

Univerza v Ljubljani  
Fakulteta za elektrotehniko

Tilen Mlakar

**ANALIZA VSEBINE SLIK S TENZORSKIM  
GLASOVANJEM**

MAGISTRSKO DELO

Mentor: doc. dr. Andrej Košir

Ljubljana, 2008



## Zahvala

Na prvem mestu se zahvaljujem mentorju doc. dr. Andreju Koširju, ki me je z nasveti, idejami in pomočjo vodil pri izdelavi magistrskega dela.

Posebej bi se rad zahvalil še prof. dr. Juriju F. Tasiču, ki mi je omogočil zaposlitev v Laboratoriju za digitalno obdelavo signalov, slik in videa na Fakulteti za Elektrotehniko v Ljubljani. Za pomoč in nasvete se zahvaljujem tudi ostalim sodelavcem iz laboratorija.

Zahvaljujem se tudi Tjaši, ki me je med pisanjem magistrske naloge vzpodbujala in mi pomagala s koristnimi nasveti.

Hvala!



## Povzetek

Vidno zaznavanje okolja je najpomembnejši in najkompleksnejši postopek zaznavanja okolja pri človeku. Od vseh osnovnih čutil nam priskrbi največ informacij. Ker je količina informacij, ki jo lahko vsebuje slika ali na splošno vidna scena velika, lahko slike oziroma video posnetke izkoristimo kot učinkovit medij za prenos informacije. Zaradi razmaha množičnih medijev in vse večje količine slik ter informacij, ki jih nosijo, se je pojavila potreba po avtomatski analizi in opisu vidne scene, ki je zajeta na sliki. V magistrskem delu predstavljamo metodo za analizo in opis vidne scene z vloženim grafom, kjer krivulje objektov, zapisane s kontrolnimi točkami B-zlepkov, predstavimo s povezavami v grafu, presečišča in končne točke krivulj pa predstavimo z vozlišči grafa. Z vložitvijo grafa določimo položaj grafa v ravnini. Vložen graf lahko uporabimo kot vhod v grafovski algoritem za iskanje poti in kot vhod v algoritem za razpoznavo objektov iz kontrolnih točk B-zlepkov. Tako lahko objekte v sliki razpoznamo ter določimo njihove geometrijske in topološke lastnosti. Segmentacijo slike ter določanje krivulj in presečišč izvedemo z metodo tenzorskega glasovanja.

B-zlepki so najbolj optimalen parametrični zapis krivulj s stališča minimalne podpore pri maksimalni gladkosti. Lastnosti B-zlepkov so prostorska edinstvenost, omejenost, nepretrganost, lokalno vplivanje na obliko in neodvisnost od afinih transformacij. Zaradi teh lastnosti se B-zlepke uporablja v računalniško podprtem načrtovanju in računalniški grafiki, uporabimo pa jih lahko tudi pri razpoznavi objektov. Poleg tega lahko s predstavitvijo krivulj s kontrolnimi točkami B-zlepkov krivulje predstavimo z malo točkami na zgoščen način.

Tenzorsko glasovanje je ogrodje za segmentacijo slik, ki je prilagodljivo in uporabno na več področjih računalniškega vida. Prednosti pred ostalimi segmentacijskimi metodami so, da usmeri vhodne podatke, učinkovito odpravlja šum in diskontinuitete v slikah, je neiterativen in ima samo en prost parameter, to je velikost glasovalnih polj. Metoda tenzorskega glasovanja sloni na dveh

elementih. Prvi je predstavitev vhodnih podatkov (v našem primeru slikovnih elementov v sliki) s simetričnimi tenzorji 2. reda. Drugi element je nelinearno filtriranje s tenzorskimi glasovalnimi polji. Preko teh polj si tenzorji izmenjujejo informacije o tipu strukture (krivulja, presečišče krivulj), ki ji določen tenzor pripada. S tenzorskim glasovanjem lahko v sliki uspešno zaznamo strukture kot so krivulje, presečišča in končne točke krivulj. Presečišča in končne točke krivulj predstavimo z vozlišči grafa, krivulje pa zapišemo s kontrolnimi točkami B-zlepkov ter jih predstavimo s povezavami grafa.

V okviru magistrske naloge smo implementirali metodo tenzorskega glasovanja, metodo za prileganje B-zlepkov ter metodo za izgradnjo grafa. Pri ovrednotenju predstavljenih metod smo se osredotočili na testiranje metode tenzorskega glasovanja, ki smo ga izvajali na testnih slikah. Ugotovili smo, da je metoda odporna na šum, poleg tega pa so krivulje, ki jih dobimo kot rezultat tenzorskega glasovanja, primerne za neposredno prileganje B-zlepkov. Nadaljnje delo bi bilo upravičeno na dveh področjih. Prvo je izboljšanje implementacije tenzorskega glasovanja, drugo pa razpoznava objektov ter njihovih geometričnih in topoloških lastnosti.

**Ključne besede:** analiza vsebine slik, opis vidne scene, razpoznava objektov, tenzorsko glasovanje, B-zlepki, vložen graf, segmentacija slik.

# Abstract

Visual perception is one of the most important and complex processes of human perception. It provides us with more information about the environment than any other basic human sense. Because of that an image or a video clip of a visual scene has a huge informational value. Therefore it can be used as an efficient medium for information transmission. Because of the proliferation of communication media and a huge amount of images that carry information, the need for automatic image analysis and visual scene description has arisen. This master thesis presents an approach for image analysis and scene description with the use of embedded graphs. Object contours or curves, which are represented by B-splines, can be presented with graph edges, while junctions and intersections can be presented by graph vertices. Object position can be encoded by graph embedding. An embedded graph could be later used for obtaining geometrical and topological information, while B-splines could be used for object recognition. For image segmentation and curves extraction a tensor voting is proposed.

A curve can be represented with B-spline control points in a compressed way. B-splines are regarded as one of the most efficient curve representation and possess very attractive properties such as spatial uniqueness, boundedness and continuity, local shape controllability and invariance to affine transformations. They have been extensively used in computer-aided design and computer graphics and can be also used for recognition purposes.

Tensor voting is an image segmentation framework, which can be adopted and used in many computer vision applications. Main advantages of tensor voting, when compared to other segmentation techniques, are: orientation of input data, robustness to noise and discontinuities in input data, the fact that it is non-iterative and has only one free parameter, the size of voting fields. Tensor voting is founded on two components: symmetric second order tensor calculus for data representation and non-linear voting for data communication. Tensor

voting generates descriptions in terms of curves, junctions and curve endpoints from noisy images. Junctions and curve endpoints are later presented by graph vertices, while curves are represented by B-splines control points and presented by graph edges.

A tensor voting algorithm, B-splines control points estimation algorithm and graph construction algorithm have been implemented in this thesis. We focused on evaluation of tensor voting segmentation, which was performed on a set of test images. Tensor voting proved to be robust to noise. Besides that, curves which were extracted by tensor voting could be directly used in a B-splines control points estimation algorithm. Further research would be interesting in two directions; the first one is the search for improvements of tensor voting implementation and the second one is extraction of objects geometric and topological information and objects recognition.

**Keywords:** image content analysis, visual scene description, object recognition, tensor voting, B-splines, embedded graph, image segmentation.



# Vsebina

<b>Seznam slik</b>	<b>xiii</b>
<b>Seznam tabel</b>	<b>xvii</b>
<b>1. Uvod</b>	<b>1</b>
<b>2. Analiza vidne scene na sliki</b>	<b>5</b>
2.1 Opis analize vizualne scene . . . . .	5
2.2 Predlagana rešitev . . . . .	6
2.3 Opis implementacije . . . . .	7
2.4 Nosilna literatura . . . . .	8
<b>3. Predlagana rešitev</b>	<b>9</b>
3.1 Vpeljava oznak in pojmov . . . . .	9
3.2 Potek predlagane rešitve . . . . .	15
3.3 Način ovrednotenja rezultatov . . . . .	17
<b>4. Obstoječe stanje na področju analize vidne scene</b>	<b>19</b>
4.1 Primeri uporabe analize vidne scene v realnem okolju . . . . .	19
4.2 Dosedanje delo na področju analize vidne scene . . . . .	20
4.2.1 Raziskovalno delo . . . . .	20
4.2.2 Obstoječi izdelki . . . . .	22
4.3 Parametrični zapis krivulj . . . . .	26
	ix

4.3.1	Hermitske krivulje . . . . .	27
4.3.2	Bezierove krivulje . . . . .	28
<b>5.</b>	<b>Analiza vidne scene</b>	<b>31</b>
5.1	Segmentacija s tenzorskim glasovanjem . . . . .	31
5.1.1	Uvod . . . . .	31
5.1.2	Zapis vhodnih podatkov . . . . .	34
5.1.3	Glasovalna polja . . . . .	38
5.1.4	Tenzorsko glasovanje . . . . .	41
5.1.5	Analiza glasov . . . . .	45
5.2	Določanje posameznih struktur . . . . .	46
5.2.1	Iskanje presečišč . . . . .	47
5.2.2	Sledenje krivuljam . . . . .	50
5.3	Zapis z B-zlepki . . . . .	53
5.3.1	Predstavitev B-zlepkov . . . . .	53
5.3.2	Uniformni kubični B-zlepki . . . . .	54
5.3.3	Ocenjevanje kontrolnih točk B-zlepkov iz podatkovnih točk	56
5.4	Predstavitev scene z vloženim grafom . . . . .	57
5.4.1	Grafi . . . . .	58
5.4.2	Opis scene z vloženim grafom . . . . .	60
<b>6.</b>	<b>Eksperimentalni rezultati</b>	<b>63</b>
6.1	Testni podatki . . . . .	63
6.2	Testno okolje . . . . .	65
6.3	Opis testiranja in rezultati . . . . .	65
6.3.1	Testiranje na umetno pridobljeni sliki . . . . .	68
6.3.2	Testiranje na realni sliki . . . . .	73
6.3.3	Prikaz opisa slike z vloženim grafom . . . . .	77

---

<b>7. Zaključek in nadaljnje delo</b>	<b>79</b>
7.1 Sklep . . . . .	79
7.2 Nadaljnje delo . . . . .	81
<b>Literatura</b>	<b>83</b>



## Seznam slik

1.1	Naloge sistemov za računalniški vid. . . . .	2
2.1	Predlagana rešitev razpoznavе objektov na vidni sceni. . . . .	7
3.1	Primer točke na krivulji ter tangente na krivuljo v tej točki. . . . .	10
3.2	Električno polje naelektrenega delca. . . . .	11
3.3	Primer diagrama grafa. . . . .	14
3.4	Primer B-zlepka. . . . .	15
3.5	Segmentacija s tenzorskim glasovanjem. . . . .	16
3.6	Analiza tenzorskih polj in izgradnja grafa. . . . .	17
3.7	Zapis krivulj z B-zlepki. . . . .	17
4.1	Uporabniški vmesnik programskega paketa Aimetis AIRA v5. . . . .	23
4.2	Sistem Hitachi ITS 21. . . . .	24
4.3	Uporabniški vmesnik sistema IDENTRACE in navidezna področja. . . . .	25
4.4	Program imgSeek za iskanje po slikah. . . . .	25
4.5	Spletna storitev Behold za iskanje po slikah. . . . .	26
4.6	Krivulja, določena z začetno in končno točko in odvodoma v teh točkah. . . . .	27
4.7	Krivulja, določena s štirimi točkami. . . . .	28
4.8	Zlepek treh Bezierovih krivulj. . . . .	29

5.1	Gestalt zakoni: a.) objekte, ki so si podobni, združimo v stolpce, b.) objekte, ki so si blizu, združimo v skupino, c.) črtam sledimo po najbolj gladki poti, d.) združene objekte obravnavamo kot celoto, e.) kompleksne objekte dojemamo čimbolj preprosto - vidimo 5 krogov namesto več bolj zapletenih objektov. . . . .	32
5.2	Potek tenzorskega glasovanja. . . . .	35
5.3	Elipsa in njen lastni sistem. . . . .	37
5.4	Razstavljanje poljubnega tenzorja na paličasti in okrogli tenzor. . . . .	38
5.5	Konstrukcija glasovalnega polja 2. reda. . . . .	39
5.6	Konstrukcija glasovalnega polja 2. reda. . . . .	40
5.7	Velikosti tenzorjev paličastega glasovalnega polja v 2-D. . . . .	41
5.8	Orientacije tenzorjev paličastega glasovalnega polja v 2-D. . . . .	41
5.9	Velikosti tenzorjev okroglega glasovalnega polja v 2-D. . . . .	42
5.10	Orientacije tenzorjev okroglega glasovalnega polja v 2-D. . . . .	42
5.11	Neorientirani žetoni. . . . .	43
5.12	Prerez izrazitosti žetonov. . . . .	44
5.13	Smer vektorjev v glasovalnem polju prvega reda. . . . .	44
5.14	Prerez smernosti žetonov. . . . .	45
5.15	Testna slika za prikaz določanja presečišč in krivulj. . . . .	47
5.16	Paličasto tenzorsko polje, predstavljeno z vektorji. . . . .	47
5.17	Povečan izsek slike 5.16. . . . .	48
5.18	Okroglo tenzorsko polje, predstavljeno z vektorji. . . . .	48
5.19	Polje smernih vektorjev. . . . .	49
5.20	Vrednosti $\lambda_2$ po tenzorskem glasovanju in dekompoziciji. . . . .	49
5.21	Absolutne vrednosti smernih vektorjev po glasovanju 1. reda. . . . .	49
5.22	Vrednosti $\lambda_1 - \lambda_2$ . . . . .	50
5.23	Izrez vektorskega polja, ki prikazuje različne smeri vektorjev. . . . .	51
5.24	Točke, ki pripadajo krivuljam. . . . .	52

5.25	Krivulje, presečišča in končne točke. . . . .	52
5.26	Bazna funkcija $Q_{0,4}(t)$ . . . . .	55
5.27	Kubične bazne funkcije . . . . .	56
5.28	Kontrolne točke in B-zlepki testne slike. . . . .	58
5.29	Diagram grafa testne slike. . . . .	60
6.1	Postopek pridobivanja testne slike. . . . .	64
6.2	Slika letala, pridobljena z digitalnim fotoaparatom. . . . .	64
6.3	Testno okolje. . . . .	66
6.4	Primerjava paličastih glasovalnih polj velikosti 30 in 60. . . . .	66
6.5	Razlika ploščin, ki jih oklepajo določene krivulje kot mera napake. . . . .	67
6.6	Rezultati tenzorskega glasovanja in določanja krivulj na sliki brez dodanega šuma. . . . .	68
6.7	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.01$ . . . . .	69
6.8	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.05$ . . . . .	69
6.9	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.1$ . . . . .	69
6.10	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.15$ . . . . .	69
6.11	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.2$ . . . . .	70
6.12	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.25$ . . . . .	70
6.13	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.3$ . . . . .	70
6.14	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.35$ . . . . .	70

6.15	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.4$ . . . . .	71
6.16	Velikost glasovalnih polj in čas glasovanja za določanje presečišč v odvisnosti od variance. . . . .	72
6.17	Velikost glasovalnih polj in čas glasovanja za določanje presečišč v odvisnosti od variance. . . . .	72
6.18	Velikost glasovalnih polj in čas glasovanja za določanje presečišč v odvisnosti od variance. . . . .	73
6.19	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki brez dodanega šuma. . . . .	74
6.20	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.01$ . . . . .	74
6.21	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.02$ . . . . .	74
6.22	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.03$ . . . . .	75
6.23	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.04$ . . . . .	75
6.24	Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom $\sigma^2 = 0.05$ . . . . .	75
6.25	Velikost glasovalnih polj in čas glasovanja za določanje presečišč v odvisnosti od variance. . . . .	76
6.26	Velikost glasovalnih polj in čas glasovanja za določanje krivulj v odvisnosti od variance. . . . .	76
6.27	Relativna napaka določenih krivulj in čas določanja krivulj v odvisnosti od variance. . . . .	77
6.28	Prikaz grafa, ki ima povezave labelirane s koeficienti B-zlepkov. . . . .	78



## Seznam tabel

5.1	Zapis vhodnih struktur s simetričnim tenzorjem drugega reda. . .	37
5.2	Analiza glasov in lastnosti 2-D struktur. . . . .	46
6.1	Rezultati tenzorskega glasovanja in določanja krivulj. . . . .	71
6.2	Rezultati tenzorskega glasovanja in določanja krivulj. . . . .	75



# 1. Uvod

Vidno zaznavanje je eno izmed petih osnovnih človekovih čutil. Je najpomembnejši in najkompleksnejši postopek zaznavanja okolja. Priskrbi nam informacije, ki jih potrebujemo tako za enostavne naloge, kot so na primer razpoznavanje objektov, kot tudi za kompleksnejše naloge, kot so hoja, tek, sklepanje odločitev in tako dalje. O količini informacij, ki jih pridobimo z opazovanjem vizualne scene oziroma ki je zapisana v sliki ali v videu posnetku, priča kitajski pregovor: “Slika je vredna več kot tisoč besed” [1]. Slike oziroma video posnetke lahko zato s pridom izkoristimo za prenos informacij. Mediji, kot na primer časopisi, televizija in v zadnjem času Internet, uporabljajo slike kot nosilce informacij. To je vodilo do velikih količin informacij, ki so zapisane v slikah in do potrebe po analizi, obdelavi in shranjevanju teh informacij. Z razvojem digitalnih računalnikov je tako nastala nova veja znanosti, ki ji pravimo *digitalna obdelava slik*. To je obdelava signalov, kjer je vhod v obdelavo slika, izhod pa je lahko slika, nabor značilnik, vezanih na vsebino slike ali simbolični opis slike. Razvoj digitalne obdelave slik se je začel v 60-tih letih prejšnjega stoletja, od takrat pa je doživela veliko rast in hiter razvoj. Dandanes nas digitalna obdelava slik spremlja skoraj na vsakem koraku, pojavlja pa se v telekomunikacijah, v znanosti, oddajanju TV programov, v medicini, v grafičnem oblikovanju in tako dalje.

Človekovo zaznavanje okolja je zelo kompleksen in še precej neznan proces, vendar se kljub temu pri digitalni obdelavi slik skušamo zgledovati po človekovem zaznavanju. Zaradi te podobnosti s človeškim vidom temu področju pravimo tudi računalniški vid. Ena izmed razvrstitev računalniškega vida [2] deli to področje na tri sklope:

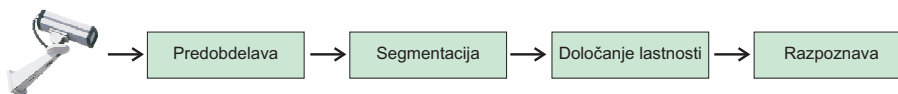
- nizko-nivojski,
- srednje-nivojski ter

- visoko-nivojski računalniški vid.

Z nizko-nivojskimi algoritmi ponavadi izboljšamo sliko, vhod in izhod za te algoritme pa so digitalne slike. S srednje-nivojskimi algoritmi sliko zapišemo z nizko-nivojskimi značilkami, kot so npr. konture ali maske objektov. Z visoko-nivojskimi algoritmi, ki poizkušajo oponašati človekovo zaznavanje, pa skušamo razpoznati vsebino slike. Ti algoritmi za vhod in izhod uporabljajo simbolične predstavitve ter so tesno povezani s področji umetne inteligence in razpoznavanja vzorcev. Arhitekture sistemov za računalniški vid se med seboj zelo razlikujejo glede na aplikacijo, v kateri se uporabljajo, vsem so pa značilne naslednje naloge:

- zajem slike,
- predobdelava,
- detekcija objektov ali segmentacija,
- določanje lastnosti in
- razpoznavna.

Če te naloge umestimo v prejšnjo delitev, potem v nizko-nivojski računalniški vid spadata zajem in predobdelava slik. Segmentacija in določanje lastnosti spada v srednje-nivojski, razpoznavna pa v visoko-nivojski računalniški vid.



Slika 1.1: Naloge sistemov za računalniški vid.

Slika 1.1 prikazuje korake sistemov za računalniški vid. Najprej moramo s slikovnim senzorjem sliko zajeti. Ponavadi so zajete slike podvržene raznim popačenjem, zato jih moramo izboljšati. Izboljšavo slik dosežemo s postopki predobdelave, ki jih v grobem delimo na postopke obnavljanja in na postopke izboljšanja kakovosti. V naslednjem koraku moramo sliko razčleniti ali segmentirati na manjše enote ali objekte in te ločiti od ozadja. Rezultat so samostojni objekti, ki so povezani med seboj. Nato moramo določiti lastnosti objektov in

zapisati relacije med njimi v taki obliki, da jih lahko uporabimo v zadnji fazi. Zadnja faza je razpoznavna, kjer lastnosti neznanih objektov primerjamo s prej shranjenimi lastnostmi poznanih objektov. Rezultat celotnega postopka so razpoznani objekti in relacije med njimi. Z omenjenimi postopki sliko analiziramo in rezultat zapišemo v obliki, ki je primerna za uporabo v določeni vrsti aplikacij.

V magistrskem delu bomo opisali le postopek segmentacije in določanje lastnosti ter opis regij objektov in relacij med njimi (vidne scene) na način, ki je primeren za visoko-nivojsko razpoznavo objektov [3]. Kompleksno sceno na sliki predstavimo z vloženim grafom, katerega vložene povezave potekajo po robovih regij objektov. Ta predstavitev je primerna za visoko-nivojsko analizo, ki jo v delu le nakazujemo.

Magistrsko delo je sestavljeno iz 7 poglavij. V drugem poglavju bomo opisali problem analize vidne scene in predlagali našo rešitev. Navedli bomo razloge za izbiro posameznih metod in programskih orodij. V tretjem poglavju bomo podrobneje predstavili našo rešitev ter podali opis oznak in pojmov. V četrtem poglavju bomo predstavili področja uporabe, opisali dosedanje delo na industrijskem in raziskovalnem področju ter omenili nekaj metod za analizo in opis vidne scene. Opisali bomo tudi metode za zapis krivulj. V petem poglavju bomo podrobno opisali segmentacijo s tenzorskim glasovanjem, zapis krivulj z B-zlepki ter zapis vidne scene z grafi. Potek testiranja, eksperimentalne rezultate in opis testnega okolja bomo podali v šestem poglavju. V sedmem poglavju bomo delo zaključili, podali sklepne ugotovitve in opisali možnosti za nadaljnje delo.



## 2. Analiza vidne scene na sliki

V tem poglavju bomo predstavili analizo vizualne scene na slikah. Opisali bomo postopke, ki so potrebni za opis scene ter predlagali ustrezne metode, ki jih bomo v nadaljevanju dela tudi opisali. Podali bomo način implementacije in razloge za izbrano implementacijo. Predstavili bomo tudi nosilno literaturo, po kateri smo se pri implementaciji zgledovali.

### 2.1 Opis analize vizualne scene

Kot smo omenili v 1. poglavju, se bomo v magistrski nalogi ukvarjali le s postopkom segmentacije in opisom vidne scene z vloženim grafom. Nad sliko, ki jo zanjamo s slikovnim senzorjem, moramo izvesti določeno zaporedje operacij. Sliko je najprej potrebno prenesti v računalnik, jo ustrezno obdelati, nato pa z uporabo razčlenjevanja ali segmentacije poiskati oziroma detektirati področja ali objekte, ki nas zanimajo. Segmentacija je postopek, v katerem sliko razčlenimo na regije, ki so sestavljene iz med seboj podobnih slikovnih elementov. Kriterijev za merjenje podobnosti slikovnih elementov je več, na njihovi izbiri temelji tudi izbira metode segmentacije. Cilj segmentacije je torej poenostaviti oziroma spremeniti zapis slike na tak način, da omogočimo nadaljnjo analizo slike. Po segmentaciji oziroma detekciji zanimivih področij ali objektov sledi proces določanja lastnosti ter njihov zapis za kasnejšo rabo pri razpoznavi objektov. Obenem želimo objekte zapisati tudi na tak način, da ohranimo medsebojne relacije. Naša naloga je bila določitev krivulj in presečišč, ki ustrezajo robovom objektov in njihov kompakten in učinkovit zapis.

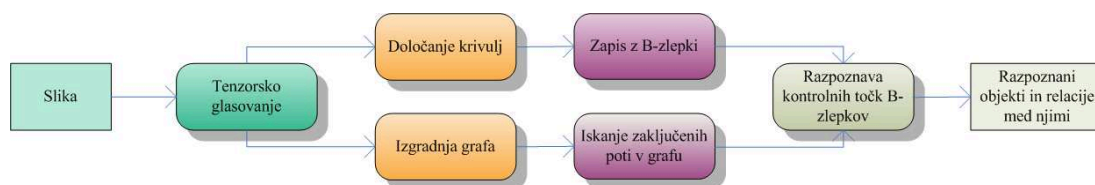
## 2.2 Predlagana rešitev

Sistem, ki bi se zavedal celotnega okolja, je trenutno še neizvedljiv, zato izbira metod za segmentacijo slik ter razpoznavo objektov temelji na izbranem področju uporabe. Kljub temu je bilo nekaj korakov v to smer že narejenih. Medtem, ko nekatere metode za segmentacijo temeljijo na izbranem področju uporabe, kjer se lahko kriterij za podobnost slikovnih elementov od metode do metode razlikuje (videz, oblika, gibanje,...), je uspelo Medioniju in drugim [4] narediti ogrodje za segmentacijo, ki je prilagodljivo in uporabno v več namenih. To ogrodje se imenuje tenzorsko glasovanje in se lahko uporablja v številnih aplikacijah, kot so na primer obnavljanje in restavriranje slik in videa, iskanje struktur v slikah in 3-D podatkih, iskanje sorodnih slikovnih točk pri stereo vidu ter segmentacija na podlagi gibanja v videu. Glavni prednosti tenzorskega glasovanja pred drugimi metodami segmentacije sta v tem, da usmeri vhodne podatke in učinkovito odpravlja šum in diskontinuitete v slikah. Zaradi usmerjanja vhodnih podatkov dobimo več informacij o tipu strukture, ki ji posamezen slikovni element pripada. Zaradi teh lastnosti smo tenzorsko glasovanje uporabili za segmentacijo slik pri opisu vidne scene. S tenzorskim glasovanjem sicer lahko uspešno zaznamo strukture, kot so krivulje, površine in presečišča ter izločimo šum, vendar moramo znati te strukture učinkovito zapisati za poznejšo razpoznavo. V našem primeru, kjer določamo le krivulje ter presečišča, lahko za zapis krivulj uporabimo kontrolne točke B-zlepkov. Na tak način lahko krivulje zapišemo z majhnim številom točk na zgoščen način. B-zlepki so se na področju opisa krivulj izkazali za najbolj optimalen zapis krivulj s stališča minimalne podpore pri maksimalni gladkosti [5]. B-zlepke med drugim uporabljamo v računalniško podprtem načrtovanju in računalniški grafiki, nekaj pozornosti pa jim je bilo namenjeno tudi pri razpoznavi objektov.

