijo za Houghovo transformacijo za krožnice (glej 2.2.2). S tem poiščejo krožnico kovanca in pripadajoči radij, s katerim uspešno izrežejo kovanec iz sivinske slike. Izrezani sliki izračunajo Fourierove koeficiente v polarni obliki (glej 2.2.7), ki jih dobijo s pomočjo Fourierove transformacije. V zadnjem koraku dobljeni vektor Fourierovih koeficientov uporabijo pri računanju evklidskih razdalj (glej 2.3.1) med testno sliko in ostalimi učnimi slikami. Pri učni sliki, v kateri je razdalja najmanjša in je hkrati manjša od pragovne vrednosti, testna slika pripada istemu razredu kot učna.

Na sliki 2.1 so prikazane vse štiri opisane metode, vsaka v svoji barvi. Modra predstavlja Chenovo metodo [2, 11], vijolična Chalechaleovo metodo [1], zelena Guptovo [6] in rdeča Martensovo metodo [8]. Dobro je razvidno, v čem so si metode različne ali podobne. Pri tem tudi vidimo povezave med koraki ter kako lahko dele metod uporabimo in sestavimo novo. V našem primeru sta si prvi dve metodi zelo blizu (modri in vijolični stolpec), razlikujeta pa se samo v srednji fazi.



Slika 2.1: Struktura metod, ki se uporabljajo pri prepoznavanju kovancev.

## 2.2 Obdelava slike

Preden lahko določimo razred objekta, je prej potrebno obdelati sliko do te mere, da iz nje dobimo neko uporabno obliko zapisa objekta, s katero v naslednji fazi klasificiramo dani objekt. V nadaljevanju, bomo opisali pod metode, ki se uporabljajo pri tej obdelavi slike.

### 2.2.1 Cannyjev filter

Cannyjev filter [12] je eden najbolj razširjenih detektorjev robov. Z njim lahko detektiramo robove v sliki tudi, če je prisoten šum. Deluje na osnovi odvodov slike in Gaussovega filtra.



Slika 2.2: Prikaz detekcije robov s Cannyjevim filtrom [1].

### 2.2.2 Houghova transformacija

Houghova transformacija [12, 9] je glasovalna metoda, ki se uporablja za iskanje parametričnih oblik, kot so npr. ravne črte in krogi. Pri iskanju kovancev v sliki nam pomaga poiskati njim pripadajoče krožnice. Njena glavna prednost je, da lahko detektiramo več kovancev hkrati, slabost pa, da je časovno zelo zahtevna.



Slika 2.3: Iskanje krožnic kovancev s Houghovo transformacijo [13].

### 2.2.3 Segmentacija slike

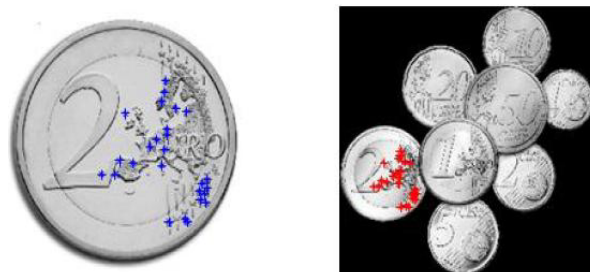
Segmentacija slike [6] nam pomaga poiskati kovanec na sliki. Sama metoda deluje tako, da sivinski sliki povečujemo kontrast do te mere, da lahko s pomočjo pragovne vrednosti pretvorimo sliko v binarno sliko. Tako dobimo območje kovanca, v katerem je vrednost slikovnega elementa enaka 1 oziroma je bele barve. Najmanjša in največja oddaljenost belega slikovnega elementa od izhodišča, ki je na sliki levo zgoraj, nam pove meje kovanca, s katerim izračunamo njegov radij, ki ga lahko uporabimo pri rezanju kovanca iz slike. Prednost te metode je v enostavnosti in časovni nezahtevnosti, vendar pa je občutljiva na šum.



Slika 2.4: Prikaz korakov segmentacije slike [6].

### 2.2.4 SIFT

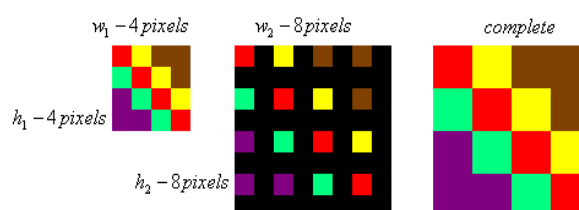
SIFT [7, 8] je metoda, ki transformira sliko v zbirko lokalnih vektorskih značilnic, s katerimi lahko v kasnejših fazah poiščemo podobne značilnice objekta na testni sliki. Metoda je invariantna na translacijo, povečavo/pomanjšavo, rotacijo ter delno na spremembo svetlosti.



Slika 2.5: Prikaz uporabe SIFT metode na sliki z večjim številom kovancev [8].

### 2.2.5 Interpolacija po metodi najbližjega soseda

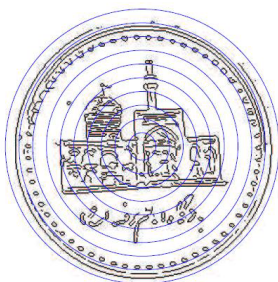
Interpolacija po metodi najbližjega soseda [16] je ena najpogostejše uporabljanih metod spreminjanja velikosti slike. Deluje tako, da se nova slika zgradi iz vrednosti slikovnih elementov osnovne slike, kateri slikovni element se uporabi, pa izračunamo po metodi najbližjega soseda. Dobra stran te metode je, da je lahka za implementacijo in ni časovno potratna. Njena največja pomanjkljivost pa, da ob zmanjšanju slike izgubimo informacijo o vrednostih slikovnih elementov, kar nam ne dovoli vrnitve na prvotno sliko.



Slika 2.6: Prikaz spreminjanja velikosti slike iz 4 x 4 slikovnih elementov na 8 x 8 slikovnih elementov s pomočjo interpolacije najbližjega soseda [16].

### 2.2.6 Spiralna dekompozicija

Spiralni dekompoziciji [1] gre za razdelitev slike na enako široke kolobarje, kot to vidimo na sliki 2.7, v vsakem izmed njih štejemo vse robne točke. Tako dobimo vektor vsot robnih točk, kar potem uporabimo v eni od klasifikacijskih metod, s katero primerja razlike med vzporednima elementoma. Prednost te metode je, da ni občutljiva na rotacijo kovanca.



Slika 2.7: Prikaz razdelitve kovanca na enakomerne kolobarje [1].

### 2.2.7 Fourierjevi koeficienti v polarni obliki

Fourierjevi koeficienti v polarni obliki [2, 11] je metoda, pri kateri najprej koordinate slike pretvorimo v polarno obliko in ji po Fourierjevi transformaciji izračunamo pripadajoči vektor za potrebe klasifikacije. Metoda je invariantna na rotacijo, da pa je še invariantna na odboj svetlobe na površini kovanca, se poleg teh koeficientov uporabi še Fourierjeva aproksimacija.

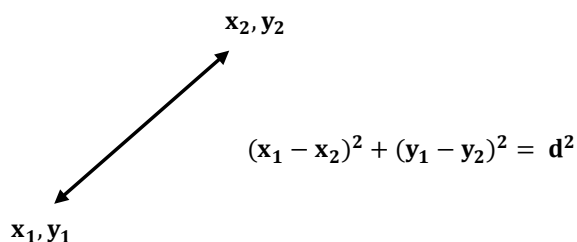
## 2.3 Klasifikacija kovancev

Po obdelavi slike uporabimo klasifikacijske metode, s katerimi lahko ugotovimo, kateremu razredu pripada iskani kovanec. Na tem področju poznamo več pristopov. V nadaljevanju bomo prikazali najpogostejše metode na področju klasifikacije kovancev.



### 2.3.1 Evklidska razdalja

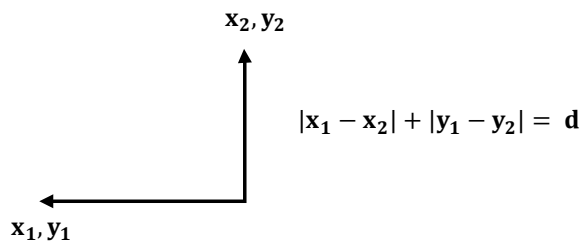
Evklidska razdalja [14] je ena najbolj poznanih razdalj, ki se uporablja na večjih področjih, kjer v osnovi merimo razdaljo med dvema točkama v prostoru, med katerima ni ovir.



Slika 2.8: Evklidska razdalja med dvema točkama v 2D prostoru [14].

### 2.3.2 City-block razdalja

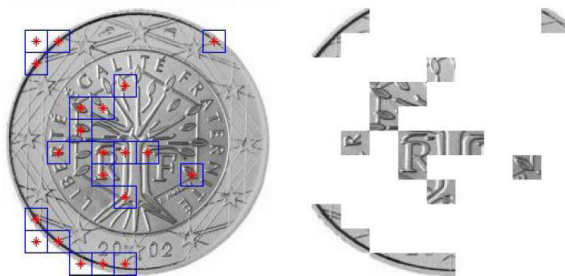
City-block razdalja ali Manhattnova razdalja [14] pokriva podoben problem kot evklidska razdalja, vendar upošteva ovire na poti, zaradi česar ne gre direktno, vendar po ovinkih, kot da bi jo na poti ovirali bloki in jih je potrebno obiti.



