

UNIVERZITET U BEOGRADU

MAŠINSKI FAKULTET

Veljko M. Marković

**PREPOZNAVANJE
GEOMETRIJSKIH PRIMITIVA
IZ TRODIMENZIONALNOG
OBLAKA TAČAKA**

doktorska disertacija

Beograd, 2021

UNIVERSITY OF BELGRADE
FACULTY OF MECHANICAL ENGINEERING

Veljko M. Marković

**RECOGNITION OF
GEOMETRIC PRIMITIVES
FROM THREE-DIMENSIONAL
POINT CLOUD**

Doctoral Dissertation

Belgrade, 2021

Komisija za pregled, ocenu i odbranu doktorske disertacije:

Mentor:

dr Živana Jakovljević, redovni profesor
Univerzitet u Beogradu, Mašinski fakultet

Članovi komisije:

dr Bojan Babić, redovni profesor
Univerzitet u Beogradu, Mašinski fakultet

dr Zoran Miljković, redovni profesor
Univerzitet u Beogradu, Mašinski fakultet

dr Saša Živanović, redovni profesor
Univerzitet u Beogradu, Mašinski fakultet

dr Igor Budak, redovni profesor
Univerzitet u Novom Sadu, Fakultet tehničkih nauka

Datum odbrane:

Izjave zahvalnosti

Želeo bih da izrazim neograničenu zahvalnost svim pojedincima koji su doprineli realizaciji ove doktorske disertacije.

Posebnu zahvalnost dugujem svom mentoru, prof. dr Živani Jakovljević, na nesebičnoj i stručnoj pomoći tokom izrade ove doktorske disertacije. Takođe, želeo bih da joj zahvalim na pruženom znanju i savetima tokom dugogodišnje zajedničke saradnje.

Zahvaljujem članovima stručne komisije, prof. dr Bojanu Babiću, prof. dr Zoranu Miljkoviću, prof. dr Saši Živanoviću i prof. dr Igoru Budaku na svom angažmanu i sugestijama kojima su doprineli kvalitetu ove disertacije.

Veliko hvala porodici i prijateljima na bezuslovnoj podršci.

Beograd, februar 2021.

Veljko M. Marković

PREPOZNAVANJE GEOMETRIJSKIH PRIMITIVA IZ TRODIMENZIONALNOG OBLAKA TAČAKA

Sažetak

Intenzivan razvoj tehnika reverznog inženjerstva doneo je značajne benefite i mašinskoj industriji. Pored tradicionalne upotrebe u projektovanju proizvoda, ove tehnike svoje mesto često pronalaze i u automatizaciji proizvodnih procesa. Međutim, za većinu primena u aplikacijama koje zahtevaju rad u realnom vremenu glavno ograničenje predstavlja potreba značajne korisničke asistencije. U skladu sa ovim ograničenjem i potrebom da se ono prevaziđe, prisutan je porast istraživanja i razvoja algoritma koji se koriste za automatsku obradu podataka u procesima reverznog inženjerstva. Ova doktorska disertacija je upravo usmerena na istraživanje u oblasti automatskog procesiranja podataka dobijenih digitalizacijom trodimenzionalnih (3D) objekata. Kako su procesi 3D digitalizacije u mašinskom inženjerstvu često svedeni na upotrebu različitih oblika 3D skeniranja, oblast istraživanja se odnosi na procesiranje oblaka tačaka.

Disertacijom su obuhvaćene tri teme iz oblasti prepoznavanja strukture skeniranih mašinskih objekata. Prva tema usmerena je na tehnike inteligentne redukcije broja tačaka inicijalnih oblaka. Druga tema orijentisana je na metode procesiranja oblaka tačaka u cilju prepoznavanja gradivnih geometrijskih primitiva posmatranih objekata, dok se poslednje obrađena tema odnosi na primenu tehnika veštačke inteligencije i mašinskog učenja za potrebe povećanja stepena autonomije pomenutih metoda prepoznavanja.

Doktorska disertacija takođe obuhvata razvoj i eksperimentalnu verifikaciju četiri originalne metode (algoritma) za obradu oblaka tačaka. Prva metoda (poglavlje 5) koristi regresiju nosećim vektorima kako bi u konturama struktuiranog oblaka tačaka identifikovala i zadržala tačke od najvećeg značaja za rekonstrukciju geometrije skeniranog objekta. U disertaciji su detaljno opisani glavni parametri regresije kao i njihov uticaj na redukciju sa visokim stepenom očuvanja inicijalne informacije o geometriji. Metoda je eksperimentalno verifikovana upotrebom tri realna oblaka tačaka koji predstavljaju različite objekte karakteristične geometrije. Takođe je, upotrebom razvijene metodologije zasnovane na neuniformnom racionalnom B-splajnu (NURBS), ocenjen pomenuti stepen prezervacije inicijalne informacije o geometriji nakon izvršene redukcije i izvedeni su upotrebljivi zaključci. Performanse predložene metode su takođe upoređene sa metodom redukcije uniformnim uzorkovanjem.

Druga razvijena metoda koja je predstavljena u disertaciji (poglavlje 6) obezbeđuje prepoznavanje jedne vrste površi drugog reda (kvadrika) iz struktuiranih oblaka tačaka. Metoda je zasnovana na segmentaciji skeniranih kontura upotrebom osobina singularnosti informacione matrice sistema tačaka tokom regresije elipse metodom najmanjih kvadrata. Pored detaljne prezentacije algoritma prepoznavanja izvršena je i njegova verifikacija procesiranjem nekoliko sintetizovanih i jednog realnog oblaka tačaka, čime je dokazana njegova mogućnost prepoznavanja elipsoida, eliptičnih cilindara, eliptičnih torusa, elipsoida ali i cilindara. Takođe, prikazani su i rezultati prepoznavanja u vidu histograma pozicija prelaza i estimiranih parametara prepoznatih površi.

Treća metoda (poglavlje 7) obezbeđuje robusno prepoznavanje cilindara i ravni u nestruktuiranim oblacima tačaka koji reprezentuju tipične objekte u mašinskom inženjerstvu. Osnov metode predstavljaju algoritmi segmentacije i robusnog prepoznavanja cilindara. Segmentacija oblaka je bazirana na osobinama singularnosti informacione matrice tokom regresije opšte površi drugog reda metodom najmanjih kvadrata. Metoda je verifikovana upotrebom tri realna oblaka tačaka. Kao značajan deo eksperimentalne verifikacije izvršena je uporedna analiza razvijene metode i dve referentne metode drugih autora. Za potrebu poređenja njihovih performansi kreirana je posebna metodologija bazirana na vrednostima devijacija parametara prepoznatih primitiva. Rezultati ove uporedne analize su pokazali značajan potencijal pomenute tehnike prepoznavanja.

Kako bi se razmotrila mogućnost povećanja autonomije razvijenih automatskih algoritama prepoznavanja primitiva, sprovedena su i dodatna istraživanja na temu mogućnosti upotrebe tehnika veštačke inteligencije. Poslednja metoda prezentovana u ovoj disertaciji (poglavlje 8) odnosi se na upotrebu unapred obučениh veštačkih neuronskih mreža za potrebe generisanja i podešavanja parametara prethodno pomenutog algoritma prepoznavanja jedne vrste kvadraka iz struktuiranih oblaka tačaka. Na ovaj način obezbeđen je potpuno autonoman algoritam koji je u procesu verifikacije korišćenjem realnog oblaka pokazao značajno unapređenje performansi polaznog metoda pri prepoznavanju cilindara.

Pored navedenih istraživanja i razvijenih metoda, u disertaciji su razmatrane tehnike struktuiranja oblaka i predstavljena je metodologija struktuiranja upotrebom z bafer algoritma (poglavlje 4). Korišćenjem ove metodologije moguće je nestruktuirane oblake prevesti u struktuiranu formu, pogodnu za obradu nekom od tehnika namenjenih procesiranju struktuiranih oblaka. Takođe, pomenuta metoda struktuiranja može izvršiti sintetizovanje oblaka tačaka na osnovu postojećeg CAD modela objekta. Navedeni procesi korišćeni su za potrebe nekoliko eksperimentalnih verifikacija algoritama prepoznavanja iz ove disertacije.

Potrebno je naglasiti da su svi razvijeni algoritmi implementirani i verifikovani u Matlab programskom paketu. Takođe, i ostale metode drugih autora koje su upotrebljene za uporednu analizu su na ovaj način uvedene u upotrebu. Simulacioni i eksperimentalni rezultati predstavljeni u disertaciji su potvrdili postavljene polazne hipoteze.

Ključne reči: reverzno inženjerstvo, procesiranje oblaka tačaka, prepoznavanje geometrijskih primitiva, segmentacija.

Naučna oblast: Mašinsko inženjerstvo

Uža naučna oblast: Proizvodno mašinstvo

UDK: 004.93:62(043.3)

RECOGNITION OF GEOMETRIC PRIMITIVES FROM THREE-DIMENSIONAL POINT CLOUD

Abstract

Intensive development of reverse engineering techniques has brought significant benefits to the mechanical industry. In addition to the traditional utilization in product design, these techniques are increasingly employed in the automation of production processes. However, their use in most real-time applications is limited due to required significant user assistance. In accordance with this limitation and the need to overcome it, there is an increase in research and development of algorithms that can be used to automatically process data in reverse engineering. This doctoral dissertation is focused on research in the field of automatic processing of data obtained by digitization. As the digitization processes in the mechanical engineering are often reduced to various forms of 3D scanning, the research area refers to the point cloud processing.

The dissertation covers three topics related to the recognition of scanned mechanical objects' structure. The first topic is focused on intelligent point cloud simplification techniques. The second topic refers to point cloud processing in order to recognize geometric primitives of observed objects, while the last topic deals with the application of artificial intelligence and machine learning for increasing of primitive recognition methods autonomy.

The doctoral dissertation also includes the development and experimental verification of four novel methods (algorithms). The first method (chapter 5) refers to the structured point clouds and employs support vector regression to identify and retain points with high importance for the reconstruction of the scanned objects geometry. The dissertation describes main regression parameters and their influence on point cloud reduction with high level of initial information preservation. The developed simplification method was experimentally verified using three real point clouds that represent objects with characteristic geometry. Also, utilizing the developed NURBS (non-uniform rational B-spline) methodology, the quality of initial geometry preservation after reduction process was examined and usable conclusions were drawn. The performance of the proposed method was compared with uniform sampling method as well.

The second developed method (chapter 6) provides recognition of one type of second degree surfaces (quadrics) from structured point clouds. The method is based on the scanned contours segmentation using the singularity of the scatter matrix during the least squares ellipses regression. In addition to the detailed presentation of the recognition algorithm, its verification was performed using several synthesized and one real point cloud. This experimental verification proved methods ability to recognize ellipsoids, elliptical cylinders, elliptical torii, ellipsoids and cylinders. Furthermore, the recognition results in the form of histograms of transition points and estimated parameters of recognized surfaces are presented.

The third method (chapter 7) provides robust recognition of cylinders and planes in unstructured point clouds that represent typical objects in the mechanical engineering. The method is based on algorithms for segmentation and robust cylinder recognition. Point cloud segmentation employs properties of scatter matrix during least squares regression of general second-order surface. The method is verified using three real point clouds. As a significant part of the experimental verification, a comparative analysis of the developed method and two reference methods of other authors were performed. To compare their performance, a special methodology based on the deviations of the recognized primitives' parameters is created. The results of this comparative analysis showed a significant potential of the proposed recognition technique.

In order to consider the possibility of increasing the autonomy of the developed automatic primitive recognition algorithms, an additional research on the possibility to use artificial intelligence tech-

niques is done. The last presented method (chapter 8) exploits pre-trained artificial neural networks for generating and adjusting the parameters of the previously described algorithm for recognizing one type of quadrics from structured point clouds. In that way, a completely autonomous process is provided. The experimental verification results showed a significant improvement in the performance of the initial method in the case of recognition cylinders in real point cloud.

In addition to the mentioned developed methods, the dissertation refers to point cloud structuring techniques and presents one structuring methodology that employs the z buffer algorithm (chapter 4). This methodology converts unstructured point clouds into structured which are appropriate for processing using a wide range of techniques intended for structured clouds. Also, the mentioned structuring process can synthesize point cloud from the existing CAD model. Described processes were used in several experimental verifications in this dissertation.

It should be emphasized that all developed algorithms are implemented and verified in the Matlab software environment. All other methods that were used for comparative analysis were also introduced in this way. Simulation and experimental results presented in the dissertation confirmed initial hypotheses.

Keywords: reverse engineering, point cloud processing, recognition of geometric primitives, segmentation.

Scientific discipline: Mechanical engineering

Scientific subdiscipline: Production engineering

UDC: 004.93:62(043.3)

Sadržaj

1	Uvod	1
1.1	Predmet istraživanja	1
1.2	Cilj istraživanja	1
1.3	Polazne hipoteze i naučni doprinos	2
1.4	Struktura rada	2
2	Reverzno inženjerstvo geometrijskih modela	4
2.1	Faze reverznog inženjerstva geometrijskih modela	4
2.2	Tehnike za 3D digitalizaciju objekata	5
2.3	Oblak tačaka	11
2.4	Pregled primena metoda za 3D digitalizaciju objekata u mašinskoj industriji	13
3	Pregled postojećih metoda za procesiranje 3D oblaka tačaka	16
3.1	Metode za redukciju broja tačaka u okviru oblaka tačaka	16
3.2	Metode za segmentaciju i prepoznavanje geometrijskih primitiva iz oblaka tačaka	20
4	Strukturiranje oblaka tačaka	26
5	Redukcija broja tačaka u okviru 3D oblaka	30
5.1	Regresija nosećim vektorima	30
5.2	Definicija metode i algoritma	33
5.3	Eksperimentalna verifikacija metode	34
5.3.1	Studija slučaja 1: Stanford zmaj	34
5.3.2	Studija slučaja 2: Lopatica turbine	41
5.3.3	Studija slučaja 3: Objekat sa ostrim ivicama	44
5.4	Ocena kvaliteta predložene metode redukcije	45
6	Prepoznavanje jedne klase površi drugog reda iz strukturiranih oblaka tačaka	50
6.1	Segmentacija eliptičnih regiona iz skeniranih linija	50
6.2	Definicija metode i algoritma	51
6.3	Verifikacija metode simulacijom	54
6.3.1	Segmentacija i prepoznavanje elipsi u konturi	54
6.3.2	Segmentacija i prepoznavanje eliptičnih cilindara	56
6.3.3	Segmentacija i prepoznavanje elipsoida	57
6.3.4	Segmentacija i prepoznavanje eliptičnih torusa	59
6.4	Eksperimentalna verifikacija metode	59
7	Prepoznavanje cilindara i ravni iz nestrukturiranih oblaka tačaka	63
7.1	Struktura metode i algoritma	63
7.2	Segmentacija	64
7.3	Algoritam spajanja i estimacija parametara površina	68
7.3.1	Modifikovani proces robusnog prepoznavanja cilindara iz oblaka tačaka	68
7.3.2	Procena parametara ravni	71
7.3.3	Definicija algoritma spajanja	71
7.4	Ekstrakcija cilindara i ravni	74
7.5	Eksperimentalna verifikacija metode	76
7.5.1	Prepoznavanje primitiva test objekta	76
7.5.2	Prepoznavanje primitiva vratila	80

7.5.3	Prepoznavanje primitiva prirubnice	83
8	Optimizacija parametara algoritama za prepoznavanje geometrijskih primitiva	87
8.1	Uticaj pragova na ishod rezultata segmentacije	87
8.2	Određivanje pragova segmentacije upotrebom veštačkih neuronskih mreža	89
8.3	Eksperimentalna verifikacija metode	93
9	Zaključak	95
10	Literatura	98

Spisak slika

2.1	Faze reverznog inženjerstva geometrijskih modela [4].	4
2.2	Podela kontaktnih metoda 3D skeniranja.	6
2.3	Podela beskontaktnih metoda 3D skeniranja.	7
2.4	Princip linijskog skenera sa struktuiranom svetlošću [2].	8
2.5	Ilustracija principa više-linijskog skenera sa struktuiranom svetlošću [23].	9
2.6	Primer optičkog industrijskog skenera - Atos Compact Scan 5M [24].	10
2.7	Vrste oblaka tačaka prema strukturi: a) izgled površine objekta; b) struktuirani oblak tačaka; c) nestruktuirani oblak tačaka.	12
2.8	Razlika između 3D oblaka tačaka i mreže: a) oblak tačaka; b) poligonalna (trougona) mreža nastala procesiranjem oblaka.	12
4.1	Metoda struktuiranja oblaka tačaka upotrebom paralelnih ravni u zadatom rasteru [72].	26
4.2	Grafički prikaz struktuiranja nestruktuiranih oblaka tačaka korišćenjem z bafer metodologije.	27
4.3	Pseudokod algoritma struktuiranja oblaka tačaka korišćenjem z bafer metodologije. .	28
4.4	Grafički prikaz sintetizovanja oblaka tačaka korišćenjem STL modela objekta i z bafer metodologije	29
5.1	Grafički prikaz modela ε -SVR [114].	30
5.2	Ilustracija primene ε -SVR nad skeniranom konturom.	33
5.3	Stanford zmaj: a) 3D mreža; b) struktuirana oblast od interesa.	35
5.4	Struktuirani oblak rastera 0,05 mm duž x ose : a) struktuirani oblak; b) izdvojena kontura oblaka.	35
5.5	Efekti izbora kernela na redukciju: a) B-splajn kernel: $\varepsilon = 0,15, C = 10$, procenat redukcije 52,8%; b) splajn kernel: $\varepsilon = 0,01, C = 10$, procenat redukcije 49,0%. . . .	36
5.6	Efekti vrednosti parametra ε na redukciju: a) B-splajn kernel; b) splajn kernel. . . .	37
5.7	Efekti vrednosti parametra C na redukciju: a) B-splajn kernel; b) splajn kernel. . . .	37
5.8	ε -SVR izdvojene konture rastera 0,1 mm: a) B-splajn kernel: $\varepsilon = 0,15, C = 10$, procenat redukcije 54,9%; b) splajn kernel: $\varepsilon = 0,01, C = 10$, procenat redukcije 62,4%.	39
5.9	Redukcija kompletnog struktuiranog oblaka leđa zmaja uniformnog rastera 0,1 mm: a) B-splajn kernel; b) splajn kernel.	40
5.10	Lopatica turbine: a) fotografija realnog objekta; b) struktuirani oblak tačaka.	41
5.11	ε -SVR izdvojene konture lopatice: $\varepsilon = 0,01, C = 1.000$, splajn kernel - noseći vektori i dodatne tačke na $d_{max} = 5$ mm.	41
5.12	ε -SVR izdvojene konture lopatice: a) B-splajn kernel, $\varepsilon = 0,15$; b) splajn kernel, $\varepsilon = 0,01$	42
5.13	Redukcija kompletnog oblaka lopatice: a) B-splajn kernel, $\varepsilon = 0,05$; b) splajn kernel, $\varepsilon = 0,002$	43
5.14	Test objekat: a) fotografija objekta; b) sken objekta iz jednog pravca; c) struktuirani oblak.	44
5.15	Redukcija kompletnog oblaka test objekta: a) jedna izdvojena kontura; b) redukovani oblak upoređen sa ivicama objekta u pogledu odozgo; c) redukovani oblak u izometriji.	45
5.16	NURBS kreiran na osnovu redukovane konture ($\varepsilon = 0,025, C = 1.000$, splajn kernel) zajedno sa NURBS-om kreiranim na osnovu inicijalne konture sa slike 5.5.	46
5.17	Komparacija predložene SVR metode i uniformne redukcije na primeru konture sa slike 5.5.	47
5.18	NURBS kreiran na osnovu redukovane konture ($\varepsilon = 0,05, C = 10.000$, splajn kernel) zajedno sa NURBS-om kreiranim na osnovu inicijalne konture sa slike 5.11.	48

5.19	Komparacija predložene SVR metode i uniformne redukcije na primeru konture sa slike 5.11.	49
6.1	Pseudokod jedne iteracije algoritma segmentacije skeniranih kontura.	52
6.2	Pseudokod procesa grupisanja eliptičnih segmenata.	53
6.3	Mogući pravci skeniranja određenih kvadirka: a) skeniranje cilindra duž ose i upravno na nju; b) skeniranje cilindra u pravcima pod uglom u odnosu na osu; c) skeniranje eliptičnog torusa.	53
6.4	Prva sintetizovana kontura sa tri eliptična segmenta.	54
6.5	Druga sintetizovana kontura sa tri eliptična segmenta.	56
6.6	Površina sačinjena od eliptičnih cilindara: a) 3D model; b) sintetizovani oblak dobijen longitudinalnim struktuiranjem (skeniranjem); c) sintetizovani oblak dobijen struktuiranjem pod uglom.	56
6.7	Grafički rezultati segmentacije površine sačinjene od eliptičnih cilindara segmentirane duž x ose: a) oblak tačaka sa slike 6.6(b); b) oblak tačaka sa slike 6.6(c).	57
6.8	Površina sačinjena od sferoida: a) 3D model; b) sintetizovani oblak tačaka.	58
6.9	Grafički rezultati segmentacije površine sačinjene od sferoida: a) segmentacija duž x ose; b) segmentacija duž y ose.	58
6.10	Površina sačinjena od torusa: a) 3D model; b) sintetizovani oblak tačaka.	59
6.11	Grafički rezultati segmentacije površine sačinjene od torusa: a) segmentacija duž x ose; b) segmentacija duž y ose; c) konačni rezultati segmentacije.	60
6.12	Histogrami detektovanih prelaznih tačaka oblaka sa slike 6.11 : a) prelazne tačke duž x ose; b) prelazne tačke duž y ose.	60
6.13	Segmentacija realnog oblaka koji reprezentuje površinu sačinjenu od cilindara: a) površina od interesa označena na test objektu sa slike 5.14(a); b) rezultati segmentacije izdvojene površine duž x ose.	61
6.14	Histogrami detektovanih prelaza sa slike 6.13(b): a) između segmenata 1 i 2; b) između segmenata 2 i 3; c) između segmenata 3 i 4; d) između segmenata 4 i 5.	61
7.1	Struktura predložene metode prepoznavanja cilindara i ravni iz nestruktuiranih oblaka tačaka.	64
7.2	Uklanjanje malih regiona: ilustracija u 2D prostoru.	66
7.3	Pseudokod jedne iteracije segmentacije oblaka tačaka	67
7.4	Procesiranje klastera koji reprezentuje cilindar upotrebom RPCA algoritma ($\alpha = 0,75$; $k = 3$): a) oblak tačaka; b) procesirani klaster.	69
7.5	Struktura algoritma spajanja klastera.	72
7.6	Pseudokod za spajanje klastera	73
7.7	Pseudokod za ekstrakciju površina	75
7.8	Test objekat: a) fotografija realnog objekta sa akcentom na region karakterizovan lošom mašinskom obradom, b) kompletan nestruktuirani oblak tačaka dobijen 3D skeniranjem objekta.	77
7.9	Grafička reprezentacija rezultata prepoznavanja primitiva iz oblaka tačaka sa slike 7.8(b): a) Predložena metoda, b) EoRP, c) eRANSAC.	78
7.10	Vratilo: a) fotografija realnog objekta, b) oblak tačaka.	81
7.11	Grafička reprezentacija rezultata prepoznavanja primitiva iz oblaka tačaka sa slike 7.10(b): a) Predložena metoda, b) EoRP, c) eRANSAC.	81
7.12	Prirubnica: a) fotografija realnog objekta, b) oblak tačaka.	83
7.13	Grafička reprezentacija rezultata prepoznavanja primitiva iz oblaka tačaka sa slike 7.12(b): a) Predložena metoda, b) EoRP, c) eRANSAC.	84
8.1	Sintetizovana površina koja se sastoji od G1 kontinualna tri cilindra: a) CAD model objekta; b) sintetizovani struktuirani oblak tačaka.	88

8.2	Segmentirana kontura oblaka sa slike 8.1: a) pragovi koje je korisnik odredio metodom pretraživanja; b) promenjeni pragovi u cilju ilustracije osetljivosti segmentacije.	89
8.3	Segmentirane reprezentativne konture za obuku veštačkih neuronskih mreža.	90
8.4	Rezultati segmentacije sintetizovane konture upotrebom algoritma sa automatskim izborom pragova.	92
8.5	Rezultati segmentacije sintetizovanog oblaka tačaka upotrebom algoritma sa automatskim izborom pragova.	93
8.6	Rezultati segmentacije realnog oblaka tačaka upotrebom algoritma sa automatskim izborom pragova.	93
8.7	Histogrami detektovanih prelaza sa slike 8.6: a) između segmenata 1 i 2; b) između segmenata 2 i 3; c) između segmenata 3 i 4; d) između segmenata 4 i 5.	94

Spisak tabela

5.1	Pregled rezultata procesiranja u slučaju konture zmaja sa različitim brojem tačaka.	39
5.2	Rezultati redukcije kompletnog oblaka leđa zmaja uniformnog rastera 0,1 mm.	40
5.3	Rezultati redukcije kompletnog oblaka lopatice.	44
5.4	Estimacija kvaliteta regresije konture sa slike 5.5.	46
5.5	Uniformna redukcija broja tačaka konture sa slike 5.5.	47
5.6	Estimacija kvaliteta regresije konture sa slike 5.11.	48
5.7	Uniformna redukcija broja tačaka konture sa slike 5.11.	49
6.1	Prva sintetizovana kontura - parametri i granice segmenata.	55
6.2	Rezultati segmentacije prve sintetizovane konture.	55
6.3	Druga sintetizovana kontura - parametri i granice segmenata.	55
6.4	Rezultati segmentacije druge sintetizovane konture.	56
6.5	Estimirani parametri prepoznatih eliptičnih cilindara.	57
6.6	Sintetizovani i estimirani parametri prepoznatih sferoida.	58
6.7	Procenjene vrednosti parametara prepoznatih cilindara.	60
7.1	Predložena metoda: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.8(b)	77
7.2	EoRP: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.8(b)	77
7.3	Estimirani radijusi cilindara [mm] za oblak tačaka sa slike 7.8(b)	79
7.4	Estimirane normale ravni za oblak tačaka sa slike 7.8(b)	79
7.5	Predložena metoda: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.10(b)	82
7.6	EoRP: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.10(b)	82
7.7	Estimirani radijusi cilindara [mm] za oblak tačaka sa slike 7.10(b)	82
7.8	Estimirane normale ravni za oblak tačaka sa slike 7.10(b) (vratilo bez kanala za prstenasti uskočnik)	82
7.9	Predložena metoda: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.12(b)	85
7.10	EoRP: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.12(b)	85
7.11	Estimirani radijusi cilindara [mm] za oblak tačaka sa slike 7.12(b)	85
7.12	Estimirane normale ravni za oblak tačaka sa slike 7.12(b)	85
8.1	Rezultati segmentacije sintetizovnog struktuiranog oblaka tačaka upotrebom neuronskih mreža.	92
8.2	Rezultati segmentacije realnog struktuiranog oblaka tačaka upotrebom neuronske mreže.	93

Spisak oznaka

α, β	pridruženi parametri
ϵ	procenat tačkaka koje predstavljaju grešku skeniranja
$\lambda, \lambda_i, \lambda_i^*$	Lagranžovi množioc
$\lambda_0, \lambda_1, \lambda_2$	sopstvene vrednosti
c	centar bazne kružnice cilindra
DN_i	dizajnirani pravac normale
EN_i	estimirani pravac normale
e	jedinični vektor normale ravni
n	vektor normale ravni
p	vektor ose cilindra
v₀, v₁, v₂	sopstveni vektori
σ	srednja gustina oblaka tačkaka
a	vektor parametara opšte jednačine elipse/kvadrice
A₁, A₂, A₃, A₄	matrice za izračunavanje prodora prave kroz trougao
C	matrica uslova regresije elipse/kvadrice
D	matrica sistema
I	ulazna matrica obučavanja mreža
i, j, k	jedinični vektori Dekartovog koordinatnog sistema
O	izlazna matrica obučavanja mreža
P_i	tačka
S	informaciona matrica sistema
u	vektor čvorova
v	vektor posmatranja
x	vektor promenljivih
ϵ	margina dozvoljene greške regresije (SVR parametar)
ξ_i, ξ_i^*	promenljive
a, b, c, d	parametri jednačine ravni
a_0, b_0	koordinate centra inicijalne kružnice
a_i	parametri opšte jednačine elipse/kvadrice
AND	srednje odstupanje normala (engl. <i>Average Normal Deviation</i>)
ARD	srednje odstupanje vrednosti radijusa (engl. <i>Average Radius Deviation</i>)
asr	srednje kvadratno odstupanje za h tačkaka
C	cena za grešku (SVR parametar)
c_x, c_y, c_z	koordinate centra bazne kružnice cilindra

D	distanca između tačke i ravni
D_v	parametar pravca najkraćeg rastojanja između tačke i ravni
d_{max}	maksimalno rastojanje između tačaka konture
dev_a	srednje apsolutno odstupanje dve NURBS krive
dev_m	maksimalno apsolutno odstupanje dve NURBS krive
dev_t	ukupno kvadratno odstupanje dve NURBS krive
DR_i	dizajnirana vrednost radijusa cilindra
e_i^2	kvadratna odstupanja (reziduali)
ER_i	estimirana vrednost radijusa cilindra
$f(x)$	linearna funkcija
h	broj $n/2$ tačaka sa najmanjim kvadratnim odstupanjima
h_{sr}	srednja visina konture
I_n	broj iteracija
K	kernel
k	faktor redukcije
m	ukupan broj tačaka izabranog rastera za poređenje NURBS kriva
m_c	minimalni broj tačaka klastera
m_{ic}	minimalni broj tačaka inicijalnog klastera
N	broj tačaka za obučavanje mreže
n	broj tačaka
n_1, n_2	brojevi tačaka dva susedna eliptična segmenta
n_c	broj prepoznatih cilindra
n_p	broj prepoznatih ravni
p_r	verovatnoća pronalaženja minimalnog skupa tačaka
p_x, p_y, p_z	komponente vektora ose cilindra
R	radijus cilindra
r_0	radijus inicijalne kružnice
r_j	raster za poređenje NURBS kriva
R_{ti}	najkraće rastojanje tačke do ose cilindra
S	prostor ulaznih parametara
sd	najkraće rastojanje tačke od omotača cilindra
t_{gr}	prag za grupisanje eliptičnih segmenata
t_{ov}	prag za sopstvene vrednosti pri segmentaciji eliptičnih segmenata
t_{rc}	prag za recipročni faktor uslovljenosti pri segmentaciji eliptičnih segmenata
t_{er}	prag za izdvajanje klaster

tm_r	maksimalni očekivani radijus cilindra
ts_e	prag za pripadnost tačke inicijalnom klasteru
ts_r	prag za distancu
ts_s	prag za segmentaciju površi drugog reda
u, v, t	parametri prodora prave kroz površ
W	valovitnost konture
w	kombinacija parametara učenja
x_i, y_i, z_i	koordinate tačke
x_p, y_p, z_p	koordinate tačke prodora vektora kroz trougao
y_{1i}, y_{2i}	originalne vrednosti y koordinata dva susedna eliptična segmenta
y_{e1i}, y_{e2i}	estimirane vrednosti y koordinata dva susedna eliptična segmenta

Spisak skraćenica

ε -SVR	ε -neosetljiva regresija nosećim vektorima (engl. ε -insensitive Support Vector Regression)
1D	jednodimenzionalni
2D	dvodimenzionalni
3D	trodimenzionalni
AGV	automatski vođena robokolica (engl. <i>Automated Guided Vehicle</i>)
AI	veštačka inteligencija (engl. <i>Artificial intelligence</i>)
CAD	projektovanje proizvoda upotrebom računara (engl. <i>Computer Aided Design</i>)
CARE	reverzno inženjerstvo upotrebom računara (engl. <i>Computer Aided Reverse Engineering</i>)
CCD	senzor sastavljen od fotosenzitivnih dioda (engl. <i>Charge-Coupled Device</i>)
CT	kompjuterska tomografija (engl. <i>Computed Tomography</i>)
CWT	kontinualna vejevlet transformacija (engl. <i>Continuous Wavelet Transform</i>)
DWT	diskretna vejevlet transformacija (engl. <i>Discrete Wavelet Transform</i>)
EoRP	ekstrakcija značajnih primitiva(engl. <i>Extraction of Reliable Primitives</i>)
eRANSAC	efikasni konsenzus za slučajno odabrane uzorke (engl. <i>efficient RANdom SAmple Consensus</i>)
fdn	algoritam suseda na definisanoj distanci (engl. <i>fixed distance neighbors</i>)
HT	Hofova transformacija (engl. <i>Hough Transform</i>)
KMM	Koordinatna Merna Mašina
knn	algoritam k najbližih suseda (engl. <i>k-nearest neighbors</i>)
LED	dioda koja emituje svetlost (engl. <i>Light-Emitting Diode</i>)
MRI	magnetna rezonanca (engl. <i>Magnetic Resonance Imaging</i>)
NUMA	Numerički Upravljana Mašina Alatka
NURBS	Neuniformni racionalni B-splajn (engl. <i>Non-Uniform Rational B-Spline</i>)
PCA	analiza glavnih komponenti sistema (engl. <i>Principal Component Analysis</i>)
RANSAC	konsenzus za slučajno odabrane uzorke (engl. <i>RANdom SAmple Consensus</i>)
RCF	robustno prepoznavanje cilindara (engl. <i>Robust Cylinder Fitting</i>)
RE	reverzno inženjerstvo (engl. <i>Reverse Engineering</i>)
RPCA	robustna analiza glavnih komponenti sistema (engl. <i>Robust Principal Component Analysis</i>)
SfM	generisanje strukture na osnovu kretanja (engl. <i>Structure from Motion</i>)
SNR	odnos šuma i signala (engl. <i>Signal to Noise Ratio</i>)
STL	fajl format namenjen stereolitografiji (engl. <i>Standard Triangle Language</i>)
SVM	mašine sa nosećim vekotrima (engl. <i>Support Vector Machines</i>)
SVR	regresija nosećim vekotrima (engl. <i>Support Vector Regression</i>)

1. Uvod

1.1. Predmet istraživanja

U savremenom projektovanju proizvoda pored klasičnog pristupa (engl. *Forward Engineering*) sve češće se primenjuju tehnike reverznog inženjerstva (engl. *Reverse Engineering*). Za razliku od konvencionalnog metoda projektovanja pristupom od ideje ka realizaciji, u reverznom inženjerstvu se primenjuje princip suprotnog smera - od fizičkog objekta ka njegovoj digitalnoj verziji.

Na osnovu opšteprihvaćene definicije reverzno inženjerstvo (RE) predstavlja proces analize posmatranog objekta u cilju identifikacije njegovih komponenti i njihovog međusobnog odnosa, kao i kreiranja reprezentacije objekta u nekoj drugoj formi ili na višem nivou apstrakcije [1]. Prema autorima definicije, RE ne obuhvata kreiranje novog (sličnog) objekta na bazi posmatranog. Ipak, jasno je da je upravo ta potreba u velikoj meri ubrzala razvoj ove oblasti i umnogome je oblikovala.

Iz industrijske perspektive reverzno inženjerstvo se odnosi na kreiranje CAD modela objekta za koji tehnička ili proizvodna dokumentacija nije dostupna [2]. Ovako dobijen model se može koristiti u različite svrhe: za proizvodnju kopije fizičkog objekta (koncept 3D replikatora), povratak izgubljenog sadržaja u smislu mogućnosti ponovne izrade različitih vrsta dokumentacije (engl. *Redocumentation, Design Recovery*), kontrolu izrađenih delova složene geometrije, buduće unapređenje izgleda i performansi objekta (reinženjering) itd. Takođe, značajna je i upotreba RE u automatizaciji velikog broja proizvodnih procesa.

Kao glavni procesi RE, izdvajaju se trodimenzionalna digitalizacija i (pre-)procesiranje podataka dobijenih digitalizacijom, a sve u cilju kreiranja potrebne (kompjuterske) reprezentacije objekta od interesa [3]. Digitalizacija predstavlja skup tehnika i procesa kojima se vrši prikupljanje podataka o (pre svega) geometriji površine posmatranog objekta. Ove informacije se inicijalno zapisuju u formi 3D koordinata tačaka detektovanih na površinama objekta. Ovakvi skupovi tačaka se zbog svoje forme tradicionalno nazivaju oblacima tačaka (engl. *point clouds*), dok se tehnike kojima se vrši njihovo generisanje (ispitivanjem realnog objekta) nazivaju tehnikama skeniranja. U skladu sa navedenim, procesiranje podataka 3D digitalizacije nazivamo procesiranjem oblaka tačaka.

Predmet istraživanja ove disertacije odnosi se upravo na različite vrste tehnika procesiranja 3D oblaka tačaka dobijenih skeniranjem. U fokusu istraživanja biće pre-procesiranje oblaka i to pre svega tehnike za inteligentnu redukciju broja tačaka u oblaku koje treba da obezbede smanjenje količine informacija uz istovremeno zadržavanje informacionog sadržaja. Pored toga, istraživaće se i tehnike za prepoznavanje određenih geometrijskih primitiva iz 3D oblaka tačaka.

1.2. Cilj istraživanja

Upotrebom računara za sprovođenje procesa RE nastao je koncept tzv. kompjuterski podržanog reverznog inženjerstva (CARE) [2]. U okviru ovog koncepta prisutno je postojanje velikog broja softvera koji su namenjeni, sa jedne strane procesu skeniranja, a sa druge strane, procesiranju podataka dobijenih skeniranjem. Algoritmi i tehnike namenjene procesiranju oblaka tačaka su često integrisane u komercijalne CAD softvere čineći na taj način njihove posebne module namenjene RE. Međutim, procesi prevođenja oblaka tačaka u kompjuterske modele upotrebom trenutno dostupnih tehnika i softverskih rešenja zahtevaju značajnu asistenciju samog korisnika. Ovaj zahtev umnogome otežava svakodnevni rad korisnika, ali takođe i limitira implementaciju tehnika RE za rad u realnom vremenu za potrebe u drugim oblastima mašinskog inženjerstva.

Zbog navedene činjenice o značajnom udelu korisničke asistencije u procesima RE, razvoj potpuno automatskih metoda procesiranja oblaka tačaka predstavlja značajan savremeni istraživački izazov. Cilj ove doktorske disertacije predstavlja istraživanje u oblasti obrade oblaka tačaka i on obuhvata:

- razvoj metoda za redukciju broja tačaka iz oblaka uz istovremeno zadržavanje informacionog

sadržaja,

- razvoj metoda za prepoznavanje geometrijskih primitiva iz strukturiranih i nestruktuiranih oblaka tačaka,
- verifikaciju razvijenih metoda na generisanim oblacima tačaka,
- eksperimentalnu verifikaciju razvijenih metoda na oblacima tačaka dobijenim skeniranjem reprezentativnih objekata.

1.3. Polazne hipoteze i naučni doprinos

Uzimajući u obzir sve izraženije potrebe industrije za primenom sistema trodimenzionalne (3D) digitalizacije objekata u upravljanju resursima kao i analize trenutnog stanja u oblastima obrade 3D oblaka tačaka (3D signala), mašinskog učenja i veštačke inteligencije, u okviru ove disertacije definišu se sledeće polazne hipoteze:

- **Prva hipoteza:** Primenom tehnika obrade signala i mašinskog učenja (prvenstveno mašina sa nosećim vektorima) moguće je formirati nove robusne metode i algoritme za automatsko redukovanje broja tačaka u okviru trodimenzionalnog oblaka tačaka uz istovremeno zadržavanje neophodnog informacionog sadržaja.
- **Druga hipoteza:** Tehnike obrade signala, analitičke i statističke metode i mašinsko učenje mogu se efikasno upotrebiti za razvoj novih robusnih metoda za automatsko prepoznavanje geometrijskih primitiva iz trodimenzionalnog oblaka tačaka; na osnovu prepoznatih geometrijskih primitiva moguće je izvršiti parametarsku reprezentaciju modela digitalizovanih objekata.
- **Treća hipoteza:** Tehnike veštačke inteligencije predstavljaju dovoljnu osnovu za kreiranje inteligentnih metoda za automatsko podešavanje potrebnih parametara prilikom prepoznavanja geometrijskih primitiva iz trodimenzionalnog oblaka tačaka.

1.4. Struktura rada

Tekst doktorske disertacije je strukturiran u devet osnovnih poglavlja u kojima su predstavljena istraživanja, opservacije i doprinos izabranoj naučnoj oblasti. Pored navedenih devet, u desetom poglavlju je dat i spisak korišćene literature.

U uvodnom poglavlju prezentovana je problematika kojom se disertacija bavi, a koja je vezana za oblast procesiranja oblaka tačaka u cilju redukcije broja tačaka i automatskog prepoznavanja gradivnih primitiva skeniranih objekata. Takođe, u ovom poglavlju date su polazne hipoteze koje su tokom istraživanja i dokazane.

U drugom poglavlju predstavljeni su koncept i naučni okvir reverznog inženjerstva geometrijskih modela zajedno sa pripadajućim tehnikama za 3D digitalizaciju objekata i njihovom primenom u mašinskoj industriji. Pored toga, dati su praktični aspekti 3D digitalne reprezentacije realnih objekata upotrebom oblaka tačaka, kao i njihova struktura.

U trećem poglavlju prezentovan je pregled relevantnih postojećih metoda iz oblasti procesiranja 3D oblaka tačaka i to: tehnika za kreiranje 3D poligonalnih mreža, tehnika za redukciju broja tačaka oblaka i metoda za segmentaciju i prepoznavanje geometrijskih primitiva iz oblaka.

Četvrto poglavlje se odnosi na oblast strukturiranja oblaka tačaka i u okviru njega se razmatraju neke od postojećih metoda, ali i predstavlja jedan drugačiji pristup.

Nakon četvrtog poglavlja sledi niz poglavlja u kojima se predstavlja nekoliko originalnih metoda za procesiranje oblaka tačaka koje su razvijene u okviru ove disertacije, kao i istraživanja na temu njihove primene u realnom okruženju. Razvijene metode predstavljaju osnovni naučni doprinos ove disertacije i mogu predstavljati osnov za potencijalno pronalaženje rešenja nekih od trenutno defini-

sanih problema u oblasti procesiranja oblaka tačaka kao i primene tehnika reverznog inženjerstva u mašinskoj industriji.

Poglavlje pet se bavi problemom redukcije broja tačaka struktuiranih oblaka sa visokim stepenom održanja informacionog sadržaja. Za ovu namenu kreirana je metoda redukcije zasnovana na upotrebi mašina sa nosećim vektorima (tehnika mašinskog učenja). Takođe, prikazani su određeni rezultati primene ove metode u tri realna slučaja, ali i njegova uporedna analiza sa tehnikom uniformne redukcije na osnovu definisanih ocena kvaliteta.

U šestom poglavlju prezentovana je originalna metoda za prepoznavanje jedne klase površi drugog reda (kvadrata) iz struktuiranih oblaka tačaka. Date su osnovne teorijske postavke korišćenih tehnika i prezentovana je kompletna strategija metode. Prikazani su rezultati verifikacije predložene metode pri njenoj primeni nad znatnom broju simuliranih oblaka tačaka, ali i oblaka tačaka dobijenih trodimenzionalnim skeniranjem realnih objekata.

U sedmom poglavlju se predstavlja jedna robusna metoda prepoznavanja dva najfrekventnije korišćena primitiva u mašinstvu - cilindara i ravni. Predstavljeni su svi razvijeni algoritmi za ovu namenu kao i rezultati njihove upotrebe na realnim oblacima tačaka. Metoda za prepoznavanje cilindara i ravni je upoređena sa dve postojeće metode.

Osmo poglavlje pokazuje mogući način optimizacije parametara algoritma prepoznavanja geometrijskih primitiva upotrebom određenih tehnika veštačke inteligencije. U konkretnom slučaju predstavlja se unapređenje metode iz šestog poglavlja upotrebom veštačkih neuronskih mreža za potrebe automatskog određivanja pragova korišćenih u algoritmu.

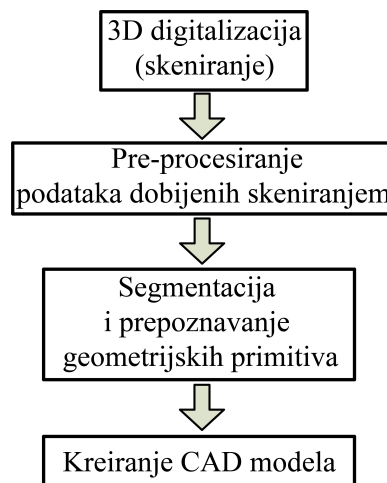
Na samom kraju, u devetom poglavlju, izvršena je diskusija rezultata upotrebe svih metoda procesiranja oblaka tačaka razvijenih u okviru ove disertacije. Svaka od metoda je posebno analizirana, definisane su sve njihove prednosti i mane i izneti su relevantni zaključci kao i smernice za budući rad.

2. Reverzno inženjerstvo geometrijskih modela

Ukoliko u okviru šire oblasti RE posmatramo samo kreiranje CAD modela na bazi posmatranog realnog objekta onda možemo govoriti o tzv. reverznom inženjerstvu geometrijskih modela. Cilj ovog procesa je generisanje, pre svega, informacija o geometrijskoj strukturi objekta, kako bi te informacije mogle biti iskorišćene za kreiranje različitih standardnih kompjuterskih formi modela. U ovom poglavlju biće predstavljene osnovne faze reverznog inženjerstva geometrijskih modela. Takođe, biće reči i o oblacima tačaka, kao i o široj upotrebi procesa 3D digitalizacije u mašinskoj industriji.

2.1. Faze reverznog inženjerstva geometrijskih modela

Proces reverznog inženjerstva geometrijskih modela se prema [4] može podeliti na četiri faze (slika 2.1). Prva faza procesa predstavlja već pomenutu 3D digitalizaciju, odnosno generisanje informacija o objektu u formi, pre svega, oblaka tačaka. Tehnike 3D digitalizacije kao i vrste oblaka tačaka biće detaljno objašnjene u narednim segmentima ovog poglavlja.



Slika 2.1: Faze reverznog inženjerstva geometrijskih modela [4].

Pre-procesiranje oblaka tačaka predstavlja drugu fazu. U ovoj fazi pristupa se izvođenju različitih procesa koji obezbeđuju lakše sprovođenje narednih faza. Savremeni 3D skeneri obezbeđuju veoma precizne i guste oblake tačaka koji veoma često sadrže i previše redundantnih informacija o samoj geometriji skeniranog objekta [3, 5]. Ovakvi oblaci mogu biti veoma otežavajući faktor za dalje procese RE, pogotovo kod tehnika koje zahtevaju značajne računarske i vremenske resurse. Zbog toga se u ovoj fazi najčešće pribegava različitim tehnikama redukovanja broja tačaka polaznog (sirovog) oblaka (engl. *point cloud simplification*). Potrebno je pomenuti i da se pored redukcije broja tačaka u određenim situacijama vrši i proširenje polaznog oblaka novim tačkama radi obezbeđivanja boljeg prepoznavanja informacije o geometriji skeniranog objekta u narednim fazama. Takođe, u ovoj fazi se po potrebi primenjuju i različite tehnike tzv. struktuiranja polaznog oblaka u cilju obezbeđivanja boljeg, pre svega geometrijskog, rasporeda tačaka čime se omogućava i olakšava primena tehnika procesiranja u sledećoj fazi.

Kao poseban deo faze pre-procesiranja oblaka tačaka izdvajaju se razne tehnike registracije i integracije oblaka, ali i tehnike formiranja tzv. poligonalnih mreža (engl. *3D meshing*) čijom se upotrebom 3D digitalizovani objekti mogu, bez ostalih faza RE, lako prevesti u određene CAD modele. Najčešće korišćena interpolacija geometrije objekta vrši se upotrebom trouglova. Iako estetski prihvatljivi, ovakvi oblici reprezentacije ipak ne sadrže dovoljno korisnih informacija za sve potrebne primene u industriji. Zbog toga se na putu do potpuno funkcionalnog CAD modela u inženjerskoj praksi pristupa ostalim fazama RE.

Treća faza je suštinski podeljena na dva koraka: 1) segmentaciju oblaka tačaka na regione različite po svojim geometrijskim karakteristikama, i 2) prepoznavanje geometrijskih primitiva na osnovu segmentiranih regiona i estimaciju njihovih parametara. Segmentacija predstavlja jedan od najzahtevnijih koraka pri procesiranju oblaka tačaka [6]. U komercijalnim CAD softverima (u njihovim modulima za RE) segmentacija oblaka se gotovo uvek izvodi uz asistenciju samog korisnika. Najčešće korisnik mora vizuelno da prepozna različite regione u geometriji površine skeniranog objekta i da ih označi selektovanjem određenog broja tačaka na njima. Neretko korisnik treba da obezbedi i samu informaciju o tipu primitiva čije se prepoznavanje očekuje u označenoj oblasti. Nakon ove korisnikove intervencije softver će biti u mogućnosti da sam (automatski) izvrši drugi korak odnosno da prepozna traženi primitiv, kao i da proceni njegove parametre. Estimacija parametara primitiva se obično izvodi korišćenjem nekog od matematičkih metoda regresije. Imajući u vidu navedeno, lako je primetiti da razvoj poptuno automatskih procesa segmentacije i prepoznavanja geometrijskih primitiva predstavlja veliki izazov savremenih istraživanja.

Nakon prepoznavanja geometrijskih primitiva od kojih je sačinjena struktura posmatranog (skeniranog) objekta, u završnoj fazi se pristupa kreiranju CAD modela u potrebnoj formi. Sama forma odnosno tip CAD modela, koji predstavlja izlazni produkt iz procesa reverznog inženjerstva geometrijskih modela, veoma zavisi od njegove namene u daljem korišćenju. Zbog toga, pri kreiranju modela, različite informacije o skeniranom objektu mogu biti kombinovane sa onima dobijenim iz faze prepoznavanja geometrijskih primitiva.

Na osnovu svega navedenog, pojam reverznog inženjerstva geometrijskih modela možemo posmatrati kao skup svih procesa koji se odnose na stvaranje potrebnog digitalnog oblika određenog predmeta samo na osnovu već postojeće fizičke realizacije istog. Dosadašnja istraživanja u ovoj oblasti dovela su do razvoja značajnog broja različitih komercijalnih rešenja. Međutim, gotovo sva ona zahtevaju asistenciju samog korisnika procesa/softvera. Zbog toga, razvoj potpuno automatskog procesa kreiranja kompjuterskog modela određenog realnog objekta je veliki izazov. Tema ove disertacije je upravo usmerena na oblast automatskog procesiranja podataka dobijenih 3D digitalizacijom (skeniranjem) objekata, a u cilju dobijanja upotrebljivih informacija potrebnih za kreiranje njihovih geometrijskih (kompjuterskih) modela sa krajnjim ciljem upotrebe ovih informacija u automatizaciji proizvodnih procesa. U ovom radu ukazaće se i na određene probleme u ovoj oblasti i biće predložena rešenja za neke od njih. Takođe, biće predstavljeni i rezultati istraživanja kao i eksperimentalnih verifikacija predloženih metoda čime će biti prikazan osnovni naučni doprinos ove doktorske disertacije.

2.2. Tehnike za 3D digitalizaciju objekata

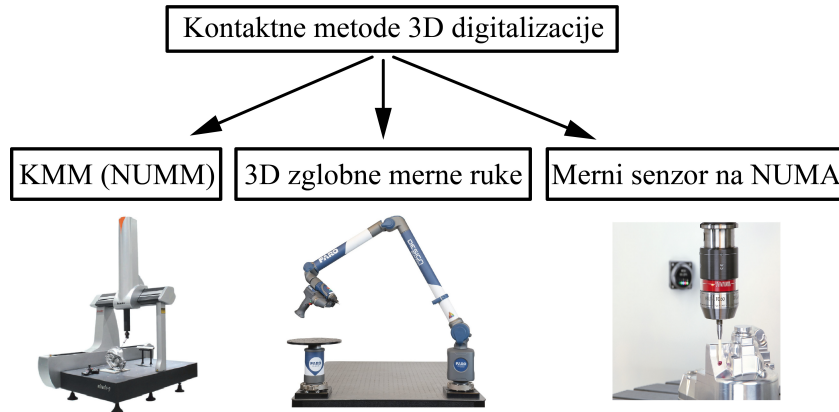
3D digitalizacijom nazivamo procese i metode akvizicije podataka o geometriji posmatranih objekata i njihovo prevođenje u računarski pogodan oblik. Ovaj oblik se naziva digitalnim pa otuda i sam naziv digitalizacija, odnosno prevođenje u digitalni oblik. Najčešći digitalni oblik koji je zastupljen u reverznom inženjerstvu je svakako oblak tačaka. Njega sačinjavaju skupovi 3D koordinata tačaka (u Dekartovom koordinatnom sistemu), koje predstavljaju određene pozicije na površinama posmatranih objekata. Zbog same prirode procesa dobijanja oblaka tačaka, 3D digitalizaciju dovodimo u vezu sa pojmom 3D skeniranja, dok se uređaji koji se koriste u ovim postupcima nazivaju 3D skenerima.

Na osnovu načina generisanja informacija o strukturi objekata, nedestruktivne metode 3D digitalizacije možemo podeliti na:

- kontaktne i
- beskontaktne.

Kontaktne metode informacije o geometriji generišu se direktnim dodiranjem (kontaktnom) između određenog senzora i površine posmatranog objekta. U praksi se sreću tri vrste kontaktnih metoda 3D digitalizacije (slika 2.2). Kao senzori kontakta najčešće se koriste tzv. merne igle (merni pipci). Merne igle mogu biti postavljene na koordinatne merne mašine (KMM), zglobove merne ruke ali i

na numerički upravljane mašine alatke (NUMA) pa otuda i podela na ove tri oblasti kontaktne 3D digitalizacije. U ovim slučajevima se mernom iglom, prema određenom planu kretanja, ostvaruju i detektuju kontakti sa površinom posmatranog objekta dok se preko mehaničkog sistema prepoznaju i generišu prostorne koordinate kontakata, odnosno tačaka na objektu. Ovakve vrste skeniranja su po pravilu vremenski zahtevne i izvode se tzv. principom od tačke do tačke. Mana je i nemogućnost skeniranja objekata sačinjenih od mekih materijala. Međutim, prednosti ovih metoda se ogledaju u visokoj preciznosti skeniranja, niskim troškovima, mogućnosti skeniranja dubokih žlebova i otvora, kao i neosetljivosti na izgled površine i loše osvetljenje [2].



Slika 2.2: Podela kontaktnih metoda 3D skeniranja.

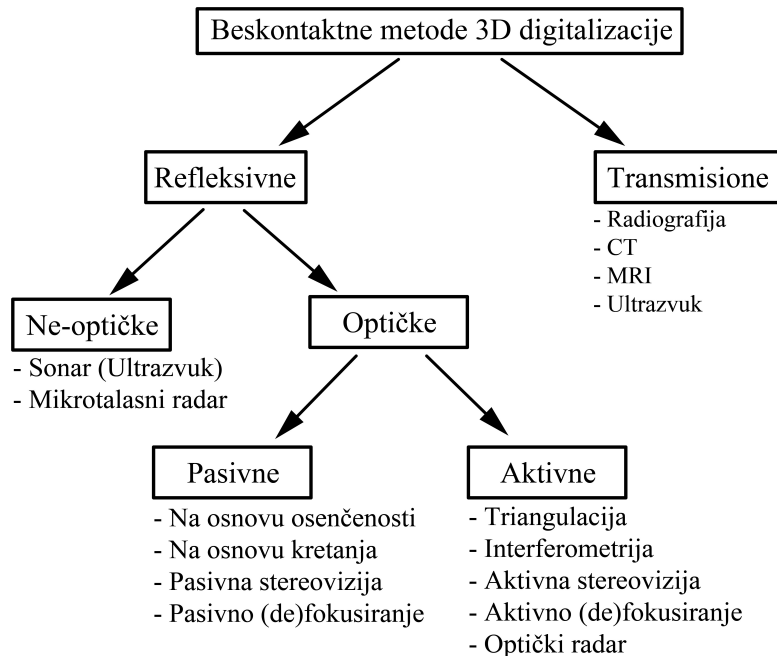
Beskontaktno metode se ogledaju u tome što im za generisanje informacije o geometriji nije potreban fizički kontakt sa objektom. Pored ove prednosti izdvaja se i njihova brzina digitalizacije, dobra tačnost i rezolucija za većinu aplikacija, sposobnost detektovanja boja, kao i mogućnost skeniranja malih promena na geometriji objekta (mesta za koje je vrh merne igle preveliki). Sa druge strane, mane ovih metoda mogu biti problem sa skeniranjem objekata koji poseduju površine određene boje, transparentnosti i reflektivnosti, kao i niža tačnost skeniranja u odnosu na kontaktne metode [2]. Za razliku od kontaktnih, postoji znatno veći broj beskontaktnih metoda koje se međusobno razlikuju i mogu se podeliti u više grupa. Podela beskontaktnih metoda 3D digitalizacije data je na slici 2.3. Na ovoj slici su takođe navedene i najpoznatije tehnike iz svake grupe pojedinačno. Zbog svoje zastupljenosti u mašinskoj industriji u ovom radu akcentat je na aktivnim optičkim metodama 3D skeniranja, ali biće navedene i osnovne osobine ostalih beskontaktnih metoda.

Prvi nivo podele beskontaktnih metoda odnosi se na način emitovanja određenog signala i njegovu upotrebu za generisanje informacije o geometriji posmatranog objekta. Transmisionne tehnike se baziraju na emitovanju energetske signala kroz objekat i obradi njegovog neapsorbovanog dela. Refleksivne metode skeniranja se ogledaju u projektovanju određene vrste signala na objekat i obrade reflektovanog dela signala u cilju dobijanja informacije o obliku [7]. Najvažnija razlika između metoda ove dve grupe je da transmisionne tehnike obezbeđuju informaciju o obliku unutrašnjosti objekta dok refleksivne mogu dati informacije samo o njegovoj spoljašnosti.

U transmisionne metode spadaju industrijska radiografija, kompjuterska tomografija (CT), magnetna rezonanca (MRI) i ultrazvuk. Sve ove tehnike su prvenstveno i prvobitno korišćene u medicinske svrhe, ali su kasnije našle svoju ulogu i u reverznom inženjerstvu [8]. Prve dve navedene tehnike koriste svojstva x-zraka pri čemu detektuju neapsorbovani nivo zračenja pri prolasku kroz posmatrani objekat odnosno njegovo slabljenje (atenuaciju).

Radiografija spada u 2D tehnike rekonstrukcije i može se podeliti na digitalnu i filmsku [7]. Filmska radiografija za detekciju nivoa oslabljenog zračenja (nakon prolaska kroz objekat) koristi posebne fotoosetljive filmove za jednokratnu upotrebu tj. za dobijanje samo jedne ekspozicije. Sa druge strane, tehnološki naprednija, digitalna radiografija koristi posebne radiografske panel detektore. Osnovu detektora sačinjavaju matrice fotoosetljivih dioda koje se nalaze ispod tzv. sintilatora. Sintilator, odnosno

hemijska jedinjenja od kojih je sačinjen, poseduje svojstvo pretvaranja primljene količine x-zračenja u svetlost [9]. Ovom promenom prirode signala obezbeđuje se da fotoosetljive diode zabeleže nivo svetlosti koji je srazmeran količini neapsorbovanog zračenja koju je detektor primio. Uprošćeno se može reći da nivo detektovane svetlosti (zračenja) u svakoj pojedinačnoj diodi ujedno označava i vrednost u svakom pojedinačnom pikselu neprocesirane digitalne radiografske slike. Treba naglasiti da se u industrijskoj praksi sreću i tehnike koje koriste detektore sačinjene od samo jednog reda (jedne linije) receptora (dioda) - tzv. linijski detektori [10].



Slika 2.3: Podela beskontaktnih metoda 3D skeniranja.