Prvi korak predlagane rešitve je, da s tenzorskim glasovanjem v sliki določimo krivulje in presečišča med njimi. Nato te krivulje zapišemo s kontrolnimi točkami B-zlepkov, odvisnosti med njimi pa z uporabo grafov, kjer presečišča med krivuljami in končne točke krivulj predstavljajo vozlišča, krivulje pa povezave. Tak graf vložimo, kar pomeni, da vsakemu vozlišču priredimo koordinate v slikovni ravnini, povezavam pa vektorje kontrolnih točk B-zlepkov ter s tem opišemo vidno sceno na sliki. Pri razpoznavi nato uporabimo grafovski algoritem [6], ki najde



vse zaključene poti po celotnem grafu. Vsako pot, ki je opisana s kontrolnimi točkami B-zlepkov, lahko nato primerjamo s kontrolnimi točkami shranjenih objektov. Za razpoznavo kontrolnih točk lahko uporabimo algoritem, opisan v [5], katerega rezultat je neodvisen od rotacije, skaliranja in premika objektov, ki jih razpoznavamo. Celoten postopek grafično prikazuje še slika 2.1. Postopki, ki so pobarvani z enako barvo, lahko potekajo vzporedno, ostali pa zaporedno. V magistrski nalogi razpoznave objektov ne bomo opisovali ali izvajali, saj je namen naloge le segmentacija in določanje lastnosti. Posamezni postopki so bolj podrobno opisani v 5. poglavju, grafično pa so predstavljeni na slikah 3.5 do 3.7.



Slika 2.1: Predlagana rešitev razpoznave objektov na vidni sceni.

## 2.3 Opis implementacije

Za implementacijo uporabljenih metod smo uporabili programsko okolje The MathWorks Matlab, ki je programski jezik in obenem okolje za numerično računanje [7], [8]. Zelo dobro podpira računanje z matrikami, grafični prikaz podatkov in funkcij, interakcijo z ostalimi programskimi jeziki ter izdelavo uporabniških vmesnikov. Na voljo so tudi vgrajena orodja, ki olajšajo delo pri določenih nalogah (na primer *Image processing Toolbox*, ki omogoča delo s slikami). Za implementacijo v Matlabu smo se odločili zaradi lažjega in hitrejšega programiranja kot je mogoče v nižje nivojskem programskem jeziku, kot na primer Java ali C++. Poleg tega je vizualizacija rezultatov v Matlabu hitrejša in preprostejša zaradi že obstoječih funkcij za prikaz slik, grafov in rezultatov. Slabost implementacije v Matlabu je sicer počasnejše izvajanje programov ter slabša izkoriščenost sistemskih virov, vendar naša naloga ni bila čim hitrejše izvajanje programov, temveč ocena uspešnosti delovanja implementiranih metod.

Pri implementaciji tenzorskega glasovanja smo se zgledovali po že napisanem

orodju za Matlab, to je Tensor Voting Toolbox, ki ga je izdelal Gabriel Peyré [9]. Ker večina funkcij v tem orodju ni delovala v skladu z našimi pričakovanji, smo jih prilagodili ali na novo napisali. Potrebno je bilo dodati tudi funkcije za analizo dobljenih tenzorskih polj in določanje krivulj. Implementirali smo tudi funkcije za prileganje kontrolnih točk B-zlepkov iz podatkovnih točk, izris B-zlepkov ter zapis rezultata segmentacije z vloženim grafom.

## 2.4 Nosilna literatura

Pri implementaciji omenjenih metod smo se zgledovali po nosilni literaturi na posameznem področju. Nosilna literatura za tenzorsko glasovanje je poglavje o tenzorskem glasovanju v knjigi “Emerging topics in computer vision” [4], knjiga “A Computational Framework for Segmentation and Grouping” [10] ter članek “First Order Augmentation to Tensor Voting for Boundary Inference and Multiscale Analysis in 3D” [11]. Pri implemetaciji opisa krivulj z B-zlepki smo se zgledovali po članku “Invariant Matching and Identification of Curves Using B-Splines Curve Representation” [5]. Pri opisu teorije grafov pa smo se zgledovali po knjigi “Combinatorial Optimization: Theory and Algorithms” [6].

### 3. Predlagana rešitev

V tem poglavju bomo predstavili pojme in oznake, ki se v delu večkrat ponavljajo in katerih predstavitev je ključna za razumevanje metode, opisane v magistrskem delu. Nekatere pojme bomo predstavili tudi s preprostimi primeri. V nadaljevanju bomo predstavili predlagano rešitev za segmentacijo in opis vizualne scene. Opisali bomo posamezne dele rešitve in njihov potek ponazorili na diagramu. Podali bomo tudi način ovrednotenja končnih rezultatov.

#### 3.1 Vpeljava oznak in pojmov

V tem podpoglavju bomo vpeljali pojme, ki se pojavljajo v delu, ter opisali matematične oznake, ki jih uporabljamo.

##### Skalarne vrednosti

Skalarne vrednosti in funkcije zapišemo z malimi, ležečimi in nepoudarjenimi črkami, na primer  $x$  ali  $f(x)$ . Na skalarno vrednost znotraj vektorja ali matrike se sklicujemo z indeksom spodaj, na primer  $x_i$  ali  $m_{i,j}$ .

##### Vektorji

Vektorje zapišemo z malimi, ležečimi in poudarjenimi črkami, na primer  $\mathbf{q}$ . Definiramo jih kot stolpčne vektorje

$$\mathbf{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} = [q_1 \ \dots \ q_n]^T. \quad (3.1)$$

## Matrike

Matrike zapišemo z velikimi, poudarjenimi pokončnimi črkami, na primer  $\mathbf{M} = [m_{i,j}]$ .

## Točke v ravnini

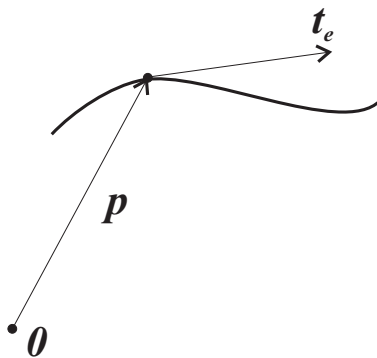
Točko v ravnini zapišemo in predstavimo kot vektor  $\mathbf{p} = [x, y]^T$ , katerega komponente predstavljajo položaj točke v ravnini.

## Krivulje v ravnini

Za potrebe tenzorskega glasovanja potrebujemo točko s smerjo. To je par  $(\mathbf{p}, \mathbf{t}_e)$ , kjer je  $\mathbf{t}_e$  tangenta na krivuljo v točki  $\mathbf{p}$ . Krivuljo, ki je rezultat tenzorskega glasovanja in analize tenzorskih ter vektorskih polj, predstavimo kot zaporedje točk. To zaporedje točk zapišemo v vektor

$$\mathbf{r} = [\mathbf{r}_1 \dots \mathbf{r}_N], \quad (3.2)$$

kjer je  $\mathbf{r}_i = [x_i, y_i]^T$ ,  $i = 1 \dots N$ .



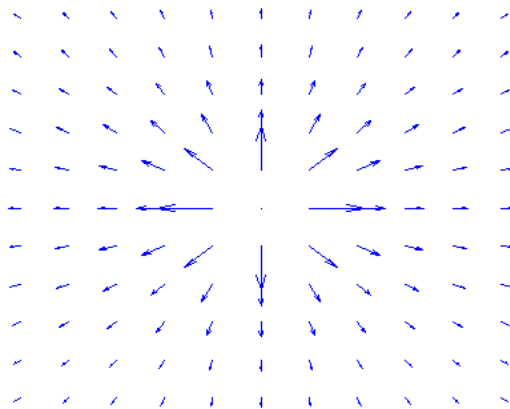
Slika 3.1: Primer točke na krivulji ter tangente na krivuljo v tej točki.

## Vektorska polja

Vektorsko polje definiramo kot preslikavo

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

ki vsakemu  $\mathbf{x} \in \mathbb{R}^n$  priredi vrednost vektorske funkcije  $\mathbf{f}(\mathbf{x})$  na tem mestu. Vektorska funkcija je funkcija ene ali več spremenljivk, ki imajo v primerjavi s spremenljivkami pri skalarnih funkcijah, zalogo vrednosti dvo ali več dimenzijsko (v splošnem  $n$ -dimenzijsko). Vektorska polja se pogosto uporablja na primer v fiziki, kjer želimo predstaviti smer in hitrost premikanja tekočin v prostoru ali moč in smer sil, kot so električna ali magnetna. Na sliki 3.2 je prikazan primer vektorskega polja, ki prikazuje jakost električnega polja v prostoru zaradi naelektrenega delca.



Slika 3.2: Električno polje naelektrenega delca.

## Tenzorji

Tenzor je matematični objekt in je posplošitev skalarjev in vektorjev. Lahko zapišemo:

- skalar je tenzor 0. reda (ima samo velikost),
- vektor je tenzor 1. reda (ima velikost in eno smer),
- tenzor 2. reda ima 2 velikosti in 2 smeri.

Denimo, da  $\mathbf{e}_i$ ,  $i = 1, 2$ , tvorijo ortonormirano bazo  $\mathbb{R}^2$  prostora. Tenzor 2. reda ima 2 velikosti ter 2 smeri, zaradi česar ima v 2-dimenzijskem prostoru 4 komponente. Komponente tenzorja lahko izpeljemo kot tenzorski produkt dveh vektorjev. Če sta

$$\mathbf{u} = u_1\mathbf{e}_1 + u_2\mathbf{e}_2 \text{ in } \mathbf{v} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2,$$

potem je tenzorski produkt definiran kot

$$\mathbf{T} = \mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^\top = u_1v_1\mathbf{e}_1\mathbf{e}_1^\top + u_1v_2\mathbf{e}_1\mathbf{e}_2^\top + u_2v_1\mathbf{e}_2\mathbf{e}_1^\top + u_2v_2\mathbf{e}_2\mathbf{e}_2^\top \text{ oz.}$$

$$\mathbf{u} \otimes \mathbf{v} = t_{11}\mathbf{e}_1\mathbf{e}_1^\top + t_{12}\mathbf{e}_1\mathbf{e}_2^\top + t_{21}\mathbf{e}_2\mathbf{e}_1^\top + t_{22}\mathbf{e}_2\mathbf{e}_2^\top.$$

Skalarne vrednosti  $t_{ij}$  lahko zapišemo kot  $2 \times 2$  matriko, ki jo imenujemo matrika tenzorja  $\mathbf{T}$ .

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \quad (3.3)$$

S to matriko lahko predstavimo tenzor v dani bazi prostora. Za tenzor, predstavljen z matriko, veljajo vse operacije iz matrične algebre, kot so npr. seštevanje matrik ali spektralna analiza, s katero matriko razstavimo na lastne vrednosti in lastne vektorje. Tenzor zapišemo tako kot matriko, z velikimi, poudarjenimi pokončnimi črkami.

Tenzorski produkt vektorjev ni komutativen. Za tenzor ranga 1, ki je oblike  $\mathbf{T} = \mathbf{u}\mathbf{v}^\top$ , in vektor  $\mathbf{s}$  vpeljemo tenzorski produkt z desne in leve, dana sta z

$$\mathbf{s} \otimes \mathbf{T} = (\mathbf{s}^\top \mathbf{u})\mathbf{v} = \varphi\mathbf{v}, \quad \mathbf{T} \otimes \mathbf{s} = (\mathbf{s}^\top \mathbf{v})\mathbf{u} = \lambda\mathbf{u}.$$

Če vektor množimo z desne, dobimo vektor katerega dolžina je  $\varphi = \mathbf{s}^\top \mathbf{u}$ , smer pa določa  $\mathbf{v}$ . Če vektor množimo z leve strani, je rezultat vektor, katerega dolžina je  $\lambda = \mathbf{s}^\top \mathbf{v}$ , smer pa določa  $\mathbf{u}$ .

Pri množenju vektorja s skalarjem dobimo nov vektor, ki ima enako smer kot originalni vektor, drugačno ima lahko le dolžino. Rezultat množenja vektorja s tenzorjem 2. reda pa je nov vektor, ki se od originalnega lahko razlikuje tako po dolžini, kot tudi po smeri. Torej je tenzor 2. reda linearni operator, ki vektor preslika v drug vektor. V magistrskem delu se bomo ukvarjali le s simetričnimi tenzorji drugega reda, zato bomo v nadaljevanju tak tenzor opisovali le z besedo tenzor. Pri metodi tenzorskega glasovanja izkoristimo naslednje lastnosti tenzorjev:

- tenzor ima 2 velikosti in 2 smeri,
- seštevanje tenzorjev je identično seštevanju matrik,
- nad tenzorjem je mogoče izvesti spektralno analizo.

Omenjene lastnosti in njihovo uporabo pri metodi tenzorskega glasovanja bomo podrobneje opisali v poglavju 5.

### Tenzorska polja

Tenzorsko polje je posplošitev vektorskega polja. Tenzorsko polje vsaki točki ravnine (v splošnem prostora) priredi tenzor. Je preslikava

$$\mathbf{T} : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m},$$

ki vsakemu  $\mathbf{X} \in \mathbb{R}^n$  priredi vrednost tenzorske funkcije  $\mathbf{T}(\mathbf{X})$  na tem mestu. Tenzorska polja se tako kot vektorska uporabljajo med drugim v fiziki, npr. za preučevanje napetosti, sil in obremenjenosti v različnih materialih.

### Žetoni

Z žetoni predstavimo elemente, ki sodelujejo pri nastanku struktur iz podatkov. Pri slikah, kjer iščemo krivulje in presečišča, ki jih tvorijo slikovni elementi, so žetoni kar slikovni elementi slike. V magistrskem delu bomo torej izraz žeton enačili s slikovnim elementom. Žetoni oziroma slikovni elementi lahko pripadajo krivulji. Takim žetonom pravimo, da so usmerjeni. Žetoni lahko pripadajo tudi presečiščem krivulj ali nepovezanim točkam, ki so posledica šuma. Takim žetonom pravimo neusmerjeni žetoni. Žetoni so vhodni elementi pri metodi tenzorskega glasovanja. Zapišemo jih s tenzorji, ki nato glasujejo in si izmenjujejo informacije.

### Digitalne slike

Način zapisa digitalne slike je povezan s procesom zajema slike v slikovnem sensorju. Slika se formira v slikovni ravnini slikovnega sensorja po prehodu svetlobe skozi optični sistem. Intenziteto vpadne svetlobe v slikovni ravnini zapišemo z dvodimenzijsko funkcijo  $g(x, y) \in R$ , kjer sta  $x$  in  $y$  koordinati točke v slikovni ravnini. Slikovni sensor, ki leži v slikovni ravnini, predstavlja pravokotno mrežo točk, v katerih vzorčimo intenziteto vpadne svetlobe, rezultat vzorčenja pa so diskretne vrednosti intenzitete vpadne svetlobe. Dimenzije pravokotne mreže točk označimo z  $M$  in  $N$ .

Digitalna slika predstavlja preslikavo, ki vsaki točki slike  $[x, y]^T$  priredi vrednost svetilnosti (intenzitete, sivega nivoja) iz nabora diskretnih vrednosti  $G_L$

$$g : I_M \times I_N \rightarrow G_L,$$

kjer predstavljata  $I_M = \{0, \dots, M - 1\}$  in  $I_N = \{0, \dots, N - 1\}$  podmnožici celih nenegativnih števil,  $I_M \times I_N$  pa je njun kartezični produkt, ter  $G_L = \{0, \dots, L - 1\}$  množico diskretnih vrednosti svetilnosti slike.  $g(x, y) \in G_L$  torej označuje vrednost svetilnosti v točki slike  $(x, y) \in I_M \times I_N$ . Sliko predstavimo v obliki pravokotne matrike, katere elementi definirajo vrednosti slikovnih elementov

$$\mathbf{G} = [g_{xy}]_{M \times N}; g_{xy} = g(x, y).$$

## Grafi

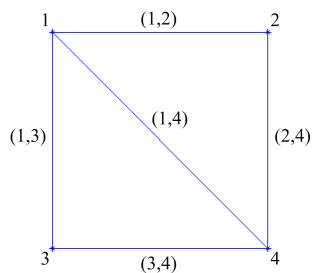
Graf zapišemo kot urejeno dvojico  $(V, E)$ , kjer je  $V = \{v_1, \dots, v_n\}$  neprazna, končna množica vozlišč grafa in  $E = \{e_1, \dots, e_m\}$  množica povezav vozlišč grafa, kjer je  $e_i = (v_k, v_l)$ ,  $v_k, v_l \in V$ . Grafe ponazorimo grafično z diagrami. Teorija grafov se uporablja na primer pri načrtovanju elektronskih vezij, pri telekomunikacijskih omrežij, pri prometu in podobno. Podrobneje jo bomo predstavili v poglavju 5.4, zato v tem poglavju prikazujemo le enostaven primer grafa z množico vozlišč

$$V = \{1, 2, 3, 4\}$$

in množico povezav

$$E = \{(1, 2), (1, 4), (3, 1), (2, 4), (4, 3)\}.$$

Diagram grafa je prikazan na sliki 3.3.



Slika 3.3: Primer diagrama grafa.



## B-zlepki

B-zlepki so eden izmed najbolj optimalnih parametričnih zapisov krivulj [5]. Uporabljajo se pri računalniško podprtem načrtovanju in računalniški grafiki. B-zlepek zapišemo kot

$$\mathbf{r}(t) = \sum_{i=0}^{n+k-1} \mathbf{c}_{i-1} Q_{i,k}(t), \quad (3.4)$$

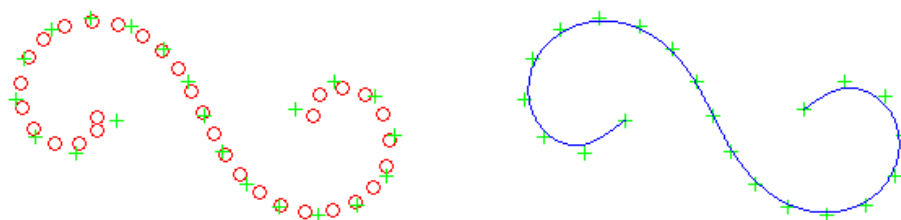
kjer so

$$\mathbf{c}_i = [c_{ix}, c_{iy}]^T \quad (3.5)$$

kontrolne točke,  $Q_{i,k}(t)$  pa je bazna funkcija  $k$ -tega reda. V okviru tega dela smo predpostavljali, da zleпки ležijo v  $\mathbb{R}^2$ . Kontrolne točke lahko zapišemo kot vektor

$$\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_n], \quad (3.6)$$

s katerim lahko na zgoščen način predstavimo krivuljo. B-zlepke bomo podrobneje obravnavali v poglavju 5.3, zato bomo na sliki 3.4 prikazali le primer B-zlepka.



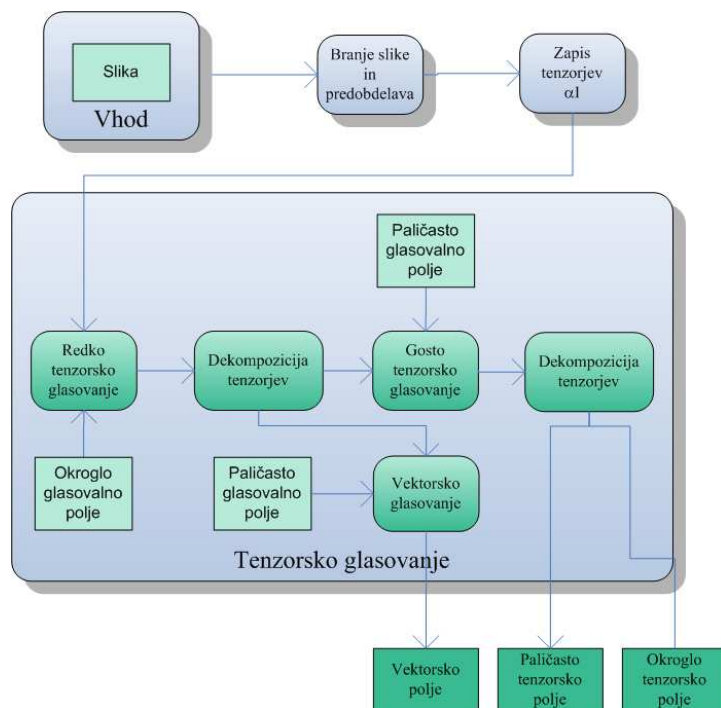
Slika 3.4: Primer B-zlepka.

Z rdečimi krogi so predstavljeni vhodni podatki, to so točke krivulje. Z zelenimi plusi so prikazane kontrolne točke, ki so rezultat prilaganja B-zlepkov, z modro krivuljo pa je prikazan B-zlepek, ki je določen s kontrolnimi točkami.

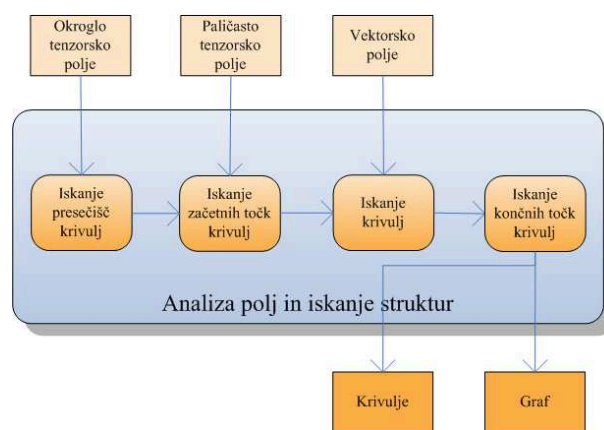
## 3.2 Potek predlagane rešitve

Na spodnjih slikah je prikazan potek celotne metode, predlagane v magistrski nalogi. Vhod v predstavljeno metodo je digitalna slika, zajeta s senzorjem, ki jo ustrezno predobdelamo. Prvi del (slika 3.5) predlagane rešitve obsega segmentacijo s tenzorskim glasovanjem. Ta del zajema branje slike s trdega diska računalnika, kodiranje slike s tenzorji v tenzorsko polje, konstrukcijo tenzorskih

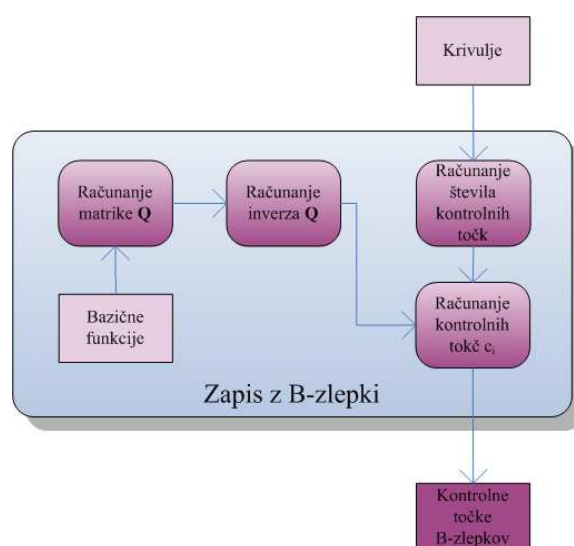
in vektorskih glasovalnih polj, tenzorsko in vektorsko glasovanje ter dekompozicijo dobljenih tenzorskih polj. Ta del je podrobneje opisan v poglavju 5.1. Drugi del (slika 3.6) obsega analizo tenzorskih in vektorskih polj ter iskanje struktur kot so presečišča, končne točke ter krivulje. Na podlagi lastnosti tenzorskih polj poiščemo presečišča krivulj, nato pa začetne točke krivulj, ki jim sledimo in sproti iščemo končne točke krivulj na podlagi vektorskega polja. V procesu iskanja teh struktur gradimo tudi graf vozlišč in povezav. Ta del je podrobneje opisan v poglavju 5.2. Tretji del (slika 3.7) je učinkovit zapis krivulj s koeficienti B-zlepkov, ki je opisan v poglavju 5.3. Rezultat celotne rešitve je opis vizualne scene z vloženim grafom (poglavje 5.4), kjer vozlišča predstavljajo presečišča in končne točke krivulj, povezave pa predstavljajo krivulje med temi točkami. Povezave v grafu so labelirane z vektorjem koeficientov B-zlepkov. Slike, prikazane v tem poglavju, bodo vodile opis posameznih delov v naslednjem poglavju.



Slika 3.5: Segmentacija s tenzorskim glasovanjem.



Slika 3.6: Analiza tenzorskih polj in izgradnja grafa.



Slika 3.7: Zapis krivulj z B-zlepki.

### 3.3 Način ovrednotenja rezultatov

Rezultate predlagane rešitve bomo ovrednotili na naslednji način. Na izbranih predobdelanih testnih slikah bomo izvedli metodo tenzorskega glasovanja. Preizkusili bomo vpliv velikosti glasovalnih polj ter vpliv dodajanja šuma v vhodnih slikah na rezultate. Rezultate segmentacije s tenzorskim glasovanjem bomo podali grafično in tabelarično. Kot mero za uspešnost metode tenzorskega glasovanja bomo uporabili razliko med ploščinami objektov, ki jih določajo krivulje v originalni sliki in sliki, ki ji bomo v postopku dodajali šum. Raziskali bomo tudi vpliv velikosti glasovalnih polj na časovno zahtevnost. V magistrskem delu pred-

lagane metode ne bomo primerjali z nobeno od že obstoječih metod za analizo in opis slik. Razlog v tem je v naravi problema opisa slike, ki je zelo zahteven in kompleksen, rezultate celotne rešitve pa je težko ovrednotiti. Lahko bi le primerjali segmentacijo s tenzorskim glasovanjem s kako drugo metodo segmentacije, vendar zaradi začetne faze implementacije tenzorskega glasovanja tega nismo izvajali. Rezultati predlagane metode so podani v poglavju 6.

## 4. Obstoječe stanje na področju analize vidne scene

V tem poglavju bomo predstavili izbrana področja oziroma aplikacije, v katerih se uporablja opis vidne scene. Opisali bomo dosedanje delo na področju opisa vidne scene, tako v industrijskem kot tudi v raziskovalnem okolju. Omenili bomo nekatere postopke, ki so jih uporabili drugi avtorji za opis vidne scene. Ker bomo v delu za zapis krivulj uporabljali B-zlepke, bomo v tem poglavju naredili tudi kratek pregled parametričnih zapisov krivulj.

### 4.1 Primeri uporabe analize vidne scene v realnem okolju

Analiza vsebine vidne scene je pomemben sestavni del komunikacije človeka s strojem ali stroja z okoljem. Uporablja se na različnih področjih:

- Inteligentni uporabniški vmesniki [12]: računalniki, stroji in roboti postajajo vedno bolj kompleksne naprave. Zaradi lažjega upravljanja s temi napravami želimo poenostaviti njihove uporabniške vmesnike. Z uporabo kamere in analize slike, ki jo zajamejo, lahko naprava razpozna obraz ali gesto uporabnika, nato pa ustrezno ukrepa.
- Iskanje po bazah slik in video posnetkov [13], [14]: za iskanje določenih prizorov v sliki moramo indeksirati velike množice slik in video posnetkov. S pomočjo analize vidne scene lahko ta postopek vsaj v neki meri avtomatiziramo.
- Varovanje in nadzor [15]: z analizo vidne scene lahko spremljamo določeno področje in spremljamo sumljivo obnašanje oseb, merimo gostoto ljudi ali pomikanje množice. Analizo vidne scene lahko uporabimo tudi v prometu

za spremljanje vozil na nevarnih odsekih cest (tuneli). Sistem lahko ob zaznani nevarnosti (vozilo vozi v napačni smeri) sam sproži alarm.