Slika 2.9: City-block razdalja med dvema točkama v 2D prostoru [14].

### 2.3.3 Križna korelacija

Križna korelacija [8] je metoda, ki z računsko zahtevno operacijo izračuna podobnosti dveh slik oziroma njunih delov in hkrati odpravi občutljivost na svetlobo. Na sliki 2.10 je bila uporabljena metoda le nad deli, kjer so bile najdene značilnice.



Slika 2.10: Uporaba križne korelacije na izbranih delih [8].

### 2.3.4 Odštevanje slike z invariantnostjo rotacije

Odštevanje slike z invariantnostjo rotacije [6] je klasifikacijska metoda, ki s pomočjo več sivinskih učnih slik pri različnih kotih primerja razliko med vsotama sivinske testne slike in učne sivinske slike pri nekem kotu. Med temi vsotami poišče minimum; če je minimum manjši od pragovne vrednosti, pomeni, da smo našli pravi kovanec. Metoda je časovno zahtevnejša, saj potrebuje večje število vhodnih slik, vendar pa je tudi občutljiva na šum in na svetlost.

## Poglavje 3

# Izbira ustrezne metode

### 3.1 Problemi pri detekciji

Problemi pri detekciji se pojavijo že na samem začetku, in sicer pri zajemu slik. Pri tem imamo opraviti z odbojnim svetlobnim virom zaradi odbojne kovinske površine, kar pomeni, da v primeru ko je površina kovanca boljše očiščena, je tudi odboj svetlobe večji. Problem lahko omilimo z večjim številom svetlobnih virov, ki oddajajo svetlobe z različnih strani, pri čemer je svetlobni vir bela svetloba. Pri tem lahko uporabimo krožni polarizacijski filter, ki ga pritrdimo na objektiv kamere. Poleg tega se tu pojavijo še problem različnih oddaljenosti od kamere, različne lokacije kovanca na sliki, različne rotacije in šum pri zajemu slike.

### 3.2 Kriteriji izbire

Pri izbiri ustrezne metode smo uporabili spoznanja iz člankov [1, 2, 6, 8, 11]. Naš cilj je bil, da najdemo čim ustrežnejšo metodo, ki bo razmeroma enostavna za implementacijo in čim robustnejša ter invariantna na rotacijo, premik, povečavo/pomanjšavo, poleg tega pa kar se da neobčutljiva na šum in spremembo svetlobe.

Tako se lahko lotimo ocenjevanja ustreznosti prve metode iz članka [1]. Pri tej metodi Chalechale razdeli postopek na 5 korakov, ki jih ni težko implementirati in so tudi razmeroma hitri. Pri tem upošteva vse kriterije, ki smo jih podali, razen invariantnost premika, ki ga v članku niti ne potrebuje, saj obdeluje že prej obrezane slike kovancev. Metoda se je v članku izkazala za izrazito učinkovito. Vendar obstaja pomislek glede testov, saj so bile za testne

slike uporabljene zasukane učne slike, obrnjene s pomočjo metode integer pixel shifts za faktorje  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ . S tem so dokazali dobro invariantnost na rotacije. Vendar obstaja dvom o robustnosti glede na vhodne testne slike, ki niso zasukane učne slike.

Naslednja metoda, ki se pojavi v članku [6], je sestavljena iz mnogo manjših korakov, ki so tudi enostavni za implementacijo. Kar se tiče pokrivanja kriterijev, je pokrit večji del, vendar se v delu, v katerem režemo kovanec iz slike, pojavi prevelika občutljivost na šum, ker pri računanju radija kovanca poišče lokacijo najmanj in najbolj oddaljene robne točke od zgornjega levega kota slike. Ob veliki verjetnosti napačne šumne robne točke dobimo napačno lokacijo kovanca in njegovega radija. Metoda tudi ni invariantna na povečavo/pomanjšavo, ker ločuje kovance po njihovem radiju. Je tudi časovno zelo potratna, saj mora za vsak testni kovanec pregledati veliko bazo kovancev v učni množici, v kateri za vsakega hrani sliko za vseh 360 kotov.

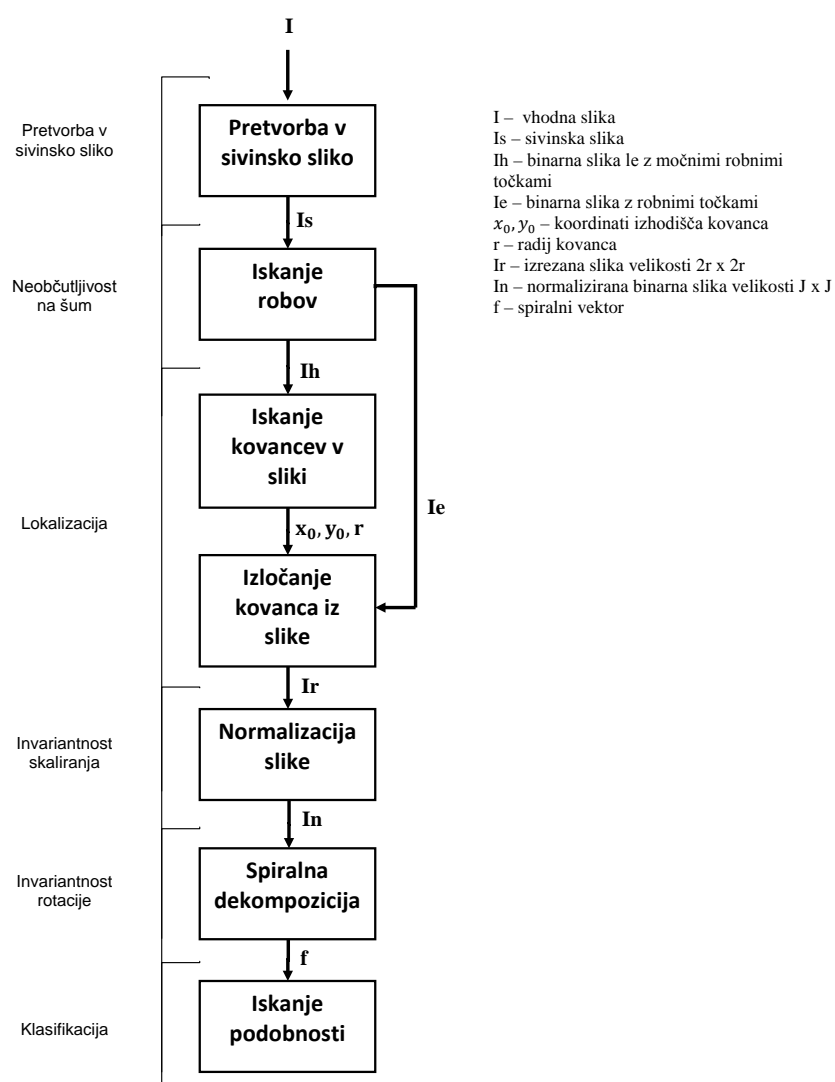
Metoda iz članka [8], sestavljata dva koraka, od katerih je prvi zelo kompleksen. Ta prek SIFT metode poišče značilnice na sliki. Metoda ustreza kar vsem občutljivostim kriterija, vendar pa je kompleksna za implementacijo. V članku pogrešamo le rezultate testiranja, kar bi potrdilo njeno učinkovitost.

Zadnja metoda iz člankov [2, 11], po njihovih testiranjih deluje zelo robustno. Ustreza večini našim kriterijem, vendar pa kot večina metod tudi ta ni invariantna na povečavo/pomanjšavo. Glede implementacije pa ni tako kompleksna in izvajanje algoritma nima zelo velike časovne odvisnosti. V članku [1], sta bili skupaj testirani ta in prva metoda, ob čemer se je izkazalo, da se prva metoda bolje obnese tudi pri različni rotaciji kovanca.

Po temeljitnem pregledu se je izkazalo, da danim kriterijem najbolj ustreza Chalechalejeva metoda [1], ki je sestavljena tako, da z vsakim korakom premaga ali pa vsaj omili eno težavo. Tako se pri prvem koraku, ko pretvori barvno sliko v sivinsko, omili občutljivost na odboj svetlobe. S pomočjo Cannyjevega filtra (glej 2.2.1 omilimo šum na sliki, pri naslednjem koraku, ko uporabimo interpolacijo po metodi najbližjega sosedu (glej 2.2.5) se znebimo težave z različnimi velikostmi slik, ter še pri predzadnjem koraku, ko uporabimo spiralno dekompozicijo (glej 2.2.6) in z njo dosežemo invariantnost rotacije. Vendar nam na koncu ostane odprto le še pokritje kriterija invariantnosti premika kovanca, pri čemer imamo na voljo več načinov, s katerimi smo se srečali pri prejšnjih metodah. Da bi bila naša metoda čim manj modificirana in bi pri tem najmanj izgubili na času, si lahko pomagamo s sliko povezav med metodami 2.1, kjer vidimo, da je najbolj smiselna uporaba Houghove transformacije (glej 2.2.2), kajti predhodni korak Houghove transformacije je nek detektor robov, kar pa naša metoda že ima v osnovni obliki.