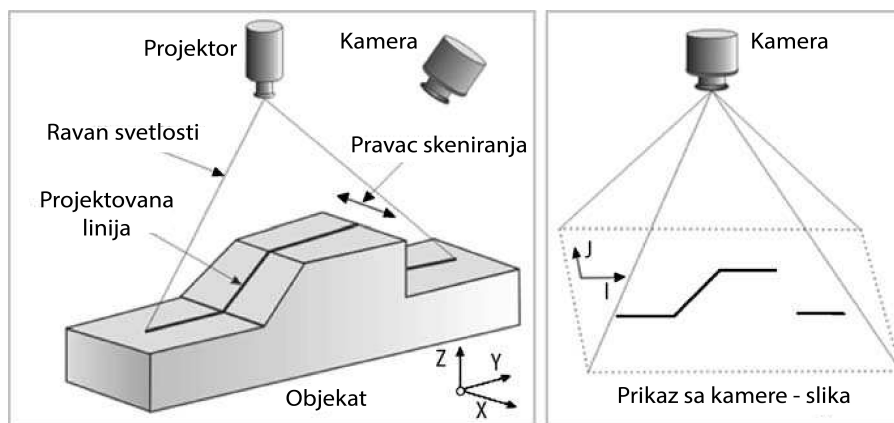
Kompjuterska tomografija za 3D rekonstrukciju objekta koristi metode digitalne obrade velikog broja 2D snimaka. Ovi snimci mogu biti generisani tokom kretanja (rotacije i translacije) rendgenskog sistema i/ili kretanja samog objekta. Kao i kod radiografije i ovde je moguće primeniti linijske detektore pa u tom slučaju možemo govoriti o linijskim skenerima (engl. *fan/line beam scanners*) naspram onih koji koriste 2D detektore (engl. *cone beam scanners*) [10]. CT predstavlja vrlo preciznu tehniku 3D digitalizacije. Moguće je snimanje velikog broja raznovrsnih materijala kao i objekata koji sadrže visoko reflektivne površine koje mogu predstavljati problem za reflektivne metode. Ipak najveća prednost tomografije ogleda se u tome što je ovo tehnika koja bez razaranja može rekonstruisati i unutrašnjost posmatranog objekta, dok je njena mana visoka cena same opreme [2].

Magnetna rezonanca za razliku od radiografskih tehnika kombinuje osobine magnetnih polja i radiofrekventne elektromagnetne talase [2]. Osovni princip rada se zasniva na osobinama protona atoma vodonika. Ovi protoni smešteni u jakom stacionarnom magnetnom polju (i do 8.5 T) pri izloženosti spoljnim radiofrekventnim elektromagnetnim talasima (jasno određene frekvencije) imaju sposobnost da svojim ponašanjem indukuju električnu struju u kalemovima (receptorima) smeštenim u okolini posmatranog objekta. Obradom (u više ravni) strujnih signala generisanih pri različitim frekvencijama dodatih elektromagnetnih talasa mogu se dobiti informacije o strukturi posmatranog objekta. Naravno, MRI je u širokoj primeni u medicinskoj dijagnostici, ali sve više nalazi svoju primenu i u reverznom inženjerstvu. Ovde se pre svega misli na RE tehnike u savremenoj medicini, a pre svega u ortopediji i stomatologiji [11]. Pored mnogobrojnih prednosti vezanih za skeniranje pre svega mekih tkiva, tetiva i kostiju, glavna mana ove tehnologije je veoma visoka cena opreme i softvera.

Refleksivne metode 3D skeniranja se dele na optičke i ne-optičke. Optičke reflektivne metode spadaju u najbrojniju grupu tehnika za 3D digitalizaciju. Njihova glavna odlika je brzina akvizicije. Mogu se podeliti na aktivne i pasivne. Glavna razlika je što pasivne optičke metode ne koriste pomoćni

izvor svetlosti kako bi je dodatno projektovale na posmatrani objekat. Zbog toga se izdvajaju relativno pristupačnom opremom, ali i spadaju u grupu najnepreciznijih metoda pa o njima neće biti više reči u ovom radu. Sa druge strane, aktivne optičke tehnike zbog svoje upotrebe dodatnih izvora svetlosti prevazilaze mnoge probleme pasivnih metoda i obezbeđuju značajno višu tačnost skeniranja.

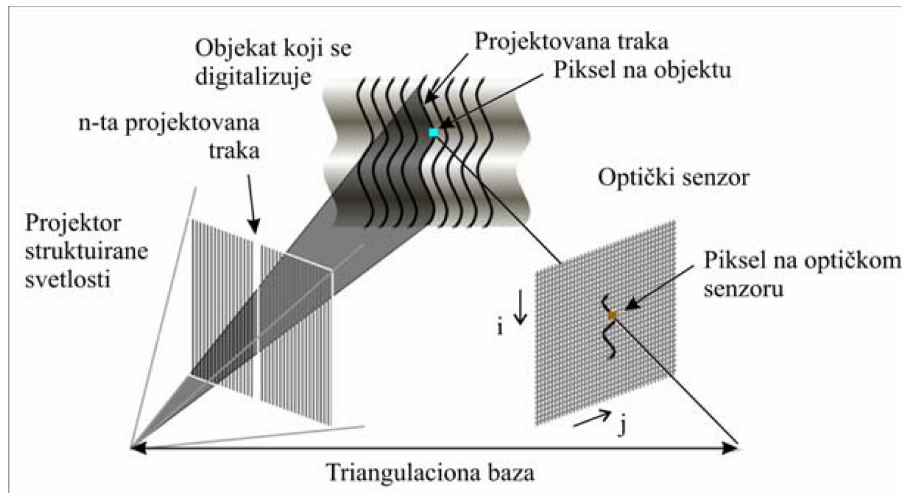
Optičke metode 3D digitalizacije zasnovane na geometrijskoj triangulaciji [12] koriste unapred poznate pozicije i uglove između izvora svetlosti i foto-osetljivog receptora (najčešće kamere) za izračunavanje pozicija određenih tačaka u prostoru [4]. Izvor svetlosti emituje fokusiranu svetlost na površinu objekta pod određenim uglom dok receptor prikuplja reflektovani signal čime su stvorene mogućnosti za izračunavanje distance od osnovne ravni do pozicija na površini objekta. Ove metode su dodatno klasifikovane na laserske triangulacione senzore (skenere) i one koje koriste strukturiranu svetlost [7]. Iako metode iz obe ove grupe u osnovi koriste triangulaciju, zbog načina njene primene kao i različitosti prirode svetlosnih izvora mnogi autori ih svrstavaju u posebne grupe optičkih metoda [2, 4]. Laserski triangulacioni skeneri [13, 14] mogu projektovati lasersku svetlost na objekat u obliku tačke ili linije. Sa druge strane, postoje konfiguracije laserskih senzora sa jednim i dva foto-osetljiva receptora pri čemu se, u zavisnosti od metoda, kao receptori koriste fotodiode ili matični sistemi fotodiode odnosno CCD senzori (kamere). Preciznost skeniranja ovakvih metoda uslovljena je pre svega rezolucijom receptora kao i udaljenošću objekta od samog uređaja. Laserski triangulacioni senzori se mogu naći na različitim vrstama mobilnih (pokretnih) platformi ili pak mogu biti stacionarni dok se objekat kreće. Takođe, mogu se kombinovati sa KMM-a ali i sa antropomorfnim industrijskim robotima [15] čime se dobijaju nove mogućnosti skeniranja. Prednosti laserskih skenera su veoma dobra digitalizacija udaljenih objekata kao i mala osetljivost na ambijentalno osvetljenje. Glavni nedostaci su potreba upotrebe zaštitnih mera kod lasera čije su talasne dužine u vidljivom ili ultraljubičastom delu spektra kao i manja brzina akvizicije u odnosu na npr. skenere koji koriste strukturiranu svetlost.



Slika 2.4: Princip linijskog skenera sa strukturiranom svetlošću [2].

Skeneri sa strukturiranom svetlošću [16, 17, 18, 19] koriste analizu izgleda (slike) projektovanog svetlosnog šablona na površine objekta kako bi generisali informaciju o geometriji. Projektovani šabloni mogu biti izgrađeni od tačaka, linija, pruga, pravougaone mreže, ali i kompleksno kodiranih formi. Sa stanovišta opreme ovakvi skeneri se uglavnom sastoje od projektor i kamera za posmatranje scene čiji broj zavisi od same tehnike. Tipični skener koristi jedan centralni projektor i jednu kameru za snimanje scene postavljenu pod poznatim uglom u odnosu na projektor (slika 2.4). Projektor na površinu objekta emituje svetlost u obliku ravni paralelne sa postavljenim koordinatnim sistemom skenera. Kamera detektuje presek svetlosti sa površinom posmatranog objekta u vidu pruga čija pozicija i dužina zavisi od oblika površina objekta. Proces skeniranja se odvija pomeranjem skenera ili samog objekta i snimanjem više uzastopnih slika. Korišćenjem geometrijske triangulacije i daljom obradom svih slika generiše se veliki broj tačaka na prugama čime se dobija precizna informacija o obliku površine posmatranog objekta. Kao varijanta ove tehnike postoji i metoda koja koristi dva projektor od kojih je jedan postavljen pod uglom [20]. Sa druge strane, da bi se ubrzao proces skeniranja

primenjuju se i tzv. tehnike više-linijske triangulacije [7]. Ovakvi skeneri projektuju odjednom više linija (pruga) na objekat pri čemu se oblik šablona menja u vremenu. Ovim postupkom svaka linija posmatrana kroz sve sekvence dobija jedinstveni iluminacioni kod. Ilustracija ove metode je prikazana na slici 2.5. Nešto kompleksniji skeneri ovog tipa koriste različite vrste kodiranih monohromatskih, ali i šablona u boji [18, 19, 21, 22].



Slika 2.5: Ilustracija principa više-linijskog skenera sa struktuiranom svetlošću [23].

U industrijskoj praksi često su u upotrebi skeneri koji projektuju kodirane šablone plave svetlosti. Svetlost ove talasne dužine je najmanje prisutna u okolini pa su samim tim skeneri ovakvog tipa otporniji na svetlosni šum. Čest je slučaj da ovakvi uređaji poseduju dve ili više kamera postavljenih pod uglom u odnosu na centralni projektor. Primer jednog industrijskog skenera ovakvog tipa [24] dat je na slici 2.6. Glavna odlika skenera ove grupe je veoma brza akvizicija - generisanje po nekoliko miliona tačaka u sekundi. Takođe, okarakterisani su visokom rezolucijom i tačnošću skeniranja. Sa druge strane mana im je potrebna priprema za skeniranje pre svega u vidu postavljanja određenih markera na objekat kao i eventualne pripreme površine radi smanjenja refleksije. Potrebno je napomenuti da je većina realnih oblaka tačaka korišćenih u ovoj disertaciji generisana upravo upotrebom dva skenera koji koriste tehnike ove grupe.

Nakon tehnika sa struktuiranom svetlošću značajno mesto zauzima interferometrija. Kada se sa stanovišta 3D digitalizacije govori o interferometriji, obično je reč o tehnikama zasnovanim na Moiré tehnici [25] koja je proistekla iz konvencionalne holografske interferometrije. Osnovni šablon paralelnih ekvidistantnih traka se projektuje na površinu objekta. Scena se zatim pod određenim uglom (u odnosu na pravac projekcije) fotografiše. Ovako dobijena fotografija daje prikaz tzv. Moiré šablona koji za posmatranu neravnu površinu objekta predstavljaju skup zakrivljenih traka (pruga). Procesiranjem ovako dobijene slike (šablona) i njenim upoređivanjem sa osnovnim šablonom generišu se informacije o geometriji površine posmatranog objekta. Interferometrija spada u brze tehnike 3D digitalizacije. Pojedini skeneri koji eksploatišu ovu tehniku mogu generisati i preko milion tačaka u sekundi. Skeneri ove grupe se uspešno primenjuju za digitalizaciju pre svega objekata i površina malih dimenzija jer se sa porastom veličine povećava potrebno udaljenje opreme od objekta čime se smanjuje preciznost same metode.

Stereovizijske tehnike su razvijene na osnovu ideje generisanja informacije o trećoj dimenziji (daljini, dubini) samo na osnovu dva (ili više) 2D prikaza istog objekta. Ove tehnike se zasnivaju na obradi i upoređivanju slika posmatranog objekta dobijenih snimanjem iz dve različite, ali poznate pozicije i orijentacije kamere. Ovakvi sistemi skeniranja mogu i u isto vreme posedovati fizički dve kamere u poznatom prostornom odnosu [26]. Glavna razlika između pasivne i aktivne stereovizije je u korišćenju dodatnih veštačkih izvora osvetljenja. Aktivne metode koriste prednosti dodatnog osvetljenja scene, ali neretko koriste i projektovanje određenih svetlosnih šablona što može doneti prednosti pri

obradi podataka. Sa stanovišta savremene 3D digitalizacije bitna tehnika streovizije je tzv. fotogrametrija. Ova tehnika koristi obradu dve ili obično više fotografija objekta koje su generisane snimanjem jednom kamerom iz različitih pozicija i pod različitim uglovima. Slično nekim prethodno navedenim sistemima 3D digitalizacije i ovde se obrada fotografija zasniva na određenim geometrijskim odnosima. Uprošćeno govoreći, kod sistema sa dve fotografije jedna uočena tačka na objektu odgovara različitim pozicijama na fotografijama pa se njene koordinate u pridruženom opštem koordinatnom sistemu mogu izračunati iz geometrijskih odnosa [7]. U praksi se međutim često koristi znatno veći broj fotografija jednog objekta za njegovu 3D digitalizaciju. Za razliku od klasične fotogrametrije, tehnike zasnovane na SfM principu ne zahtevaju prethodnu kalibraciju kamere, već se njeni parametri izračunavaju u procesu obrade slika detektujući značajna obeležja na objektu - autokalibracija [7]. Za primenu i tačnost fotogrametrijskih algoritama pored veličine samog objekta važno je posedovanje što većeg broja njegovih fotografija (iz raznih uglova) visoke rezolucije sa znatnim preklapanjem prikaza. Mana ovakvih metoda se ogleda u lošijoj 3D digitalizaciji objekata koji poseduju jednolične visoko-refleksivne površine. Ova ograničenja se pak u nekim slučajevima mogu prevazići upotrebom projektovanja određenih tekstura na objekat, ali čak i njihovim fizičkim nanošenjem (nalepnice ili određeni premazi).



Slika 2.6: Primer optičkog industrijskog skenera - Atos Compact Scan 5M [24].

Tehnike 3D digitalizacije koje određuju daljinu fokusiranjem/defokusiranjem koriste optičke zakonitosti kako bi generisale informacije o udaljenju (dubini) određenih tačaka na posmatranom objektu [27]. Njihova podela na pasivne i aktivne ogleda se u prisutnosti veštačkog izvora osvetljenja. Osnovni princip fokusiranja temelji se na korišćenju finih konveksnih sočiva (bez prisustva aberacije). Ukoliko se posmatrana tačka nađe u fokusu sočiva onda se njeno udaljenje od istog može sračunati prostim geometrijskim odnosom. Koristeći ovu osobinu kreira se niz fotografija (i preko deset) sa različitim udaljenjima sočiva od senzora slike. Ove fotografije se zatim obrađuju čime se u svakom pikselu (koji predstavlja tačku na površini objekta) određuje fotografija na kojoj je imao najveću oštrinu. Samom tom informacijom zajedno sa pomenutim zakonitostima sočiva lako je izračunati udaljenosti tačaka na površini posmatranog objekta. Sa druge strane, tehnike defokusiranja unapređuju postupak i smanjuju broj potrebnih fotografija. Kod njih se posmatra tačka na objektu kada se ne nalazi u fokusu. U tom položaju tačka se na slici reprezentuje zamućenom (engl. *blur*) mrljom određenog prečnika. Oslanjajući se na prečnik ove mrlje, matematički opis funkcije promene udaljenja sočiva od senzora slike kao i parametara samog sočiva, distanca posmatrane tačke do sočiva se lako može izračunati. Ovaj postupak se izvodi samo na osnovu dve fotografije dobijene sa različitim udaljenjima sočiva, ali postoje i metode koje koriste samo jednu fotografiju. Tehnike fokusiranja kao i defokusiranja se ne ogledaju visokom tačnošću 3D digitalizacije pa se ne koriste u oblastima RE gde je visoka tačnost neophodna. Njihova prednost se ogleda u relativno jednostavnoj i dostupnoj opremi čija kalibracija ne predstavlja poteškoću u radu.

Kao posebne optičke tehnike digitlizacije izdvajaju se različite vrste radara. Optički radari su najčešće zasnovani na upotrebi laserskog izvora svetlosti [2]. Laserski radari su razvijeni na dva osnovna principa. Prvi princip je zasnovan na merenju vremena za koje impuls emitovane laserske svetlosti pređe put od emitera do tačke na površini objekta i nazad - tzv. vreme preleta. Generisanje informacije o geometriji posmatranog objekta se svodi na proračunavanje udaljenosti (distance) od emitera do tačaka na površini objekta, a na osnovu dobijenog vremena preleta i poznate konstantne brzine prostiranja laserskog impulsa (svetlosti) kroz vazduh. Drugi princip je zasnovan na amplitudnoj modulaciji signala. Naime, ka tački objekta se generiše konstantni laserski snop čija je jačina sinusno modulirana tokom vremena. U ovom slučaju reflektovani signal je fazno pomeren od emitovanog i udaljenost ciljane tačke od emitera je u direktnoj zavisnosti od ove fazne razlike [7]. Glavna prednost laserskih radara ogleda se u preciznom skeniranju objekata različite veličine (pogotovo velikih dimenzija) na velikim udaljenostima.

Od ne-optičkih metoda izdvajaju se sonar i mikrotalasni radar. Obe ove tehnike koriste vreme preleta (kao kod prve grupe optičkih radara) emitovanog impulsa zraka za izračunavanje udaljenja tačaka na objektu. U slučaju sonara reč je o ultrazvučnim impulsima, dok je u slučaju mikrotalasnog radara reč naravno o impulsima mikrotalasne energije. U tehnikama RE sonar se najčešće koristi za podvodna 3D mapiranja terena dok je upotreba radara šira zbog njegove osobine detekcije i skeniranja objekata na velikoj udaljenosti od samog uređaja. Prednost uređaja iz ove grupe ogleda se u pristupačnoj ceni, ali je mana nešto lošija rezolucija skeniranja. Takođe, sonar se odlikuje i sporijim procesom akvizicije, ali i osetljivošću na akustičnu interferenciju i spoljašnji šum [2].

2.3. Oblak tačaka

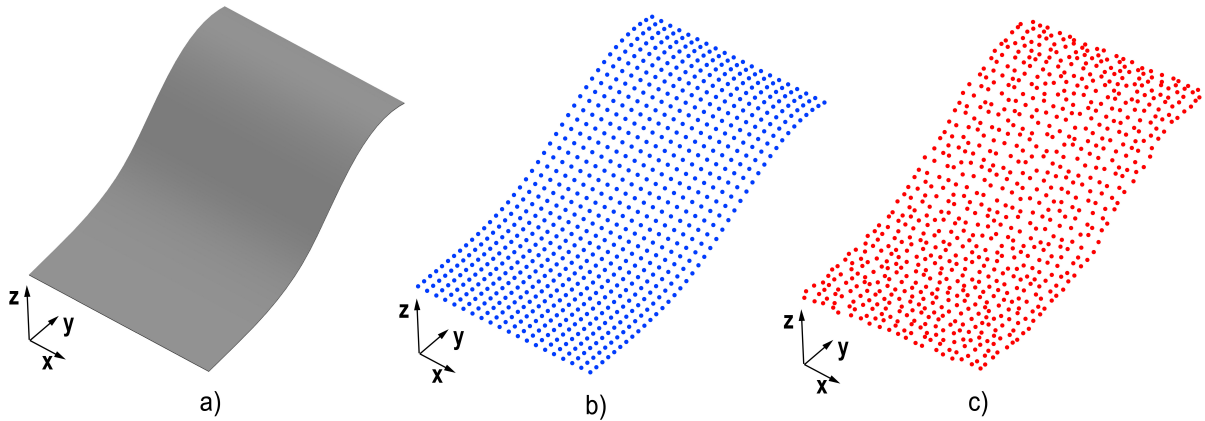
Kao što je već navedeno, uobičajeni rezultat procesa 3D digitalizacije (skeniranja) predstavlja forma koju nazivamo 3D oblak tačaka ili skraćeno oblak tačaka. Prema međusobnom odnosu i uređenosti redosleda generisanih tačaka, oblaci tačaka se mogu podeliti na:

- struktuirane (organizovane), i
- nestruktuirane.

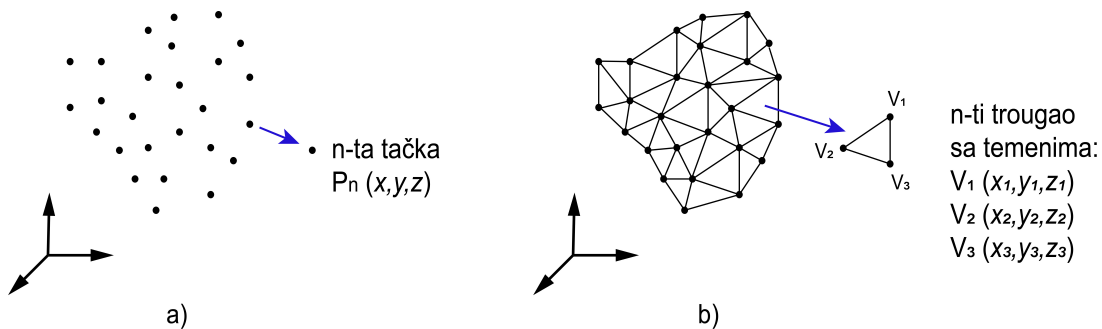
Pod struktuiranim oblacima podrazumevamo one oblake kod kojih su međusobni geometrijski odnosi (pre svega distance) između tačaka unapred poznati i uređeni prema nekom pravilu u pridruženom koordinatnom sistemu. Često se mogu susresti struktuirani oblaci kod kojih su tačke prostorno uređene u ekvidistantnom rasteru. Takođe, struktuirani oblaci se mogu i sastojati iz nizova paralelnih skeniranih linija odnosno grupa tačaka generisanih duž jednog definisanog pravca. Sama struktuiranost oblaka umnogome zavisi od tehnike skeniranja kao i algoritama generisanja tačaka. Sa druge strane, nestruktuirani oblaci, kako im samo ime govori, ne poseduju jasnu strukturu tačaka. Zbog ove osobine oni su zahtevniji za dalje procesiranje. Vizuelne razlike između struktuiranog i nestruktuiranog oblaka su prikazane na primeru jedne površine na slici 2.7.

Osnovna forma opšteg oblaka tačaka sadrži samo 3D koordinate (x, y, z) tačaka na površinama objekta zapisane u matricnom obliku sa tri kolone odnosno formata $n \times 3$, gde je n broj tačaka oblaka. 3D koordinate su obično izražene u pridruženom Dekartovom koordinatnom sistemu. Za mnoge aplikacije ovakva forma oblaka predstavlja dovoljni izvor podataka za dalje procesiranje, ali sa napretkom uređaja i tehnika skeniranja sve češće je prisutno proširivanje formi zapisa. Obično se koordinatama pridružuju informacije (tri dodatne kolone brojeva) o boji svake pojedinačne tačke iskazanoj u RGB modelu označavanja. Takođe, pored boje, tačkama mogu biti pridružene i koordinate vektora normale u svakoj tački (još tri dodatne kolone u zapisu) koje su izračunate nekim od metoda diferencijalne geometrije na osnovu lokalnih osobina okruženja svake pojedinačne tačke oblaka. Zavisno od formata zapisa može postojati i zaglavlje sa informacijama o samom fajlu, strukturi navedenog zapisa oblaka, uređaju ili softveru koji je generisao fajl kao i drugi podaci. Zbog načina razvoja oblasti 3D skeniranja, ali i mnogih velikih kompanija koje se bave ovom oblašću u upotrebi je veliki broj standardizovanih

formata digitalnog zapisa oblaka tačaka. Glavna razlika među njima je u upotrebi ASCII ili binarnog koda, ali postoje i fajl formati koji kombinuju oba. Neki poznati formati zapisa oblaka tačaka su OBJ, PLY, PTX, PCD, FLS, LAS itd. Treba napomenuti da su određeni formati predviđeni samo za strukturane oblake tačaka dok drugi podržavaju i nestruktuirane. Takođe, neki formati kao izlaz iz procesa 3D digitalizacije sadrže poligonalne (najčešće trougaone) mreže koje predstavljaju već procesirane koordinate tačaka iz početnog (sirovog) oblaka. Razlika u reprezentaciji oblaka tačaka i trougaone mreže prikazana je na slici 2.8.



Slika 2.7: Vrste oblaka tačaka prema strukturi: a) izgled površine objekta; b) strukturani oblak tačaka; c) nestruktuirani oblak tačaka.



Slika 2.8: Razlika između 3D oblaka tačaka i mreže: a) oblak tačaka; b) poligonalna (trougaona) mreža nastala procesiranjem oblaka.

Potrebno je pomenuti da je za dobijanje oblaka tačaka koji u celini reprezentuje određeni objekat u praksi često potrebno izvršiti više skeniranja iz više pravaca čime se dobija i više nezavisnih oblaka tačaka. Da bi se svi nezavisni oblaci precizno grupisali u jedan kompletan oblak potrebno je primeniti tehnike njihovog spajanja. Ove procese nazivamo registracijom i integracijom oblaka. Takođe, mnogi skeneri na nižem nivou generišu tačke u strukturanom rasporedu, ali zbog naknadnih različitih postupaka registracije i integracije konačno formirani oblaci ipak poseduju haotičnu strukturu, odnosno postaju nestruktuirani oblaci.

Kada govorimo o oblacima tačaka potrebno je osvrnuti se i na njihovu sposobnost vernog reprezentovanja skeniranog objekta. Iako su savremeni 3D skeneri sposobni za veoma preciznu akviziciju ogromnog broja tačaka u malom vremenskom intervalu, ovo ne znači da generisani oblaci tačaka ne nose sa sobom određene nesavršenosti (artefakte) koje u manjoj ili većoj meri mogu uticati na kvalitet daljeg procesiranja i njihove upotrebe za određene namene. Najčešći problem oblaka tačaka svakako predstavljaju nasumično generisane greške (tačke) prilikom akvizicije (skeniranja). One mogu biti dodate oblaku u procesu skeniranja iz raznih razloga, ali najuticajniji su tip i performanse skenera, struktura objekta (oblik, materijal, boja, kvalitet površine itd.) kao i uslovi skeniranja (osvetljenost,

refleksija sa površine objekta, direktan vizuelni pristup skenera objektu, udaljenost skenera od objekta itd.). Greške koje se javе prilikom akvizicije manifestuju se kao tačke koje ne pripadaju površini posmatranog objekta (engl. *outliers*). Njihovim uzimanjem u razmatranje pri procesiranju može se stvoriti loša slika o geometriji objekta i generisati dalja greška. Takođe, jedan deo pogrešno generisanih tačaka predstavlja i šum. Iz razloga postojanja ovih nesavršenosti akvizicionog procesa 3D skeneri se konstantno unapređuju, ali se uporedo radi i na razvoju raznih tehnika koje odstranjuju šum i nepoželjne tačke iz oblaka. Još jedan od potencijalnih problema može biti i nejednaka gustina (rezolucija) oblaka koja se javlja usled nepovoljne orijentacije površina objekta u odnosu na pravac skeniranja, udaljenosti skenera od objekta, ali i samih karakteristika površina objekata. Zbog navedenog, mnoge tehnike 3D digitalizacije zahtevaju skeniranje iz više pravaca u uslovima gde je to moguće. Na primer, u situacijama gde su određeni delovi površina skeniranih objekata opisani segmentima oblaka niže rezolucije, moguće je da su određeni sitni detalji na površini objekta reprezentovani sa svega nekoliko tačaka čime se onemogućuje njihovo dalje prepoznavanje i gubi se informacija o njihovoj geometriji. Većina ostalih mogućih nesavršenosti oblaka tačaka kao i razlozi njihovog nastanka jasno su opisani u [28].

2.4. Pregled primena metoda za 3D digitalizaciju objekata u mašinskoj industriji

Savremena industrija poslednjih godina spoznaje benefite 3D digitalizacije i tehnika RE i kao takve ih integriše u svoje tradicionalne procese stvarajući time nove koncepte. Pomenuta integracija je u velikoj meri ostvarena prvenstveno u procesima kreiranja i razvoja proizvoda. Veliki broj savremenih CAD programskih paketa sadrži module koji podržavaju reverzno inženjerstvo geometrijskih modela čijim se korišćenjem oblak tačaka može prevesti u upotrebljiv kompjuterski model objekta. Ovakvi procesi, iako danas opšteprisutni u svakodnevnom inženjerskom radu, zbog svoje potrebe za obimnom korisnikovom asistencijom i nedostatkom kvalitetnih algoritama procesiranja, ipak nisu na očekivanom nivou za upotrebe u aplikacijama gde je potrebno reagovanje u realnom vremenu. Međutim, moderni senzori (pre svega laserski trinagulacioni) i 3D skeneri poseduju visoku tačnost skeniranja i pružaju i više nego dovoljno informacija o geometriji posmatranih objekata. Kao takvi oni se uz razvoj odgovarajućih metoda i algoritama mogu koristiti za veliki broj aplikacija u savremenoj automatizovanoj industriji što je i tema ovog poglavlja.

Tehnike 3D digitalizacije su već dugo prisutne u primeni metrologije u oblasti kontrole kvaliteta proizvoda. Pored konvencionalnih KMM u ovoj oblasti su zastupljene optičke tehnike 3D digitalizacije (laserske triangulacione, interferometrija, fotogrametrija), ali i transmisione radiografske. Poseban izazov predstavlja inspekcija delova složene geometrije. U [5] je izvršena opširna uporedna analiza najzastupljenijih metoda za ovu namenu. Ovaj rad definiše prednosti i mane najrelevantnijih metoda geometrijske inspekcije posmatrajući ceo spektar tačnosti, veličina, oblika, kvaliteta površina i materijala objekata. Sa druge strane, istraživanja u oblasti 3D digitalizacije direktno mogu poboljšati i upotrebu KMM. Ovde se pre svega misli na poboljšanje procesa planiranja strategije dimenzione kontrole mašinskih delova na KMM korišćenjem pogodnosti laserskih trinagulacionih skenera [29, 30]. Kod ovakvih hibridnih metoda planiranje strategije se zasniva na kombinovanju informacija dobijenih skeniranjem objekta i samog njegovog CAD modela. 3D skeneri se mogu kombinovati i sa NUMA u cilju poboljšanja postupaka lokalizacije priprema, planiranja putanja alata pri obradi, ali i inspekciji mera obratka na samoj mašini. U [31] autori su detaljno opisali postupak integracije laserskog 3D skenera i mašinskog obradnog centra pri čemu se oblak tačaka dobijen skeniranjem priprema direktno registruje u koordinatnom sistemu mašine čime se obezbeđuju navedene mogućnosti planiranja i kontrole obrade.

Laserski triangulacioni senzori imaju primenu i u mnogim automatizovanim procesima kontrole kvaliteta dimenzija izrađenog proizvoda. Jedan primer realne industrijske upotrebe opisan je u [32, 33]. U ovim radovima su predstavljene dve metode kontinualne kontrole debljine gumiranog korda tokom

njegove proizvodnje postupkom kalandriranja. Takođe, često se u industriji susreću i nedestruktivna ispitivanja nadvišenja šava kod zavarenih konstrukcija upotrebom laserskih senzora. Autori u [34] su opisali jedan ovakav sistem. Kod njega se laserski zrak projektuje na oblast zavarivanja u vidu linije dok senzor (pozicioniran pod odgovarajućim uglom) detektuje njegov reflektovani oblik koji odgovara geometriji samog šava. Relativnim kretanjem zavarenog dela i uređaja za 3D digitalizaciju ostvaruje se skeniranje potrebne oblasti zavarivanja. Ovim postupkom je moguće lako uočavanje nepravilnosti u obliku i poziciji šava na zavarenoj konstrukciji.

Zavarivanje svakako predstavlja još jednu oblast mašinske industrije u kojoj su tehnike 3D skeniranja pomogle unapređenju automatizacije procesa. Kao što je poznato automatsko (robotizovano) zavarivanje zahteva precizno pozicioniranje i visok kvalitet pripreme elemenata koji se zavaruju čime se povećavaju troškovi samog procesa. Upotreba tehnika 3D digitalizacije u ovoj oblasti je upravo orijentisana na mogućnost automatskog zavarivanja lošije pripremljenih elemenata kao što je to slučaj kod manuelnog zavarivanja. U [35] je predstavljena metoda automatske korekcije putanje gorionika na osnovu vizelne inspekcije međusobne pozicije, stanja i oblika ivica elemenata koji se zavaruju čime je omogućena lošija priprema delova i smanjenje troškova procesa zavarivanja. Vizelna inspekcija se u ovom slučaju izvodi 3D digitalizacijom korišćenjem linijskog laserskog izvora i CCD kamere. Još jedna referentna metoda za korekciju off-line programiranih putanja automata za zavarivanje na osnovu laserskog senzora i CCD kamere predstavljena je u [36].

Aktuelna istraživanja na temu upotrebe 3D digitalizacije u procesima zavarivanja se pre svega odnose na procese inspekcije i održavanja upotrebom savremenih robotizovanih sistema u oblasti procesne industrije. U [37] je prezentovan sistem za detekciju linija šavova pri inspekciji zavarenih konstrukcija velikih tankova specijalizovanim daljinski upravljanim robotom (engl. *wall-climbing robot*). Sistem je baziran na korišćenju laserskog projektora i CCD kamere. Digitalizacija pomaže i pri održavanju pravca robota pri čišćenju korozije i peskarenja šava na tankovima. Jedan ovakav sistem baziran na dve CCD kamere i dodatnom LED osvetljenju predstavljen je u [38]. Sa druge strane tehnike skeniranja se mogu uočiti i kod rešavanja problema lokalizacije i navigacije ovakvih specijalnih robota pri unutrašnjoj inspekciji sfernih tankova velike zapremine kao što je slučaj u [39].

Kada se govori o praktičnim primerima 3D skeniranja ne može se izbeći automobilska industrija. Široko je poznata upotreba 3D skeniranja za potrebe inspekcije delova šasije i kompleksnih proizvodnih alata [40, 41], ali 3D digitalizacija može pomoći i tamo gde naizgled nije upotrebljiva. Interesantan je npr. slučaj kontrole oblika zakrivljenih staklenih površina tipa vetrobranskog stakla. Ovakav jedan princip korišćenja laserskog skenera sa struktuiranom svetlošću i dve kamere je detaljno opisan u [42]. Transmisione tehnike digitalizacije takođe imaju značajan udeo u automobilskoj industriji [43]. One se koriste za nedestruktivnu inspekciju delova. Ova oblast je sa pojavom raznih naprednih tehnika obrade slike i mašinskog učenja i danas ostala aktuelna u istraživanju [44].

Još jedan od reprezentativnih primera upotrebe digitalizacije je svakako transport delova i sirovina u proizvodnim pogonima koji se odvija nekim od vidova automatski vođenih mobilnih robotizovanih sistema/kolica (AGV) [45]. Trenutni standard za navođenje i lokalizaciju AGV predstavljaju različite vrste magnetnih ili induktivnih sistema koji zahtevaju znatan nivo ulaganja u infrastrukturu i održavanje, a ne obezbeđuju i zavidan nivo fleksibilnosti. Takođe, nisu pogodni za nediskretizovana i dinamična proizvodna okruženja. Iz ovih razloga proteklih godina prisutan je porast istraživanja u oblasti upotrebe metoda 2D/3D digitalizacije za lokalizaciju, navođenje i detektovanje prepreka pri kretanju AGV [46], ali i za potrebe razvoja inteligentnih algoritama i softvera koji obezbeđuju visok nivo autonomije ovakvih mobilnih sistema. Mnogi praktični problemi prevaziđeni su integracijom AGV i laserskih sistema navođenja i detektovanja prepreka [47, 48], ali su neki od njih morali biti rešeni i upotrebom dodatnih senzora i tehnika akvizicije [49, 50].

Pored oblasti fleksibilnog transporta delova u savremenim proizvodnim procesima izdvaja se još jedna značajna oblast upotrebe tehnika 3D digitalizacije. Reč je prepoznavanju položaja (pozicije i orijentacije) nasumično postavljenih objekata u definisanoj sceni u cilju obezbeđivanja informacija za dalje autonomne manipulacije unutar fleksibilnih proizvodnih ćelija (engl. *bin picking*) [51]. Osnov

ovih metoda prepoznavanja su algoritmi za automatsko procesiranje podataka dobijenih 3D digitalizacijom scene. Pored tipa korišćenog algoritma, metodi prepoznavanja položaja se izdvajaju prema načinu digitalizacije scene. Tako se scena može digitalizovati korišćenjem neke od postojećih tehnika 3D skeniranja [52, 53, 54], ali i korišćenjem jedne kamere i procesa obrade slika [51, 55]. Mnogi metodi ove grupe porede podatke dobijene 3D digitalizacijom sa CAD modelima očekivanih objekata u sceni. Na ovaj način vrši se pojednostavljenje procesa prepoznavanja i povećava njegova tačnost. Sa druge strane, postoji i mali broj metoda koji ne koriste ovu metodologiju poređenja [56].

Pored navedenih oblasti primene, postoje druge oblasti mašinske industrije kod kojih bi se mogla izvršiti unapređenja procesa primenom tehnika 3D digitalizacije. Mnoge od njih ipak zahtevaju kvalitetnije prepoznavnje i matematičko opisivanje geometrijske strukture pomatranih objekata kao i njihovo tačno prevođenje u odgovarajuće kompjuterske modele. Zbog toga razvoj automatskih algoritama i metoda procesiranja oblaka tačaka u cilju ekstrakcije gradivnih geometrijskih primitiva skeniranih objekata kao i prevođenja oblaka u potrebni CAD model predstavljaju aktuelnu temu istraživanja širom sveta.

3. Pregled postojećih metoda za procesiranje 3D oblaka tačaka

Kao što je objašnjeno u poglavlju 2, na putu od 3D digitalizacije do kreiranja kompjuterskog modela realnog objekta susrećemo se sa različitim procesima obrade oblaka tačaka. Kao relevantne tehnike ove oblasti izdvajaju se tehnike za redukciju broja tačaka oblaka kao i one za segmentaciju i prepoznavanje gradivnih primitiva skeniranog objekta. U ovom poglavlju biće predstavljena aktuelna istraživanja vezana za pomenute dve vrste tehnika procesiranja oblaka tačaka.

3.1. Metode za redukciju broja tačaka u okviru oblaka tačaka

Tehnike 3D skeniranja (pogotovo optičke) odlikuju se visokom rezolucijom i tačnošću. Visoka rezolucija po pravilu znači i generisanje oblaka tačaka velike gustine pri čemu su površine skeniranih objekata reprezentovane velikim brojem tačaka. Ovakva reprezentacija površina svakako nosi i više nego dovoljno informacija o njihovom obliku, ali u velikom broju slučajeva takva gustina i broj tačaka nisu neophodne za detektovanje i opisivanje same geometrije. Kao prost primer može se navesti skeniranje jednostavnijih objekata izgrađenih od površina oblika ravni. Sama gustina oblaka i veliki broj tačaka nisu od presudnog značaja za prenos informacije o skeniranim ravnima pošto se ravni mogu precizno opisati i sa svega nekoliko tačaka. Glavni problem kod oblaka sa previše tačaka ogleda se u tome da on zahteva znatno veće računarske i memorijske resurse za dalje procesiranje što umnogome može usporiti i otežati nastavak procesa rada. Takođe, potrebno je razmišljati i o savremenim mogućnostima čuvanja ili transfera podataka putem određenih *cloud* ili *web* servisa za koje bi redukovani (smanjeni) oblaci bili znatno pogodniji. Zbog nevedenih razloga, koji predstavljaju otežavajuće okolnosti pri manipulaciji oblakom, javila se potreba za redukcijom broja tačaka oblaka. Glavni cilj redukcije predstavlja smanjenje broja tačaka uz zadržavanje visokog nivoa inicijalne informacije o skeniranim površinama. Tačnije, potrebno je selektivno odstraniti tačke koje nemaju (ili imaju vrlo mali) uticaj na nošenje informacije o geometriji skeniranog objekta što nije uvek lak zadatak. Uopšteno govoreći, potrebno je za datu površinu *Surf*, rekonstruisanu na osnovu oblaka tačaka P , i željene veličine redukovanog oblaka $n < |P|$, generisati oblak P' ($|P'| = n$) takav da je distanca ϵ između njegove rekonstruisane površine $Surf'$ i date *Surf* minimalna [57]. Takođe, relevantan je i pristup u kome se zadaje željena minimalna distanca između površina, a traži maksimalna moguća redukcija.

Postojeće metode redukcije broja tačaka oblaka (uprošćavanja oblaka) mogu se podeliti u dve osnovne grupe na:

- metode koje redukuju poligonalne 3D mreže (engl. *mesh-based simplification methods*), i
- metode koje sprovode direktno redukovanje tačaka oblaka (engl. *direct point cloud simplification methods*).

Prva grupa metoda se bavi procesiranjem prethodno formiranih poligonalnih (najčešće trougaonih) mreža i njihovim uprošćavanjem baziranim na različitim kriterijumima. Pregled tehnika ove grupe prikazan je u [58, 59]. Ove metode moraju sadržati prethodno implementirane algoritme za generisanje (engl. *meshing*) i preoblikovanje (engl. *re-meshing*) 3D mreža na osnovu oblaka tačaka i kao takve su računarski i vremenski veoma zahtevne [57]. Zbog navedene osobine ova grupa metoda se neće razmatrati u ovom radu.

Metode koje direktno redukuju oblake tačaka bez prethodnog generisanja poligonalnih mreža mogu imati značajniju upotrebu u praksi zbog svoje brzine izvršavanja kao i zbog nekorišćenja tehnika kreiranja 3D mreža. Najintuitivnija i najjednostavnija metoda ove grupe je svakako uniformno smanjenje (engl. *uniform downsampling*) broja tačaka oblaka na osnovu zadatog faktora redukovanja (engl. *sampling rate*). Postoje i metode koji vrše uniformnu redukciju zadržavajući minimalnu unapred zadatu distancu između sačuvanih tačaka što se može uporediti sa pojmom smanjenja rezolucije oblaka. Međutim, imajući u vidu zahtev o održanju visokog nivoa informacije o geometriji možemo zaključiti da za objekte komplikovanije geometrije ove metode ne proizvode adekvatne rezultate. Takođe, za

njihovu upotrebu potrebni su oblaci sa uređenim redosledom tačaka. Zbog svega navedenog pribegava se razvoju metoda koji uvode razne kriterijume odabira tačaka i ocenjuju njihov uticaj na konačne rezultate procesa rekonstrukcije objekta. Prema strukturi oblaka koji obrađuju, metode iz ove grupe mogu se razvrstati na one koji procesiraju nestruktuirane i one koje procesiraju struktuirane oblake tačaka.

Direktno procesiranje nestruktuiranih oblaka u cilju njihovog redukovanja se po pravilu izvodi metodama grupisanja (klasterovanja) tačaka na osnovu određenih kriterijuma i izdvajanja reprezenata. Autori u [60] su predstavili postupak redukcije oblaka dobijenih laserskim skeniranjem koji se zasniva na dekompoziciji prostora neuniformnim 3D koordinatnim mrežama (rešetkama). Ova dekompozicija je zasnovana na principu oktalnog stabla (engl. *octree*) korišćenjem kriterijuma standardne devijacije vektora normala u svakoj tački. Grafički pojednostavljeno, ovakava dekompozicija prostora se može zamisliti kao opisivanje paralelopipeda (obično kvadra ili kocke) oko oblaka tačaka i njegovo deljenje na inicijalno osam segmenata (oktanta) prostora. Dalja podela oktanta se vrši prema kriterijumu standardne devijacije normala tačaka koje mu pripadaju i koja se računa za svaki oktant posebno. Ukoliko je devijacija veća od unapred definsiane, posmatrani oktant se dalje deli na sledećih osam i postupak se ponavlja sve do postizanja krajnje neuniformne diskretizacije oblaka. Na kraju se redukovani oblak generiše tako što se iz svake ćelije zadrži po jedna reprezentativna tačka čija normala najbliže odgovara srednjoj vrednosti normala tačaka iz te ćelije. Treba naglasiti da je postupak izračunavanja vrednosti normala u tačkama baziran na 2D triangulaciji prema predloženoj inovativnoj strategiji.

U [57] autori implementiraju i upoređuju tri metode direktne redukcije oblaka od kojih je najbrža i računarski najefikasnija ona bazirana na grupisanju tačaka oblaka, odnosno na izdvajanju klastera. Autori u ovom radu razgraničavaju dve tehnike klasterovanja. U prvom slučaju klasteri se formiraju iterativnim rastom regiona od izabranih polaznih tačaka (engl. *seed points*) na osnovu zadate maksimalne varijacije σ_{max} klastera duž normale, odnosno od tangentne ravni centroida. Ova vrednost se u svakoj iteraciji upoređuje sa varijacijom σ_n koja se izračunava pomoću prve tri sopstvene vrednosti sistema n do tada izdvojenih tačaka. U drugom slučaju vrši se hijerarhijsko deljenje oblaka tačaka. Ovim postupkom se obrazuje binarno stablo (engl. *binary tree*) gde svaki njegov ogranak obuhvata jedan klaster tačaka. Kriterijumi podele oblaka tačaka su zadata maksimalna veličina klastera i pomenuta maksimalna varijacija σ_{max} pa se ćelije prostorne rešetke oblaka dele sve dok ova dva uslova ne budu zadovoljena. Nakon procesa formiranja svih klastera, u oba slučaja, redukovani oblak će sadržati samo njihove centroide.

Sa druge strane, jedan rekurzivni postupak redukcije predstavljen je u [61]. U ovom radu se takođe vrši deljenje oblaka na grupe tačaka, ali ovoga puta korišćenjem klasterovanja metodom k srednjih vrednosti (engl. *k-means clustering*) [62] koje se široko koristi u prepoznavanju oblika i mašinskom učenju. Kriterijum podele na klaster je devijacija ugla normala u tačkama oblaka/klastera, a na osnovu predefinisane praga od strane korisnika. Postupak grupisanja počinje izborom određenog broja inicijalnih tačaka (uniformno raspoređenih u oblaku) koje postaju budući centroidi i u čijoj se okolini, na osnovu pomenutog kriterijuma, formiraju klasteri koji se dalje mogu deliti rekurzivno. Na kraju postupka redukovani oblak čine centroidi finalno izdvojenih klastera. Ovako postavljen algoritam je sklon gubljenju ivica objekta odnosno formiranju klastera koji pripadaju dvema površinama. Da bi se ovo predupredilo, autori su razvili i postupak za detekciju ivica objekta koji se primenjuje pre samog postupka klasterovanja. Takođe su razvili i dopunski algoritam za ujednačavanje gustine i rasporeda tačaka redukovane oblaka kako bi se izbeglo njihovo nepotrebno nagomilavanje u oblastima znatne promene geometrije kao i blizu ivica. Ovaj algoritam se izvršava nakon postupka redukovanja.

Još jedna metoda koja se zasniva na grupisanju tačaka u klaster i izdvajanju centroida kao njihovih reprezenata u redukovanom oblaku predstavljena je u [63]. Ovoga puta za klasterovanje se koristi adaptivni *mean-shift clustering* algoritam, dok su za kriterijume grupisanja iskorišćeni vektori normala i lokalne pozicije tačaka (njihova relativna udaljenost od zadatog broja okolnih tačaka). Treba naglasiti i da vektor normale koji se pridružuje svakom centroidu u redukovanom oblaku predstavlja sopstveni vektor koji odgovara najmanjoj sopstvenoj vrednosti sistema tačaka klastera. Rad [64], sa

druge strane, opisuje potrebu kreiranja redukovanog oblaka nejednake gustine sa srazmerno većom količinom zadržanih (ili generisanih) tačaka na mestima značajne lokalne promene geometrije. Podela polaznog oblaka na klasterne i izdvajanje referentnih tačaka se, između ostalog, izvodi korišćenjem prilagođenog algoritma fazi c srednjih vrednosti (engl. *fuzzy c-means clustering*). Ovim postupkom se obezbeđuje balansirani raspored tačaka redukovanog oblaka koji zadovoljava polazne pretpostavke. Međutim, potrebno je naglasiti da je ova metoda pogodna samo za oblake tačaka dovoljno velike gustine i visokog nivoa aproksimacije skeniranog objekta koji mogu omogućiti kvalitetno izračunavanje svih potrebnih parametara procesa. Značajno poboljšanje generalizacije metode i postizanje značajnijih rezultata ostvareno je uvođenjem filtriranja oblaka pre samog postupka redukcije.

Jedna iterativna metoda redukcije oblaka pomoću grupisanja tačaka u klasterne predstavljen je u [65]. Klasterovanje se vrši korišćenjem metodologije formiranja tzv. Voronoijevih ćelija (engl. *Voronoi cells*) za izabrane reprezentativne tačke iz oblaka. Ovo znači da će se klasteri formirati tako da će svaka pripadajuća tačka jednog klastera biti bliže svom reprezentu nego reprezentu bilo kog drugog okolnog klastera. Postupak je iterativan i nakon inicijalne podela iz svakog klastera se izdvajaju tačke koje poseduju najmanju distancu do tangentne ravni tog klastera. One se proglašavaju novim reprezentativnim tačkama i postupak klasterovanja se ponavlja sve dok novoformirani klasteri ne zadovolje zadatu definisanu minimalnu geometrijsku devijaciju (u odnosu na polazni oblak). Za konačno izabrani redukovani oblak proglašava se skup poslednje izabranih reprezentativnih tačaka. Međutim, ova tehnika može dovesti do gubljenja značajnih tačaka koje reprezentuju ivice površina skeniranih objekata. Zbog ove nepovoljne osobine isti autori u radu [66] predstavljaju dopunjenu tehniku uvođeci algoritam ekstrakcije ivica iz oblaka tačaka koji primenjuju pre samog postupka redukcije. Rad predstavljen u [67] donosi sličan iterativni pristup prethodo opisanom postupku. Međutim, ovi autori kao kriterijum za uklanjanje tačaka iz polaznog oblaka koriste prednosti diferencijalne geometrije, odnosno razliku vektora glavnih krivina (engl. *principal curvatures*) površi u određenim tačkama pre i posle redukcije.

Metode redukcije struktuiranih oblaka tačaka se uglavnom zasnivaju na procesiranju skeniranih linija (engl. *scan lines/curves*). Redukcija broja tačaka skeniranih linija se najjednostavnije može izvršiti prostim uniformnim uzorkovanjem svake k -te tačke, gde k predstavlja faktor redukcije. Zbog značajnih nedostataka ovakvog izbora značajnih tačaka, postoje i druge razvijene metode uzorkovanja zasnovane na geometrijskoj oceni [3].

Jedna poznata metoda redukcije skeniranih linija zasniva se na pojmu ravnosti krive koju tačke konture opisuju (engl. *straightness method*). Ova metoda se oslanja na proveru da li svaka posmatrana tačka konture, u korelaciji sa dve njene najbliže (prethodna i naredna u konturi), može kreirati prividnu pravu liniju sa zadatom tolerancijom ravnosti (uglova preloma). Geometrijski uslov ove metode predstavljen je i matematički definisan u [68]. U radu su opisana i dva izvedena parametra koja se mogu porediti sa unapred zadatom vrednošću (pragom) kojom se proces redukcije može kontrolisati. Provera uslova se vrši za sve tačke u konturi redom i ukoliko je bilo koji od ova dva parametra veći od zadatog praga tačka se zadržava, dok se u suprotnom eliminiše iz konture.

Još jedna široko korišćena metoda je zasnovana na Euklidovoj distanci između tačaka (engl. *spatial method*). U ovoj metodi se počev od prve tačke, sve naredne, koje su na međusobno manjem rastojanju od zadatog, odstranjuju iz oblaka. Takođe, poznata je i metoda redukcije na osnovu devijacije visine tetive (engl. *chordal method*). Ova metoda razmatra najkraća rastojanja tačaka od određenih tetiva skenirane linije. U ovom slučaju, tačka konture biće izdvojena kao reprezentativna ukoliko je njena najkraća distanca od tetive (kreirane između njene dve najbliže tačke) duža od zadatog praga. Kao dodatni uslov može se koristiti i distanca merena od posmatrane tačke do tetive konstruisane između njene prethodne i druge naredne u nizu. U radu [69] objašnjeni su nedostaci dve prethodno opisane metode redukcije i predstavljena je metoda zasnovana na razlici visina (engl. *height decision method*) dve susedne tačke u konturi, a na osnovu predefinisane distance. Ideja redukcije je da se posmatrana tačka, ukoliko se nađe na manjoj visinskoj razlici (posmatrano duž ose pravca skeniranja laserskim skenerom) od poslednje zadržane tačke u konturi, odstranjuje iz konačnog skupa. Sa druge strane, uko-

liko se u duplom opsegu zadate distance od prethodno zadržane tačke ne nađe naredna reprezentativna tačka konture, linearnom interpolacijom po obe koordinatne ose, kreira se nova dodatna tačka.

Kod navedenih metoda redukcije skeniranih linija vrlo je teško spoznati vezu između zadatih parametara metode (nivoa redukcije) i nivoa devijacije koja se na taj način unosi u informaciju o skeniranom objektu. Tačnije, ne postoji jasna povratna informacija koja bi omogućila neku vrstu odlučivanja tokom same primene metode. Zbog ovog nedostatka, značajan broj postojećih metoda redukcije je dopunjen nekim naprednijim metodama odlučivanja. U [70] predstavljeno je unapređenje redukcije na osnovu minimalne dozvoljene Euklidove distance korišćenjem fazi logike za generisanje bročane ocene kvaliteta same redukcije. Ova ocena predstavlja vrstu povratne sprege pri podešavanju ulaznog parametara redukcije i čijim poređenjem sa unapred zadatim pragom se značajno povećava sofisticiranost samog procesa. Za formiranje fazi skupova podataka se, pored vrednosti minimalne distance između tačaka, koristi i maksimalna greška redukcije koja se dobija iz procesa upoređivanja regresionih kriva trećeg reda (metodom najmanjih kvadrata) dobijenih od tačaka skenirane konture pre i posle redukcije. Takođe, sam proces podešavanja funkcija pripadnosti i mehanizam odlučivanja (*Mamdani* metod) su detaljno objašnjeni u navedenom radu. U radovima [71] i [72] predstavljena je slična metodologija korišćenja fazi logike, ali ovoga puta za unapređenje metode redukcije proverom tetivne visine i proverom ravnosti, respektivno.

Primer upotrebe neuniformnih 2D mreža pri redukciji oblaka tačaka predstavljen je u [73]. Autori ovog rada predlažu projektovanje strukturiranog oblaka tačaka (dobijenog laserskim skenerom) na ravan podeljenu neuniformnom 2D mrežom čiji pravci prostiranja odgovaraju pravcima skeniranih linija oblaka. Nakon projektovanja oblaka, iz svakog polja mreže se izdvajaju tačka sa srednjom vrednošću visine od ravni projektovanja kao i tačka sa najvećim uglom vektora normale dobijenim njenim upoređivanjem sa okolne dve tačke u skeniranoj liniji (engl. *angular deviation method*). Ove dve tačke zajedno sa izdvojenim tačkama iz svih ostalih polja čine redukovani oblak tačaka. Treba napomenuti da je formiranje 2D mreža u pomenutom radu delimično zasnovano na korisničkoj asistenciji, kao i da se može napraviti podela na dva pristupa u zavisnosti od oblika korišćene mreže - metoda sa jednodirekcionim i bidirekcionim mrežama.

Kao posebna grupa tehnika redukcije skeniranih linija izdvajaju se one koje koriste osobine postojećih regresionih metoda da bi izvršili selekciju i odstranjivanje tačaka sa najnižim uticajem na informaciju o geometriji samih kontura. Jedan reprezentativni rad iz ove grupe je predstavljen u [74]. Autori u ovom radu koriste osobine Akima splajna kako bi izvršili redukciju oblaka tačaka dobijenih skeniranjem kontaktnim metodama, odnosno korišćenjem mernog senzora na NUMA. Za svaku skeniranu konturu polazni korak procesa redukcije predstavlja izbor određenog broja izabranih tačaka na osnovu kojih će biti konstruisan Akima splajn. Izbor ovih tačaka se vrši na osnovu poređenja uprošćeno formulisanih derivata regresione funkcije izračunatih u svake dve uzastopne tačke konture. Ukoliko njihov odnos zadovolji postavljeni početni uslov, prva od te dve tačke se dodaje u grupu izabranih. U drugom koraku se određenom strategijom proveravaju devijacije (najkraće distance) većeg broja ostalih originalnih tačaka konture od Akima splajna konstruisanog na osnovu prethodno izabrane grupe tačaka. Tačka sa najvećom vrednošću devijacije se upoređuje sa unapred zadatim pragom i, ukoliko je devijacija veća, ta tačka se dodaje polaznom skupu izabranih tačaka za konstruisanje splajna. Postupak konstruisanja Akima splajna i dalji izbor tačaka se iterativno ponavljaju sve dok se ne nađe tačka sa distancom većom od zadatog praga. Konačno izabrani skup tačaka ujedno predstavlja i traženu redukovanu konturu. Naravno postupak se ponavlja za svaku skeniranu konturu dok se ne dobije kompletan redukovani oblak tačaka. Autori ovog rada posebno naglašavaju da je za konstruisanje Akima splajna neophodno imati tačke skenirane linije sortirane u rastućem nizu posmatrano po apscisi pa je u svakoj iteraciji postupka pri dodavanju nove tačke u polazni skup potrebno pravilno sortirati redosled tačaka. Ova metoda je direktno upoređena sa metodom devijacije visine tetive i pokazala je značajno bolje rezultate u smislu ostvarivanja zadate devijacije znatno većim stepenom redukcije oblaka. Slična grupa autora u svom radu [75] predstavlja unapređenje ove metode regresije upotrebom tzv. bi-Akima splajna. Ova metoda je unapređena i u smislu mogućnosti procesiranja i tzv.

3D skeniranih kontura odnosno kontura čije se tačke ne moraju sadržati u istoj ravni (npr. kontura oblika helikoide).

Postoje i metode redukcije oblaka tačaka koje za procesiranje skeniranih kontura oblaka koriste poznate tehnike obrade signala. Ovakvom strategijom se u određenim slučajevima može doći do informacija koje mogu pomoći u prepoznavanju i odabiru značajnih tačaka, odnosno onih koje treba zadržati nakon procesa redukcije. Jedan primer ovakve metode predstavljen je [76]. U ovom radu autori koriste tzv. kontinualnu vejevlet transformaciju (CWT) za procesiranje 2D skeniranih kontura 3D oblaka tačaka dobijenih nakon struktuiranja. Pošto se ovako definisana transformacija može primeniti samo na jednodimenzionalne signale, 2D skenirane konture se posmatraju i procesiraju kao dva nezavisna 1D signala duž dva pravca - x i y . Na ovaj način moguće je generisati vejevlet koeficijente detalja (u svakoj tački dekomponovnih signala) čije lokalne promene vrednosti ukazuju na mesta (tačke) značajne promene oblika signala (geometrije konture oblaka). Uzimajući u obzir željeni stepen redukcije, odnosno zadati nivo vrednosti koeficijenta detalja, značajne tačke konture se mogu detektovati i izdvojiti u grupu tačaka redukovano oblaka, dok se duplikati (koji se mogu javiti zbog procesiranja duž oba pravca) odstranjuju.

Kao što je naglašeno u poglavlju 2.3 mnogi 3D skeneri, pri skeniranju iz jednog pravca (engl. *single view scanning*), primarno formiraju struktuirane oblake tačaka. Ovi oblaci nakon procesa registracije i integracije ipak poprimaju nestruktuirani oblik. Međutim, tehnike koje redukuju struktuirane oblake se mogu primeniti neposredno posle skeniranja, a pre procesa registracije i integracije. Zbog ove mogućnosti, ali i uporedivo veće brzine izvršavanja kao i lakše implementacije, tehnike za redukciju struktuiranih oblaka ne gube na značaju u odnosu na one koje procesiraju nestruktuirane. Takođe, pre upotrebe metode redukcije, mogu se koristiti razne metode uklanjanja šuma ili odstranjivanja tačaka koje predstavljaju grešku skeniranja i ne pripadaju skeniranoj geometriji (engl. *outliers, spikes*) [73]. Sa druge strane, u mnogim industrijskim aplikacijama, struktuirani oblak dobijen skeniranjem iz samo jednog pravca je dovoljan za ostvarivanje predviđenih ciljeva.

Na kraju ovog segmenta treba navesti da, pored metoda redukcije broja tačaka, postoje i potrebe da se u fazi pre-procesiranja izvrši proširivanje, odnosno inteligentna dopuna oblaka dodatnim tačkama. Tehnike razvijene u ove svrhe mogu se koristiti zarad obezbeđivanja bolje uniformnosti i gustine oblaka u situacijama kada je on kreiran u niskoj rezoluciji, kao i zbog omogućavanja i olakšavanja izvođenja dalje obrade oblaka nekom od potrebnih tehnika procesiranja. Takođe, u nekim slučajevima ovakve tehnike mogu doprineti i popunjavanju nekih segmenata oblaka na mestima gde je, usled određenog problema pri skeniranju, određeni manji segment oblaka izostao. Kao reprezentativne primere značajnih naučnih doprinosa u ovoj oblasti možemo navesti radove [77, 78]. Zbog veoma retke potrebe za upotrebom ove vrste procesiranja oblaka tačaka u industrijskoj primeni reverznog inženjerstva u ovom radu neće biti detaljnije diskutovano o tehnikama iz ove grupe.