- Avtomatizacija procesov [16]: v industriji lahko z analizo slik, zajetih s kamero, zagotavljamo kakovost izdelkov.
- Medicina [17]: z analizo vidne scene lahko sistem samodejno interpretira slike, dobljene iz različnih medicinskih slikovnih senzorjev in pacientom postavi diagnozo, ki jo mora nato potrditi tudi človek.

## 4.2 Dosedanje delo na področju analize vidne scene

V tem podpoglavju bomo naredili pregled dosedanjega dela na področju analize vidne scene. Predstavili bomo raziskovalno delo na tem področju in opisali nekaj izdelkov, ki se jih da dobiti na trgu.

### 4.2.1 Raziskovalno delo

Večina znanstvenih člankov na področju analize vidne scene priča o tem, da se avtorji največ ukvarjajo s problemom analize in opisa slik za iskanje glede na vsebino po bazah slik in videa. Cristani in drugi [18] pa opisujejo tudi detekcijo dogodkov v video nadzornih sistemih, vendar poleg analize slik in videa uporabljajo tudi analizo zvoka, nato pa obe modalnosti združijo. Njihova metoda zna detektirati prihod človeka v video sceno. Video analiza poteka z detekcijo sprememb v slikovnih elementih slike in v histogramu celotne slike. Čeprav ta metoda ne omogoča popolnega opisa vidne scene, ga omenjamo zaradi fuzije podatkov iz slikovnih in audio senzorjev. Še ena metoda za detekcijo dogodkov v videu je opisana v [19]. Avtorji opisujejo metodo, ki regije roji istočasno v prostoru in času z GMM (angl. *Gaussian Mixture Model*) modelom. Rezultat so regije, ki so uniformne tako v posamezni sliki videa kot tudi časovno preko celotnega videa. Medtem, ko so pri analizi videa na voljo tudi časovne informacije, teh pri posameznih slikah ni. Pregled področja analize in iskanja slik glede na vsebino je narejen v [14]. Avtorji pri analizi slik (oziroma analizi multimedijskega materiala) izpostavljajo dva glavna problema:

- detekcija objektov iz kompleksnih ozadij in določitev nizko-nivojskih značilk,
- omogočanje prehoda med nizko-nivojskimi značilkami ter visoko-nivojskimi značilkami, ki so uporabne za človeka (angl. *bridging the semantic gap*). To zajema učenje sistema in razpoznavo objektov, zapisanih v prostoru značilk (angl. *feature space*).

Prvi problem je stvar digitalne obdelave slik, metode za predobdelavo, segmentacijo in določanje značilk pa so dobro opisane v več knjigah, npr. [2], [13], [20], [21], [22] in [23]. Večina metod, predlaganih v člankih, za nizko-nivojske značilke v glavnem uporabljajo barvo, teksturo, obliko in velikost objektov. Smith in Li sta opisala postopek za opis slike s kompozitnimi šablonami regij (angl. *Composite Region Templates*) [24]. Metoda zajema segmentacijo na osnovi barve, zapis posameznih barvnih področij z nizi regij in zapis slike z CRT vzorci. Iskanje slik z določenim motivom nato izvajata z iskanjem po CRT vzorcih. Chengliang in drugi opisujejo segmentacijo za opis slike na podlagi segmentacije robov in regij [25]. Segmentacijo izvajajo z *watershed* algoritmom.

Drugi problem je problem področij umetne inteligence, razpoznavanja vzorcev in rudarjenja s podatki. Therrien za učenje in razpoznavo me drugim omenja Karhunen-Loeve transformacijo ali metodo glavnih komponent [26]. Izboljšavo te metode so predlagali Capelli in drugi z vpeljavo večprostorske KL transformacije (angl. *multispace KL*) [27]. Zhou in Huang sta za učenje in razpoznavo primerjala SVM (angl. *Support Vector Machine*) in različne diskriminantne metode [28]. Lew in Denteneer sta za uporabo v ta namen opisala FLT (angl. *Fisher's Linear Discriminant*) metodo [29]. Krishnapuram in drugi so za opis scene v sliki uporabili podoben pristop, kot ga predlagamo v magistrski nalogi. Vidno sceno so zapisali z uporabo grafov, kjer so posamezne objekte predstavili z vozlišči grafa, odnose med njimi pa s povezavami med vozlišči [30].

Fan in drugi so opisali celotno ogrodje za analizo in iskanje slik [31], kjer so rešili oba omenjena problema. Za nizko-nivojske značilke so uporabili tako imenovane izrazite objekte (angl. *salient objects*), ki so jih dobili s tremi različnimi algoritmi: detekcija na osnovi segmentacije, detekcija na osnovi mreže in detekcija na osnovi posameznih delov. Izrazite objekte so nato zapisali z 139 dimenzijskimi

značilkami, ki med drugim vključujejo barvni histogram, dominantno barvo in različne oblike značilk, dobljenih iz tekstur. Za določanje relacij med posameznimi objekti in pomenskimi koncepti so razvili PoM (angl. *Product of Mixture-experts*) algoritem. Avtorji so razvili tudi ontologijo za koncepte, ki jih določijo v slikah. Na tak način lahko sistem v sliki zazna npr. računalnik, monitor, tipkovnico in mizo, na podlagi ontologije pa sklepa, da je na sliki pisarna. Razvili so tudi tako imenovano hiperbolično predstavitev slik za sestavljanje poizvedb in prikaz rezultatov poizvedb.

#### 4.2.2 Obstoječi izdelki

Ker bi bil pregled celotnega področja analize vidne scene preveč obsežen, smo se pri industrijskih produktih, ki so večinoma na voljo kot programska oprema, osredotočili le na izdelke za video nadzor, ki sceno snemajo, analizirajo in ob določenih dogodkih ustrezno ukrepajo ter na sisteme za iskanje slik po bazah. Idealni sistem za iskanje slik po bazah bi omogočal semantično iskanje, kar pomeni, da bi lahko za iskalni niz vpisali npr. “najdi vse slike avtomobilov” ali “najdi vse slike konjev”. Ker je to zaradi različnih pojavitev iskanih objektov pretežko, večina obstoječih sistemov uporablja za iskanje in ujemanje nižje-nivojske značilke, kot so barva, tekstura in oblika. Nekateri sistemi omogočajo iskanje po podobnih slikah (če želimo najti slike sončnega zahoda, moramo sistemu podati referenčno sliko sončnega vzhoda), pri nekaterih pa lahko narišemo preprosto skico objekta, ki ga iščemo. Obstajajo tudi sistemi, ki omogočajo iskanje po medsebojnih prostorskih odnosih (avto pred hišo ipd.), vendar je treba poizvedbo zopet podati kot sliko ali 3D sceno, v kateri so narisani iskani objekti.

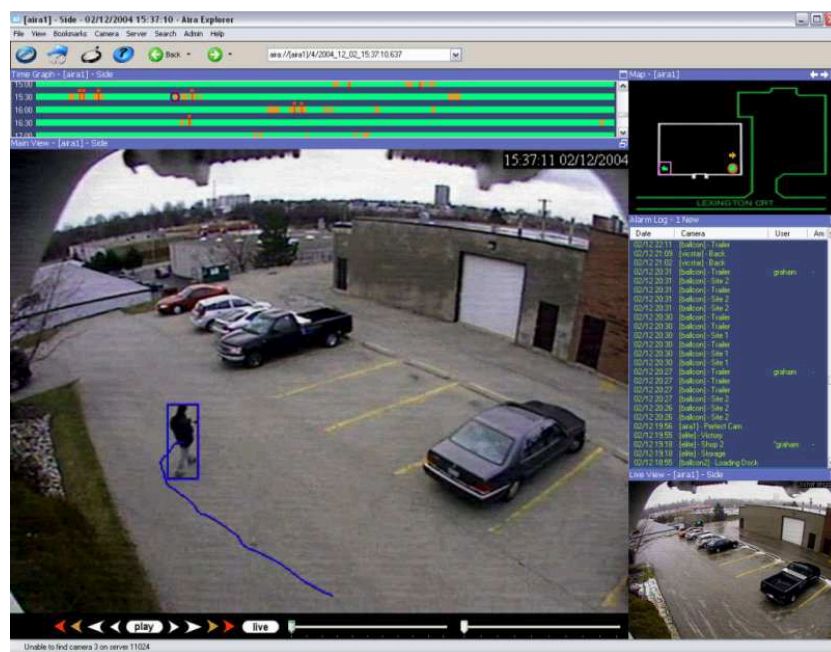
#### Aimetis AIRA v5

Aimetis je podjetje, ki se ukvarja z storitvami inteligentnega nadzora na podlagi računalniškega vida. Njihov produkt AIRA v5 [32] je programski paket za analizo videa in upravljanje kamer, ki se ga da namestiti na osebni računalnik z operacijskim sistemom Microsoft Windows 2000 ali novejšim. AIRA je kompatibilen z analognimi, mrežnimi (IP) in PTZ (angl. *pan, tilt, zoom*) kamerami in omogoča hkratno snemanje, video analizo v realnem času in oddaljen dostop do živega in



posnetega videa s kateregakoli omreženega računalnika. AIRA lahko avtomatsko sledi in uvrsti objekte, kot so pešci in avtomobili in posreduje vsebino varnostnemu osebju. Omogoča tudi štetje ljudi in iskanje zapuščenih predmetov. Slika 4.1 prikazuje uporabniški vmesnik paketa AIRA.

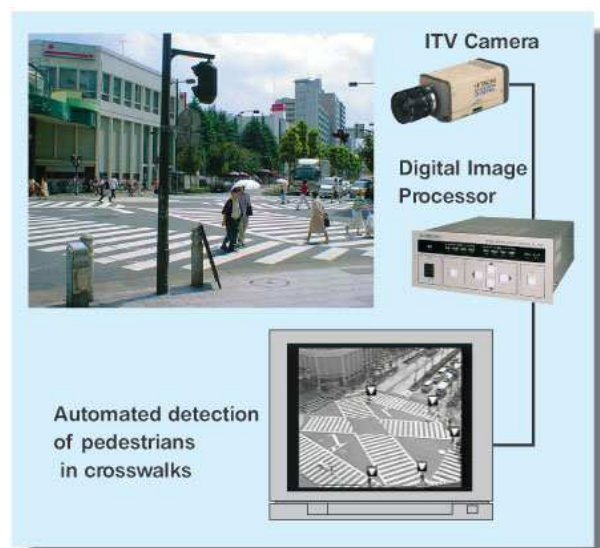
AIRA je zasnovana na lastnih algoritmih za sledenje, ki delujejo v različnih vremenskih pogojih ter pri spremembah v osvetlitvi.



Slika 4.1: Uporabniški vmesnik programskega paketa Aimetis AIRA v5.

## Hitachi ITS21

Hitachi Intelligent Transport System ITS21 [33] je sistem, ki združuje sistem za video nadzor in metode za računalniški vid. Omogoča detekcijo hitrosti posameznih vozil, zgostitev prometa, dolžine vrst vozil ter pešce na prehodih za pešce. Samodejno zazna napačno parkirana vozila in kršitve prometnih znakov, razpozna pa tudi registrske tablice na vozilih. Arhitektura sistema je prikazana na sliki 4.2.



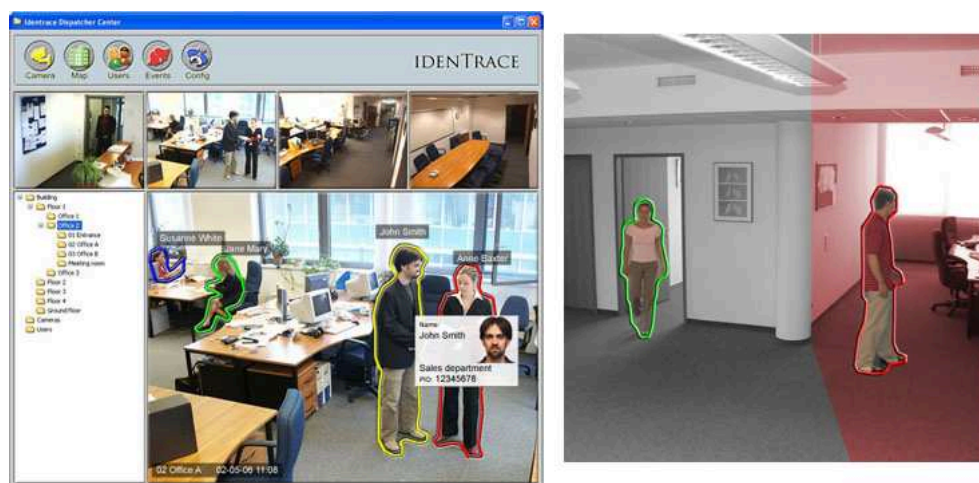
Slika 4.2: Sistem Hitachi ITS 21.

## IDENTRACE - Intelligent Video Surveillance System

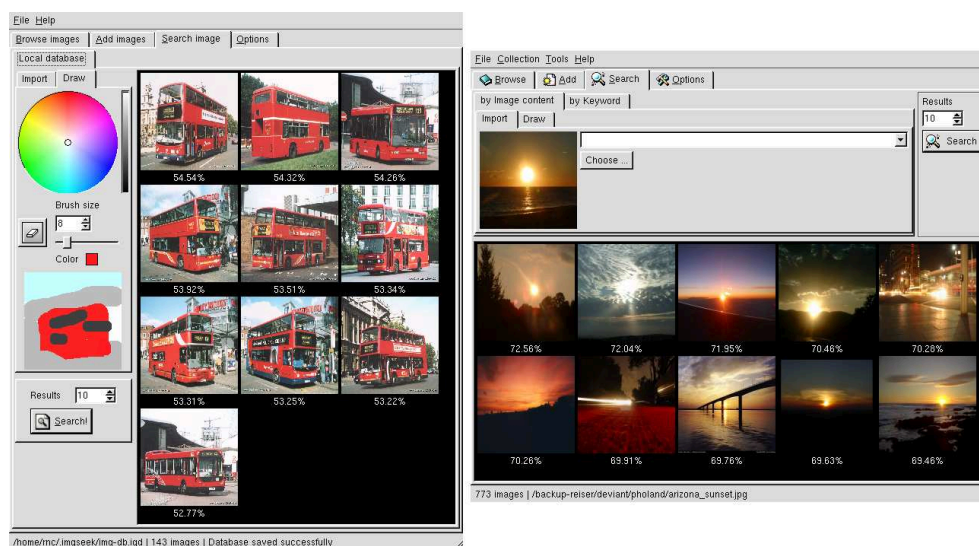
IDENTRACE [34] je prototip, ki je rezultat raziskovalnega projekta IBAR04. Omogoča združevanje obstoječih sistemov za video nadzor in omejevanje dostopa z identifikacijskimi metodami. Glavna naloga sistema je sledenje in identifikacija oseb v prostorih. Identifikacija oseb temelji na razpoznavi obrazov, ušes, merjenju višine ter barvi oblačil. Poleg sledenja in identifikacije omogoča tudi vzpostavitev 3D modelov stavb, v katere lahko postavi osebe, ki jim sledi. Omogoča tudi vzpostavitev navideznih področij (slika 4.3), ki jih določene osebe ne smejo prečkati, čeprav ta področja niso fizično ločena od drugih.

## imgSeek

imgSeek [35] je odprtokodni program za upravljanje s slikami, ki omogoča tudi iskanje po zbirkah slik. Omogoča iskanje po meta podatkih slik in poizvedbe s podajanjem preprostih shem, ki jih uporabnik nariše, ali s podajanjem slik, ki vsebujejo iskane objekte (slika 4.4). imgSeek za ujemanje slik uporablja večločljivostno dekompozicijo valčkov (angl. *multiresolution wavelet decomposition*) referenčne slike in slik v bazi.



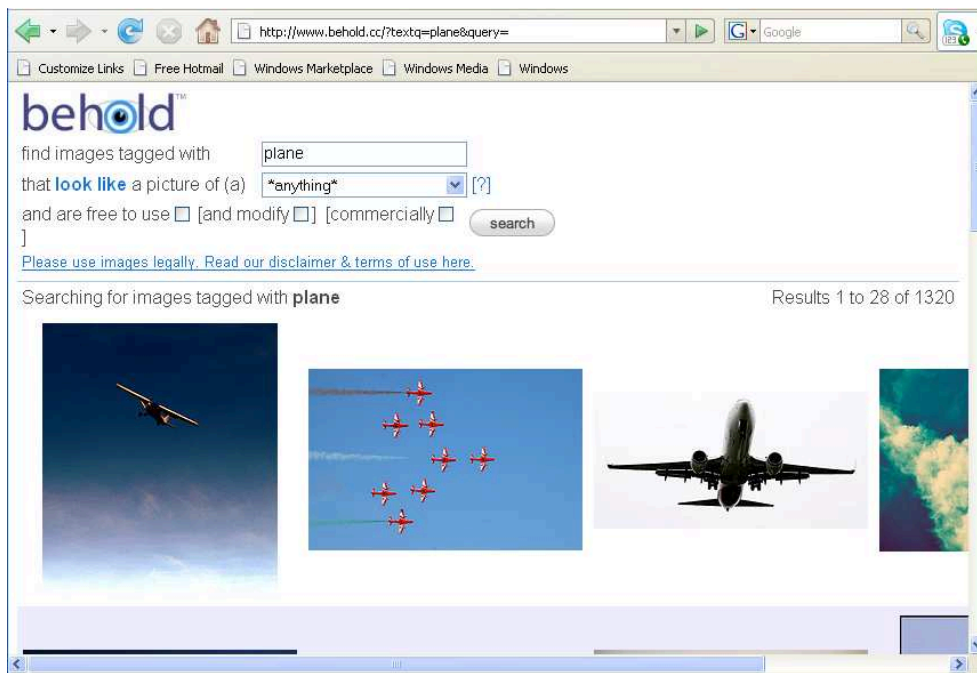
Slika 4.3: Uporabniški vmesnik sistema IDENTRACE in navidezna področja.



Slika 4.4: Program imgSeek za iskanje po slikah.

## Behold

Behold [36] je spletna storitev, ki omogoča iskanje po slikah znotraj aplikacije Flickr [37]. Behold uporablja metode računalniškega vida za analizo slik in razpoznavo objektov na slikah. Slike označi s pomenskimi koncepti, ki jih najde v slikah in omogoči iskanje po teh konceptih (slika 4.5). Za obdelavo slik Behold uporablja Amazonovo spletno storitev za porazdeljeno računanje Amazon Elastic Compute Cloud [38].



Slika 4.5: Spletna storitev Behold za iskanje po slikah.

### 4.3 Parametrični zapis krivulj

V tem poglavju bomo podali kratek pregled parametričnih zapisov krivulj. Omenili bomo Hermitske ter Bezierove krivulje ter motivacijo za uporabo B-zlepkov.

Krivulje lahko zapišemo na eksplicitni (4.1), implicitni (4.2) ali parametrični (4.3) način:

$$y = kx + n, \quad (4.1)$$

$$f(x, y) = 0, \quad (4.2)$$

$$x = f(t), y = g(t), t \in [a, b]. \quad (4.3)$$

Čeprav pri eksplicitnem zapisu najlažje izračunamo vrednosti  $y$ , je ta zapis v določenih primerih neuporaben. Na tak način ne moremo zapisati krivulje, ki ima v neki točki navpično tangento. Zaradi težjega računanja vrednosti, ki jih zavzamejo točke na krivulji, tudi implicitni način ni primeren. Zato za zapis kompleksnejših krivulj uporabimo parametrični zapis, ki je neodvisen od koordinatnega sistema. Ta zapis med drugim omogoča enostavno rotacijo in premik koordinatnega sistema, zapis sklenjenih krivulj in krivulj z navpično tangento. Tak zapis krivulj se uporablja predvsem v računalniško podprtem načrtovanju.

Parametrični zapis nas privede tudi do odsekoma polinomskih funkcij (zlepkov), s katerimi lahko zapišemo krivuljo poljubne oblike.

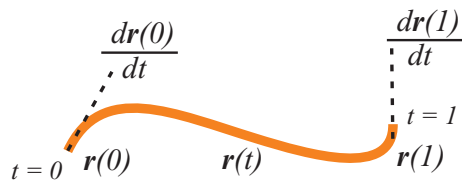
Parametrično krivuljo tretjega reda zapišemo kot

$$\mathbf{r}(t) = \mathbf{c}_3 t^3 + \mathbf{c}_2 t^2 + \mathbf{c}_1 t + \mathbf{c}_0, \quad \mathbf{r}(t) = [x(t), y(t)]^\top, \quad \mathbf{c}_i = [c_{ix}, c_{iy}]^\top, \quad (4.4)$$

kjer so  $\mathbf{c}_i$  kontrolne točke, ki določajo obliko krivulje, določimo jih pa z reševanjem enačb, ki jih določajo podane točke in zahteve za gladkost krivulje. Glede na podane pogoje za krivuljo lahko med drugim uporabimo Hermitske ali Bezierove krivulje.

### 4.3.1 Hermitske krivulje

Hermitska krivulja tretjega reda je določena z začetno in končno točko in odvodoma v teh točkah (slika 4.6). Enačbo (4.4) lahko zapišemo tudi kot



Slika 4.6: Krivulja, določena z začetno in končno točko in odvodoma v teh točkah.

$$\mathbf{r}(t) = [t^3 \ t^2 \ t \ 1] \begin{bmatrix} \mathbf{c}_3 \\ \mathbf{c}_2 \\ \mathbf{c}_1 \\ \mathbf{c}_0 \end{bmatrix} = [t^3 \ t^2 \ t \ 1] \mathbf{C}. \quad (4.5)$$

Z upoštevanjem začetne in končne točke ter odvodov lahko zapišemo

$$\begin{bmatrix} \mathbf{r}(0) \\ \mathbf{r}(1) \\ \frac{d\mathbf{r}(0)}{dt} \\ \frac{d\mathbf{r}(1)}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}_3 \\ \mathbf{c}_2 \\ \mathbf{c}_1 \\ \mathbf{c}_0 \end{bmatrix} \quad \text{ali } \mathbf{G}_H = \mathbf{B}_H \mathbf{C}. \quad (4.6)$$

Od tod dobimo

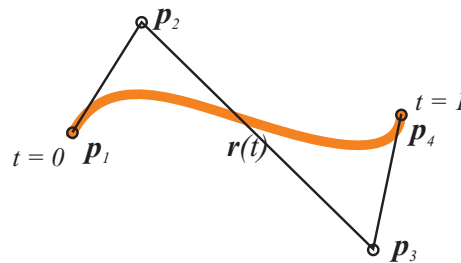
$$\mathbf{r}(t) = [t^3 \ t^2 \ t \ 1] \mathbf{B}_H^{-1} \mathbf{G}_H. \quad (4.7)$$

Če izračunamo inverz matrike  $\mathbf{B}_H$ , dobimo končno enačbo za Hermitsko krivuljo:

$$\mathbf{r}(t) = \begin{bmatrix} 2t^3 - 3t^2 + 1 \\ -2t^3 + 3t^2 \\ t^3 - 2t^2 + t \\ t^3 - t^2 \end{bmatrix} \begin{bmatrix} \mathbf{r}(0) \\ \mathbf{r}(1) \\ \frac{d\mathbf{r}(0)}{dt} \\ \frac{d\mathbf{r}(1)}{dt} \end{bmatrix}. \quad (4.8)$$

### 4.3.2 Bezierove krivulje

Kubična Bezierova krivulja je določena s štirimi kontrolnimi točkami, od katerih sta dve začetna in končna točka krivulje, drugi dve pa določata obliko krivulje (slika 4.7). Izkaže se, da lahko odvode Bezierove krivulje v začetni in končni točki



Slika 4.7: Krivulja, določena s štirimi točkami.

zapišemo kot

$$\frac{d\mathbf{r}(0)}{dt} = 3(\mathbf{p}_2 - \mathbf{p}_1), \quad \frac{d\mathbf{r}(1)}{dt} = 3(\mathbf{p}_4 - \mathbf{p}_3). \quad (4.9)$$

Iz teh dveh pogojev sledi, da lahko Bezierovo krivuljo zapišemo na podoben način kot smo prej Hermitsko.

$$\begin{bmatrix} \mathbf{r}(0) \\ \mathbf{r}(1) \\ \frac{d\mathbf{r}(0)}{dt} \\ \frac{d\mathbf{r}(1)}{dt} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \mathbf{p}_4 \end{bmatrix} \text{ ali } \mathbf{G}_H = \mathbf{H}\mathbf{P}. \quad (4.10)$$

Če v enačbi (4.7)  $\mathbf{G}_H$  nadomestimo z enačbo (4.10), dobimo

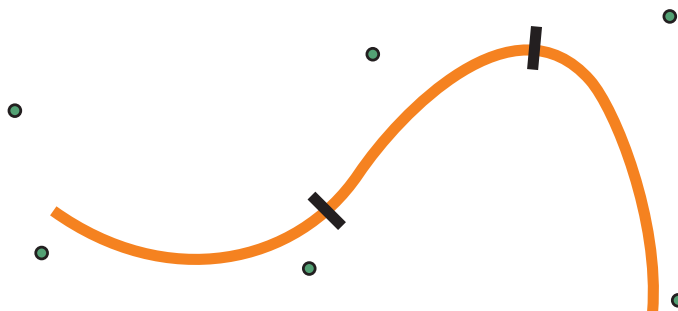
$$\mathbf{r}(t) = [t^3 \ t^2 \ t \ 1] \mathbf{B}_H^{-1} \mathbf{H}\mathbf{P}. \quad (4.11)$$

Če izraz izračunamo, dobimo končno enačbo za Bezierovo krivuljo:

$$\mathbf{r}(t) = \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \mathbf{p}_4 \end{bmatrix}. \quad (4.12)$$

Polinomom, ki definirajo Bezierove krivulje, pravimo Bernsteinovi polinomi, njihova značilnost pa je, da so na intervalu  $[0, 1]$  pozitivni, njihova vsota pa je enaka 1.

Slabost obeh tipov krivulj je, da moramo za bolj kompleksne krivulje povečati število kontrolnih točk in s tem red krivulj. Poleg tega sprememba v položaju ene kontrolne točke vpliva na celo krivuljo, torej sprememba ni lokalna. Zato pogosto uporabljamo zlepke takih krivulj, pri katerih v vozlih posameznih krivulj poskrbimo za gladkost določenega reda. Primer so B-zlepki, ki jih podrobneje opisujemo v poglavju 5.3. Slika 4.8 prikazuje primer zlepka, zlepljenega iz treh Bezierovih krivulj ter kontrolne točke. Več o zapisu krivulj z Bezierovimi in



Slika 4.8: Zlepek treh Bezierovih krivulj.

Hermitskimi krivuljami najdemo v [39].





## 5. Analiza vidne scene

Cilj poglavja je podrobno opisati sestavne dele rešitve za analizo in opis vidne scene na sliki. Podali bomo opis segmentacije s tenzorskim glasovanjem, ki ga v glavnem povzemamo po [4] in [10]. V podpoglavju bomo opisali tudi B-zlepke, katerih opis temelji na članku [5]. Predstavili bomo osnove teorije grafov, ki nam bo omogočila učinkovit in zgoščen opis slike. Teorijo grafov bomo povzeli po [6].