### 3.3 Struktura

V tem pod poglavju bomo prikazali strukturo naše izbrane metode.



Slika 3.1: Prikaz strukture delovanja metode.

Na sliki 3.1 je prikazana struktura metode, ki na vходу dobi navadno sliko ( $I$ ), na kateri je kovanec. To sliko v prvi fazi spremeni v sivinsko sliko ( $Is$ ), iz katere potem naslednja faza iskanja robov po metodi Cannyjevega filtra (glej 2.2.1) ustvari binarno sliko. Prva binarna slika ( $Ie$ ) je namenjena kasnejši

uporabi pri rezanju slike. Druga binarna slika ( $I_h$ ), na kateri je dobro vidna le krožnica, pa se uporabi v fazi iskanja kovancev v sliki, ko s pomočjo Houghove transformacije (glej 2.2.2) poiščemo koordinati središča krožnice ( $x_0, y_0$ ) ter njen radij ( $r$ ). S pomočjo teh koordinat, radija ter binarne slike robov ( $I_e$ ), ki nastopajo kot vhodni podatki faze izločanja kovanca iz slike, dobimo izrezan kovanec v binarni sliki robov ( $I_r$ ). Nastalo sliko ( $I_r$ ) potem normaliziramo na fiksno velikost  $J \times J$  s pomočjo interpolacije najbližjega sosedu (glej 2.2.5). Dobljeno normalizirano sliko ( $I_n$ ) v fazi spiralne dekompozicije (glej 2.2.6) razdelimo na enakomerne kolobarje, pri čemer vsakemu posamezno izračunamo vsoto robov in dobimo spiralni vektor ( $f$ ) z dolžino števila kolobarjev. Do te faze sodijo pod obdelavo slike. Iz tega dela metode se lahko premaknemo v zadnjo fazo iskanja podobnosti s pomočjo city-block razdalje (glej 2.3.2), ko dobljeni spiralni vektor ( $f$ ) primerjamo z učno množico že pripravljenih spiralnih vektorjev iste dolžine. Iz tega dela klasifikacije dobimo podatek, kateremu razredu je najbližji testni primerek. Te faze so v podpoglavju 4.2 tudi prikazane s primeri. Na sliki levo od faz tudi vidimo kriterije, ki jih te faze zagotavljajo, in v nadaljevanju bomo tudi teoretično podrobno opisali njihovo delovanje.

## 3.4 Opis metode

Osnovna oblika izbrane metode [1] ne vsebuje rezanja kovancev iz slike, saj v članku Chalechale [1] uporabi že vnaprej obdelane baze slik in kovanec pokriva celotno sliko. V našem primeru bomo uporabili baze slik, v katerih je kovanec na poljubni lokaciji. Zaradi tega je pri tej metodi potrebna modifikacija. V ta namen smo dopolnili metodo s Houghovo transformacijo (glej 2.2.2) za krožnice, ki nam bo poiskala lokacijo kovancev poljubne velikosti. Kot smo že omenili, je naša metoda razdeljena na več delov oziroma korakov, ki jih bomo v nadaljevanju podrobneje opisali.

### 3.4.1 Pretvorba v sivinsko sliko

V prvemu koraku vhodno barvno sliko pretvorimo v sivinsko. Barvna slika tipa RGB vsebuje tri barvne kanale: rdečega (Red), zelenega (Green) in modrega (Blue). Pri sivinski barvi imamo samo en barvni kanal, zato moramo tri kanale zmanjšati na enega. To naredimo tako, da vse tri kanale med seboj isto ležne slikovne elemente seštejemo in jih normaliziramo. Vendar pri tem uporabimo določene deleže posameznih barvnih kanalov, vrednost teh deležev

nam prikazuje enačba 3.1. Smisel tega postopka je v osnovi, da imamo opraviti zgolj z enim kanalom, kar nam olajša delo in pospeši nadaljnje operacije.

$$S = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (3.1)$$

### 3.4.2 Neobčutljivost na šum

Pri drugem koraku se metoda ukvarja z iskanjem robov v sliki. V ta namen uporabi Cannyjev filter [1, 12], kot smo ga že omenili v podpoglavju 2.2.1, je eden najbolj uporabljenih filtrov za iskanje robov v sliki. V našem primeru smo robove izrazili s Cannyjevim operatorjem pri  $\sigma = 1$  in Gaussovi maski dolžine = 9. Da lahko opišemo močne robove, potrebujemo Gaussov 1D filter  $G$ , in njegovo jedro  $g$ . Enačba, ki jo potrebujemo se glasi:

$$H(k) = \sum_i G(i)g(k + 1 - i). \quad (3.2)$$

Ta predstavlja 1-D konvolucijo med Gaussovim filtrom  $G$  in njegovim jedrom  $g$ . Potem filtriramo sliko preko osi  $X$  in  $Y$  z naslednjima enačbama:

$$X(u, v) = \left[ \sum_{j=1}^V I'(u, j) \right]', \quad (3.3)$$

$$Y(u, v) = \sum_{i=1}^U I(i, v)H(u - i), \quad (3.4)$$

kjer je  $I$  sivinska slika,  $X$  je parcialni odvod sivinske slike po  $x$  osi in  $Y$  je parcialni odvod sivinske slike po  $y$  osi ter pri čemer je  $u = 1, 2, \dots, U$  in  $v = 1, 2, \dots, V$ .  $U$  predstavlja število vrstic in  $V$  predstavlja število stolpcev v sliki  $I$ . V naslednjem koraku izračunamo moč robov  $\Gamma$ , ki pomenijo moč robov v sliki  $I$ , to izračunamo z enačbo:

$$\Gamma(u, v) = \sqrt{X(u, v)^2 + Y(u, v)^2}. \quad (3.5)$$

Če želimo na sliki dobiti le pravilne robove brez šuma, moramo določiti spodnjo in zgornjo pragovno vrednost. V članku [1] Chalechale predlaga, da se meji dobijo s pomočjo kumulativnega 64-celičnega histograma iz sivinske slike. Iz tega bi poiskali indeks  $\iota$ , ki je na lokaciji minimuma vrednosti kumulativnega histograma in je večji od  $\alpha * U * V$ , pri čemer je  $\alpha$  delež ne robnih točk na sliki. Za večino primerov se izkaže, da je najboljša izbira tega deleža vrednost 0.7. Tako lahko določimo zgornjo pragovno vrednost, ki se glasi  $\beta * \iota$  in spodnja

pragovna vrednost, ki je  $0.4 * \beta * \iota$ , pri čemer je  $\beta$  parameter, ki kontrolira stopnjo vidnih robov na sliki: večji ko je, manj robov se prikaže. Pravilo, ki določa, kaj se detektira kot rob, se glasi:

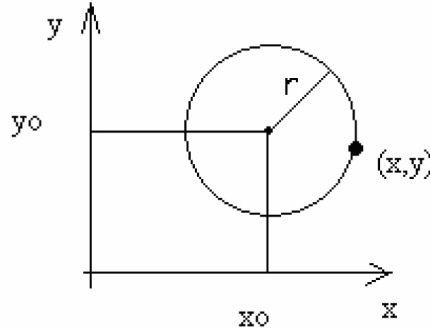
$$Ie(u, v) = \begin{cases} 1; & 0.4 * \iota * \beta \leq \Gamma(u, v) \leq \iota * \beta \\ 0; & \text{sicer} \end{cases} \quad (3.6)$$

Pri tem je  $\iota$  indeks komutativnega grafa, kjer je minimum deleža sivinske slike.