3.2. Metode za segmentaciju i prepoznavanje geometrijskih primitiva iz oblaka tačaka

Metode za segmentaciju i prepoznavanje geometrijskih primitiva zauzimaju značajno mesto u istraživanjima i razvoju u oblasti RE. Iako je veliki broj istraživanja na ovu temu rađen u bliskoj prošlosti, ova oblast i danas zauzima značajno mesto u istraživačkim okvirima i kao takva je vrlo aktuelna. Ovo proizilazi iz činjenice da se većina do sada razvijenih tehnika u velikoj meri oslanja na korisničku asistenciju. Glavni cilj savremenih istraživanja predstavljaju algoritmi koji bi obezbedili autonomni (automatski) proces prepoznavanja i ekstrakcije geometrije iz oblaka tačaka. U skladu sa ovim napomenama u nastavku je dat pregled najznačajnijih dosadašnjih dostignuća u ovom segmentu RE.

Klasifikacija metoda za prepoznavanje primitiva može se izvršiti po nekoliko kriterijuma. Neki od njih se primenjuju nad izvornim oblacima tačaka, dok su drugi projektovani da bi se koristili za procesiranje 3D mreža. Sa druge strane, jedna grupa tehnika prepoznaje i izdvaja samo jedan ciljani tip primitiva (npr. samo ravni ili samo cilindre), dok druga grupa može izdvajati više različitih primitiva

skvencijalno ili pak simultano. Treba naglasiti da je dizajniranje i razvoj algoritama prepoznavanja što više različitih tipova primitiva u direktnoj vezi sa stepenom mogućnosti njihove generalne primene. Iako je pomenuta generalna upotreba algoritama prepoznavanja bila (i još uvek je) cilj mnogih istraživanja u ovoj oblasti, do sada su razvijene zadovoljavajuće metode koji se mogu primeniti samo u specifičnim i kontrolisanim uslovima.

Posmatrajući strukturu i procese do sada razvijenih metoda prepoznavanja primitiva, njihova klasifikacija se može izvršiti na:

- metode kojima je potrebna prethodna segmentacija oblaka tačaka, i
- metode koje vrše direktnu ekstrakciju primitiva iz oblaka.

Karakteristika metoda prve grupe je inicijalna segmentacija (podela) oblaka na regione (klastere) koji reprezentuju delove različitih gradivnih primitiva skeniranog objekta. Segmentacija može biti izvršena direktno nad oblakom ili 3D mrežom i po pravilu je zasnovana na strategiji širenja regiona (engl. *region growing*). Proces širenja regiona odnosno formiranja klastera tačaka počinje izborom inicijalne tačke (engl. *seed point*), grupe tačaka ili pak inicijalne triangularne površine (pri segmentaciji mreža). Iterativnim postupkom se zatim klaster širi dodavanjem okolnih tačaka (ili triangularnih površina), a po određenom geometrijskom kriterijumu. Cilj tehnika ove grupe je da oblak tačaka podeli na klastere koji predstavljaju gradivne primitive posmatranog skeniranog objekta. Glavni nedostatak ove grupe metoda je visoka zavisnost od kvaliteta samog procesa segmentacije ali u velikom broju slučajeva i velika dužina trajanja izvršavanja algoritama.

Jednu od poznatih metoda navedene grupe predstavili su T. Rabbani i saradnici u [6]. Ova metoda segmentacije oblaka zasnovana je na uslovu kontinualnosti površine (engl. *smoothness constraint*) koji pogoduje detekciji glatko spojenih regija (površina) u oblaku. Rast regiona pri formiranju klastera je zasnovan na poređenju vrednosti normala u tačkama i njihovih reziduala, sračunatih pri regresiji ravni u izabranoj okolini posmatrane tačke, sa unapred zadatim pragovima. Okolina posmatrane tačke se određuje unapred zadatim parametrima na osnovu jednog od dva opšte poznata algoritma - k najbližih suseda (knn) i suseda na definisanoj distanci (fdn). Dodatni uslov pripadnosti neke tačke klasteru je njena udaljenost od lokalnih suseda (susednih tačaka). Korišćenjem predstavljenih uslova mogu se detektovati sve tačke koje pripadaju istoj glatkoj površini, dok se izborom vrednosti potrebnih pragova može kontrolisati pojava presegmentacije.

U [79] predstavljena je segmentacija oblaka zasnovana na fazi logici. Za formiranje fazi skupova podataka autori koriste tri geometrijska parametra koji opisuju oblik i karakteristike površine u okolini posmatrane tačke. Fazi logikom se kreiraju pravila po kojima se za određenu kombinaciju vrednosti tri navedena parametra svaka tačka oblaka može klasifikovati u jednu od 11 nezavisnih grupa, odnosno tipova površi. U postupcima rasta regiona za svaku navedenu klasu proveravaju se pripadnosti tačaka redom, počev od inicijalne. Inicijalna tačka postaje ona tačka koja poseduje najveći stepen pripadnosti traženoj klasi površine. Ista grupa autora je predstavila sličnu metodu i u [80], ali ovoga puta je prikazana njena upotreba za segmentaciju određenih površina drugog reda.

U [81] autori su predstavili metodu za prepoznavanje ravni u struktuiranim oblacima tačaka zasnovanu na vejevlet transformaciji. U ovoj metodi upotrebljena je diskretna vejevlet transformacija (DWT) odnosno specifična osobina ortonormalnosti tzv. Dobeši db2 vejevleta (engl. *Daubechies wavelet*). Posmatranjem skeniranih linija oblaka kao 2D signala i njihove obrade vejevlet transformacijom moguće je, posmatranjem vrednosti koeficijenata detalja u svakoj tački, uočiti pozicije značajne promene geometrije odnosno granične tačke segmenata kontura. Na ovaj način moguće je izvršiti preciznu segmentaciju svih kontura oblaka čime se dolazi do klastera tačaka koji reprezentuju različite geometrijske primitive. Takođe, uzastopne varijacije vrednosti koeficijenata detalja mogu identifikovati i odvojiti segmente koji reprezentuju ravni (prave linije) od onih koji reprezentuju geometriju višeg reda. Nakon ovakve segmentacije svih skeniranih linija, estimacija parametara ravni može se vršiti standardnim tehnikama regresije.

Još jedna zanimljiva metoda za struktuirane oblake predstavljena je u [82]. Autori u ovom radu

prezentuju rezultate primene svog algoritma za prepoznavanje ravni, cilindara i sfera u struktuiranim oblacima dobijenim skeniranjem laserskim skenerom pričvršćenim na industrijskom antropomorfnom robotu. Metoda se temelji na inicijalnoj segmenaciji 2D skeniranih kontura oblaka na linijske i eliptične segmente. Nakon izvršene segmentacije svih kontura pristupa se grupisanju njihovih izdvojenih segmenata čime se dobijaju 3D klasteri koji reprezentuju delove traženih vrsta površi. Proces 3D grupisanja počinje od automatskog izbora jednog segmenta neke od skeniranih kontura i izvodi se procesom rasta regiona odnosno iterativnim dodavanjem ostalih segmenata okolnih kontura prema definisanom kriterijumu koplanarnosti (engl. *co-planarity*).

Metode koje vrše segmentaciju i ekstrakciju primitiva iz 3D mreža koriste podatke o normalama u tačkama kao i parametrima krivina (dobijenih uglavnom postupcima diferencijalne geometrije) za različite strategije rasta regiona. U [83] autori predstavljaju metodu segmentacije 3D mreža zasnovanu na rastu regiona počev od inicijalne grupe temena trouglova. Inicijalne grupe se formiraju na osnovu upoređivanja znakova Gausovih i srednjih vrednosti krivina (engl. *Gaussian and Mean curvatures*) i vektora normala izračunatih u svakom temenu na osnovu aproksimacije polinomima drugog stepena. Rast regiona je zatim zasnovan na iterativnom postupku pri čemu se u prvoj iteraciji nad inicijalnom grupom temena formira bikubna Bezijerova (fran. *Bézier*) površ i zatim se u tekući klaster dodaju do tada neiskorišćene tačke koje po određenom kriterijumu pripadaju (odnosno bliske su) kreiranoj površi. Postupak se dalje ponavlja sve do postizanja određenog geometrijskog kriterijuma, s tim što se u svakoj iteraciji površ ponovo kreira nad celokupnom do tada izdvojenom grupom tačaka.

Sa druge strane u [84] predložena je metoda prepoznavanja kvadraka u 3D mrežama. Ova metoda zasnovana je na segmentaciji koja se sprovodi u dve faze. U prvoj fazi se oko inicijalnih temena formira grupa tačaka rastom regiona na osnovu parametara zakrivljenosti koji se poredi sa unapred zadatim pragom. Na osnovu toga se formiraju manje grupe tačaka kako bi se odredio tip površi. U drugoj fazi segmentacije vrši se finalno formiranje klastera. Ova faza se izvodi širenjem regiona tako što se u svakom potencijalno pridruženom temenu poredi njegova normala sa srednjom vrednošću normale u do tada izdvojenom klasteru. Wang i saradnici su u [85] predstavili svoj pristup 3D rekonstrukciji modela koji se takođe zasniva na segmentaciji mreže. Proces počinje prvobitnim filtriranjem inicijalne 3D mreže, odnosno uklanjanjem prepoznatog šuma nastalog u procesu skeniranja. Zatim se vrši segmentacija različitih tipova površi sekvencijalno u tri faze. Prvo se vrši segmentacija ravni zasnovana na strategiji rasta regiona poređenjem pravaca normala okolnih trougaonih segmenata počev od jednog nasumično izabranog trougla mreže. Nakon toga sledi segmentacija kvadraka zasnovana na statističkim metodama i Gausovim mapama [86] i na kraju izdvajanje tzv. površina slobodne forme (engl. *free form*) korišćenjem regresije površina na osnovu bikubnog B-splajna (engl. *bicubic B-spline*). Poslednja faza ove metode posvećena je kompjuterskom modeliranju prepoznatih površina i njihovom povezivanju u jedinstveni model posmatranog objekta.

Jedna drugačija metoda prepoznavanja primitiva koja se zasniva takođe na segmentaciji 3D mreža predstavljena je u [87]. Predložena segmentacija se karakteriše kao hijerarhijsko deljenje (klasterovanje) i u tom procesu se formira binarno stablo klastera. Na početku procesa se svaki pojedinačni trougao mreže smatra posebnim klasterom, a zatim se iterativno klasteri proširuju. U svakoj iteraciji vrši se aproksimacija ravni, cilindara i sfera na osnovu do tada formiranog klastera i dodatnog susednog trougla. Korišćenjem iste metrike vrši se izračunavanje greške aproksimacije u sva tri slučaja i donosi se odluka o tipu površi kao i o tome da li će se do tada formirani klaster dalje proširiti. Na kraju procesa dobijaju se konačno obrazovani klasteri trouglova koji reprezentuju tri izabrana tipa površina.

Kada je reč o tehnikama koje vrše direktnu ekstrakciju primitiva iz oblaka, svakako najznačajnije mesto zauzimaju one zasnovane na tzv. RANSAC strategiji. RANSAC je prvobitno nastao za potrebe prepoznavnja geometrijskih oblika u 2D prostoru tačaka ili piksela slike pa je kao takav našao upotrebu i u 3D prostoru. Ova strategija primenjuje iterativne procedure u kojima se vrši nasumično izdvajanje minimalnog skupa tačaka, estimacija parametara potrebnog primitiva, a zatim i izdvajanje svih tačaka koje, po unapred zadatim kriterijumima, zadovoljavaju uslov pripadnosti tekućem primitivu. Sve izdvojene grupe tačaka, odnosno potencijalne površi, se porede na osnovu grešaka aproksimacije i

broja izdvojenih tačaka, na osnovu čega se donosi odluka o tome koja od njih je najbolja da bi postala konačno izdvojeni segment. Broj iteracija za izdvajanje svih potrebnih površi obično se izračunava upotrebom nekog od modela teorije verovatnoće. Treba naglasiti da, samim tim što opisani proces zavisi od nasumično izabranih tačaka i potrebnog velikog broja iteracija, on može (u zavisnosti od vrednosti unapred zadatih pragova) često dovesti i do pogrešno prepoznatog i izdvojenog tipa primitiva. Zbog toga mnoge metode ove grupe poseduju različite dodatne tehnike validacije, odnosno provere kvaliteta izdvojenih površina. Pionirsko istraživanje na temu primene RANSAC-a za ove potrebe u 3D prostoru prezentovano je u [88]. Schnabel i njegovi saradnici su značajno kasnije u [89] predstavili unapređenu metodu koju su nazvali efikasni RANSAC (eRANSAC). Ova automatska metoda prepoznavanja primitiva se izdvaja kao jedna od najrelevantnijih u svojoj grupi i poseduje mogućnost prepoznavanja većeg broja osnovnih oblika (ravni, cilindra, torusa, sfera i konusa). Kroz opisane iterativne postupke (specifične za RANSAC metodologiju) prikupljaju se kandidati odnosno grupe tačaka sa njihovim pripadajućim normalama i vrši se njihova ocena podobnosti funkcijama posebno razvijenim za svaki od traženih oblika pojedinačno (engl. *score functions*). Autori rada su ipak zaključili da njihov algoritam postaje nestabilan u uslovima prisustva povećanog nivoa šuma u oblaku i da je tada teško adekvatno podesiti potrebne parametre (pragove). Međutim, u nekim regularnim situacijama daje veoma dobre rezultate. Takođe potrebno je istaći da je ovaj algoritam specifičan i po razvijenoj metodi donekle supervizijskog izbora minimalnog skupa nasumičnih tačaka pri traženju pojedinih vrsta primitiva.

Značajno mesto u razvoju tehnika direktne ekstrakcije primitiva iz oblaka zauzima i Hofova transformacija (HT) koja, u određenim situacijama, može biti računarski i memorijski vrlo zahtevna [90]. U [91] autori su iskoristili Hofovu transformaciju za detekciju cilindra, odnosno estimaciju njihovih pet parametara - orijentacije ose cilindra u sfernim koordinatama (θ , ϕ), radijusa i pozicije cilindra u ravni upravnoj na njegovu osu. U opštem slučaju za estimaciju ovih parametara potrebna je Hofova transformacija u 5D prostoru koja bi bila izuzetno računski zahtevna. Međutim, autori ove tehnike koriste princip estimacije parametara u dve faze. U prvoj fazi koriste Gausovu sferu [86] i 2D HT za estimaciju orijentacije ose cilindra. U sledećoj fazi koristi se 3D HT za generisanje informacija o poziciji i radijusu cilindra. Navedeni pristup direktno omogućava smanjenje vremena potrebnog za izvršavanje opisanog algoritma. Treba napomenuti da Gausova sfera predstavlja grafičku interpretaciju svih vrhova jediničnih vektora normala u svakoj tački oblaka reprezentovanih u vidu tačaka u istom Dekartovom koordinatnom sistemu. Ovakav oblik za neki kompleksniji objekat ima formu sfere pa otuda i sam naziv pojma. Upotrebom ovakve interpretacije grafički se mogu sagledati glavni pravci orijentacije gradivnih primitiva (skeniranog) objekta koji će lako biti detektovani kao oblasti nagomilavanja tačaka na omotaču Gausove sfere. Još jedan primer korišćenja HT u svrhe ekstrakcije primitiva prikazan je u [92]. Autori su upotrebili 3D HT za detektovanje ravni u oblacima tačaka za potrebe kalibracije optičkog skenera dok su isti autori u [93] optimizovali 3D HT za detekciju sfera.

U [94] autori su iskoristili benefite svoja dva prethodna rada [95, 96] i kreirali algoritam za detekciju cilindra, sfera i ravni iz nestruktuiranih oblaka tačaka. Na početku procesa vrši se nasumični izbor inicijalnih tačaka čiji je broj jednak 0,5% ukupnog broja tačaka u oblaku. Pretpostavka je da će se ovim pristupom na svakoj gradivnoj površini objekta (na svakom primitivu) izdvojiti bar jedna inicijalna tačaka čime bi bilo omogućeno da se ta površina prepozna i izdvoji, a samim tim i sve ostale. Zatim se u okolini inicijalnih tačaka izdvaja unapred definisani broj njihovih najbližih tačaka čime se formiraju inicijalni klasteri. Nad svakim klasterom se primenjuju metode regresije sva tri tipa primitiva i izračunavaju se vrednosti odstupanja. Poređenjem ovih vrednosti i izborom najmanje dolazi se do tipa primitiva. Klasteri se zatim procesiraju iterativno vodeći računa o njihovom redosledu. Prvo se vrši izdvajanje cilindra i sfera, a zatim ravni. U svakoj iteraciji se inicijalni klaster proširuje na osnovu estimiranih parametara površi i izdvaja iz oblaka kao prepoznati segment gradivnog primitiva. Glavna karakteristika ove metode je postojanje kreiranih mehanizama validacije, odnosno provere da li je do tada izdvojen primitiv (cilindar ili sfera) zaista strukturno podoban ili je možda došlo do greške u ekstrakciji. Takođe metoda sadrži i postupak spajanja udaljenih delova površi koje pripadaju istom

cilindru, a koji je razvijen na osnovu poznatog *Mean Shift clustering* algoritma [97]. Izdvajanje ravni iz koje se primenjuje na kraju iz preostalih tačaka oblaka je zasnovano na RANSAC metodologiji. Na samom kraju, validacija ravni izvodi se upotrebom algoritma koji uzima u obzir najveće povezane komponente (engl. *largest connected component*) na isti način kao u [89].

Iz velikog broja do sada razvijenih metoda na temu ekstrakcije različitih vrsta primitiva i njihove potencijalne upotrebe kao posebnu grupu možemo izdvojiti one koje su razvijene sa ciljem prepoznavanja cilindara u veoma velikim oblacima (više miliona tačaka), odnosno prepoznavanja cevovoda pri reverznom inženjerstvu velikih industrijskih postrojenja. Ova popularna oblast istraživanja donosi mnoga rešenja i koristi do sada uglavnom poznate tehnike prepoznavanja cilindara, ali ovog puta prilagođene za rad sa veoma velikim brojem tačaka u realnom vremenu. Takođe treba navesti da standardne industrijske cevovode možemo uprošćeno posmatrati kao cilindre čije se ose najčešće prostiru duž tri upravna pravca pa su u skladu sa ovim mnoga pojednostavljena u metodama prepoznavanja donela računске benefite. Algoritam predložen u [98] prvo prepoznaje cevovode (cilindre) upravne na ravan poda postrojenja, a zatim vrši prepoznavnje onih paralelnih tog ravni u dva upravna pravca. Metod za određivanje ravni poda i dva međusobno upravna pravca paralelna sa podom je zasnovan na korišćenju osobina već ranije pomenute Gausove sfere. Nakon određivanja ovih glavnih ravni, cilindri se detektuju na osnovu RANSAC strategije, ali ovoga puta u 2D prostoru, odnosno detekcijom kružnica iz projekcija svih tačaka na tri pomenute glavne ravni. Sa druge strane, algoritam predložen u [99] koristi prednosti više poznatih postojećih tehnika uz izvršene potrebne modifikacije za primenu nad velikim brojem tačaka. Ovaj rad je inspirisan postupkom prikazanim u [91]. Polazni proces ovog metoda je segmentacija oblaka po ugledu na metod prezentovan u [6]. Nakon segmentacije u dve faze vrši se estimacija osa cilindara na osnovu Gausove sfere i HT sa tzv. neuniformnom gustinom čelija. Pozicije cilindara i njihovi radijusi se pronalaze upotrebom algebarske regresije kružnice u skupu tačaka projektovanih na ravan upravnu na estimiranu osu. Autori navode da je glavni nedostatak ove metode veoma veliko vreme potrebno za procesiranje velikog broja tačaka što ograničava njegovu eventualnu praktičnu upotrebu.

Potrebno je napomenuti i da postoje određena istraživanja na temu upotrebe različitih vrsta neuronskih mreža za potrebe procesiranja 3D oblaka tačaka [100, 101, 102]. Međutim, ovi postupci nisu orijentisani samom prepoznavanju primitiva već više semantičkoj segmentaciji prostora i kao takvi zahtevaju dobro registrovane oblake. Njihova upotreba je razmatrana samo u domenu manjih oblaka [103]. Takođe, kod njih su neophodni veoma zahtevni procesi obučavanja mreža i imaju malu sposobnost generalizacije pa nisu pogodni za industrijsku upotrebu u domenu reverznog inženjerstva mašinskih delova.

Ukoliko govorimo o mašinskim delovima, treba napomenuti da je jedan od najzastupljenijih gradivnih primitiva svakako cilindar. Zbog toga se njemu u mnogim metodama prepoznavanja i poklanja najviše pažnje. Imajući u vidu prethodno navedena istraživanja, možemo zaključiti da kvalitet ekstrakcije cilindara iz oblaka zavisi od precizno određenih pravaca njihovih osa. Mnoge tehnike, nakon izvršenog inicijalnog klasterovanja i primene neke od metoda prepoznavanja tipa površi, u slučaju cilindra, prvo vrše estimaciju pravca njegove ose. Ovo umnogome olakšava dalju procenu vrednosti radijusa cilindra zato što je nakon prepoznavanja pravca ose moguće sve pripadajuće tačke klastera projektovati u neku ravan upravnu na tu osu i procesirati nekim od brojnih tehnika za regresiju kružnice. Ovakvim postupcima se 3D problem prepoznavanja radijusa cilindra svodi na 2D problem koji je lakši i precizniji i upravo iz ovih razloga biće skrenuta pažnja na neke od relevantnih tehnika za estimaciju osa cilindara kao i na neke radove kojima je prepoznavanje cilindara u fokusu.

Već pomenuta upotreba Gausove sfere i RANSAC strategije za određivanje pravca ose cilindra prvi put je predstavljena u [104]. Postupak je predviđen za primenu nad Gausovom sferom klastera tačaka za koji se unapred zna da predstavlja deo površi jednog ili više paralelnih cilindara. Teorijski gledano, Gausova sfera skupa tačaka (sa pridruženim normalama) koji reprezentuje cilindar poseduje oblast oblika prstena na kome postoji visoka koncentracija tačaka - znatno gušća nego u ostalim zonama na površini sfere. Imajući ovo u vidu potrebno je naći ravan koja najbolje opisuje ovaj prsten (krug)

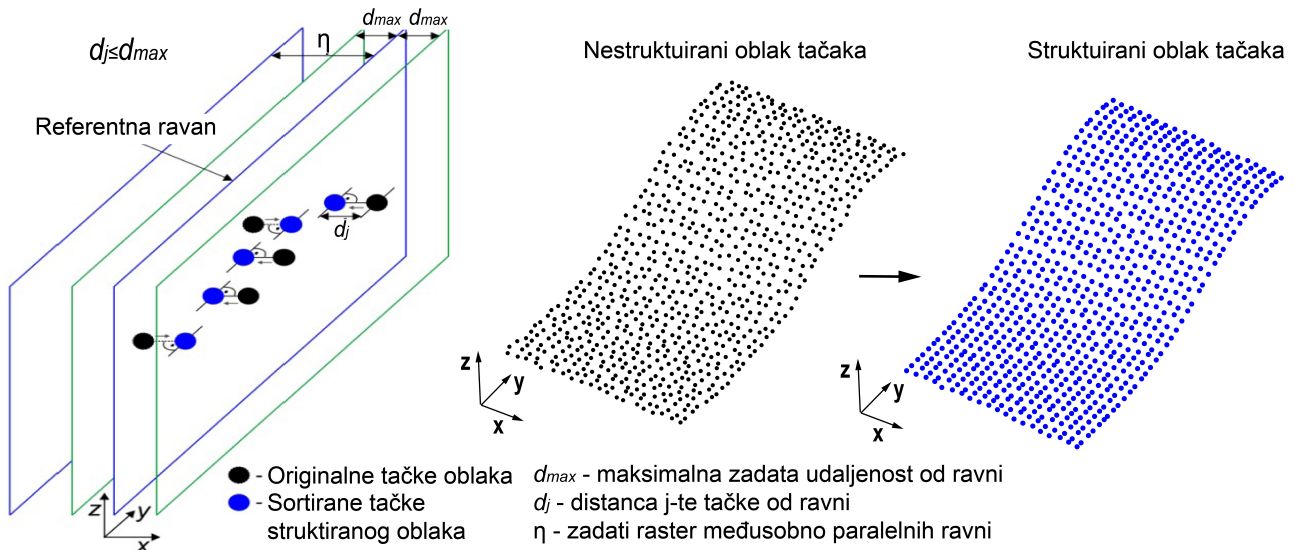
tj. koja u svojoj unapred definisanoj okolini sadrži najveći broj pomenutih tačaka prstena. Samim tim, pravac normale ove ravni ujedno predstavlja i pravac ose cilindra kog klaster opisuje. Sam postupak iz [104] je iterativan. U svakoj iteraciji se nasumično vrši izbor dve tačke iz sfere na osnovu kojih se, zajedno sa koordinatnim početkom, vrši estimacija parametara zamišljene ravni. Zatim se u unapred definisanoj bliskoj okolini ravni proverava broj pripadajućih tačaka (engl. *inliers*). Nakon završetka svih (unapred zadati broj) iteracija, bira se najbolja ravan, odnosno ona koja je sadržala najveći broj pripadajućih tačaka i njena normala samim tim predstavlja i osu cilindra. Ovakav postupak, iako je pokazao neke svoje dobre osobine, ipak se zasniva na nasumično izabranim tačkama i kao takav se ne smatra naročito pouzdanim u opštoj upotrebi.

U [95] autori za estimaciju pravca ose cilindra koriste pravac sopstvenog vektora koji odgovara najmanjoj sopstvenoj vrednosti kovarijantne matrice skupa normala tačaka. Sa druge strane, za ove svrhe u radovima [105, 106] je iskorišćena analiza glavnih komponenti (PCA) sistema tačaka. Prva glavna komponenta (vektor) sistema tačaka definiše pravac najviše varijabilnosti podataka što su u ovom slučaju koordinate tačaka. Zbog ove osobine pravac ovog vektora se može smatrati i pravcem ose cilindra koga zadati skup tačaka reprezentuje. Pored PCA postoji i njena robusnija verzija (RPCA) [107] koja je na isti način iskorišćena za potrebe određivanja ose cilindra u [108]. Obe metode analize glavnih komponenti koje prirodno pokazuju glavne pravce varijabilnosti koordinata tačaka mogu dovesti do greške pri njihovoj upotrebi za estimaciju osa niskih cilindara (pre svega većeg prečnika) zbog jasnog nedostatka dovoljne varijanse koordinata u potrebnom pravcu. O ovome će npr. biti više reči u poglavlju 7.3. Zbog svega navedenog, trenutno prisutne tehnike estimacije pravaca osa cilindara prepoznatih u oblacima tačaka ne pokazuju zadovoljavajući nivo potencijalne generalne upotrebe i zbog toga su i dalje u fokusu mnogih istraživanja.

4. Strukturiranje oblaka tačaka

Kao što je već naglašeno u poglavlju 2, u fazi pre-procesiranja oblaka tačaka pored raznih drugih metoda često se upotrebljava i proces strukturiranja. Strukturiranje oblaka tačaka, kao što i samo ime govori, predstavlja proces (pre)uređivanja pozicija tačaka i konfiguracije njihovog digitalnog zapisa za potrebe mogućnosti primene nekih narednih tehnika obrade koje zahtevaju uređenu formu. Glavni cilj ovakvog procesiranja je da se promena pozicija (ili dodavanje novih) tačaka izvrši tako da se u najvećoj mogućoj meri zadrži originalna informacija dobijena samim postupkom skeniranja. Ovo znači da se generisani nestruktuirani (haotični) oblaci mogu, upotrebom ove vrste tehnika, pretvoriti u strukturane forme. Međutim, i pored česte potrebe za primenom strukturiranja, u praksi se ne može naći mnogo referentnih radova koji se bave konkretno ovom temom. Od nekolicine postojećih tehnika strukturiranja, u ovom poglavlju biće pojašnjena tehnika korišćena u [72]. Takođe, u nastavku ovog poglavlja biće prezentovana i jedna metoda strukturiranja oblaka koja je opisana u [81].

Strukturiranje oblaka tačaka iz [72] može se smatrati vrlo intuitivnom metodologijom. Ova metoda je zasnovana na formiranju skeniranih linija korišćenjem zamišljenih paralelnih ravni određenog predefinisiranog razmaka (rastera). Proverom svih okolnih tačaka jedne posmatrane ravni na osnovu unapred definisanog maksimalnog rastojanja (tačke od ravni) odlučuje se da li će se te tačke translirati i na taj način pridružiti toj ravni. Postupak se izvršava za sve definisane ravni pri čemu broj ravni određuje broj generisanih skeniranih linija strukturiranog oblaka na izlazu. Korisnik procesa ima mogućnost da definisanjem rastera između ravni i maksimalne distance tačaka od tih ravni postigne zadovoljavajuću strukturu. Treba napomenuti da se orijentacija zamišljenih ravni može definisati po potrebi, ali da se iz praktičnih razloga često koriste pravci paralelni sa nekom od globalnih koordinatnih ravni oblaka. Takođe, ova metoda nudi strukturiranje samo po, uslovno rečeno, jednoj osi, odnosno formira se uniformna distanca samo duž pravca normala zadatih ravni. Ovo znači da razlika koordinata tačaka posmatrano duž pravca paralelnog ravnima ostaje onakva kakva je bila na originalnom nestruktuiranom oblaku, odnosno neuniformna. Grafički prikaz predstavljene metode strukturiranja dat je na slici 4.1.

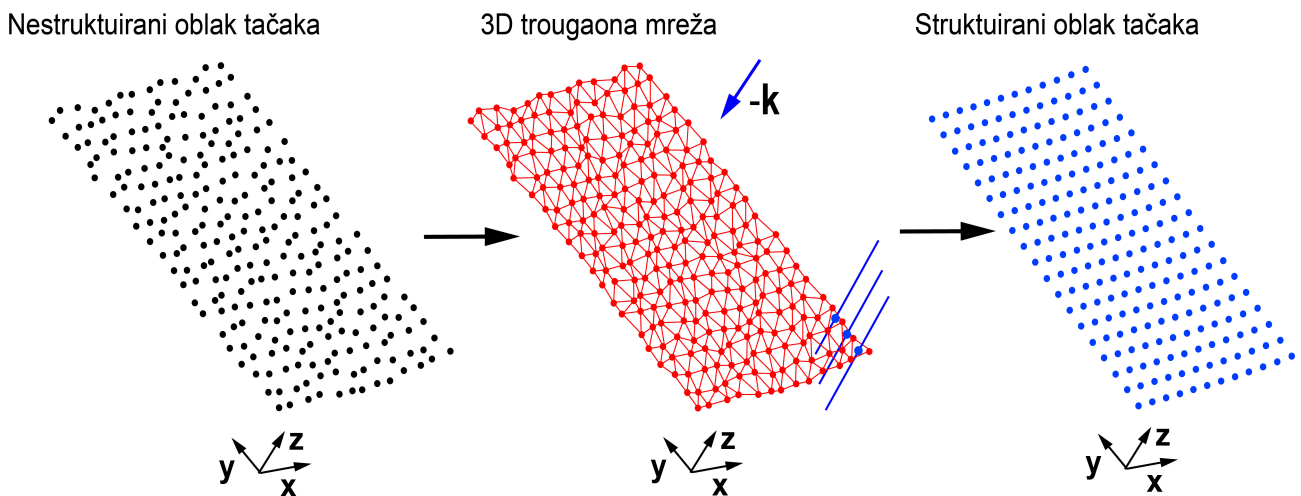


Slika 4.1: Metoda strukturiranja oblaka tačaka upotrebom paralelnih ravni u zadatom rasteru [72].

Metoda strukturiranja oblaka koja je prezentovana u ovom radu temelji se na ideji iz [81]. Osnovu metode predstavlja upotreba tzv. z bafer (engl. *z-buffer*) algoritma [109]. Ovaj algoritam je kreiran za potrebe detekcije vidljivih površina objekata, a za izabrani pravac posmatranja. Algoritam je baziran na detekciji prodora vektora pravca posmatranja kroz geometrijski opisane fragmente površina kompjuterskih objekata. Ukoliko se matematički dokaže da prodor vektora posmatranja, odnosno njegovog

pravca (prave definisane u prostoru) kroz površinu postoji, onda se ta površina objekta smatra vidljivom iz posmatranog pravca i kao takva se može npr. prikazati na ekranu. Iako deluje vrlo intuitivno, pojava inicijalnog z bafer algoritma je donela značajne inovacije u kompjuterskoj grafici.

U algoritmu strukturiranja oblaka tačaka osobine z bafera su iskorišćene na nešto drugačiji način. Nestruktuirani oblaci generisani modernim 3D skenerima često su odmah nakon skeniranja tretirani metodama kreiranja poligonalnih mreža [110, 111, 112] čime se dobijaju najčešće 3D trougaone mreže. Takođe, kao čest izlaz iz procesa skeniranja može se susresti i forma opisivanja skeniranih površina skupovima trouglova čime se formira poznati STL fajl format. Imajući ovo u vidu, jasno je da se za zadati vektor posmatranja mogu pronaći pozicije svih postojećih tačaka njegovog prodora kroz trouglove koji opisuju površine skeniranog objekta. Dodajući pritom i određeni inicijalni raster slike, odnosno skup strukturno raspoređenih tačaka u kojima će se kroz iteracije pozicionirati prave pravca vektora posmatranja, moguće je generisati strukturirani oblak tačaka koji sa određenim stepenom aproksimacije reprezentuje površine skeniranog objekta.



Slika 4.2: Grafički prikaz strukturiranja nestruktuiranih oblaka tačaka korišćenjem z bafer metodologije.

Detaljno predstavljanje algoritma treba početi od matematičkog opisa problema određivanja postojanja i pozicije tačke prodora pravca zadatog vektora kroz trougao u opštem slučaju. Neka su dati vektor $\mathbf{v} = [a \ b \ c]^T$ i tačka $\mathbf{P}_0(x_0, y_0, z_0)$ kroz koju pravac ovog vektora prolazi. Takođe, neka je proizvoljan trougao u prostoru opisan temenima $\mathbf{P}_1(x_1, y_1, z_1)$, $\mathbf{P}_2(x_2, y_2, z_2)$ i $\mathbf{P}_3(x_3, y_3, z_3)$ i neka su matrice potrebne za izračunavanje parametara prodora oblika:

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} x_1 - x_0 & x_1 - x_3 & a \\ y_1 - y_0 & y_1 - y_3 & b \\ z_1 - z_0 & z_1 - z_3 & c \end{bmatrix}, & \mathbf{A}_2 &= \begin{bmatrix} x_1 - x_2 & x_1 - x_0 & a \\ y_1 - y_2 & y_1 - y_0 & b \\ z_1 - z_2 & z_1 - z_0 & c \end{bmatrix}, \\ \mathbf{A}_3 &= \begin{bmatrix} x_1 - x_2 & x_1 - x_3 & a \\ y_1 - y_2 & y_1 - y_3 & b \\ z_1 - z_2 & z_1 - z_3 & c \end{bmatrix}, & \mathbf{A}_4 &= \begin{bmatrix} x_1 - x_2 & x_1 - x_3 & x_1 - x_0 \\ y_1 - y_2 & y_1 - y_3 & y_1 - y_0 \\ z_1 - z_2 & z_1 - z_3 & z_1 - z_0 \end{bmatrix}. \end{aligned} \quad (4.1)$$

U tom slučaju parametri prodora u i v se mogu izračunati kao [113]:

$$u = \frac{|\mathbf{A}_1|}{|\mathbf{A}_3|}, \quad v = \frac{|\mathbf{A}_2|}{|\mathbf{A}_3|}. \quad (4.2)$$

Tačka prodora zadatog vektora posmatranja \mathbf{v} (koji prolazi kroz tačku P_0) kroz zadati trougao

postoji ukoliko su ispunjeni sledeći uslovi:

$$u, v, u + v \in (0, 1). \quad (4.3)$$

Ukoliko prodor postoji parametar t možemo izračunati kao:

$$t = \frac{|\mathbf{A}_4|}{|\mathbf{A}_3|}, \quad (4.4)$$

dok vrednosti koordinata tačke prodora $\mathbf{P}_p(x_p, y_p, z_p)$ vektora kroz trougao možemo izraziti kao:

$$x_p = x_0 + at, \quad y_p = y_0 + bt, \quad z_p = z_0 + ct. \quad (4.5)$$

ULAZ: **mesh** – struktura koja sadrži temena trouglova; mesh.X, mesh.Y i mesh.Z predstavljaju matrice formata 3xn koje u svakoj koloni sadrže odgovarajuće koordinate tri temena posmatranog trougla
vec=[a b c] – vektor prodora
xs – raster duž x ose
ys – raster duž y ose

```

m=1;
zs=max(mesh.Z);
for i = 1:length(ys)
    for j = 1:length(xs)
        tacka=[xs(j) ys(i) zs];   ▷ tačka definisane strukture kroz koju prolazi vektor prodora
        for k = 1:length(mesh.X)
            trougao=[mesh.X(:,k); mesh.Y(:,k); mesh.Z(:,k)];             ▷ izdvojeni trougao
            generisati matrice A1, A2 i A3 prema jednačinama (4.1)
            generisati parametre u i v prema jednačinama (4.2)
            if u<1 & u>0 & v<1 & v>0 & u+v<1 & u+v>0
                generisati matricu A4 prema (4.1)
                izračunati parametar t prema (4.4)
                izračunati koordinate tačke prodora xp, yp i zp prema (4.5)
                cloud(m,:)= [xp yp zp];   ▷ tačka prodora postaje tačka strukturiranog oblaka
                m=m+1;
            end if
        end for
    end for
end for
end for

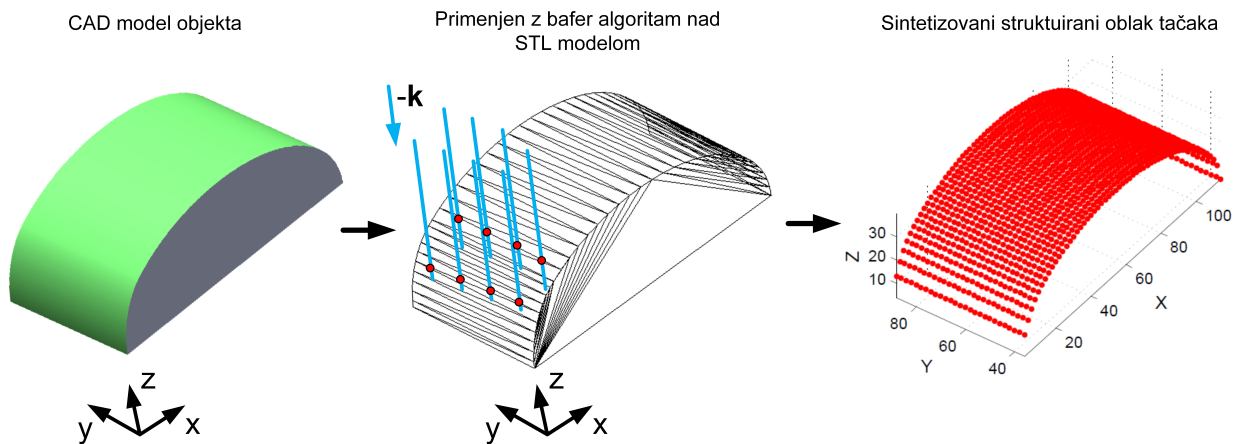
```

IZLAZ: **cloud** = [xp yp zp] – strukturirani oblak tačaka

Slika 4.3: Pseudokod algoritma strukturiranja oblaka tačaka korišćenjem z bafer metodologije.

Kao što se može zaključiti, upotrebom obrazaca (4.1) - (4.5) lako je izračunati vrednosti koordinata tačke prodora bilo kog vektora kroz bilo koji trougao ukoliko taj prodor zapravo postoji. Na osnovu navedene matematičke postavke oformljena je metoda strukturiranja nestruktuiranih oblaka tačaka. Osnovni uslov za primenu ovog algoritma je polazna struktura oblaka u formi 3D trougaone mreže. Mnogi moderni 3D skeneri na svom izlazu pružaju mogućnost direktnog generisanja ovakve forme oblaka, ali u situacijama gde to nije slučaj potrebno je sirovi oblak procesirati nekom od tehnika kreiranja 3D mreža kao što su npr. one definisane u [110, 111, 112]. Ovakva forma zapisa upravo obezbeđuje informacije o koordinatama temena svih trouglova kojima su opisane površine skeniranog objekta. Na početku algoritma potrebno je zadati određenu pravilnu strukturu tačaka (mrežu) sa uniformnim ili neuniformnim rasterom duž dve ose osnovnog koordinatnog sistema ulazne 3D mreže, dok je treća koordinata proizvoljno odabrane fiksne vrednosti i ista je za sve tačke. Takođe, potrebno je definisati i željeni pravac posmatranja, odnosno pravac vektora za koji se traže tačke prodora trouglova mreže, odnosno u ovom slučaju tačke na aproksimiranoj površini skeniranog objekta. Na

osnovu definisane strukture tačaka i zadatog pravca vektora posmatranja iterativnim postupkom u svakoj zadatoj tački rastera proverava se postojanje prodora kroz svaki trougao mreže. Ukoliko prodor postoji, izračunava se tačka prodora koja ujedno postaje i tačka u izlaznom strukturiranom oblaku. Algoritam se završava nakon procesiranja svih trouglova u svakoj tački zadate mreže. Intuitivno je da se za vektor posmatranja bira jedinični vektor negativnog pravca z ose koordinatnog sistema 3D mreže ($-\mathbf{k}$), dok se struktura rastera tačaka zadaje duž x i y osa. U ovakvoj formi vrši se pojednostavljenje metode i izbegava se računanje x i y koordinata tačaka prodora zato što su one prosto već definisane polaznom strukturom. Grafička interpretacija ovako postavljene metode je data na slici 4.2. Sa druge strane, na slici 4.3 prikazan je i pseudokod opisanog algoritma. Veoma je važno napomenuti da ovaj pseudo kod predstavlja algoritam strukturiranja kod koga se izlaznom oblaku pridružuju sve detektovane tačke prodora kroz sve trouglove mreže. Naravno, u zavisnosti od potreba mogu se javiti i varijacije algoritma. Na primer ukoliko 3D mreža trouglova definiše neki objekat zatvorene forme, a vektor posmatranja je postavljen na pravac paralelan z osi, moguće je strukturirati samo površine koje su najviše na objektu. U ovom slučaju u svakoj tački rastera od svih detektovanih tačaka prodora kroz razne trouglove koji se nalaze na tom pravcu treba zadržati samo onu sa najvećom z koordinatom. Opisani postupak strukturiranja baziran na z bafer algoritmu korišćen je u nekim od narednih poglavlja u ovoj disertaciji.



Slika 4.4: Grafički prikaz sintetizovanja oblaka tačaka korišćenjem STL modela objekta i z bafer metodologije

Pored navedenog, potrebno je dodati da je osim procesiranja 3D mreža, predstavljenim algoritmom moguće vršiti obradu i STL formata CAD objekata. Na ovaj način moguće je, za potrebe različitih simulacija, isprojektovati STL model željenog objekta i izvršiti sintetizovanje njegovog strukturiranog oblaka tačaka. Naravno, kako bi se sintetizovani oblak približio realnom, nakon primene algoritma strukturiranja potrebno je dodati određeni nivo veštačkog šuma koordinatama tačaka oblaka. Takođe, ukoliko se formira i primeni određeni npr. haotični (nasumični) raster tačaka pri strukturiranju moguće je simulirati i nestruktuirani oblak tačaka. Postupak sintetizovanja oblaka tačaka na osnovu STL formata objekta je grafički prikazan na slici 4.4.

5. Redukcija broja tačaka u okviru 3D oblaka

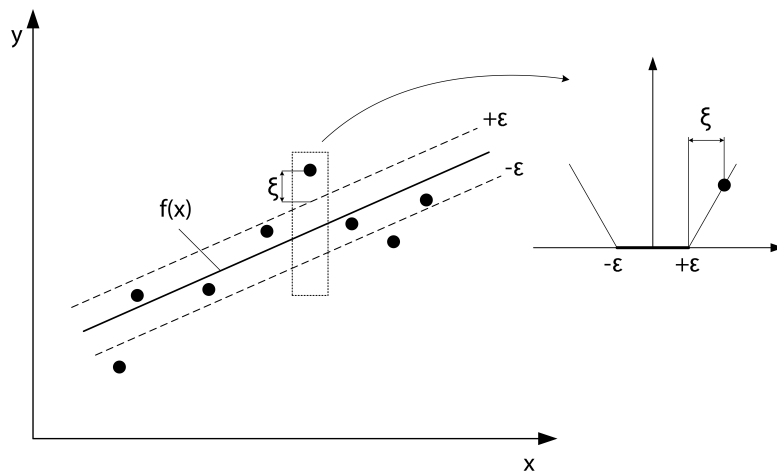
U poglavlju 3.1 bilo je reči o potrebi redukcije broja tačaka inicijalnih (sirovih) oblaka kao i o postojećim relevantnim tehnikama iz ove oblasti. U skladu sa tim u ovoj disertaciji biće predstavljena jedna posve drugačija metoda redukcije strukturiranih oblaka tačaka. Ova metoda se zasniva na tzv. regresiji nosećim vektorima (SVR) [114] pojedinačnih skeniranih kontura oblaka. SVR je tehnika proistekla iz šireg koncepta tehnike mašinskog učenja koju nazivamo mašine sa nosećim vektorima (SVM) [115, 116]. Tvorac ovog koncepta je Vladimir Vapnik koji ovaj pojam uvodi 90-tih godina prošlog veka. Čvrstu teorijsku osnovu za mašine sa nosećim vektorima on pronalazi u VC teoriji statističkog učenja (engl. *Vapnik–Chervonenkis theory*) čija je suština zasnovana na tzv. VC dimenziji kojom se opisuje kompleksnost nekog sistema. U nastavku biće predstavljen koncept regresije nosećim vektorima, a zatim i postavke metode redukcije oblaka kao i rezultati njene eksperimentalne verifikacije i realne primene.

5.1. Regresija nosećim vektorima

Mašine sa nosećim vektorima spadaju u tehnike mašinskog učenja sa nadzorom koje su prvobitno razvijene za potrebe klasifikacije. Međutim, pokazano je da su SVM pogodne i za procese regresije [114]. Ovakva formulacija mašina sa nosećim vektorima za potrebe regresije dovela je do pojave pomenute metode koju nazivamo regresijom nosećim vektorima (SVR). Kako bi se oformio SVR proces bilo je potrebno u matematički model uključiti i tzv. funkciju gubitka (engl. *loss function*) [117]. U upotrebu je uvedeno nekoliko funkcija gubitaka od kojih se do sada izdvojila tzv. ε -neosetljiva (engl. *ε -insensitive*). U ovom radu se koristi regresija nosećim vektorima zasnovana upravo na ovoj funkciji gubitaka, tako da je u narednim poglavljima korišćen pojam ε -neosetljive regresije nosećim vektorima (ε -SVR). Izlaganje u okviru ovog odeljka bazirano je pre svega na [114] i [115].

Osnovni cilj ε -SVR je da se na osnovu skupa obučavajućih parova $\{(x_1, y_1), \dots, (x_l, y_l)\} \subset S \times \mathbb{R}$ (S je prostor ulaznih parametara x_i) pronade funkcija $f(x)$ takva da je od svih $y_i, i = 1, \dots, l$ udaljena maksimalno za vrednost ε , ali koja je ujedno i ravna što je više moguće [114]. Ovo znači da se vrednost greške može zanemariti ukoliko je odstupanje funkcije $f(x)$ od svakog y_i manje od unapred definisane vrednosti za ε . Međutim odstupanja funkcije koja su veća od ε nisu dozvoljena. Takođe, može se primetiti da u zavisnosti od dimenzije prostora S nezavisni parametri x_i mogu biti i vektori (npr. kada je $S = \mathbb{R}^n$). Posmatrajmo prvo linearne funkcije oblika:

$$f(x) = \langle w, x_i \rangle + b, \quad w \in S, b \in \mathbb{R}. \quad (5.1)$$



Slika 5.1: Grafički prikaz modela ε -SVR [114].

U ovom slučaju funkcija je ravna ukoliko parametar w ima malu vrednost. Da bi se osigurala što veća ravnost funkcije $f(x)$, potrebno je minimizirati normu $1/2 \|w\|^2$ tako da su sva odstupanja manja ili jednaka definisanom odstupanju ε . Ovaj zaključak se može definisati kao konveksni optimizacioni problem oblika:

$$\begin{aligned} &\text{minimizirati} && \frac{1}{2} \|w\|^2 \\ &\text{prema} && |y_i - \langle w, x_i \rangle - b| \leq \varepsilon. \end{aligned} \quad (5.2)$$

Međutim, u radu sa realnim sistemima često se ne može naći ravna funkcija koja bi od svih y_i bila udaljena najviše ε , pa se prethodni optimizacioni problem prema [114] može preformulisati kao:

$$\begin{aligned} &\text{minimizirati} && \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ &\text{prema} && \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (5.3)$$

gde su ξ_i i ξ_i^* novouvedene promenljive koje omogućavaju pokretljivost granica pri rešavanju postavljenog optimizacionog problema. Ovakvom konstrukcijom se obezbeđuje mogućnost da se u realnim uslovima, a u skladu sa određenim dozvoljenim odstupanjima, zasigurno nađe tražena funkcija. Dozvoljena odstupanja se kontrolišu parametrom $C > 0$ koji predstavlja balans (engl. *trade-off*) između ravnosti funkcije i veličine odstupanja od uslova minimizacije. Upravo ovakva postavka problema odgovara pomenutoj ε -neosetljivoj funkciji gubitaka koja se može definisati kao:

$$|\xi|_\varepsilon := \begin{cases} 0 & \text{ako je } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{ako je } |\xi| > \varepsilon. \end{cases} \quad (5.4)$$

Potrebno je istaći da, ukoliko bismo grafički posmatrali problem, uočili bismo da je potrebno naći funkciju $f(x)$ za koju će se u ε okolini naći najveći broj obučavajućih parova, a samo mali dozvoljeni broj njih naći van te oblasti. Ovo je ilustrovano na slici 5.1. Treba naglasiti da se u trodimenzionom prostoru ε okolina funkcije može zamisliti kao cev poluprečnika ε (engl. *ε -insensitive tube*).

Sa druge strane, formulaciju (5.3) možemo definisati u dualnoj formi:

$$\begin{aligned} &\text{maksimizirati} && \begin{cases} -\frac{1}{2} \sum_{i,j=1}^l (\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*) \langle x_i, x_j \rangle \\ -\varepsilon \sum_{i=1}^l (\lambda_i + \lambda_i^*) + \sum_{i=1}^l y_i (\lambda_i - \lambda_i^*) \end{cases} \\ &\text{prema} && \sum_{i=1}^l (\lambda_i - \lambda_i^*) = 0 \quad \text{i} \quad \lambda_i, \lambda_i^* \in [0, C], \end{aligned} \quad (5.5)$$

gde su parametri λ_i i λ_i^* tzv. Lagranžovi množioci.

Takođe, prema [114] može se definisati w kao kombinacija parametara učenja x_i , $i = 1, \dots, l$:

$$w = \sum_{i=1}^l (\lambda_i - \lambda_i^*) x_i, \quad (5.6)$$

tako da se tražena funkcija može zapisati u obliku:

$$f(x) = \sum_{i=1}^l (\lambda_i - \lambda_i^*) \langle x_i, x \rangle + b. \quad (5.7)$$

Kombinacija jednačina (5.6) i (5.7) naziva se ekspanzijom nosećih vektora (engl. *support vector expansion*) i njena kompleksnost zavisi direktno od dimenzije prostora S .

Na osnovu izloženog može se uočiti da su Lagranžovi množiocci jednaki nuli za svaki vektor koji je od regresione funkcije udaljen manje od ε , dok su različiti od nule u slučaju kada je vektor izvan ε prostora oko funkcije. Vektori za koje su Lagranžovi množiocci različiti od nule, odnosno koji se nalaze izvan ε okoline regresione funkcije nazivaju se noseći vektori na osnovu kojih je ceo koncept i dobio ime.

Nakon definisanja pojma nosećih vektora za linearni slučaj (regresiona funkcija je bila linearna), potrebno je definisati koncept za nelinearne funkcije koje se češće susreću u praksi. U primeni SVR algoritma za nelinearne slučajeve zastupljena je jedna osnovna ideja: određenom funkcijom prelaza ϕ prevesti ulazne parametre x_i u prostor više dimenzije gde bi slučaj bio linearan i gde bi regresija nosećim vektorima bila linearna. Međutim, u [115] je pokazano da takav prostor više dimenzije nije neophodno definisati u eksplicitnom obliku već je dovoljno uvesti tzv. kernel (engl. *kernel*), odnosno funkciju jezgra $K(x_i, x_j)$ koja implicitno definiše skalarni proizvod u prostoru više dimenzije. Nakon uvođenja ovakve ideje optimizacioni problem u dualnoj formi (5.5) se može definisati kao:

$$\begin{aligned} \text{maksimizirati} \quad & \begin{cases} -\frac{1}{2} \sum_{i,j=1}^l (\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*) K(x_i, x_j) \\ -\varepsilon \sum_{i=1}^l (\lambda_i + \lambda_i^*) + \sum_{i=1}^l y_i (\lambda_i - \lambda_i^*) \end{cases} \\ \text{prema} \quad & \sum_{i=1}^l (\lambda_i - \lambda_i^*) = 0 \quad \text{i} \quad \lambda_i, \lambda_i^* \in [0, C]. \end{aligned} \quad (5.8)$$

Regresiona funkcija je u ovom slučaju definisana kao:

$$f(x) = \sum_{i=1}^l (\lambda_i - \lambda_i^*) K(x_i, x) + b, \quad (5.9)$$

dok izraz (5.6) postaje:

$$w = \sum_{i=1}^l (\lambda_i - \lambda_i^*) \phi(x_i), \quad (5.10)$$

Ukoliko je prostor u kome se vrši linearna regresija dimenzije više od tri onda govorimo o hiperprostoru, dok je u tom slučaju regresiona funkcija opisana jednačinom (5.9) hiperavan. Kernel može biti bilo koja simetrična i pozitivno definitna funkcija koja zadovoljava uslove Mercerove teoreme (engl. *Mercer's theorem*) [114]. Drugim rečima funkcija oblika $K(u, v)$ zadovoljava postavljene uslove ukoliko izraz:

$$\iint K(u, v) g(u) g(v) du dv < 0, \quad (5.11)$$

važi za svako $g \neq 0$ za koje je $\iint g^2(u) du < \infty$ [115].

Radi boljeg razumevanja funkcije i forme kernela daje se njegova opšta matematička formulacija kao:

$$K(x, x') := \langle \phi(x), \phi(x') \rangle, \quad (5.12)$$

gde sa x' označavamo obučavajuće parametre u prostoru više dimenzije.

U upotrebi SVR se mogu susresti veoma različiti kerneli: polinomni kerneli, Gausov kernel, sig-

moidna funkcija, splajn (engl. *spline*) kernel, B-splajn (engl. *B-spline*) kernel itd. Potrebno je naglasiti da se linearnim kombinacijama navedenih funkcija mogu dobiti neki novi kerneli.

Ne može se sa sigurnošću tvrditi korišćenje kog kernela je pogodno u određenim slučajevima regresije, ali postoje izvesne preporuke u tom pravcu. Ukoliko imamo određena očekivanja i ocenu kakvu bi regresionu funkciju mogli da dobijemo, možemo izabrati krenel koji bi joj po obliku bio blizak. U ovom radu, a u skladu sa postavljenom problematikom biće korišćeni splajn i B-splajn kernel. Splajn reda $k \geq 1$ koji na intervalu T_s poseduje N čvorova može se prema [117] predstaviti kao:

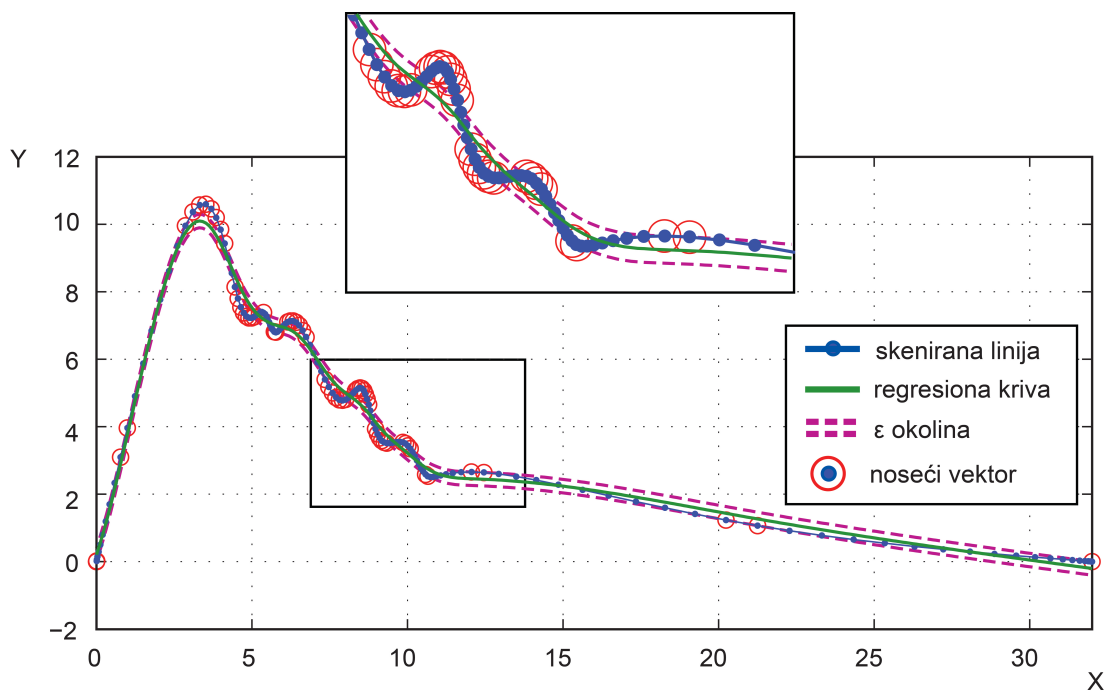
$$K(x, x') = \sum_{r=0}^k x^r x'^r + \sum_{s=1}^N (x - T_s)_+^k (x' - T_s)_+^k, \quad (5.13)$$

gde je $(\cdot)_+^k = \max\{(\cdot)^k, 0\}$. Sa druge strane B-splajn kernel reda $2n + 1$ definisan na intervalu $[-1, 1]$ se može definisati kao:

$$K(x, x') = B_{2n+1}(x - x'). \quad (5.14)$$

5.2. Definicija metode i algoritma

Metoda redukcije broja tačaka skeniranih kontura koja je kreirana u ovoj disertaciji temelji se na upotrebi ε -SVR. Ova vrsta regresije se primenjuje nad skupovima podataka (tačkama) u 2D prostoru, odnosno u prostoru dimenzije \mathbb{R}^2 . Naime, nad svakom skeniranom konturom strukturiranog oblaka sprovodi se ε -SVR, ali potrebni rezultati se ne ogledaju u samoj regresionoj funkciji (krivoj) generisanoj tom prilikom. Osnov upotrebe regresije se sastoji u detektovanju tačaka (nosećih vektora) koji se nalaze van unapred zadate ε oblasti oko te funkcije (krive). U ovom slučaju koristi se svojstvo ravnosti (engl. *flatness property*) ε -SVR. U tom smislu, regresiona kriva biće što je moguće ravnija i „dobro“ će aproksimirati skupove tačaka u ravnim oblastima ili oblastima konstantne promene geometrije - u njima neće izdvajati veliki broj nosećih vektora. Sa druge strane, u oblastima gde skenirane tačke reprezentuju zone visoke promene geometrije (zakrivljenosti) regresiona kriva će „teže“ moći da izvrši aproksimaciju i veći broj tačaka u tim zonama biće identifikovan kao noseći vektori. Grafički prikaz rezultata upotrebe ε -SVR nad jednom proizvoljnom konturom dat je na slici 5.2.



Slika 5.2: Ilustracija primene ε -SVR nad skeniranom konturom.

Detektovani noseći vektori se mogu koristiti kao indikatori značajnosti tačaka u skeniranim konturama, što znači da će tačke koje ne budu noseći vektori biti odstranjene iz skenirane konture. Zadržavanjem samo identifikovanih nosećih vektora, odnosno većeg broja tačaka u visoko zakrivljenim zonama i malog broja tačaka u ravnim, dobijaju se redukovane konture koje poseduju visok stepen održanja informacije o geometriji skeniranog objekta. Sam kvalitet održanja informacije nakon redukcije biće diskutovan u poglavlju posvećenom verifikaciji metode.