### 5.1 Segmentacija s tenzorskim glasovanjem

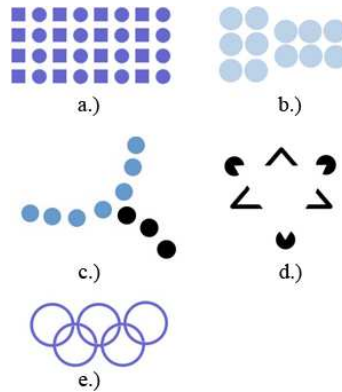
#### 5.1.1 Uvod

Načrtovalci sistemov računalniškega vida se že od začetka načrtovanja zgledujejo po človeškem vidu in skušajo iz človekovih opažanj sveta določiti omejitve, ki sodelujejo pri človekovem zaznavanju in razumevanju. Velik izziv predstavlja uporaba in implementacija teh omejitev pri računalniškem vidu. Ena izmed najbolj pogostih omejitev je strnjenost (angl. *continuation*), ki je izpeljana iz lastnosti, da je snov povezana (angl. *the matter is cohesive*). Ta omejitev predpostavlja, da so objekti v resničnih scenah skoraj povsod strnjeni. Večino problemov računalniškega vida lahko predstavimo kot probleme percepcijske organizacije (angl. *perceptual organization problems*), zato lahko omejitve poiščemo pri psihologih, ki se ukvarjajo s človekovim dojemanjem sveta. Že pred letom 1900 so Gestalt psihologi predstavili nekaj zakonov, imenovanih Gestalt zakoni, ki vodijo človeka pri njegovi zaznavi [40]. Ti zakoni so:

- zakon podobnosti (angl. *similarity*),
- zakon bližine (angl. *proximity*),
- zakon strnjenosti (angl. *continuation*),

- zakon zaprtosti (angl. *closure*) in
- zakon preprostosti (angl. *simplicity*).

Našteti zakoni oziroma omejitve sodelujejo pri nastanku izrazitih struktur iz opazovanih žetonov (slika 5.1). Predlaganih je bilo veliko metod, kako te omejitve



Slika 5.1: Gestalt zakoni: a.) objekte, ki so si podobni, združimo v stolpce, b.) objekte, ki so si blizu, združimo v skupino, c.) črtam sledimo po najbolj gladki poti, d.) združene objekte obravnavamo kot celoto, e.) kompleksne objekte dojemamo čimbolj preprosto - vidimo 5 krogov namesto več bolj zapletenih objektov.

uvesti v sisteme računalniškega vida. Ena izmed njih je tudi metoda tenzorskega glasovanja (angl. *tensor voting*), ki jo predstavljamo v nadaljevanju magistrske naloge. Metoda temelji na predstavitvi vhodnih podatkov v obliki tenzorjev in nelinearnem glasovanju za posredovanje informacije, ki jo nosijo tenzorji. Oba elementa skupaj zagotavljata enotno ogrodje za zaznavo različnih struktur, kot so krivulje, ploskve, regije in presečišča iz zašumljenih podatkov, ki jih dobimo z različnimi slikovnimi senzorji in/ali predobdelavo. Za ovrednotenje te metode je bilo razvitih več algoritmov kot so algoritmi za percepcijsko združevanje v 2-D in 3-D, stereo vid, združevanje na osnovi gibanja ter algoritmi za segmentacijo.

### Problem sistemov računalniškega vida

Vrednosti posameznih slikovnih elementov slike, ki nastane na slikovnem senzorju so odvisne od samega senzorja in postavitve scene. Z upoštevanjem zgoraj

našteti omejitve oziroma zakonov lahko sceno rekonstruiramo. V splošnem so variacije sosednjih slikovnih elementov majhne. Nenadna sprememba v intenziteti slikovnih elementov je posledica spremembe v sceni, ko nek slikovni element pripada drugi strukturi. Problem sistemov računalniškega vida je ravno detekcija teh sprememb v sliki oziroma zaznava percepcijsko pomembnih informacij iz zašumljenih podatkov. Primer takšnega problema je na primer detekcija robov z zaznavanjem sprememb v intenziteti sivinske slike ter povezovanje teh robov, primerjanje dveh slik s korelacijo ter izbira in interpolacija ustreznih slikovnih elementov pri stereo vidu.

### Zapis vhodnih podatkov

Ob upoštevanju zgoraj napisanega moramo uporabiti tak zapis vhodnih podatkov, ki bo zajel:

- izrazitost (angl. *saliency*) neke strukture oziroma verjetnost, da se je ta struktura res pojavila,
- ter geometrijske lastnosti te strukture, če se je pojavila.

Geometrijski objekti, ki so pomembni za opis 2-D struktur, torej struktur, ki nastopajo v sliki, so segmenti krivulj, presečišča in končne točke krivulj. Informacije, ki jih moramo opisati na ustrezen način, so torej orientacije, velikosti in položaj tangent krivulj ter položaj točk. Smiselno bi bilo, da bi velikost in orientacijo struktur zapisali z vektorjem, vendar kot bomo prikazali v nadaljevanju, na tak način ne moremo zapisati tipa strukture. Prednost metode tenzorskega glasovanja je ravno predstavitev podatkov na tak način, da lahko združimo informacije o orientaciji in izrazitosti vhodnih podatkov z informacijo o tipu strukture. Pri elementih, ki predstavljajo krivuljo, je izrazita le ena orientacija, pri elementih, ki pa predstavljajo diskontinuitete (končne točke, presečišča, ...), pa je izrazitih več orientacij. Vhodne podatke moramo torej zapisati na tak način, ki omogoča zapis več orientacij. Izkaže se, da je za to primerna predstavitev v obliki elipse, oziroma simetričnega tenzorja drugega reda, ki smo ga opisali v 3. poglavju. Oblika elipse pove informacijo o orientaciji strukture (podolgovata elipsa - ena orientacija, okrogla elipsa - več orientacij), velikost elipse pa izrazitost strukture.

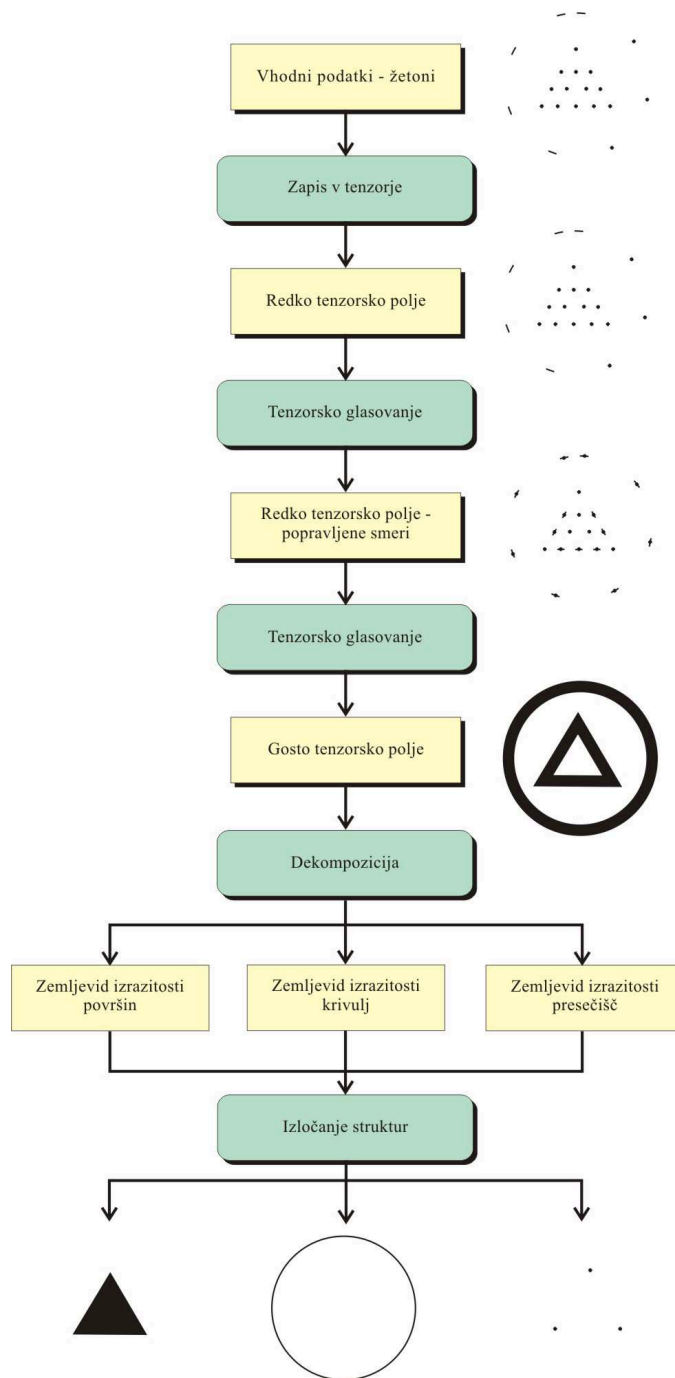
## Povzetek metode glasovanja tenzorjev

Povzetek delovanja metode je prikazan na sliki 5.2. Metoda sloni na dveh elementih, predstavitvi vhodnih podatkov s tenzorji in nelinearnem glasovanju tenzorjev za posredovanje informacije, ki jo nosijo. Vsak vhodni element posreduje informacijo, ki jo nosi, v okolico. Ta informacija je zapisana s tenzorjem, ki je odvisen od vnaprej definiranih tenzorskih glasovalnih polj. Vsaka točka zbira in analizira informacije, ki so jih posredovale ostale točke, ter gradi zemljevid izrazitosti (angl. *saliency map*) posameznih elementov struktur. Nato določimo izrazite strukture kot lokalne ekstreme v teh zemljevidih. Vhodne podatke, ki so lahko usmerjeni ali neusmerjeni žetoni, zapišemo s tenzorji ter dobimo redko tenzorsko polje. V prvi fazi tenzorskega glasovanja žetoni komunicirajo med seboj preko vnaprej definiranih polj in si izmenjujejo informacije. Na tak način izboljšajo informacijo, ki jo nosijo. Po tej fazi postane vsak žeton tenzor drugega reda, ki vsebuje podatke o tipu strukture, in o izrazitosti te strukture. V uvodu omenjene omejitve upoštevamo pri obliki glasovalnega polja. V drugi fazi žetoni preko tenzorskih polj poplavijo informacijo v okolico, kar vodi do gostega zemljevida izrazitosti struktur. Celotna domena prostora je razdeljena v posamezne celice in po končanem glasovanju vsaka celica nosi informacijo o tipu strukture, ki ji pripada. Glede na obliko in velikost tenzorja, lahko sklepamo na tip strukture, kateri pripada določen žeton oziroma celica.

Lastnosti metode tenzorskega glasovanja so neiterativnost, učinkovitost, zaznavanje različnih struktur istočasno, samo en prosti parameter (velikost glasovalnega polja) ter odsotnost kakršnegakoli kritičnega praga.

### 5.1.2 Zapis vhodnih podatkov

Naloga sistema računalniškega vida je opis opazovane scene z geometrijskimi strukturami, kot so krivulje in presečišča med njimi. Točke zapišemo le z njenimi koordinatami, krivuljo pa zapišemo s koordinatami točke in tangento na krivuljo v tej točki. Ker ne vemo vnaprej, kakšni strukturi pripada določena točka ali žeton, želimo vhodne žetone zapisati ne glede na tip strukture, ki ji pripada, skupaj z mero za izrazitost oziroma verjetnost, da ta struktura res obstaja. Potrebno je najti tak matematični objekt, ki bo omogočal zapis tipa strukture in njene



Slika 5.2: Potek tenzorskega glasovanja.

izrazitosti.

## Predstavitev vhodnih podatkov z vektorji

Če bi vedeli, kakšni strukturi pripada določen žeton, bi lahko za predstavitev vhodnih podatkov uporabili tenzor prvega reda, to je vektor. Delček krivulje v 2-D lahko predstavimo s točko  $\mathbf{p} = [x, y]^T$  in s tangento na krivuljo v tej točki, ki je enotin vektor  $\mathbf{t}_e = \mathbf{t}/\|\mathbf{t}\|$ . Dolžina vektorja  $\|\mathbf{t}\|$  pove izrazitost oziroma verjetnost pravilne ocene smeri tangente v točki  $P$ . Podobno lahko tudi v 3-D z normalo predstavimo ploskev. Omejitev take predstavitve se pojavi, ko želimo z vektorjem zapisati različne tipe struktur. Če bi želeli predstaviti neorientirano strukturo, kot je točka, z vektorjem, bi morali nastaviti  $\|\mathbf{t}\| = 0$ . Vendar bi s tem na 0 nastavili tudi izrazitost oziroma stopnjo zaupanja o tipu strukture. Torej ne bi mogli trditi “popolnoma smo prepričani da je točka  $\mathbf{p} = [x, y]^T$  neusmerjena”.

## Predstavitev vhodnih podatkov s tenzorji

Za predstavitev vhodnih podatkov uporabimo simetrični tenzor drugega reda, ki zajema tip strukture, usmerjenost in stopnjo zaupanja, da ta struktura res obstaja. Tak tenzor lahko grafično predstavimo kot elipso v 2-D, kjer velikost elipse ustreza velikosti tenzorja, glavni smeri elipse pa smerem tenzorja. Oblika elipse torej predstavlja tip strukture, velikost pa izrazitost strukture. Zelo izrazit element krivulje lahko v 2-D predstavimo s podolgovato elipso, katere smer glavne osi predstavlja ocenjeno smer tangente na krivuljo, dolžina glavne osi pa predstavlja izrazitost elementa krivulje. Tako elipso oziroma tenzor bomo poimenovali paličasti tenzor (angl. *stick tensor*). Neorientirano točko lahko predstavimo kot krog, njegov radij pa zopet predstavlja izrazitost točke. Tak tenzor bomo poimenovali okrogli tenzor (angl. *ball tensor*).

## Matematična predstavitev tenzorjev v 2-D

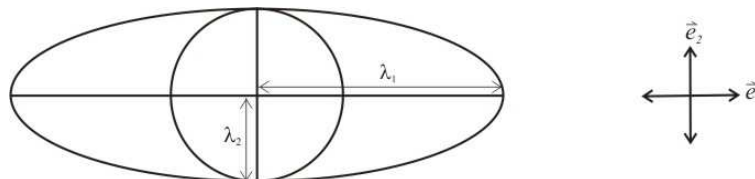
Simetrični tenzor drugega reda zapišemo z matriko velikosti 2x2. S spektralno analizo tenzor zapišemo kot

$$\mathbf{T} = [ \mathbf{e}_1 \quad \mathbf{e}_2 ] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^\top \\ \mathbf{e}_2^\top \end{bmatrix}, \quad (5.1)$$

ali kot

$$\mathbf{T} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^\top + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^\top, \quad (5.2)$$

kjer je  $\lambda_i$   $i$ -ta lastna vrednost matrike  $\mathbf{T}$  in  $\mathbf{e}_i$  njej pripadajoč lastni vektor. Ker je tenzor simetričen in pozitiven, sta lastni vrednosti realni in nenegativni, lastna vektorja pa tvorita ortonormirano bazo. Lastna vektorja predstavljajta glavni osi elipse, lastni vrednosti pa velikost in obliko (slika 5.3). Način zapisa vhodnih



Slika 5.3: Elipsa in njen lastni sistem.

žetonov, kot so točke in elementi krivulj, je prikazan v tabeli 5.1. Daljša os elipse je poravnana s smerjo tangente  $\mathbf{t}$ . Tak zapis vhodnih podatkov omogoča zapis različnih struktur na unikatni način.



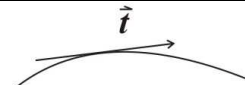

Struktura	Tenzor	Lastne vrednosti	Kovariančna matrika elipse
$\mathbf{P}$  točka neorientirana	 okrogli	$\lambda_1 = \lambda_2 = 1$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\vec{\mathbf{t}}$  krivulja orientirana	 paličasti	$\lambda_1 = 1, \lambda_2 = 0$ $\mathbf{e}_1 = \mathbf{t}$	$\begin{bmatrix} t_x^2 & t_x t_y \\ t_y t_x & t_y^2 \end{bmatrix}$

Tabela 5.1: Zapis vhodnih struktur s simetričnim tenzorjem drugega reda.

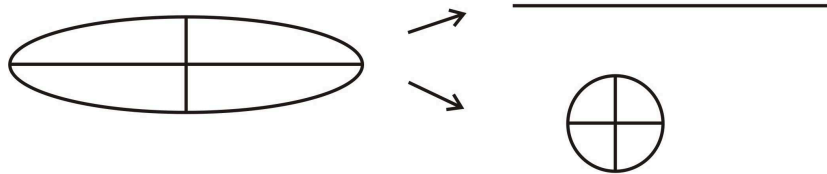
## Dekompozicija tenzorjev

Do sedaj smo predstavili posebne primere tenzorjev, kot sta okrogli in paličasti tenzor. Kot bomo pokazali v nadaljevanju, je rezultat glasovanja poljubni tenzor. Zato je potrebno kakršenkoli tenzor obravnavati kot kombinacijo zgoraj

omenjenih tenzorjev. Enačbo (5.2) lahko zapišemo tudi kot linearno kombinacijo prej omenjenih tenzorjev.

$$\mathbf{T} = (\lambda_1 - \lambda_2)\mathbf{e}_1\mathbf{e}_1^\top + \lambda_2(\mathbf{e}_1\mathbf{e}_1^\top + \mathbf{e}_2\mathbf{e}_2^\top) \quad (5.3)$$

Prvi del enačbe (5.3) ustreza paličastemu tenzorju in drugi del okroglemu tenzorju (slika 5.4).



Slika 5.4: Razstavljanje poljubnega tenzorja na paličasti in okrogli tenzor.

Na ta način lahko na vsaki lokaciji ocenimo tip strukture, ki jo opisuje tenzor:

- točka, usmerjenosti ni, izrazitost oz. stopnja zaupanja je  $\lambda_2$ ,
- krivulja, usmerjenost je  $\mathbf{e}_1$ , izrazitost oz. stopnja zaupanja je  $\lambda_1 - \lambda_2$ .

### 5.1.3 Glasovalna polja

V prejšnjem podpoglavju smo opisali predstavitev vhodnih podatkov s tenzorji. Ker so vhodni podatki redki in pogosto neorientirani, je potrebno v prvem koraku glasovanja vhodnim žetonom prirediti najbolj verjetno smer. To imenujemo redko glasovanje (angl. *sparse voting*). Nato moramo v vseh ostalih celicah, ki ne vsebujejo žetonov, izvesti ekstrapolacijo in določiti najbolj verjetne tenzorje, ki predstavljajo tip strukture v celicah. To je gosto glasovanje (angl. *dense voting*).

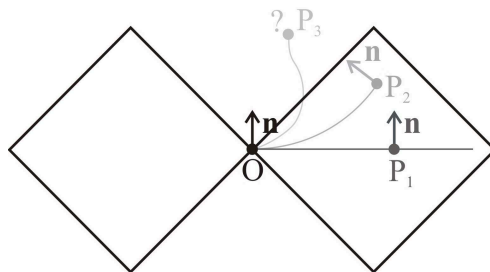
Definirajmo tenzorsko glasovalno jedro, ki upošteva omejitve, omenjene v uvodu. Jedro predstavimo kot območje s tenzorji v vsaki točki. V prvem koraku glasovanja žetoni zbirajo informacijo, ki jo ostali žetoni posredujejo preko glasovalnega polja. Rezultat je redko tenzorsko polje, kjer je vsak tenzor na določenem mestu vsota glasov, ki so jih na to mesto oddali ostali tenzorji. V drugem koraku se preko tenzorskega polja določijo tudi vrednosti v vseh ostalih točkah ravnine.



Če je vhodna struktura usmerjena, jo zapišemo s paličastim tenzorjem. Če pa je neusmerjena, jo zapišemo z okroglim tenzorjem (tabela 5.1). Tenzorji nato glasujejo z ustreznim poljem. V 2-D obstajata dve polji. Paličasti tenzorji glasujejo s paličastim poljem, okrogli pa z okroglim glasovalnim poljem.

### Paličasto glasovalno polje

Izpeljali bomo paličasto glasovalno polje, s katerim glasujejo paličasti tenzorji. Za lažjo predstavitev smo za usmerjenost struktur uporabili strukturi pripadajočo normalo. Če ima struktura (v tem primeru krivulja) v točki  $O$  normalo  $\mathbf{n}$  in pripada isti strukturi (npr. krivulji) kot žeton v točki  $P$ , kakšno informacijo mora posredovati žeton iz točke  $O$  žetonu v točki  $P$ ? Glede na Gestalt zakon o gladki strnjivosti, je najbolj verjetna pot iz točke  $O$  v točko  $P$  po loku kroga, ki ima v točki  $O$  enako normalo kot krivulja. Glas, ki ga tenzor, ki se nahaja v točki  $O$ , pusti v točki  $P$  je tudi tenzor, usmerjen pa je v smeri normale kroga, ki gre skozi točki  $O$  in  $P$ . Če je pot med točkama  $O$  in  $P$  ravna, se krog spremeni v premico (slika 5.5). Amplituda glasov mora z ukrivljenostjo upadati, ker imajo



Slika 5.5: Konstrukcija glasovalnega polja 2. reda.

ravne poti prednost pred ukrivljenimi, poleg tega pa žeton glasuje le za točke  $P$ , v katerih tangenta na krivuljo v tej točki in tangenta na krivuljo v točki  $O$  oklepata kot manjši od  $45^\circ$ . Amplituda glasov mora upadati tudi z oddaljenostjo od tenzorja, ki glasuje, zato da zadostimo zakonu bližine. Funkcija, po kateri glasovalno polje upada, je Gaussova funkcija

$$df(s, \kappa, \sigma_v) = e^{-\frac{s^2 + c\kappa^2}{\sigma_v^2}}, \quad (5.4)$$

kjer je  $s$  dolžina loka  $OP$ ,  $\kappa$  je ukrivljenost,  $c$  je stopnja upadanja glede na ukrivljenost,  $\sigma_v$  pa določa velikost glasovalnega polja in je edini prosti parameter

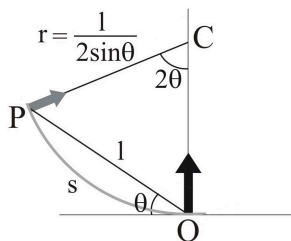
metode tenzorskega glasovanja. Vrednosti  $s$  in  $\kappa$  z uporabo geometrije in slike 5.6 izračunamo kot

$$s = r \cdot 2\theta = \frac{\theta l}{\sin \theta}, \quad (5.5)$$

$$\kappa = \frac{1}{r} = \frac{2 \sin \theta}{l}, \quad (5.6)$$

vrednost  $c$  pa smo empirično določili na

$$c = 0.02. \quad (5.7)$$



Slika 5.6: Konstrukcija glasovalnega polja 2. reda.

Tenzorje v paličastem polju izračunamo po enačbi

$$\mathbf{S}(s, \kappa, \sigma_v, \theta) = df(s, \kappa, \sigma_v) \begin{bmatrix} -\cos(2\theta) \\ -\sin(2\theta) \end{bmatrix} [-\cos(2\theta) - \sin(2\theta)] \quad (5.8)$$

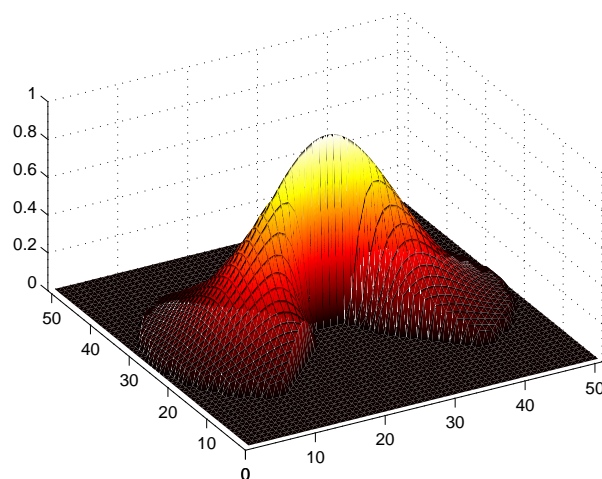
Velikosti tenzorjev v paličastem glasovalnem polju so prikazane na sliki 5.7, orientacije pa na sliki 5.8.

Katerokoli polje (tudi v več dimenzijah) lahko izpeljemo iz paličastega glasovalnega polja [4], [10].

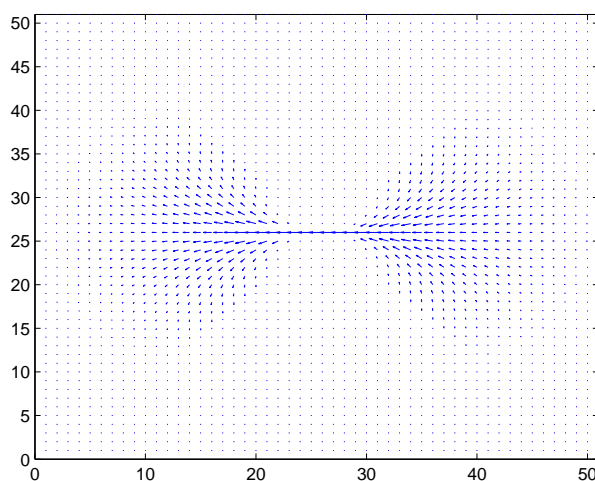
## Okroglo glasovalno polje

Okroglo 2-D glasovalno polje lahko izpeljemo z rotacijo in vsoto paličastih glasovalnih polj. Glas okroglega tenzorja v točki  $O$ , ki glasuje za točko  $P$ , je enak vsoti glasov paličastih tenzorjev, ki raztezajo celoten prostor vseh možnih smeri. V tem primeru je to enako rotaciji paličastih tenzorjev, s čimer opišemo krog. Tenzorji v okroglem glasovalnem polju imajo vrednost

$$\mathbf{B}(s, \kappa, \sigma_v, \theta) = \sum_{\phi=0}^{2\pi} \mathbf{S}_{\phi}(s, \kappa, \sigma_v, \theta), \quad (5.9)$$



Slika 5.7: Velikosti tenzorjev paličastega glasovalnega polja v 2-D.

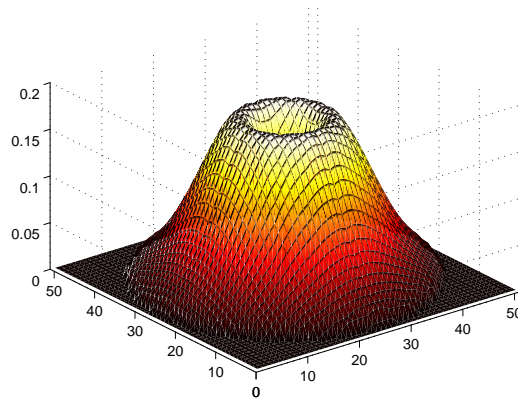


Slika 5.8: Orientacije tenzorjev paličastega glasovalnega polja v 2-D.