### 3.4.3 Lokalizacija

V nadaljevanju metoda preide v razreševanje lokalizacije kovanca na sliki; v Chalechalovem [1] članku so bile slike prej pripravljene tako, da so bili kovanci že izrezani iz slike, pri čemer ni bilo potrebno uporabiti te metode. Vendar pa se po navadi srečamo s slikami, v katerih je kovanec na poljubni lokaciji. Ena od metod za iskanje kovancev v sliki je Houghova transformacija za krožnice [9](glej 2.2.2), osnovna oblika transformacije je za detekcijo črt, vendar je v našem primeru potrebno detektirati krožnice. Kot vhodni podatek v Houghovo transformacijo dobi binarno sliko robov, podobno kot je bilo s Cannyjevim filtrom, vendar z drugačnim pravilom določanja robov. V našem primeru,  $\Gamma(u, v) \geq 0.7 * \max \Gamma$ , dobimo čim manj robov, ki še določajo iskano krožnico. Glavni razlog, da imamo tako malo robov, je, da hitreje preiščemo iskani prostor, saj imamo tridimenzionalen problem, v katerem iščemo središče krožnice, ki je v točki  $S$ . Ta ima koordinati  $x_0$  in  $y_0$  ter radij krožnice  $r$ , ki ga izračunamo po formuli 3.7, pri čemer koordinati  $x$  in  $y$  predstavljata robno točko na krožnici, kot vidimo na sliki 3.2. Vendar vse točke ne ležijo na iskani krožnici. Da bi ugotovili, katere točke ležijo na krožnici, moramo uporabiti glasovalni način, ki ga uporablja Houghova transformacija. Glasovanje deluje tako, da najprej ustvari tridimenzionalno matriko, velikosti  $X * Y * R$ , v kateri so vsi elementi enaki 0, pri čemer je  $X$  število vrstic in  $Y$  število stolpcev v sliki,  $R$  pa število različnih celoštevilskih radijev velikosti  $R = \frac{X}{2}$ . Tej matriki pravimo tudi akumulator. Zgornja meja je postavljena na  $\frac{X}{2}$ , ker radij kovanca ne more biti večji kot polovica slike, pri čemer predpostavljamo, da se na sliki kovanec vedno prikaže v celoti. Potem pa po enačbi 3.7 izračunamo radije oz. razdalje med ne robnimi in robnimi točkami in vse radije, manjše od  $\frac{X}{2}$ , akumulator jim povečamo za ena. Na koncu pregleda dobimo največjo vrednost akumulatorja. Njegova pozicija v tridimenzionalni matriki nam pove, v katerih koordinatah je izhodišče kovanca in njegovega radija. S pomočjo teh treh parametrov izrežemo kovanec iz slike, pri katerem se koordinate X-a nahajajo med  $x_0 - r$  in  $x_0 + r$  ter koordinate Y-a med  $y_0 - r$  in  $y_0 + r$ .





Slika 3.2: Prikaz koordinatnega sistema krožnice [9].

$$r = \sqrt{(x_0 - x)^2 + (y_0 - y)^2} \quad (3.7)$$

### 3.4.4 Invariantnost skaliranja

Ko imamo izrezano binarno sliko, se lotimo normalizacije velikosti slike kovanca, s katero se izognemo občutljivosti na različno oddaljenost kovanca od kamere. Normalizacija poteka s pomočjo interpolacije najbližjega sosedu (glej 2.2.5), ki za vhod vzame želeno velikost nove spremenjene slike, kot sta število vrstic  $h_2$  in število stolpcev  $w_2$  ter referenčno sliko. V našem primeru je to izrezana binarna slika. Pri njej preverimo njeno velikost, kot sta število vrstic  $h_1$  in število stolpcev  $w_1$ . S pomočjo teh dveh velikosti lahko z enačbama 3.8 izračunamo koeficienta razmerij  $x_k$  v smeri x in  $y_k$  v smeri y.

$$\left. \begin{aligned} x_k &= \frac{h_1}{h_2} \\ y_k &= \frac{w_1}{w_2} \end{aligned} \right\} w_2, h_1 \neq 0 \quad (3.8)$$

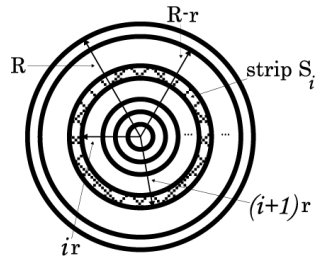
Potem sliko  $I_r$  spremenimo v sliko  $I_n$  velikosti  $J \times J$ , s pomočjo transformacijske enačbe

$$I_n(i, j) = I_r(i * x_k, j * y_k). \quad (3.9)$$

Pri tem indeksa nove slike  $I_n$  zavzemata zalogo vrednosti pri  $i = 1, 2, \dots, J$  in pri  $j = 1, 2, \dots, J$ .

### 3.4.5 Invariantnost rotacije

Naslednji korak je, da iz slike pridobimo uporaben podatek za kasnejšo kvalifikacijo. V ta namen se v tem delu uporabi metoda spiralne dekompozicije (glej 2.2.6), ki zgradi spiralni vektor. Spiralna dekompozicija temelji na dejstvih, da se ob vsakem zasuku slike kovanca, gledano iz izhodišča krožnice, premakne nek slikovni element z radijem  $\rho$  in kotom  $\theta$  na novo lokacijo, ob tej spremembi ostane radij, enak  $\rho$ , pri kotu pa se spremeni na kot  $\theta + \tau$ . Iz tega izhajamo, ko razdelimo kovanec na več kolobarjev. Na način, ki je prikazan na sliki 3.3, kjer so vsi kolobarji širine  $r$ . Tako se kolobar  $S_i$  prične pri radiju  $ir$  in konča pri radiju  $(i+1)r$ , z upoštevanjem da je  $i = 1, 2, \dots, N-1$ , kjer je  $N$  število vseh kolobarjev. Največji kolobar je čisto zunanji na robu krožnice, ki se prične pri radiju  $R-r$  in konča pri radiju  $R$ , ki je tudi radij krožnice.



Slika 3.3: Prikaz razdelitev kovanca na enakomerne kolobarje [1].

Naš spiralni vektor izračunamo s pomočjo enačbe 3.10, s katero v vsakem kolobarju  $S_i$  izračunamo vsoto vseh njegovih slikovnih elementov na normalizirani sliki  $I_n$  pri različnih radijih  $\rho$  in kotih  $\theta$ , pri čemer gredo indeksi spiralnega vektorja  $\vec{f}$  od  $k = 1, 2, \dots, N$ .

$$\vec{f}[k] = \sum_{\rho=(k-1)\frac{R}{N}}^{k\frac{R}{N}} \sum_{\theta=0}^{\theta=2\pi} I_n(\rho, \theta) \quad (3.10)$$

### 3.4.6 Klasifikacija

V zadnjem koraku pride na vrsto sama klasifikacija kovanca. V naši metodi se za ta namen uporabi City-block razdalja (glej 2.3.2). Preden lahko začnemo s klasifikacijo potrebujemo imeti zgrajeno matriko učnih spiralnih vektorjev  $U$ :

$$U_{N,L} = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & \cdots & u_{1,L} \\ u_{2,1} & \ddots & & & \vdots \\ \vdots & & u_{k,l} & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ u_{N,1} & \cdots & \cdots & \cdots & u_{N,L} \end{pmatrix} \quad (3.11)$$

Pri tem je  $u_{k,l}$  vsota slikovnih elementov pri  $k$ -tem kolobarju in  $l$ -tem učnem spiralnem vektorju ter  $N$  je število vseh kolobarjev in  $L$  število učnih spiralnih vektorjev. Potem lahko izračunamo razdaljo med testnim spiralnim vektorjem  $\vec{f}$  in učnimi spiralnimi vektorji v matriki  $U$  po enačbi:

$$\vec{d}_c[l] = \sum_{k=1}^N |u_{k,l} - f[k]|. \quad (3.12)$$

Pri tem je  $l = 1, 2, \dots, L$ ,  $L$  je število učnih spiralnih vektorjev, ter  $k = 1, 2, \dots, N$ , kjer je  $N$  število kolobarjev. Ko izračuna vse elemente vektorja razdalj  $\vec{d}_c$ , poiščemo testnemu spiralnemu vektorju  $\vec{f}$  pripadajoči razred  $l$ , pri katerem velja, da je  $\vec{d}_c[l] = \min \vec{d}_c$  ter  $\vec{d}_c[l] < \varepsilon$ . Pri čemer je  $\varepsilon$  neka zgornja meja razdalje, s katero lahko trdimo, da smo našli pravi kovanec v naši učni množici, v nasprotnem primeru ga zavrnamo.

# Poglavje 4

## Rezultati in analiza

### 4.1 Strojna in programska oprema

Strojna oprema, ki smo jo uporabili, je bila razdeljena v dva dela: prvi del je bil namenjen zajemu slik kovancev, drugi del pa opravljanju testov s pridobljenimi in zajetimi slikami kovancev. Za zajemanje slik smo uporabili digitalno kamero (glej dodatek A sliko A.2) z objektivom (glej dodatek A sliko A.1), ki smo jo pritrdili na improvizirano stojalo, kar kaže slika 4.1. Kot svetlobni vir smo uporabili kar dnevno svetlobo, ki je bila najprimernejša ob oblačnih dnevih, ko smo imeli na voljo belo svetlobo in najmanj odboja. Za dodatno zmanjšanje odboja smo na objektiv pritrdili krožni polarizatorski filter.



Slika 4.1: Stojalo s kamero, ki je zajemala slike s kovanci.

Strojna oprema, ki je služila za testiranje programov, na katero je bila priključena digitalna kamera, je bil stacionarni računalnik s procesorjem Intel Core 2 s frekvenco delovanja 2GHz ter z dinamičnim pomnilnikom velikosti 2GB.