Broj izdvojenih nosećih vektora se može kontrolisati pažljivim zadavanjem vrednosti osnovnih parametara ε -SVR - ε i C . Što je veća zadata vrednost parametra ε (margine dozvoljene greške) to znači da će ε oblast oko regresione krive biti šira i broj identifikovanih nosećih vektora biti manji. Ovo u prenesenom smislu znači da se povećanjem pomenute granice regresionoj funkciji obezbeđuje veća sloboda pri aproksimaciji tretiranog skupa tačaka, odnosno dozvoljava se veća greška aproksimacije. Sa druge strane, dozvoljena greška regresije se kontroliše parametrom C čime se obezbeđuje potreban balans. Što je veća vrednost parametra C broj izdvojenih nosećih vektora biće manji i regresiona funkcija će se „više potruditi“ da kvalitetno aproksimira zadatu geometriju. U suprotnom javlja se veći broj nosećih vektora. Zbog ove osobine parametar C se često naziva cenom za grešku (engl. *error cost*). Pored pomenuta dva parametra, na rezultate ε -SVR utiče i izbor kernela. U skladu sa oblikom regresione krive koja se u opštem slučaju očekuje za ovu metodu, odabrana su i testirana dva kernela - splajn i B-splajn kernel. U nastavku će kroz verifikaciju postavljene metode redukcije biti predstavljeni uticaji vrednosti parametara ε i C kao i različitosti pri upotrebi oba izabrana kernela.

5.3. Eksperimentalna verifikacija metode

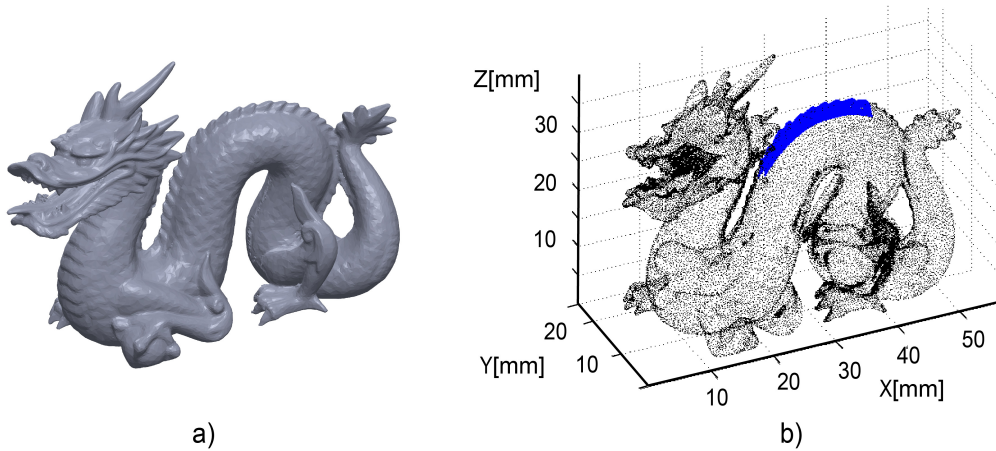
Za potrebu verifikacije predstavljene metode redukcije broja tačaka biće upotrebljena tri karakteristična realna oblaka tačaka. Prva dva poseduju kompleksnu geometriju, ali bez oštih ivica dok treći oblak tačaka reprezentuje manje složenu geometriju, ali koja sadrži upečatljive oštre prelaze u skeniranim konturama. Potrebno je naglasiti da je za potrebe implementacije metode korišćen softverski paket Matlab [118], dok je za upotrebu SVR u okviru ovog programskog paketa korišćen poseban SVR toolbox [119]. Procesiranje implementiranih metoda vršeno je na 1,7 GHz Intel Core i3 računarskoj platformi. Rezultati će biti prezentovani i diskutovani kroz tri odvojene studije slučaja.

5.3.1. Studija slučaja 1: Stanford zmaj

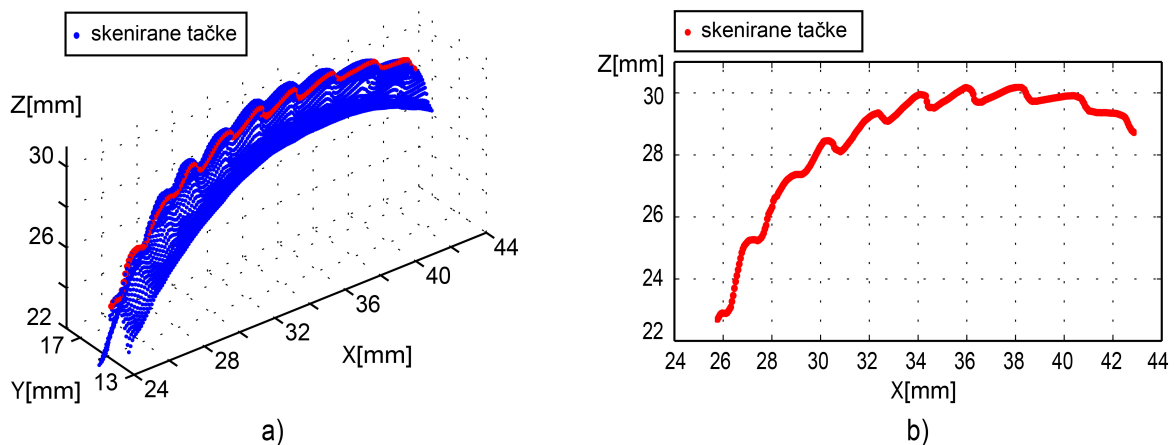
Prvi oblak tačaka koji je iskorišćen za verifikaciju predstavljene metode reprezentuje tzv. Stanford zmaja (slika 5.3). Ovaj oblak je prvi put upotrebljen za potrebe istraživanja predstavljenog u [120] i danas se smatra jednim od standarda pri verifikaciji različitih algoritama u kompjuterskoj grafici. Geometrija skenirane statue zmaja se, po svojoj prirodi, sastoji od velikog broja zakrivljenih oblasti sa čestom (ali kontinualnom) promenom geometrije. Oblak sadrži preko 2,7 miliona tačaka i dobijen je registracijom oko 70 nezavisnih skenova. Za potrebe ovog rada ovaj nestruktuirani oblak (slika 5.3(a)), u formi trougaone mreže (PLY format), preuzet je sa 3D repozitorijuma Stanford univerziteta [121] i potom je konvertovan u binarni STL format. Ovim postupkom se došlo do standardne forme 3D trougaone mreže kod koje svaka tačka, generisana pri skeniranju, predstavlja teme nekog od trouglova u mreži. Zbog potrebe za generisanjem struktuiranog oblaka, potrebno je bilo izvršiti struktuiranje upotrebom metode predstavljene u poglavlju 4. Za vektor prodora izabran je vektor $-\mathbf{k}(0, 0, -1)$. Treba naglasiti da je binarni STL format oblaka učitao u Matlab upotrebom eksterno preuzete funkcije iz [122]. Takođe, izvršeno je struktuiranje samo dela leđa zmaja koji predstavlja oblast sa najizraženijom promenom geometrije (slika 5.3(b)).

Zbog potrebe jasne percepcije mogućnosti ε -SVR, u ovom primeru izabrano je da se generišu dva struktuirana oblaka različite gustine. Raster prvog (gušćeg) struktuiranog oblaka (slika 5.4(a)) je podešen na 0,05 mm duž x , a na 0,1 mm duž y ose koordinatnog sistema polazne trougaone mreže. Na ovaj način je dobijen oblak sa 7.935 tačaka. Iako ovako gust raster duž x ose nema realnu

primenu, ovaj oblak je kreiran da bi se određeni skoro vertikalni segmenti kontura reprezentovali sa što više tačaka radi boljeg predstavljanja performansi postavljene metode redukcije. Sa druge strane, zbog jednostavnosti reprezentacije ε -SVR parametara na rezultate, biće korišćena samo jedna njegova izdvojena kontura prikazana na slici 5.4(b).



Slika 5.3: Stanford zmaj: a) 3D mreža; b) struktuirana oblast od interesa.

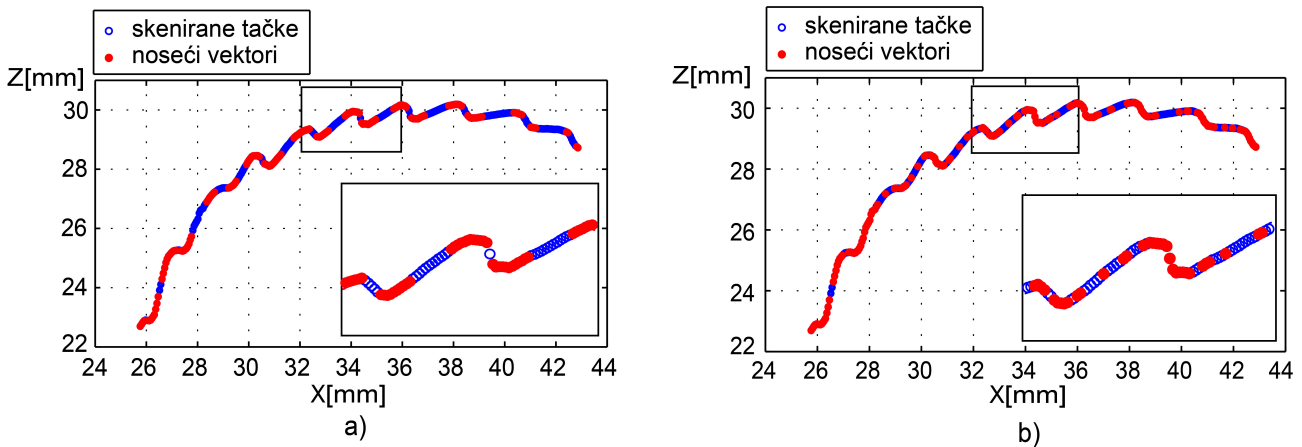


Slika 5.4: Struktuirani oblak rastera 0,05 mm duž x ose : a) struktuirani oblak; b) izdvojena kontura oblaka.

Drugi struktuirani oblak je generisan upotrebom uniformnog rastera od 0,1 mm u oba pravca. Ovaj oblak sadrži 3.979 tačaka odnosno 23 skenirane konture od po 173 tačke. Kreiran je da bi se njegovim procesiranjem predstavilo interesantno poređenje vremena procesiranja u slučaju manjeg broja tačaka u konturama. Ovde se pre svega misli na vreme regresije nosećim vektorima u slučaju različite gustine tačaka na konturi iste geometrije. Diskusija i upoređivanje rezultata vremena izvršavanja algoritma biće predstavljeni nakon ispitivanja performansi metode.

Prvo će na primeru pomenute izdvojene konture sa slike 5.4(b) (rastera 0,05 mm i broja tačaka 345) biti predstavljen uticaj vrste kernela (splajn i B-splajn) na broj izdvojenih nosećih vektora, odnosno tačaka koje se zadržavaju u konturi nakon redukcije. Tokom ove analize u oba slučaja vrednost cene za grešku odnosno parametra C je postavljena na 10. Sa druge strane, a zbog prirode samih izabranih kernela, margina greške regresije (parametar ε) je u slučaju B-splajn kernela postavljena na vrednost 0,15, a u slučaju splajn kernela na znatno manjih 0,01. Ovim vrednostima obezbeđen je približno jednak procenat redukcije kako bi se stvorili uslovi za adekvatnije poređenje ova dva slučaja. Procenat redukcije je definisan kao procentualno izražen broj zadržanih tačaka u odnosu na ukupan broj tačaka konture. Rezultati ε -SVR nad izdvojenom skeniranom konturom, a za predstavljene vrednosti ε i C ,

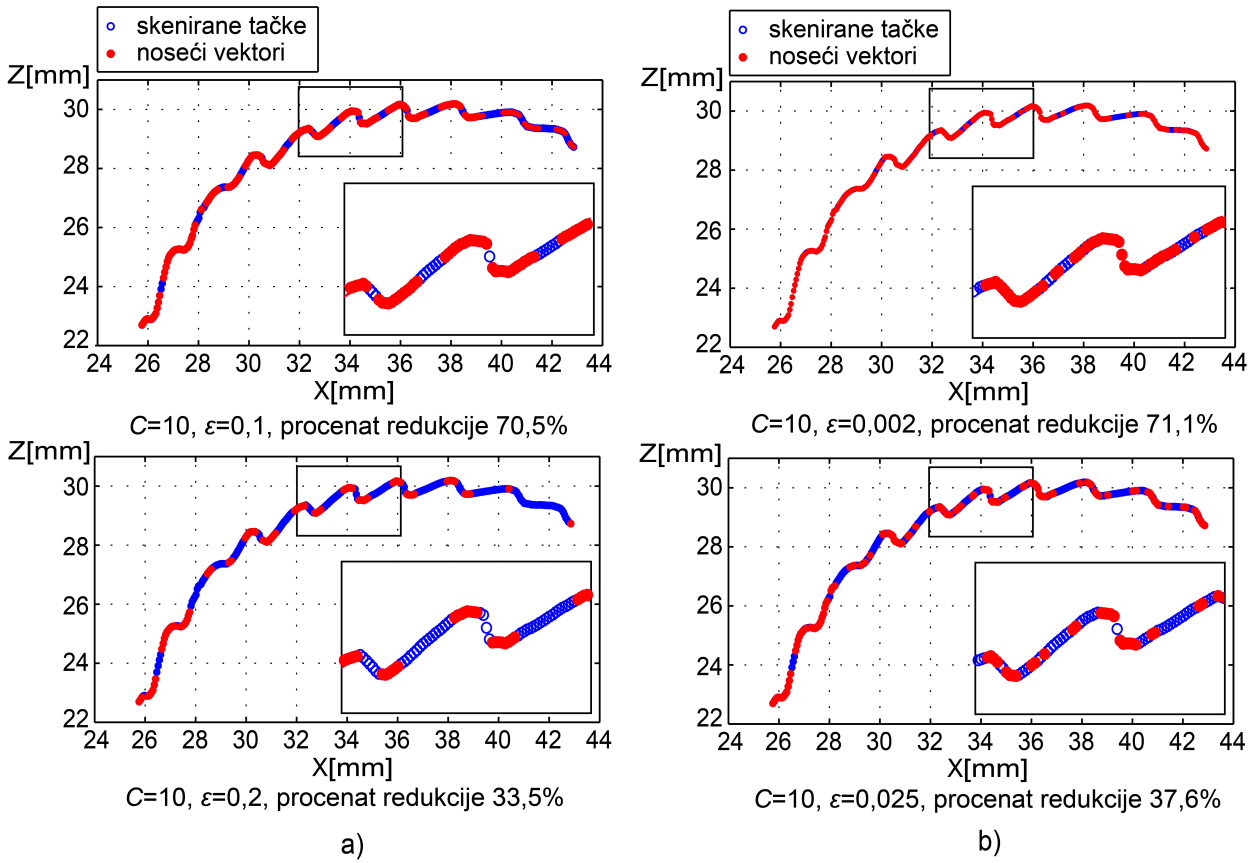
u slučaju primene B-splajn kernela obezbeđuju procenat redukcije od 52,8%. U slučaju splajn kernela ovaj procenat je 49,0%. Treba istaći i da je pri upotrebi B-splajn kernela potrebno definisati i stepen polinoma (koji je za sve primere u ovom radu postavljen na 3). Takođe, pošto je B-splajn definisan samo na intervalu $[-1,1]$, pre njegove upotrebe bitno je izvršiti normalizaciju svih vrednosti koordinata tačaka duž apscise u ovaj definsani interval.



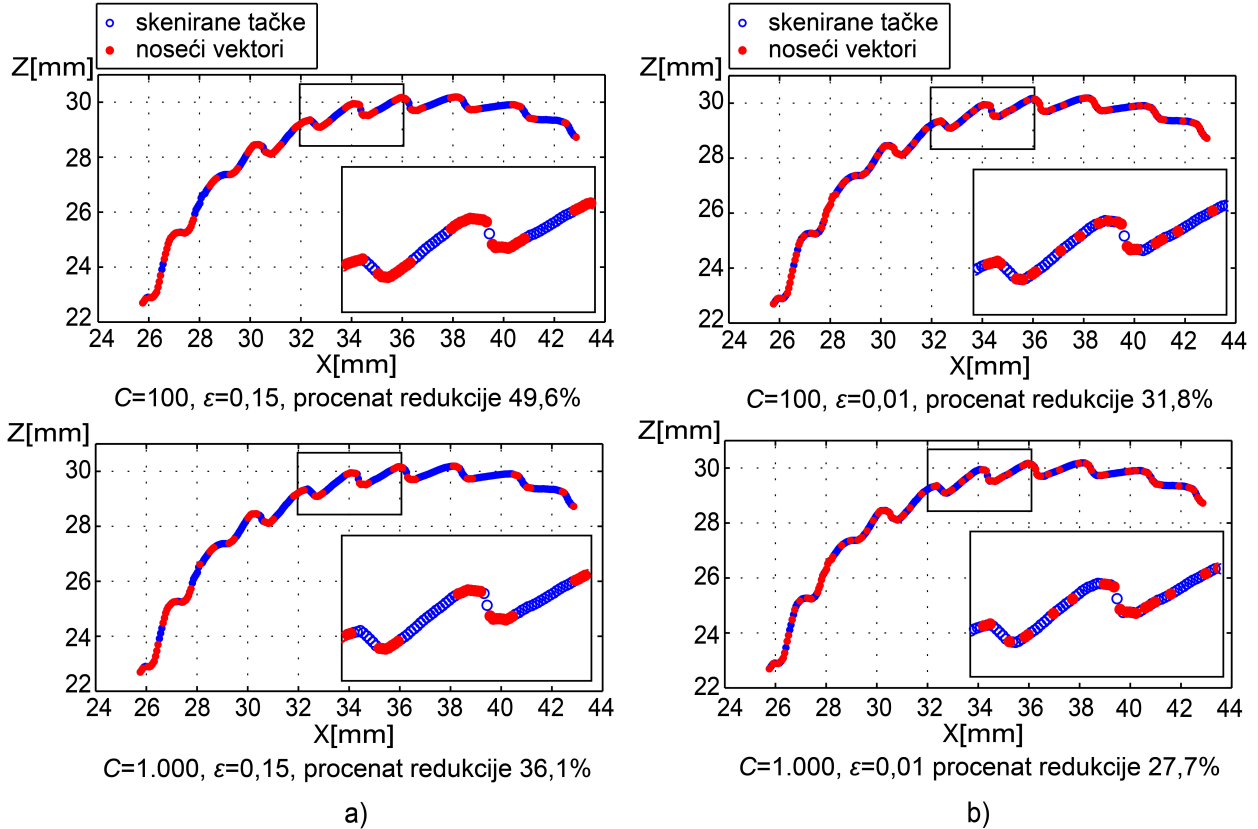
Slika 5.5: Efekti izbora kernela na redukciju: a) B-splajn kernel: $\varepsilon = 0,15$, $C = 10$, procenat redukcije 52,8%; b) splajn kernel: $\varepsilon = 0,01$, $C = 10$, procenat redukcije 49,0%.

Grafički prikaz uticaja vrste kernela na rezultate redukcije dat je na slici 5.5. Analizom ovog prikaza može se uočiti da je u oba slučaja upotreba ε -SVR identifikovala značajne tačke (noseće vektore) u zonama visoke zakrivljenosti konture, dok je u ravnijim oblastima primećeno značajno odsustvo istih. Ovime je potvrđena i polazna pretpostavka i ispunjen je zahtev za redukcijom takvom da u visokoj meri sačuva informaciju o geometriji skenirane konture. O samom kvalitetu regresije, odnosno o stepenu održane informacije i njenoj oceni biće više reči u narednoj sekciji ovog rada. Međutim, u slučaju upotrebe splajn kernela u pomenutim ravnim segmenata ipak je zadržano i nekoliko nosećih vektora. Ovo ponašanje može se opravdati osobinom samog kernela, ali i značajnom činjenicom da je pri korišćenju splajn kernela margina greške regresije bila reda veličine uža nego u slučaju upotrebe B-splajna. Ova osobina, odnosno potreba upotrebe užih margina pri korišćenju splajn kernela, predstavlja glavnu razliku u analiziranom slučaju. Takođe, izdvajanje sporadičnih nosećih vektora na ravnijim segmentima može se smatrati i dobrom osobinom SVR, ali o tome će biti nešto više reči kasnije.

Sa druge strane, uticaji vrednosti parametara ε i C na regresiju se mogu smatrati intuitivnim. Pravi izbor ovih vrednosti umnogome može doprineti kvalitetnoj redukciji i zbog toga je u narednim primerima detaljnije pojašnjen njihov uticaj. Prvo se razmatra uticaj margine greške odnosno parametra ε . Ukoliko ovaj parametar poseduje nisku vrednost, ε -SVR će, u zadatim granicama greške regresije $[-\varepsilon, +\varepsilon]$ i uz npr. visoku cenu za grešku (C), imati problem da konstruiše dobru regresionu krivu kroz zadate tačke konture. Ovo značajno može da utiče na broj identifikovanih nosećih vektora kao i na vreme procesiranja. Zbog navedenog, veoma je bitno pronaći balans između zadate vrednosti za ε i željenog stepena redukcije kontura (količine izdvojenih nosećih vektora odnosno zadržanih tačaka konture). Što je zadata vrednost parametra ε niža, margina greške biće uža. U tom slučaju i broj izdvojenih nosećih vektora (tačaka koje izlaze izvan margina greške) biće veći. Ovo dovodi do smanjenja stepena redukcije, odnosno do povećanja procenta zadržanih tačaka u odnosu na njihov ukupan broj. Naravno, za šire margine, regresija će biti olakšana i broj nosećih vektora biće manji. Pojašnjeni uticaj vrednosti parametra ε na regresiju prikazan je na slici 5.6. Ipak, navedene tvrdnje ne mogu biti striktno definisane bez razmatranja uticaja vrednosti drugog značajnog parametara regresije - cene za grešku C . Takođe, značajnu ulogu u izboru vrednosti za ε ima i tip kernela koji se pri regresiji koristi. Kao što je pokazano, za isti željeni stepen redukcije, u slučaju korišćenja B-splajn kernela potrebno je zadati šire margine greške u odnosu na slučaj upotrebe splajn kernela.



Slika 5.6: Efekti vrednosti parametra ϵ na redukciju: a) B-splajn kernel; b) splajn kernel.



Slika 5.7: Efekti vrednosti parametra C na redukciju: a) B-splajn kernel; b) splajn kernel.

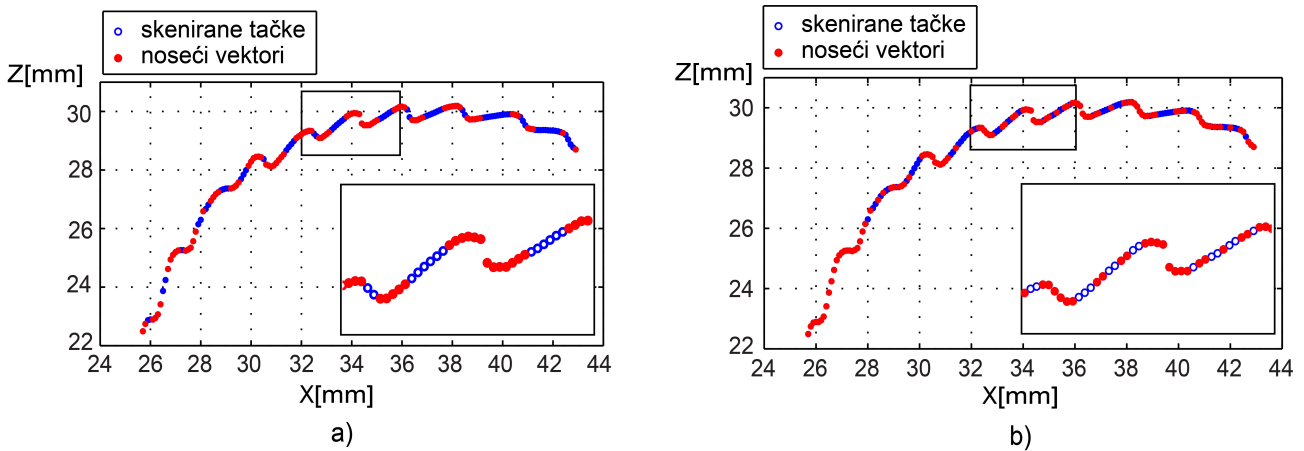
Vrednost parametra C (cene za grešku) takođe ima značajan uticaj na rezultate regresije, odnosno u ovom slučaju redukcije. Ovaj parametar, kao što je ranije pomenuto, predstavlja balans između ravnosti regresione funkcije i devijacije (greške) od ulaznih podataka (tačkaka). Teorijski posmatrano, niska vrednost ovog parametra (a pri određenoj fiksnoj vrednosti margine ε) dovodi do ravnije regresione krive (funkcije), a samim tim i do detektovanja većeg broja nosećih vektora kao posledice dozvole većeg stepena devijacije od ulaznih podataka. Ova osobina uticaja vrednosti cene za grešku na izdvajanje nosećih vektora značajno je izražena u zakrivljenim segmentima skeniranih kontura kao i na mestima oštih prelaza geometrije što predstavlja prednost za ovakvu upotrebu ε -SVR. Sa druge strane, za visoke vrednosti parametra C , broj nosećih vektora biće značajno manji, ali regresiona kriva neće biti dovoljno ravna već će u značajnoj meri aproksimirati tačke konture.

Efekat parametra C na redukciju izdvojene konture zmaja je grafički prikazan na slici 5.7. Za ovu ilustraciju izabrana su po dva primera (za oba kernela) sa značajno različitim vrednostima za C i odgovarajućim fiksnim vrednostima za ε . Posmatrajnem ovih rezultata može se uočiti pomenuti pad broja nosećih vektora za višu vrednost cene za grešku kod korišćenja B-splajna. Međutim, kod splajn kernela je vrednost cene za grešku u gornjem delu slike 5.7(b) deset puta manja ali, zbog značajno uske margine ε , razlika u izdvojenim nosećim vektorima nije toliko izražena kao u pomenutom slučaju kada se koristio B-splajn. Ovo samo dokazuje do sada više puta pomenutu tvrdnju da se uticaj parametra C ne može razmatrati bez njegove korelacije sa marginom greške ε i obrnuto. Iako ne postoji neko striktno pravilo, radi jednostavnosti podešavanja parametara pri postizanju željenog efekta redukcije, poželjno je jedan od ova dva parametra fiksirati i vršiti promenu samo drugog. Npr. u slučajevima kontura koje su reprezentivi geometrija visokog stepena zakrivljenosti pogodno je fiksirati određenu nižu vrednost cene za grešku, a podešavati njenu marginu. Sa druge strane, kod kontura koje sadrže ravne geometrije poželjno je fiksirati marginu greške, a željeni stepen redukcije postizati promenom parametra C . Navedene tvrdnje su i obrazložene određenim rezultatima redukcija navedenim u nastavku ovog poglavlja.

Nakon provere teorijskih osnova SVR regresije u konkretnom slučaju jedne izvojene skenirane konture i razmatranja uticaja vrednosti parametara na produkte regresije, potrebno je sagledati i potrebno vreme procesiranja. Za početni primer može se uzeti procesiranje konture u slučaju predstavljenom na slici 5.5. Uzevši u obzir oba slučaja, odnosno upotrebu oba kernela primećeno je da je, u slučaju ove konture veoma malog rastera po apscisi (0,05 mm), potrebno izrazito puno vremena za procesiranje. U slučaju splajn kernela vreme procesiranja iznosi 55,1 s, dok u slučaju B-splajna čak 81,8 s. Ovo se za konturu sa 345 tačkaka ne može smatrati dovoljnim za neku primenu u sistemima koji zahtevaju rad u realnom vremenu. Međutim, vreme regresije ne zavisi samo od broja tačkaka već i od njihove međusobne udaljenosti što će biti pokazano u nastavku. Ipak, ovako visoko vreme procesiranja može se delimično opravdati činjenicom da algoritam nije implementiran na visokom profesionalnom nivou kao i da je računar na kome su vršena procesiranja skromnih mogućnosti. Takođe, prilikom izvršavanja koda pisanog u Matlabu mnogi drugi procesi vezani za operativni sistem kao i ostale programe mogu biti paralelno u funkciji što takođe utiče na povećanje vremena procesiranja.

Situacija sa vremenom procesiranja može se učiniti i lošijom ukoliko posmatramo kompletan struktuirani oblak sa slike 5.4(a) koji poseduje 23 guste konture slične geometrije. U tom slučaju vreme procesiranja svih kontura oblaka korišćenjem splajn kernela (za iste parametre kao u primeru jedne konture - $\varepsilon = 0,01$, $C = 10$) bi iznosilo i preko 1260 s. Međutim, bitan detalj je da se SVR odlikuje time da, pored smanjenja broja tačkaka, i povećanje njihove međusobne udaljenosti utiče na višestruko smanjenje vremena procesiranja. Imajući ovaj detalj u vidu, na samom početku poglavlja naglašeno je da su kreirana dva struktuirana oblaka od kojih je drugi uniformnog rastera od 0,1 mm. Ovo znači da njegova izdvojena kontura (koja odgovara onoj sa slike 5.4(b)) sadrži skoro duplo manje tačkaka koje su na duplo većem međusobnom rastojanju. Pri procesiranju ovakve konture upotrebom ε -SVR se znatno brže dolazi do odgovarajuće regresione krive koja zadovoljava postavljene uslove (zadate vrednosti parametara). Radi lakšeg razumevanja, u tabeli 5.1 dat je uporedni pregled vremena procesiranja u slučaju obe konture - one rastera 0,05 mm i one rastera 0,1 mm. Treba istaći da, zbog

relevantnosti podataka, vremena prikazana u ovoj tabeli predstavljaju srednje vrednosti pet nezavisnih izvršavanja algoritma. Radi vizuelnog upoređivanja na slici 5.8 dati su i rezultati ε -SVR nad konturom rastera 0,1 mm. Parametri regresije su isti kao pri procesiranju gušće konture prikazane na slici 5.5.



Slika 5.8: ε -SVR izdvojene konture rastera 0,1 mm: a) B-splajn kernel: $\varepsilon = 0,15$, $C = 10$, procenat redukcije 54,9%; b) splajn kernel: $\varepsilon = 0,01$, $C = 10$, procenat redukcije 62,4%.

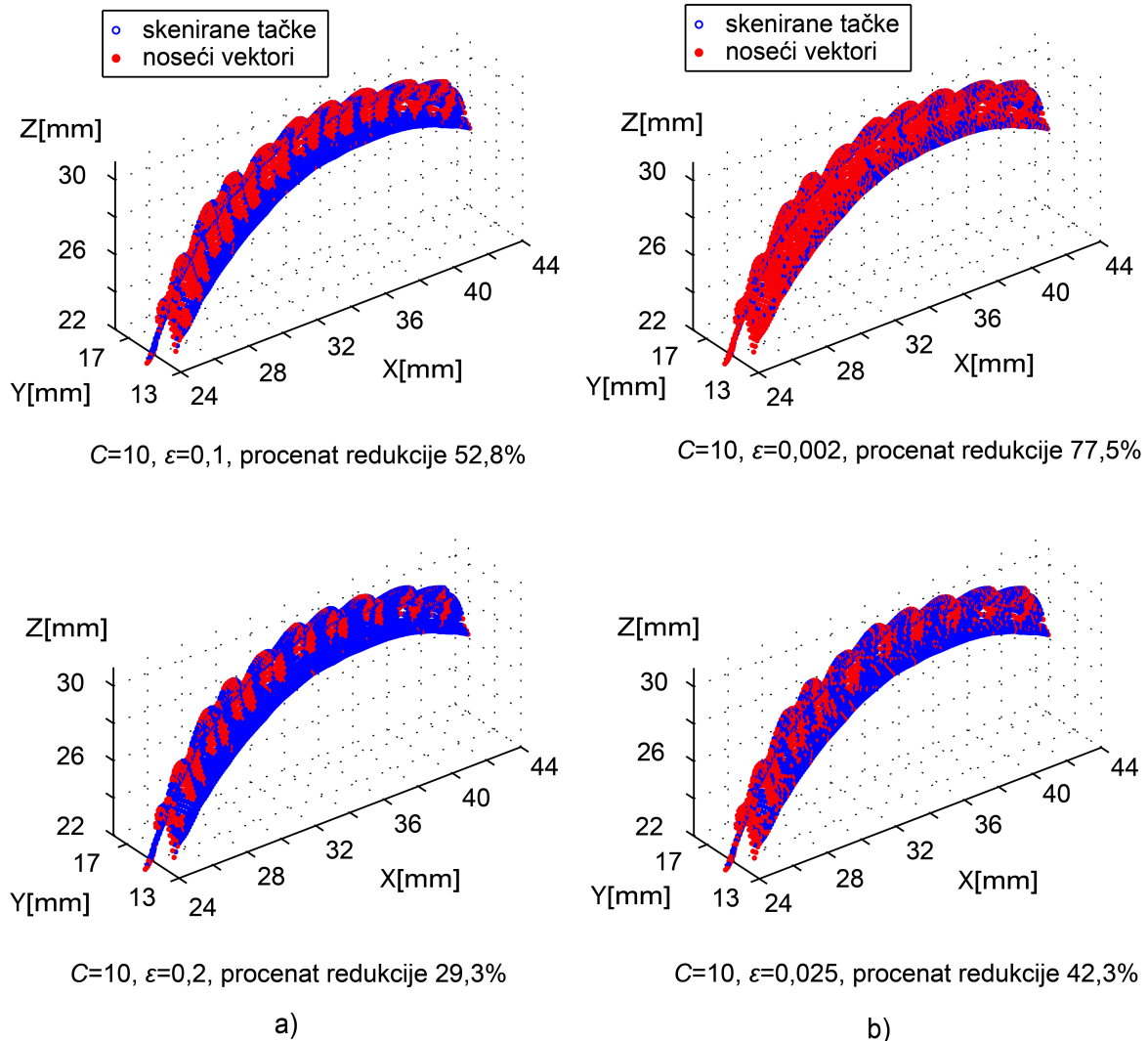
Tabela 5.1: Pregled rezultata procesiranja u slučaju konture zmaja sa različitim brojem tačaka.

Kernel	ε	C	Kontura rastera 0,05 mm (345 tačaka)		Kontura rastera 0,1 mm (173 tačke)	
			Noseći vektori [%]	Vreme [s]	Noseći vektori [%]	Vreme [s]
B-splajn	0.15	10	52,8	81,8	54,9	12,3
splajn	0.01	10	49,0	55,1	62,4	7,1

Posmatrajući prikazane tabelarne i grafičke rezultate, može se zaključiti da pomenuto smanjenje broja tačaka konture za blizu 50% (korišćenje ređe konture) ne smanjuje vreme procesiranja linearno već znatno više. Npr. u slučaju sa slike 5.8 vreme procesiranja splajn kernelom je 7,1 s u odnosu na prethodnih 55,1 s. Takođe, i u slučaju B-splajna vreme izvršavanja algoritma se redukovalo sa 81,8 s na 12,3 s. U ovom primeru, ali i pri generisanju drugih rezultata u ovom radu, primećeno je da je algoritmu redukcije pri korišćenju B-splajna u odnosu na splajn potrebno nešto više vremena za izvršavanje. Ova posledica proizilazi iz toga da je B-splajnu pri njegovom korišćenju za regresiju potrebno zadati stepen polinoma što može dovesti do sporije regresije (u ovom radu je izabrano da se koristi B-splajn reda 3). Takođe, određeno vreme se gubi i pri normalizaciji koordinata po apscisi u opseg $[-1,1]$ kao i ponovnom konvertovanju u originalni raspon nakon izvršene regresije. Vizuelno je primetna i razlika u procentima izdvojenih nosećih vektora pri poređenju ova dva slučaja. Naravno da, ukoliko se razmišlja o kvalitetu informacije dobijene skeniranjem, u slučaju konture sa manje tačaka ova vrednost opada. Međutim, obe vrednosti korišćenih rastera za struktuiranje (0,05 i 0,1 mm) se mogu smatrati veoma niskim i nepovoljnim za realnu primenu čak i kod test objekta veoma sitnih detalja i one su u ovom slučaju generisane samo za potrebe istraživanja. Šta više, na slikama koje reprezentuju rezultate procesiranja tj. na njihovim uvećanim segmentima, može se uvideti da u jednoj oblasti konture (koja je gotovo vertikalno orijentisana) u oba slučaja nedostaju tačke. Ovo indikuje da samo smanjenje rastera i povećanje gustine tačaka pri struktuiranju (ili skeniranju) u nekim situacijama nije od značaja za povećanje informacije o geometriji posmatranog objekta. Zbog svega navedenog se u narednim delovima ovog poglavlja predstavljaju i procesiraju još dva realna oblaka tačaka kod kojih su rasteri struktuiranih oblaka prilagođeniji realnim potrebama i čijim rezultatima redukcije se eventualne dileme oko mogućnosti upotrebe predstavljene metode mogu otkloniti.

Tabela 5.2: Rezultati redukcije kompletnog oblaka leđa zmaja uniformnog rastera 0,1 mm.

Kernel	ϵ	C	Noseći vektori [%]	Vreme procesiranja [s]
B-splajn	0,1	10	52,8	283,6
B-splajn	0,2	10	29,3	303,1
splajn	0,002	10	77,5	165,1
splajn	0,025	10	42,3	167,9



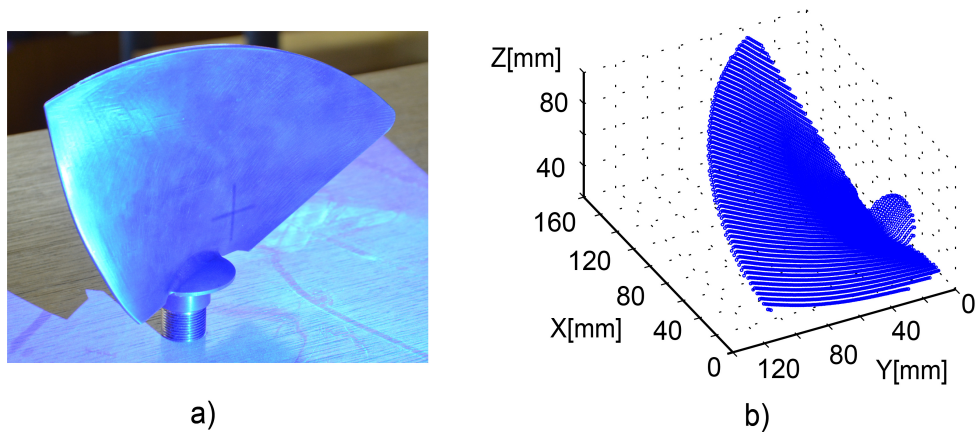
Slika 5.9: Redukcija kompletnog strukturiranog oblaka leđa zmaja uniformnog rastera 0,1 mm: a) B-splajn kernel; b) splajn kernel.

Nakon diskusije na temu vremena procesiranja u tabeli 5.2 su predstavljeni rezultati redukcije svih kontura oblaka uniformnog rastera 0,1 mm. Vrednosti margine greške ϵ su proizvoljno izabrane dok je vrednost parametra C u svim slučajevima jednaka 10. SVR je primenjena nad skeniranim linijama duž x ose. Vremena procesiranja u ovim slučajevima se mogu smatrati prihvatljivim jer bi profesionalna implementacija algoritma svakako doprinela njihovom višestrukom smanjenju. Ipak, ostaje činjenica da je u slučaju B-splajna regresija nešto sporija. Grafički rezultati koji prate tabelu 5.2 su prikazani na slici 5.9. Ovde se može zapaziti prethodna preporuka za izbor parametara. U slučaju ovakve izrazito promenljive geometrije izabrano je da se cena za grešku postavi na fiksnu malu vrednost, a da se željeni rezultati redukcije ostvaruju promenom margine greške. Na ovoj slici je moguće uočiti i željeni efekat redukcije tj. zadržavanje velikog broja tačaka u segmentima visoke varijacije geometrije, a

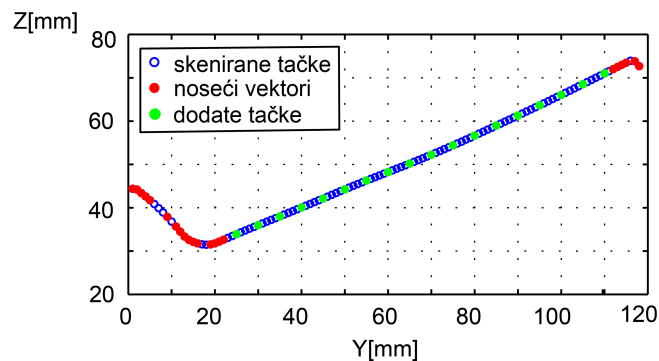
odstranjivanje većeg broja tačaka na ravnijim. Sa druge strane, sam kvalitet održanja informacije o skeniranom obliku nakon redukcije biće proveren i diskutovan u poglavlju 5.4.

5.3.2. Studija slučaja 2: Lopatica turbine

Drugi realni oblak koji se koristi za verifikaciju algoritma redukcije predstavlja jedan karakterističan mašiniski element. Reč je o lopatici turbine (slika 5.10(a)) koja sadrži upečatljivu oblast ravne geometrije, ali i visoko promenljivu geometriju na svojim krajevima. Lopatica je skenirana optičkim skenerom naziva *ATOS Compact Scan 5M* [24]. Deklarisani opseg rezolucije ovog uređaja je 0,017-0,481 mm, dok tačnost skeniranja nije definisana od strane proizvođača. Ovaj skener, sa svojim pripadajućim softverom, kao izlaz iz procesa skeniranja lopatice formirao je nestruktuirani oblak tačaka u formi 3D trougaone mreže. Nakon primene metode struktuiranja iz poglavlja 4, dobijen je struktuirani oblak (slika 5.10(b)) koji sadrži 5.210 tačaka. Treba naglasiti da je globalni koordinatni sistem mreže izabran tako da se dominantna promena geometrije oslikava duž njegovih glavnih osa. Takođe, pri struktuiranju raster je podešen na 3 mm duž x ose i na 1 mm duž y ose.



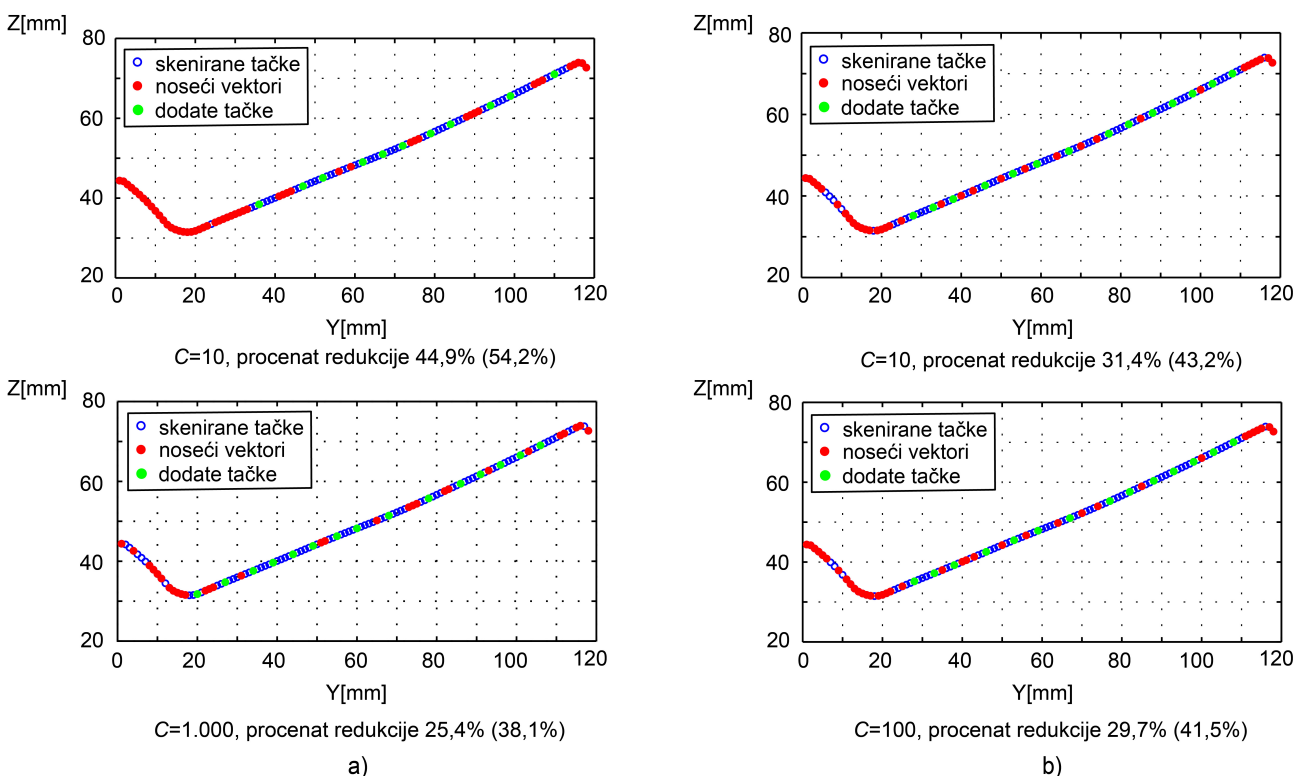
Slika 5.10: Lopatica turbine: a) fotografija realnog objekta; b) struktuirani oblak tačaka.



Slika 5.11: ϵ -SVR izdvojene konture lopatice: $\epsilon = 0,01$, $C = 1.000$, splajn kernel - noseći vektori i dodatne tačke na $d_{max} = 5$ mm.

Za razliku od prethodno procesiranog oblaka tačaka, ovde oblak lopatice nije iskorišćen za prikaz uticaja parametara SVR redukcije već su pokazani nešto drugačiji primeri. Na slici 5.11 su predstavljeni rezultati implementacije regresije nad samo jednom izdvojenom konturom lopatice koja je upotrebljena radi lakše vizuelne prezentacije rezultata. Parametri regresije su grubo podešeni ($\epsilon = 0,01$, $C = 1.000$) i iskorišćen je splajn kernel. Kao što se očekivalo, na ravnom segmentu konture nisu identifikovani noseći vektori, dok je njihov značajan broj, na početku i kraju, odnosno u oblastima znatne promene geometrije. Ovakvi rezultati obezbeđuju dobar stepen očuvanja informacije o geometriji.

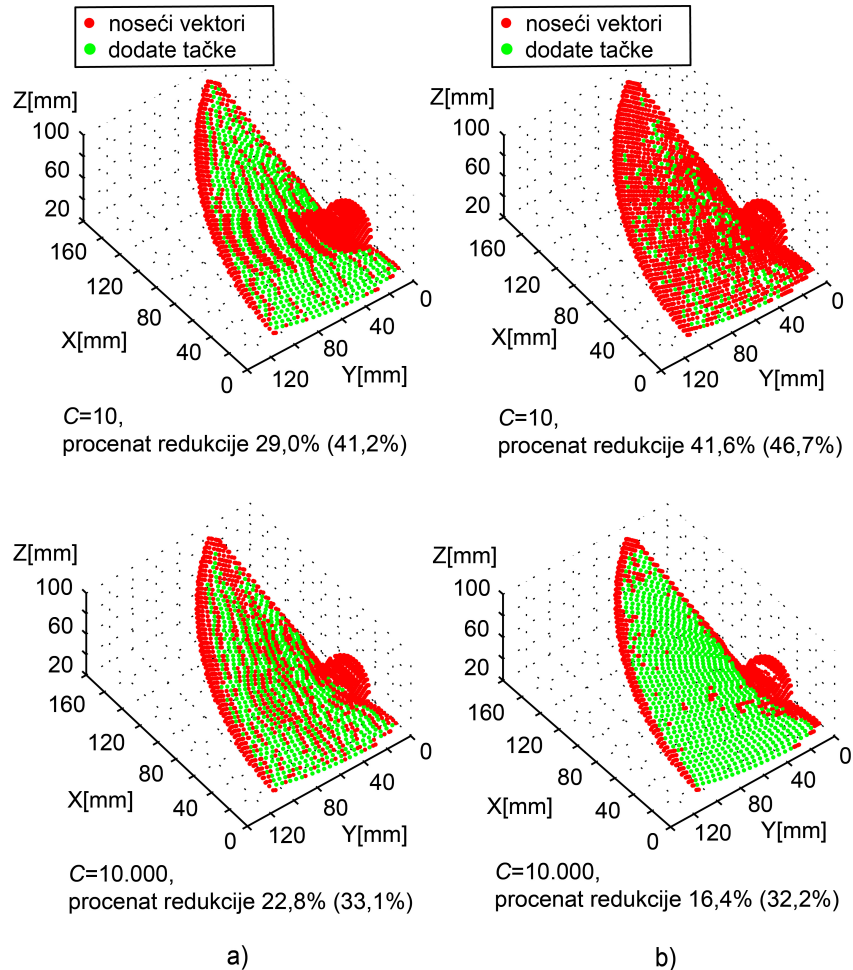
Imajući u vidu potrebu da se redukovani oblak u daljoj eksploataciji upotrebi za primenu neke od metoda rekonstrukcije površi, bitno je ostvariti i određenu uniformnost rasporeda zadržanih tačaka. U tim situacijama nije pogodno ostavljati ravne segmente bez tačaka. Iako bi se zadavanjem izrazito uskih margina greške ili visoke cene za grešku identifikovali noseći vektori i u ravnim segmentima u istoj situaciji bi se značajno povećao i njihov broj u zakrivljenim segmentima. Ovo se ne bi moglo smatrati pogodnim rešenjem pa je zbog toga polazna metoda redukcije proširena i adaptirana. Pored identifikovanih nosećih vektora, u zonama ravne geometrije izdvajaju se i dodatne tačke (označene zelenom bojom na slici 5.11). Ukoliko se u konturi od poslednje identifikovanog nosećeg vektora (posmatrano duž apscise) do unapred definisanog maksimalnog rastojanja (d_{max}) ne identifikuje novi, vrši se izdvajanje jedne postojeće tačke konture i njeno dodavanje u skup redukovanih tačaka. Izabrano je da se dodaje tačka koja je najbliža polovini maksimalnog zadatog rastojanja od poslednje identifikovanog nosećeg vektora. Postupak se dalje ponavlja počev od nove dodate tačke i, ukoliko je praznina između dva noseća vektora veća, dodaju se nove tačke po istom principu. Ovom strategijom se obezbeđuje da se prva dodata tačka u prostoru između udaljenih nosećih vektora nalazi bliže poslednje dodatom nosećem vektoru, a sve naredne u lancu biće na rastojanju jednakom maksimalno zadatom. Treba napomenuti da prag za maksimalno rastojanje treba zadati tako da je reprezentovan brojem koji je deljiv vrednošću rastera oblaka duž konture bez ostatka. U tom slučaju se taj prag može iskazati i fizičkom distancom ili pak brojem uzoraka u konturi. Za primer regresije konture prikazane na pomenutoj slici 5.11 izabrano je da maksimalno rastojanje (u ovom slučaju duž y ose) bude 5 mm odnosno 5 uzoraka. Imajući u vidu zadate parametre regresije i maksimalno rastojanje, izvršena je redukcija konture sa ukupno zadržanih 33,9% tačaka, dok su noseći vektori činili 18,6%.



Slika 5.12: ϵ -SVR izdvojene konture lopatice: a) B-splajn kernel, $\epsilon = 0,15$; b) splajn kernel, $\epsilon = 0,01$.

Nakon adaptacije metode redukcije, na slici 5.12 su prikazani još neki primeri regresije iste izdvojene konture lopatice za različite vrednosti parametara C i ϵ . Pored procenata izdvojenih nosećih vektora u zagradi su dati i procenti ukupno izdvojenih tačaka (i noseći vektori i dodatne tačke). Na ovoj slici je moguće uočiti ranije pomenutu lakšu kontrolu broja nosećih vektora promenom samo vrednosti parametra C dok je vrednost za ϵ konstantna. Naravno, za niske vrednosti cene za grešku na ravnijim delovima se izdvaja veći broj nosećih vektora. U slučaju potrebe povećanja redukcije, odnosno sma-

njenja broja izdvojenih tačaka, dovoljno je povećati samo vrednost parametra C dok nije potrebno menjati marginu greške. Ovako se može kontrolisati sam kvalitet redukcije, ali i smanjiti potrebno vreme procesiranja. Takođe, ovime se omogućava i da regresiona funkcija bude što je moguće ravni-ja. Ovde treba istaći da je B-splajn kernel u ovom slučaju nepovoljniji kod upotrebe nižih vrednosti cene za grešku. Na primeru sa pomenute slike 5.12 gde je redukcija izvršena korišćenjem vrednosti $C = 10$ (u gornjem levom uglu), na ravnom segmentu jasno se može uočiti periodično pojavljivanje malih grupa nosećih vektora, a ne samo individualnih tačaka kao kod splajn kernela. Zbog ovakvog ponašanja može se zaključiti da je u ovim situacijama splajn kernel u prednosti.



Slika 5.13: Redukcija kompletnog oblaka lopatice: a) B-splajn kernel, $\varepsilon = 0,05$; b) splajn kernel, $\varepsilon = 0,002$.

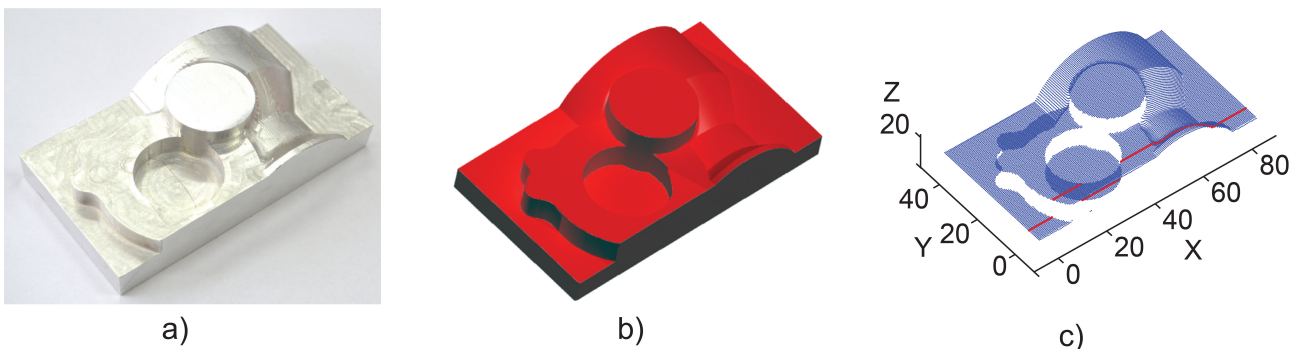
Nakon diskusije na primeru jedne konture, na slici 5.13 prikazano je nekoliko različitih rezultata regresije kompletnog strukturiranog oblaka lopatice. Parametri regresije su podešeni tako da se ostvari približno isti procenat redukovanih tačaka u slučaju oba kernela kako bi poređenje bilo relevantno. Posmatrajući ove rezultate, takođe se može uočiti prethodna tvrdnja da je B-splajn kernel više podložan izdvajanju grupa nosećih vektora na ravnim segmentima. Pored grafičke interpretacije na pomenu-toj slici, u tabeli 5.3 dat je procentualno izražen prikaz broja nosećih vektora i ukupno redukovanih tačaka kao i vremena procesiranja u svakom pojedinačnom slučaju. Potrebno je napomenuti da su zbog relevantnosti podataka predstavljena vremena i ovde data kao srednja vrednost u pet nezavisnih procesiranja. Posmatrajući ova vremena ponovo se može primetiti već izrečena konstatacija o tome da korišćenje B-splajna u odnosu na splajn kernel povećava potrebno vreme izvršavanja, odnosno redukovanja oblaka.

Tabela 5.3: Rezultati redukcije kompletnog oblaka lopatice.

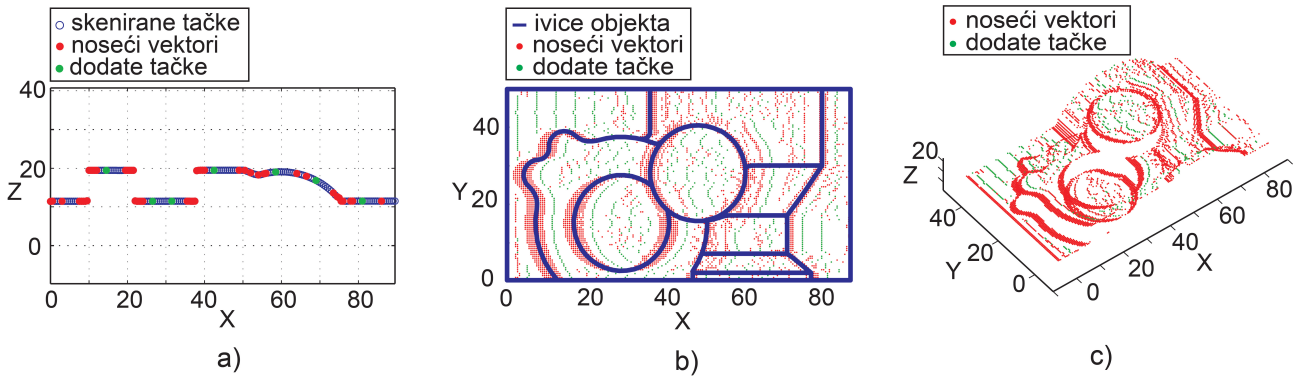
Kernel	ε	C	Noseći vektori [%]	Ukupno sačuvane tačke [%]	Vreme procesiranja [s]
B-splajn	0,05	10	29,0	41,2	189,0
B-splajn	0,05	10.000	22,8	33,1	202,1
splajn	0,002	10	41,6	46,7	93,7
splajn	0,002	10.000	16,4	32,2	114,6

5.3.3. Studija slučaja 3: Objekat sa oštrim ivicama

Treći realni oblak iskorišćen za verifikaciju predstavljene metode redukcije reprezentuje određeni test objekat (slika 5.14(a)) čija se geometrija sastoji od nekoliko osnovnih vrsta primitiva, ali se takođe odlikuje i velikim brojem oštih prelaza u oblaku (oštrim ivicama). Ovaj test objekat predstavlja jedan od standarda za testiranje algoritama u kompjuterskoj grafici i reverznom inženjerstvu [81, 123, 124, 125]. U ovom poglavlju razmatramo samo jedan njegov sken (engl. *single view scan*) kreiran skeniranjem odozgo (slika 5.14(b)). Objekat je skeniran istim optičkim skenerom [24] kao u slučaju lopatice i generisana je 3D trougaona mreža. Da bi se dobile uređene skenirane konture, odnosno strukturirani oblak (slika 5.14(c)) izvršeno je strukturiranje kreirane mreže na isti način kao u prethodnim primerima. U ovom slučaju izabrano je da raster i duž x (pravac skeniranih linija) i duž y ose koordinatnog sistema mreže bude jednak i iznosi 0,5 mm.


Slika 5.14: Test objekat: a) fotografija objekta; b) sken objekta iz jednog pravca; c) strukturirani oblak.

Na slici 5.15(a) prikazani su rezultati primene predstavljenog algoritma redukcije na jednoj konturi oblaka (obeležena crvenom bojom na slici 5.14(c)), dok je na ostatku slike predstavljen kompletno redukovan oblak. U ovom slučaju pri upotrebi splajn kernela korišćene su sledeće vrednosti parametra: $\varepsilon = 0,05$, $C = 100$ i $d_{max} = 5$ mm (10 uzoraka). Pri regresiji ovim parametrima ostvaren je procenat zadržanih tačaka od 29,0%. Predstavljeni proces redukcije je grafički prikazan u cilju prezentacije još jedne pogodne osobine postavljenog modela redukcije baziranom na ε -SVR. Naime, kao što je ranije pokazano, SVR izdvaja noseće vektore na mestima znatne promene geometrije, odnosno u oblastima gde generisana regresiona funkcija ima poteškoća da prati raspored tačaka konture. Zbog ove osobine, pri redukciji kontura koje sadrže oštre prelaze, znatan broj tačaka se zadržava upravo na mestima ovih prelaza. Ovime se dolazi do redukovanog oblaka koji imaju dobru gustinu tačaka na ivicama (sa obe njihove strane) objekta bez obzira na stepen same redukcije što se može smatrati prednošću ove metode i povoljnim slučajem za održanje visokog stepena informacije o geometriji posmatranih objekata. Sam kvalitet prezentovane metode redukcije biće proveren i diskutovan u narednom segmentu ovog poglavlja.



Slika 5.15: Redukcija kompletnog oblaka test objekta: a) jedna izdvojena kontura; b) redukovani oblak upoređen sa ivicama objekta u pogledu odozgo; c) redukovani oblak u izometriji.

5.4. Ocena kvaliteta predložene metode redukcije

Nakon predstavljanja algoritma i njegove eksperimentalne verifikacije, potrebno je sagledati i oceniti sam kvalitet postavljene metode redukcije. Redukcija se može smatrati kvalitetnom ukoliko se nakon značajnog smanjenja broja tačaka oblaka informacija o geometriji skeniranog objekta neznatno umanjuje. Pošto je metoda zasnovana na redukovanju jedne po jedne konture strukturiranog oblaka i postavljena metoda ocene kvaliteta biće bazirana na proveru količine zadržane informacije u pojedinačnoj skeniranoj konturi. Za potrebe procene kvaliteta redukcije u ovom radu predložena je metoda zasnovana na NURBS krivama koje se često koriste u kompjuterskoj grafici [126].