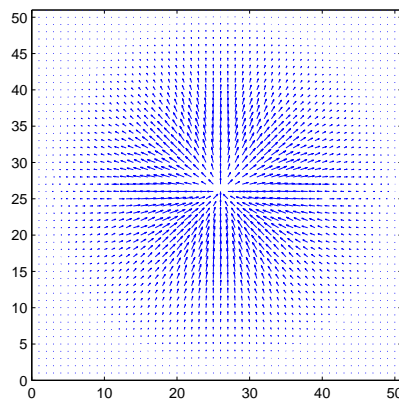
kjer je  $\mathbf{S}_\phi(s, \kappa, \sigma_v, \theta)$  vrednost paličastega tenzorja na mestu, ki ga določajo  $s$ ,  $\kappa$ ,  $\sigma_v$  in  $\theta$  v smeri  $\phi$ . Slika 5.9 prikazuje velikosti tenzorjev okroglega glasovalnega polja, slika 5.10 pa orientacije tenzorjev okroglega glasovalnega polja.

#### 5.1.4 Tenzorsko glasovanje

V nadaljevanju bomo opisali tenzorsko glasovanje v 2-D. Postopek je grafično prikazan na slikah 3.5 in 5.2.



Slika 5.9: Velikosti tenzorjev okroglega glasovalnega polja v 2-D.



Slika 5.10: Orientacije tenzorjev okroglega glasovalnega polja v 2-D.

### Tenzorsko glasovanje drugega reda

Glasovanje se najprej izvede le v položajih na sliki, ki so bili že na začetku zakodirani s tenzorji. Temu glasovanju pravimo redko tenzorsko glasovanje, med njim pa si žetoni izmenjujejo informacije, popravljajo začetne vrednosti tenzorjev in se usmerjajo v najbolj verjetni smeri. Usmerjeni žetoni, ki smo jih zapisali s paličastimi tenzorji, glasujejo s paličastim glasovalnim poljem, neusmerjeni, ki smo jih zapisali z okroglimi tenzorji, pa z okroglim glasovalnim poljem. V našem primeru smo žetone (slikovne elemente) v sliki zapisali z neorientiranimi žetoni, ker nismo imeli podane začetne usmerjenosti žetonov. Zato se pri redkem glasovanju izvede le glasovanje z okroglim glasovalnim poljem. Vrednost

tenzorskega glaslu paličastega tenzorja na določenem mestu je dana z vrednostjo enačbe (5.4) pomnoženo z velikostjo tenzorja, ki glasuje, medtem ko je vrednost glaslu okroglega tenzorja enaka vsoti glasov rotirajočih se paličastih tenzorjev, ki raztezajo celoten prostor vseh možnih smeri, pomnoženo z velikostjo tenzorja, ki glasuje. Velikost tenzorja na določenem mestu je za paličasti tenzor dana z  $\lambda_1 - \lambda_2$ , za okrogli pa z  $\lambda_2$ . Vsak tenzor odda glas drugim tenzorjem, ki tenzorske glasove seštevajo. V praksi to izvedemo z 2-D konvolucijo. Ko dobimo redko tenzorsko polje, tenzorje razstavimo na okroglo in paličasto komponento (enačba 5.3). Nato izvedemo še gsto tenzorsko glasovanje, pri katerem glasujejo le paličasti tenzorji preko paličastega glasovalnega polja. Pri tem se informacija prenese tudi na točke, ki na začetku niso bile zakodirane v tenzorje. Izvede se torej ekstrapolacija. Po gostem glasovanju se glasovi zopet seštevajo s tenzorskim seštevanjem. Rezultat tega je gsto tenzorsko polje, v katerem lahko po analizi glasov določimo različne strukture.

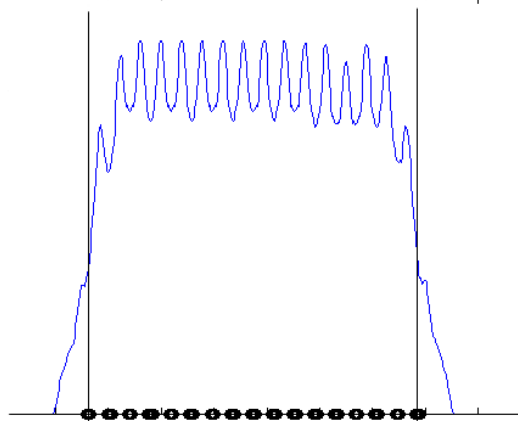
### Tenzorsko glasovanje prvega reda

Namen gostega glasovanja drugega reda je ekstrapolacija. Torej se informacija zapiše tudi na točke, v katerih na začetku ni bilo žetonov. Na sliki 5.11 je prikazan primer testne slike, na kateri bomo pojasnili potrebo po glasovanju prvega reda oz. glasovanju vektorjev. Na sliki 5.12 vidimo, da na podlagi glasov 2. reda ne



Slika 5.11: Neorientirani žetoni.

moremo natančno določiti končne točke neke krivulje. Zato moramo uporabiti tudi predstavitev žetonov s smernimi vektorji (angl. *polarity vectors*) ter tenzorsko glasovanje 1. reda ali vektorsko glasovanje. Izvedeno je na enak način, kot glasovanje drugega reda, torej z 2-D konvolucijo, kjer se za jedro uporabi glasovalno polje prvega reda. Rezultat tega glasovanja so smerni vektorji, ki povedo, iz katere smeri je prišla večina glasov. Za točko, ki je del neke krivulje ali regije, je smerni vektor enak 0, ker se bodo prispevki iz različnih smeri izičili. Za točko,

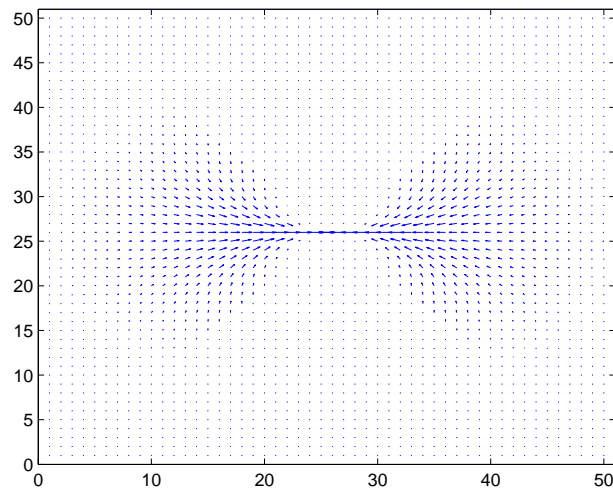


Slika 5.12: Prerez izrazitosti žetonov.

ki leži na koncu krivulje ali na robu regije, bo smerni vektor različen od 0, ker bodo glasovi prišli le z ene smeri. Vektorsko glasujejo le vhodni žetoni, vendar po procesu usmerjanja oz. popravljanja začetnih smeri, to je po redkem tenzorskem glasovanju. Za tenzorsko glasovanje 1. reda uporabimo enako paličasto glasovalno polje, kot pri glasovanju 2. reda, s tem da žetone predstavimo z vektorji namesto s tenzorji. Glasovalno polje izračunamo po naslednji enačbi

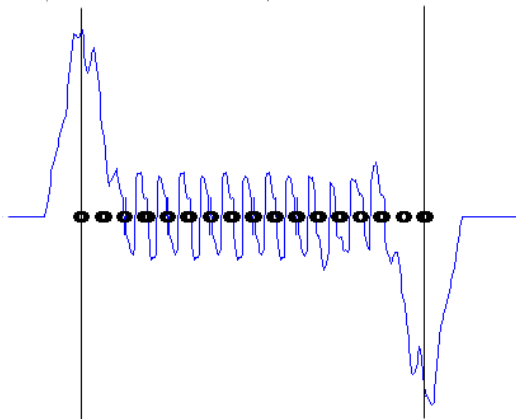
$$\mathbf{s}(s, \kappa, \theta) = df(s, \kappa, \theta) \begin{bmatrix} -\cos(2\theta) \\ -\sin(2\theta) \end{bmatrix}. \quad (5.10)$$

Velikost glasovalnega polja je enaka kot velikost glasovalnega polja drugega reda (slika (5.7)), smer vektorjev pa je prikazana na sliki 5.13. Med glasovanjem se



Slika 5.13: Smer vektorjev v glasovalnem polju prvega reda.

glasovi 1. reda seštevajo z vektorskim seštevanjem. Velikost glasov 1. reda je tudi dana z enačbo (5.4) in velikostjo smernega vektorja žetona, ki glasuje. Rezultat glasovanja prvega reda je prikazan na sliki 5.14, kjer je razvidno, da je velikost smernega vektorja na kočni točki krivulje maksimalna.



Slika 5.14: Prerez smernosti žetonov.

### 5.1.5 Analiza glasov

Ko izvedemo glasovanje, dobimo gosto tenzorsko polje, ki ga moramo analizirati. V vsaki diskretni celici vhodne slike imamo žeton, ki predstavlja tip strukture, ki se nahaja na tem mestu. Analizirati moramo tako glasove 1. kot tudi 2. reda. Rezultat analize glasov so zemljevidi izrazitosti posameznih struktur (slika 5.2).

#### Analiza glasov 2. reda

Glasove lahko analiziramo, ko izračunamo lastni sistem tenzorjev in jih razstavimo po enačbi (5.3). Če je  $\lambda_1 \gg \lambda_2$ , potem pripada žeton usmerjeni strukturi (paličasti tenzor), ki ima smer tangente enako  $\mathbf{e}_1$ . Če je  $\lambda_1 \simeq \lambda_2$ , potem je žeton neusmerjen (okrogli tenzor), bodisi zato, ker so vse smeri enako verjetne, bodisi zato, ker na tem mestu obstaja več smeri. Žeton, ki je neusmerjen, lahko pripada notranjosti regije, kjer se smeri sosedov izničijo, ali pripada presečišču dveh krivulj, kjer obstajata dve smeri, ali pa pripada zunanji točki, ki leži izven regij in ni del krivulje in ki je lahko posledica šuma. Med temi tremi primeri lahko ločimo glede na velikost lastnih vrednosti.

## Analiza glasov 1. reda

Glasovi 1. reda se seštevajo z vektorskim seštevanjem. Rezultat v neki točki je vektor, ki kaže v smer masnega centra žetonov, ki so glasovali za to točko. Ker tudi velikost glasov 1. reda upada z oddaljenostjo in ukrivljenostjo, kaže smerni vektor v smer, od koder so prišli najmočnejši glasovi. Nizka smernost pomeni žeton, ki je na krivulji ali v notranjosti ploskve, visoka smernost pa pomeni žeton, ki je na robu področja ali pa je končna točka krivulje. Meje področij in končne točke krivulj lahko določimo na podlagi maksimalne smernosti v smeri polarnega vektorja.

V tabeli 5.2 je prikazana analiza glasov ter lastnosti posameznih struktur. S pomočjo tabele naredimo zemljevide izrazitosti posameznih struktur, ki jih uporabimo za določanje struktur, kot so krivulje, ploskve in presečišča. S pomočjo

2-D struktura	Izrazitost	Smer tenzorja	Smernost	Smer
krivulja	velik $\lambda_1 - \lambda_2$	$\mathbf{e}_1$	nizka	-
k. t. krivulje	velik $\lambda_1 - \lambda_2$	$\mathbf{e}_1$	visoka	$\mathbf{e}_2$
regija	velik $\lambda_2$	-	nizka	-
meja regije	$\lambda_2$	-	visoka	$\perp$ na mejo
presečišče	lokalno max $\lambda_2$	-	nizka	-
zunanja točka	nizka	-	nizka	-

Tabela 5.2: Analiza glasov in lastnosti 2-D struktur.

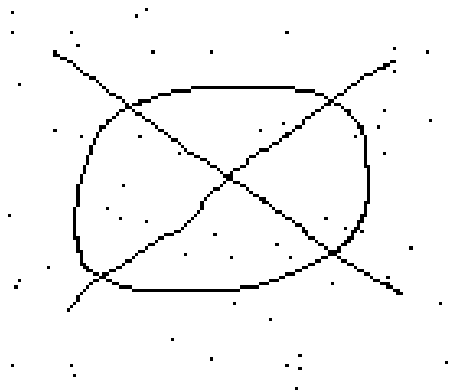
metode tenzorskega glasovanja lahko določimo tudi ploskve, vendar smo se v magistrski nalogi ukvarjali le z določanjem krivulj, presečišč in končnih točk krivulj.

## 5.2 Določanje posameznih struktur

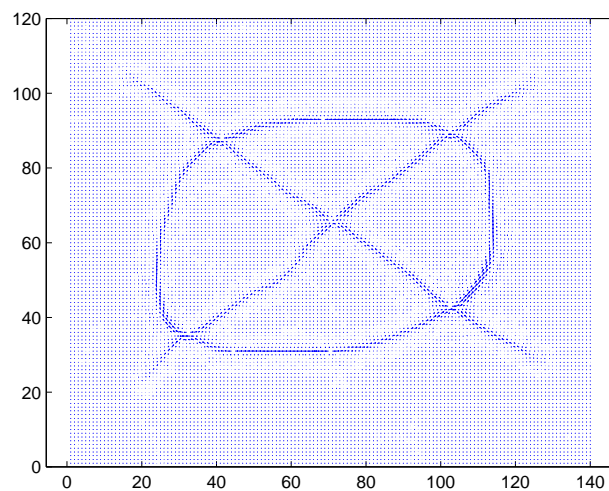
Zemljevidi izrazitosti posameznih struktur, ki jih dobimo po analizi glasov, so vektorska polja. To so vhodni podatki za algoritem, ki na podlagi ekstremnosti točk poišče presečišča in krivulje. Postopek algoritma je grafično prikazan na sliki 3.6. S pomočje slike 5.15 ga bomo v nadaljevanju predstavili ter prikazali delovanje tenzorskega glasovanja in določanje posameznih struktur. Rezultati tenzorskega glasovanja in dekompozicije, to so paličasto in okroglo tenzorsko polje



ter polje smernih vektorjev, so prikazani na slikah od 5.16 do 5.19. Paličasto tenzorsko polje je predstavljeno z vektorji, katerih smer ustreza smeri tenzorja. V okroglem tenzorskem polju je vsak tenzor predstavljen z dvema vektorjema, ki ustrezata glavnima smerema tenzorja. Ker se vektorji na slikah slabo vidijo, je na sliki 5.17 za lažjo predstavitev prikazan povečan odsek tenzorskega polja, predstavljenega z vektorji.



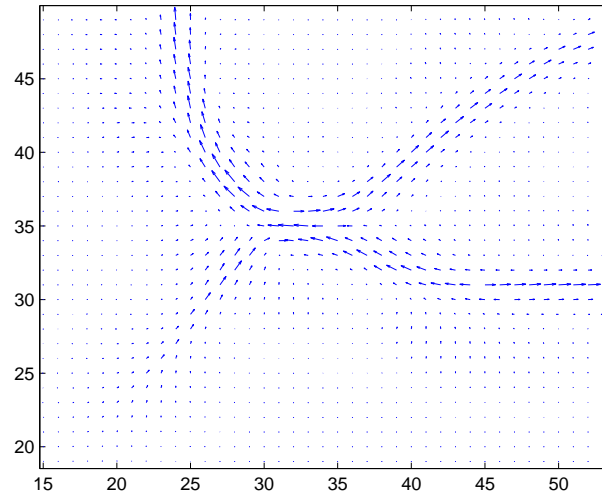
Slika 5.15: Testna slika za prikaz določanja presečišč in krivulj.



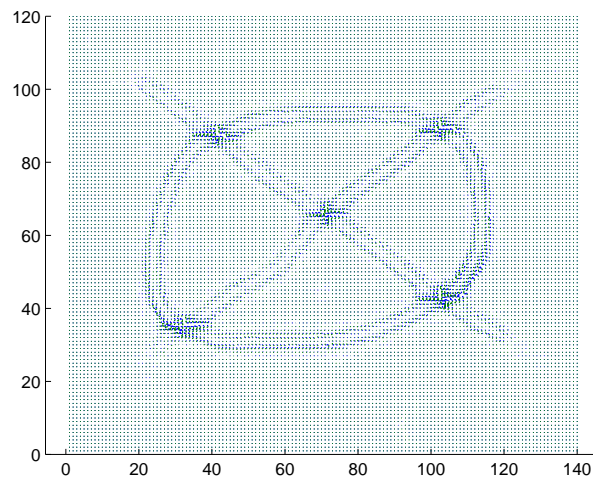
Slika 5.16: Paličasto tenzorsko polje, predstavljeno z vektorji.

### 5.2.1 Iskanje presečišč

Tabela 5.2 kaže na to, da je ekstremnost točk, ki nakazujejo presečišča, definirana kot lokalni maksimumi vrednosti  $\lambda_2$ . Tabela kaže tudi na to, da je smernost, ki



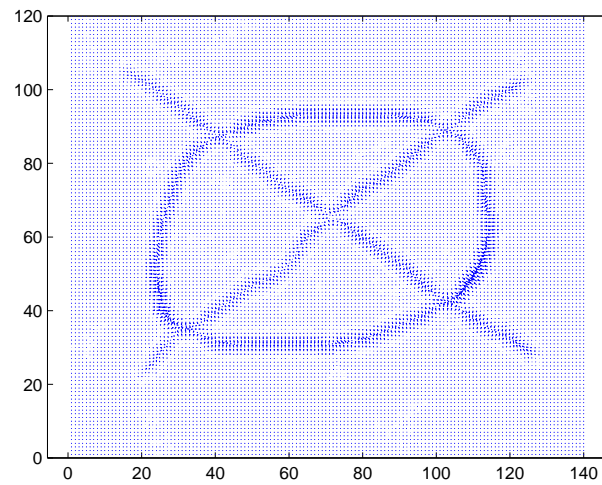
Slika 5.17: Povečan izsek slike 5.16.



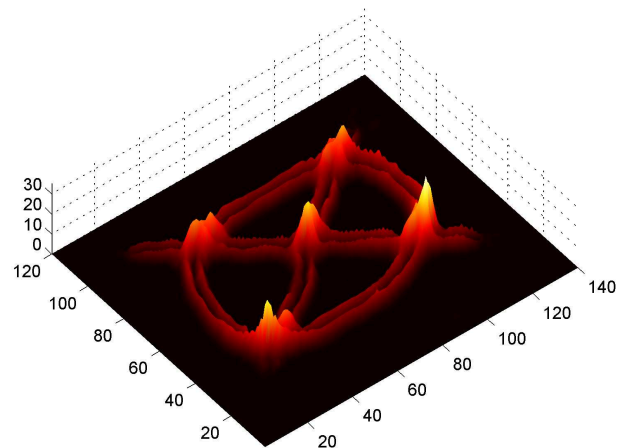
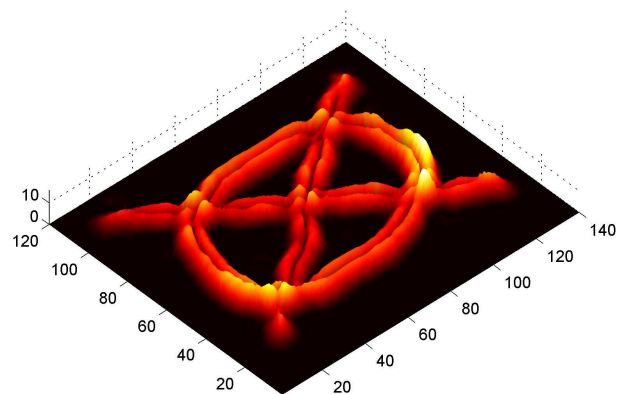
Slika 5.18: Okroglo tenzorsko polje, predstavljeno z vektorji.

jo označimo s črko  $s$ , v teh točkah nizka. Na sliki 5.20, ki prikazuje polje  $\lambda_2$ , je vidno, da lokalni maximumi ustrezajo presečiščem, nizke vrednosti pa ustrezajo šumu. Na sliki 5.21, ki prikazuje polje absolutnih vrednosti smernih vektorjev  $s$ , pa je razvidno, da presečiščom ustrezajo nizke vrednosti, končnim točkam krivulj pa maksimalne vrednosti smernosti v smeri krivulje. Obe polji smo dobili po tenzorskem glasovanju prvega in drugega reda ter dekompoziciji tenzorjev.

Presečišča imajo visoko vrednost  $\lambda_2$  in nizko vrednost  $s$ . Zato za iskanje



Slika 5.19: Polje smernih vektorjev.

Slika 5.20: Vrednosti  $\lambda_2$  po tenzorskem glasovanju in dekompoziciji.

Slika 5.21: Absolutne vrednosti smernih vektorjev po glasovanju 1. reda.

presečišč definiramo kompozitno mero

$$p = \alpha \frac{s}{\max(s)} + (1 - \alpha) \left(1 - \frac{\lambda_2}{\max(\lambda_2)}\right), \quad (5.11)$$

kjer je  $\alpha$  vrednost med 0 in 1,  $s$  je polje absolutnih vrednosti smernih vektorjev,  $\lambda_2$  pa je izrazitost točk. S primernim pragom, ki lahko zavzame vrednosti med 0 in 1, lahko določimo kandidate za presečišča. Pri implementaciji smo empirično določili  $\alpha = 0.5$ . Ker dobimo za posamezno presečišče več kandidatov, z uporabo nelinearnega uvrščevalnega filtra poiščemo točko z največjo vrednostjo  $p$  v okolici posameznega presečišča in jo shranimo kot presečišče.

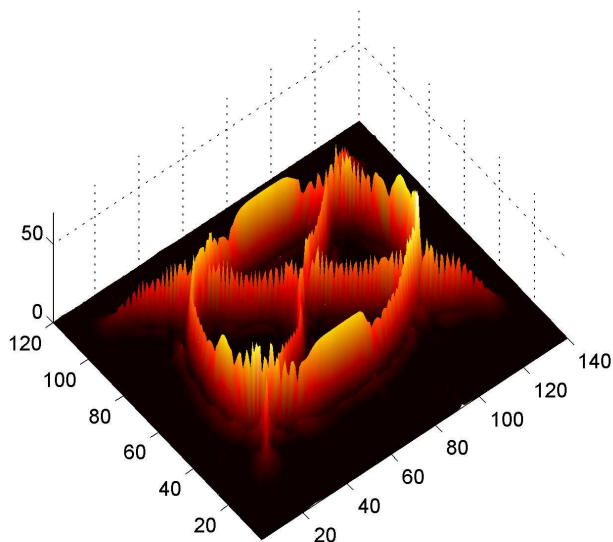
### 5.2.2 Sledenje krivuljam

Za opis slike moramo določiti še krivulje. Vsaka točka v vektorskem polju je predstavljena s parom  $(s, \mathbf{t})$ , kjer je  $s$  velikost polja na tem mestu,  $\mathbf{t}$  pa smer tangente na krivuljo. Po tabeli 5.2 so za iskanje krivulj pomembni

$$s = (\lambda_1 - \lambda_2) \text{ in}$$

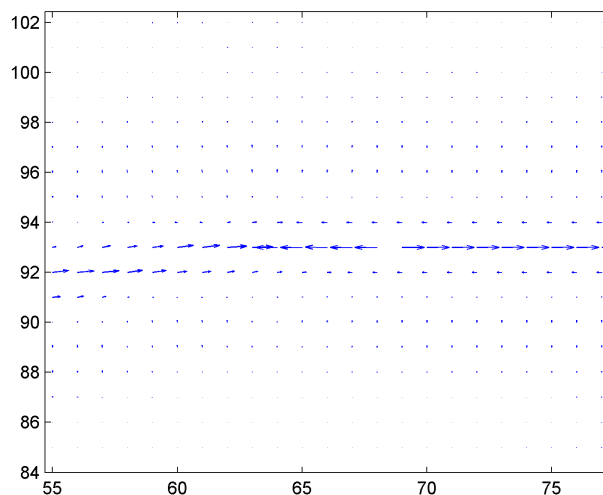
$$\mathbf{t} = \mathbf{e}_1,$$

kjer  $\mathbf{e}_1$  predstavlja zeleno orientacijo krivulje. Slika 5.22 prikazuje vrednosti  $\lambda_1 - \lambda_2$ . Točka  $s(s, \mathbf{t})$  je del maksimalne krivulje, če se katerikoli odmik od



Slika 5.22: Vrednosti  $\lambda_1 - \lambda_2$ .

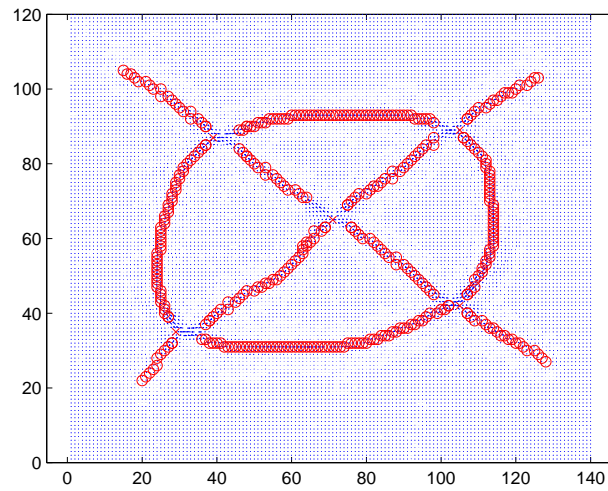
te točke na premici, ki je pravokotna na  $\mathbf{t}$ , odraža v nižji vrednosti  $s$ . To je potrební pogoj, da točka pripada krivulji. Iskanje krivulj je implementirano tako, da izberemo seme, to je točka  $s = \max(\lambda_1 - \lambda_2)$ , nato pa iz te točke potujemo v smeri  $\mathbf{t}$ . V naslednji točki preverimo, če je vrednost  $\lambda_1 - \lambda_2$  še dovolj velika. Če je, se premaknemo v to točko in postopek nadaljujemo. V vsaki točki preverimo ali smo v smeri pravokotno na  $\mathbf{t}$  na maksimalni vrednosti  $s$ . Če nismo, se nanjo premaknemo, nato pa nadaljujemo v smeri  $\mathbf{t}$ . Če pridemo do končne točke krivulje ali do presečišča, se vrnemo v začetno seme, in postopek ponovimo še v smeri  $-\mathbf{t}$ . Ta postopek nato ponovimo v naslednjem semenu, ki ima drugo največjo vrednost  $\lambda_1 - \lambda_2$ . Ker so vektorji v vektorskem polju v splošnem lahko obrnjeni v smeri  $\mathbf{t}$  ali  $-\mathbf{t}$  (slika 5.23), moramo pri sledenju krivulje to upoštevati. Sledenje krivuljam ponovimo za vsa semena, ki imajo dovolj visoko vrednost  $\lambda_1 - \lambda_2$ . Če se krivulja konča v enem izmed drugih presečišč, kar preverimo z Evklidovo razdaljo med vsako točko krivulje in vsemi presečišči, potem krivuljo zaključimo. Če je vrednost  $\lambda_1 - \lambda_2$  padla pod določen prag, potem sklepamo da smo prišli do končne točke krivulje. Poiskati moramo še natančen položaj končne točke,



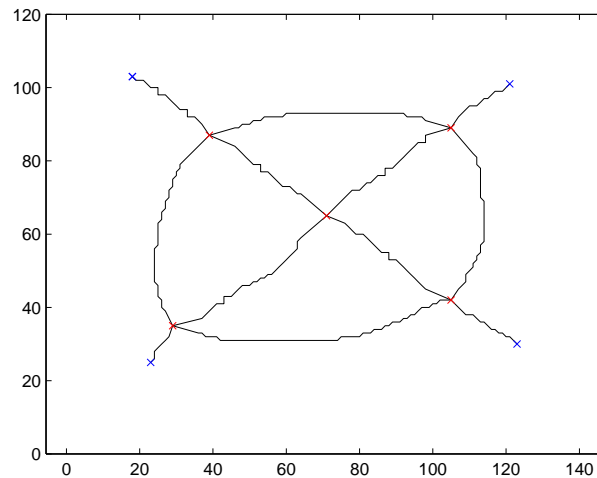
Slika 5.23: Izrez vektorskega polja, ki prikazuje različne smeri vektorjev.

ki je po tabeli 5.2 definirana kot maksimum smernosti v smeri krivulje. Torej moramo med vsemi točkami, ki sestavljajo krivuljo, poiskati tisto, ki ima maksimalno smernost. Sledenje krivuljam smo izvajali z natančnostjo enega slikovnega elementa. Preizkusili smo tudi sledenje z natančnostjo večjo od enega slikovnega elementa, kar smo dosegli z interpolacijo, vendar je bila izboljšava krivulj pre-

majhna, da bi opravičili za en velikostni čas daljše določanje krivulj. Slika 5.24 prikazuje točke, ki pripadajo krivuljam, slika 5.25 pa določene krivulje skupaj s presečišči, ki so na sliki označena z rdečimi križci ter končnimi točkami, ki so označene z modrimi križci.