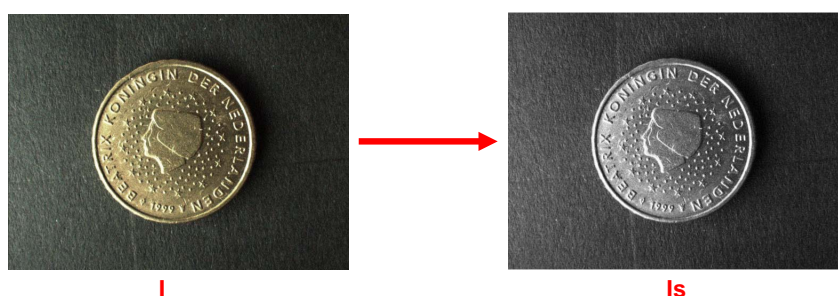
Pri programski opremi smo za delovno okolje uporabili operacijski sistem Microsoft Windows XP Profesional s 3. servisnim paketom. Program za zajemanje slik iz kamere je bil Point Grey Fly Capture 2, s katerim smo slikali slike s kovanci, resolucije 640 x 480, v barvnem kanalu RGB. Za poganjanje in pisanje naših programov smo uporabili Matlab verzije 7.11.0(R2010b) za delo v 32-bitnem okolju.

## 4.2 Implementacija

V strukturi metode (glej 3.3) vidimo, da metoda poteka v sedmih korakih, ki jih bomo v nadaljevanju tudi natančno opisali s pomočjo nazornih slik rezultatov korakov, ki smo jih uporabili pri diplomskem delu. Pri tem moramo izpostaviti, da smo za jasnejši prikaz celotne slike, ki prikazuje slike robov, uporabili inverzne slike in jih pri tem še dodatno poudarili

### 4.2.1 Pretvorba v sivinsko sliko

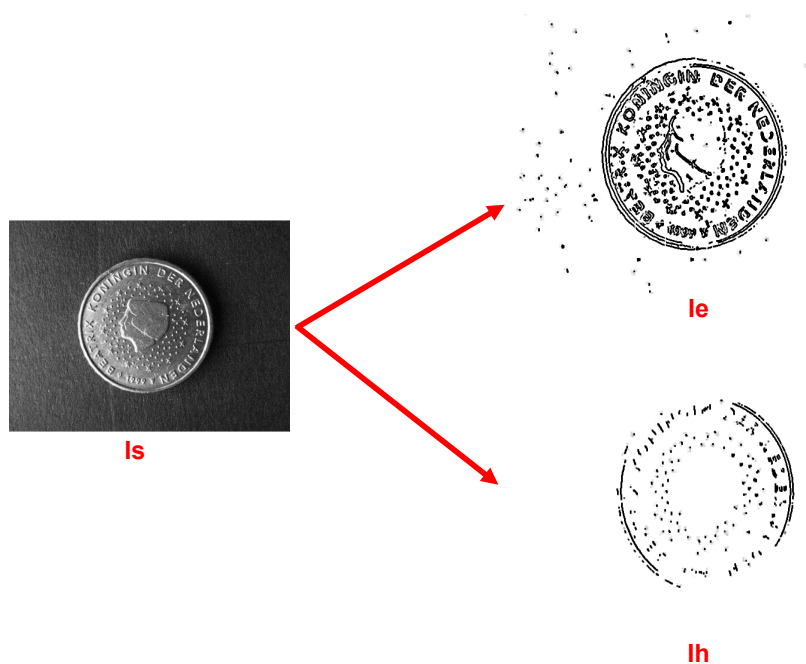
Prvi korak je najenostavnejši. Kot vidimo na sliki 4.2 vhodno sliko zgolj spremenimo v sivinsko sliko ob pogoju, da je bila ta prej barvna; če ugotovimo, da je že sivinska, jo pustimo takšno, kot je, in preskočimo ta korak. Pri implementaciji smo za to pretvorbo uporabili integrirano Matlab funkcijo `rgb2gray`, ki za pretvorbo uporablja enačbo 3.1, ki smo jo prej bolj podrobno opisali.



Slika 4.2: Prikaz transformacije slike I v sivinsko sliko Is.

### 4.2.2 Iskanje robov

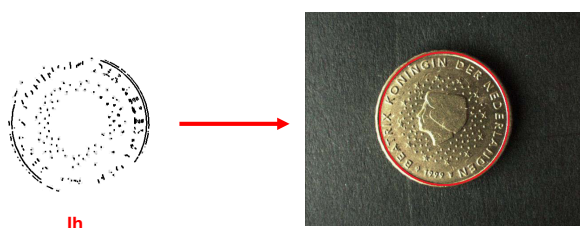
Pri sliki 4.3 vidimo, da pri drugem koraku vhodno sivinsko sliko  $I_s$  pretvorimo v binarni sliko robov  $I_h$  in  $I_e$ . Prva binarna slika robov  $I_h$  vsebuje le najmočnejše robove, da je jasno vidna krožnica. Z njeno pomočjo bomo lažje in hitreje našli središče kovanca v koraku 4.2.3. Pri drugi binarni sliki robov  $I_e$  pa imamo večje število robov, tako da so dobro vidne značilnice kovanca, ki ga bomo kasneje uporabili za obdelavo v koraku 4.2.4. Pri tem smo uporabili Cannyjev operator  $\sigma = 1$  in Gaussovo masko dolžine 9. Pravila 3.6, ki se uporablja za določanje robov, nismo uporabili, ker se med testom ni izkazalo kot dobra izbira. Zato smo za določitev spodnje pragovne vrednosti pri binarni sliki  $I_e$  vzeli  $\Gamma(u, v) > 45$ , pri binarni sliki  $I_h$  pa vzeli  $\Gamma(u, v) > 90$ , zgornje pragovne vrednosti nismo določili. Spodnja pragovna vrednost, ki smo jo izbrali, se je dobro obnesla v večina naših primerih.



Slika 4.3: Prikaz transformacije sivinske slike  $I_s$  v binarni sliki  $I_e$  in  $I_h$ .

### 4.2.3 Iskanje kovancev v sliki

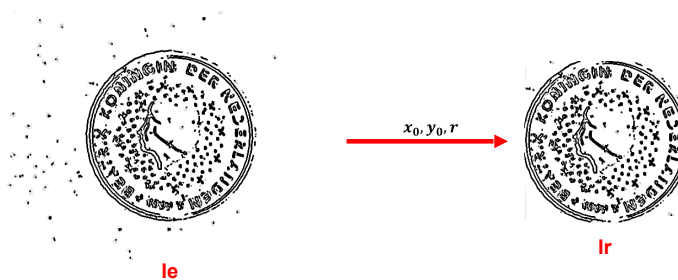
Na sliki 4.4 vidimo, da metoda v tem koraku uporabi binarno sliko robov  $I_h$ , pri kateri poišče središče krožnice s koordinatami  $x_0$  in  $y_0$ , ter njen polmer  $r$ . Pri implementaciji smo do teh atributov prišli, kakor je opisano v podpoglavju 3.4.3, pri čemer smo preiskovali v ožjem območju slike, da smo meje postavili na delih, kjer sta minimalna in maksimalna koordinata  $x$ -a in  $y$ -a robnih točk. Po tem omejenem območju smo se sprehajali v smeri  $x$ -a in  $y$ -a s korakom 2.



Slika 4.4: Prikaz iskanja središča krožnice ter radija v binarni sliki robov  $I_h$ .

### 4.2.4 Izločanje kovanca iz slike

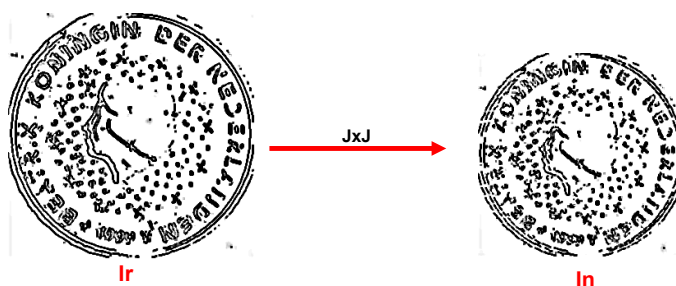
Pri tem koraku, kot vidimo na sliki 4.5, izločanja kovanca iz binarne slike  $I_e$  in dobimo novo binarno sliko  $I_r$ . To smo naredili s pomočjo v prejšnjem koraku izračunane koordinate središča krožnice  $x_0$  in  $y_0$  ter radija  $r$ .



Slika 4.5: Prikaz izločanja kovanca iz slike s pomočjo izhodiščih koordinat  $x_0$  in  $y_0$  ter radija  $r$ .

### 4.2.5 Normalizacija slike

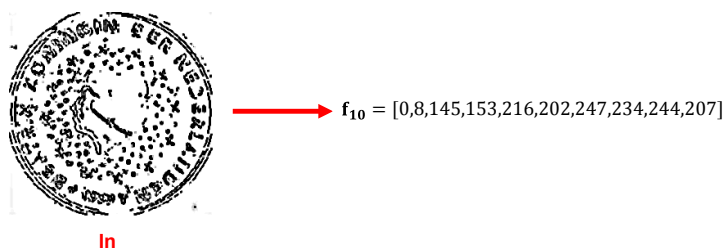
V tem koraku normaliziramo sliko na neko poljubno fiksno velikost  $J \times J$ , tako kot je prikazano na sliki 4.6, v katerem izrezano binarno sliko  $I_r$  pomanjšamo in pri tem dobimo novo binarno sliko  $I_n$ , velikosti  $J \times J$ . Pri implementaciji smo uporabili velikost  $J = 129$ , kot je bila podana v članku [1].