Pri proveru kvaliteta redukcije jedne izdvojene konture biće generisane dve NURBS krive:

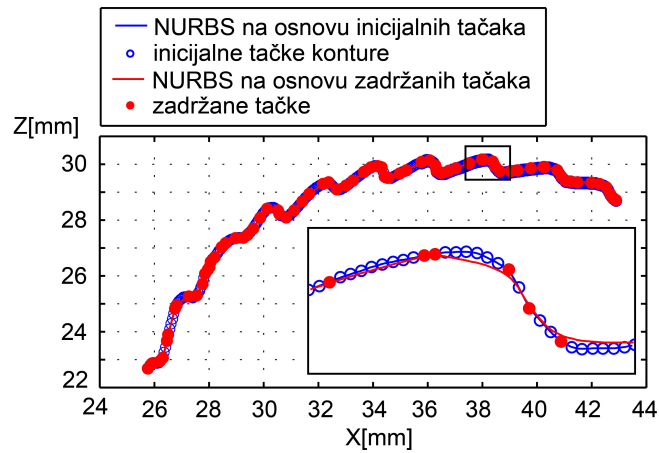
- $NURBS_1$ - kriva konstruisana nad tačkama polazne (inicijalne) konture,
- $NURBS_2$ - kriva konstruisana nad tačkama zadržanim nakon redukcije (redukovanim tačkama).

Estimacija kvaliteta redukcije vrši se upoređivanjem ove dve NURBS krive. Visok nivo podudaranja njihovog oblika direktno znači i visok nivo održanja informacije o geometriji nakon izvršenog procesa redukcije. Sa druge strane, nizak nivo podudaranja ove dve krive implicira loš kvalitet redukcije, odnosno neadekvatno gubljenje većeg dela informacije o obliku koja je inicijalno dobijena skeniranjem.

Prilikom konstruisanja NURBS kriva treba definisati njihov skup kontrolnih tačaka, vektor čvorovnih tačaka, i težinske faktore za svaku kontrolnu tačku. U ovom radu izabrano je da za $NURBS_1$ kontrolne tačke budu sve skenirane tačke konture, dok su u slučaju $NURBS_2$ to redukovane tačke. Vektor čvorova \mathbf{u} sadrži $n + 3$ elemenata pri čemu n predstavlja broj kontrolnih tačaka svake krive. Vrednosti čvorova su postavljene u ekvidistantnom rastućem nizu unutar intervala $[0, 1]$ izuzev prva tri koji imaju vrednost 0 i poslednja tri koji imaju vrednost 1. Ovime se obezbeđuje da NURBS kriva prolazi kroz prvu i poslednju tačku konture. Težine svih kontrolnih tačaka su postavljene na vrednost 1 kako bi sve tačke konture imale jednak uticaj na oblik krive. Na slikama 5.16 i 5.18 predstavljene su NURBS krive koje su kreirane na osnovu inicijalnih i redukovanih kontura zmaja i lopatice turbine, respektivno.

Poređenje oblika definisanih NURBS kriva je moguće izvršiti upoređivanjem y koordinata odgovarajućih tačaka na njima. Zbog toga je potrebno za te dve NURBS krive ($NURBS_1$ i $NURBS_2$) generisati dva skupa tačaka na njima. Kako bi se ova dva skupa mogla međusobno upoređivati, potrebno je da tačke na krivama budu izračunate u istim pozicijama duž apscise odnosno u identičnom unapred definisanom rasteru $r_j, j = 1, \dots, m$. Raster se može definisati proizvoljno, ali svakako je najprirodnije izabrati ga tako da odgovara rasteru inicijalne (skenirane konture). Navedenim postupkom dobijaju se dva skupa koordinata po ordinati podudarnih tačaka na krivama - $NURBS_1(r_j)$ i $NURBS_2(r_j)$.

Za estimaciju kvaliteta redukcije potrebno je porediti NURBS krive pre i posle izvršenog procesa,



Slika 5.16: NURBS kreiran na osnovu redukovane konture ($\varepsilon = 0,025$, $C = 1.000$, splajn kernel) zajedno sa NURBS-om kreiranim na osnovu inicijalne konture sa slike 5.5.

odnosno njihove skupove podudarnih tačaka u zadatom rasteru. Za ovu namenu su predložene tri vrste ocena (metrika). Prva ocena je vrednost maksimalne apsolutne devijacije (odstupanja) dev_m koja predstavlja najveću razliku ove dve krive u jednoj pojedinačnoj tački rastera. Sledeća ocena je ukupno kvadratno odstupanje dev_t koje možemo definisati kao:

$$dev_t = \sum_{j=1}^m [NURBS_1(r_j) - NURBS_2(r_j)]^2, \quad (5.15)$$

gde m predstavlja broj tačaka izabranog rastera. Treća ocena je srednje apsolutno odstupanje dev_a koje predstavlja osrednjenu vrednost svih odstupanja u svakoj tački rastera posebno i definiše se kao:

$$dev_a = \frac{\sum_{j=1}^m |NURBS_1(r_j) - NURBS_2(r_j)|}{m}. \quad (5.16)$$

Tabela 5.4: Estimacija kvaliteta regresije konture sa slike 5.5.

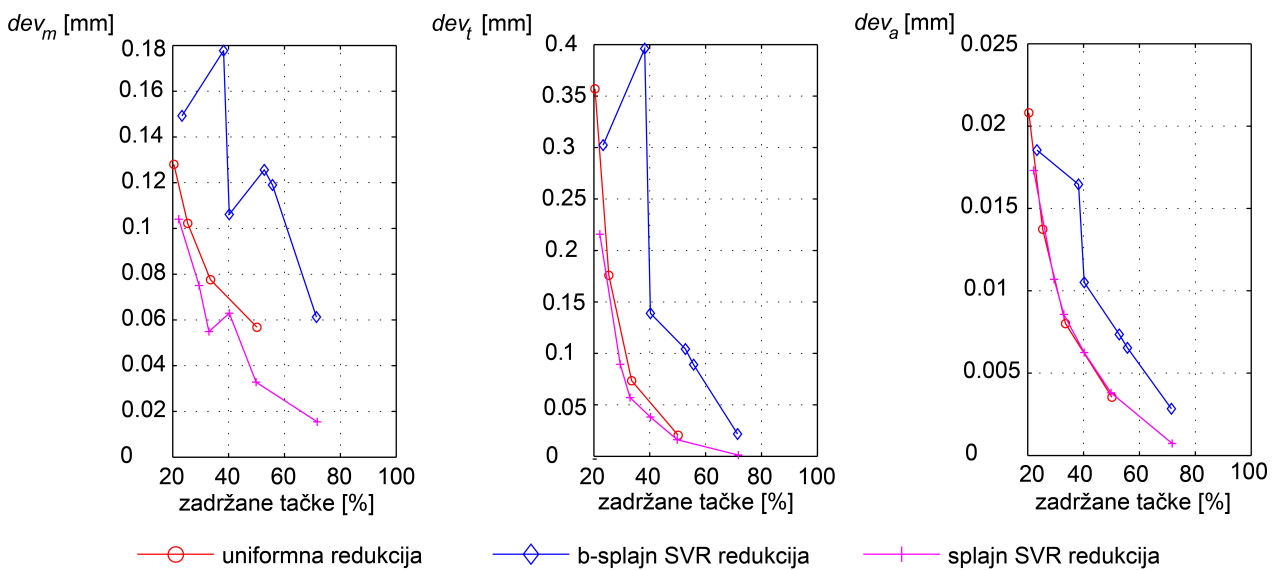
Kernel	ε	C	Noseći vektori [%]	Ukupno zadržane tačke [%]	Maksimalno odstupanje [mm]		Ukupno kvadratno odstupanje [mm]		Srednje odstupanje [mm]	
					Zadrž. tačke	Noseći vektori	Zadrž. tačke	Noseći vektori	Zadrž. tačke	Noseći vektori
B-splajn	0,1	10	70,6	71,4	0,0613	0,0608	0,0217	0,0227	0,0028	0,0032
B-splajn	0,15	10	52,8	55,7	0,1190	0,1901	0,0891	0,2527	0,0065	0,0113
B-splajn	0,15	100	49,6	52,8	0,1256	0,2076	0,1043	0,3060	0,0073	0,0129
B-splajn	0,15	1.000	36,2	40,2	0,1061	0,1061	0,1389	0,2599	0,0105	0,0179
B-splajn	0,2	10	33,5	38,2	0,1777	0,3185	0,3960	1,9488	0,0165	0,0391
B-splajn	0,15	10.000	18,1	23,3	0,1492	0,3040	0,3023	1,7207	0,0185	0,0448
splajn	0,002	10	71,1	71,7	0,0154	0,0154	0,0012	0,0013	0,0007	0,0008
splajn	0,01	10	49,0	49,9	0,0328	0,0327	0,0161	0,0185	0,0038	0,0043
splajn	0,025	10	37,6	40,2	0,0630	0,0901	0,0382	0,1069	0,0062	0,0096
splajn	0,01	100	31,8	32,9	0,0549	0,0550	0,0571	0,0631	0,0086	0,0093
splajn	0,01	1.000	27,7	29,4	0,0751	0,0751	0,0894	0,1256	0,0107	0,0131
splajn	0,025	1.000	18,9	22,2	0,1040	0,1040	0,2158	0,3105	0,0173	0,0222

U ovom poglavlju biće izvršena estimacija već korišćenih karakterističnih kontura zmaja i lopatice turbine koje su prikazane na slikama 5.5 i 5.11, respektivno. Definisane tri devijacije (dev_m , dev_t , i dev_a) pri redukciji broja tačaka konture zmaja za različite vrednosti SVR parametara prikazane su u tabeli 5.4. Kao dodatni pokazatelj, pored ocena za NURBS krive konstruisane nad svim zadržanim

tačkama (zajedno sa dodatim tačkama na $d_{max} = 0,5$ mm, odnosno na 10 uzoraka) u tabeli su date i vrednosti generisane samo za detektovane noseće vektore. Posmatrajući navedenu tabelu i uzimajući u razmatranje veličinu skenirane geometrije kao i veoma gust raster konture, može se zaključiti da su vrednosti definisanih odstupanja niske. Sa druge strane, primetno je da je za približno isti stepen redukcije splajn kernel u prednosti u odnosu na B-splajn zbog nešto manjih devijacija.

Tabela 5.5: Uniformna redukcija broja tačaka konture sa slike 5.5.

Faktor uniformne redukcije	Zadržane tačke [%]	Maksimalno odstupanje [mm]	Ukupno kvadratno odstupanje [mm]	Srednje odstupanje [mm]
2	50,1	0,0568	0,0205	0,0035
3	33,5	0,0776	0,0735	0,0080
4	25,4	0,1023	0,1759	0,0137
5	20,4	0,1280	0,3570	0,0208

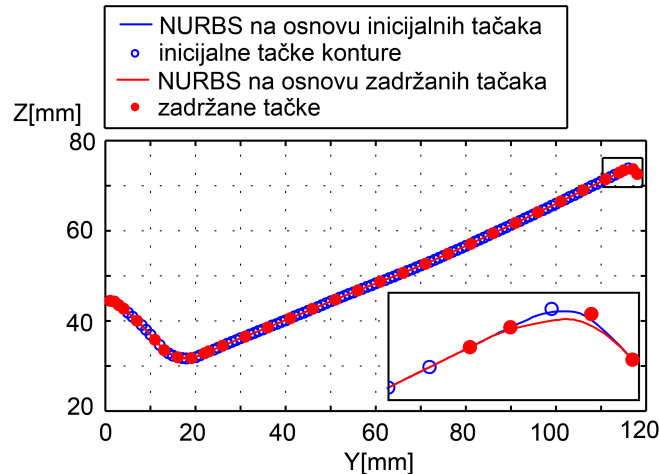


Slika 5.17: Komparacija predložene SVR metode i uniformne redukcije na primeru konture sa slike 5.5.

Iako prikazani rezultati potvrđuju prethodne navode o visokom stepenu očuvanja informacije o geometriji nakon primene predloženog metoda redukcije, performanse metoda su po istom principu upoređene i sa metodom uniformne redukcije (engl. *uniform down-sampling*). Definisane vrednosti devijacija za izvršenih nekoliko uniformnih redukcija broja tačaka različitim faktorima (koracima) date su u tabeli 5.5. Kao što se može primetiti, ove vrednosti su uporedivih veličina sa onima dobijenim predloženom SVR redukcijom. Radi boljeg sagledavanja na slici 5.17 dat je grafički uporedni prikaz obe metode korišćenjem podataka iz tabela 5.4 i 5.5.

Posmatranjem uporednog prikaza rezultata predložene SVR i uniformne redukcije (slika 5.17) može se zaključiti da je SVR splajn kernelom dala najbolje rezultate u slučaju maksimalne i ukupne kvadratne devijacije dok su vrednosti srednje devijacije vrlo slične izvršenoj uniformnoj regresiji nad istom konturom. Sa druge strane, SVR regresija upotrebom B-splajna nad ovom vrlo gustom konturom visokog stepena promene geometrije pokazuje lošije rezultate od ostale dve vrste redukcije. Takođe, posmatrajući srednje apsolutno odstupanje, rezultati SVR B-splajn regresije, za razliku od ostalih, ne pokazuju jasno uočljiv trend linarnog povećanja greške usled smanjenja broja tačaka. Sa druge strane, treba istaći činjenicu da se uniformnom regresijom upotrebom određenih faktora (2, 3, 4...) ne može jasno podešavati željeni stepen redukcije i ne može se dobiti više od 50% zadržanih tačaka. Imajući u vidu navedene nedostatke uniformne regresije, upotrebom predložene SVR (podešavanjem njenih

parametara) se može ostvariti željeni procenat regresije (naravno i veći od 50%) i u tom slučaju se mogu ostvariti zanemarljivi gubici informacije za manji procenat redukcije koji sa druge strane može biti veoma značajan kod oblaka sa veoma velikim brojem tačaka. Takođe, SVR se može koristiti i kao adaptivna metoda. Kao povratna informacija se može zadati vrednost neke od željenih devijacija (sa određenom tolerancijom) i automatskim podešavanjem parametara u iterativnom postupku se može izvršiti regresija u željenom kvalitetu. Međutim, iako su izvršena inicijalna istraživanja ovakve upotrebe SVR, njene prednosti i mane kao i mogućnost inteligentnog postavljanja pragova neće biti prezentovane u ovoj disertaciji.



Slika 5.18: NURBS kreiran na osnovu redukovane konture ($\epsilon = 0,05$, $C = 10.000$, splajn kernel) zajedno sa NURBS-om kreiranim na osnovu inicijalne konture sa slike 5.11.

Tabela 5.6: Estimacija kvaliteta regresije konture sa slike 5.11.

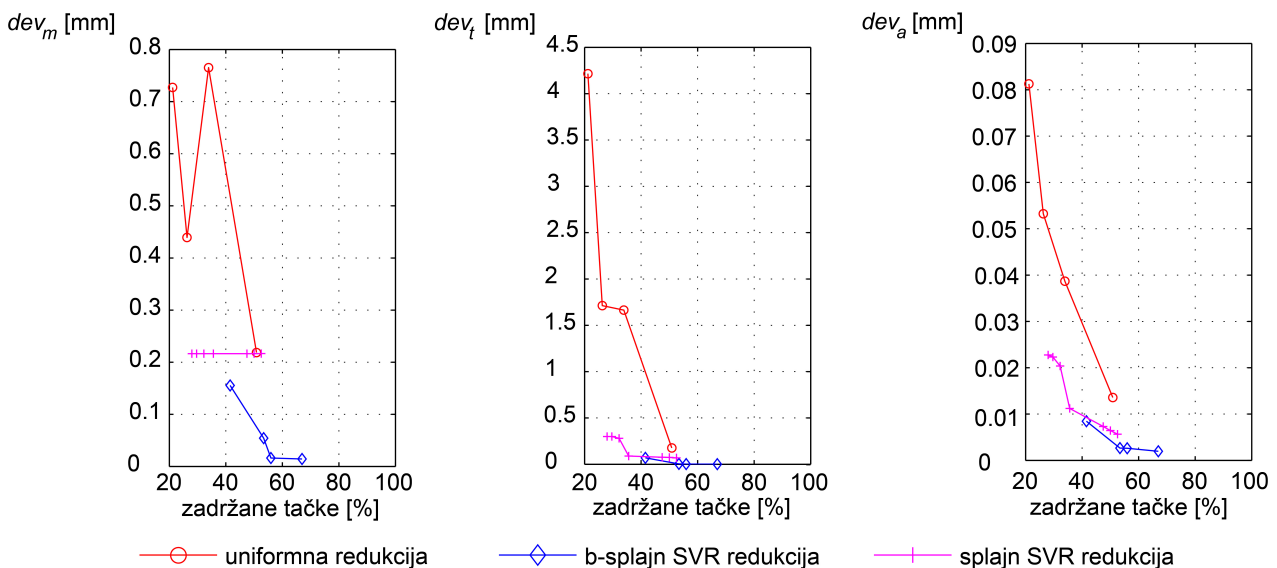
Kernel	ϵ	C	Noseći vektori [%]	Ukupno zadržane tačke [%]	Maksimalno odstupanje [mm]		Ukupno kvadratno odstupanje [mm]		Srednje odstupanje [mm]	
					Zadrž. tačke	Noseći vektori	Zadrž. tačke	Noseći vektori	Zadrž. tačke	Noseći vektori
B-splajn	0,05	10	60,2	66,9	0,0144	0,0740	0,0017	0,0496	0,0024	0,0020
B-splajn	0,05	100	48,3	55,9	0,0161	0,1270	0,0022	0,1669	0,0025	0,0026
B-splajn	0,05	1.000	44,1	53,4	0,0542	0,0540	0,0046	0,0148	0,0038	0,0027
B-splajn	0,05	10.000	28,8	41,5	0,1555	0,4325	0,0694	0,6634	0,0184	0,0085
splajn	0,002	10	50,8	52,5	0,2165	0,2164	0,0699	0,0701	0,0057	0,0060
splajn	0,002	100	47,4	50,0	0,2165	0,2164	0,0744	0,0756	0,0064	0,0073
splajn	0,002	1.000	38,9	47,5	0,2164	0,2164	0,0759	0,4366	0,0073	0,0355
splajn	0,002	10.000	22,0	35,6	0,2164	0,7859	0,0895	18,135	0,0112	0,2575
splajn	0,05	10	18,6	32,2	0,2164	0,2497	0,2801	1,8849	0,0204	0,0899
splajn	0,05	100	15,2	29,7	0,2164	0,3761	0,2994	4,7092	0,0223	0,1477
splajn	0,05	1.000	11,9	28,0	0,2164	1,4806	0,2998	79,505	0,0228	0,6490
splajn	0,05	10.000	11,9	28,0	0,2164	1,4806	0,2998	79,505	0,0228	0,6490

Nakon estimacije kvaliteta izdvojene konture visokog stepena promene geometrije izvršena je i ocena konture lopatice turbine koja poseduje značajan ravni segment (kontura sa slike 5.11). Rezultati vrednosti devijacija pri SVR regresiji konture lopatice različitim parametrima su prikazani u tabeli 5.6. Posmatranjem prikazanih devijacija može se zaključiti da vrednosti pre svega maksimalnog odstupanja ni u jednom slučaju ne prelaze opseg rezolucije skenera kojime je generisan oblak tačaka lopatice. Ukoliko se, sa druge strane, fokusiramo na kvalitet redukcije kontura sa izraženim pravolinijskim segmentima (kakve su uostalom i konture razmatrane lopatice), može se zaključiti da B-splajn poseduje određene prednosti u odnosu na splajn kernel. Interesanto je zapaziti i situaciju pri visokim stepenima

redukcije odnosno malom broju zadržanih tačaka kod konutra koje sadrže znatne ravne segmente. U pomenutoj tabeli može se uočiti da u slučaju korišćenja splajn kernela sa nešto širom marginom greške ($\varepsilon = 0,05$) i pri niskim procentima izdvojenih nosećih vektora, situacija pri povećanju cene za grešku se ne menja znatno (posmatrajući procenete ukupno zadržanih tačaka i vrednosti devijacija). Ovo se može objasniti time da se na ravnom segmentu konture noseći vektori ne izdvajaju uopšte, već da se situacija neznatno menja samo na zakrivljenim delovima što dovodi do veoma malih varijacija. Šta više, odstupanja za dve poslednje navedene regresije (splajn sa $C = 1.000$ i $C = 10.000$) su identična.

Tabela 5.7: Uniformna redukcija broja tačaka konture sa slike 5.11.

Faktor uniformne redukcije	Zadržane tačke [%]	Maksimalno odstupanje [mm]	Ukupno kvadratno odstupanje [mm]	Srednje odstupanje [mm]
2	50,8	0,2185	0,1770	0,0136
3	33,9	0,7652	1,6644	0,0387
4	26,3	0,4394	1,7123	0,0532
5	21,2	0,7271	4,2164	0,0813



Slika 5.19: Komparacija predložene SVR metode i uniformne redukcije na primeru konture sa slike 5.11.

Kao i u prethodnom primeru i u slučaju konture lopatice upoređeni su rezultati SVR regresije sa uniformnom. Devijacije i procenti redukcije za uniformnu metodu su prikazani odvojeno u tabeli 5.7 dok su grafički uporedni rezultati obe vrste regresije dati na slici 5.19. Može se zaključiti da su u slučaju korišćenja SVR B-splajn metode za slične procenete redukcije ostvarene znatno manje devijacije nego pri uniformnoj redukciji, a čak i nešto niže u odnosu na SVR splajn kernelom. Sa druge strane, u ovom slučaju se jasno vidi da su vrednosti srednjeg apsolutnog odstupanja predložene SVR regresije i više od dva puta manje nego kod uniformne.

Analizom prikazanih rezultata estimacije kvaliteta u slučaju dve konture vrlo različitog oblika, gustine i broja tačaka utvrđeno je da predložena metoda redukcije zasnovana na ε -SVR ima potencijal za dalju upotrebu u sistemima reverznog inženjerstva. Naravno, upotreba ovako postavljene metode redukcije kod sistema koji zahtevaju rad u realnom vremenu mora biti detaljnije istražena i prilagođena pre svega u smislu implementacije algoritama na višem nivou. Na kraju poglavlja treba napomenuti da su delovi istraživanja na temu predložene ε -SVR redukcije strukturiranih oblaka tačaka tokom izrade ovog rada objavljeni u [127].

6. Prepoznavanje jedne klase površi drugog reda iz struktuiranih oblaka tačaka

Posmatrano iz prizme procesa reverznog inženjerstva geometrijskih modela (poglavlje 2) redukcija broja tačaka oblaka, čija je analiza izvršena u prethodnom poglavlju, se može svrstati u fazu pre-procesiranja podataka dobijenih skeniranjem. Sledeća faza na putu ka dobijanju CAD modela skeniranog objekta je segmentacija oblaka i prepoznavanje njegovih gradivnih geometrijskih primitiva. U skladu sa ciljem ovog rada u narednim poglavljima biće predstavljena istraživanja na ovu temu kao i neki od predloženih pristupa rešavanju mogućih problema u ovoj oblasti. U ovom konkretnom poglavlju biće reči o segmentaciji struktuiranih oblaka tačaka u cilju prepoznavanja određenih gradivnih površi drugog reda.

Većina objekata u mašinstvu je, pored ravni, izgrađena i od površina koje se sastoje iz delova površi drugog reda - tzv. kvadrika (cilindar, sfera, konus, elipsoid, hiperboloid...) i torusa. Osnov metode koja je tema ovog poglavlja je segmentacija eliptičnih segmenata iz 2D skeniranih kontura. Zbog toga (kao što će biti kasnije pokazano) on je sposoban za prepoznavanje kvadrika koje se u skeniranim linijama struktuiranog oblaka reprezentuju upravo kao eliptični segmenti. U ovu grupu spadaju pre svih cilindar i sfera (tzv. prirodne kvadrike), ali i njihovi opšti oblici kao što su eliptični cilindar i elipsoid (tzv. opšte kvadrike). Zbog navedenog, u ovom poglavlju, kada se bude govorilo o kvadrikama misliće se pre svega na prethodno pobrojane tipove. Ipak i neke druge kvadrike (npr. konus) u zavisnosti od načina skeniranja mogu biti predstavljene eliptičnim segmentima u skeniranim linijama i kao takve mogu biti prepoznate. Međutim ovi slučajevi neće biti istraženi u ovom radu.

Prema [123] tehnike segmentacije površina iz oblaka tačaka se mogu podeliti na:

- tehnike zasnovane na detektovanju ivica, i
- tehnike zasnovane na regionima.

Pri korišćenju tehnika zasnovanih na detekciji ivica, nakon prve faze, u kojoj se pronađu tačke oblaka koje reprezentuju ivice, moguće je doći do odvojenih segmenata oblaka koji su određeni izdvojenim ivicama. Ove metode su pogodne za segmentaciju površina koje poseduju G0 kontinualne prelaze (koje je lako uočiti), dok za slučajeve G1 kontinualnosti ne daju dobre rezultate (veoma je teško uočiti same ivice). Treba napomenuti da veliki broj delova u mašinstvu, pored oštih ivica poseduje i površine koje se sastoje od G1 kontinualnih segmenata. Zbog navedenog, metoda segmentacije koja se razmatra u ovom poglavlju je orijentisana pre svega na pomenute slučajeve površina G1 kontinualnosti i pripada grupi tehnika zasnovanih na regionima.

6.1. Segmentacija eliptičnih regiona iz skeniranih linija

Segmentacija eliptičnih regiona iz skeniranih linija koja je kreirana u ovom radu zasnovana je na direktnoj regresiji elipsi metodom najmanjih kvadrata [128]. Da bi ovaj proces bio bliže pojašnjen potrebno je poći od jednačine elipse u implicitnom obliku koja se može predstaviti preko opšte jednačine konusnih preseka:

$$a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6 = 0, \quad (6.1)$$

s tim što za elipsu mora biti zadovoljen dodatni uslov:

$$a_2^2 - 4a_1a_3 < 0. \quad (6.2)$$

U jednačini (6.1) promenljive x i y reprezentuju koordinate tačke na konusnom preseku, dok a_i , $i = 1, \dots, 6$ predstavljaju parametre krive. Ovu jednačinu možemo zapisati i u matričnoj formi kao:

$$\mathbf{x} \cdot \mathbf{a} = 0, \quad (6.3)$$

pri čemu je $\mathbf{x} = [x^2 \ xy \ y^2 \ x \ y \ 1]$ vektor promenljivih, dok je $\mathbf{a} = [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6]^T$ matrica parametara (koeficijenata) elipse.

Da bi se pronašla elipsa koja predstavlja najbolju aproksimaciju datog skupa tačaka (x_j, y_j) , $j = 1, \dots, n$ neophodno je izvršiti estimaciju parametara a_i , $i = 1, \dots, 6$ odnosno članova matrice \mathbf{a} . Korišćenjem metode najmanjih kvadrata postavljeni problem može biti sveden na problem optimizacije (minimizacije) [128]:

$$\min_{\mathbf{a}} \|\mathbf{D}\mathbf{a}\|^2, \quad (6.4)$$

gde je sa \mathbf{D} obeležena matrica sistema (engl. *design matrix*) oblika:

$$\mathbf{D} = \begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1 \end{bmatrix}. \quad (6.5)$$

Da bi proces minimizacije (6.4) učinili efikasnijim autori u [128] vrše normalizaciju uvodeći uslov:

$$4a_1a_3 - a_2^2 = 1, \quad (6.6)$$

koji se u matricnoj formi može zapisati kao:

$$\mathbf{a}^T \mathbf{C} \mathbf{a} = 1, \quad (6.7)$$

gde je \mathbf{C} matrica dimenzije 6x6 sa elementima $\mathbf{C}(1, 3) = \mathbf{C}(3, 1) = 2$, $\mathbf{C}(2, 2) = -1$ i svim ostalim jednakim 0.

Postavljeni problem minimizacije se sada može predstaviti sledećim sistemom jednačina:

$$\begin{aligned} \mathbf{S}\mathbf{a} &= \lambda \mathbf{C}\mathbf{a} \\ \mathbf{a}^T \mathbf{C} \mathbf{a} &= 1, \end{aligned} \quad (6.8)$$

gde je $\mathbf{S} = \mathbf{D}^T \mathbf{D}$ tzv. informaciona matrica [129], a λ predstavlja uvedeni Lagranžov množilac.

Autori u [128] su pokazali da sistem jednačina (6.8) ima tačno jedno rešenje u vidu sopstvenog vektora koji odgovara pozitivnim sopstvenim vrednostima matrica \mathbf{S} i \mathbf{C} . Članovi ovog sopstvenog vektora ujedno predstavljaju vrednosti traženih parametara elipse a_i , $i = 1, \dots, 6$. Ovo znači da ukoliko su tačke (x_j, y_j) , $j = 1, \dots, n$ uzorkovane sa neke elipse bez bilo kakvog šuma onda će njihove koordinate zadovoljavati postavljenu jednačinu elipse. U tom slučaju informaciona matrica \mathbf{S} biće singularna i neće biti moguće izvršiti estimaciju parametara elipse metodom najmanjih kvadrata [129]. Međutim, ukoliko dati skup tačaka približno definiše određenu elipsu, matrica \mathbf{S} neće biti singularna već bliska singularnoj i u tom slučaju je moguće rešiti sistem (6.8) i dobiti ocenu vrednosti parametara te elipse. Kako u realnosti reverznog inženjerstva tačke koje se dobiju skeniranjem predstavljaju određenu aproksimaciju skenirane geometrije i sadrže grešku merenja (skeniranja) i određeni nivo šuma, one u skeniranim konturama oblaka nikada egzaktno ne predstavljaju eliptične segmente. Upravo ova osobina omogućava primenu prikazane metodologije regresije elipse (estimacije parametara jednačine elipse) za potrebe formiranja procesa segmentacije eliptičnih segmenata iz skeniranih kontura koji će biti prezentovan u nastavku.

6.2. Definicija metode i algoritma

Primenom direktne regresije elipse metodom najmanjih kvadrata može se kreirati metoda segmentacije eliptičnih regiona iz skeniranih linija koji predstavlja osnov metode prepoznavanja i estimacije parametara kvadrata iz struktuiranih oblaka tačaka. Na osnovu pojašnjenja iz poglavlja 6.1, segmenta-

cija eliptičnih regiona je zasnovana na činjenici da je, kada se radi o proceni parametara elipse iz skupa tačaka koje tačno pripadaju elipsi, informaciona matrica \mathbf{S} singularna. Ukoliko sa druge strane skup skeniranih tačaka aproksimativno predstavlja neku elipsu informaciona matrica je bliska singularnoj i u tom slučaju biće ispunjena dva uslova:

- recipročni faktor uslovljenosti matrice \mathbf{S} biće blizak nuli,
- sopstvene vrednosti matrice $\mathbf{S}^{-1}\mathbf{C}$ mogu biti bliske nuli ili čak manje od nule [130].

```

ULAZ:   N tačaka konture, koordinate  $\mathbf{x}$ ,  $\mathbf{y}$ 
          t_rc, t_ov - pragovi za segmentaciju


---


for i=1:N
    x1=x(1 : i);
    y1=y(1 : i);
    D=[x1^2 x1*y1 y1^2 x1 y1 ones(i)];
    S=D'*D;
    if rcond(S)>t_rc
        prelaz=i; break
    else
        [svck, svr]=eig(inv(S)*C)
        sv=abs(svr(find(svr>0)))
        if sv<t_ov
            prelaz=i; break
        end if
    end if
end for


---


IZLAZ:  prelaz - detektovane tačke prelaza između segmenata

```

Slika 6.1: Pseudokod jedne iteracije algoritma segmentacije skeniranih kontura.

Metoda segmentacije koja se predlaže u ovom poglavlju je upravo zasnovana na uslovu singularnosti informacione matrice, odnosno na dva prethodno navedena uslova. Algoritam ove metode se može svrstati u one koji koriste strategiju rasta regiona počev od jedne inicijalne tačke. Naime, skenirane tačke svake konture se obrađuju jedna po jedna, počev od prve u nizu. U svakoj iteraciji se računaju vrednosti matrice \mathbf{D} , a samim tim se dobijaju i vrednosti matrice \mathbf{S} čije se osobine u tom trenutku ispituju. Ukoliko vrednost recipročnog faktora uslovljenosti informacione matrice ili njene sopstvene vrednosti budu u određenoj poziciji (više, odnosno niže) u odnosu na unapred zadate vrednosti pragova za segmentaciju (t_{rc} i t_{ov}), može se reći da se matrica \mathbf{S} udaljava od singularnosti, odnosno da tekuća tačka konture ne pripada skupu do tada obrađenih tačaka koje aproksimiraju određenu elipsu. Ta tačka skenirane konture se označava kao prelaz između dva eliptična regiona i proglašava se prvom tačkom naredne konture. Postupak segmentacije se ponavlja za tu narednu tačku, i tako redom dok se ne obrade sve tačke skenirane linije. Pseudokod za jednu iteraciju opisane metode segmentacije skenirane linije dat je na slici 6.1. Treba naglasiti da se pragovi za vrednosti recipročnog faktora uslovljenosti i sopstvenih vrednosti mogu podešavati u zavisnosti od nivoa šuma u konturi (2D signalu).

Tokom testiranja predloženog algoritma segmentacije kontura, primećeno je da je algoritam u prisustvu višeg nivoa šuma podložan izdvajanju većeg broja (od potrebnog) segmenata na konturi - presegmentaciji. Zbog toga je kreirana dodatna procedura grupisanja prekobrojnih segmenata. Nakon prve faze segmentacije pristupa se upoređivanju svaka dva susedna segmenta počev od prvog. Zbog permanentne prisutnosti određenog šuma u konturama, za svaki izdvojeni segment moguće je izvršiti estimaciju parametara elipsi korišćenjem postupka iz [128]. Na osnovu procenjenih parametara elipsi se za x koordinate tačaka dva susedna segmenata računaju nove y koordinate. Odluka o tome da li je ta dva segmenta potrebno spojiti ili ne donosi se na osnovu uporedne analize tih novo izračunatih y koordinata, a na osnovu određene norme. Ukoliko y koordinate dva susedna segmenta, estimirane korišćenjem parametara \mathbf{a} , obeležimo sa y_{e1} i y_{e2} , originalne sa y_1 i y_2 , a ukupne brojeve tačaka tih

segmenata sa n_1 i n_2 , respektivno, onda se norma za spajanje segmenata na osnovu zadatog praga t_{gr} može opisati na sledeći način:

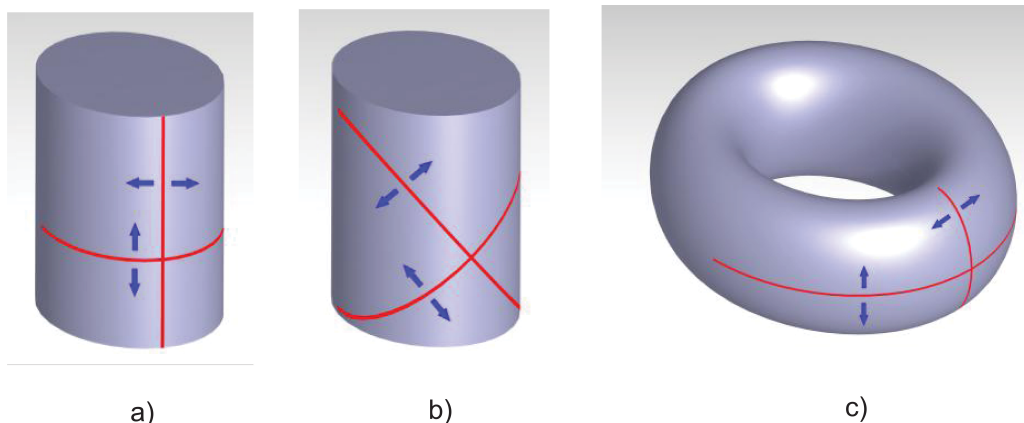
$$\left| \frac{\sum_{i=1}^{n_1} |y_{e1i} - y_{1i}|}{n_1} - \frac{\sum_{i=1}^{n_2} |y_{e2i} - y_{2i}|}{n_2} \right| > t_{gr}. \quad (6.9)$$

Pseudokod predstavljenog algoritma grupisanja dat je na slici 6.2.

```

ULAZ:   x, y - koordinate tačaka konture
           t_gr - prag za grupisanje
           prelaz - vektor pozicija tačaka prelaza
    k=1; j=prelaz(2);
    for i=1:length(prelaz)
        x1=x(k : prelaz(i)); y1=y(k : prelaz(i));
        proceniti a1
        izračunati procenjene y1_e
        x2=x(prelaz(i) : j); y2=y(prelaz(i) : j);
        proceniti a2
        izračunati procenjene y2_e
        if abs(sum(abs(y1_e-y1))/length(y1)-
        abs(sum(abs(y2_e-y2))/length(y2))>t_gr
            kon_prelaz(k)=prelaz(i);
        end if
        k=prelaz(i);
        if i<length(prelaz)-1
            j=prelaz(i+2);
        else
            j=length(x);
        end if
    end for
IZLAZ:   kon_prelaz - konačne pozicije tačaka pre-
           laza nakon grupisanja
    
```

Slika 6.2: Pseudokod procesa grupisanja eliptičnih segmenata.



Slika 6.3: Mogući pravci skeniranja određenih kvadirka: a) skeniranje cilindra duž ose i upravno na nju; b) skeniranje cilindra u pravcima pod uglom u odnosu na osu; c) skeniranje eliptičnog torusa.

Ekstrapolacijom metode za segmentaciju eliptičnih regiona iz skeniranih linija na dve dimenzije, moguće je kreirati algoritam za segmentaciju delova površi koje pripadaju eliptičnom cilindru ili elipsoidu. Naime, u strukturiranom oblaku tačaka dobijenom skeniranjem elipsoida svaka skenirana

linija biće u obliku elipse. Sa druge strane, pri skeniranju eliptičnog cilindra moguće su dve situacije u zavisnosti od načina skeniranja:

- da skenirane linije u jednom pravcu predstavljaju duži, a u drugom elipse (slika 6.3(a)),
- da skenirane linije u oba pravca predstavljaju elipse (slika 6.3(b)).

Naravno, postoje i druge površi koje mogu u skeniranim linijama biti reprezentovane elipsama u oba pravca – tipičan primer je opšti eliptični torus (slika 6.3(c)).

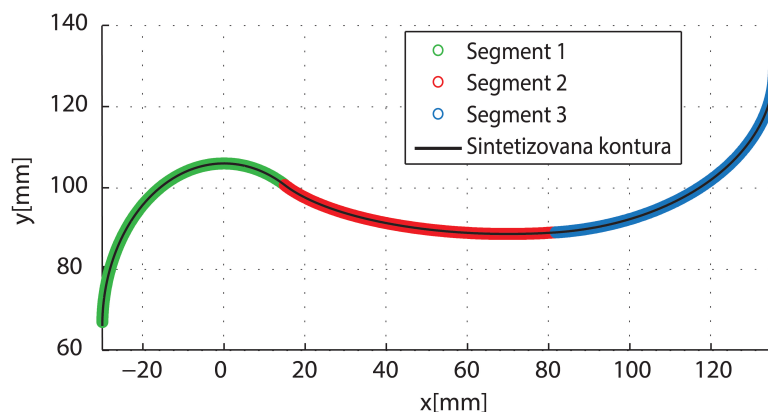
Kako se uniformno strukturani oblak tačaka sastoji iz dva niza ukrštenih skeniranih linija na svaku od ovih linija se može primeniti opisana metoda segmentacije eliptičnih regiona i to u pravcu jedne, a zatim i druge ose. Primenom metode na sve linije duž jedne ose oblaka dobija se segmentacija skenirane površine u jednom pravcu, odnosno oblak tačaka se u posmatranom pravcu segmentira na određeni broj regiona. Kada se zatim procesiraju i skenirane linije duž druge ose skenirana površ se segmentira i u tom pravcu. Ukrštanjem rezultata segmentacije po obe ose oblaka, odnosno traženjem preseka izdvojenih skupova tačaka po jednoj sa onim po drugoj osi dobijaju se konačni segmenti koji aproksimiraju regione od kojih se površina sastoji.

6.3. Verifikacija metode simulacijom

Da bi se proverile performanse predložene metode segmentacije i prepoznavanja kvadraka u strukturiranom oblaku tačaka, algoritam je podvrgnut testiranju u simuliranim uslovima. Proces segmentacije eliptičnih regiona prvo je proveren na primerima procesiranja dve konture koje su sačinjene od eliptičnih segmenata, a nakon toga, celokupna metoda prepoznavanja kvadraka je testirana u slučajevima nekoliko simuliranih oblaka tačaka. Ovi strukturirani oblaci su kreirani tako da sadrže delove različitih kvadraka od interesa i za njihovo sintetizovanje su korišćeni CAD modeli u STL formatu nad kojima je izvršen proces strukturiranja predstavljen u poglavlju 4. Raster strukturiranja za obe ose je u slučaju svih oblaka podešen na 0,5 mm. Takođe, da bi se približili realnosti, svim sintetizovanim oblacima je dodat šum od 120 dB SNR. Svi rezultati i zapažanja pri upotrebi algoritma prepoznavanja su kreirani, strukturirani i predstavljeni u nekoliko posebnih studija slučajeva koje su prezentovane u nastavku.

6.3.1. Segmentacija i prepoznavanje elipsi u konturi

U prvom slučaju proverene su mogućnosti algoritma segmentacije procesiranjem dve sintetizovane konture koje sadrže po tri eliptična segmenta G1 kontinualne geometrije. Treba napomenuti da tačke na konturama nisu ekvidistantne.



Slika 6.4: Prva sintetizovana kontura sa tri eliptična segmenta.

Prva sintetizovana kontura prikazana je na slici 6.4. Radi bolje ilustracije u tabeli 6.1 prikazani su

parametri jednačina koje matematički opisuju elipse koje sačinjavaju ovu konturu. Takođe, date su i tačne pozicije prelaza između segmenata predstavljene rednim brojem tačke u nizu. Pri procesiranju ove sintetizovane konture bez prisustva šuma algoritam segmentacije određuje prelazne tačke bez devijacije (izdvojeni segmenti na slici 6.4). Međutim, radi provere robusnosti algoritma u prisustvu različitog nivoa šuma, kontura je procesirana u dva slučaja: sa dodatim šumom nivoa 120 dB i sa 100 dB. Rezultati segmentacije u oba pomenuta slučaja dati su u tabeli 6.2. Upoređivanjem ove tabele sa tabelom 6.1 moguće je uvideti da vrednosti prepoznatih granica u prisustvu šuma neznatno odudaraju od idealnih. Maksimalna devijacija je na konturi sa dodatim SNR od 100 dB i iznosi 7 uzoraka (tačaka) što npr. predstavlja 0,6 mm duž x ose konture. Ova devijacija se s obzirom na visok nivo dodatog šuma može smatrati vrlo prihvatljivom.

Da bi postignuti zadovoljavajući rezultati segmentacije bili potvrđeni procesirana je još jedna sintetizovana kontura prikazana na slici 6.5. Tačke prelaza između segmenata kao i parametri elipsi su prikazani u tabeli 6.3. Primena algoritma segmentacije nad ovom konturom za slučaj bez šuma kao i u prethodnom slučaju daje dobre rezultate, odnosno identifikuje upravo potrebne tačke prelaza sintetizovane konture. Da bi se dodatno proverile performanse na isti način kao i u prethodnom slučaju i ova sintetizovana kontura je kontaminirana šumom nivoa 120 i 100 dB. Nakon primene postupka segmentacije konture sa slike 6.5 pri oba dodata nivoa šuma dobijeni su rezultati prikazani u tabeli 6.4. Može se uočiti da dobijene pozicije prelaza ponovo neznatno odstupaju od idealnih. Maksimalno odstupanje u ovom slučaju se može uočiti kod prvog prelaza u slučaju šuma nivoa 100 dB i iznosi 13 uzoraka. Međutim, ukoliko ovaj rezultat pogledamo kroz prizmu odstupanja u milimetrima po x osi, ono iznosi 0,4 mm što se smatra vrlo dobrim u prisustvu šuma visokog nivoa.

Tabela 6.1: Prva sintetizovana kontura - parametri i granice segmenata.

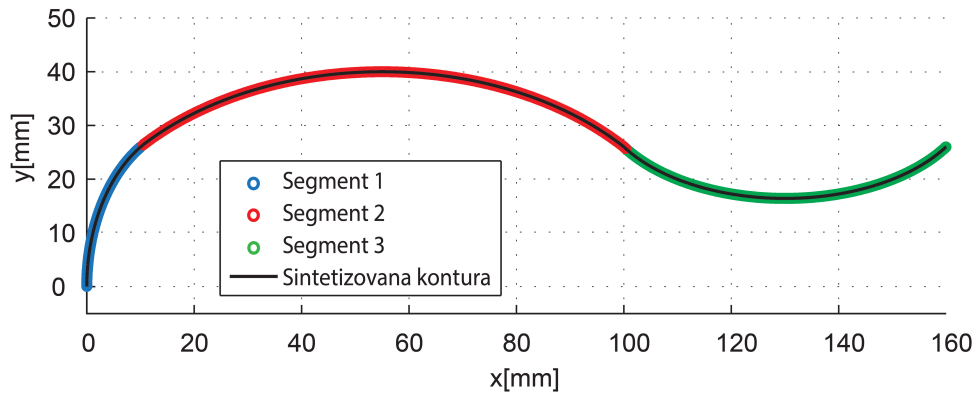
Br. seg.	Parametri elipse $a_i \times 10^2, i = 1, \dots, 6$	Redni broj tačke	
		Početak segmenta	Kraj segmenta
1.	[0,111 0,000 0,062 0,000 -8,250 172,250]	1	1201
2.	[0,028 0,000 0,250 -3,889 -54,318 2986,500]	1202	1973
3.	[0,028 0,000 0,062 -4,195 -16,100 1095,330]	1974	2820

Tabela 6.2: Rezultati segmentacije prve sintetizovane konture.

Br. seg.	Redni broj tačke			
	Kontura sa SNR 120 dB		Kontura sa SNR 100 dB	
	Početak segmenta	Kraj segmenta	Početak segmenta	Kraj segmenta
1.	1	1202	1	1199
2.	1203	1966	1200	1979
3.	1967	2820	1980	2820

Tabela 6.3: Druga sintetizovana kontura - parametri i granice segmenata.

Br. seg.	Parametri elipse $a_i \times 10^2, i = 1, \dots, 6$	Redni broj tačke	
		Početak segmenta	Kraj segmenta
1.	[0,250 0,000 0,118 -10,000 0,000 0,001]	1	617
2.	[0,028 -0,002 0,062 -3,137 0,127 -13,725]	618	1599
3.	[0,082 0,000 0,381 -21,224 -26,806 1751,410]	1560	2785



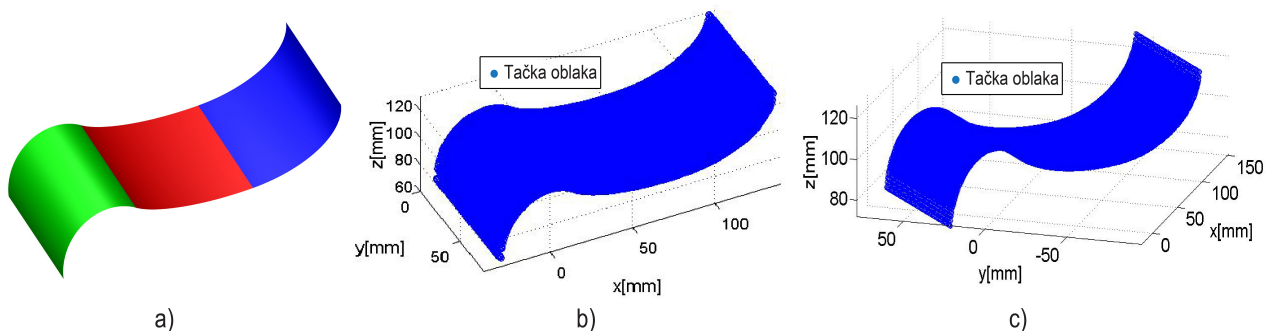
Slika 6.5: Druga sintetizovana kontura sa tri eliptična segmenta.

Tabela 6.4: Rezultati segmentacije druge sintetizovane konture.

Br. seg.	Redni broj tačke			
	Kontura sa SNR 120 dB		Kontura sa SNR 100 dB	
	Početak segmenta	Kraj segmenta	Početak segmenta	Kraj segmenta
1.	1	617	1	604
2.	618	1601	605	1601
3.	1602	2785	1602	2785

6.3.2. Segmentacija i prepoznavanje eliptičnih cilindara

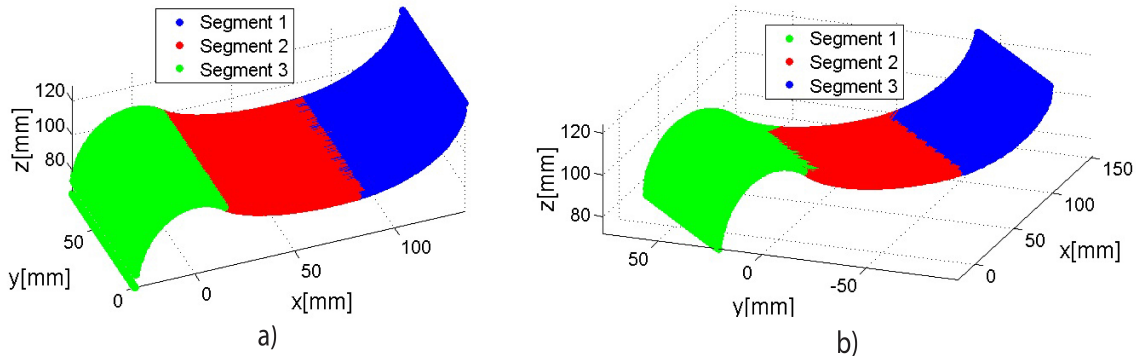
U ovoj studiji slučaja prezentovane su performanse predložene metode za potrebe segmentacije i prepoznavanja eliptičnih cilindara. U te svrhe iskorišćena su dva sintetizovana oblaka tačaka koji reprezentuju istu G1 kontinualnu površinu koja je sastavljena od tri segmenta eliptičnih cilindara (slika 6.6). Jedina razlika između ova dva oblaka je u simulaciji pravca skeniranja. Pošto se eliptični cilindri u oba strukturirana oblaka reprezentuju eliptičnim segmentima u skeniranim linijama, nad oba oblaka moguće je upotrebiti predloženi algoritam segmentacije. Prvi strukturirani oblak (slika 6.6(b)) sadrži skenirane linije u pravcu upravnom na ose eliptičnih cilindara, dok je u drugom slučaju izvršeno simuliranje skeniranja pod uglom od 45° u odnosu na pomenute ose. Naravno oba oblaka su kontaminirana šumom sa SNR 120 dB.



Slika 6.6: Površina sačinjena od eliptičnih cilindara: a) 3D model; b) sintetizovani oblak dobijen longitudinalnim strukturiranjem (skeniranjem); c) sintetizovani oblak dobijen strukturiranjem pod uglom.

Rezultati segmentacije skeniranih linija oblaka sa slika 6.6(a) i 6.6(b) duž njihovih x osa su prikazani na slikama 6.7(a) i 6.7(b), respektivno. Algoritam je u ovom slučaju izvršio pravilnu segmentaciju na tri oblasti koje predstavljaju tri različita eliptična cilindra. Radi dalje provere performansi algoritma, izvršena je i segmentacija duž y ose. Interesantno je istaći i da, u slučaju oblaka sa slike 6.6(a),

skenirane konture u pravcu y ose reprezentuju prave linije. Predloženi algoritam je u tom slučaju pokazao dobre performanse i izvršio prepoznavanje samo jednog segmenta u svakoj skeniranoj liniji tj. nije prepoznao nikakve granične tačke u pravolinijskim konturama. Ova osobina algoritma proizilazi iz činjenice da se prava može opisati i kao specijalni slučaj jednačine kvadrrike u implicitnom obliku kod koje su parametri $a_1 = a_2 = a_3 = 0$.



Slika 6.7: Grafički rezultati segmentacije površine sačinjene od eliptičnih cilindara segmentirane duž x ose: a) oblak tačaka sa slike 6.6(b); b) oblak tačaka sa slike 6.6(c).

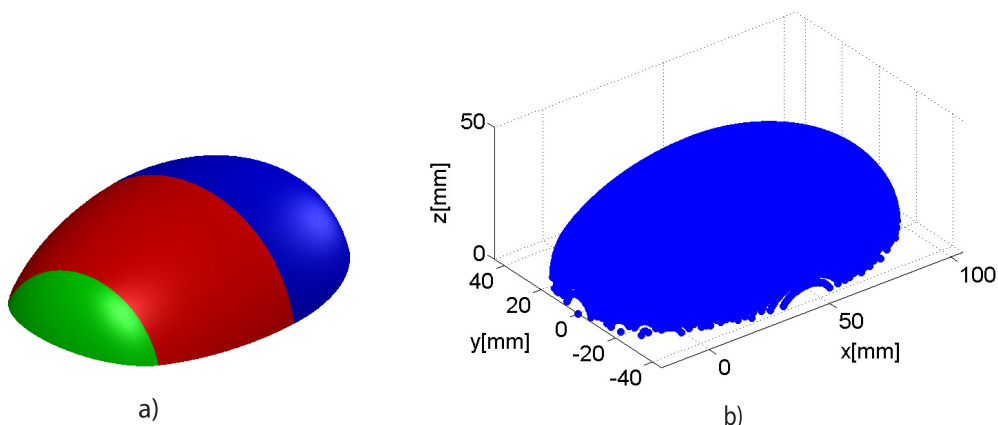
U tabeli 6.5 predstavljeni su estimirani parametri prepoznatih eliptičnih cilindara u slučaju segmentacije oba oblaka predstavljena na slici 6.7. Pokazane vrednosti parametara su dobijene regresijom metodom najmanjih kvadrata kroz tačke izdvojenih segmenata oblaka. Upoređivanjem dobijenih parametara u oba slučaja može se zaključiti da je predloženi algoritam pokazao očekivano dobre rezultate za sve segmente. Devijacija je manja od 1% osim u slučaju parametra a_7 kod segmenta broj 2. U slučaju oblaka sa slike 6.7(b) estimirani parametri eliptičnih cilindara su u saglasnosti sa pravcem skeniranja. Naime, vrednosti parametara a_1 i a_2 su jednake što indikuje da je ugao skeniranja bio 45° . Takođe, ovi parametri (a_1 i a_2) poseduju i dva puta veće vrednosti od istih parametara segmenata u slučaju segmentacije oblaka sa pravcem skeniranja upravnom na ose eliptičnih cilindara - $\sin^2(45^\circ) = 1/2$. Sa druge strane, parametar a_3 je isti kod svih adekvatnih segmenata oba oblaka.

Tabela 6.5: Estimirani parametri prepoznatih eliptičnih cilindara.

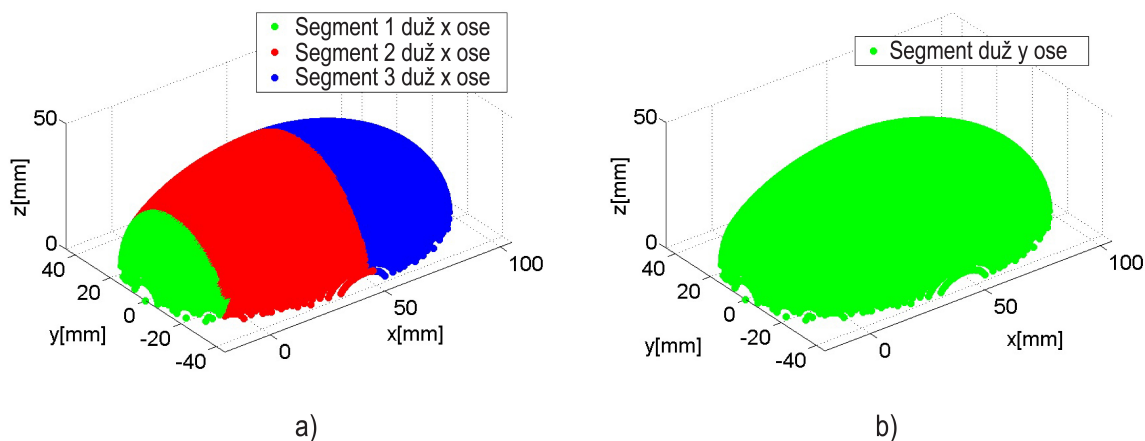
Oblak tačaka	Jednačina kvadrrike:	
	$a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5yz + a_6xz + a_7x + a_8y + a_9z + a_{10} = 0$	Parametri $a_i \times 10^2, i = 1, \dots, 10$
Slika 6.7(a)	Seg. 1	[0,111 -0,000 0,062 -0,000 0,000 0,000 -0,003 -0,000 -8,249 172,110]
	Seg. 2	[0,027 -0,000 0,250 -0,000 -0,000 0,002 -4,043 0,006 -54,229 2976,380]
	Seg. 3	[0,028 0,000 0,062 0,000 -0,000 0,000 -4,197 0,000 -16,105 1095,660]
Slika 6.7(b)	Seg. 1	[0,050 0,050 0,062 -0,100 0,004 -0,003 0,250 -0,353 -8,689 218,930]
	Seg. 2	[0,015 0,015 0,250 -0,030 0,003 -0,004 -2,662 2,754 -54,951 3055,720]
	Seg. 3	[0,014 0,014 0,062 -0,028 -0,000 0,000 -2,967 2,967 -16,101 1095,380]

6.3.3. Segmentacija i prepoznavanje elipsoida

Treća studija slučaja prezentuje rezultate upotrebe algoritma za prepoznavanje elipsoida. Elipsoidi nisu kvadrrike koja se često sreće u mašinskom inženjerstvu. Sa druge strane, sferoidi kao specijalni slučaj elipsoida se češće sreću u praksi. Zbog navedenog, za dalje testiranje predloženog algoritma je iskorišćen sintetizovan oblak koji reprezentuje G1 kontinualnu površinu sastavljenu od tri sferoida (slika 6.8). Oblak je generisan strukturiranjem (tehnika iz poglavlja 4) CAD modela dobijenog rotacijom dizajnirane konture oko x ose. I u ovom slučaju dobijeni oblak je kontaminiran šumom nivoa SNR 120 dB.



Slika 6.8: Površina sačinjena od sferoida: a) 3D model; b) sintetizovani oblak tačaka.



Slika 6.9: Grafički rezultati segmentacije površine sačinjene od sferoida: a) segmentacija duž x ose; b) segmentacija duž y ose.

Tabela 6.6: Sintetizovani i estimirani parametri prepoznatih sferoida.

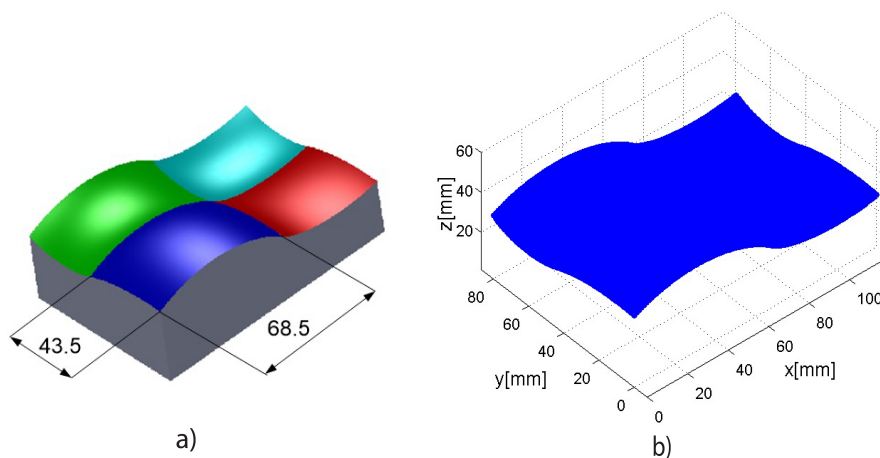
		Jednačina kvadratika: $a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5yz + a_6xz + a_7x + a_8y + a_9z + a_{10} = 0$ Parametri $a_i \times 10^2, i = 1, \dots, 10$									
Sintetizovani parametri	Seg. 1	[0,2500	0,1100	0,1100	0,0000	-0,0000	0,0000	0,0000	0,0000	0,0000	-100,0000]
	Seg. 2	[0,0178	0,0625	0,0625	-0,0000	0,0000	-0,0000	-1,6495	-0,0000	-0,0000	-61,7384]
	Seg. 3	[0,0278	0,0625	0,0625	0,0000	0,0000	-0,0000	-2,5000	-0,0000	0,0000	-43,7500]
Estimirani parametri	Seg. 1	[0,2504	0,1100	0,1100	0,0000	-0,0000	0,0000	0,0070	0,0000	0,0010	-100,0000]
	Seg. 2	[0,0179	0,0625	0,0627	-0,0000	0,0000	-0,0000	-1,6517	-0,0000	-0,0042	-61,7384]
	Seg. 3	[0,0281	0,0625	0,0625	0,0000	0,0000	-0,0000	-2,5000	-0,0000	0,0000	-43,7520]

Rezultati segmentacije skeniranih linija ovog oblaka duž oba pravca su grafički prikazani na slici 6.9. Njihovim posmatranjem može se zaključiti da je algoritam duž x ose izvršio ispravnu segmentaciju na tri regiona dok procesiranjem duž y ose nije izvršio nikakve podele što je i bilo očekivano. Estimirani parametri prepoznatih sferoida su, zajedno sa projektovanim (sintetizovanim), prikazani u tabeli 6.6. Njihovim upoređivanjem može se zaključiti da je algoritam dao vrlo ubedljive rezultate. Naime, vrednosti estimiranih parametara su vrlo bliske projektovanim vrednostima.