Slika 5.24: Točke, ki pripadajo krivuljam.



Slika 5.25: Krivulje, presečišča in končne točke.

## 5.3 Zapis z B-zlepki

V tem poglavju bomo podali osnoven opis B-zlepkov in opisali algoritem za prileganje B-zlepkov ter možnost uporabe na krivuljah, določenih s tenzorskim glasovanjem.

### 5.3.1 Predstavitev B-zlepkov

Zlepki so parametrični zapis krivulje  $\mathbf{r}(t) = [x(t), y(t)]^\top$ .  $t$  je parameter, ki se s premikanjem po krivulji spreminja,  $x$  in  $y$  pa sta koordinati, ki sta funkciji  $t$ . Zlepek reda  $k$  je odsekoma polinomska funkcija, sestavljena iz polinomskih segmentov, ki so združeni v vozlih. Vsak segment je reda  $k$ . Red zlepkov je ponavadi kvadratičen ( $k = 3$ ) ali kubičen ( $k = 4$ ). Polinom  $p(x) = a + bx + cx^2$  je reda  $k = 3$ , stopnja polinoma pa je najvišja potenca  $x$ , torej 2, v splošnem  $k - 1$ . V delu bomo predstavili B-zlepke, ki so optimalna predstavitev zlepkov s stališča minimalne podpore pri maksimalni gladkosti. Lastnosti B-zlepkov so [5]:

- prostorska edinstvenost (angl. *spatial uniqueness*),
- omejenost (angl. *boundedness*),
- nepretrganost (angl. *continuity*),
- lokalno vplivanje na obliko (angl. *local shape controllability*) in
- neodvisnost od afinih transformacij (angl. *invariance to affine transformations*).

B-zlepek predstavimo kot uteženo vsoto baznih funkcij, vsaka bazna funkcija pa je sestavljena iz  $k$  polinomov, spojenih v vozlih. Polinomi so definirani na določenem razponu  $t$ . V magistrski nalogi bomo uporabljali kubične uniforme B-zlepke reda  $k = 4$ . Čeprav so tudi B-zlepki reda  $k = 3$  za našo uporabo dovolj gladki, smo zlepke reda  $k = 4$  uporabili za rezervo pri nadaljnjem delu.

### 5.3.2 Uniformni kubični B-zlepki

#### B-zlepki

Krivuljo lahko predstavimo z B-zlepkom  $k$ -tega reda. B-zlepek predstavimo z  $n + 1$  kontrolnimi točkami  $\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n$ . Celotno krivuljo  $\mathbf{r}(t)$  zapišemo kot

$$\mathbf{r}(t) = \sum_{i=0}^n \mathbf{r}_i(t-i) = \sum_{i=0}^{n+k-1} \mathbf{c}_{i-1} Q_{i,k}(t), \quad (5.12)$$

kjer je  $0 \leq t \leq n + 1$ ,  $Q_{i,k}(t)$  pa je bazna funkcija  $k$ -tega reda. Od tu naprej upoštevamo  $k = 4$ .  $\mathbf{r}_i(t-i)$  je različen od 0 samo na intervalu  $i \leq t \leq i + 1$ . Manjkajoče vrednosti  $\mathbf{c}_i$  za sklenjene krivulje določimo po modulu  $n + 1$

$$\mathbf{c}_{-1} = \mathbf{c}_n, \quad (5.13)$$

$$\mathbf{c}_{n+k-3} = \mathbf{c}_0 \text{ in}$$

$$\mathbf{c}_{n+k-2} = \mathbf{c}_1,$$

za odprte krivulje pa jih izračunamo

$$\mathbf{c}_{-1} = 2\mathbf{c}_0 - \mathbf{c}_1, \quad (5.14)$$

$$\mathbf{c}_{n+k-3} = 2\mathbf{c}_{n+k-4} - \mathbf{c}_{n+k-5} \text{ in}$$

$$\mathbf{c}_{n+k-2} = 2\mathbf{c}_{n+k-3} - \mathbf{c}_{n+k-4}.$$

Manjkajoče vrednosti  $\mathbf{c}_i$  so izračunane na način, da se krivulja začne v  $\mathbf{c}_0$  in konča v  $\mathbf{c}_n$ . Enačbo (5.12) lahko zapišemo tudi v matrični obliki:

$$\mathbf{r}(t) = \begin{bmatrix} Q_0(t) & Q_1(t) & \dots & Q_{n+k-2}(t) & Q_{n+k-1}(t) \end{bmatrix} \begin{bmatrix} \mathbf{c}_{-1} \\ \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_{n+k-3} \\ \mathbf{c}_{n+k-2} \end{bmatrix} \quad (5.15)$$

#### Bazne funkcije

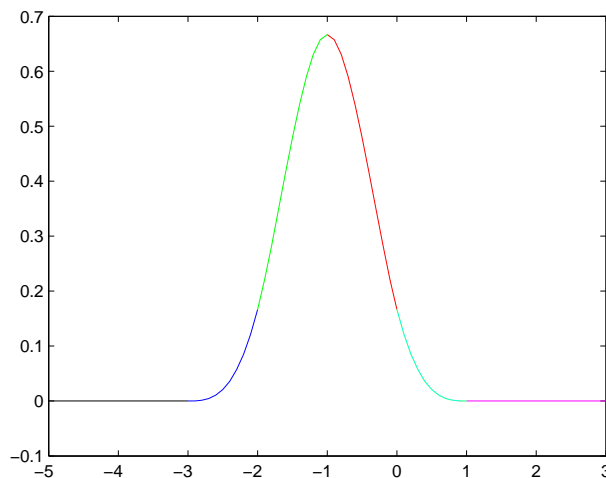
Bazne funkcije  $Q_{i,k}(t)$  so odvisne od položaja vozlov na parametrizacijskem intervalu  $T_i \in [a, b]$ , pri katerih so spojeni polinomi, ki sestavljajo bazno funkcijo. V



primeru uniformnih B-zlepkih, kjer so vozli med seboj razmaknjeni za enako vrednost  $\Delta t = 1$ , so vrednosti  $T_i = \{-k + 1, \dots, 0, \dots, n + k\}$ . Bazne funkcije konstruiramo glede na izbran red  $k$ , poiščemo pa jih lahko z eksplcitnim reševanjem enačb ali de Boorovim algoritmom [41]. Bazne funkcije za  $k = 4$  so

$$Q_{i,4}(t) = \begin{cases} \frac{1}{6}(t-i+3)^3 & -3 \leq t < -2 \\ \frac{2}{3} - 2(t-i+3) + 2(t-i+3)^2 - \frac{1}{2}(t-i+3)^3 & -2 \leq t < -1 \\ -\frac{22}{3} + 10(t-i+3) - 4(t-i+3)^2 + \frac{1}{2}(t-i+3)^3 & -1 \leq t < 0 \\ \frac{32}{3} - 8(t-i+3) + 2(t-i+3)^2 - \frac{1}{6}(t-i+3)^3 & 0 \leq t < 1 \\ 0 & \text{drugje} \end{cases} \quad (5.16)$$

Na sliki 5.26 je prikazana bazna funkcija  $Q_{0,4}(t)$ . Zaradi uniformnosti B-zlepkih



Slika 5.26: Bazna funkcija  $Q_{0,4}(t)$

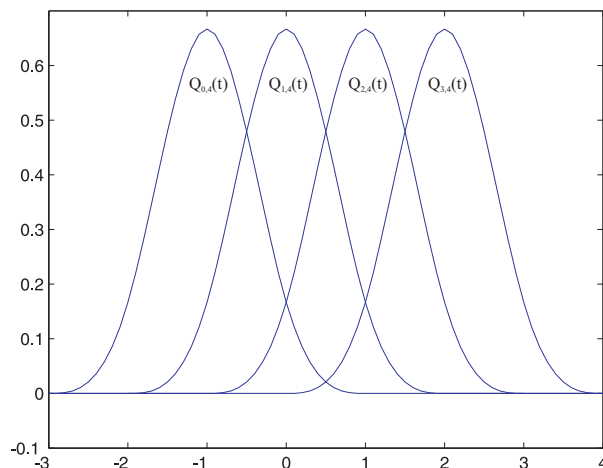
so ostale bazne funkcije le premaknjene kopije funkcije  $Q_{0,4}(t)$ . Zapišemo lahko

$$Q_{i,k}(t) = Q_{0,k}(t - i). \quad (5.17)$$

Na sliki 5.27 so prikazane bazne funkcije  $Q_{i,4}(t)$ .

Prostorske točke, ki jih obiše krivulja (5.12) pri vozlih (ko je parameter eden od vozlov) imenujemo vozne točke (ang. knot points)  $\mathbf{s}_i = \mathbf{r}(T_i)$ ,  $i = 0, \dots, n$ . Povezava med vektorjem vozelnih točk  $\mathbf{s} = (\mathbf{s}_0, \dots, \mathbf{s}_n)$  in vektorjem kontrolnih točk  $\mathbf{c} = (\mathbf{c}_0, \dots, \mathbf{c}_n)$  je linearna,

$$\mathbf{s} = \mathbf{A}\mathbf{c}, \quad (5.18)$$



Slika 5.27: Kubične bazne funkcije

kjer je  $\mathbf{A}$  kvadratna matrika, določimo jo neposredno iz baznih B-zlepkov.

$$\mathbf{A} = \begin{bmatrix} Q_0(T_0) & Q_1(T_0) & \dots & Q_{n+k-2}(T_0) & Q_{n+k-1}(T_0) \\ Q_0(T_1) & Q_1(T_1) & \dots & Q_{n+k-2}(T_1) & Q_{n+k-1}(T_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_0(T_n) & Q_1(T_n) & \dots & Q_{n+k-2}(T_n) & Q_{n+k-1}(T_n) \end{bmatrix} \quad (5.19)$$

### 5.3.3 Ocenjevanje kontrolnih točk B-zlepkov iz podatkovnih točk

Krivuljo zapišemo kot nabor  $m$  točk, kjer je  $m > n + 1$

$$\mathbf{r} = [\mathbf{r}_1 \dots \mathbf{r}_m] = \begin{bmatrix} x_1 & \dots & x_m \\ y_1 & \dots & y_m \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \quad (5.20)$$

Tej krivulji želimo prilagoditi B-zlepek, torej moramo poiskati nabor kontrolnih točk  $\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n$ . Točkam  $\mathbf{r}$  prilagodimo tak zlepek, da bo napaka med originalno krivuljo in zlepkom čim manjša. Problem je optimizacijske narave, ena od njegovih rešitev vodi do MMSE (angl. *minimum mean square estimation*) algoritma [42]. To je postopek, ki minimizira srednjo kvadratično napako in je pogosta mera za kakovost. V članku [5] je opisan postopek, ki temelji na iskanju MMSE rešitve in na izboljšavi parametričnih vrednosti  $t_1, \dots, t_m$ . Ker je naš cilj opis vizualne scene z množico kontrolnih točk, ne pa natančno modeliranje regij objektov, algoritem za ocenjevanje kontrolnih točk priredimo in

izboljšavo parametričnih vrednosti izpustimo ter uporabimo le prvi MMSE približek. Parametrične vrednosti izračunamo na podlagi ločne dolžine po enačbi

$$t_j = t_{j-1} + \|\mathbf{r}_j - \mathbf{r}_{j-1}\| \frac{t_{max}}{l}, \quad (5.21)$$

kjer je  $t_1 = 0$ ,  $t_{max} = n$  in

$$l = \sum_{j=2}^m \|\mathbf{r}_j - \mathbf{r}_{j-1}\|. \quad (5.22)$$

Potrebno je omeniti, da enačbi (5.21) in (5.22) veljata le za nesklenjene krivulje, medtem ko za sklenjene namesto  $n$  in  $m$  uporabimo  $n + 1$  in  $m + 1$ . Rešitev MMSE problema nam za izračun kontrolnih točk da naslednjo enačbo

$$\mathbf{c}_f = \mathbf{Q}^\dagger \mathbf{r}_f = (\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{Q}^T \mathbf{r}_f, \quad (5.23)$$

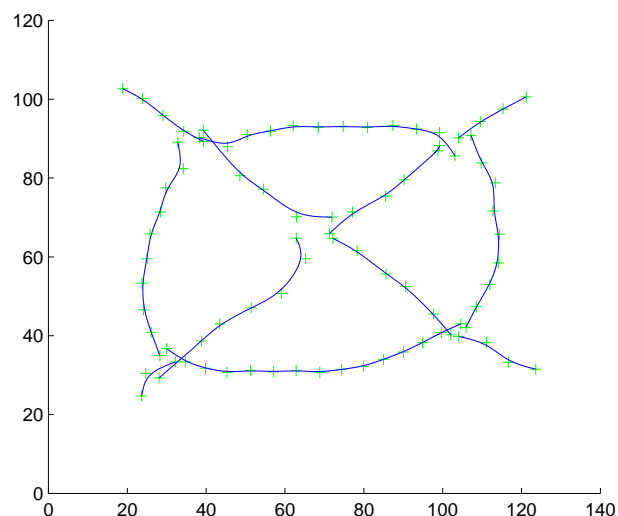
kjer je  $f = x$  ali  $f = y$ , s  $\mathbf{Q}^\dagger$  pa označimo Moore-Penrose-ov psevdoinverz matrike  $\mathbf{Q}$  [43]. Matriko  $\mathbf{Q}$  izračunamo z vstavljanjem parametričnih vrednosti v bazne funkcije.

$$\mathbf{Q} = \begin{bmatrix} Q_0(t_0) & Q_1(t_0) & \dots & Q_{n+k-2}(t_0) & Q_{n+k-1}(t_0) \\ Q_0(t_1) & Q_1(t_1) & \dots & Q_{n+k-2}(t_1) & Q_{n+k-1}(t_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_0(t_n) & Q_1(t_n) & \dots & Q_{n+k-2}(t_n) & Q_{n+k-1}(t_n) \end{bmatrix} \quad (5.24)$$

Vsaki krivulji, ki je rezultat tenzorskega glasovanja, lahko torej z zgoraj opisanim postopkom približamo B-zlepek. Slika (5.28) prikazuje kontrolne točke testne slike, ter B-zlepke, ki jih določajo. Kontrolne točke so označene z zelenimi plusi, B-zlepki pa z modro krivuljo.

## 5.4 Predstavitev scene z vloženim grafom

V poglavju bomo podali nekaj osnovnih pojmov o grafih in opisali njihovo uporabo pri opisu kompleksne vidne scene. Opis teorije grafov povzemamo po [6] in [44].



Slika 5.28: Kontrolne točke in B-zlepki testne slike.

### 5.4.1 Grafi

Graf je matematična struktura za modeliranje relacij med določenimi objekti. Zapišemo ga kot množico točk ali vozlišč, ki so povezana z množico povezav. Graf  $G$  je urejen par

$$G = (V, E), \quad (5.25)$$

kjer je  $V = \{v_1, \dots, v_n\}$  neprazna, končna množica vozlišč grafa in  $E = \{e_1, \dots, e_m\}$  množica povezav ali robov med vozlišči grafa, kjer je  $e_i = (v_k, v_l)$ ,  $v_k, v_l \in V$ . Graf je lahko neusmerjen, kar pomeni, da je povezava med A in B ista kot povezava med B in A. Graf je lahko usmerjen, kjer se različno usmerjeni povezavi med dvema točkama smatrata za različni. Takemu grafu pravimo tudi digraf. Grafe lahko ponazorimo grafično z diagrami. Za vsako vozlišče grafa narišemo točko ter povezavo med točkami, če ta obstaja. Če je graf usmerjen, smer povezave označimo s puščico. Posamezni povezavi lahko pripišemo tudi določeno utež ali oznako. V nalogi se bomo osredotočili na neusmerjene grafe, zato bomo v nadaljevanju opis skrčili le na usmerjene grafe.

Grafe lahko predstavimo tudi z različnimi matematičnimi strukturami, ki so primerne za zgoščen zapis podatkov. Nekateri načini predstavitve grafov so:

- zapis z incidenčno matriko,
- zapis z matriko sosednosti,

- zapis z incidenčnim seznamom,
- zapis s seznamom sosednosti.

Incidenčna matrika neusmerjenega grafa  $G$  je matrika, ki ima za vsako vozlišče eno vrstico in za vsako povezavo en stolpec:

$$\mathbf{A} = (a_{v,e})_{v \in V(G), e \in E(G)} \quad (5.26)$$

$$a_{i,j} = \begin{cases} 1; & v_i \in e_j \\ 0; & v_i \notin e_j \end{cases} \quad (5.27)$$

Matrika sosednosti neusmerjenega grafa  $G$  je kvadratna matrika, ki ima za vsako vozlišče eno vrstico in en stolpec:

$$\mathbf{A} = (a_{v,w})_{v,w \in V(G)} \quad (5.28)$$

$$a_{i,j} = \begin{cases} 1; & (v_i, v_j) \in E(G) \\ 0; & (v_i, v_j) \notin E(G) \end{cases} \quad (5.29)$$

Zapis grafov z matrikami je primeren za goste grafe, medtem ko je za zapis redkih grafov bolj primeren zapis s seznamami. Incidenčni seznam je seznam, kjer vsakemu vozlišču pripišemo množico povezav, katerih krajišče je. Tak seznam lahko zapišemo v obliki tabele:

$$\begin{array}{c|c} v_1 & e_j; v_1 \in e_j, \\ \vdots & \vdots \\ v_n & e_j; v_n \in e_j, \end{array} \quad (5.30)$$

Seznam sosednosti je seznam, kjer vsakemu vozlišču pripišemo množico vozlišč, s katerimi je sosedni oziroma povezan. Tak seznam zapišemo kot

$$\begin{array}{c|c} v_1 & v_j; (v_1, v_j) \in E(G), \\ \vdots & \vdots \\ v_n & v_j; (v_n, v_j) \in E(G), \end{array} \quad (5.31)$$

Slabost zapisa s seznamom sosednosti je redundanca, ker se sosednost dveh vozlišč v seznamu ponovi. Zato bomo v delu za zapis grafov uporabili incidenčni seznam.

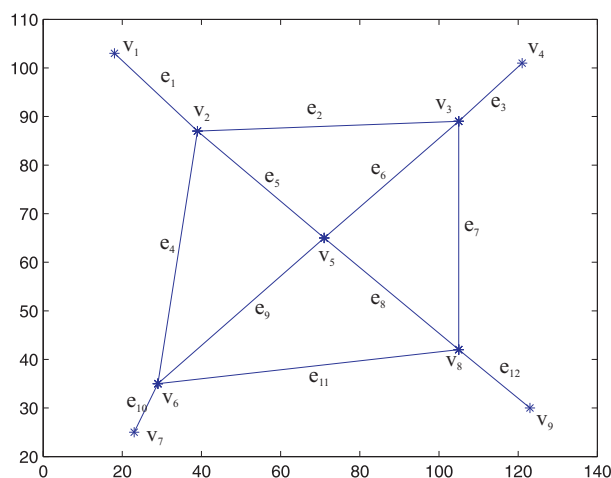
### 5.4.2 Opis scene z vloženim grafom

Rezultati tenzorskega glasovanja in analize dobljenih polj nad vidno sceno so presečišča, krivulje in mejne točke krivulj. Zaradi tega lahko vidno sceno opišemo z grafom, kjer so vozlišča grafa presečišča in končne točke krivulj, povezave pa krivulje med temi točkami. Ker sama vozlišča in povezave v grafu ne morejo enolično opisati scene, je potrebno vozlišča predstaviti s koordinatami v  $\mathbb{R}^2$  prostoru (vložitev grafa), posamezni povezavi pa prirediti labelo oziroma oznako, ki pove, kateri krivulji ta povezava pripada. To pomeni, da topološko informacijo o vidni sceni podamo z vložitvijo grafa in labeliranjem povezav s kontrolnimi točkami B-zlepkov. Definirajmo preslikavi

$$\phi : V(G) \rightarrow \mathbb{R}^2, \quad (5.32)$$

$$\psi : E(G) \rightarrow \Gamma, \quad (5.33)$$

kjer je  $\Gamma$  množica label, ki so v našem primeru kar vektorji kontrolnih točk B-zlepkov, ki predstavljajo krivuljo te povezave. Graf gradimo sproti med iskanjem presečišč, končnih točk in krivulj. Celoten rezultat opisa vidne scene so torej koordinate presečišč in končnih točk, kontrolne točke B-zlepkov za vsako določeno krivuljo ter vložen graf. Tak opis scene je nato primeren za višje nivojske algoritme za razpoznavo vizualne scene. Slika (5.29) prikazuje rezultat opisa testne slike z grafom. Graf opišemo z incidenčnim seznamom ter preslikavami  $\phi$  in  $\psi$ :



Slika 5.29: Diagram grafa testne slike.

$v_1$	$e_1$				
$v_2$	$e_1$	$e_2$	$e_4$	$e_5$	
$v_3$	$e_2$	$e_3$	$e_6$	$e_7$	
$v_4$	$e_3$				
$v_5$	$e_5$	$e_6$	$e_8$	$e_9$	
$v_6$	$e_4$	$e_9$	$e_{10}$	$e_{11}$	
$v_7$	$e_{10}$				
$v_8$	$e_7$	$e_8$	$e_{11}$	$e_{12}$	
$v_9$	$e_{12}$				

$$\begin{aligned} \phi : v_1 \mapsto (18, 103), v_2 \mapsto (39, 87) v_3 \mapsto (105, 89), v_4 \mapsto (121, 101), v_5 \mapsto (71, 65) \\ v_6 \mapsto (29, 35), v_7 \mapsto (23, 25), v_8 \mapsto (105, 42), v_9 \mapsto (123, 30) \end{aligned}$$

$$\begin{aligned} \psi : e_1 \mapsto \mathbf{C}_1, e_2 \mapsto \mathbf{C}_2, e_3 \mapsto \mathbf{C}_3, e_4 \mapsto \mathbf{C}_4, e_5 \mapsto \mathbf{C}_5, e_6 \mapsto \mathbf{C}_6, \\ e_7 \mapsto \mathbf{C}_7, e_8 \mapsto \mathbf{C}_8, e_9 \mapsto \mathbf{C}_9, e_{10} \mapsto \mathbf{C}_{10}, e_{11} \mapsto \mathbf{C}_{11}, e_{12} \mapsto \mathbf{C}_{12}, \end{aligned}$$

kjer so  $\mathbf{C}_1, \dots, \mathbf{C}_{12}$  matrice kontrolnih točk posameznega B-zlepka.





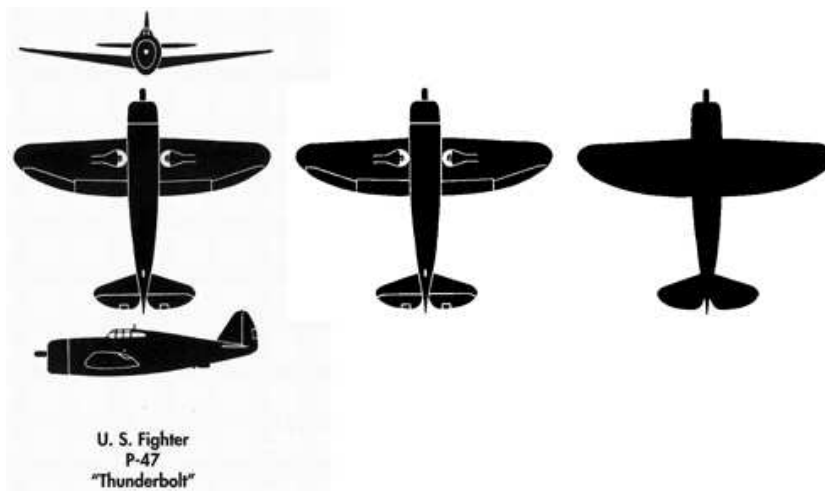
## 6. Eksperimentalni rezultati

V tem poglavju bomo podali eksperimentalne rezultate metode tenzorskega glasovanja. Ker smo želeli preveriti odpornost metode tenzorskega glasovanja na šum v slikah, smo se pri testiranju osredotočili na dodajanje šuma testnim slikam. Pri tem smo opazovali, kako uspešna je metoda pri določanju krivulj iz vhodne slike. Z večanjem šuma v vhodnih slikah smo večali tudi velikost glasovalnih polj ter opazovali vpliv na čas trajanja tenzorskega glasovanja in določanja krivulj, ob tem pa spremljali, kako dobro lahko določimo krivulje in presečišča v slikah. V nadaljevanju bomo opisali testne podatke, testno okolje in potek testiranja, na koncu pa bomo prikazali in komentirali rezultate.

### 6.1 Testni podatki

Zaradi pomanjkanja ustreznih testnih podatkov smo se odločili, da za testne podatke uporabimo skice letal iz druge svetovne vojne, ki smo jih pridobili na spletni strani [45]. Podobne skice so za testne slike uporabili tudi nekateri avtorji [5], [46] pri razpoznavi objektov iz obrisov teh objektov. Slike vsebujejo tri poglede na letala; frontalni, stranski in tloris (s spodnje strani). Za potrebe testiranja smo uporabili le skico s tlorisnim pogledom, ki smo jo s predhodno obdelavo ustrezno obdelali. Predobdelava zajema obstoječe, preproste in dobro znane postopke obdelave slik. Sliko smo najprej obrezali, da smo dobili le tlorisni pogled, nato smo jo upragovili, da smo dobili binarno sliko, nato pa smo s programom za grafično oblikovanje zapolnili praznine v sliki. Tako smo dobili binarno masko objekta, ki smo jo uporabili v postopku, ki je opisan v naslednjih dveh poglavjih. Končna vhodna slika je bila velikosti  $564 \times 474$  slikovnih elementov. Slika 6.1 prikazuje postopek obdelave testne slike.