Slika 4.6: Prikaz normalizacije slike na velikost  $J \times J$ .

### 4.2.6 Spiralna dekompozicija

Na sliki 4.7, je prikazano delovanje tega koraka, ki vzame za vhod normalizirano binarno sliko  $I_n$  ter jo razdeli na enakomerno široke kolobarje, v našem primeru na 10 kolobarjev. Za vsak tak kolobar dobi vsoto slikovnih elementov, ki jih vsebuje, in iz teh vsot sestavi naš spiralni vektor  $f_{10}$ . V implementaciji smo računali spiralni vektor  $f$  za od 10 do 100 kolobarjev v razmiku 10.



Slika 4.7: Prikaz pridobitve spiralnega vektorja pri 10 kolobarjih, v katerih je vsaka vrednost vsota slikovnih elementov v posamičnem kolobarju.



### 4.2.7 Iskanje podobnosti

V zadnjem koraku pa spiralni vektor  $f$  primerjamo z ostalimi spiralnimi vektorji iz učne množice in pri tem dobimo pripadajoči razred. Pri implementaciji smo uporabili način, ki je podrobno opisan v podglavju 3.4.6.

## 4.3 Zajem testnih podatkov

Za potrebe testiranja naše metode uporabljamo dve različni podatkovni bazi slik kovancev. Prva je Muscle podatkovna baza kovancev, ki smo jo pridobili iz Musclove spletne strani[15] in je last avstrijske Dunajske tehnične univerze. To podatkovno bazo so uporabili za tekmovanja, tako so temu primerno na nekaterih slikah kovanci slabše prepoznavni. V tej podatkovni bazi je 207 učnih slik ter 1100 testnih slik, velikosti 640 x 576, vse so že spremenjene v sivinske slike (glej sliko 4.8 a). Kovanci na slikah pa so evropski kovanci pred uvedbo evra iz 15 držav, in sicer Avstrije, Belgije, Češke, Francije, Grčije, Hrvaške, Italije, Irske, Nizozemske, Slovenije, Španije, Švice in Velike Britanije.



Slika 4.8: Slika levo (a) je iz Musclove podatkovne baze, desna slika (b) je iz Evro podatkovne baze.

Pri drugi podatkovni bazi slik smo predhodno zajeli slike s pomočjo kamere, ki je na sliki 4.1. Z njo smo zajeli 103 učnih in 1072 testnih slik kovancev, velikosti 640 x 480, in jih shranili v RGB barvnem prostoru (glej sliko 4.8 b). Kovanci na slikah so evro kovanci z nacionalno stranjo iz 13 držav Avstrije, Belgije, Finske, Francije, Grčije, Italije, Irske, Luksemburga, Nemčije, Nizozemske, Portugalske, Slovenije in Španije.

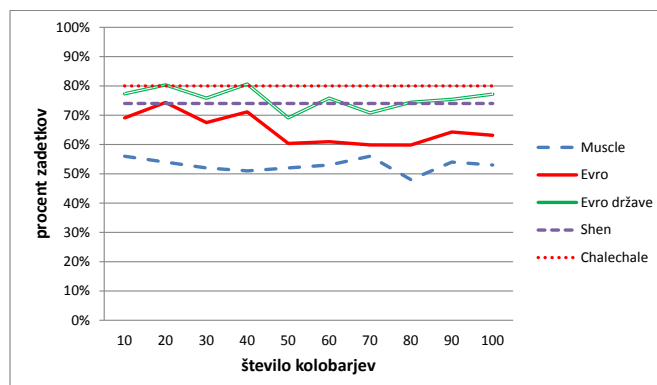
## 4.4 Rezultati testiranja

### 4.4.1 Postopek testiranja

Pred testiranjem smo vse slike iz obeh podatkovnih baz obdelali z našo metodo, pri čemer smo dobili pripadajoče spiralne vektorje, ki smo jih zapisali v datoteke. Datoteke so bile ločene glede na bazo slik, tip množice - ali gre za učno ali testno množico - ter na število kolobarjev. Tako je imela datoteka v vsaki vrstici zapisane spiralne vektorje vseh slik, ki so bili enake dolžine in so pripadali enemu tipu množice in bazi. V datoteki, v kateri so bili zapisani spiralni vektorji dolžine 10, smo še dodatno zapisali pripadajoči razred te slike. Vsako testiranje se je začelo z ustvarjanjem matrike z vsemi spiralnimi vektorji učne množice ter njihovimi pripadajočimi razredi, ki smo jih dobili z branjem prej ustvarjenih datotek. Nadaljevanje testiranja je bilo odvisno od tega, kaj smo hoteli testirati in katere testne spiralne vektorje smo pri tem potrebovali, da smo lahko z njimi primerjali učno množico ter pri tem dokazali uspešnost naših napovedi.

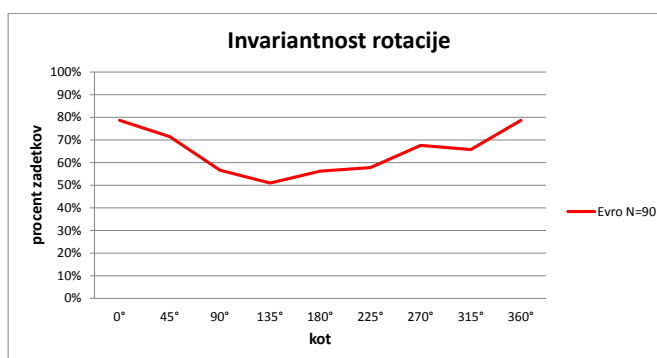
### 4.4.2 Grafi

Na grafu 4.9 vidimo prikaz uspešnosti detekcije kovancev pri različnem številu kolobarjev. Rdeča črta pripada bazi slik evrov, ki smo jih zajeli z našo kamero, modra črtkana črta pripada Musclovi bazi slik, zelena črta pripada prav tako naši bazi, vendar prikazuje uspešnost glede na zadetek države. Najprej opazimo, da nobeno testiranje ni bilo tako zelo uspešno kot v Chalechalevem članku [1], ki je na grafu označeno z rdečo pikčasto črto, kar smo tudi nekako pričakovali, glede na to da je uporabil za testne slike iste slike, kot so bile učne, vendar rotirane. Pri obeh bazah smo bili v več kot polovici primerov uspešni. Slabšo uspešnost Musclove baze od naše bi lahko pripisali še enkrat večji učni množici ter slabše vidnim slikam kovancev, vendar je rezultat lahko primerljiv rezultatom iz Shenovega članka [10], ki je tudi uporabil Musclove bazo. Njegov rezultat je na grafu prikazan z vijolično črtkano črto. Višja uspešnost na naši bazi glede na zadetek države pa potrjuje prisotnost podvojenih podob na nekaterih posamičnih množicah kovancev neke države.



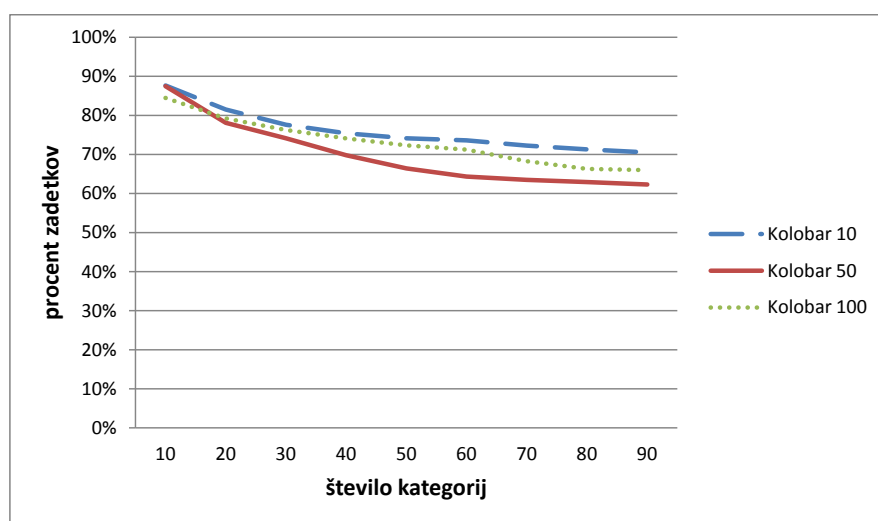
Slika 4.9: Prikaz uspešnosti metod pri detekciji.

Graf 4.10 prikazuje invariantnost metode v odvisnosti od kota pri 90 kolo-  
barjih. Kot  $45^\circ$  pomeni, da so vse testne slike premaknjene za kot  $45^\circ$  glede  
na pozicijo učnih slik, ki so seveda vse na isti poziciji. Iz grafa je možno raz-  
brati, da ima kot vpliv na uspešnost detekcije, nekje pade uspešnost celo za  
dobrih 20 %. Vendar kljub temu lahko še trdimo, da je metoda invariantna na  
različne rotacije kovancev. Minimum pri kotu  $135^\circ$  in ne pri kotu  $180^\circ$ , kot bi  
pričakovali, pa gre pripisati šumnim podatkom.