6.3.4. Segmentacija i prepoznavanje eliptičnih torusa

Nakon testiranja prepoznavanja eliptičnih cilindara i elipsoida, performanse algoritma su proverene u slučaju sintetizovanog oblaka tačaka koji predstavlja G1 kontinualnu površinu sačinjenu od četiri segmenata opštih eliptičnih torusa. Kompjuterski model površine objekta, koji je iskorišćen za generisanje oblaka tačaka, kreiran je provlačenjem profila koji se sastoji od dva eliptična regiona po putanji koja takođe sadrži dva eliptična regiona. Promena geometrije objekta je izražena po sve tri ose prostora i treba naglasiti da je ostvarena G1 kontinualnost površine. 3D model objekta zajedno sa pozicijama prelaza između njegovih površina, kao i sintetizovani oblak tačaka njegove površine (kontaminiran šumom nivoa 120 dB) prikazani su na slici 6.10.

Na slici 6.11 dat je grafički prikaz rezultata segmentacije primenom predložene metode, po obe ose pojedinačno, kao i konačna segmentacija na četiri grupe tačaka odnosno na četiri izdvojena segmenta od kojih se površina sastoji. Sve skenirane linije po jednoj osi su procesirane identičnim pragovima za segmentaciju i grupisanje, a u cilju održanja visokog nivoa automatizma samog procesa segmentacije.

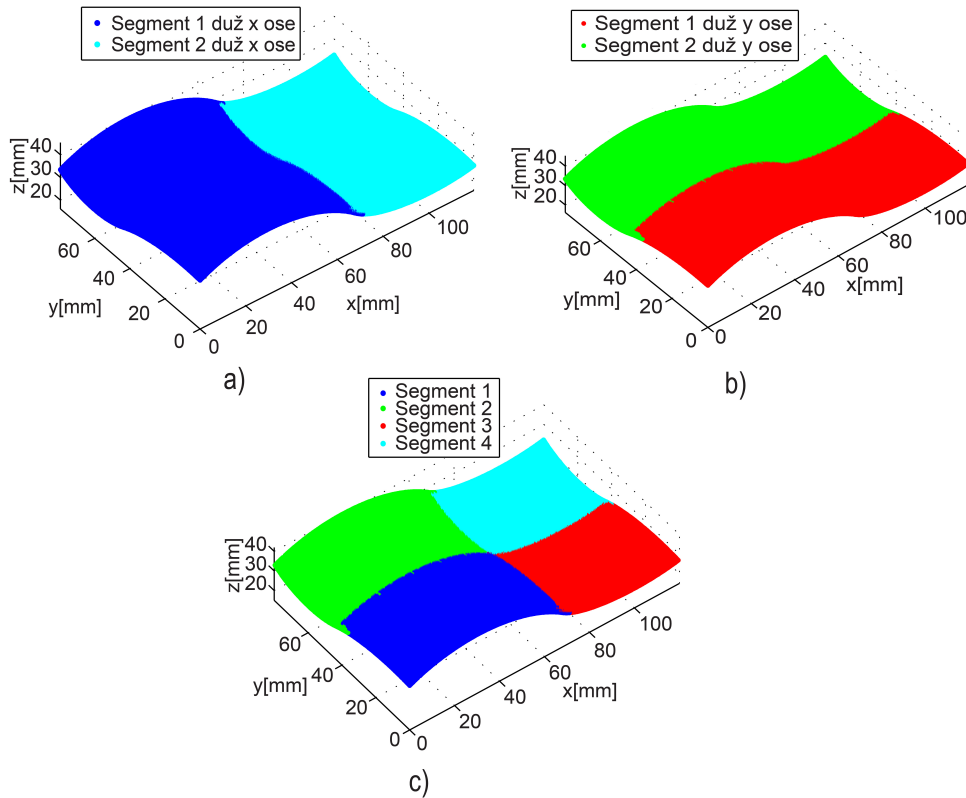


Slika 6.10: Površina sačinjena od torusa: a) 3D model; b) sintetizovani oblak tačaka.

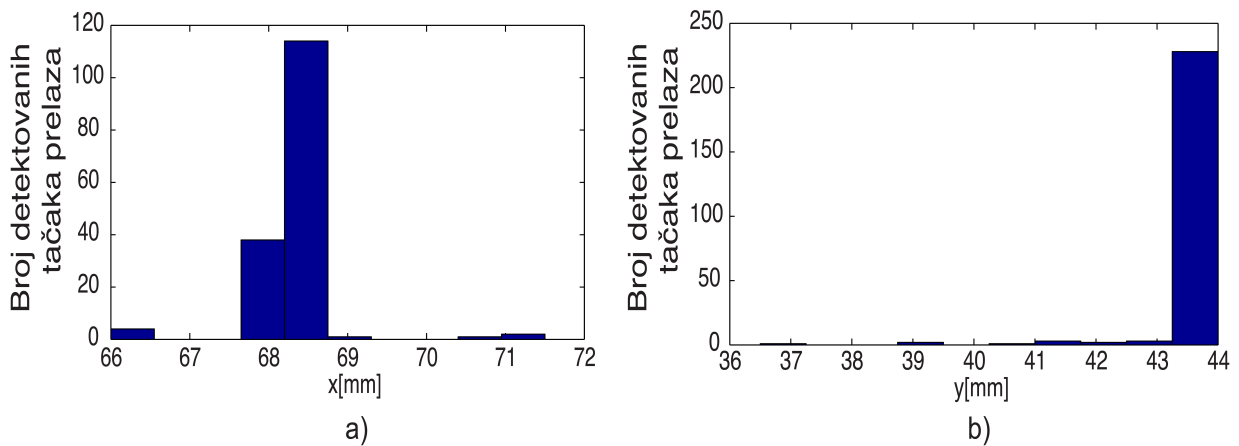
Posmatranjem grafičkih rezultata na slici 6.11 može se zapaziti da postoji izvesna devijacija u detektovanim tačkama prelaza u skeniranim konturama međusobno. Ova devijacija se može zapaziti, pre svega, na krajnjim konturama pri segmentaciji linija duž x ose i posledica je prisutnog šuma u oblaku. Međutim, ukoliko se posmatraju detektovane pozicije prelaza u svim konturama ponaosob, može se primetiti značajna preciznost u velikom broju slučajeva segmentacije. Ovi indikatori su prikazani upotrebom histograma pozicija tačaka prelaza u svakoj konturi. Izgledi ovih histograma su dati na slici 6.12. Na njima se može videti da je veliki broj kontura ispravno segmentiran, s obzirom da su tačne pozicije projektovanih prelaza 43,5 mm duž x ose i 68,5 mm duž y ose.

6.4. Eksperimentalna verifikacija metode

U cilju dalje verifikacije predstavljenog algoritma segmentacije iskorišćen je realni oblak tačaka test objekta koji je takođe korišćen u poglavlju o redukciji broja tačaka i čiji je izgled bio predstavljen na slici 5.14(a). Deo je obrađen na 3-osnom obradnom centru i sadrži veći broj površi drugog reda uključujući i ravne površi. Prilikom skeniranja ranije već pomenutim *ATOS Compact Scan 5M* skenerom [24] kreirana je 3D mreža trouglova. Za potrebe testiranja algoritma izdvojena je oblast objekta koja reprezentuje površinu sastavljenu od delova pet cilindara (obeležena crveno na slici 6.13(a)) i na već opisan način generisan je njen strukturani oblak tačaka pri čemu je izabran gust raster od 0,05 mm duž obe ose. Treba napomenuti da cilindri predstavljaju poseban slučaj eliptičnih cilindara zbog čega je moguće iskoristiti predloženi algoritam za njihovu segmentaciju u strukturanim oblacima. Takođe, važan aspekt je da su cilindri primitiv koji se pored ravni najčešće sreće u mašinskom inženjerstvu.



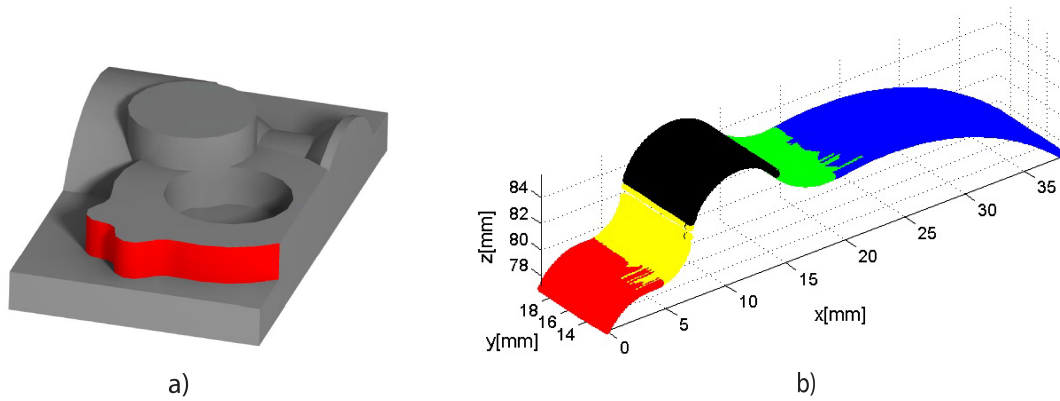
Slika 6.11: Grafički rezultati segmentacije površine sačinjene od torusa: a) segmentacija duž x ose; b) segmentacija duž y ose; c) konačni rezultati segmentacije.



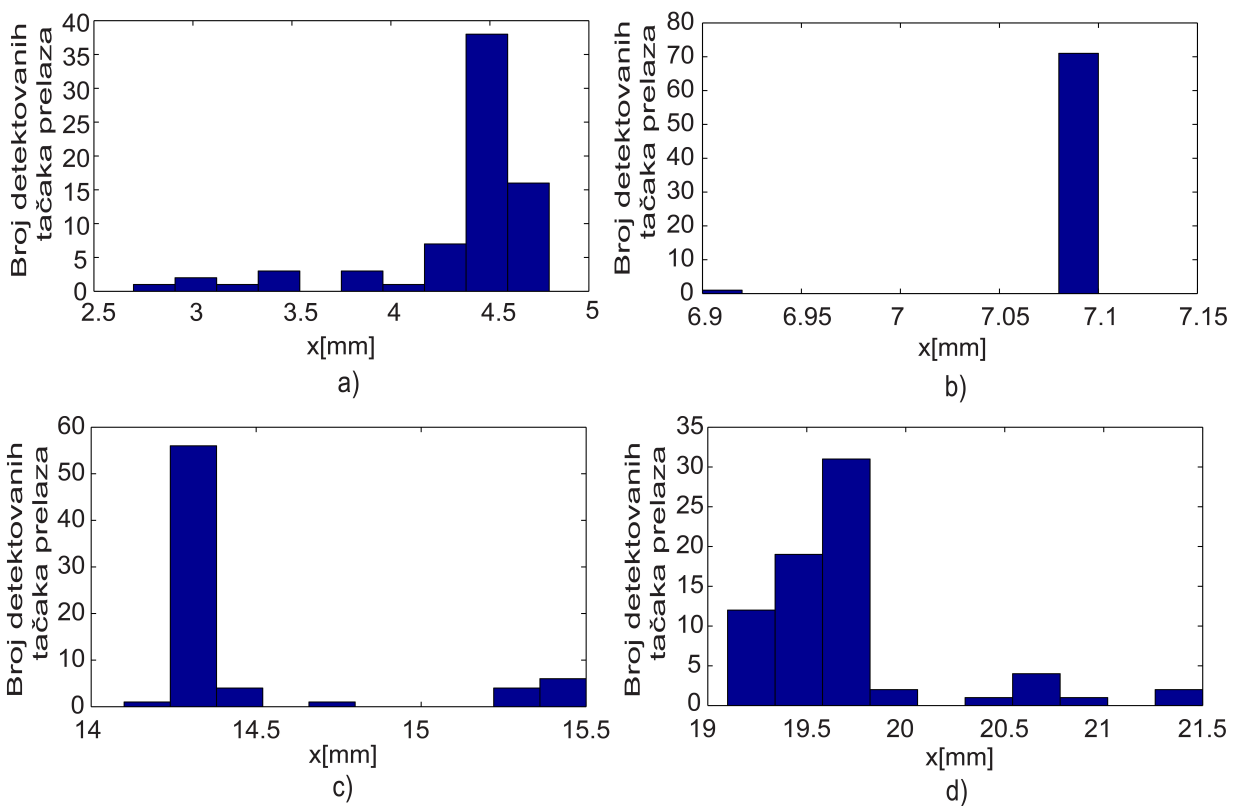
Slika 6.12: Histogrami detektovanih prelaznih tačaka oblaka sa slike 6.11 : a) prelazne tačke duž x ose; b) prelazne tačke duž y ose.

Tabela 6.7: Procenjene vrednosti parametara prepoznatih cilindara.

Jednačina cilindra:	
$a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5yz + a_6xz + a_7x + a_8y + a_9z + 1 = 0$	
Parametri $a_i \times 10^2, i = 1, \dots, 9$	
Segment 1	[0,003 0,000 0,017 0,000 0,000 -0,016 1,271 -0,013 -2,572]
Segment 2	[0,015 0,000 0,014 0,000 0,000 0,013 -1,105 -0,006 -2,340]
Segment 3	[0,015 0,000 0,015 0,000 0,000 0,002 -0,504 0,006 -2,396]
Segment 4	[0,000 0,000 0,013 0,000 0,000 0,009 -0,687 -0,047 -2,292]
Segment 5	[0,000 0,000 0,012 0,000 0,000 0,007 -0,589 -0,028 -2,195]



Slika 6.13: Segmentacija realnog oblaka koji reprezentuje površinu sačinjenu od cilindara: a) površina od interesa označena na test objektu sa slike 5.14(a); b) rezultati segmentacije izdvojene površine duž x ose.



Slika 6.14: Histogrami detektovanih prelaza sa slike 6.13(b): a) između segmenata 1 i 2; b) između segmenata 2 i 3; c) između segmenata 3 i 4; d) između segmenata 4 i 5.

Rezultati segmentacije skeniranih linija realnog oblaka odabrane površine duž x ose su grafički predstavljeni na slici 6.13(b). Vrednosti parametara segmentiranih površina koje su procenjene metodom najmanjih kvadrata su prikazane u tabeli 6.7. Procenjene vrednosti parametara ukazuju na to da se radi o profilima koji su ekstrudirani duž y ose jer su vrednosti parametara a_2 , a_4 , a_5 i a_8 bliske nuli. Slika 6.14 prikazuje histograme raspodele detektovanih tačaka prelaza između pojedinih segmenata.

Na osnovu predstavljenih rezultata segmentacije realnog oblaka može se zaključiti da predstavljeni algoritam ima potencijal za pravilnu segmentaciju i prepoznavanje cilindara pogotovo ako se u obzir uzmu estimirani parametri prepoznatih cilindara. Međutim, izvesno je da poseduje određena ograničenja u smislu potrebnog finog podešavanja vrednosti pragova algoritma od strane korisnika što u prisustvu šuma može predstavljati poteškoću i u velikoj meri uticati na rezultate. Zbog toga

je predstavljeni algoritam unapređen razvijenim dopunskom metodom automatskog izbora pragova koja se temelji na upotrebi tehnika mašinskog učenja. Njegova upotreba obezbeđuje prevazilaženje pomenutih problema određivanja pragova i obezbeđuje generisanje još tačnijih rezultata prepoznavanja potrebnih geometrijskih primitiva u struktuiranim oblacima. Osnovi ove dopunske metode kao i rezultati njene upotrebe biće predstavljeni u poglavlju 8.

Na kraju je potrebno istaći činjenicu da, iako teorijska osnova dozvoljava, eventualna upotreba predstavljenog algoritma segmentacije i prepoznavanja u slučajevima gde prelazi između kvadrata nisu prave linije nije do kraja istražena. Takođe, potrebno je naglasiti da su delovi rezultata istraživanja iz ovog poglavlja od strane autora predstavljeni i u ranije objavljenim naučnim radovima [131, 132, 133].

7. Prepoznavanje cilindara i ravni iz nestruktuiranih oblaka tačaka

U prethodnom poglavlju predstavljena je metoda prepoznavanja određenih vrsta površi drugog reda u struktuiranim oblacima tačaka. Predložena metoda je za svaku izdvojenu površ generisala parametre opšte jednačine kvadrike koji su dobijeni regresijom metodom najmanjih kvadrata nad skupovima tačaka iz prethodno izvršene segmentacije oblaka. Sami tipovi primitiva, kao i specifični parametri koji bi ih opisivali mogu se indirektno izračunati iz pomenute opšte jednačine površi drugog reda. Zbog toga, prepoznata je potreba razvoja robusnije metode prepoznavanja primitiva čijim korišćenjem bi bilo moguće direktno generisanje parametara specifičnih za svaki izdvojeni tip primitiva posebno. Takođe, izabrano je da se istraživanja na ovu temu usmere na nestruktuirane oblake tačaka kako bi se obezbedila veća mogućnost generalizacije metode.

Obrada nestruktuiranih oblaka, ali i potreba generisanja parametara prema svakom tipu primitiva posebno, znatno usložnjavaju proces ekstrakcije i zahtevaju tačno određivanje vrste primitiva veoma rano u samom procesu prepoznavanja. Ukoliko se tip ne prepozna adekvatno, može doći do tzv. konfuzije modela u prepoznavanju i ekstrakcije pogrešnih primitiva iz oblaka. Jasno je da je u opštem slučaju, gde postoji veoma veliki broj sličnih tipova geometrijskih primitiva koji se mogu pojaviti u strukturi nekog objekta, teško sprovesti robusnu ekstrakciju bez grešaka u prepoznavanju.

Zbog navedenog, pogodno je uvesti određena ograničenja metode u smislu tipova primitiva koji mogu biti prepoznati. Kako je tema ovog rada usmerena na primenu tehnika 3D digitalizacije u mašinskoj industriji, metoda prepoznavanja koja se prezentuje u ovom poglavlju je usmerena na dve vrste površina koje se mogu najčešće sresti u morfologiji mašinskih delova. Reč je o cilindrima i ravnima.

Kao polazni zahtevi razvoja metode prepoznavanja u ovom slučaju mogu se istaći potrebe prepoznavanja cilindara i ravni u realnim nestruktuiranim oblacima tačaka koji reprezentuju određene mašinske elemente. Potrebno je prepoznati sve cilindre i ravni od kojih se struktura objekta sastoji, razvrstati ih u posebne grupe, ali i razdvojiti od ostalih regiona koji mogu reprezentovati neke druge tipove površi koje ne pripadaju navedenim. Takođe, za sve prepoznate regione koji predstavljaju cilindre potrebno je generisati njihove osnovne parametre kao što su radijus cilindra i pravac njegove ose, dok je za ravni potrebno odrediti parametre njihovih jednačina.

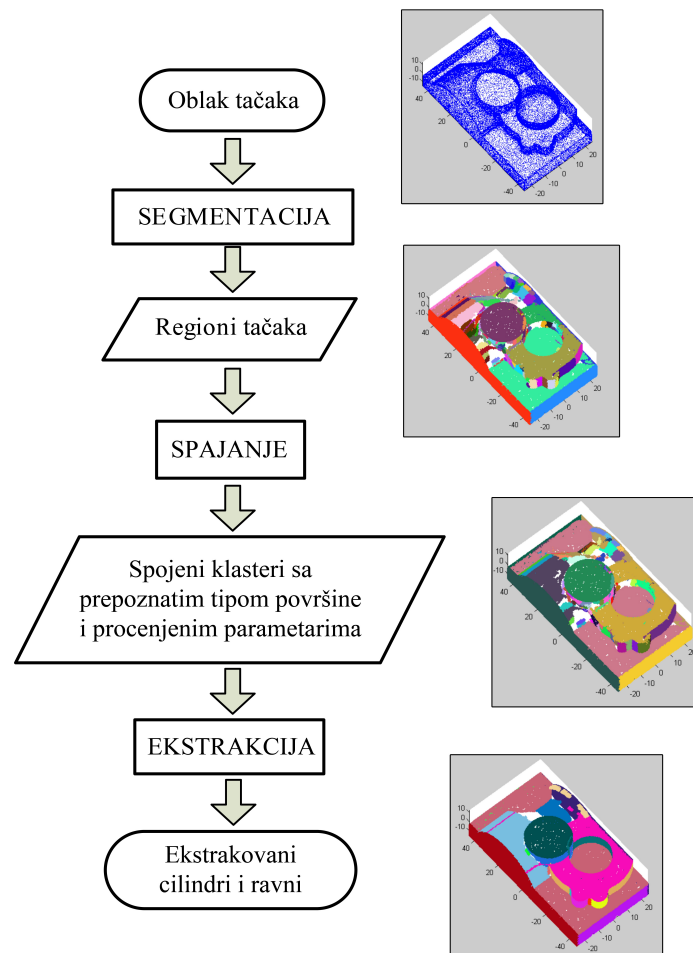
7.1. Struktura metode i algoritma

U skladu sa navedenim zahtevima izvršeno je kreiranje metode koja simultano procesira cilindre i ravni i ujedno identifikuje segmente koji reprezentuju neke druge tipove površi (algoritam ne prepoznaje egzaktno tipove ostalih površi). Kao i svi razvijeni algoritmi prepoznavanja predstavljeni u ovom radu i ovaj spada u grupu onih koji su zasnovani na prethodnoj segmentaciji oblaka tačaka. Segmentacija se sprovodi direktno na sirovom oblaku čime se vrši njegovo razdvajanje na klastere tačaka različite po svojim geometrijskim karakteristikama. Algoritam dalje obrađuje ove klastere razvrstavajući ih u tri grupe (cilindre, ravni i ostale površine), a zatim vrši robusnu estimaciju parametara prepoznatih cilindara i ravni. Na kraju, da bi se dobili što precizniji rezultati prepoznavanja, vrši se završni proces ekstrakcije prepoznatih primitiva. U skladu sa navedenim, opisani algoritam prepoznavanja se može podeliti u tri glavne faze (slika 7.1):

- segmentacija oblaka tačaka,
- spajanje presegmentiranih regiona i estimacija parametara površina, i
- ekstrakcija cilindara i ravni iz oblaka.

U prvoj fazi procesa, kao što je već objašnjeno, oblak tačaka se deli na određeni broj različitih klastera. U prisustvu relativno visokog nivoa šuma i većeg broja tačaka koje ne pripadaju geometriji skeniranog objekta proces segmentacije je podlozan presegmentaciji. Iz ovog razloga je uvedena druga

faza odnosno spajanje presegmentiranih regiona koje je objedinjeno sa postupcima prepoznavanja cilindara i ravni i estimacije njihovih parametara. U poslednjoj (trećoj) fazi se, na osnovu prepoznatih regiona i procenjenih parametara njihovih površi, vrši direktna ekstrakcija cilindara i ravni iz polaznog oblaka tačaka čime se dobijaju precizniji rezultati, ali i odstranjuju eventualno prepoznati beznačajno mali segmenti površina. Takođe se, na samom kraju, vrši i ponovno izračunavanje parametara površi čime se dobijaju još precizniji konačni rezultati prepoznavanja. U nastavku će u posebnim sekcijama biti detaljno objašnjene sve tri pomenute faze metode.



Slika 7.1: Struktura predložene metode prepoznavanja cilindara i ravni iz nestrukturiranih oblaka tačaka.

7.2. Segmentacija

Algoritam segmentacije koji se koristi u metodi prepoznavanja cilindara i ravni spada u grupu algoritama sa primenom strategije rasta regiona koja je pomenuta u prethodnim poglavljima. Sam kriterijum za rast regiona u ovom poglavlju je odabran nakon intenzivnih istraživanja čiji su pojedini rezultati prezentovani u radovima [134, 135, 136].

Za potrebe segmentacije oblaka koji reprezentuju cilindrične objekte u [134], pri inicijalnom širenju regiona izabran je uslov singularnosti informacione matrice pri regresiji kvadrata metodom najmanjih kvadrata [137]. U ovom slučaju pokazalo se da ovakva segmentacija poseduje značajne performanse, ali treba istaći da je testirana samo na ograničenom broju oblaka.

U radovima [135, 136] se, sa druge strane, pristupilo nešto kompleksnijem rešenju. Za inicijalno širenje regiona iskorišćen je recipročni faktor uslovljenosti informacione matrice pri procesu regresije elipsoida metodom najmanjih kvadrata iz [138]. Segmentacija je dizajnirana tako da se region širi

tačku po tačku na osnovu pomenutog kriterijuma, ali upotrebom tri različite strategije širenja i biranja inicijalnih tačaka. Ovakav način je poboljšao performanse metode, ali je računski i vremenski opteretio izvršavanje algoritma.

Nastavak istraživanja na ovu temu kao i dodatna iskustava u upotrebi predstavljenih algoritama segmentacije dovela su do zaključka da veći benefit može doneti manje robusna, ali i računski manje zahtevna metoda inicijalnog širenja. Na ovaj način bi se obezbedila dovoljno dobra inicijalna segmentacija nestruktuiranih oblaka koja bi kasnijim koracima algoritma mogla biti poboljšana, uobličena i sa uspehom korišćena za prepoznavanje cilindara i ravni. Na osnovu ovih zaključaka formiran je i proces segmentacije koji je predstavljen u nastavku.

Formiranje klastera počinje od izabranih inicijalnih tačaka dodavanjem okolnih u dve faze. Prva faza predstavlja inicijalno širenje regiona. Proces širenja sadrži dva koraka. U konkretnom slučaju izabrano je da se inicijalni region širi tačku po tačku na osnovu osobina singularnosti informacione matrice sistema generisane tokom regresije opšte površine drugog reda metodom najmanjih kvadrata (cilindri takođe spadaju u površine drugog reda). Na ovaj način formira se inicijalni klaster tačaka (određene zadate veličine) na osnovu koga se mogu proceniti parametri tekuće površine i, računski manje zahtevnim postupkom, izdvojiti sve ostale tačke oblaka koje pripadaju toj površini. Ovo dopunsko izdvajanje ujedno predstavlja pomenuti drugi korak procesa segmentacije.

Kako bi se postupak bolje objasnio potrebno je krenuti od jednačine opšte kvadrike u implicitnom obliku:

$$a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5yz + a_6xz + a_7x + a_8y + a_9z + a_{10} = 0, \quad (7.1)$$

gde su sa $a_i, i = 1, \dots, 10$ obeleženi parametri površine, dok (x, y, z) predstavljaju koordinate tačke na toj površini. Ovu jednačinu možemo zapisati i u matricnoj formi kao:

$$\mathbf{x} \cdot \mathbf{a} = 0, \quad (7.2)$$

pri čemu je $\mathbf{x} = [x^2 \ y^2 \ z^2 \ xy \ yz \ xz \ x \ y \ z \ 1]$ vektor promenljivih, dok je vektor parametara elipsoida $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_{10}]^T$.

Estimacija članova vektora \mathbf{a} se može izvršiti metodom najmanjih kvadrata. U ovom slučaju koristimo opštu regresiju površi drugog reda (kvadrike) i problem optimizacije možemo predstaviti kao:

$$\begin{aligned} &\text{minimizirati } \|\mathbf{D}\mathbf{a}\|^2 \\ &\text{na osnovu } \mathbf{a}^T \mathbf{C}\mathbf{a} = 1, \end{aligned} \quad (7.3)$$

gde je matrica sistema \mathbf{D} sa n tačaka oblika:

$$\mathbf{D} = \begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & x_1y_1 & y_1z_1 & x_1z_1 & x_1 & y_1 & z_1 & 1 \\ x_2^2 & y_2^2 & z_2^2 & x_2y_2 & y_2z_2 & x_2z_2 & x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & y_n^2 & z_n^2 & x_ny_n & y_nz_n & x_nz_n & x_n & y_n & z_n & 1 \end{bmatrix}, \quad (7.4)$$

dok matrica \mathbf{C} predstavlja matricu 10×10 sa svim elementima jednakim nuli osim $C(10, 10) = 1$.

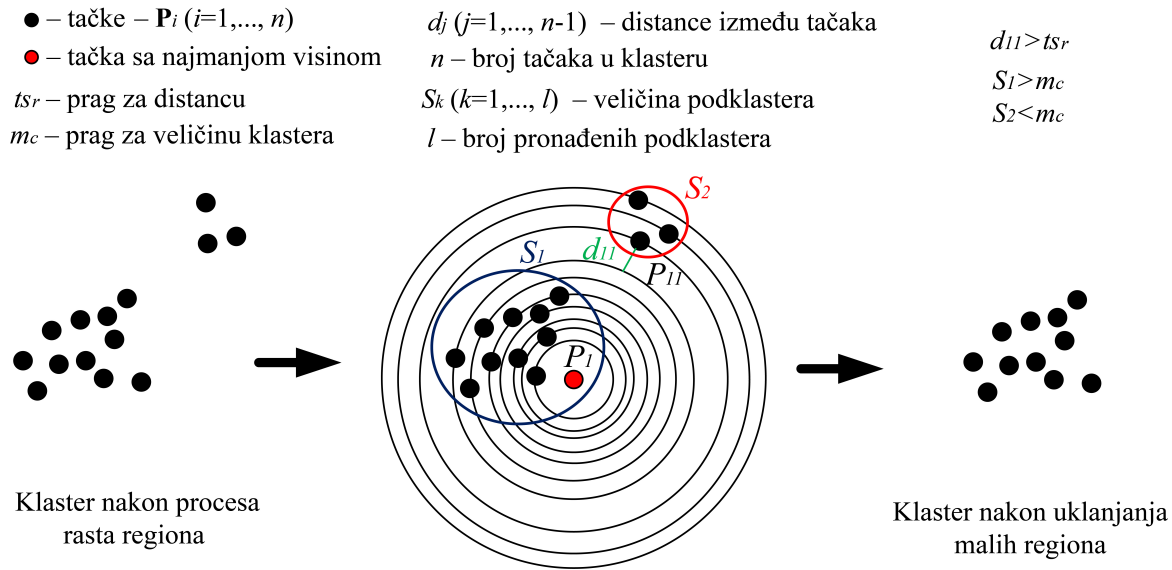
U ovom slučaju problem minimizacije (7.3) se može svesti na jednačinu:

$$\mathbf{S}\mathbf{a} = \lambda\mathbf{C}\mathbf{a}, \quad (7.5)$$

pri čemu je informaciona matrica oblika $\mathbf{S} = \mathbf{D}^T\mathbf{D}$, a λ je Lagranžov množilac.

Jednačina (7.5) ima tačno jedno rešenje u vidu vektora koji sadrži 10 vrednosti koje predstavljaju parametare $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_{10}]^T$ jednačine kvadrike. Kao i u slučaju segmentacije struktuiranih oblaka tačaka (poglavljje 6) i u ovoj metodi segmentacija je zasnovana na tome da je informaciona matrica \mathbf{S} singularna kada sve tačke sistema leže egzaktno na površini, a bliska singularnosti kada tačke

aproksimiraju tu površinu (npr. u realnim uslovima kada su tačke uzorkovane sa određenim greškama i šumom). U slučaju kada je matrica \mathbf{S} bliska singularnosti, njen recipročni faktor uslovljenosti je blizak nuli i ovu osobinu koristimo za inicijalno širenje regiona.



Slika 7.2: Uklanjanje malih regiona: ilustracija u 2D prostoru.

Sam proces segmentacije je iterativan i svaka iteracija počinje izborom inicijalne tačke koja predstavlja najvišu tačku (tačku sa najvišom z koordinatom) do tada neiskorišćenog dela oblaka. Ovakvim izborom se povećava verovatnoća da upravo ta tačka predstavlja lokalni ekstrem neke nove površine, odnosno novog segmenta oblaka. Tokom svake iteracije, metoda koristi knn algoritam [139, 140, 141] kojim se u 3D okruženju određuju tačke sa najmanjom algebarskom distancom od inicijalne. Distanca može biti određena korišćenjem različitih poznatih funkcija (npr. Euklidove, Menhetn, Lagranžove) [139], ali se u ovom slučaju koristi knn sa Euklidovom algebarskom distancom. Na taj način vrši se sortiranje svih tačaka oblaka na osnovu rastojanja od izabrane inicijalne u rastućem nizu, odnosno od najbliže ka najdaljoj.

U nastavku procesa se iz sortirane grupe tačaka u sistem dodaje tačka po tačka pri čemu se u svakom koraku vrši izračunavanje recipročnog faktora uslovljenosti matrice \mathbf{S} i njegova vrednost poredi sa unapred zadatom veoma niskom vrednošću praga za segmentaciju (ts_s). Ukoliko vrednost recipročnog faktora uslovljenosti nije veća od zadatog praga, tekuće dodata tačka se pridružuje do tada formiranom klasteru tačaka. Kada se dođe do tačke koja ne ispunjava pomenuti uslov, inicijalna segmentacija, odnosno rast regiona se prekida. Tada se pristupa drugom koraku segmentacije. Treba naglasiti da postoji i prag za minimalan broj tačaka u inicijalnom klasteru (m_{ic}) i njegova vrednost zavisi od nivoa šuma u oblaku. Ukoliko se pri inicijalnom širenju regiona ne dostigne ovaj zadati broj tačaka proces se prekida, bira se nova inicijalna tačka i iteracije se ponavljaju.

U pomenutom drugom koraku segmentacije se do tada izdvojeni klaster tačaka koristi kako bi se izvršila estimacija parametara pripadajuće površine metodom najmanjih kvadrata. Zatim se iz preostalog oblaka tačaka vrši izdvajanje dodatnih tačaka po uslovu pripadnosti toj površini. Uslov pripadnosti tačke površini je zadovoljavanje jednačine kvadrike sa procenjenim parametrima, a na osnovu neke tolerancije odnosno praga ts_e . Vrednost ovog praga je, na osnovu iskustava sa procesiranjem različitih oblaka, fiksno postavljen na $ts_e = 0,5 \cdot 10^{-4}$. Takođe, treba istaći da u ovom koraku figurira još jedan prag, a to je minimalni broj tačaka finalno izdvojenog klastera u vidu parametra m_c . Ovim parametrom algoritma se smanjuje mogućnost presegmentacije i sama njegova vrednost zavisi od gustine procesiranog oblaka i od broja i veličine očekivanih konačno pronađenih površina.

```

ULAZ:      cloud = [x y z] – registrovani oblak tačaka
              tss – prag za segmentaciju
              alpha – pridruženi parametar praga za segmentaciju
              sigma – srednja vrednost distance između tačaka u oblaku
              mic – minimalni broj tačaka u inicijalnom klasteru
              mc – minimalni broj tačaka u klasteru

```

```

m = 0; trials = 0;
while trials < 10
    [_,seed] = max(z);
    idx=knn(seed,cloud);
    temp_cluster(1,:) = cloud(seed,:);
    for i = 1:length(idx)
        surf_aux = [temp_cluster; cloud(idx(i,:))];
        izračunati rcond(S) za surf_aux
        if rcond(S) < tss
            temp_cluster = [temp_cluster; cloud(idx(i,:))];
        else
            brake
        end if
    end for
    if length(temp_cluster) > mic
        l1 = 0; l2 = 10;
        while l1 < l2
            l1 = length(temp_cluster);
            izračunati a korišćenjem temp_cluster
            for i = 1:length(cloud)
                izračunati distance(i)
                if distance(i) < 0.5e-4
                    temp_cluster = [temp_cluster; cloud(i,:)]
                end if
            end for
            l2 = length(temp_cluster);
        end while
        temp_cluster = pruning(temp_cluster,alpha*sigma,mc);
    end if
    if length(temp_cluster) >= mc
        m = m+1;
        surfaces(m).cluster = temp_cluster;
        cloud = setdiff(cloud,surfaces(m).cluster);
    else
        trials = trials + 1;
    end if
    if length(cloud) < 2
        trials=10;
    end if
end while
calcPoints()

```

```

IZLAZ:      surfaces – struktura koj sadrži segmentirane regione

```

Slika 7.3: Pseudokod jedne iteracije segmentacije oblaka tačaka

Nakon izvršenja oba koraka segmentacije i formiranja finalnog klastera, proces kreće u sledeću iteraciju i traženje novog regiona. Ovaj postupak se ponavlja sve dok se ne izvrši segmentacija kompletnog oblaka. Iako dopunsko izdvajanje tačaka, odnosno drugi korak segmentacije uobličava klaster i dovodi do boljih rezultata, ono takođe može dovesti do izdvajanja malih nepovezanih regiona u samom klasteru. Ovi regioni mogu biti i tačke koje ne pripadaju geometriji, već su proizvod greške skeniranja ili šuma u oblaku pa samim tim mogu dovesti do problema u daljem procesiranju.

Da bi se izbegla navedena situacija, na kraju postupka segmentacije uvodi se proces uklanjanja ovih potencijalno izdvojenih malih nepovezanih regija iz svakog od klastera. Pod malim nepovezanim grupama tačaka smatraju se one koje sadrže manje od m_c tačaka, a udaljene su od veće grupe tačaka više od unapred definisanog praga $ts_r = \alpha \cdot \sigma$, gde je σ srednja međusobna udaljenost susednih tačaka u oblaku, odnosno srednja gustina oblaka, a α pridruženi parametar.

Procedura uklanjanja navedenih malih regiona iz klastera počinje procesiranjem od najniže tačke klastera (tačka sa najmanjom z koordinatom) - \mathbf{P}_1 i koristi knn algoritam da sračuna Euklidove distance od nje do ostalih tačaka klastera, kao i da ih sortira po udaljenosti u uzlaznom rasporedu. Ove sračunate Euklidove distance predstavljaju radijus zamišljenih sfera definisanih centrom u tački \mathbf{P}_1 i ostalim tačkama $\mathbf{P}_i, i = (2, \dots, n)$, gde n predstavlja ukupan broj tačaka u klasteru. Razlika vrednosti radijusa između dve susedne tačke u nizu (\mathbf{P}_i i \mathbf{P}_{i+1}) koja je veća od ts_r pokazuje da se radi o značajnom razmaku između dve tačke. U tom slučaju se prethodno procesirana grupa tačaka odvaja od narednih tačaka, odnosno dobijaju se dve grupe tačaka - $\mathbf{P}_1 - \mathbf{P}_i$ i $\mathbf{P}_{i+1} - \mathbf{P}_n$. Procesiranjem svih tačaka klastera vrši se krajnja podela na konačan broj nezavisnih grupa - podklastera. Na kraju procedure se iz klastera uklanjaju svi podklasteri koji sadrže manji broj tačaka od prethodno definisanog praga m_c . Radi lakšeg razumevanja navedenog, na slici 7.2 prikazana je 2D reprezentacija objašnjene procedure. Takođe, na slici 7.3 dat je pseudokod jedne iteracije kompletnog procesa segmentacije opisane u ovoj sekciji.

7.3. Algoritam spajanja i estimacija parametara površina

Kao i metoda segmentacije predstavljena u poglavlju 6, i poslednje opisani proces se koristi za podelu oblaka tačaka na regione koji pripadaju različitim građivnim primitivima skeniranog objekta. Međutim, pored jasnog razdvajanja primitiva, ovaj proces je, u slučajevima povišenog šuma, podložan izdvajanju prekobrojnih regiona na istim površinama. Znajući da je u realnim oblacima tačaka pojava šuma česta i značajna, za razmatranu metodu prepoznavanja primitiva kreiran je dodatni proces spajanja prekobrojnih segmenata. Osnov algoritma spajanja predstavlja metoda robusnog prepoznavanja cilindara (RCF) [108] koja je za ove potrebe modifikovana i prilagođena. Ovom modifikacijom je RCF metodi povećana sposobnost generalizacije i primene o čemu će između ostalog biti i reči u nastavku. Pored spajanja presegmentiranih regiona algoritmu je dodata i uloga estimacije parametara prepoznatih površina. U skladu sa polaznim zahtevom da metoda prepoznaje cilindre i ravni i činjenice da sama RCF metoda vrši prepoznavanje parametara cilindara bilo je potrebno dodati i proces estimacije parametara ravni. Za tu svrhu razvijena je tehnika zasnovana na RANSAC-u koja će takođe biti objašnjena u ovom poglavlju.

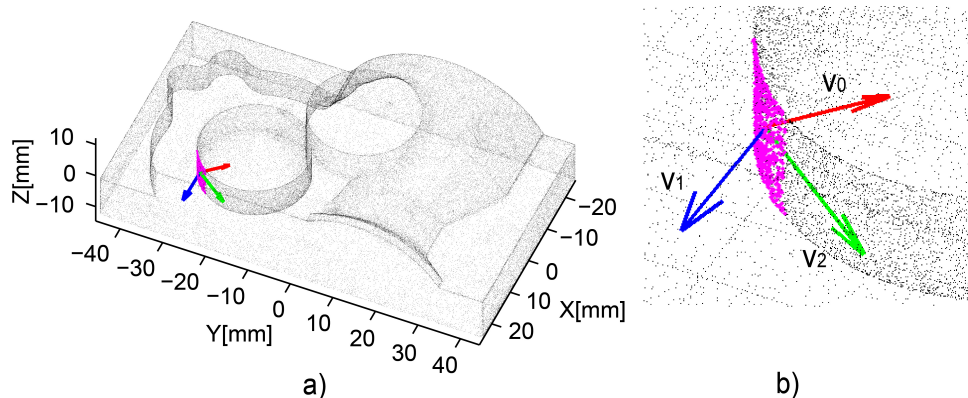
7.3.1. Modifikovani proces robusnog prepoznavanja cilindara iz oblaka tačaka

Metoda robusnog prepoznavanja cilindara u oblacima tačaka iz [108], odnosno skraćeno RCF metoda, razvijena je za procesiranje oblaka tačaka dobijenih mobilnim laserskim skenerom montiranim na vozilu u pokretu. Prevažodno je namenjena prepoznavanju cilindara sa značajno većom visinom u odnosu na njihov radijus (stubovima saobraćajne signalizacije i sličnim objektima na kolovozu) i kao takva koristi određene matematičke aparate i pojednostavljenja koji su orijentisani takvoj vrsti upotrebe. Treba naglasiti da je RCF metoda namenjena samo robusnoj proceni parametara cilindra iz već izdvojenih oblasti tačaka koje reprezentuju ovaj tip primitiva. Ova metoda ne sadrži nikakvu vrstu prethodne segmentacije oblaka. Sa druge strane RCF za procenu pravca osa cilindara koristi poznatu tehniku robusnog određivanja glavnih komponenti sistema (RPCA) [107] o kojoj je već bilo reči u poglavlju 3.2.

Izvorni RCF algoritam iz [108] sadrži dve glavne faze nakon kojih obezbeđuje estimaciju četiri glavna parametra cilindra - ose cilindra (vektora koji reprezentuje njen pravac), centra osnovne kružnice, radijusa i visine cilindra. U prvoj fazi algoritam procesira izdvojeni klaster oblaka tačaka korišćenjem RPCA [107] čime generiše vrednosti tri sopstvena vektora ($\mathbf{v}_i, i = 2, 1, 0$) koja odgova-

raju sopstvenim vrednostima ($\lambda_i, i = 2, 1, 0$, obično je $\lambda_2 \geq \lambda_1 \geq \lambda_0$) kovarijantne matrice sistema. Ova tri sopstvena vektora predstavljaju pravce u kojima dolazi do najviše varijabilnosti koordinata tačaka klastera. Pošto je inicijalna RCF metoda namenjena prepoznavanju cilindara sa izraženo većom visinom od vrednosti radijusa za autore ove metode je bilo praktično da vektor \mathbf{v}_2 (pravac najviše promene koordinata) koriste u svim slučajevima kao osu prepoznatog cilindra. Na osnovu toga, osnovnu kružnicu cilindra je uvek moguće naći u projekciji tačaka klastera na ravan koju definišu druga dva sopstvena vektora - \mathbf{v}_1 i \mathbf{v}_0 .

Navedena pretpostavka o permanentnoj upotrebi vektora \mathbf{v}_2 kao vektora pravca ose cilindra nije upotrebljiva u opštem slučaju, a pre svega u slučajevima kada klaster tačaka reprezentuje samo mali deo površine omotača nekog cilindra. Kako bi se ilustrovao ovaj problem, na slici 7.4 je prikazan klaster koji je izdvojen iz nestruktuiranog oblaka tačaka test objekta koji je već korišćen u prethodnim poglavljima (slika 5.14(a)). Izdvojeni klaster reprezentuje deo cilindra kod koga visina nije značajno veća od radijusa. Klaster je procesiran RPCA tehnikom i na prikazu se mogu videti dobijena tri osnovna vektora. Prostim vizuelnim pregledom može se ustanoviti da ni jedan od generisanih vektora ne odgovara traženom pravcu ose cilindra. Zbog mogućih čestih susreta sa cilindrima ovakve geometrije, izvorni RCF algoritam nije mogao biti upotrebljen za potrebe razvijene metode prepoznavanja i morao je biti modifikovan kako bi se povećala sposobnost generalizacije.



Slika 7.4: Procesiranje klastera koji reprezentuje cilindar upotrebom RPCA algoritma ($\alpha = 0,75$; $k = 3$): a) oblak tačaka; b) procesirani klaster.

Nakon dubljeg istraživanja na temu određivanja ose cilindra iz klastera tačaka u opštem slučaju došlo se do zaključka da postavljeni problem nikako nije trivijalan i da generalno rešenje još uvek ne postoji. Kako je ovaj rad orijentisan na primenu metode procesiranja oblaka tačaka za potrebe mašinske industrije, ipak se mogu uvesti i neka pojednostavljenja. Pošto je većina mašinskih delova izgrađena od cilindara čije se ose prostiru u tri ortogonalna pravca, metoda se može ograničiti na ekstrakciju cilindara upravo duž tih pravaca. Pomoć u implementaciji ovog pojednostavljenja predstavljaju moderne tehnike registracije oblaka kojima se nakon skeniranja određenog objekata, ose koordinatnog sistema (pridruženog oblaku) lako mogu postaviti duž željenih pravca. U tom slučaju predstavljena metoda bi imala lakši zadatak i prepoznavala bi samo cilindre koji poseduju bazne kružnice u osnovnim ravnima oblaka. Automatsko određivanje ravni u kojoj će se naći potrebna kružnica je drugi problem sa kojim se treba suočiti. Imajući sve navedeno u vidu kao i činjenicu da RPCA ima svoja ograničenja u opštem slučaju, prva faza RCF procesa je modifikovana za potrebe primene u predloženoj metodi prepoznavanja cilindara iz ovog poglavlja.

Radi pojednostavljenja daljeg izlaganja, uvodi se pojam tzv. najbolje ravni. Naime, potrebno je za svaki klaster tačaka koji reprezentuje određeni cilindar (koji je dobijen nakon segmentacije) odrediti pravac prostiranja njegove ose koji se nalazi duž jednog od tri osnovna pravca oblaka. Ovo se može izvesti projektovanjem klastera na sve tri osnovne ravni i izborom najpogodnije ravni, odnosno ravni u kojoj se realno i najpreciznije može pronaći bazna kružnica cilindra. Ovu najpogodniju ravan u ovom

slučaju nazivamo najboljom ravni.

Osnovna ideja pronalaženja najbolje ravni je procesiranje sve tri projekcije tekućeg klastera (u tri osnovne ravni oblaka) korišćenjem procesa druge faze izvornog RCF algoritma i pronalaženje one kružnice kod koje je odstupanje od skupa tačaka najmanje. Time se dolazi do izbora najbolje ravni čija normala ujedno predstavlja pravac ose cilindra (vektor \mathbf{i} , \mathbf{j} ili \mathbf{k}), a samim tim i nađena najbolja kružnica postaje bazna kružnica cilindra.

Kako bi se pojasnio sam proces druge faze RCF algoritma u nastavku je prikazano detaljnije objašnjenje. Procesiranje se izvodi u tri glavna koraka:

1. U prvom koraku algoritam koristi tzv. hiper regresiju kružnice [142] na osnovu tri nasumično izabrane tačke projekcije klastera na ravan. Vršiti se izračunavanje koordinata centra (a_0, b_0) i radijusa r_0 inicijalne kružnice kao i kvadratnih odstupanja e_i^2 za sve tačke (x_i, y_i) , $i = 1, \dots, n$ (n je ukupni broj tačaka) projekcije. Za izračunavanje kvadratnih odstupanja upotrebljava se obrazac:

$$e_i^2 = [\sqrt{(x_i - a_0)^2 + (y_i - b_0)^2} - r_0]^2, \quad i = 1, \dots, n. \quad (7.6)$$

2. U drugom koraku vrši se sortiranje kvadratnih odstupanja (reziduala) u rastućem nizu i bira se polovina ($h = n/2$) tačaka sa najmanjim vrednostima reziduala.
3. U trećem koraku algoritam ponovo koristi hiper regresiju kružnice i izračunava koordinate centra (a_0, b_0) i radijus r_0 , ali ovoga puta na osnovu h izabranih tačaka iz prethodnog koraka. Nakon toga se ponovo na osnovu obrasca (7.6) za sve tačke projekcije izračunavaju kvadratna odstupanja, sortiraju se u uzlaznom nizu i izračunava se njihova suma za prvih h tačaka.

Opisani koraci se iterativno ponavljaju I_n puta, pri čemu je I_n broj iteracija koji se određuje na osnovu Monte Karlo statističke metode po obrascu:

$$I_n = \frac{\log(1 - p_r)}{\log(1 - (1 - \epsilon)^{h_0})}. \quad (7.7)$$

U prethodnoj jednačini p_r je verovatnoća pronalaženja minimalnog skupa tačaka koji pripadaju skeniranoj geometriji (engl. *outliers-free minimal subset*) u klasteru sa ϵ procenata tačaka koje predstavljaju grešku skeniranja (engl. *outliers*). U ovom radu p_r je podešeno na 0,999, dok je $\epsilon = 0,6$.

Sume kvadratnih odstupanja (iz trećeg koraka algoritma) iz svih I_n iteracija se porede i h tačaka iz najbolje iteracije (sa najmanjom vrednošću navedene sume) se uzima za finalnu estimaciju parametara bazne kružnice ponovnim korišćenjem hiper metode regresije. Objašnjena procedura se ponavlja za sve tri projekcije tekućeg klastera i najboljom projekcijom (ravni) se proglašava ona čija finalna kružnica poseduje najmanju vrednost sume kvadratnih odstupanja (za h najboljih tačaka). Estimirani radijus, koordinate centra finalne bazne kružnice i odgovarajući vektor normale (pravac ose cilindra) iz najbolje ravni ujedno postaju i parametri cilindra koga klaster reprezentuje. Takođe, na kraju procesa se kao dopuna RCF metode, a na osnovu sume kvadratnih odstupanja vrši i izračunavanje srednjeg kvadratnog odstupanja pomenutih h tačaka najbolje ocenjene projekcije kao:

$$asr = \frac{\sum_{i=1}^h e_i^2}{h}. \quad (7.8)$$

Treba naglasiti da će vrednost ovako generisanog parametra asr biti od značaja u daljem toku algoritma prepoznavanja.

7.3.2. Procena parametara ravni

Kao što je već naglašeno, metoda prepoznavanja primitiva pored prepoznavanja cilindara vrši i prepoznavanje ravni. Pošto modifikovana RCF metoda vrši robusnu estimaciju parametara cilindara, potrebno je definisati i metodu koja će vršiti procenu parametara površi koje se budu detektovale kao ravni. U tu svrhu razvijen je jednostavan proces zasnovan na geometrijskim i RANSAC principima. Kako bi ovaj proces bio bliže objašnjen potrebno je poći od jednačine ravni koja može biti definisana sledećim oblikom:

$$ax + by + cz + d = 0. \quad (7.9)$$

Parametri a , b i c predstavljaju koordinate vektora normale ravni $\mathbf{n} = [a \ b \ c]$, dok je parametar d definisan kao skalarni proizvod tog vektora normale i vektora koji definiše poziciju nasumično izabrane tačke na toj ravni.

Posmatranjem jednačine ravni (7.9) može se uočiti njena jasna sličnost sa implicitnom jednačinom opšte površine drugog reda (jednačina (7.1)). Naime, ukoliko su parametri a_1, \dots, a_6 opšte jednačine kvadrike jednaki nuli onda ova jednačina dobija oblik jednačine ravni.

Predložena metoda za estimaciju parametara ravni (a, b, c, d) predstavlja iterativni proces sa I_n iteracija (jednačina (7.7)). U svakoj iteraciji se vrši nasumični izbor tri tačke ($\mathbf{P}_1, \mathbf{P}_2$, i \mathbf{P}_3) iz datog klastera i na osnovu njih se računa vektor normale kao:

$$\mathbf{n} = \mathbf{P}_1\mathbf{P}_2 \times \mathbf{P}_1\mathbf{P}_3, \quad (7.10)$$

gde $\mathbf{P}_i\mathbf{P}_j$ predstavlja vektor od tačke \mathbf{P}_i do tačke \mathbf{P}_j . Definisanjem vektora normale dobijaju se prva tri parametra (a, b, c) jednačine ravni. Parametar d se dodatno računa na osnovu skalarnog proizvoda vektora normale i vektora \mathbf{P}_1 (vektor od koordinatnog početka do tačke \mathbf{P}_1):

$$d = \mathbf{n} \cdot \mathbf{P}_1. \quad (7.11)$$

Takođe, u svakoj iteraciji algoritam vrši izračunavanje distanci između svih tačaka klastera i ravni sa prethodno procenjenim parametrima a, b, c, d . Distanca između svake tačke $\mathbf{P}_i(x_i, y_i, z_i)$, $i = 1, \dots, n$ (n je ukupni broj tačaka klastera) i ravni se računa kao:

$$D = |D_v \cdot \mathbf{e}|, \quad (7.12)$$

gde je D_v :

$$D_v = \frac{ax_i + by_i + cz_i + d}{\sqrt{a^2 + b^2 + c^2}}, \quad (7.13)$$

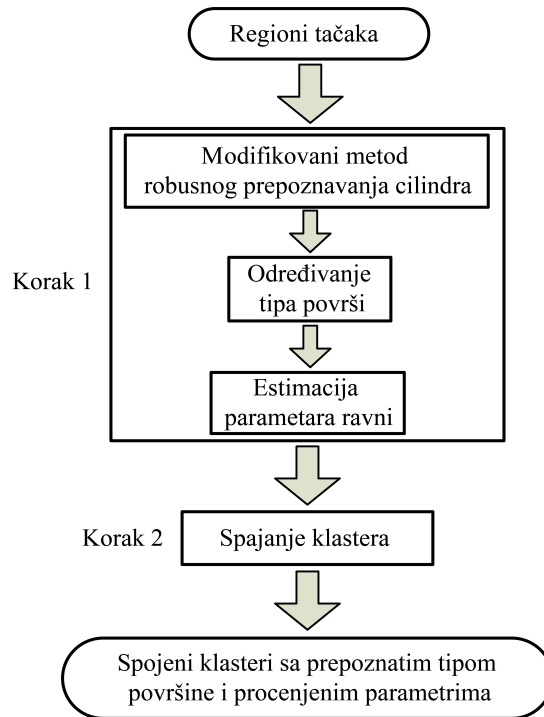
dok je $\mathbf{e} = \mathbf{n}/|\mathbf{n}|$ jedinični vektor normale ravni.

Nakon izračunavanja distanci svih tačaka do ravni, vrši se definisanje broja tačaka koje su od ravni udaljene manje ili jednako od unapred zadatog malog rastojanja (praga) koji je u ovom radu postavljen na 0,01. Nakon poređenja svih iteracija bira se najbolja, odnosno ona kod koje je najveći broj tačaka na definisanom rastojanju od ravni. Parametri ravni iz najbolje ocenjene iteracije ujedno postaju i konačno izabrani (procenjeni) parametri koje predstavljaju proces generiše na izlazu.

7.3.3. Definicija algoritma spajanja

Nakon pojašnjenja modifikovane RCF metode i procesa estimacije parametara ravni, može se definisati algoritam spajanja presegmentiranih regiona oblaka. Algoritam se u osnovi sastoji od dva koraka. U prvom koraku se svi klasteri dobijeni nakon segmentacije sortiraju u tri grupe prema tipu površine koju reprezentuju na cilindre, ravni i ostale površine (površine nekog drugog tipa). U ovom koraku se vrši i estimacija parametara cilindara i ravni. U drugom koraku se zatim potrebni klasteri spajaju na osnovu tipa i parametara dobijenih u prvom koraku. Takođe se i za spojene klasterne ponovo

vrši estimacija parametara. Struktura algoritma spajanja je grafički prikazana na slici 7.5.



Slika 7.5: Struktura algoritma spajanja klastera.

Klasifikacija klastera u pomenute tri grupe vrši se tako što se svi inicijalni klasteri prvo procesiraju modifikovanim RCF algoritmom iz poglavlja 7.3.1 čime se pored ostalih parametara dobijaju procenjene vrednosti radijusa potencijalnih cilindara kao i vrednost srednjeg kvadratnog odstupanja asr (jednačina (7.8)). Razgraničenje između traženih primitiva (cilindara i ravni) i ostalih tipova površi, koje se eventualno mogu naći u oblaku, vrši se upravo pomenutom asr vrednošću. Ukoliko je ova vrednost veća od $0,02\sigma$ (σ je srednja gustina oblaka) klaster se svrstava u grupu onih koji reprezentuju neke druge tipove površi i odstranjuje se iz oblaka. Nakon odstranjivanja drugih površi pristupa se razdvajanju cilindara i ravni. Ako se ima u vidu da će RCF algoritam u slučajevima klastera koji reprezentuju ravni generisati veoma velike vrednosti radijusa onda se prostom selekcijom na osnovu unapred zadatog praga (tm_r) može izvršiti razvrstavanje između cilindara i ravni - ukoliko je estimirana vrednost radijusa manja od tm_r reč je o cilindru, dok je u suprotnom reč o ravni. Ovaj prag je na osnovu iskustva prilikom razvoja algoritma postavljen na vrednost $tm_r = 1.000$ i kao takav se koristi za sva procesiranja predstavljena u ovom radu.

Nakon definisanja tipova površina i odstranjivanja onih koje nisu cilindri ili ravni, pristupa se estimaciji parametara površi. Kako je modifikovani RCF algoritam već primenjen nad svim klasterima i kako se njime generišu parametri cilindara, površine koje su deklarirane kao cilindri ne zahtevaju dodatnu estimaciju parametara. Sa druge strane, površine za koje se ustanovi da pripadaju ravnima zahtevaju izračunavanje parametara. U tom slučaju primenjuje se postupak procene parametara iz poglavlja 7.3.2.

Procedura spajanja adekvatnih presegmentiranih oblasti, koja se primenjuje nakon razvrstavanja površina po tipu i procene njihovih parametara, može se podeliti na dva izdvojena paralelna procesa - spajanje ravni i spajanje cilindara. Oba procesa su iterativna i upoređuju redom po dva po tipu odgovarajuća klastera i izvršavaju se sve dok postoji mogućnost spajanja. Proces spajanja ravni u razmatranom paru površina vrši izračunavanje distanci (jednačina (7.12)) tačaka drugog razmatranog klastera u odnosu na ravan koja je reprezentovana prvim klasterom. Izračunate distance se zatim upoređuju sa unapred definisanim pragom koji je jednak duploj vrednosti prosečne gustine oblaka koji se procesira - 2σ . Ukoliko se 90% tačaka drugog klastera nalazi na udaljenosti manjoj od definisanog

praga dolazi se do zaključka da oba razmatrana klastera reprezentuju u stvari istu ravan i u tom slučaju se vrši njihovo spajanje uz naknadnu ponovnu estimaciju parametara novoformiranog regiona.