Kot vhod v tenzorsko glasovanje bi lahko podali tudi neobdelano barvno



Slika 6.1: Postopek pridobivanja testne slike.

oziroma sivinsko sliko, vendar bi bila analiza take slike v trenutni fazi implementacije prezahtevna. Prvi problem bi bil določanje regij, ki ga nismo implementirali. Drugi problem bi bilo daljše izvajanje tenzorskega glasovanja, ker bi glasoval vsak tenzor na mestu slikovnih elementov z intenziteto različno od 0 (tudi slikovni elementi ozadja in notranjosti regij objektov).

Maske objektov bi lahko dobili tudi iz realnih slik, zajetih s slikovnim senzorjem, vendar smo se zaradi pomanjkanja ustreznih slik in slabe kvalitete (ločljivosti) odločili za zgoraj opisan postopek ter za testiranje metode tenzorskega glasovanja le na eni realni sliki. Slika 6.2 prikazuje realno sliko, zajeto z digitalnim aparatom, iz katere lahko dobimo masko objekta. Sliko smo našli na spletni strani [47]. Ločljivost slike je  $470 \times 300$  slikovnih elementov.



Slika 6.2: Slika letala, pridobljena z digitalnim fotoaparatom.

## 6.2 Testno okolje

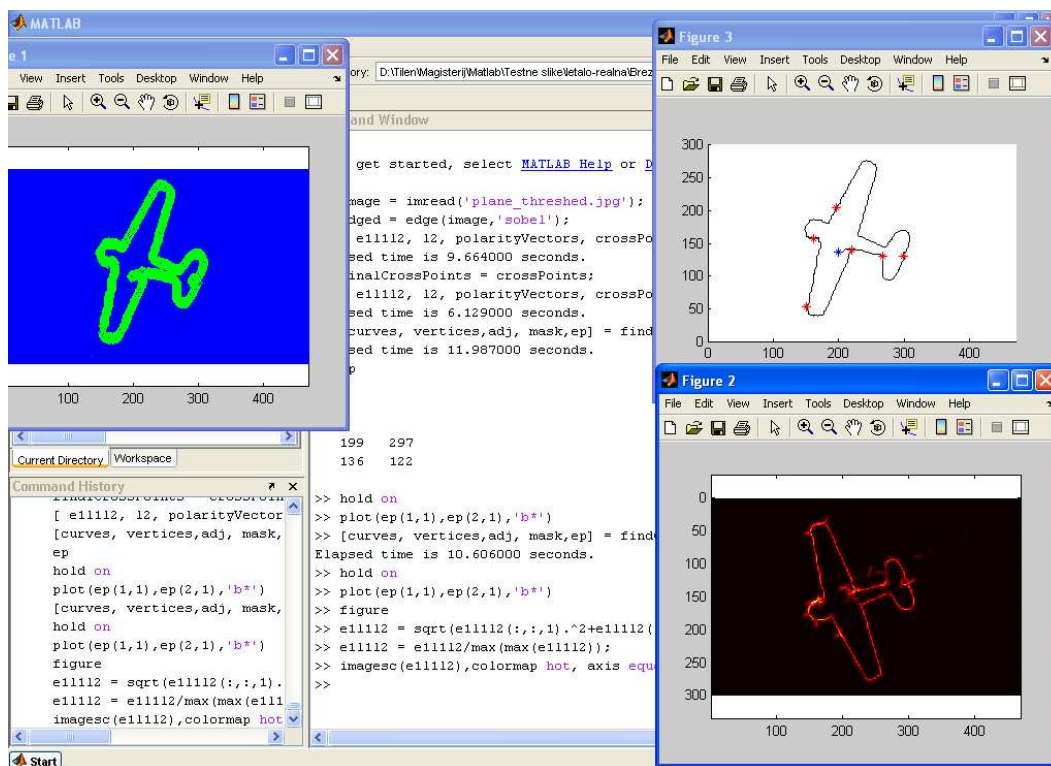
Za testno okolje smo uporabili program MathWorks Matlab 7 [7]. V tem okolju smo napisali funkcije za tenzorsko glasovanje, določanje krivulj in konstrukcijo grafov, uporabili pa smo tudi funkcije iz Matlabovega vgrajenega orodja za obdelavo slik, to je Image Processing Toolbox. Iz ukaznega načina Matlaba smo poganjali funkcije, v katerih smo implementirali v prejšnjih poglavjih opisane postopke. Izvajanje smo razdelili na tri korake:

- nalaganje slik z diska in robljenje s Sobelovim [2] operatorjem,
- izvajanje tenzorskega glasovanja in
- določanje krivulj iz tenzorskih in vektorskih polj ter konstrukcija grafa.

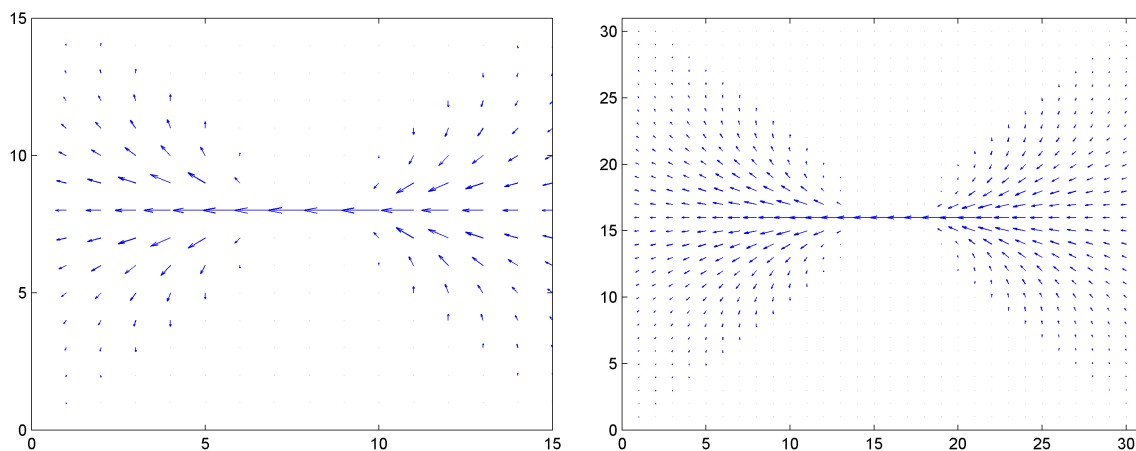
Rezultate smo spremljali s pomočjo izpisa v ukazni vrstici in z izrisom z vgrajenimi Matlabovimi funkcijami. Za prikaz rezultatov smo uporabljali funkcije *plot*, *quiver*, *image* oziroma *imagesc* ter *gplot*, za merjenje časa pa smo uporabili funkciji *tic* in *toc*. Slika 6.3 prikazuje testno okolje. Testiranje smo izvajali na osebнем računalniku s procesorjem Intel Pentium M 1.5GHz s 512 MB pomnilnika in operacijskim sistemom Microsoft Windows XP Professional.

## 6.3 Opis testiranja in rezultati

S testiranjem smo želeli ugotoviti odpornost metode tenzorskega glasovanja na šum v vhodni sliki ter vpliv velikosti tenzorskih polj na rezultate glasovanja. Predvidevali smo, da se z večanjem glasovalnih polj poveča odpornost na šum, vendar na račun izgube detajlov na sliki. Velikost glasovalnih polj smo pri testiranju podajali kot dvakratno dimenzijo glasovalnega jedra v slikovnih elementih. Če je kot velikost glasovalnega polja podano na primer 30, to pomeni da je glasovalno polje velikosti  $15 \times 15$  slikovnih elementov. Parameter  $\sigma_v$ , ki je definiran v poglavju 5.1 v enačbi (5.4), se je pri večanju velikosti glasovalnega polja ustrezno manjšal, da so bili tenzorji razporejeni po celotnem glasovalnem polju. Primer dveh različno velikih glasovalnih polj je prikazan na sliki 6.4. Spremljali smo tudi čas izvajanja glasovanja v odvisnosti od velikosti glasovalnih polj. Vhodni



Slika 6.3: Testno okolje.



Slika 6.4: Primerjava paličastih glasovalnih polj velikosti 30 in 60.

sliki smo dodajali Gaussov šum v vedno večji meri, nato pa opazovali rezultate. Razlog za dodajanje ravno Gaussovega šuma je, da je ta tip šuma prisoten pri vseh slikovnih senzorjih, zlasti pa postane opazen pri visokih ISO vrednostih in kratkih osvetlitvenih časih pri zajemu slik [48]. Gaussov šum opišemo s srednjo vrednostjo  $\mu$  in varianco  $\sigma^2$ . Slikovni elementi, katerim dodajamo Gaussov šum,

zavzemajo vrednosti med 0 in 1. Vrednost variance lahko torej interpretiramo na naslednji način. Če sliki dodamo šum s srednjo vrednostjo  $\mu = 0$  in  $\sigma^2 = 0.1$ , je verjetnost, da se bodo vrednosti slikovnih elementov povečale ali zmanjšale za 0.2, enaka 95%.

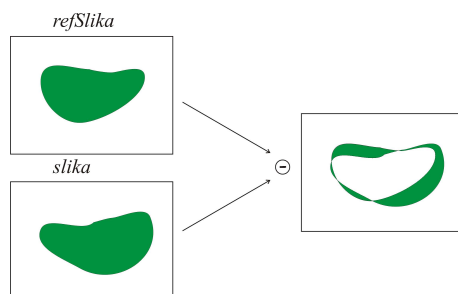
Uspešnost tenzorskega glasovanja smo merili na dva načina. Prvi način je bil opazovanje izrisanih krivulj in subjektivna ocena o uspešnosti, drugi pa empirični, to pomeni merjenje napake med določenimi krivuljami v referenčni sliki in v slikah z dodanim šumom. Napako smo merili kot razliko površin, ki jo oklepajo določene krivulje v referenčni sliki in določene krivulje v sliki, ki smo ji dodajali šum,

$$err = \sum_{i=1}^{n \times m} refSlika(i) - slika(i), \quad (6.1)$$

kjer je  $i$  indeks slikovnega elementa,  $n \times m$  je velikost slike,  $refSlika$  je slika določenih krivulj na referenčni sliki,  $slika$  pa je slika določenih krivulj na slikah z dodanim šumom. Napako smo podali kot relativno napako

$$\Delta err = \frac{err}{n \times m} \cdot 100\%. \quad (6.2)$$

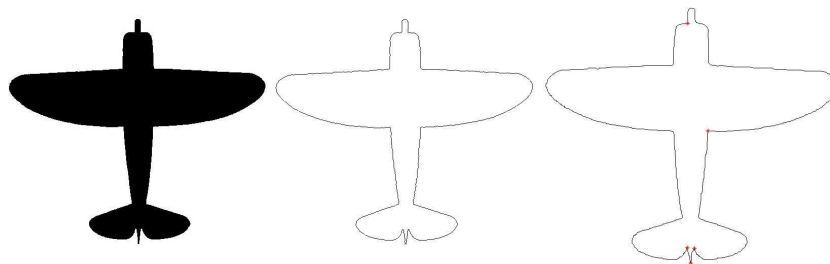
Če določene krivulje niso bile sklenjene, smo kot ploščino uporabili konveksno ogrinjačo. Slika 6.5 grafično prikazuje postopek izračuna napake med določenimi krivuljami. V nadaljevanju bomo predstavili rezultate testiranja na sliki, ki smo jo umetno pridobili iz skic letal ter na realni sliki. Prikazali bomo tudi primer opisa slike z vloženim grafom, kjer so povezave labelirane s kontrolnimi točkami B-zlepkov. Na podlagi te slike bomo v poglavju 7. ponovno opisali ideje za nadaljnje delo.



Slika 6.5: Razlika ploščin, ki jih oklepajo določene krivulje kot mera napake.

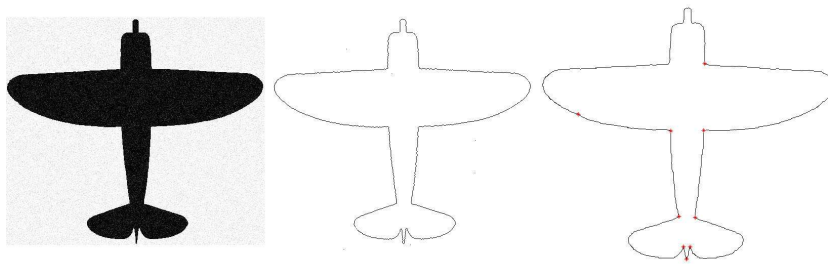
### 6.3.1 Testiranje na umetno pridobljeni sliki

Najprej smo metodo tenzorskega glasovanja preizkusili na umetno pridobljeni sliki. Sliki smo dodajali Gaussov šum s srednjo vrednostjo 0 in različno varianco. Za dodajanje šuma smo uporabili Matlabovo funkcijo *imnoise*. Nad zašumljenimi slikami smo izvedli robljenje s Sobelovim operatorjem z funkcijo *edge*. Praga pri robljenju nismo določili, temveč smo Matlabu prepustili avtomatski izračun praga. Nato smo nad robljeno sliko dvakrat izvedli tenzorsko glasovanje z različnimi velikostmi glasovalnih polj. Po prvem glasovanju z manjšimi glasovalnimi polji smo iz slike dobili možne kandidate za presečišča, po drugem glasovanju z večjimi glasovalnimi polji pa smo v sliki določili še krivulje ter zgradili vloženi graf. Razlog za dvakratno glasovanje je v tem, da smo pri iskanju presečišč potrebovali manjša glasovalna polja kot pri iskanju krivulj. Z naraščanjem velikosti glasovalnih polj narašča tudi izguba podrobnosti, čemur pa smo se želeli pri iskanju presečišč izogniti, obenem pa se večja odpornost metode na šum. Na ta način smo presečišča določili z večjo natančnostjo, pri določanju krivulj pa smo na račun večjih polj lažje določili krivulje. Na spodnjih slikah so rezultati prikazani grafično. Slike od 6.6 - 6.15 prikazujejo vhodno sliko z dodanim šumom, robljeno sliko ter določene krivulje in presečišča, ki so rezultat tenzorskega glasovanja. Z rdečo zvezdico so označena presečišča krivulj, z modro pa končne točke. Varianco šuma smo večali le do vrednosti 0.4, kjer je bil rezultat že slab, povečal pa se je tudi čas določanja krivulj.

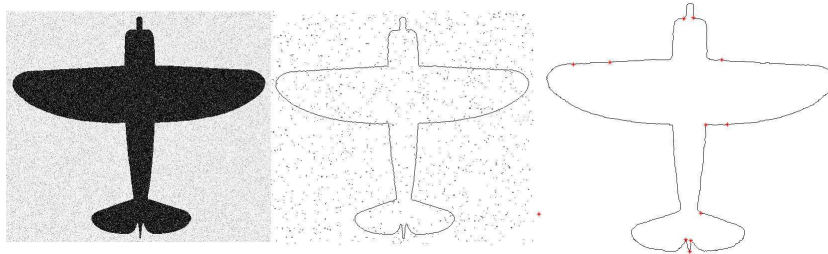


Slika 6.6: Rezultati tenzorskega glasovanja in določanja krivulj na sliki brez danega šuma.

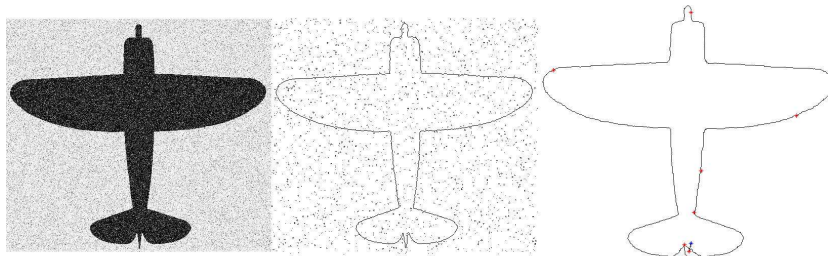
V tabeli 6.1 so prikazani še tabelarični rezultati, kjer T1 oziroma V1 pomeni tenzorsko oziroma vektorsko glasovalno polje za določanje presečišč, T2 oziroma V2 pa tenzorsko oziroma vektorsko glasovalno polje za določanje krivulj. Podana



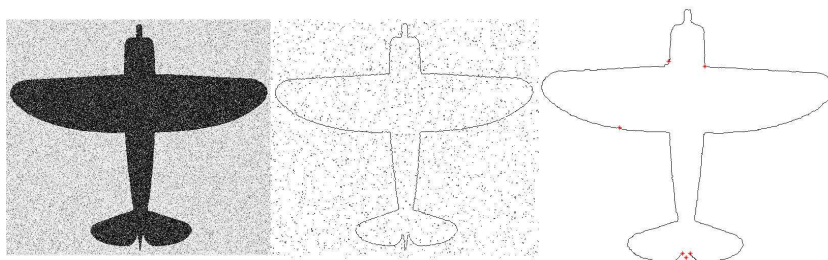
Slika 6.7: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.01$ .



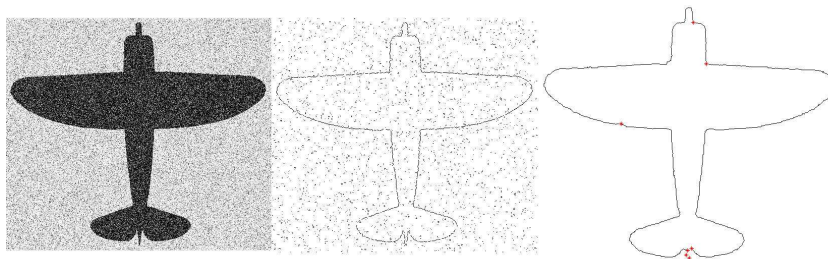
Slika 6.8: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.05$ .



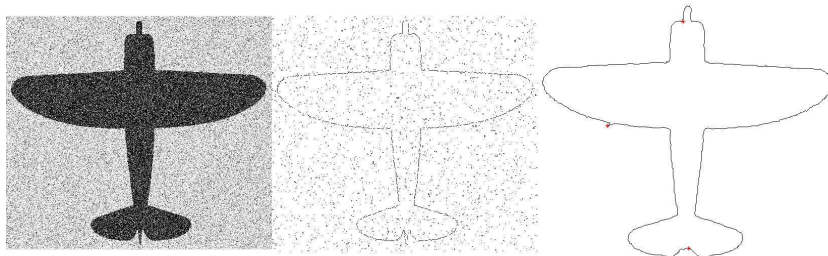
Slika 6.9: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.1$ .



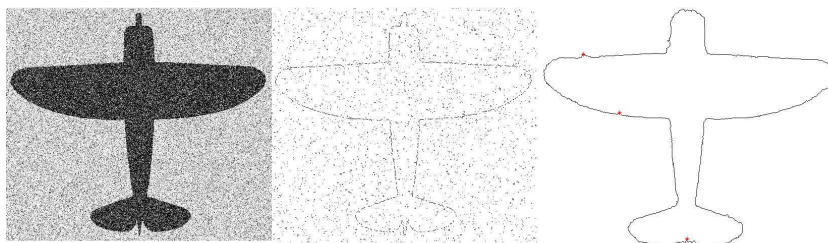
Slika 6.10: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.15$ .



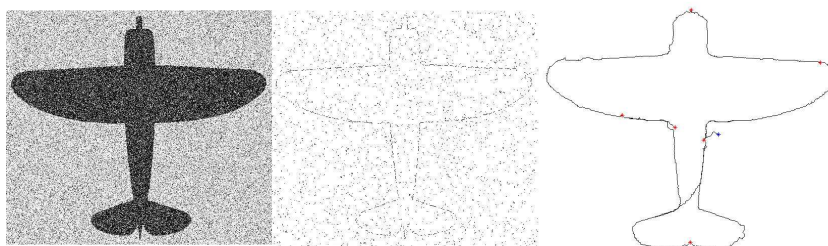
Slika 6.11: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.2$ .



Slika 6.12: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.25$ .

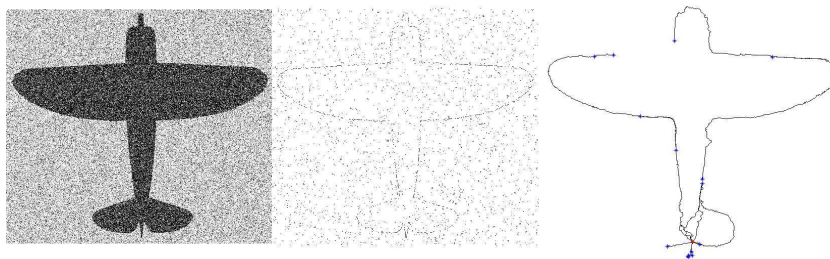


Slika 6.13: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.3$ .



Slika 6.14: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.35$ .





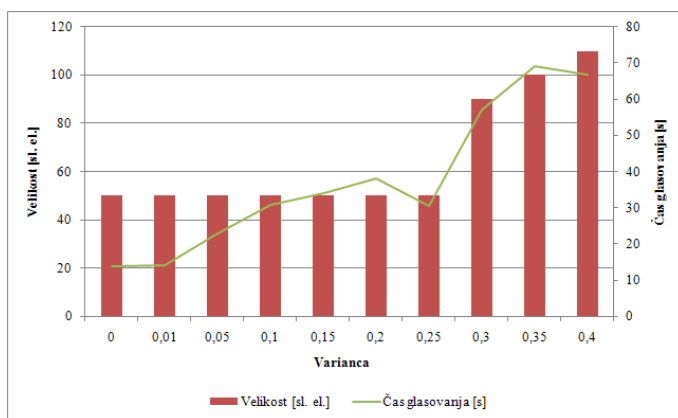
Slika 6.15: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.4$ .

sta tudi časa za obe glasovanji ter čas določanja krivulj.

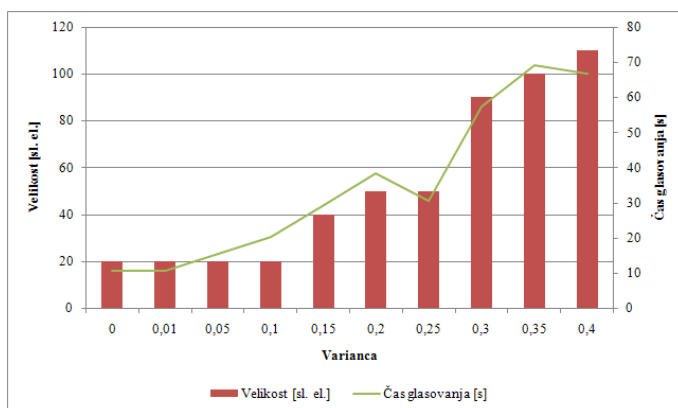
$\sigma^2$	Velikost T1	Velikost V1	Čas glasovanja [s]	Velikost T2	Velikost V2	Čas glasovanja [s]	Čas določanja krivulj [s]	$\Delta err$ [%]
0	20	30	10.5	50	50	13.9	41.8	/
0.01	20	30	10.6	50	30	14.2	42.5	0.32
0.05	20	30	15.4	50	30	23	41.5	0.34
0.1	20	30	20.3	50	30	31	43.7	0.36
0.15	40	30	29.2	50	30	34.1	41.6	0.32
0.2	50	30	38.3	50	30	38.3	41.6	0.35
0.25	50	30	30.5	50	30	30.5	41.6	0.37
0.3	90	30	57.3	90	30	57.3	40.5	0.6
0.35	100	30	69.2	100	30	69.2	69.6	0.75
0.4	110	30	66.9	110	30	66.9	238.8	3.12

Tabela 6.1: Rezultati tenzorskega glasovanja in določanja krivulj.

Kot smo domnevali, je potrebno ob povečanem šumu vhodne slike povečati velikost glasovalnih polj. Na ta račun izgubimo podrobnosti v sliki (na zgornjih slikah je to vidno pri repu in kljunu letala), čas glasovanja pa se podaljša. Od šuma z varianco višjo od 0.20, smo za iskanje presečišč in krivulj uporabljali enako velika glasovalna polja, ker pri iskanju presečišč visoke natančnosti nismo več potrebovali (na zgornjih slikah je to zopet vidno pri repu letala, kjer namesto treh presečišč najdemo le še eno). Na grafih 6.16 in 6.17 sta prikazana velikost glasovalnih polj in čas tenzorskega glasovanja v odvisnosti od šuma.



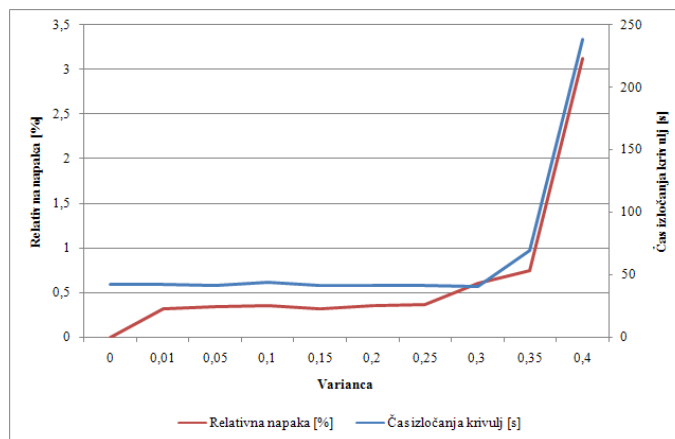
Slika 6.16: Velikost glasovalnih polj in čas glasovanja za določanje presečišč v odvisnosti od variance.



Slika 6.17: Velikost glasovalnih polj in čas glasovanja za določanje presečišč v odvisnosti od variance.

Ob večanju šuma smo večali tenzorska glasovalna polja tako, da smo vizualno preverjali določene krivulje in velikost prilagajali toliko časa, dokler nismo dobili optimalnega rezultata. Vektorskih glasovalnih polj nismo večali, ker jih potrebujemo le za natančno določitev končnih točk krivulj. V našem primeru smo se ukvarjali s sklenjenimi krivuljami, zato vpliva velikosti vektorskih glasovalnih polj nismo preverjali. Ob večanju šuma se je večal tudi čas glasovanja. Vzroka sta dva. Prvi je, da smo pri večjem šumu uporabili večja glasovalna polja. Ker je glasovanje izvedeno kot 2-D konvolucija, glasovanje z večjimi polji traja dlje. Drugi razlog je, da se z večanjem šuma večja število žetonov (slikovnih elementov), ki glasujejo. Na grafih je vidno, da je kljub večanju glasovalnih polj, čas glasovanja pri  $\sigma^2 = 0.25$  in  $\sigma^2 = 0.4$  upadel, nato pa pri  $\sigma^2 = 0.3$  znova pričel

naraščati. Razlog je v tem, da pri robljenju nismo določili pragu, ampak ga je Matlab nastavil avtomatično. Na ta način je po robljenju ostalo manj žetonov, ki glasujejo, vendar je bil na račun močnejšega šuma obris letala manj izrazit.



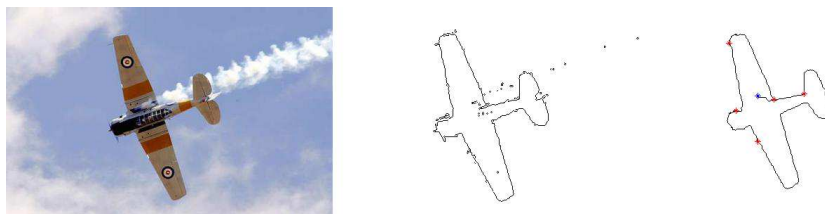
Slika 6.18: Velikost glasovalnih polj in čas glasovanja za določanje presečišč v odvisnosti od variance.