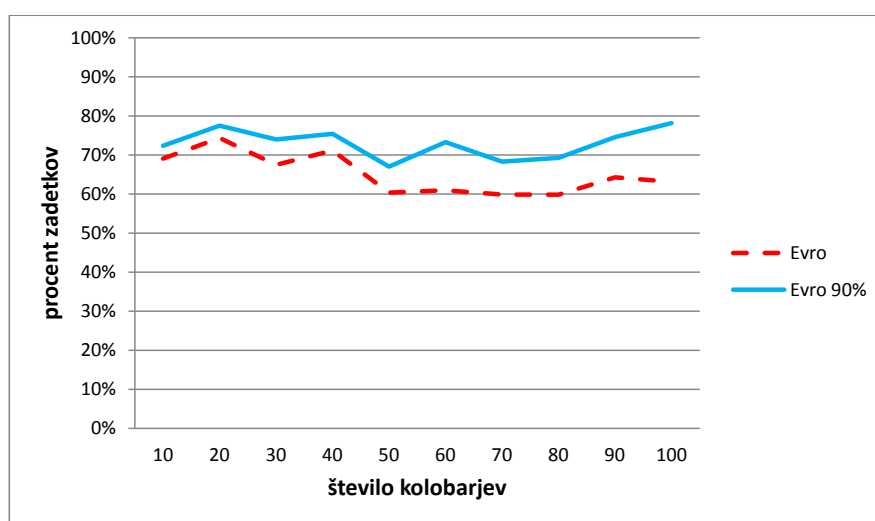
Slika 4.10: Prikaz invariantnosti rotacije.

V grafu 4.11 je prikazana uspešnost metode v odvisnosti števila kategorij pri številu kolobarjev 10, 50 in 100. Iz slike lahko razberemo, da imamo pri 10 učnih slikah zelo visoko uspešnost, okoli 90 %. Ob povečevanju baze učnih slik pa začne graf počasi padati, vendar se kmalu stabilizira, kar je posledica tega, da se z večanjem števila kategorij kovancev zmanjšuje razlika oziroma razdalja med njimi, s tem pa povečuje verjetnost napak pri detekciji.



Slika 4.11: Prikaz vpliva velikosti učne množice.

Graf 4.12 prikazuje vpliv morebitnih izboljšav metode. Ena od izboljšav bi lahko bila predpostavka, da se prava informacija skriva v kovancu. Zunanji del kovanca naj bi vseboval nepotrebne informacije, kot so razni napisi držav in letnice, ki s seboj prinašajo ogromno število slikovnih elementov, ti pa imajo lahko pri računanju razdalje usoden vpliv. Vendar se je v naših testiranjih, ko smo vzeli notranjih 90 % kolobarjev in jih 10 % zavrgli, izkazalo, da smo pri tem za malenkost izboljšali svoj rezultat.



Slika 4.12: Prikaz vpliva izboljšav metode.

## Poglavje 5

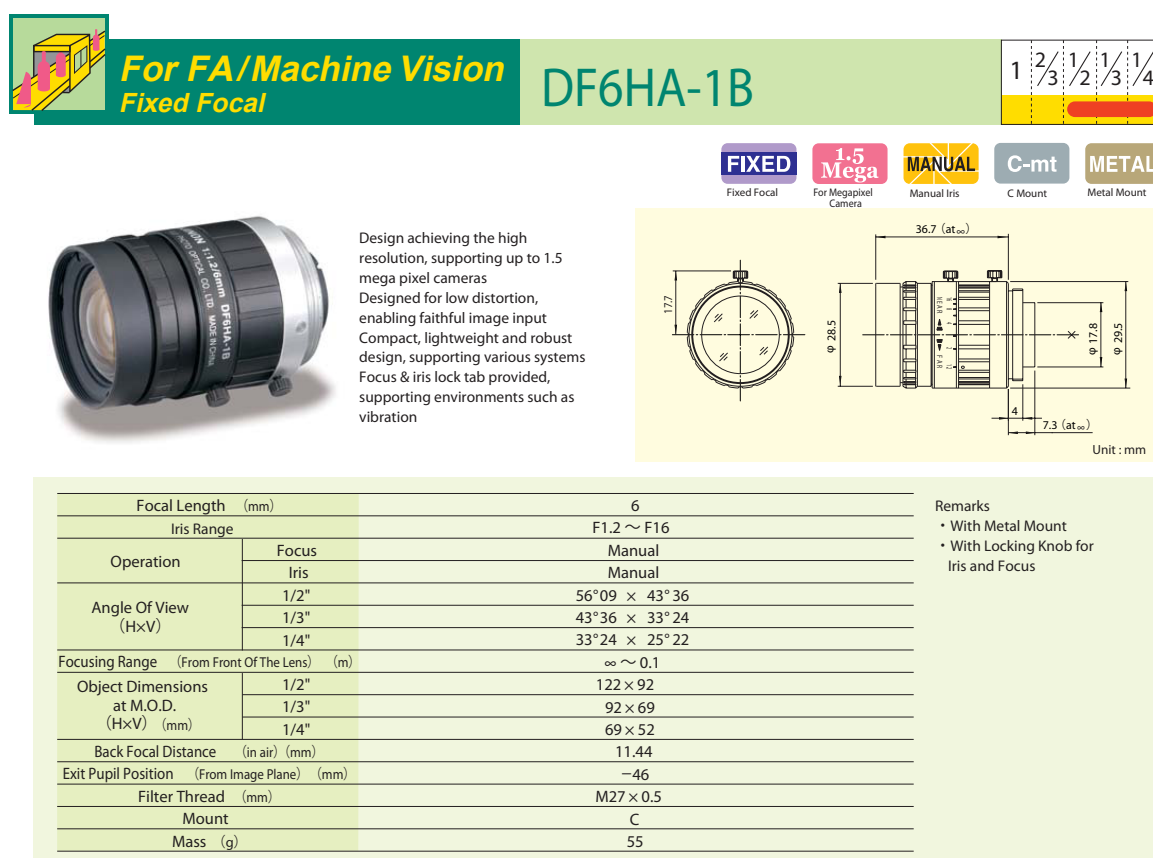
# Sklepne ugotovitve

V diplomski nalogi smo pregledali obstoječe metode s področja vizualne detekcije kovancev, ki bi se lahko v bodoče implementirale v komercialne avtomate kot dodatni mehanizem za preverjanje pravih kovancev. Med njimi smo izbrali najprimernejšo metodo oziroma pristop, za katero smo tudi naredili svojo implementacijo. Metodo smo testirali na Musclovi bazi in na naši bazi slik, ki smo jo zajeli z digitalno kamero. Poleg tega smo testirali vpliv posamične rotacije kovancev na uspešnost pravilne detekcije kovancev in pri tem ugotovili, da je metoda invariantna na rotacijo. Naslednji test je bil namenjen preverjanju vpliva povečevanja kategorij. Ugotovili smo naslednje: če se na začetku povečevanja začne manjši upad uspešnosti detekcije, se ta začne kmalu tudi stabilizirati. Zadnji je bil opravljen test vpliva morebitnih izboljšav, kot je na primer izpustitev 10 % zunanjega dela kovanca, kjer je po navadi le besedilo. S tem posegom smo uspeli za malenkost izboljšati rezultate. Med samimi testiranjem smo imeli velike težave s pravilnim izločanjem kovancev iz slike, saj metoda ne dopušča prevelikih napak. Drugi problem je bil iskanje prave mere za določitev robnih točk, premalo robnih točk je namreč povzročilo veliko napak pri detekciji.

Metoda se je obnesla solidno, vendar je še veliko prostora za nadaljnje raziskave. Med njimi je prav gotovo pospešitev preverjanja, kar bi lahko storili že z vnaprej znanimi radii, s čimer bi tudi odpravili problem izločanja kovancev iz slike. Druga izboljšava bi lahko bila grupiranje kategorij glede na nominalno vrednost kovanca. Ugotavljanje skupine, v katero sodi vhodni kovanec, bi lahko določili s pomočjo radija. S pomočjo teh izboljšav in še s katerimi drugimi, bi bila morda metoda zrela za uporabo v komercialnih avtomatih.

# Dodatek A

## Dodatek



Slika A.1: Specifikacije objektiva kamere, ki smo jo uporabljali pri zajemu slike kovancev [17].

# Flea<sup>®</sup> 2

## ULTRA-COMPACT + VERSATILE + 1394B

- 12 different models, 0.3 MP to 5.0 MP
- Smallest 1394b camera in the world
- High speed 1394b 800 Mb/s digital interface
- Metal case with locking screw connection
- Ideal for industrial machine vision

With resolutions ranging from 0.3MP (VGA) to 5.0MP and 12 different models to choose from, the compact, versatile Flea<sup>®</sup> camera system is a complete, cost effective and reliable IEEE-1394b solution for demanding image applications such as semiconductor inspection and high-speed assembly.