ULAZ:	surfaces – segmentirani regioni sigma – srednja vrednost distance između tačaka u oblaku tmr – maksimalni očekivani radijus cilindra
--------------	--

```

for i = 1:length(surfaces)
    izvršiti estimaciju parametara cilindra (radijus, osa, centar baznog kruga) i srednju vrednost
    kvadratnih odstupanja (asr) korišćenjem procedure iz poglavlja 7.3.1
    if asr < 0.02*sigma
        if radius <= tmr
            surface(i).type = 'cilindar'
        else
            surface(i).type = 'ravan'
            izvršiti estimaciju parametara ravni korišćenjem metode iz poglavlja 7.3.2
        end if
    else
        surface(i).type = 'drugi tip površine'
    end if
end for
while nema više površina za spajanje
    for i = 1:length(surfaces)
        for j = 1:length(surfaces)
            if surface(i) i surface(j) su cilindri sa podudarnom osom
                izračunati ctcd i rd          ▷ distanca između centara baznih kružnica i razlika radijusa cilindra
                threshold = 0.05*surface(i).radius;
                if ctcd <= threshold & rd <= threshold
                    spojiti površine
                    izračunati parametre cilindra spojene površine korišćenjem modifikovanog RCF procesa
                    za poznatu osu
                end if
            end if
            if surface(i) i surface(j) su ravni
                izračunati d                    ▷ distanca između surface(i) i tačaka surface(j)
                dis=find(d <= tmp);
                if length(dis) > 0.9*length(surface(j))
                    spojiti površine
                    izračunati parametre ravni spojene površine
                end if
            end if
        end for
    end for
end while

```

IZLAZ:	surfaces – struktura koja sadrži regione i parametre površina
---------------	---

Slika 7.6: Pseudokod za spajanje klastera

Sa druge strane, spajanje cilindara je zasnovano na tri kriterijuma. Prvi kriterijum je podudarnost osa, drugi je rastojanje između centara baznih kružnica, dok je treći kriterijum razlika vrednosti radijusa. Kao što je već naglašeno, nakon procesiranja svih klastera modifikovanim RCF algoritmom dobijamo estimirane vrednosti radijusa, pozicije (koordinate) centara bazne kružnice i ose orijentacije svih segmentiranih cilindara. Kako je modifikacija izvorne RCF metode izvršena kako bi se ekstrahovali cilindri sa međusobno upravnim osama čiji pravci odgovaraju osama osnovnog koordinatnog sistema oblaka, jasno je da je za pravce osa moguće dobiti samo tri jedinična vektora (**i**, **j** i **k**) čiju je međusobnu podudarnost moguće trivijalno proveriti. Ovakav način određivanja pravaca osa cilindara znači da se tokom RCF procesa klasteri projektuju na osnovne koordinatne ravni oblaka i bazna kružnica sa njenim centrom i radijusom se pronalazi u jednoj od te tri projekcije. Ovo znači da se svi

pronađeni centri baznih kružnica cilindara koji poseduju podudarne ose nalaze u istoj ravni i njihovo međusobno Euklidovo rastojanje je moguće lako izračunati.

Sam proces spajanja cilindara je takođe iterativan i izvodi se jednostavnim poređenjem svih cilindara (klastera) sve dok postoji potreba za njihovim spajanjem. U svakoj iteraciji dva cilindra se porede u dva koraka. Prvo se vrši poređenje na osnovu pravaca osa i samim tim dva cilindra čiji se pravci osa ne podudaraju se ne mogu spojiti u jedan. Ukoliko se stupi u drugi korak procesa, prvo se vrši kalkulacija rastojanja između pripadajućih centara baznih kružnica. Zatim se vrši poređenje ovog rastojanja kao i apsolutne vrednosti razlike radijusa oba cilindra sa definisanim pragom koji fiksno iznosi 5% vrednosti radijusa prvog od dva razmatrana cilindra. Ukoliko su i vrednost distance između centara i razlika radijusa manje ili jednake predviđenom pragu, algoritam donosi odluku o spajanju razmatrana dva cilindra (klastera) u jedan. Radi preciznijeg određivanja parametara, nakon spajanja novonastali klaster se opet tretira modifikovanim RCF procesom u cilju generisanja novih parametara površine. Treba istaći da je, radi povećanja ukupne brzine izračunavanja, na ovom mestu RCF proces moguće pojednostaviti oduzimajući mu inicijalni proces biranja prave projekcije zbog toga što je u ovoj fazi najbolja projekcija već poznata. Na slici 7.6 prikazan je kompletan pseudokod za definisani proces spajanja klastera i estimaciju parametara prepoznatih površina.

7.4. Ekstrakcija cilindara i ravni

Nakon procesa segmentacije i spajanja presegmentiranih oblasti izvodi se treća faza algoritma prepoznavanja koja predstavlja ekstrakciju prepoznatih cilindara i ravni iz procesiranog oblaka tačaka. Ovom fazom vrši se generisanje konačnih rezultata prepoznavanja, ali se po potrebi dolazi i do preciznije estimiranih parametara ravni. Ova faza se može podeliti na tri koraka, a to su:

- sortiranje klastera,
- ekstrakcija primitiva iz oblaka, i
- uklanjanje malih regiona korišćenjem procedure iz 7.2.

U prvom koraku vrši se sortiranje svih klastera (dobijenih nakon faze spajanja) na osnovu njihove veličine (broja sadržanih tačaka) u silaznom nizu (od najvećeg ka najmanjem). U drugom koraku se ovako uređeni klasteri biraju jedan po jedan počev od najvećeg. Na osnovu parametara njihovih površi, proverava se pripadnost (afilijacija) svih ostalih tačaka inicijalnog oblaka tom izabranom klasteru odnosno površini koju on reprezentuje. Ukoliko tačka po zadatim kriterijumima (koji će biti kasnije navedeni) pripada tekućoj površini, ona se izuzima iz inicijalnog oblaka, pridružuje tekućem (finalno izdvojenom) klasteru, ali i oduzima nekom od narednih manjih klastera (ukoliko im je pripadala). Zbog ovakve procedure, prethodnim sortiranjem je data prednost većim (značajnijim) klasterima, koji zasigurno predstavljaju osnov morfologije skeniranog objekta kako bi se eliminisali eventualni mali klasteri koji mogu predstavljati grešku prethodno sprovedenih procesa. Takođe, treba naglasiti da tokom procesiranja jednog po jednog klastera broj dostupnih tačaka inicijalnog oblaka u svakoj iteraciji postaje sve manji i manji za grupu tačaka do tada ekstrahovanih klastera.

Afilijacija neke tačke površini se posebno definiše za ravni, a posebno za cilindre. U slučaju ravni, tačka pripada površini ukoliko se nalazi na distanci (jednačina (7.12)) manjoj od duple vrednosti srednje gustine oblaka - 2σ . Sa druge strane, u slučaju cilindra se proverava blizina tačke njegovom omotaču. Ukoliko je ta distanca takođe manja od 2σ onda se smatra da razmatrana tačka pripada tekućem cilindru (klasteru).

Kako bi se potrebna distanca tačke od omotača cilindra definisala, treba poći od najkraćeg rastojanja R_{ti} neke tačke $\mathbf{P}_i(x_i, y_i, z_i)$ od ose cilindra $\mathbf{p} = [p_x, p_y, p_z]$. Ovo rastojanje može biti izračunato na osnovu:

$$R_{ti} = |\mathbf{cP}_i - (\mathbf{cP}_i \cdot \mathbf{p})\mathbf{p}|, \quad (7.14)$$

pri čemu je \mathbf{cP}_i vektor od centra bazne kružnice cilindra $\mathbf{c}(c_x, c_y, c_z)$ do tačke \mathbf{P}_i . Na osnovu toga

potrebno najkraće rastojanje tačke P_i od omotača cilindra je:

$$sd = |R - R_{ti}| \quad (7.15)$$

gde R predstavlja procenjenju vrednost radijusa posmatranog cilindra.

ULAZ: cloud = [x y z] – inicijalni registrovani oblak tačaka
surfaces – struktura sa segmentiranim regionima
sigma – srednja vrednost distance između tačaka u oblaku
alpha – parametar praga za segmentaciju
beta – parametar praga za izdvajanje
mc – minimalni broj tačaka u klasteru

```

sortirati površine po veličini u opadajućem nizu
j = 0; k = 0;
for i = 1:length(surfaces)
    surface(i).cluster = setdiff(surface(i).cluster, cloud);    ▷ ukolniti iz surface(i).cluster tačke koje nisu prisutne
                                                                u oblaku

    if length(surface(i).cluster) >= mc
        if surface(i).type == 'ravan'
            j = j+1;
            izdvojiti plane(j).cluster iz oblaka
            cloud = setdiff(cloud, plane(j).cluster);    ▷ uklanjanje iz oblaka tačaka koje pripadaju ravni plane(j)
        end if
        if surface(i).type == 'cilindar'
            k = k+1;
            izdvojiti cylinder(k).cluster iz oblaka
            cloud = setdiff(cloud, cylinder(k).cluster);    ▷ uklanjanje iz oblaka tačaka koje pripadaju cilindru
                                                                cylinder(k)
        end if
    end if
end for
for i = 1:j
    plane(i).cluster = pruning(plane(i).cluster, alpha*sigma, mc);    ▷ uklanjanje malih nepovezanih regiona
    if length(plane(i).cluster) >= mc
        izvršiti estimaciju parametara ravni korišćenjem metode iz poglavlja 7.3.2
        dodati plane(i) u final_planes
    end if
end for
for i = 1:k
    cylinder(i).cluster = pruning(cylinder(i).cluster, alpha*sigma, mc);    ▷ uklanjanje malih nepovezanih regiona
    if length(cylinder(i).cluster) >= mc
        izvršiti estimaciju parametara cilindra korišćenjem RCF procesa za poznatu osu i izračunati srednje
        vrednost kvadratnog odstupanja (asr) korišćenjem procedure iz poglavlja 7.3.1
        if cylinder(i).asr < beta*sigma
            dodati cylinder(i) u final_cylinders
        end if
    end if
end for
end for

```

IZLAZ: final_planes – struktura koja sadrži regione i parametre ravni
final_cylinders – struktura koja sadrži regione i parametre cilindara

Slika 7.7: Pseudokod za ekstrakciju površina

Nakon postupka ekstrakcije primitiva, odnosno uobličavanja njihovih grupa pripadajućih tačaka, sprovodi se proces uklanjanja malih nepovezanih regiona iz svakog klastera ponaosob. U ovu svrhu koristi se već ranije upotrebljavana procedura opisana u poglavlju 7.2. Njeno korišćenje doprinosi boljoj grafičkoj reprezentaciji finalno izdvojenih klastera, ali i eventualnom uklanjanju nepotrebnih tačaka koje bi mogle uticati na proces procene finalnih parametara cilindara i ravni. Klasteri koji u

ovom koraku izgube znatan broj tačaka i smanji im se veličina ispod očekivane (ispod praga m_c) se odstranjuju.

Na samom kraju, na osnovu do tada zadržanih klastera, vrši se odabir i formiranje konačno odabranih skupova ravni i cilindara. Takođe vrši se i ponovna estimacija njihovih parametara. Za klaster koji reprezentuju ravni u ovom koraku nema prepreke, dok je poslednja prepreka za cilindre njihova vrednost srednjeg kvadratnog odstupanja (asr iz jednačine (7.8)). Cilindar će postati član grupe konačno izdvojenih klastera samo ukoliko mu je vrednost asr manja od unapred definisanog praga $te_r = \beta \times \sigma$, gde je β još jedan pridruženi parametar. Ova prepreka je postavljena iz razloga što je tokom razvoja i testiranja algoritma prepoznavanja ustanovljeno da u izuzetnim slučajevima, granično mali klasteri cilindara (koji sadrže broj tačaka nešto veći od m_c) mogu proći proces sve do ovog koraka, ali će posedovati nešto veću asr vrednost pa ih je na ovaj način moguće ukloniti. Ovo se pre svega odnosi na regione realnih objekata koji imaju greške u geometriji izazvane npr. lošom mašinskom izradom o čemu će biti reči u poglavlju verifikacije predstavljenog algoritma prepoznavanja izabranih primitiva. Takođe, kompletan pseudokod faze ekstrakcije je predstavljen na slici 7.7.

7.5. Eksperimentalna verifikacija metode

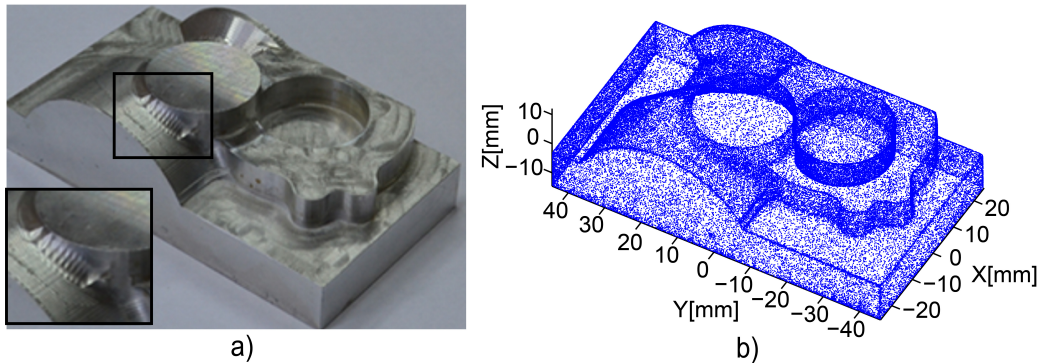
Metoda prepoznavanja cilindara i ravni je verifikovana na više realnih nestruktuiranih oblaka tačaka koji reprezentuju različite objekte čiju morfologiju pretežno sačinjavaju traženi primitivi. U ovom radu biće predstavljeni rezultati procesiranja u tri studije slučaja. Prva studija slučaja odnosi se na prepoznavanje primitiva reprezentativnog test objekata, dok druga i treća studija razmatraju upotrebu metode za objekte koji se često sreću u mašinskoj industriji - vratila i prirubnice. Takođe, predložena metoda je direktno upoređena sa dve postojeće reprezentativne metode koje su se u dosadašnjim istraživanjima izdvojile po svojim performansama. Reč je o metodama drugih autora - metoda za ekstrakciju značajnih primitiva (EoRP) iz [94] i tzv. efikasna RANSAC metoda (eRANSAC) iz [89]. Obe metode pored cilindara i ravni mogu izdvajati i neke druge tipove kvadrata (npr. sfere i konuse) ali su za ove potrebe podešene da prepoznaju samo ravni i cilindre. Takođe treba istaći da su predložena metoda i EoRP implementirane u Matlab okruženju [118], dok je za eRANSAC iskorišćena njena komercijalna implementacija u okviru softvera *CloudCompare* [143].

7.5.1. Prepoznavanje primitiva test objekta

Prva studija slučaja odnosi se na procesiranje realnog oblaka tačaka test objekta koji je između ostalog korišćen i za verifikaciju ostalih metoda predstavljenih u poglavljima 5 i 6. Ovoga puta reč je o procesiranju nestruktuiranog oblaka kompletnog test objekta. Test objekat (slika 7.8(a)) je izrađen mašinskom obradom na 3-osnom obradnom centru, ali sa određenim oblastima lošeg kvaliteta površina. Ovo se pre svega odnosi na izdvojeni region tranzicije između dva dominantna cilindra upravnih osa (izdvojeno posebno na slici 7.8(a)). Nestruktuirani oblak ovog test objekta (slika 7.8(b)) sadrži 79.253 tačke i generisan je upotrebom već pomenutog *Atos Compact Scan 5M* skenera [24]. Srednja gustina tačaka ovog oblaka iznosi $\sigma=0,1834$ mm što je u skladu sa deklaranim opsegom rezolucije uređaja (0,017-0,481 mm). Oblak je registrovan na takav način da su dve upravne bočne ravni objekta kolinearne sa dve osnovne ravni pridruženog koordinanog sistema oblaka. Treba naglasiti da je struktura test objekta sačinjena od 11 cilindara, 2 konusa i 8 ravni, a da zbog prirode skeniranja oblak ne sadrži informacije o donjoj osnovnoj ravni.

Predstavljeni oblak test objekta je procesiran predloženom metodom za prepoznavanje cilindara i ravni, ali i sa druge dve navedene metode za upoređivanje performansi. U slučaju upotrebe predložene metode dobijeni rezultati su očekivano dobri (slika 7.9(a)). Prepoznati su svi potrebni primitivi. Jedini izuzetak je detekcija jedne dodatne ravni (obeležene tamno braon bojom na slici 7.9(a)) koja je prepoznata na delu konusa. Treba naglasiti da metoda nije prepoznala ni jedan primitiv u ranije pomenutim malim regionima karakterizovanim lošom mašinskom obradom. Vrednosti pragova koji su korišćeni

pri procesiranju su prikazne u tabeli 7.1. Sa druge strane, kako bi se istakli rezultati estimacije parametara pri upotrebi predložene metode, u tabelama 7.3 i 7.4 su (zajedno sa rezultatima druge dve metode) prikazane procenjene vrednosti radijusa cilindara i normala ravni zajedno sa njihovim projektovanim vrednostima, respektivno. Pored toga u ovim tabelama su prikazane i neke kvalitativne ocene za upoređivanje performansi metoda, ali će o njima biti reči kasnije pri uporednoj diskusiji.



Slika 7.8: Test objekat: a) fotografija realnog objekta sa akcentom na region karakterizovan lošom mašinskom obradom, b) kompletan nestrukturirani oblak tačaka dobijen 3D skeniranjem objekta.

Tabela 7.1: Predložena metoda: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.8(b)

Prag	ts_s	α	β	m_{ic}	m_c
Vrednost praga	$0,5 \cdot 10^{-18}$	1	0,002	10	50

$\sigma=0,1834$ mm

Tabela 7.2: EoRP: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.8(b)

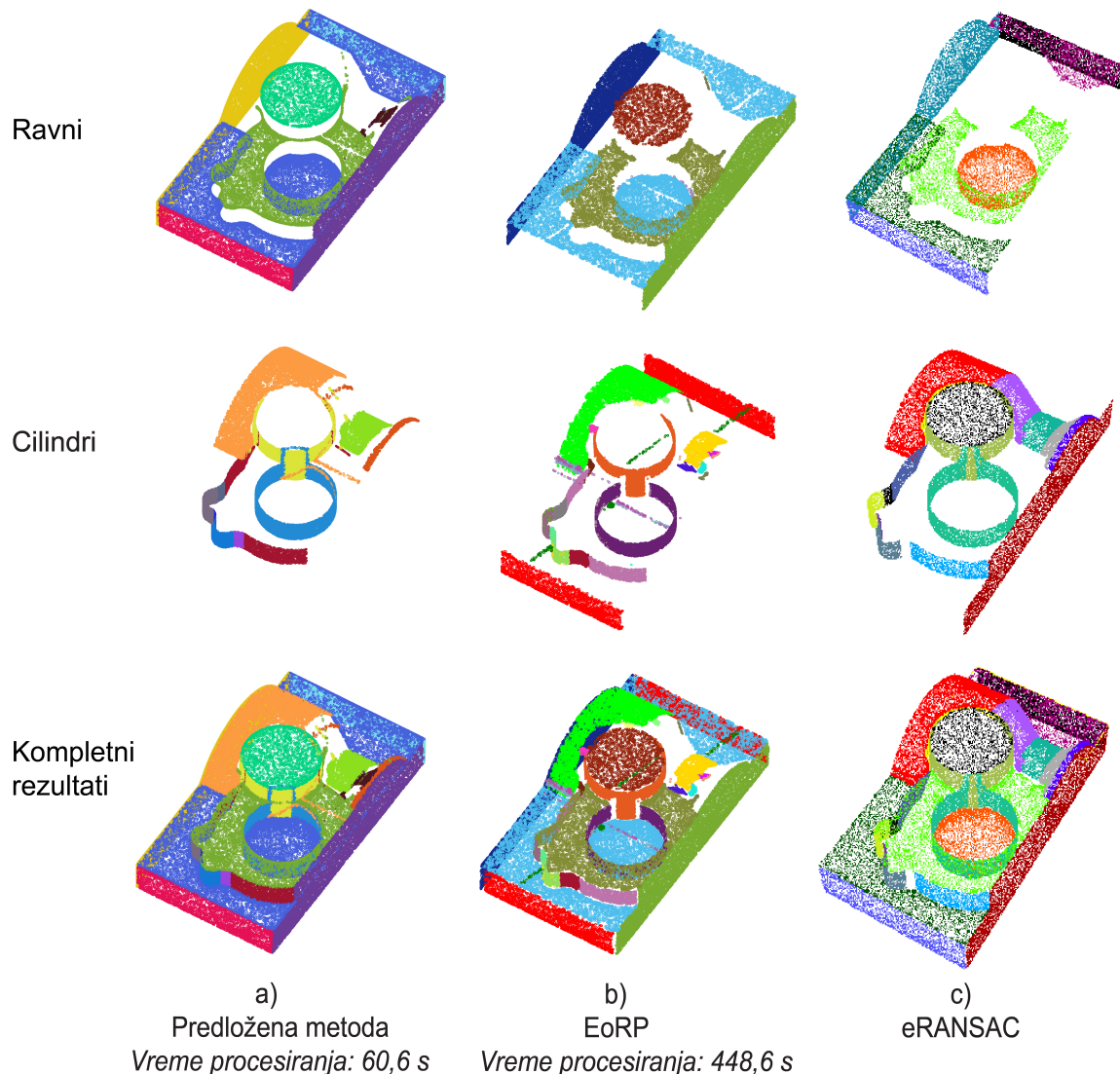
Prag	$k-nn$	α^*	$\beta^*(rad)$	θ	c_{sc}	h_1	h_2	ϵ	α^{**}	τ	cc	β^{**}
Vrednost praga	30	20	0,95	0,99	2	0,01	σ	0,5	20	200	15	3σ

*-za cilindre; **-za ravni; $\sigma=0,1834$ mm

Oblak tačaka je takođe procesiran i EoRP metodom. Ova metoda je zahtevala prethodno podešavanje čak 12 parametara (pragova) kako bi se dobili optimalni rezultati prepoznavanja. Neke vrednosti su postavljene na osnovu više uzastopnih pretraživanja dok su druge postavljene na osnovu iskustava iz [95]. Sve pomenute vrednosti koje su upotrebljene za procesiranje oblaka test objekta su prikazane u tabeli 7.2. Sa druge strane, grafički rezultati primene EoRP metode su prikazani na slici 7.9(b). Na njima se može uočiti da je ova metoda dobro ekstrahovala 10 cilindara i 5 ravni. Međutim, nije bila u stanju da prepozna cilindar radijusa 23 mm koji je reprezentovan uskim regionom tačaka na ivici objekta (na slici 7.9(a) je obeležen svetlo braon bojom). Dalje, dve ravni (obeležene crveno na slici 7.9(b)) su prepoznate kao cilindar velikog poluprečnika. Takođe, izdvojeno je i još nekoliko dodatnih cilindara (na slici 7.9(b) su označeni svetlo plavom, ljubičastom i roze bojom u blizini adekvatno izdvojenog žutog cilindra), kao i jedan cilindar na delu konusa (označen tamno zeleno). Vrednosti radijusa prepoznatih regularnih cilindara i vrednosti normala ravni date su u tabelama 7.3 i 7.4, respektivno.

Efikasna RANSAC metoda kao jedna od najpopularnijih za ovu namenu je takođe iskorišćena za procesiranje oblaka test objekta. Kao što je ranije naglašeno, za ovu svrhu iskorišćena je njena komercijalna implementacija [143]. Kao i u slučaju EoRP metode i ovde su parametri podešeni kako bi

se dobili najbolji mogući rezultati. Postavljene su sledeće vrednosti pragova: $\epsilon = 0,1$, $\beta = 0,9$, $\alpha = 25^\circ$. Verovatnoća nepronalaženja primitiva (engl. *overlocking probability*) je postavljena na 0,01, dok je za minimalnu veličinu klastera izabrano $\tau = 300$ tačaka. Softver je podešen da prepozna samo cilindre i ravni (ne i konuse). Grafički rezultati primene eRANSAC metode su dati na slici 7.9(c).



Slika 7.9: Grafička reprezentacija rezultata prepoznavanja primitiva iz oblaka tačaka sa slike 7.8(b): a) Predložena metoda, b) EoRP, c) eRANSAC.

eRANSAC metoda nije bila sposobna da prepozna jedan cilindar poluprečnika 6 mm. Dve ravni (označene crvenom i crnom bojom na slici 7.9(c)) i dva konusa (označena ljubičastom i sivom bojom na slici 7.9(c)) su prepoznata kao cilindri velikog radijusa. Jedan mali klaster (označen žutom bojom) u blizini najvećeg regularnog cilindra prepoznat je kao dodatni cilindar koji ne postoji na realnom delu, ali to se može opravdati i postojanjem već pomenutog regiona lošeg kvaliteta izrade na realnom objektu. Takođe, zbog strukture svog algoritma, eRANSAC metoda nije sposobna da prepozna više udaljenih segmenata iste ravni već je jednu ravan objekta prepoznala kao tri nezavisne ravni sa posebnim parametrima - ravni obeležene tamno zelenom, narandžastom i rozom bojom na slici 7.9(c). Ista situacija postoji i sa cilindrom radijusa 22,5 mm koga je algoritam razdvojio na dve nezavisne cilindrične površine. Rezultati estimiranih vrednosti poluprečnika prepoznatih cilindara i normale ravni su takođe dati u tabelama 7.3 i 7.4.

Posmatrajući navedene rezultate upotrebe sve tri metode prepoznavanja može se zaključiti da je

metoda predstavljena u ovom poglavlju pokazala najbolje performanse u pogledu pre svega broja pravilno izdvojenih primitiva (kvalitativna analiza). Ona je izdvojila sve zahtevane cilindre i ravni uz jednu dodatnu nepostojeću ravan na površini konusa. Sa druge strane, EoRP i eRANSAC metode su izdvojile više pogrešnih primitiva s tim što je dolazilo i do tzv. konfuzije modela (engl. *model confusion*) gde su ravni prepoznavane kao cilindri velikih prečnika. EoRP nije adekvatno prepoznala jedan traženi cilindar i dve ravni, ali je izdvojila čak 5 nepotrebnih cilindara uz dve konfuzije modela (dve ravni je prepoznala kao jedan cilindar). eRANSAC je sa druge strane propustila da izdvoji dve ravni i jedan cilindar. Ona je takođe izdvojila dve pogrešne ravni (kao jedan cilindar), jedan nepostojeći cilindar kao i dva konusa kao cilindre.

Tabela 7.3: Estimirani radijusi cilindara [mm] za oblak tačaka sa slike 7.8(b)

Metoda	Dizajnirani radijusi [mm]											ARD[mm]
	22,5	12,5	12,5	6	4	3	4	6	18	23	25,5	
Predložena	22,5390	12,5675	12,4742	5,8505	4,2972	3,1439	4,1543	6,1232	18,0379	23,1739	25,6849	0,1270
EoRP	22,5386	12,4858	12,4858	6,0867	4,0265	2,9320	4,0239	6,0867	18,0420	-	25,5823	0,0439
eRANSAC	23,7443	12,4308	12,6198	5,9518	3,9469	3,0016	3,9523	-	17,4109	21,9214	25,6911	0,3130

Tabela 7.4: Estimirane normale ravni za oblak tačaka sa slike 7.8(b)

Metoda	Dizajnirane normale ravni							AND [°]
	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	
Predložena	$\begin{bmatrix} 1 \\ 0 \\ -0,0015 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -0,0002 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0,0003 \\ 1 \\ 0,0011 \end{bmatrix}$	$\begin{bmatrix} 0,0002 \\ 1 \\ 0,0001 \end{bmatrix}$	$\begin{bmatrix} 0,0060 \\ 0,0021 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -0,0025 \\ 0,0007 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -0,0072 \\ -0,0071 \\ 1 \end{bmatrix}$	0,1811
EoRP	$\begin{bmatrix} 1 \\ 0,0006 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0,0007 \\ 0,0012 \end{bmatrix}$	-	-	$\begin{bmatrix} 0,0009 \\ -0,0009 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -0,0004 \\ -0,0008 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0,0091 \\ -0,0050 \\ 1 \end{bmatrix}$	0,1666
eRANSAC	$\begin{bmatrix} 1 \\ -0,0001 \\ -0,0001 \end{bmatrix}$	-	$\begin{bmatrix} -0,0001 \\ 1 \\ -0,0003 \end{bmatrix}$	$\begin{bmatrix} -0,0001 \\ 1 \\ 0,0003 \end{bmatrix}$	$\begin{bmatrix} -0,0005 \\ -0,0001 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -0,0002 \\ 0,0004 \\ 1 \end{bmatrix}$	-	0,0198

Da bi rezultati sve tri metode mogli biti i kvantitativno upoređeni uvedena je određena metodologija poređenja estimiranih parametara prepoznatih cilindara i ravni. Za ovu svrhu definisani se pojmovi srednje devijacije radijusa prepoznatih cilindara i srednje devijacije normala prepoznatih ravni.

Srednje odstupanje vrednosti radijusa tj. parametar ARD može se definisati kao:

$$ARD = \frac{1}{n_c} \sum_{i=1}^{n_c} |DR_i - ER_i|, \quad (7.16)$$

gde je DR_i dizajnirana vrednost radijusa i -og ekstrahovanog cilindra dok je ER_i njegova procenjena vrednost dobijena metodom prepoznavanja. Takođe, n_c predstavlja broj pravilno prepoznatih cilindara.

Srednje odstupanje normala (parametar AND) može se predstaviti kao:

$$AND = \frac{1}{n_p} \sum_{i=1}^{n_p} \arccos \left(\frac{\mathbf{DN}_i \cdot \mathbf{EN}_i}{|\mathbf{DN}_i| \cdot |\mathbf{EN}_i|} \right), \quad (7.17)$$

gde DN_i predstavlja dizajnirani pravac normale i -te ekstrahovane ravni, EN_i je njena vrednost procenjena metodom, dok je n_p ukupan broj prepoznatih ravni. Treba naglasiti da ocene ARD i AND ne predstavljaju mere preciznosti estimacije realnih radijusa i ravni zato što sama preciznost zavisi pre svega od kvaliteta procesa izrade fizičkog objekta, kao i procesa skeniranja. Međutim, kako su sve metode koje se upoređuju procesirale identičan oblak, ovako definisane ocene (vrednosti devijacija) se mogu iskoristiti za njihovu uporednu analizu.

Vrednosti parametara ARD i AND prikazane su u tabelama 7.3 i 7.4 pored projektovanih vrednosti parametara prepoznatih cilindara i ravni za svaku metodu pojedinačno. ARD vrednosti pokazuju da je predložena metoda izvršila bolju estimaciju parametara prepoznatih cilindara od eRANSAC metode. Ipak, upoređujući sve tri metode, EoRP metoda poseduje najbolju ARD vrednost. Međutim, ova vrednost je dobijena sa jednim zahtevanim cilindrom manje pošto ga EoRP metoda nije prepoznala u svom procesu. Daljom analizom pojedinačnih vrednosti radijusa može se uočiti da predložena metoda pokazuje nešto lošije performanse pri estimaciji cilindara malog poluprečnika dok je kod nešto većih cilindara ravnopravna sa EoRP algoritmom. Interesantno je istaći da su srednje vrednosti devijacija prepoznatih radijusa pri korišćenju sve tri metode uporedive sa deklarisanom rezolucijom 3D skenera (0,017-0,481 mm) kojim je oblak test objekta generisan. U konkretnom slučaju srednja vrednost gustine tačaka oblaka je 0,1834 mm što predstavlja višu vrednost od vrednosti za ARD u slučajevima upotrebe predložene i EoRP metode. Nažalost, proizvođač korišćenog skenera [24] nije definisao samu tačnost generisanja tačaka pa dalja analiza u ovom pravcu nije moguća. Takođe, nisu postojali uslovi provere fizičkih dimenzija skeniranog objekta koje su ostvarene mašinskom obradom.

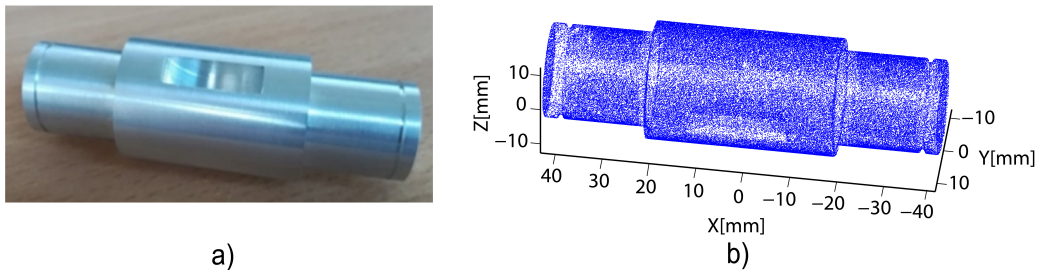
Posmatrajući ekstrakciju ravni, kao što je već rečeno, predložena metoda je prepoznala sve zahtevane ravni objekta dok druge dva metode to nisu učinile. Vrednosti srednjih devijacija normala (AND) za sve tri metode pokazuju da je predložena metoda pokazala preciznost istog nivoa kao EoRP, dok je u ovom slučaju eRANSAC pokazala najbolje performanse. Međutim, potrebno je naglasiti da su u kalkulaciju AND vrednosti za alternativne metode uvrštene samo ravni koje su one prepoznale što je manji broj od zahtevanog, odnosno broja ravni koji je predložena metoda prepoznala. Vrednosti srednjih devijacija normala pokazuju veoma zadovoljavajuće performanse za sve tri metode i ne smatraju se kritičnim za procese prepoznavanja. Ovo se može objasniti time da sve one, pri estimaciji parametara, koriste neke od mnogo puta dokazanih i opšte poznatih tehnika regresije ravni.

Na kraju, performanse korišćenih metoda možemo uporediti i po još jednom parametru, a to je vreme procesiranja. Iz ove analize ipak moramo izuzeti eRANSAC zbog njene profesionalne komercijalne izvedbe koja nije uporediva sa načinom implementacije ostale dve metode. EoRP i predložena metoda su implementirane u Matlab okruženju i korišćeni su na 3,1GHz Intel(R) Core i5-2400 platformi. Vreme procesiranja oblaka sa slike 7.8(b) korišćenjem ove dva metode pri dobijanju već prezentovanih rezultata je prikazano zajedno sa grafičkom reprezentacijom rezultata na slici 7.9. Vreme procesiranja je u slučaju upotrebe EoRP metode čak 448,6 s, dok je u slučaju predložene metode znatno povoljnije i iznosi 60,6 s. Ovime se pokazuje značajna razlika između ove dve metode i određena značajna prednost predložene. Takođe, ova informacija može biti od značaja za razmatranje eventualne upotrebe metode prepoznavanja kod sistema koji zahtevaju rad u realnom vremenu. Treba naglasiti da su predstavljena vremena procesiranja data kao srednje vrednosti pet uzastopnih izvršavanja algoritma u istim uslovima. Razlika vremena u pojedinačnim izvršavanjima se javlja zbog prisutnosti procedura nasumičnog izbora uzoraka u određenim delovima algoritama kao i različitih pozadinskih procesa koji se izvršavaju na računaru tokom pokretanja algoritma.

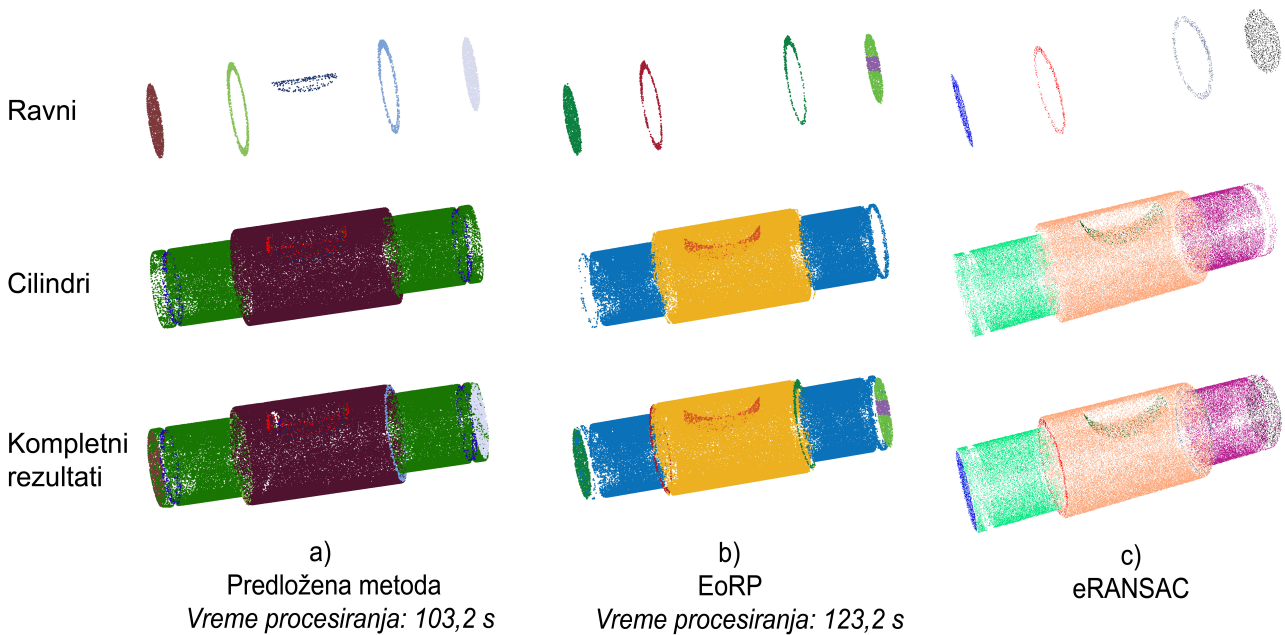
7.5.2. Prepoznavanje primitiva vratila

U drugoj studiji slučaja razmatrano je prepoznavanje primitiva jednog objekta koji je po tipu široko rasprostranjen u mašinskoj industriji. Reč je o vratilu (slika 7.10(a)) koje sadrži 10 ravni i 4 cilindra i koje je projektovano i izrađeno isključivo u svrhe ovog istraživanja. Nestruktuirani oblak ovog objekta (slika 7.10(b)) je generisan skeniranjem *Range Vision Pro* skenerom [144] i sadrži 97.688 tačaka,

dok srednja gustina tačaka iznosi $\sigma=0,1358$ mm. Deklarisana tačnost upotrebljenog 3D skenera je 0,06 mm, dok je opseg rezolucije skeniranja 0,03-0,19 mm. Treba naglasiti da geometrija vratila i ograničenja načina skeniranja nisu dozvolili generisanje velikog broja tačaka u oblastima bokova žleba za segmentni klin i u kanalima za prstenasti uskočnik. Reč je o 4 ravni i 2 cilindra koji su opisani znatno manjim brojem tačaka u poređenju sa ostalim regijama oblaka. Kao i u prethodnoj studiji slučaja i u ovoj je oblak vratila procesiran korišćenjem sve tri raspoložive metode kako bi se izvršilo njihovo poređenje.



Slika 7.10: Vratilo: a) fotografija realnog objekta, b) oblak tačaka.



Slika 7.11: Grafička reprezentacija rezultata prepoznavanja primitiva iz oblaka tačaka sa slike 7.10(b): a) Predložena metoda, b) EoRP, c) eRANSAC.

Rezultati procesiranja oblaka vratila predloženom metodom su grafički predstavljani na slici 7.11(a), dok su vrednosti pragova korišćenih u tom slučaju dati u tabeli 7.5. Predložena metoda je izvršila prepoznavanje 5 ravni i 4 cilindra. Kao što je i očekivano, metoda nije prepoznala vrlo male bočne ravni kanala za prstenaste uskočnike koji su po dimenzijama jednaki i nalaze se na oba kraja vratila. Međutim, vrlo uspešno je izvršila potrebno prepoznavanje cilindra unutar tih kanala (iako je reč o dva kanala na dva kraja vratila to je geometrijski posmatrano jedan cilindar). Takođe, nije uspela da prepozna i jednu bočnu ravan žleba za klin.

Na slici 7.11(b) su prikazani rezultati upotrebe EoRP metode, dok su vrednosti korišćenih pragova dati u tabeli 7.6. Ova metoda je korektno prepoznala 4 ravni i samo 3 cilindra. Nije uspela da prepozna ni jedan primitiv kanala za prstenaste uskočnike, a ni bočne ravni žleba za klin. Sa druge strane, ekstatovana je jedna dodatna suvišna ravan na čeonj površini objekta (na slici obeležena roze).

Rezultati treće metode (eRANSAC) su grafički predstavljeni na slici 7.11(c). Pri procesiranju su korišćeni isti parametri kao u prethodnoj studiji: $\epsilon = 0,1$, $\beta = 0,9$, $\alpha = 25^\circ$, $\tau = 300$ i verovatnoća nepronalaženja primitiva od 0,01. Metoda je ekstrahovala 4 ravni i 4 cilindra - dva dela istog cilindra su prepoznata kao dva odvojena cilindra sa posebnim parametrima (na slici obeleženi zelenom i ljubičastom bojom). Takođe, ni ova metoda nije uspeła da prepozna kritične primitive kao što su bočne ravni žleba za klin i primitivi kanala za prstenaste uskočnike.

Tabela 7.5: Predložena metoda: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.10(b)

Prag	ts_s	α	β	m_{ic}	m_c
Vrednost praga	$0,5 \cdot 10^{-19}$	3	0.05	3	10

$\sigma=0,1358$ mm

Tabela 7.6: EoRP: Vrednosti pragova za procesiranje oblaka tačaka sa slike 7.10(b)

Prag	$k-nn$	α^*	$\beta^*(rad)$	θ	c_{sc}	h_1	h_2	ϵ	α^{**}	τ	cc	β^{**}
Vrednost praga	20	10	0,95	0,99	2	0,01	σ	0,5	20	300	10	3σ

*-za cilindre; **-za ravni; $\sigma=0,1358$ mm

Tabela 7.7: Estimirani radijusi cilindara [mm] za oblak tačaka sa slike 7.10(b)

Metoda	Dizajnirani radijusi [mm]				ARD[mm]
	10	12,5	12,2	9,25	
Predložena	10,0089	12,5009	12,2116	9,2300	0,0103
EoRP	10,0054	12,5016	12,1789	-	0,0070
eRANSAC	9,9516	12,5079	12,0877	-	0,0421

Tabela 7.8: Estimirane normale ravni za oblak tačaka sa slike 7.10(b) (vratilo bez kanala za prstenasti uskočnik)

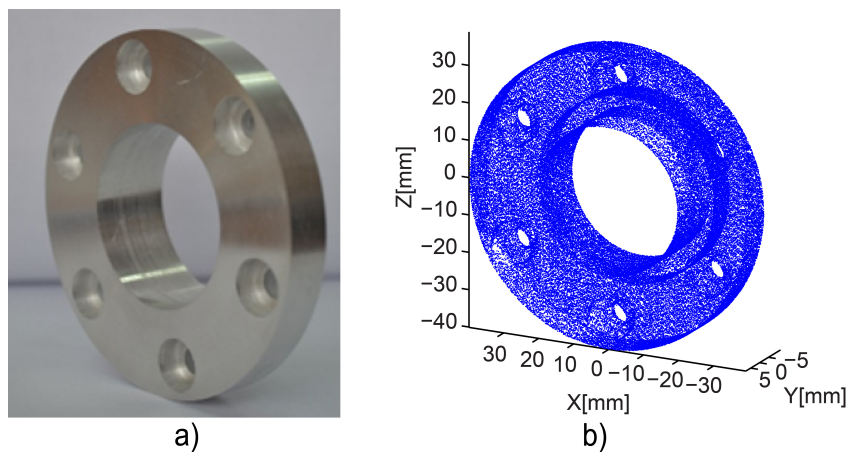
Metoda	Dizajnirane normale ravni						AND [°]
	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	
Predložena	$\begin{bmatrix} 1 \\ 0,0034 \\ 0,0011 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -0,0010 \\ 0,0013 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -0,0026 \\ -0,0055 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -0,0029 \\ -0,0126 \end{bmatrix}$	$\begin{bmatrix} 0,0005 \\ 1 \\ 0,0012 \end{bmatrix}$	-	0,2925
EoRP	$\begin{bmatrix} 1 \\ 0,0008 \\ 0,0032 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0,1769 \\ 0,1992 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0,0055 \\ -0,0074 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -0,0100 \\ -0,0072 \end{bmatrix}$	-	-	4,0852
eRANSAC	$\begin{bmatrix} 1 \\ 0,0067 \\ 0,0011 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -0,0005 \\ 0,0012 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -0,0005 \\ -0,0040 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0,0035 \\ -0,0089 \end{bmatrix}$	-	-	0,3106

Vrednosti estimiranih radijusa i normala ravni dobijenih korišćenjem sve tri metode, kao i vrednosti za *ARD* i *AND* su date u tabelama 7.7 i 7.8, respektivno. Devijacije radijusa cilindara (*ARD*) su izrazito male u sva tri slučaja. Ohrabrujuće je što su ove vrednosti značajno manje i od same tačnosti upotrebljenog 3D skenera koja iznosi pomenutih 0,06 mm. Ipak, mora se istaći da je EoRP metoda

postigla najbolje rezultate u ovom domenu, ali nije izdvojila jedan zahtevani cilindar. Sa druge strane vrednosti za *AND* favorizuju predloženu metodu, ali se mora zapaziti i značajna devijacija kod upotrebe EoRP algoritma. Kao što je ranije istaknuto, sve tri metode nisu ekstrahovale bočne ravni kanala za prstenaste uskočnike, ali je predložena metoda jedina uspela da prepozna njihov cilindar na dnu, kao i jednu ravan žleba segmentnog klina. Imajući u vidu navedeno, predložena metoda je i u kvantitativnom i u kvalitativnom poređenju pokazala vrlo dobre rezultate. Takođe, ne treba zaboraviti i uporednu analizu vremena izvršavanja pri čemu je predloženoj metodi bilo potrebno 103,2 s, naspram 123,2 s koliko je potrošila EoRP metoda da izdvoji čak i manji broj primitiva. I ovoga puta iz poznatih razloga analiza vremena izvršavanja nije uzeta u obzir u slučaju eRANSAC metode.

7.5.3. Prepoznavanje primitiva prirubnice

Slično prethodnom slučaju i u ovom se razmatra procesiranje oblaka tačaka jednog mašinskog elementa. U pitanju je prirubnica prikazana na slici 7.12(a). Ovaj objekat je konstruisan i izrađen za potrebe istraživanja navedenih u ovom radu ali poseduje tipičan oblik realno primenjivanih elemenata u industriji. Struktura ove prirubnice, koja sadrži 6 simetrično radijalno raspoređenih otvora za montažu zavrtnjevima, je sačinjena od 4 ravni i čak 15 cilindara. Oblak tačaka koji reprezentuje navedeni objekat (slika 7.12(b)) je takođe generisan upotrebom *Range Vision Pro* 3D skenera [144] i sadrži 88.178 tačaka, dok srednja gustina tačaka iznosi $\sigma=0,2125$ mm. S obzirom na to da je deklarisan opseg rezolucije skeniranja upotrebljenog skenera pomenutih 0,03-0,19 mm, može se primetiti da je srednja gustina konkretnog oblaka nešto viša. Ovo se može opravdati manjim brojem generisanih tačaka na površinama unutar velikog broja upuštenih otvora koji postoje na prirubnici. Ovaj manji broj tačaka u pomenutim oblastima je rezultat same geometrije objekta, ali i ograničenja optičke metode skeniranja. Opisani oblak tačaka prirubnice, kao i u prethodnim slučajevima, procesiran je upotrebom sve tri metode (predložene, EoRP i eRANSAC). Uporedni rezultati su prikazani na slici 7.13, kao i u tabelama 7.11 i 7.12.

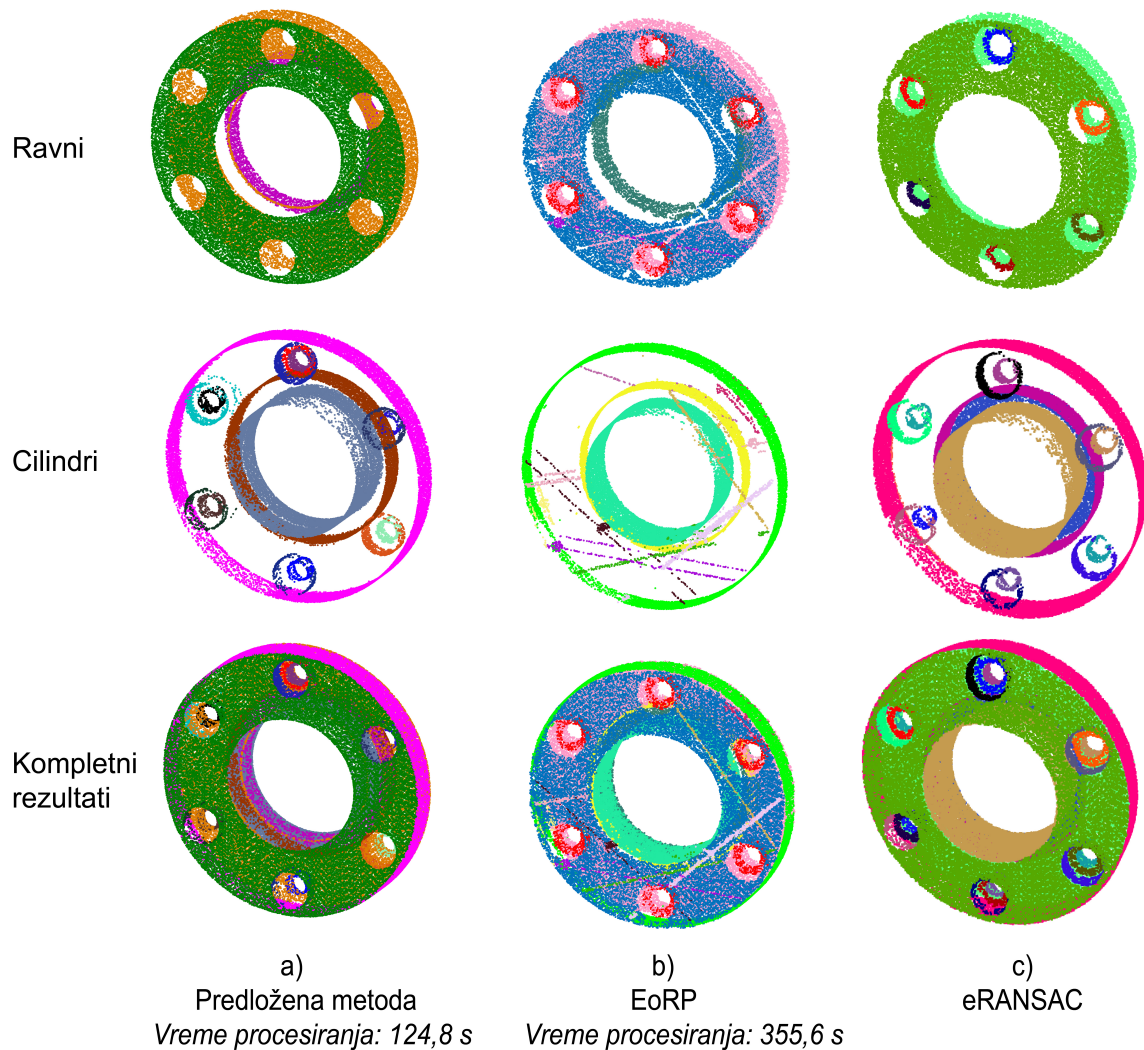


Slika 7.12: Prirubnica: a) fotografija realnog objekta, b) oblak tačaka.

Posmatranjem slike 7.13(a) moguće je utvrditi da je predloženi algoritam prilikom procesiranja oblaka tačaka prirubnice uspešno prepoznao i ekstrahovao 3 ravni i 14 cilindara. Međutim, algoritam nije prepoznao donju ravan upuštenih simetrično raspoređenih otvora. Sa druge strane, jedan deo te površine (klaster označen crvenom bojom na slici 7.13(a)) je prepoznat kao cilindar velikog poluprečnika. Vrednosti pragova korišćenih u navedenom slučaju procesiranja oblaka prirubnice predloženom metodom prikazane su u tabeli 7.9.

EoRP metoda je pri procesiranju oblaka prirubnice pokazala dosta slabosti pri prepoznavanju cilindara manjih radijusa. U ovom slučaju prepoznala je samo 4 ravni i 3 cilindra uz detektovanje većeg broja suvišnih (nepostojećih) cilindara u zonama ravnih površina. Sa druge strane, nije prepoznala ni

jedan od cilindara sa radijusima 3 i 5,75 mm. Pragovi korišćeni u ovom slučaju procesiranja su dati u tabeli 7.10. Posmatranjem potrebnih pragova, ali i same strukture EoRP algoritma [94], odnosno njegovog segmenta posvećenog cilindrima [95], može se zaključiti da postoji određeni problem u balansu između vrednosti predefinisano malog broja tačaka u inicijalnim klasterima i vrednosti praga za ocenu validnosti prepoznate cilindrične površine c_{sc} (engl. *score threshold*). Ukoliko se postavi manja vrednost za veličinu inicijalnih klastera, čime se cilja na pronalaženje cilindara malih radijusa koji su opisani manjim brojem tačaka, algoritam izdvoji veoma veliki broj dodatnih nepostojećih cilindara za određenu vrednost praga c_{sc} . Ako se pak c_{sc} smanji, algoritam iz konačnih rezultata prepoznavanja odstrani i neke od potrebnih cilindara velikog prečnika. Ovaj problem svakako zahteva dublju analizu, ali je jasno da se ovim problemom onemogućava intuitivno i lako postavljanje vrednosti pragova i da je to upravo jedan od glavnih razloga loših rezultata prepoznavanja primitiva prirubnice EoRP metodom.



Slika 7.13: Grafička reprezentacija rezultata prepoznavanja primitiva iz oblaka tačaka sa slike 7.12(b): a) Predložena metoda, b) EoRP, c) eRANSAC.

Na kraju, oblak prirubnice bio je procesiran i eRANSAC metodom. Ova metoda je adekvatno prepoznala 3 ravni, ali i sve cilindre. Ipak, zbog same prirode algoritma sve ravni na dnu upuštenih otvora su prepoznate nezavisno i sa procenjenim nezavisnim vrednostima parametara. Takođe, jedna čeona ravan (obeležena plavom bojom na slici 7.13(c)) prepoznata je kao cilindar veoma velikog radijusa (2098,7449 mm) što se može smatrati pojavom pomenute konfuzije modela. Vrednosti pragova

za procesiranje u ovom slučaju su iste kao u prethodne dve studije ($\epsilon = 0,1$, $\beta = 0,9$, $\alpha = 25^\circ$), ali je ovoga puta zbog postojanja cilindara malog prečnika (u odnosu na gabarite objekta) minimalni broj tačkaka klastera postavljen na $\tau = 100$. Verovatnoća nepronalaženja primitiva je i ovoga puta postavljena na 0,01.

Tabela 7.9: Predložena metoda: Vrednosti pragova za procesiranje oblaka tačkaka sa slike 7.12(b)

Prag	ts_s	α	β	m_{ic}	m_c
Vrednost praga	$0,5 \cdot 10^{-18}$	1	0,05	4	10

$\sigma=0,2125$ mm

Tabela 7.10: EoRP: Vrednosti pragova za procesiranje oblaka tačkaka sa slike 7.12(b)

Prag	$k-nn$	α^*	$\beta^*(rad)$	θ	c_{sc}	h_1	h_2	ϵ	α^{**}	τ	cc	β^{**}
Vrednost praga	20	10	0,95	0,99	5	0,01	σ	0,8	30	100	5	3σ

*-za cilindre; **-za ravni; $\sigma=0,2125$ mm

Tabela 7.11: Estimirani radijusi cilindara [mm] za oblak tačkaka sa slike 7.12(b)

Metoda	Dizajnirani radijusi [mm]															ARD[mm]
	39,5	25	20	3	3	3	3	3	3	3	5,75	5,75	5,75	5,75	5,75	
Predložena	39,4953	25,0781	19,9332	2,9805	3,0068	3,0370	3,0353	3,0454	3,0118	5,7937	5,7465	5,8130	5,7910	5,7375	5,8132	0,0355
EoRP	39,5256	25,0703	19,9405	-	-	-	-	-	-	-	-	-	-	-	-	0,0518
eRANSAC	39,5837	25,1332	19,9619	2,9936	3,0326	3,0058	3,0166	2,9858	3,0244	5,7418	5,7865	5,6550	5,8052	5,7833	5,8236	0,0438

Tabela 7.12: Estimirane normale ravni za oblak tačkaka sa slike 7.12(b)

Metoda	Dizajnirane normale ravni				AND [°]
	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	
Predložena	$\begin{bmatrix} 0,0005 \\ 1 \\ 0,0001 \end{bmatrix}$	$\begin{bmatrix} 0,0026 \\ 1 \\ -0,0017 \end{bmatrix}$	$\begin{bmatrix} 0,0005 \\ 1 \\ 0,0001 \end{bmatrix}$	-	0,0788
EoRP	$\begin{bmatrix} 0,0058 \\ 1 \\ 0,0017 \end{bmatrix}$	$\begin{bmatrix} 0,0006 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0,0132 \\ 1 \\ -0,0228 \end{bmatrix}$	$\begin{bmatrix} -0,0099 \\ 1 \\ 0,0064 \end{bmatrix}$	0,6413
eRANSAC	$\begin{bmatrix} 0,0026 \\ 1 \\ -0,0013 \end{bmatrix}$	$\begin{bmatrix} 0,0005 \\ 1 \\ -0,0002 \end{bmatrix}$	-	$\begin{bmatrix} 0,0068 \\ 1 \\ 0,0018 \end{bmatrix}$	0,2001

Na osnovu pojedinačno diskutovanih rezultata prepoznavanja sve tri metode, može se zaključiti da je oblak prirubnice bio znatno zahtevniji za procesiranje u poređenju sa prethodna dva slučaja. eRANSAC metoda je prepoznala sve primitive, ali sa jednom konfuzijom modela pri čemu je ravan prepoznata kao cilindar. EoRP metoda je prepoznala samo tri zahtevana cilindra uz mnoštvo drugih nepotrebnih površina (pre svega ravni). Sa druge strane, predložena metoda nije prepoznala ni jedan suvišni primitiv.

Predložena metoda postigla je najbolje *ARD* i *AND* vrednosti (tabele 7.11 i 7.12), ali, kao što je već navedeno, ipak nije prepoznala jednu zahtevanu ravan. Treba naglasiti i da je *AND* vrednost u slučaju eRANSAC metode izračunata na osnovu (jedne) najbolje od 6 prepoznatih ravni sa dna upuštenih otvora. Takođe, potrebno je uzeti u obzir i vreme procesiranja predložene i EoRP metode gde je predložena metoda svoje primitive prepoznala za 124,8 s, dok je EoRP algoritmu bilo potrebno čak 355,6 s. Sve navedene činjenice ukazuju da je u slučaju procesiranja oblaka prirubnice predloženi metod pokazao određene prednosti u odnosu na druga dva algoritma. I u ovom slučaju, bitno je istaći da su srednje vrednosti devijacija prepoznatih cilindara za sve tri metode značajno niže od tačnosti upotrebljenog 3D skenera (0,06 mm).

Na samom kraju treba naglasiti i da su tokom istraživanja na temu prezentovanu u ovom poglavlju određeni rezultati objavljeni u [133, 134, 135, 136, 145].

8. Optimizacija parametara algoritama za prepoznavanje geometrijskih primitiva

Dosadašnja referentna istraživanja na polju procesiranja oblaka tačaka (pregled dat u poglavlju 3) su uglavnom zasnovana na polu-automatskim metodama koje uz pomoć određenog nivoa korisničke asistencije postižu zapažene rezultate u različitim uslovima. Međutim, razvojem tehnika veštačke inteligencije (AI) poslednjih godina izražena je tendencija razvoja autonomnih sistema u velikom broju različitih oblasti tehnike. Ovaj trend (ali i realna potreba za razvojem autonomnih sistema) nije zaobišao ni oblasti reverznog inženjerstva i procesiranja oblaka tačaka. Iako su oblasti primene procesa 3D digitalizacije veoma široke (poglavlje 2.4), kao referentan primer u ovom slučaju mogu se navesti inteligentni mobilni tehnički sistemi koji su sve više zastupljeni u mašinskoj industriji [146, 147]. Svedoci smo postojanja potreba pravovremenog i tačnog detektovanja i prepoznavanja objekata u okruženju raznih mobilnih mašina, robota i vozila (jednom rečju mobilnih agenata) kako bi se obezbedio njihov neophodni visoki stepen autonomije u svakodnevnom radu i interakciji sa drugim objektima. Ovakvim sistemima je neophodna određena vrsta 3D digitalizacije okruženja u realnom vremenu pri čemu se, u poređenju sa kamerama i tehnikama obrade slike, prednost daje upotrebi različitih vrsta laserskih daljinomera i radara. Na taj način grubo definisano okruženje mobilnog agenta se prevodi u 3D oblake tačaka. Na osnovu navedenog može se lako zaključiti da dalji razvoj i povećanje stepena autonomije mobilnih sistema se ne može obezbediti bez razvoja i samih tehnika procesiranja oblaka tačaka.