Relativna napaka z večanjem ni naraščala tako, kot smo pričakovali (graf 6.18). Do šuma z  $\sigma^2 = 0.25$  je bila skoraj konstantna, kar priča o dobri odpornosti metode tenzorskega glasovanja na šum v vhodnih slikah. Pri šumu z  $\sigma^2 > 0.25$  je napaka začela hitro naraščati. Pri šumu z  $\sigma^2 = 0.4$  je napaka znašala že več kot 3%. Pri šumu z  $\sigma^2 > 0.4$  določanje krivulj iz tenzorskih polj v trenutni fazi implementacije ni več delovalo, zato smo pri šumu s to vrednostjo variance prenehali s testiranjem. Potrebno je omeniti, da je bila vhodna slika glede na velikost letala velika, zato je tudi relativna napaka nizka.

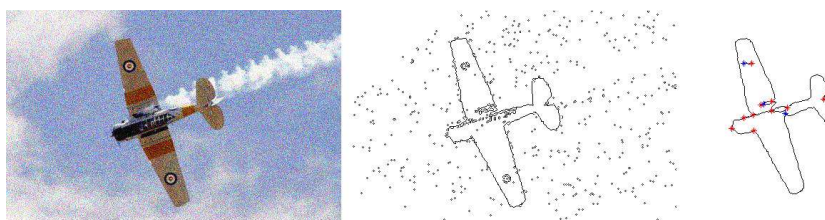
### 6.3.2 Testiranje na realni sliki

Metodo smo želeli preizkusiti tudi na realni sliki, ki smo jo opisali v poglavju 6.1. Sliki smo zopet dodajali Gaussov šum z  $\mu = 0$  in različno varianco. Zaradi zahtevnejše slike smo v tem koraku šum dodajali v manjših korakih ter testiranje končali pri precej manjši vrednosti variance dodanega šuma. Iz slike smo najprej z upravljenjem dobili binarno masko objekta. Izkazalo se je, da letalo najbolj izstopa iz ozadja v modrem kanalu, zato smo za upravljenje uporabili le modri kanal slike. Binarno masko smo nato robili s Sobelovim operatorjem, rezultat robljenja pa smo uporabili kot vhod v tenzorsko glasovanje. Tudi pri tej sliki

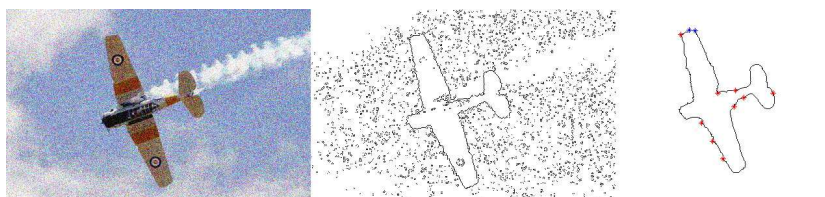
smo za določitev presečišč in krivulj tenzorsko glasovanje izvajali ločeno v dveh korakih z različno velikimi glasovalnimi polji. Na slikah 6.19 - 6.24 so prikazane vhodne slike z dodanim šumom, robovi maske objekta in rezultati tenzorskega glasovanja in določanja krivulj.



Slika 6.19: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki brez dodanega šuma.



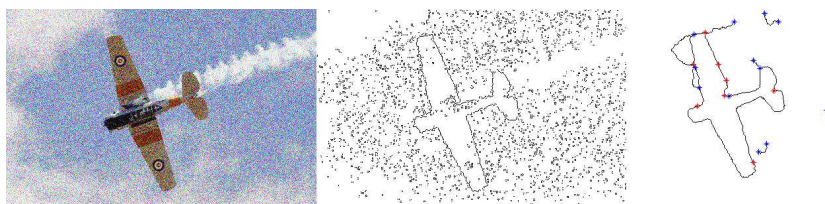
Slika 6.20: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.01$ .



Slika 6.21: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.02$ .

V tabeli 6.2 podajamo še tabelarične rezultate.

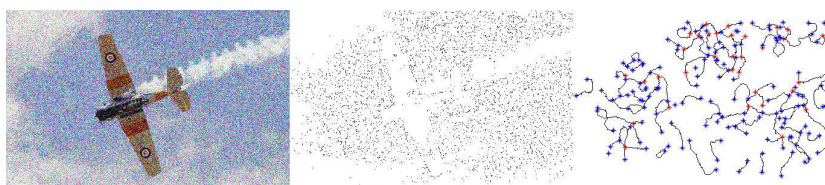
Tudi pri tej sliki smo morali ob večanju šuma večati glasovalna polja. Ker je ta slika manjša od prejšnje, smo uporabili manjša glasovalna polja. Zaradi manjše slike in manjših glasovalnih polj je bil tudi čas glasovanja krajši. Tu moramo poudariti, da je čas glasovanja odvisen tudi od števila slikovnih elementov, ki glasujejo. Na grafih 6.25 in 6.26 sta prikazana velikost glasovalnih polj in čas



Slika 6.22: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.03$ .



Slika 6.23: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.04$ .

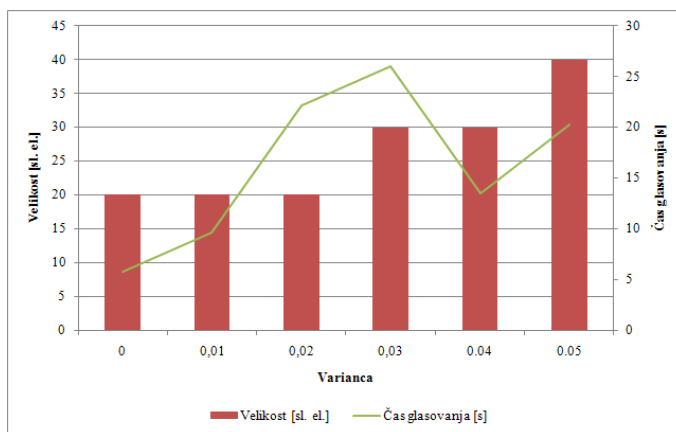


Slika 6.24: Rezultati tenzorskega glasovanja in določanja krivulj pri sliki z dodanim šumom  $\sigma^2 = 0.05$ .

$\sigma^2$	Velikost T1	Velikost V1	Čas glasovanja [s]	Velikost T2	Velikost V2	Čas glasovanja [s]	Čas določanja krivulj [s]	$\Delta err$ [%]
0	20	20	5.8	30	20	6.1	11.5	/
0.01	20	20	9.7	40	20	13.1	11.7	0.8
0.02	20	20	22.1	50	20	35.4	12.9	0.6
0.03	30	20	26	40	20	31.3	20.1	1.6
0.04	30	20	13.5	50	20	16.2	19.7	2.5
0.05	40	20	20.3	50	20	24.1	50.3	/

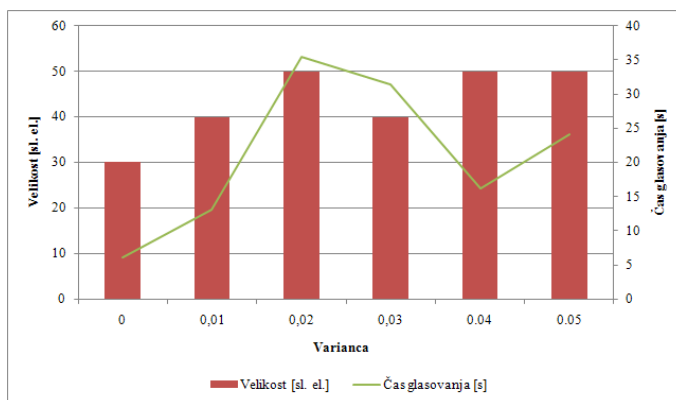
Tabela 6.2: Rezultati tenzorskega glasovanja in določanja krivulj.

glasovanja v odvisnosti od variance dodanega šuma za oba koraka tenzorskega glasovanja.



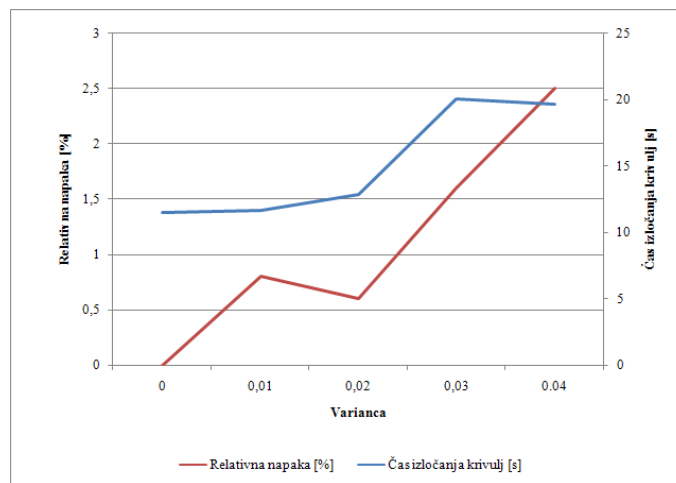
Slika 6.25: Velikost glasovalnih polj in čas glasovanja za določanje presečišč v odvisnosti od variance.

Čas glasovanja v prvem koraku se je z večanjem šuma večal. Vendar se je pri vrednosti  $\sigma^2 = 0.04$  zaradi samodejnega določanja pragu pri robljenju število slikovnih elementov, ki glasujejo, zmanjšalo, zato se je tudi čas glasovanja zmanjšal.



Slika 6.26: Velikost glasovalnih polj in čas glasovanja za določanje krivulj v odvisnosti od variance.

Čas glasovanja v drugem koraku se je ravno tako do neke mere višal, kjer je nato pričel upadati, nato pa se je znova začel večati. Za razliko od glasovanja v prvem koraku, je čas upadel že pri vrednosti  $\sigma^2 = 0.03$ , pri vrednosti  $\sigma^2 = 0.04$  se je čas še dodatno zmanjšal, pri  $\sigma^2 = 0.05$  pa je znova narasel. Vzrok za upadanje časa glasovanja pri nižji vrednosti variance je zmanjšanje velikosti glasovalnega polja pri  $\sigma^2 = 0.03$ .



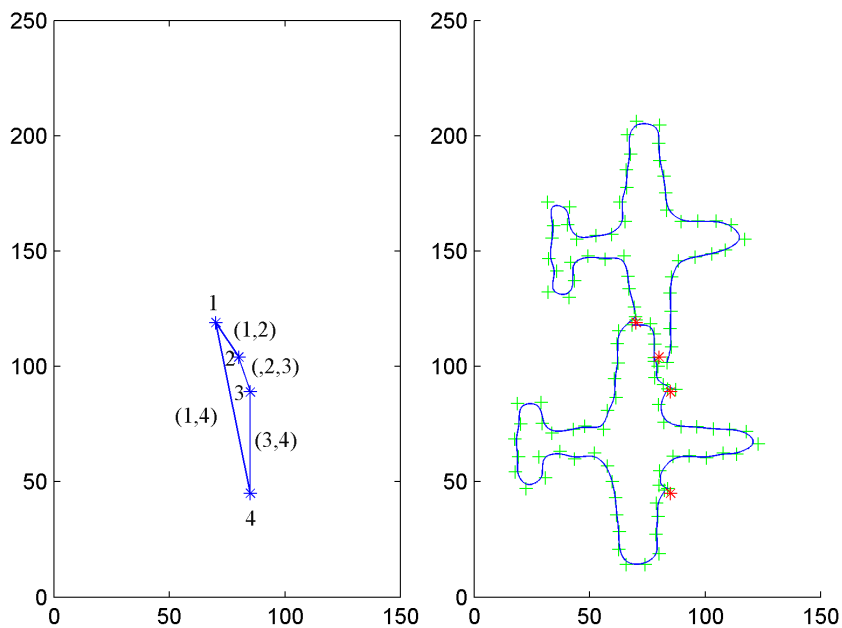
Slika 6.27: Relativna napaka določenih krivulj in čas določanja krivulj v odvisnosti od variance.

Čas določanja krivulj se je z večanjem šuma daljšal (graf 6.27). Ravno tako je z večanjem šuma naraščala relativna napaka pri določanju krivulj. Pri šumu z  $\sigma^2 = 0.03$  je napaka znašala 1.6%. Na sliki 6.22 vidimo, da je sam obris letala še vedno dobro določen, vendar se začnejo v sliki pojavljati dodatne krivulje, ki ne pripadajo letalu. Pri šumu z  $\sigma^2 = 0.04$  je ta pojav še bolj opazen (slika 6.23), napaka pa je znašala 2.5%. Na sliki 6.24 je prikazan tudi rezultat določanja krivulj na sliki z dodanim šumom s  $\sigma^2 = 0.05$ . Obrisi letala je postal neprepoznaven, rezultat so bile krivulje, ki ne pripadajo letalu. Zaradi velika števila določenih krivulj v zadnjem primeru napake nismo računali.

### 6.3.3 Prikaz opisa slike z vloženim grafom

V tem podpoglavju bomo prikazali še enostaven primer končnega rezultata postopka, ki je predstavljen v magistrski nalogi. To je opis slike z vloženim grafom, ki je labeliran s koeficienti B-zlepkov določenih krivulj. Slika 6.28 prikazuje graf z označenimi vozlišči in povezavami. Vozlišča grafa ustrezajo presečiščem in končnim točkam krivulj, povezave med vozlišči pa krivuljam. S preslikavo  $\phi$ , ki je definirana v poglavju 5.4, vozliščem grafa priredimo položaj v ravnini, kar pomeni da graf vložimo. Na sliki je prikazan tudi diagram grafa, katerega povezave smo labelirali z vektorjem kontrolnih točk B-zlepkov posameznih krivulj. Graf labeliramo s preslikavo  $\psi$ , ki je definirana v poglavju

5.4. Z rdečimi zvezdicami so prikazana vozlišča grafa, z zeleni plusi kontrolne točke, z modro krivuljo pa so narisani B-zlepki, ki jih določajo kontrolne točke.



Slika 6.28: Prikaz grafa, ki ima povezave labelirane s koeficienti B-zlepkov.

Takšen opis slike je primeren za višje-nivojske algoritme za razpoznavo objektov in relacij med njimi. Z uporabo grafovskega algoritma in algoritma za razpoznavo objektov iz kontrolnih točk B-zlepkov bi na tej sliki razpoznali dve različni letali ter ugotovili, da se eno letalo nahaja nižje in malo bolj levo kot drugo.



## 7. Zaključek in nadaljnje delo

V tem poglavju bomo podali sklepne ugotovitve, povzeli opisano rešitev ter rezultate tenzorskega glasovanja. Na kratko bomo opisali probleme, s katerimi smo se srečali pri implementaciji tenzorskega glasovanja. Opisali bomo tudi možnosti za nadaljnje delo, ki se kažejo v dveh smereh. Prva je nadaljnje delo na metodi tenzorskega glasovanja, druga pa na analizi in razpoznavi vidne scene iz vložene grafa ter kontrolnih točk B-zlepkov.

### 7.1 Sklep

V magistrskem delu smo želeli predstaviti analizo slike, katere rezultat je primeren za razpoznavo objektov. Implementirali smo segmentacijo s tenzorskim glasovanjem, zapis krivulj s kontrolnimi točkami B-zlepkov in zapis vidne scene z vložnim grafom. Pri testiranju smo se osredotočili le na testiranje metode tenzorskega glasovanja. S testiranjem smo ugotovili, da je metoda do neke mere res odporna na šum in diskontinuitete v vhodnih podatkih, ker nam je kljub visokemu šumu uspelo še vedno precej dobro določiti objekte v sliki. Poleg tega zaradi zapisa vhodnih podatkov s tenzorji pravilno sklepamo na tipe struktur, ki se nahajajo v sliki. Pri sami implementaciji tenzorskega glasovanja in določanja krivulj smo se srečali s težavami, ki upravičujejo nadaljnje delo v smeri izboljšanja implementacije. Te težave so:

- določanje velikosti glasovalnih polj,
- določanje prostih parametrov glasovanja in sledenja krivuljam,
- dolg čas izvajanja tenzorskega glasovanja,
- dolg čas določanja krivulj in

- določanje regij iz tenzorskih polj.

Največjo težavo je predstavljala določitev prostih parametrov, ki jih bomo našli v nadaljevanju. Uvedli smo jih pri tenzorskem glasovanju in določanju krivulj. Kljub temu, da se v literaturi o tenzorskem glasovanju  $\sigma_v$  pojavlja kot edini prosti parameter metode, smo morali pri implementaciji glasovanja in določanja krivulj uporabiti še veliko ostalih parametrov. To so velikost tenzorskih in vektorskih glasovalnih polj v enotah slikovnih elementov, vrednost pragov za presečišča, semena in končne točke krivulj, velikost uvrščevalnega filtra, minimalna razdalja med točko na krivulji in presečiščem ter minimalna razdalja med točko na krivulji in že obstoječo krivuljo. Te parametre smo nastavljali ročno za vsako sliko posebej, dokler nismo dobili najbolj optimalnih rezultatov. Čas izvajanja glasovanja in določanja krivulj je znašal po nekaj deset sekund, kar je za uporabo v praktične namene preveč. Kot problem se je izkazalo tudi določanje regij iz slik, zato ga v delu nismo izvajali. S težavami smo se srečali tudi pri določanju krivulj iz tenzorskih polj. Prvi večji problem je bil neurejena usmerjenost tenzorjev v bližini presečišč. Zaradi tega smo morali krivulje zaključiti že nekaj slikovnih elementov pred presečišči, drugače bi krivulja zašla iz prave smeri. Drugi večji problem se je pojavil zaradi simetričnosti tenzorjev. Ker smo uporabljali simetrične tenzorje, je imel lahko paličasti tenzor tako smer  $\mathbf{t}$  kot tudi  $-\mathbf{t}$ . Zaradi tega smo morali pri sledenju krivulj računati skalarni produkt zadnjih dveh vektorjev, da smo lahko zaznali obrat tangente. Kljub omenjenim težavam se je metoda tenzorskega glasovanja izkazalo za dobro segmentacijsko metodo zašumljenih slik. Opazili smo, da metoda do neke mere šum izloča zelo uspešno. Napaka v določenih krivuljah je znašala le 0.75% pri šumu z  $\sigma^2 = 0.35$ . Od te meje naprej pa metoda pri določanju krivulj odpove, rezultat je veliko krivulj, ki niso del iskanega objekta. Poleg odpornosti na šum je prednost tenzorskega glasovanja in določanja krivulj pred ostalimi segmentacijskimi metodami zapis krivulj na tak način, da jih lahko neposredno uporabimo pri prileganju B-zlepkov oziroma pri računanju kontrolnih točk. Celotna predlagana rešitev za analizo slike in opis vidne scene kaže velik potencial, zato bi bilo nadaljnje delo, ki ga opisujemo v naslednjem podpoglavju, upravičeno.

## 7.2 Nadaljnje delo

Možnosti za nadaljnje delo se kažejo na več področjih. Prva izmed njimi je bolj učinkovita implementacija tenzorskega glasovanja in določanja krivulj ter po-hitritev obeh postopkov. Neizkoriščena je ostala možnost določanje regij iz tenzorskih polj. Naslednja možnost je avtomatizacija določanja parametrov, ki jih je potrebno nastaviti za uspešno delovanje tenzorskega glasovanja in določanja krivulj. Na ta način bi zmanjšali potrebo po uporabnikovi intervenciji in metodo tenzorskega glasovanja bi lahko uporabili kot robustno metodo za segmentacijo. V literaturi o tenzorskem glasovanju se kot največji izzivi kažejo še v optimalnem določanju velikosti glasovalnih polj, posplošitvi in uporabi metode pri N-dimenzijskih podatkih in optimizaciji pomnilniških in računskih virov, kjer postane metoda zaradi predstavitve podatkov zelo zahtevna.

Nadaljnje delo bi bilo še posebej treba nadaljevati v smeri razpoznave objektov in relacij med njimi iz vloženi grafov. Potrebno bi bilo implementirati grafovski algoritem za iskanje zaključenih poti po grafu ter algoritem za razpoznavo objektov, katerih robovi so zapisani s kontrolnimi točkami B-zlepkov. Iz grafa in kontrolnih točk B-zlepkov bi lahko določili geometrijske in topološke lastnosti objektov, ter nato objekte tudi razpoznali in vidno sceno na sliki opisali.



## Literatura

- [1] B. E. Stevenson. *The Home Book of Proverbs, Maxims and Familiar Phrases*. Macmillan Co., 1948.
- [2] I. Pitas. *Digital Image Processing Algorithms*. Prentice Hall, 1993.
- [3] N. Pavešić. *Razpoznavanje vzorcev: Uvod v analizo in razumevanje vidnih in slušnih signalov*. Druga razširjena izdaja. Založba FE in FRI, 2000.
- [4] G. Medioni and S. B. Kang. *Emerging topics in computer vision*. Prentice Hall, New Jersey, 2005.
- [5] F. S. Cohen, Z. Huang, and Z. Yang. Invariant matching and identification of curves using b-splines curve representation. *IEEE Transactions on image processing*, 4(1), January 1995.
- [6] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2000.
- [7] Matlab - the language of technical computing.  
<http://www.mathworks.com/products/matlab/>, 2008.  
Zadnji dostop 10.3.2008.
- [8] Matlab - wikipedia, the free encyclopedia.  
<http://en.wikipedia.org/wiki/MATLAB>, 2008. Zadnji dostop 10.3.2008.
- [9] Matlab central file exchange - toolbox tensor voting.  
<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=5949&objectType=File>, 2004. Zadnji dostop 10.3.2008.
- [10] G. Medioni, M.-S. Lee, and C.-K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

- 
- [11] W.-S. Tong, C.-K. Tang, P. Mordohai, and G. Medioni. First order augmentation to tensor voting for boundary inference and multiscale analysis in 3d. *IEEE Transactions on pattern analysis and machine intelligence*, 26(5), May 2004.
- [12] B. Kisačanin, V. Pavlović, and T. S. Huang, editors. *Real-Time Vision for Human-Computer Interaction*. Springer, 2005.
- [13] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, New Jersey, 2001.
- [14] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, Feb. 2006.
- [15] P. Remagnino, G. A. Jones, N. Paragios, and C. S. Regazzoni, editors. *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*. Springer, 2001.
- [16] A. D. Marshall and R. R. Martin. *Computer Vision, Models, and Inspection*. World Scientific Publishing Company, 1992.
- [17] L. Costaridou, editor. *Medical Image Analysis Methods*. Taylor & Francis Group, 2005.
- [18] M. Cristani, M. Bicego, and V. Murino. Audio-visual event recognition in surveillance video sequences. *IEEE Transactions on Multimedia*, 9(2), February 2007.
- [19] H. Greenspan, J. Goldberger, and A. Mayer. Probabilistic space-time video modeling via piecewise gmm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3), March 2004.
- [20] D. H. Ballard and C. M. Brown. *Computer vision*. Prentice Hall, New Jersey, 1982.
- [21] R. M. Haralick and L. G. Shapiro. *Computer and robot vision*. AddisonWesley, New York, 1993.

- 
- [22] T. Acharya and A. R. Ajoy. *Image processing: Principles and Applications*. John Wiley & Sons, 2005.
- [23] W. E. Snyder and H. Qi. *Machine Vision*. Cambridge University Press, 2004.
- [24] J. R. Smith and C.-S. Li. Decoding image semantics using composite region templates. *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1998.
- [25] M. Chengliang, X. Shengli, and Y. Weiyu. Image semantics segmentation using watershed algorithm. *IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI '06.*, 2006.
- [26] C. W. Therrien. *Decision, Estimation, and Classification: An Introduction to Pattern Recognition and Related Topics*. Wiley, New York, 1989.
- [27] R. Cappeli, D. Maio, and D. Maltoni. Multispace kl for pattern representation and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9), September 2001.
- [28] X. S. Zhou and T. S. Huang. Comparing discriminating transformations and svm for learning during multimedia retrieval. *In proceedings of the ninth ACM international conference on Multimedia*, 2001.
- [29] R. Krishnapuram, S. Medasani, S. H. Jung, Y. S. Choi, and R. Balasubramaniam. Content-based image retrieval based on a fuzzy approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(8), May 2004.
- [30] M. S. Lew and D. Dentender. Fisher keys for content based retrieval. *Image and Vision Computing*, 19(10), October 2001.
- [31] J. Fan, Y. Gao, H. Luo, and R. Jain. Mining multilevel image semantics via hierarchical classification. *IEEE Transactions on Multimedia*, 10(2), February 2008.
- [32] Aimetis aira v5.  
<http://www.aimetis.com/aira/>, 2005. Zadnji dostop 11.3.2008.

- 
- [33] Hitachi its21.  
[http://www.hitachi.co.jp/en/products/its/product/service/2004030\\_12381.html](http://www.hitachi.co.jp/en/products/its/product/service/2004030_12381.html),  
2007. Zadnji dostop 11.3.2008.
- [34] Identrace - intelligent video surveillance system.  
<http://www.identrace.com/>, 2007. Zadnji dostop 12.3.2008.
- [35] imgseek.  
<http://www.imgseek.net/>, 2007. Zadnji dostop 12.3.2008.
- [36] Behold.  
<http://www.behold.cc/>, 2007. Zadnji dostop 12.3.2008.
- [37] Flickr - photo sharing.  
<http://www.flickr.com/>, 2007. Zadnji dostop 12.3.2008.
- [38] Amazon ec2, amazon elastic compute cloud, virtual grid computing.  
<http://www.amazon.com/gp/browse.html?node=201590011>, 2007. Zadnji  
dostop 12.3.2008.
- [39] J. J. McConnel. *Computer Graphics: Theory into practice*. Jones and  
Bartlett Publishers, 2005.
- [40] D. J. Levitin. *Foundations of Cognitive Psychology: Core Readings*. The  
MIT Press, 2002.
- [41] C. de Boor. *A Practical guide to Splines*. Springer-Verlag, New York, 1978.
- [42] H. Stark and J. W. Woods. *Probability and Random Processes with applica-  
tion to Signal Processing*. Prentice Hall, New Jersey, 2002.
- [43] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Ap-  
plications*. Wiley, New York, 1977.
- [44] Graph theory - wikipedia, the free encyclopedia.  
[http://en.wikipedia.org/wiki/Graph\\_theory](http://en.wikipedia.org/wiki/Graph_theory), 2008. Zadnji dostop 10.3.2008.
- [45] Aircraft spotter cards. [http://www.508pir.org/photogallery/planespotters/main-  
spotters.htm](http://www.508pir.org/photogallery/planespotters/main-spotters.htm). Zadnji dostop 10.3.2008.



- [46] S. Jaggi, W. C. Karl, S. G. Mallat, and A. S. Willsky. Silhouette recognition using high-resolution pursuit. *Pattern Recognition* 32, (5), May 1999.
- [47] Realna testna slika. [http://www.smh.com.au/ffximage/2007/08/03/plane-wide-web\\_470x300,0.jpg](http://www.smh.com.au/ffximage/2007/08/03/plane-wide-web_470x300,0.jpg). Zadnji dostop 10.3.2008.
- [48] B. Long. *Complete digital photography*. Charles River Media, Boston, 2005.



## Izjava

Izjavljam, da sem magistrsko delo izdelal samostojno pod vodstvom mentorja doc. dr. Andreja Koširja, uni. dipl. mat. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.