Specification	FL2-03S2M/C	FL2-08S2M/C	FL2-14S3M/C	FL2-20S4M/C	FL2G-13S2M/C	FL2G-50S5M/C
Image Sensor Type	Sony progressive scan interline transfer CCD's with square pixels and global shutter, monochrome or color					
Image Sensor Model	Sony ICX424 1/3"	Sony ICX204 1/3"	Sony ICX267 1/2"	Sony ICX274 1/1.8"	ICX445 1/3" EXview HAD CCD	ICX655 2/3 SuperHAD CCD
Maximum Resolution	648x488	1032x776	1392x1032	1624x1224	1288x964	2448x2048
Pixel Size	7.4 x 7.4µm	4.65 x 4.65µm	4.65 x 4.65µm	4.4 x 4.4µm	3.75 x 3.75µm	3.45 x 3.45µm
Analog-to-Digital Converter	Analog Devices 12-bit ADC					
Video Data Output	8, 12, 16 and 24-bit digital data					
Image Data Formats	Y8, Y16 (all models), RGB, YUV411, YUV422, YUV444, 8-bit and 16-bit raw Bayer data (color models)					
Color Processing	On-camera in YUV or RGB format, or on-PC in Raw format					
Digital Interface	Bilingual 9-pin IEEE-1394b for camera control, video data transmission, and power					
Transfer Rates	100, 200, 400, 800 Mbit/s					
Maximum Frame Rate	648x488 at 80 FPS	1032x776 at 30 FPS	1392x1032 at 15 FPS	1624x1224 at 15 FPS	1288x964 at 30 FPS	2448x2048 at 7.5 FPS
Partial Image Modes	pixel binning and region of interest modes via Format_7					
White Balance	automatic / manual / one-push modes, programmable via software					
General Purpose I/O Ports	8-pin Hirose HR25 GPIO connector opto-isolated pins for trigger and strobe (FL2G models only), bi-directional pins for trigger, strobe or serial port					
Gain Control	automatic / manual / one-push gain modes, programmable via software, 0dB to 24dB in 0.04dB increments					
Shutter Speed	automatic / manual / one-push modes, programmable via software, 0.02ms to greater than 10s (extended shutter mode)					
Gamma/LUT	0.50 to 4.00 / programmable lookup table					
Synchronization	via external trigger, software trigger (on same bus only), or free-running					
External Trigger Modes	DCAM v1.31 Trigger Modes 0, 1, 3, 4 and 5 (multiple exposure, 03S2 and 08S2 models only), 14 (overlapped trigger), and 15 (multi-shot trigger)					
Power Consumption	power via Vext GPIO pin or 9-pin 1394b interface: 8 to 30 V, less than 2.5 W					
Dimensions (L x W x H)	29mm x 29mm x 30mm (excluding lens holder, without optics)					
Mass	58g (without optics)					
Memory Storage	(FL2G models only) 32MB frame buffer, 512KB non-volatile data flash					
Memory Channels	3 memory channels for custom camera settings					
Camera Specification	IIDC 1394-based Digital Camera Specification v1.31, compatible with IEEE-1394b and IEEE-1394a interfaces					
Lens Mount	C-mount					
Emissions Compliance	Complies with CE rules and Part 15 Class B of FCC Rules					
Operating Temperature	0° to 45°C					
Storage Temperature	-30° to 60°C					
Warranty	2 year					

Slika A.2: Specifikacije kamere, ki smo jo uporabljali pri zajemu slike kovancev [17].



# Slike

1.1	Prvi prodajni avtomat . . . . .	3
1.2	Podobni kovanci . . . . .	4
2.1	Struktura različnih metod za detekcijo kovancev . . . . .	8
2.2	Canny filter . . . . .	9
2.3	Hough transformacija . . . . .	9
2.4	Segmentacija . . . . .	10
2.5	Sift metoda . . . . .	11
2.6	Interpolacija slike po metodi najbližjega sosedu . . . . .	11
2.7	Razdelitev kovanca na kolobarje . . . . .	12
2.8	Evklidska razdalja . . . . .	13
2.9	City-block razdalja . . . . .	13
2.10	Križna korelacija . . . . .	14
3.1	Struktura metode . . . . .	17
3.2	Krožnica v koordinatnem sistemu . . . . .	21
3.3	Razdelitev kovanca na kolobarje . . . . .	22
4.1	Kamera s stojalom . . . . .	24
4.2	Sprememba slike v sivinsko sliko . . . . .	25
4.3	Transformacija slike v binarni sliki . . . . .	26
4.4	Iskanje kovanca v sliki . . . . .	27
4.5	Rezanje kovanca iz slike . . . . .	27
4.6	Normalizacija slike . . . . .	28
4.7	Spiralni vektor . . . . .	28
4.8	Kovanca iz podatkovnih baz . . . . .	29
4.9	Graf metod . . . . .	31
4.10	Graf invariantnosti rotacije . . . . .	31
4.11	Graf vpliva velikosti učne množice . . . . .	32
4.12	Graf vpliva izboljšav . . . . .	33

A.1	Specifikacije objektiva kamere . . . . .	35
A.2	Specifikacije kamere za zajem slike . . . . .	36

# Tabele

# Literatura

- [1] A. Chalechale, “Coin recognition using image abstraction and spiral decomposition,” v zborniku *9th Intl. Symposium on Signal Processing and Its Applications*, Sharjah, United Arab Emirate, feb. 2007, str. 1-4.
- [2] C.-M. Chen, S.-Q. Zhang, Y.-F. Chen, “A coin recognition system with rotation invariance,” v zborniku *Intl. Conference on Machine Vision and Human-Machine Interface*, Kaifeng, China, apr. 2010, str. 755-757.
- [3] H. Chen, “Chinese coin recognition based on unwrapped image and rotation invariant template matching,” v zborniku *Third intl. Conference on Intelligent Networks and Intelligent Systems*, Shenyang, China, nov. 2010, str. 5-7.
- [4] D. A. Forsyth and J. Ponce, *Computer Vision: A modern approach*, Prentice Hall, 2003.
- [5] M. Fukumi, S. Omatu, F. Tekada, T. Koska, “Rotation-invariant neural pattern recognition system with application to coin recognition”, v zborniku *IEEE Trans. on Neural Networks*, Beijing, China, nov. 1992, vol.3, str. 272-278.
- [6] V. Gupta, R. Puri and M. Verma, “Prompt Indian coin recognition with rotation invariance using image subtraction technique,” v zborniku *Intl. Conference on Devices and Communications*, Ranchi, India, feb. 2011, str. 1-5.
- [7] D. G. Lowe, “Distinctive image features from scale invariant keypoints,” v reviji *The International Journal of Computer Vision*, Canada, jan. 2004, str. 60-87.
- [8] O. Martens, T. Saar, A. Gavrijaseva and A. Molder, “Variable resolution image processing for validation of coins”, v zborniku *IEEE 7th Intl. Sym-*

- posium on Intelligent Signal Processing*, Floriana, Malta, sep. 2011, str. 1-4.
- [9] M. Roushdy, "Detecting coins with different radii based on Hough transform in noisy and deformed image", v zborniku *GVIP Journal*, Cairo, Egypt, apr. 2007, vol.7, str. 25-29.
- [10] L. Shen, S. Jia, Z. Ji, W.-S. Chen, "Statistics of Gabor features for coin recognition", v zborniku *International Workshop on Imaging Systems and Techniques*, Shenzhen, China, maj 2009, str. 295-298.
- [11] P. Thumwarin, S. Malila, P. Janthawong, W. Pibulwej and T. Matsuura, "A robust coin recognition method with rotation invariance" v zborniku *Intl. Conference on Communications, Circuits and Systems Proceedings*, Guilin, China, jun. 2006, str. 520-523.
- [12] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.
- [13] (2011) Detecting coins using Hough transform Dostopno na:  
<http://www.math.tau.ac.il/~turkel/notes/hough4.html>
- [14] (2004) Distance metrics overview Dostopno na:  
<http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering-Parameters-Distance-Metrics-Overview.htm>
- [15] (2012) MUSCLE data and software repository Dostopno na:  
[https://muscle.caa.tuwien.ac.at/download\\_upload.php](https://muscle.caa.tuwien.ac.at/download_upload.php)
- [16] (2007) Nearest neighbor image scaling Dostopno na:  
[http://tech-algorithm.com/articles/nearest\\_neighbor\\_image\\_scaling.htm](http://tech-algorithm.com/articles/nearest_neighbor_image_scaling.htm)
- [17] (2012) Point Grey Dostopno na:  
<http://www.ptgrey.com>
- [18] (2011) Vending machine Dostopno na:  
[http://www.underthepier.com/19\\_vending\\_machines.htm](http://www.underthepier.com/19_vending_machines.htm)
- [19] (2012) Vending machine Dostopno na:  
[http://en.wikipedia.org/wiki/Vending\\_machine.htm](http://en.wikipedia.org/wiki/Vending_machine.htm)