U poglavljima 6 i 7 ovog rada predstavljena su istraživanja na temu procesiranja oblaka tačaka u cilju prepoznavanja geometrijskih primitiva (određenih vrsta kvadratika i ravni). Iako automatske, ove metode kao i analizirane alternativne metode, ipak zahtevaju asistenciju korisnika u vidu zadavanja i podešavanja vrednosti pragova koji se kasnije koriste u algoritmu za potrebe odlučivanja. Kako pragovi u značajnoj meri utiču na ishod rezultata pre svega segmentacije, bitno je izvršiti dobar izbor njihovih vrednosti u svakom konkretnom slučaju. Korisnik algoritma može postaviti vrednosti pragova na osnovu određenih preporuka ili pak izvršiti nekoliko procesiranja različitim vrednostima i izabrati najbolje. Ovakvu iterativnu metodu izbora vrednosti nazivamo definisanje pragova pretraživanjem.

Zbog svega navedenog sve do sada predložene metode možemo definisati kao automatske, ali ih ne možemo definisati i kao autonomne (samostalan rad bez interakcije sa korisnikom). Međutim, ukoliko bi se razvila i upotrebila dopunska metoda za inteligentno automatsko postavljanje vrednosti potrebnih pragova, sve navedene metode prepoznavanja primitiva bi se mogle koristiti u određenom autonomnom režimu. Ovo naravno ne predstavlja trivijalan problem i zavisi od velikog broja faktora kao i od samih metoda prepoznavanja. Zbog pojašnjenja navedenih tvrdnji u ovom poglavlju biće predstavljena jedna metoda automatskog izbora pragova za algoritam prepoznavanja kvadratika iz poglavlja 6. Ova dopunska metoda je zasnovana na upotrebi veštačkih neuronskih mreža (engl. *Artificial Neural Networks*), odnosno njihovom korišćenju za inteligentan izbor vrednosti pragova na osnovu prethodnog obučavanja za tipične situacije. Takođe, biće pokazani i rezultati upotrebe ovako postavljene autonomne metode u slučaju procesiranja simuliranog i realnog oblaka tačaka i biće izneti zaključci o mogućnosti generalizacije samog modela prepoznavanja.

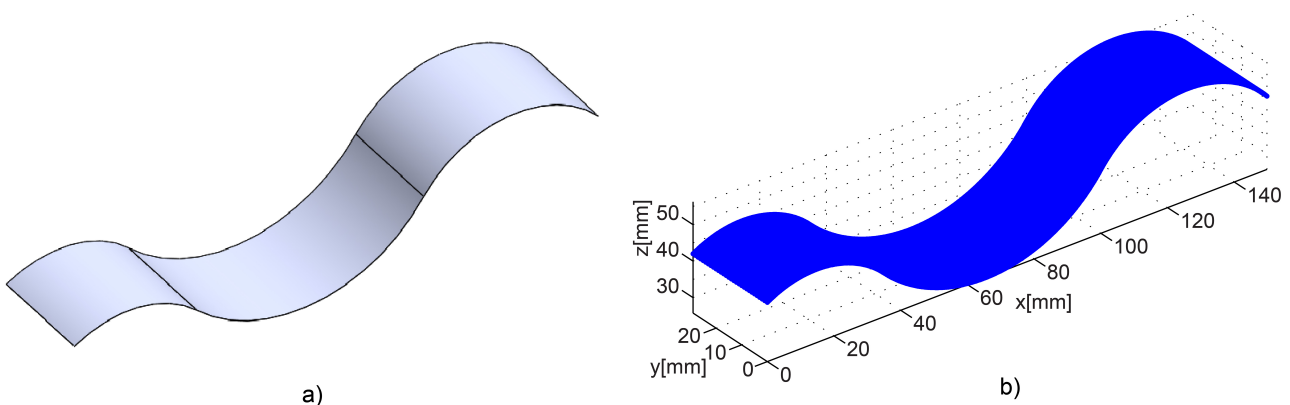
8.1. Uticaj pragova na ishod rezultata segmentacije

Kao što je i ranije naglašeno, jedan od glavnih procesa metoda prepoznavanja primitiva predstavljenih u ovom radu je svakako segmentacija. Samim tim najveći broj značajnih pragova svih metoda vezan je upravo za ovaj proces. Automatskim odabirom vrednosti pragova, metode prepoznavanja mogle bi se potpuno automatizovati što bi doprinelo njihovom potencijalu za primenu u autonomnim sistemima.

Dosadašnji razvoj postupaka segmentacije prvobitno je bio vezan za 2D obradu slike. Značajan broj istraživanja bio je usmeren na problem određivanja pragova u segmentaciji dvodimenzionalnih

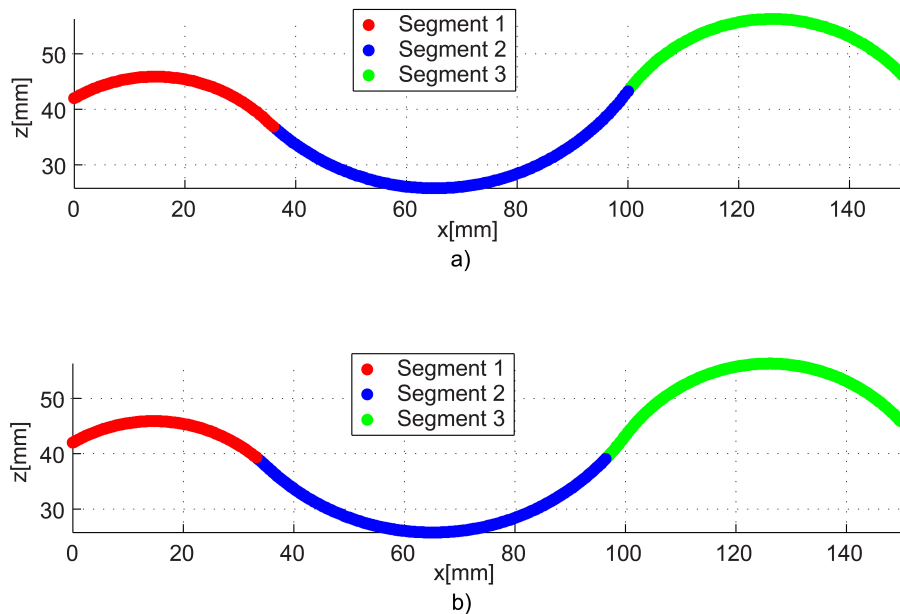
slika. Razvijen je određen broj tehnika među kojima se mogu izdvojiti tehnike zasnovane na maksimalnoj disperziji između klasa – tzv. Otsu tehnika određivanja praga [148] i tehnike zasnovane na entropiji [149] uz optimizaciju korišćenjem različitih algoritama među kojima su algoritam simuliranog kaljenja (engl. *Simulated Annealing*) [150] ili algoritam na bazi inteligencije roja čestice (engl. *Particle Swarm Optimization*) [151]. Sa druge strane, segmentacija 3D oblaka tačaka predstavlja relativno novi istraživački pravac pa problem određivanja pragova u ovoj oblasti nije dovoljno razmatran. Zbog toga veliki broj metoda koje se bave oblacima tačaka pokušava da problem 3D procesiranja, korišćenjem različitih tehnika projektovanja i geometrijskih transformacija, svede na 2D prostor i da onda iskoriste neke od dokazanih automatizovanih algoritama obrade tačaka u ravni. Kao primer može se navesti metoda iz [95] u kojoj je u jednom koraku algoritma upotrebljen Otsu princip automatskog određivanja pragova za formiranje binarne slike projekcije tačaka cilindra na određenu ravan. Ovo je izvršeno u cilju pripreme projekcije za određivanje poluprečnika bazne kružnice (cilindra) RANSAC metodom.

Tema ovog poglavlja je metoda automatskog određivanja pragova za algoritam prepoznavanja jedne vrste kvadraka iz struktuiranih oblaka tačaka koji je detaljno predstavljen u poglavlju 6. Osnov ovog algoritma predstavlja procesiranje skeniranih kontura oblaka u cilju njihove segmentacije čime se dolazi do podele oblaka na regione koji reprezentuju geometrijski različite delove površina skeniranog objekta. Zbog mogućnosti presegmentiranja kontura u prisustvu značajnog nivoa šuma u oblaku, algoritam segmentacije je dopunjen procesom grupisanja presegmentiranih regiona. Pseudokodovi procesa inicijalne segmentacije i naknadnog grupisanja su prikazani u poglavlju 6.2 na slikama 6.1 i 6.2, respektivno.



Slika 8.1: Sintetizovana površina koja se sastoji od GI kontinualna tri cilindra: a) CAD model objekta; b) sintetizovan struktuirani oblak tačaka.

Na osnovu datih postavki metode i pomenutih pseudokodova zaključuje se da je za uspešnu primenu celokupnog algoritma prepoznavanja potrebno prethodno podesiti vrednosti tri nezavisna praga. Dva praga (t_{rc} i t_{ov}) koriste se u procesu inicijalne segmentacije, po jedan za svaki slučaj singularnosti sistema (recipročni faktor uslovljenosti i sopstvene vrednosti), dok se treći (t_{gr}) koristi u procesu grupisanja (spajanja) prekobrojnih segmenata. Podsećanja radi, u svakoj iteraciji procesa inicijalne segmentacije se grupi do tada izdvojenih tačaka tekuće konture dodaje naredna tačka i proveravaju se postavljeni uslovi singularnosti. Ukoliko su vrednosti recipročnog faktora uslovljenosti informacione matrice sistema \mathbf{S} i sopstvene vrednosti matrice $\mathbf{S}^{-1}\mathbf{C}$ u određenoj korelaciji (više ili niže) od pomenutih pragova t_{rc} i t_{ov} , može se zaključiti da se matrica \mathbf{S} udaljava od svoje singularnosti, odnosno da poslednje dodata tačka ne pripada tekućem skupu tačaka koje predstavljaju eliptični segment konture, odnosno deo neke elipse. Ta tačka se u tom trenutku proglašava granicom između dva eliptična segmenta i postaje ujedno i prva tačka narednog segmenta. Nakon formiranja svih inicijalnih segmenata, međusobnom proverom svaka dva susedna na osnovu t_{gr} donosi se odluka o njihovom spajanju. Pravilno izabrane vrednosti sva tri navedena praga imaju značajnu ulogu u konačnim rezultatima primene metode.



Slika 8.2: Segmentirana kontura oblaka sa slike 8.1: a) pragovi koje je korisnik odredio metodom pretraživanja; b) promjenjeni pragovi u cilju ilustracije osetljivosti segmentacije.

Radi ilustracije uticaja pomenutih pragova na segmentaciju uporebljen je jedan sintetizovan oblak prikazan na slici 8.1(b). Sintetizovanje ovog oblaka izvršeno je upotrebom projektovane površine (slika 8.1(a)), koja se sastoji od tri G1 kontinualna cilindra, i njenog struktuiranja metodom iz poglavlja 4. Da bi se sintetisani oblak približio realnom skeniranom oblaku, on je kontaminiran šumom sa odnosom signala i šuma (SNR) od 120 dB. Raster tačaka duž x ose oblaka je ekvidistantan i iznosi 0,2 mm čime se dobija po 750 tačaka u svakoj konturi. Iz oblaka je izdvojena jedna kontura koja predstavlja pojedinačnu skeniranu liniju površine objekta i, u skladu sa geometrijom objekta, sastoji se od tri eliptična segmenta. Navedena kontura je procesirana algoritmom iz poglavlja 6 i rezultati su grafički prikazani na slici 8.2. Slika 8.2(a) prikazuje rezultate segmentacije za vrednosti pragova: $t_{rc} = 0, 15$, $t_{ov} = 10^{-13}$ i $t_{gr} = 30$. Ove vrednosti pragova su izabrane pretraživanjem (ručno) i obezbedile su segmentiranje konture na potrebna tri segmenta. Detektovane prelazne tačke se nalaze na pozicijama 176 i 500 počev od prve tačke konture. Imajući u vidu da su projektovani prelazi između segmenata 175 i 500, može se zaključiti da je u ovom slučaju greška segmentacije zanemariva i iznosi jedan uzorak. Međutim, upotrebljeni algoritam je u slučaju procesiranja ove konture prilično osetljiv na promenu vrednosti pragova. Na slici 8.2(b) prikazani su rezultati segmentacije za blago izmenjene vrednosti pragova koji iznose: $t_{rc} = 0, 20$, $t_{ov} = 1, 1 \cdot 10^{-13}$ i $t_{gr} = 35$. Generisane vrednosti prelaza su 167 i 483 pa je greška segmentacije značajna i iznosi 8 uzoraka na prvom i 17 uzoraka na drugom prelazu. Navedeni rezultati jasno ukazuju da mala razlika vrednosti pragova ima značajan uticaj na rezultate segmentacije. Iz navedenog poređenja dva slučaja sa blago različitim parametrima može se zaključiti da je za pravilnu segmentaciju neophodno tačno izabrati vrednosti pragova u zavisnosti od oblika konture i nivoa šuma u skeniranoj liniji (signalu). U osnovnom algoritmu (poglavlje 6.2) izbor vrednosti pragova je heuristički vršio korisnik.

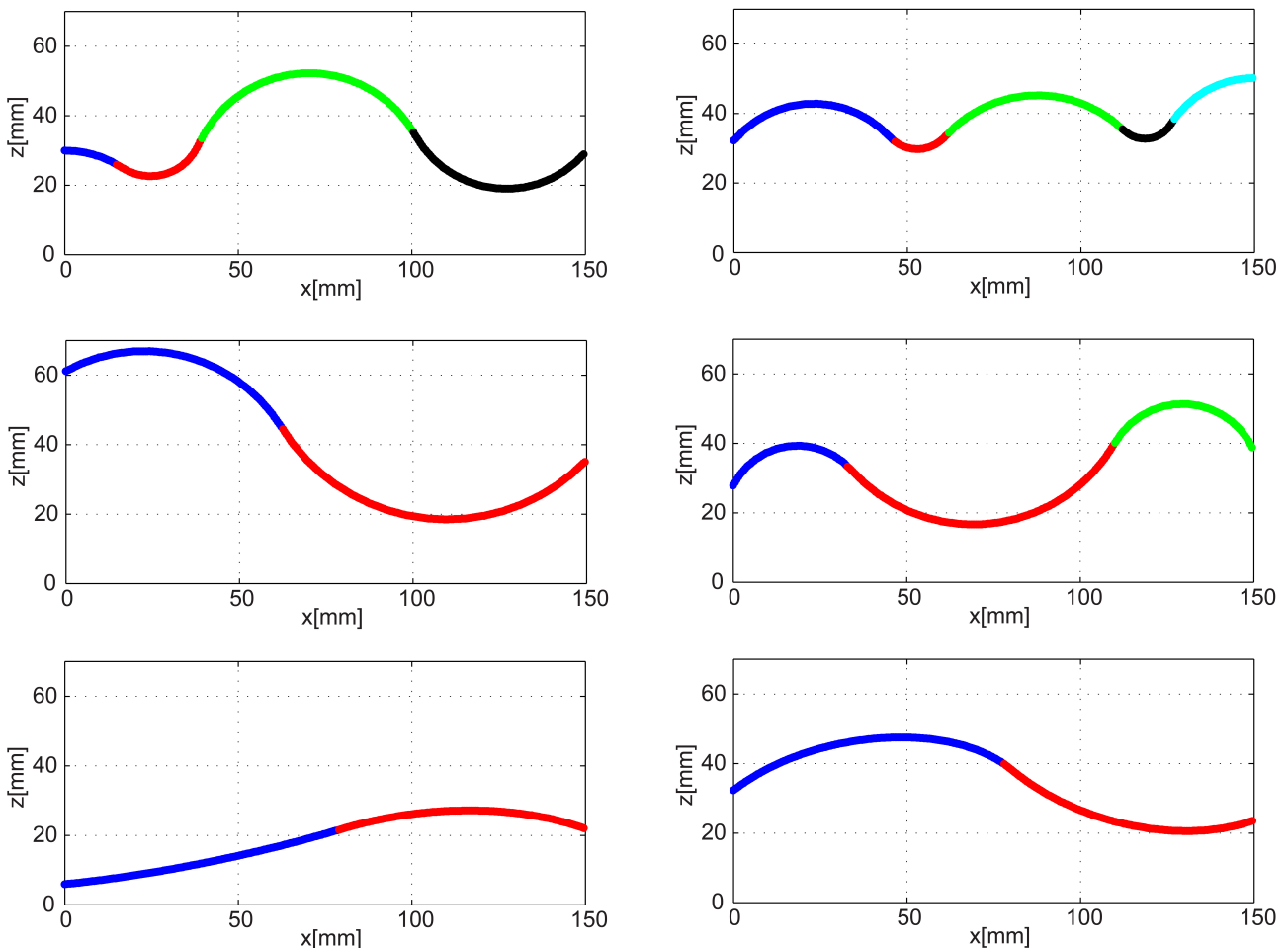
8.2. Određivanje pragova segmentacije upotrebom veštačkih neuronskih mreža

Manuelno podešavanje vrednosti pragova za segmentaciju i grupisanje može biti dugotrajno, što je važan nedostatak ako je potrebno iskoristiti algoritam prepoznavanja kvadraka za automatizaciju nekog procesa. Vrednosti pragova zavise od oblika konture čija se segmentacija vrši kao i od nivoa šuma u konturi. Međutim, nije uočena jasna korelacija između pragova i geometrijskih karakteristika kon-

ture koja se segmentira. Kako bi se podesile vrednosti parametara za zadovoljavajuću segmentaciju, korisnik obično mora da izvrši nekoliko iteracija, odnosno proba segmentacije.

Navedeni problem pronalaženja pragova stvorio je potrebu za razvojem efikasne i automatske metode za izbor njihovih vrednosti. Kao jedno od mogućih rešenja problema predstavlja se metoda određivanja vrednosti pragova segmentacije upotrebom veštačkih neuronskih mreža. Osnovna ideja metode je korišćenje unapred obučene veštačke neuronske mreže koja bi mogla da, na osnovu određenih matematički opisivih karakteristika konture, automatski izvrši estimaciju potrebnih vrednosti pragova. Kao što se može zaključiti, kritičan proces metode predstavlja sama obuka mreže, odnosno definisanje strukture obučavajućih parova. Sa druge strane, jasno je da je pri obučavanju potrebno izvršiti određeni broj simulacija primene algoritma nad određenim brojem proizvoljnih kontura različitih po svom obliku i strukturi.

U ovom radu za potrebe obučavanja mreže izvršeno je sintetisanje 6 reprezentativnih kontura. Konture sadrže različite kružne i eliptične segmente i svaka od njih sadrži po 300 ekvidistantnih tačaka. Izabrani raster je 0,5 mm, a konture su kontaminirane šumom sa SNR 120 dB. Nad konturama je primenjen algoritam segmentacije iz poglavlja 6.2 i tom prilikom su za svaku pojedinačnu konturu ručno određeni odgovarajući pragovi za segmentaciju. Grafički prikazi segmentiranih kontura za obučavanje mreža dati su na slici 8.3.



Slika 8.3: Segmentirane reprezentativne konture za obuku veštačkih neuronskih mreža.

Da bi se mreža za estimaciju pragova obučila potrebno je definisati obučavajuće parove, odnosno ulazne i izlazne vektore. Izlazni vektori treba da poseduju onakvu strukturu kakvu zapravo želimo da dobijemo od mreže kada je iskoristimo nakon obučavanja. U tom slučaju izlazni vektori predstavljaju vrednosti procenjenih pragova za segmentaciju i grupisanje i za svih 6 reprezentativnih kontura sa

slike 8.3 mogu se predstaviti matricom:

$$\mathbf{O} = \begin{bmatrix} t_{rc1} & t_{rc2} & \dots & t_{rc6} \\ t_{ov1} & t_{ov2} & \dots & t_{ov6} \\ t_{gr1} & t_{gr2} & \dots & t_{gr6} \end{bmatrix}. \quad (8.1)$$

Ulazni vektor, s druge strane treba da opiše oblik same konture. Nakon inicijalnih istraživanja na ovu temu odlučeno je da se ulazni vektor sastoji od:

- skupa z koordinata tačaka reprezentativnih kontura (koje su poredane redom od prve ka poslednjoj - tristotoj u skeniranoj liniji),
- srednje visine konture,
- parametra koji sadrži informaciju o valovitosti konture.

Srednja visina konture se može opisati kao:

$$h_{sr} = \frac{\sum_{n=1}^N z_n}{N}, \quad (8.2)$$

gde su $z_i, i = 1, \dots, N$ - z koordinate tačaka konture, a N ukupan broj tačaka (u slučaju kontura za obučavanje mreže $N = 300$).

Sa druge strane, valovitost konture se može definisati kao:

$$W = \frac{\sum_{i=1}^N |z_i - h_{sr}|}{N}.$$

Na osnovu izloženog, ulazni vektori za obučavanje mreže za svih 6 kontura sa slike 8.3 mogu se predstaviti matricom:

$$\mathbf{I} = \begin{bmatrix} h_{sr1} & h_{sr2} & \dots & h_{sr6} \\ W_1 & W_2 & \dots & W_6 \\ z_{11} & z_{12} & \dots & z_{13} \\ \vdots & \vdots & \ddots & \vdots \\ z_{N1} & z_{N2} & \dots & z_{N3} \end{bmatrix}. \quad (8.3)$$

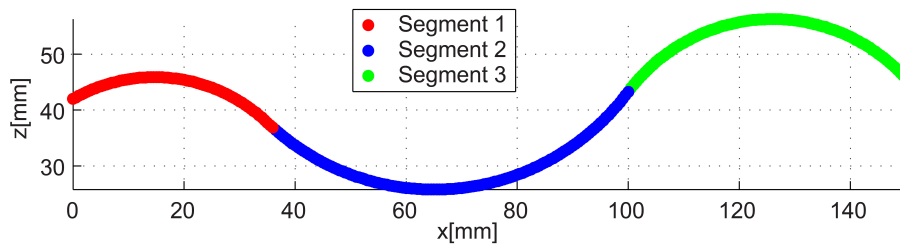
Posle definisanja obučavajućih parova potrebno je izabrati tip neuronske mreže koja će biti korišćena. Nakon inicijalnog istraživanja, odlučeno je da se za potrebe ove metode koriste veštačke neuronske mreže bazirane na algoritmima učenja sa tzv. prostiranjem greške unazad (engl. *backpropagation artificial neural network*) [146]. Za potrebe testiranja formirano je i obučeno devet veštačkih neuronskih mreža ovog tipa. Struktura ovih mreža je sa jednim, dva i tri skrivena sloja neurona, s tim što se razlikuju brojevi neurona u skrivenim slojevima. Brojevi neurona u skrivenim slojevima su proizvoljno izabrani. Nakon kreiranja sve mreže su obučene istim obučavajućim parovima i pri istim uslovima. Korišćen je Levenberg-Markeov algoritam učenja [152]. Aktivaciona funkcija neurona je bila sigmoidna. Maksimalni broj iteracija je postavljen na 100, a greška obučavanja na 0,001.

Nakon obučavanja mreža izvršeno je njihovo testiranje za estimaciju vrednosti pragova za slučaj svih kontura oblaka sa slike 8.1(b). Ulazni vektor zasnovan je na 300 tačaka sa konture. Kako je broj tačaka konture u opštem slučaju proizvoljan, neophodno je definisati način za svođenje bilo kog broja tačaka na 300. Za potrebe predloženog algoritma usvojen je sledeći princip. Ako je broj tačaka na konturi $N > 300$, uzima se prvih $300 \times k$ tačaka gde je k celobrojni rezultat deljenja N sa 300, a zatim se vrši uniformno smanjenje broja tačaka sa faktorom k . Sa druge strane, ako je $N < 300$, vrši se

uniformno dodavanje tačaka uz linearnu interpolaciju, sve dok broj tačaka ne postane veći od 300, a zatim se koristi već opisana procedura za smanjenje broja tačaka. Sve konture sa slike 8.1(b) sadrže po 750 tačaka što znači da se pri pozivanju mreže koristi svaka druga tačka od prvih 600 tačaka dok se poslednjih 150 ne uzima u razmatranje.

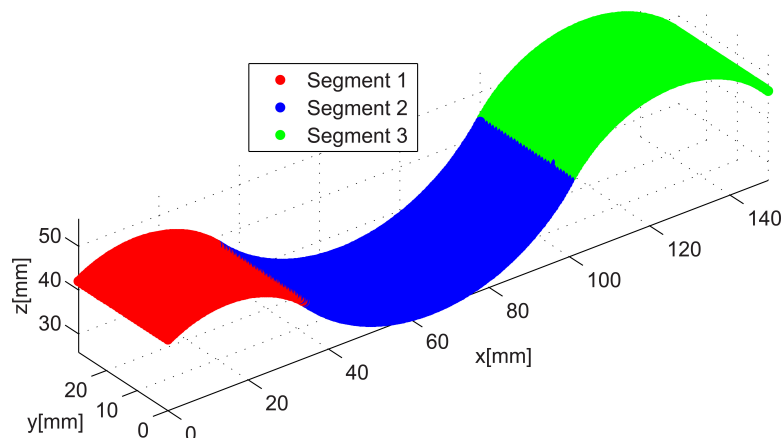
Tabela 8.1: Rezultati segmentacije sintetizovanog struktuiranog oblaka tačaka upotrebom neuronskih mreža.

Mreža	Struktura	Srednja vrednost pozicija prelaza		Standardna devijacija pozicija prelaza	
		Prelaz 1	Prelaz 2	Prelaz 1	Prelaz 2
1.	[3]	190	286	0,5	0,4
2.	[7]	178	317	0,0	0,2
3.	[10]	192	288	0,0	0,0
4.	[5,3]	176	502	0,0	0,0
5.	[3,6]	202	305	0,0	0,0
6.	[6,6]	166,5	478,5	0,5	1,4
7.	[2,5,2]	182	268	4,4	1,0
8.	[3,6,3]	180	323	0,9	2,1
9.	[4,2,5]	186	503	0,2	0,3



Slika 8.4: Rezultati segmentacije sintetizovane konture upotrebom algoritma sa automatskim izborom pragova.

Nakon upotrebe svih devet obučeni mreža za potrebe segmentacije celokupnog oblaka tačaka (slika 8.1(b)), dobijeni rezultati su predstavljeni u tabeli 8.1. U ovoj tabeli su prikazane srednje vrednosti i standardne devijacije detektovanih prelaznih tačaka za sve konture na objektu. Radi podsećanja, projektovani prelazi između segmenata su na pozicijama 175 i 500. Posmatranjem rezultata segmentacije na osnovu pragova dobijenih mrežama može se uvideti da je pri upotrebi nekoliko mreža greška i do 232 uzorka. Ipak, uočeno je da određene mreže daju očekivano dobre rezultate. Najbolji rezultati dobijeni su korišćenjem mreže sa dva skrivena sloja i to sa 5 neurona u prvom i 3 u drugom skrivenom sloju – mreža pod rednim brojem 4 u tabeli 8.1. Tom prilikom pri segmentaciji prve konture oblaka (konture sa slike 8.2) dobijeni su pragovi za segmentaciju čije su vrednosti $t_{rc} = 0,1762$, $t_{ov} = 1,0546 \cdot 10^{-13}$ i $t_{gr} = 26,5237$. Lako je zapaziti da su ove vrednosti pragova bliske vrednostima koje su heuristički određene pri segmentaciji ove konture u poglavlju 8.1. Pri pronalaženju oba prelaza korišćenjem ove mreže algoritam segmentacije je na nivou celog oblaka generisao srednje vrednosti prelaznih tačaka koje se nalaze na pozicijama 176 i 502. Ovo znači da je maksimalna greška segmentacije samo dva uzorka (tačke) odnosno 0,4 mm duž x ose (raster konture je 0,2 mm). Ovako nizak nivo greške može se smatrati odličnim rezultatom imajući u vidu nivo šuma prisutnog u signalu. Radi ilustracije rezultata segmentacije korišćenjem automatskog izbora pragova na slici 8.4 grafički je prikazana segmentacija prve konture oblaka upotrebom izdvojene mreže [5,3]. Pored toga, na slici 8.5 predstavljen je rezultat segmentacije kompletnog sintetizovanog oblaka sa slike 8.1(b). Potrebno je napomenuti da je generisanje vrednosti pragova upotrebom izabrane mreže vršeno za svaku konturu struktuiranog oblaka pojedinačno.



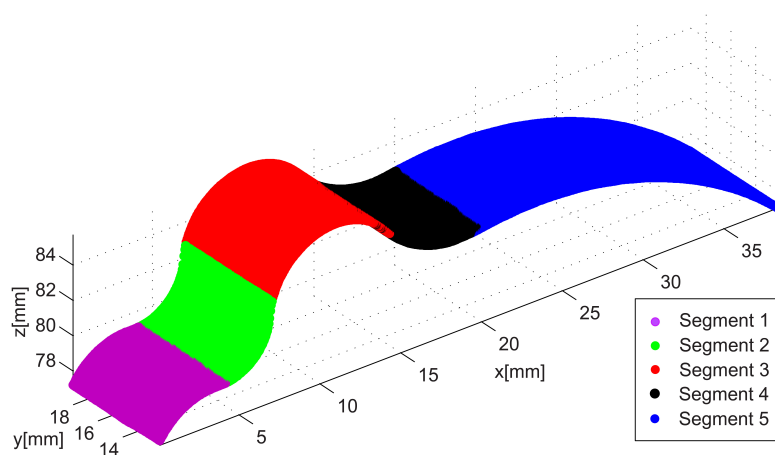
Slika 8.5: Rezultati segmentacije sintetizovanog oblaka tačaka upotrebom algoritma sa automatskim izborom pragova.

8.3. Eksperimentalna verifikacija metode

Nakon predstavljanja metode automatskog izbora pragova segmentacije i potvrde koncepta na primeru jednog sintetizovanog oblaka tačaka, predloženi princip je verifikovan na primeru realnog oblaka tačaka. Za ovu priliku upotrebljen je isti realni oblak tačaka koji je korišćen i u poglavlju 6, a koji je prikazan na slici 6.13(b). Ovaj struktuirani oblak sadrži 141 konturu sa po 765 tačaka. Raster oblaka duž x ose je ekvidistantan i iznosi 0,05 mm. Projektovani prelazi su tačke sa rednim brojem 85, 140, 285 i 400.

Tabela 8.2: Rezultati segmentacije realnog struktuiranog oblaka tačaka upotrebom neuronske mreže.

Mreža	Struktura	Srednja vrednost pozicija prelaza				Standardna devijacija pozicija prelaza			
		Prelaz 1	Prelaz 2	Prelaz 3	Prelaz 4	Prelaz 1	Prelaz 2	Prelaz 3	Prelaz 4
4.	[5,3]	83	140	285	399	0,6	0,0	1,0	2,0

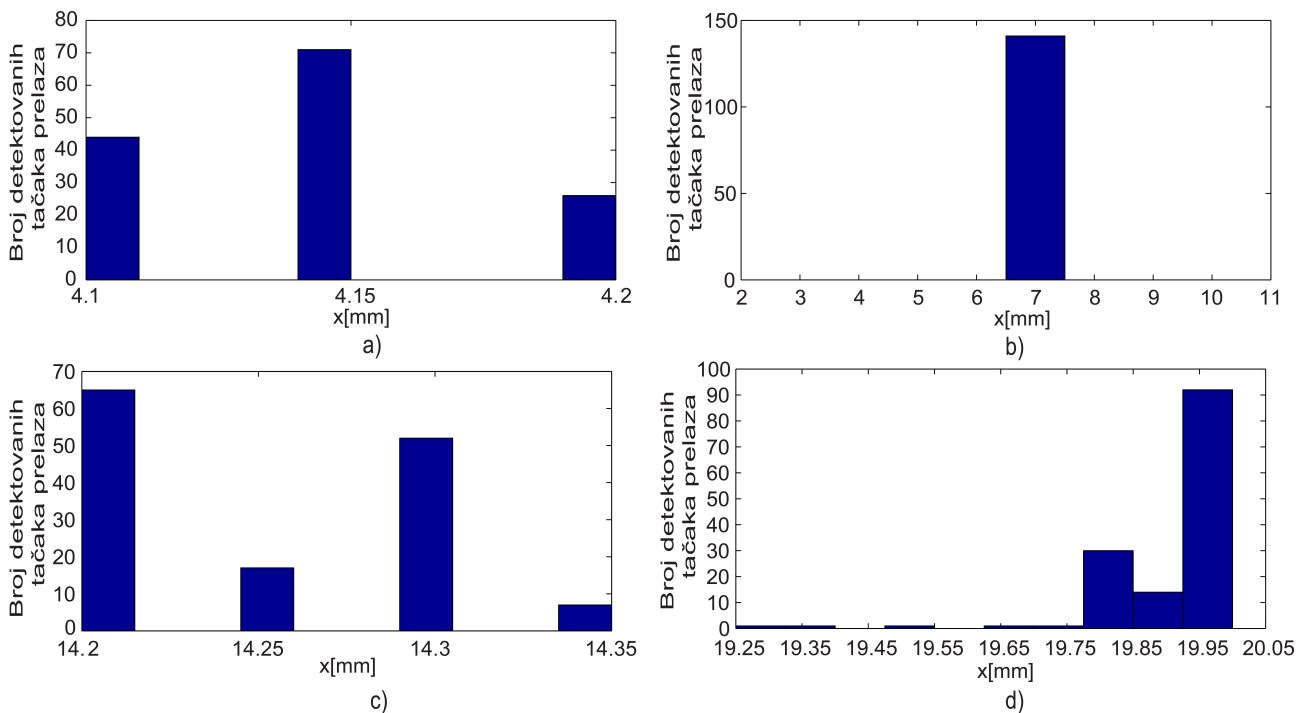


Slika 8.6: Rezultati segmentacije realnog oblaka tačaka upotrebom algoritma sa automatskim izborom pragova.

Za potrebe verifikacije predloženog metoda na realnom oblaku tačaka, iskorišćena je veštačka neuronska mreža koja je pokazala najbolje rezultate na sintetizovnom oblaku u prethodnom poglavlju. Potrebno je napomenuti da je, zbog pomenutog ograničenja odnosno potrebe za korišćenjem samo 300 tačaka konture pri pozivanju mreže, izvršena transformacija informacije o skeniranim linijama na

način kao što je predloženo u prethodnom poglavlju. Naime, signal (skenirana linija) je redukovano na dužinu od 300 tačaka - u obzir je uzeta svaka druga tačka pri čemu je poslednjih 176 tačaka izuzeto iz razmatranja.

Upotrebom predstavljene automatske metode izbora pragova na osnovu izabrane mreže dobijeni su odlični rezultati koji su prikazani u tabeli 8.2. Može se zaključiti da maksimalno odstupanje od idealnih prelaza postoji pri detektovanju prvog prelaza i iznosi 2 uzoraka. Ukoliko se posmatra fizička distanca duž x ose, ova greška segmentacije iznosi 0,1 mm (raster oblaka je 0,05 mm). Pozvana mreža je za segmentaciju npr. prve konture oblaka generisala sledeće vrednosti pragova: $t_{rc} = 0,0558$, $t_{ov} = 4,66 \cdot 10^{-15}$ i $t_{gr} = 15,4614$. Naravno vrednosti generisanih pragova za svaku dalju konturu se mogu neznatno razlikovati u zavisnosti od nivoa šuma i nepravilnosti realnog skeniranog objekta. Međutim, postupkom svođenja celokupne konture na 300 reprezentativnih tačaka na osnovu kojih mreža odlučuje o vrednostima pragova, ovaj uticaj je smanjen.



Slika 8.7: Histogrami detektovanih prelaza sa slike 8.6: a) između segmenata 1 i 2; b) između segmenata 2 i 3; c) između segmenata 3 i 4; d) između segmenata 4 i 5.

Na slici 8.6 grafički su prikazani rezultati opisane segmentacije na celom oblaku tačaka. Takođe, na slici 8.7 dati su histogrami koji reprezentuju pozicije prelaznih tačaka po dužini u milimetrima. Upoređivanjem ovih rezultata sa rezultatima segmentacije ručno podešenim pragovima iz poglavlja 6, koji su grafički prikazani na slici 6.13(b) kao i histogramima na slici 6.14, može se doći do zaključka da je u ovom slučaju korišćena neuronska mreža obezbedila bolje rezultate. Naime, ukoliko razmatramo sveukupno detektovane prelaze između segmenata na svim konturama možemo zaključiti da su u slučaju korišćenja automatskog odabira vrednosti pragova oni značano ujednačeniji. Ovakvi rezultati pokazuju da je, primenom postavljenog procesa automatskog izbora pragova korišćenjem veštačkih neuronskih mreža, metoda segmentacije i prepoznavanja jedne klase kvadraka značajno unapređena. U ovom konkretnom slučaju unapređenje se, osim u obezbeđivanju visokog stepena autonomije, ogleda i u samom povećanju kvaliteta segmentacije. Na samom kraju ovog poglavlja treba naglasiti da su tokom izrade ove disertacije delovi istraživanja i rezultati iz ovog poglavlja objavljeni u [153].

9. Zaključak

Savremene tehnike za trodimenzionalnu (3D) digitalizaciju objekata, pored tradicionalne upotrebe pri projektovanju proizvoda u okviru koncepta reverznog inženjerstva (RE), svoju ulogu sve češće pronalaze i u oblasti upravljanja proizvodnim procesima. Uređaji za 3D skeniranje, u najvećem broju slučajeva, generišu oblake tačaka koji se po pravilu odlikuju velikom gustinom i često sadrže više nego dovoljno podataka o oblicima i strukturi skeniranih objekata. Dok su za analizu jednodimenzionalnih i dvodimenzionalnih signala na raspolaganju brzi algoritmi, tehnike obrade 3D signala još uvek nisu dostigle nivo performansi koji je pogodan za primenu kod sistema koji zahtevaju rad u realnom vremenu.

Procesiranje oblaka tačaka u cilju izdvajanja potrebnih strukturnih informacija o objektu predstavlja jednu od najkompleksnijih faza obrade trodimenzionalnog signala. Kao najznačajniji koraci obrade oblaka tačaka izdvajaju se segmentacija i prepoznavanje gradivnih geometrijskih primitiva skeniranog objekta. Kompleksnost ovih procesa (pre svega segmentacije) povećava se pri postojanju površi $G1$ i više kontinualnosti. Međutim, postoje i tehnike kojima se, u određenoj meri, navedeni koraci procesiranja mogu računski relaksirati. Ovde se pre svega misli na procese redukcije broja tačaka inicijalnog oblaka pri kojima se zadržava visok stepen prezervacije informacije dobijene skeniranjem.

Dosadašnja istraživanja na temu procesiranja oblaka tačaka dovela su do razvoja praktično primenjenih metoda čiji osnovni nedostatak čini potreba značajne korisničke asistencije. Ova činjenica predstavlja glavnu prepreku za njihovu primenu u automatskim proizvodnim sistemima. Sa druge strane, potpuno autonomne metode segmentacije i prepoznavanja geometrijskih primitiva iz oblaka tačaka, koje bi bile pogodne za praktičnu primenu, predstavljaju istraživački izazov.

U skladu sa navedenim ograničenjima primene tehnika 3D digitalizacije u oblastima industrije koje zahtevaju rad u realnom vremenu, u disertaciji je analiziran problem automatskog procesiranja oblaka tačaka. Cilj istraživanja ove doktorske disertacije usmeren je na oblasti procesiranja oblaka za potrebe inteligentnog redukovanja njegovog broja tačaka, sa jedne, i na prepoznavanje gradivnih geometrijskih primitiva skeniranih objekata, sa druge strane. Na osnovu polaznih hipoteza i izvršenih istraživanja, kreirane su i predstavljene četiri metode. Prva metoda namenjena je pomenutoj redukciji broja tačaka oblaka, sledeće dve metode vrše prepoznavanje primitiva, dok se poslednja metoda odnosi na optimizaciju parametara algoritama prepoznavanja.

Zbog potrebe minimalnog gubitka informacija o skeniranim objektima tokom procesa smanjenja obimnih i gustih oblaka, razvijena je metoda redukcije tačaka (poglavlje 5) zasnovana na ε -neosetljivoj regresiji nosećim vektorima (ε -SVR) skeniranih linija struktuiranih formi oblaka tačaka. Ova tehnika regresije je sposobna da, u oblastima visoke lokalne promene geometrije kontura, detektuje i izdvoji tačke koje se nazivaju nosećim vektorima. Upravo noseći vektori predstavljaju najveće nosioce informacije o geometriji procesiranih kontura, a samim tim i o geometriji skeniranih površina objekata. U skladu sa navedenom osobinom kreirana je metodologija redukcije broja tačaka struktuiranih oblaka ε -SVR procesiranjem svih kontura oblaka ponaosob u cilju detektovanja i zadržavanja samo tačaka koje predstavljaju noseće vektore. Pored toga, kao prevencija mogućeg znatnog nedostatka izdvojenih tačaka u oblastima bez značajne promene geometrije, metoda je dopunjena uniformnim uzorkovanjem na osnovu zadatog parametra maksimalne dozvoljene distance između tačaka u redukovanim konturama oblaka.

Pored predstavljene metodologije i prikazanih rezultata ε -SVR redukcije oblaka, kroz navedene primere izvršena su i istraživanja na temu uticaja značajnih SVR parametara na krajnje rezultate primene metode. Utvrđeno je da parametri ε i C imaju znatan uticaj na ishod regresije. Njihove visoke vrednosti dovode do visokog stepena redukcije odnosno do manjeg broja zadržanih tačaka nakon izvršenog procesa. Na osnovu prirode ovih parametara pokazano je da je za upotrebu metoda znatno povoljnije postavljanje ε na određenu vrednost (shodno geometriji objekta) i uticanje na stepen željene redukcije promenom vrednosti parametra C . Naravno, pored izbora vrednosti za ε i C značajna je i uloga izbora kernel funkcije koja će se koristiti za ε -SVR. U disertaciji je izvršena uporedna analiza

upotrebe B-splajn i splajn kernela i prikazane su njihove prednosti i mane. B-splajn kernel je pokazao slabije rezultate kod primene nad skeniranim konturama visokog stepena promene geometrije, ali je njegova osobina osciliranja pogodna za održanje rasporeda izdvojenih tačaka kod objekata sa izražajem ravnom geometrijom. Sa druge strane, splajn kernel je pokazao značajan nivo sposobnosti generalizacije i izdvojio se kao pogodniji kernel za kreiranu metodu redukcije. Takođe, pored navedenih razmatranja, akcentat je pridodat i osobini da razvijena ε -SVR metoda zadržava značajni broj tačaka u oblastima koje reprezentuju ivice površina skeniranih objekata. Ovo se smatra dobrom osobinom zbog činjenice da eventualno povećanje stepena željene redukcije ne utiče osetno na potrebno očuvanje informacije o granicama između različitih površina skeniranog objekta. U disertaciji su prezentovani rezultati primene razvijene metode u slučaju procesiranja tri realna oblaka tačaka objekata različitih geometrijskih osobina. Kvalitet redukcije je ocenjen metodologijom baziranoj na NURBS krivama, ali je i upoređen sa metodom uniformne redukcije pri čemu su ostvareni zapaženi rezultati u korist razvijenog metoda i demonstrirana je validnost polazne hipoteze da je primenom tehnika obrade signala i mašinskog učenja (prvenstveno mašina sa nosećim vektorima) moguće formirati nove robustne metode i algoritme za automatsko redukovanje broja tačaka u okviru trodimenzionalnog oblaka tačaka uz istovremeno zadržavanje neophodnog informacionog sadržaja.

Prva od dve metode prepoznavanja primitiva iz oblaka tačaka, koje su predstavljene u disertaciji, namenjena je ekstrakciji jedne vrste površi drugog reda (kvadrata) iz struktuiranih oblaka tačaka (poglavlje 6). Odlika ove metode je prepoznavanje primitiva koji su u skeniranim linijama reprezentovani eliptičnim segmentima. U skladu sa tim, metoda je namenjena ekstrakciji eliptičnih cilindara, elipsoida i eliptičnih torusa. Osnov metode predstavlja proces segmentacije skeniranih linija koji se temelji na osobinama singularnosti informacione matrice sistema tačaka pri direktnoj regresiji elipsi metodom najmanjih kvadrata. Ovaj proces spada u grupu procesa koji primenjuju strategiju rasta regiona tačka po tačka sve dok su definisani uslovi rasta zadovoljeni.

U disertaciji su predstavljeni i diskutovani rezultati eksperimentalne verifikacije algoritma prepoznavanja jedne vrste kvadrata zabeleženi procesiranjem nekoliko sintetizovanih i jednog realnog oblaka tačaka objekata sastavljenih od različitih ciljanih tipova površi. Zabeleženi su značajni rezultati u smislu tačno prepoznatih pozicija prelaza između površi G1 kontinualnosti. Takođe, u određenim slučajevima predstavljene su i vrednosti estimiranih parametara prepoznatih primitiva koje su upoređene sa onim dizajniranim čime je dodatno utvrđen kvalitet performansi definisane metode prepoznavanja.

Druga metoda prepoznavanja primitiva, koja je predstavljena u poglavlju 7, vrši automatsko ekstrahovanje cilindara i ravni iz nestruktuiranih oblaka tačaka. Ona je primarno namenjena korišćenju pri rekonstrukciji mašinskih delova čija se morfologija uglavnom sastoji od ova dva široko rasprostranjena primitiva. Predstavljeni algoritam sadrži tri osnovne faze procesiranja. Prvu fazu predstavlja proces automatske segmentacije koji je zasnovan na širenju regiona na osnovu singularnosti informacione matrice sistema tačaka pri regresiji opšte površi drugog reda metodom najmanjih kvadrata. Pored segmentacije strategijom širenja regiona razvijena je i dodatna metodologija uklanjanja malih nepovezanih regiona klastera čime je povećana preciznost daljih koraka. U drugoj fazi vrši se spajanje eventualno presegmentiranih regiona i određivanje tipova površi čime se vrši selekcija na tražene primitive i one koji to nisu. Takođe, vrši se estimacija parametara cilindara i ravni. Osnovu ove faze predstavlja proces robustnog prepoznavanja cilindara [108] koji je u ovom slučaju modifikovan i prilagođen čime mu je povećana sposobnost generalizacije. U trećoj, poslednjoj fazi, izvodi se ekstrahovanje prepoznatih ravni i cilindara, ali se i određenim dodatnim procesima povećava kvalitet finalnih rezultata.

Metoda prepoznavanja cilindara i ravni je eksperimentalno verifikovana procesiranjem tri realna oblaka tačaka. Prvi oblak reprezentuje jedan test objekat, dok druga dva predstavljaju dva često korišćena mašinska elementa (vratilo i prirubnicu). Rezultati verifikacije su predstavljeni u disertaciji kao tri odvojene studije slučaja. U ovim studijama izvršena je i uporedna analiza predložene metode i dve postojeće relevantne metode drugih autora. Predložena metoda je pokazala visoke performanse i nadmašila alternativne metode u kvantitativnoj, ali i kvalitativnoj uporednoj analizi čime je potvr-

đena polazna hipoteza da se tehnike obrade signala, analitičke i statističke metode mogu efikasno upotrebiti za razvoj novih robusnih metoda za automatsko prepoznavanje geometrijskih primitiva iz trodimenzionalnog oblaka tačaka.

Obe navedene metode prepoznavanja primitiva, iako automatske, zahtevaju određeno podešavanje ulaznih parametara od strane korisnika. Ova činjenca ipak može predstavljati izvesno ograničenje za primenu u praksi. Imajući navedeno u vidu, tokom izrade doktorske disertacije vršena su i istraživanja u cilju obezbeđivanja mogućnosti automatskog izbora vrednosti parametara algoritama prepoznavanja primitiva iz struktuiranih oblaka tačaka. Na osnovu takvog istraživanja razvijena je dopunska metoda koja se temelji na upotrebi veštačkih neuronskih mreža za potrebe generisanja vrednosti pragova algoritma prepoznavanja. Na ovaj način obezbeđeno je da se celokupni proces prepoznavanja učini autonomnim, odnosno sposobnim za ekstrakciju primitiva bez bilo kakve korisničke asistencije.

Metoda obučavanja i upotrebe veštačkih neuronskih mreža kao i rezultati primene razvijenog procesa prezentovani su u poglavlju 8 ove disertacije. Autonomna metoda je, zbog poređenja rezultata i benefita, testirana nad istim realnim oblakom kao i metoda bez automatskog odabira vrednosti pragova iz poglavlja 6. Pri uporednoj analizi zaključeno je da je upotreba mreža doprinela značajno boljim rezultatima segmentacije u smislu manje disperzije prepoznatih pozicija prelaza između cilindričnih segmenata posmatranog objekta potvrđujući hipotezu da tehnike veštačke inteligencije predstavljaju dovoljnu osnovu za kreiranje inteligentnih metoda za automatsko podešavanje potrebnih parametara prilikom prepoznavanja geometrijskih primitiva iz trodimenzionalnog oblaka tačaka. Takođe, treba naglasiti da je upotreba neuronskih mreža omogućila generisanje vrednosti parametara za svaku skeniranu konturu oblaka ponaosob što pri asistenciji korisnika nije bilo moguće iz praktičnih razloga.

Razmatrajući mogućnost budućih istraživanja vezanih za obrađene teme, može se zaključiti da ova disertacija može predstavljati osnov za:

- razvoj algoritama prepoznavanja nekih drugih tipova primitiva kako u struktuiranim tako i u nestruktuiranim oblacima tačaka,
- razvoj procesa autonomnog određivanja pragova za metodu prepoznavanja cilindara i ravni iz ove disertacije,
- istraživanje na polju upotrebe SVR za procese redukcije nestruktuiranih oblaka tačaka,
- istraživanje u oblasti upotrebe različitih tehnika mašinskog učenja i veštačke inteligencije za prevazilaženje potrebe korisničke asistencije u cilju povećanja autonomnosti postojećih metoda prepoznavanja oblika.

10. Literatura

- [1] E. J. Chikofsky and J. H. Cross, "Reverse engineering and design recovery: a taxonomy," *IEEE Software*, vol. 7, no. 1, pp. 13–17, 1990.
- [2] V. Raja and K. J. Fernandes, *Reverse Engineering: An Industrial Perspective*. Springer Publishing Company, Incorporated, 2007.
- [3] I. Budak, *Reverzibilni inženjerski dizajn - preprocesiranje rezultata 3D digitalizacije*. Novi Sad: Fakultet tehničkih nauka, 2019.
- [4] T. Várady, R. R. Martin, and J. Cox, "Reverse engineering of geometric models—an introduction," *Computer-Aided Design*, vol. 29, no. 4, pp. 255 – 268, 1997.
- [5] E. Savio, L. D. Chiffre, and R. Schmitt, "Metrology of freeform shaped parts," *CIRP Annals*, vol. 56, no. 2, pp. 810 – 835, 2007.
- [6] T. Rabbani, F. Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraint," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 248–253, 2006.
- [7] I. Budak, "Reverzibilno inženjerstvo i CAQ, Predavanje: Metode 3D digitalizacije u reverzibilnom inženjerstvu." <http://www.dpm.ftn.uns.ac.rs/predmeti/Reverzibilno%20in%20C5%BEenjerstvo%20i%20CAQ/Predavanje%20%20i%203%20-%20Metode%203D%20digitalizacije.pdf>. Poslednji pristup: Mart 2020.
- [8] M. Sokac, I. Budak, M. Katic, Z. Jakovljevic, Z. Santosi, and D. Vukelic, "Improved surface extraction of multi-material components for single-source industrial x-ray computed tomography," *Measurement*, vol. 153, p. 107438, 2020.
- [9] U. Ewert, U. Zscherpel, and P. Willems, "Recent trends in digital industrial radiology," in *National Seminar of ISNT*, 2002.
- [10] A. Thompson and R. Leach, *Introduction to Industrial X-ray Computed Tomography*, pp. 1–23. Cham: Springer International Publishing, 2018.
- [11] W. Sun, B. Starly, J. Nam, and A. Darling, "Bio-CAD modeling and its applications in computer-aided tissue engineering," *Computer-Aided Design*, vol. 37, no. 11, pp. 1097 – 1114, 2005.
- [12] Ø. Hjelle and M. Dæhlen, *Triangulations and Applications*. Springer-Verlag Berlin Heidelberg, 2006.
- [13] S. Motavalli and B. Bidanda, "A part image reconstruction system for reverse engineering of design modifications," *Journal of Manufacturing Systems*, vol. 10, no. 5, pp. 383 – 395, 1991.
- [14] J. Moss, A. Linney, S. Grindrod, and C. Mosse, "A laser scanning system for the measurement of facial surface morphology," *Optics and Lasers in Engineering*, vol. 10, no. 3, pp. 179 – 190, 1989.
- [15] P. B. Petrovic, Z. Jakovljevic, M. Pilipovic, and D. V. Mikovic, "In process identification of workpiece/system geometrical deviations based on general purpose robots and laser triangulation sensors - part 1: Conceptual framework," in *Proceedings of the 10th International Scientific Conference on Flexible Technologies (MMA2009)*, pp. 174–178, 2009.

- [16] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, “Real-time 3D model acquisition,” *ACM Trans. Graph.*, vol. 21, no. 3, p. 438–446, 2002.
- [17] M. Ribo and M. Brandner, “State of the art on vision-based structured light systems for 3D measurements,” in *International Workshop on Robotic Sensors: Robotic and Sensor Environments*, pp. 2–6, 2005.
- [18] J. Pagès, J. Salvi, R. García, and C. Matabosch, “Overview of coded light projection techniques for automatic 3D profiling,” *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 133–138, 2003.
- [19] J. Salvi, J. Pagès, and J. Batlle, “Pattern codification strategies in structured light systems,” *Pattern Recognition*, vol. 37, no. 4, pp. 827 – 849, 2004.
- [20] J. Park, G. N. DeSouza, and A. C. Kak, “Dual-beam structured-light scanning for 3-D object modeling,” in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 65–72, 2001.
- [21] J. Geng, “Structured-light 3D surface imaging: a tutorial,” *Adv. Opt. Photon.*, vol. 3, no. 2, pp. 128–160, 2011.
- [22] S. Chen, Y. Li, and J. Zhang, “Vision processing for realtime 3D data acquisition based on coded structured light,” *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 17, pp. 167–76, 2008.
- [23] P. B. Petrović, J. Hodolič, A. Vićentić, M. Pilipović, Ž. Jakovljević, I. Danilov, N. Lukić, P. Baltić, D. Vukelić, I. Budak, M. Hažistević, and V. Miković, “Inteligentni robotski sistemi za ekstremno diverzifikovanu proizvodnju – TR35007,” in *Zbornik radova 38. JUPITER konferencije*, pp. UR.49– UR.66, 2012.
- [24] GOM GmbH, “Atos Compact Scan 5M.” <http://www.gom.com>. Poslednji pristup: Mart 2020.
- [25] K. Creath, J. Wyant, and E. Malacara, “Moiré and fringe projection techniques,” *Optical Shop Testing*, 1995.
- [26] C. Chen and Y. F. Zheng, “Passive and active stereo vision for smooth surface detection of deformed plates,” *IEEE Transactions on Industrial Electronics*, vol. 42, no. 3, pp. 300–306, 1995.
- [27] S. Chaudhuri and A. N. Rajagopalan, *Depth from defocus: A real aperture imaging approach*. Springer-Verlag, 1998.
- [28] F. Poux, *The Smart Point Cloud: Structuring 3D intelligent point data*. PhD thesis, University of Liege, Liege, Belgium, 2019.
- [29] H. Zhao, J.-P. Kruth, N. V. Gestel], B. Boeckmans, and P. Bley, “Automated dimensional inspection planning using the combination of laser scanner and tactile probe,” *Measurement*, vol. 45, no. 5, pp. 1057 – 1066, 2012.
- [30] M. Mahmud, D. Joannic, M. Roy, A. Isheil, and J.-F. Fontaine, “3D part inspection path planning of a laser scanner with control on the uncertainty,” *Computer-Aided Design*, vol. 43, no. 4, pp. 345 – 355, 2011.
- [31] H. Srinivasan, O. L. A. Harrysson, and R. A. Wysk, “Automatic part localization in a CNC machine coordinate system by means of 3D scans,” *The International Journal of Advanced Manufacturing Technology*, vol. 81, no. 5, pp. 1127–1138, 2015.

- [32] P. B. Petrovic, “Rubberized cord thickness measurement based on laser triangulation - Part I: Technology,” *FME Transactions*, vol. 35, no. 2, pp. 77 – 84, 2007.
- [33] P. B. Petrovic, “Rubberized cord thickness measurement based on laser triangulation - Part II: Validation,” *FME Transactions*, vol. 35, no. 2, pp. 85 – 93, 2007.
- [34] W. Huang and R. Kovacevic, “A laser-based vision system for weld quality inspection,” *Sensors*, vol. 11, no. 1, pp. 506–521, 2011.
- [35] Z. Fang, D. Xu, and M. Tan, “A vision-based self-tuning fuzzy controller for fillet weld seam tracking,” *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 3, pp. 540–550, 2011.
- [36] L. Yang, E. Li, Y. Mao, and Z. Liang, “A novel 3D measurement method of welding workpiece for robot off-line programming,” in *2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 3835–3840, 2017.
- [37] L. Zhang, W. Ke, Q. Ye, and J. Jiao, “A novel laser vision sensor for weld line detection on wall-climbing robot,” *Optics & Laser Technology*, vol. 60, pp. 69–79, 08 2014.
- [38] Z. Wang, K. Zhang, Y. Chen, Z. Luo, and J. Zheng, “A real-time weld line detection for derusting wall-climbing robot using dual cameras,” *Journal of Manufacturing Processes*, vol. 27, pp. 76 – 86, 2017.
- [39] R. Veiga, A. Oliveira, L. Arruda, and F. Junior, “Localization and Navigation of a Climbing Robot Inside a LPG Spherical Tank Based on Dual-LIDAR Scanning of Weld Beads,” in *Koubaa A. (eds) Robot Operating System (ROS)*, vol. 625, pp. 161–184, Studies in Computational Intelligence, Springer, Cham, 2016.
- [40] J. Xu, N. Xi, C. Zhang, Q. Shi, and J. Gregory, “Real-time 3D shape inspection system of automotive parts based on structured light pattern,” *Optics & Laser Technology*, vol. 43, no. 1, pp. 1 – 8, 2011.
- [41] E. Kiraci, P. Franciosa, G. A. Turley, A. Olifent, A. Attridge, and M. A. Williams, “Moving towards in-line metrology: evaluation of a laser radar system for in-line dimensional inspection for automotive assembly systems,” *The International Journal of Advanced Manufacturing Technology*, vol. 91, pp. 69–78, 2017.
- [42] J. Xu, N. Xi, C. Zhang, Q. Shi, and J. Gregory, “A robot-assisted back-imaging measurement system for transparent glass,” *IEEE-ASME Transactions on Mechatronics - IEEE-ASME TRANS MECHATRON*, vol. 17, pp. 779–788, 2012.
- [43] W. Bauer, F. T. Bessler, E. Zabler, and R. B. Bergmann, “Computer tomography for nondestructive testing in the automotive industry,” in *Developments in X-Ray Tomography IV* (U. Bonse, ed.), vol. 5535, pp. 464 – 472, International Society for Optics and Photonics, SPIE, 2004.
- [44] D. Mery, “Aluminum casting inspection using deep learning: A method based on convolutional neural networks,” *Journal of Nondestructive Evaluation*, vol. 39, no. 12, pp. 77 – 84, 2020.
- [45] M. Mitić and Z. Miljković, “Bio-inspired approach to learning robot motion trajectories and visual control commands,” *Expert Systems with Applications*, vol. 42, no. 5, pp. 2624–2637, 2015.
- [46] H. Andreasson, A. Bouguerra, M. Cirillo, D. N. Dimitrov, D. Driankov, L. Karlsson, A. J. Lilienthal, F. Pecora, J. P. Saarinen, A. Sherikov, and T. Stoyanov, “Autonomous transport vehicles: Where we are and what is missing,” *IEEE Robotics Automation Magazine*, vol. 22, no. 1, pp. 64–75, 2015.

- [47] G. Fu, P. Corradi, A. Menciassi, and P. Dario, “An integrated triangulation laser scanner for obstacle detection of miniature mobile robots in indoor environment,” *IEEE/ASME Transactions on Mechatronics*, vol. 16, pp. 778 – 783, 2011.
- [48] C. Röhrig, D. Heß, C. Kirsch, and F. Künemund, “Localization of an omnidirectional transport robot using ieee 802.15.4a ranging and laser range finder,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3798–3803, 2010.
- [49] K. Jung, J. Kim, J. Kim, E. Jung, and S. Kim, “Positioning accuracy improvement of laser navigation using UKF and FIS,” *Robotics and Autonomous Systems*, vol. 62, no. 9, pp. 1241 – 1247, 2014.
- [50] C. Li, S. Wang, Y. Zhuang, and F. Yan, “Deep sensor fusion between 2D laser scanner and IMU for mobile robot localization,” *IEEE Sensors Journal*, 2019.
- [51] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, “Fast object localization and pose estimation in heavy clutter for robotic bin picking,” *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 951–973, 2012.
- [52] O. Ghita and P. F. Whelan, “A bin picking system based on depth from defocus,” *Machine Vision and Applications*, vol. 13, no. 4, pp. 234–244, 2003.
- [53] M. Nieuwenhuisen, D. Droeschel, D. Holz, J. Stückler, A. Berner, J. Li, R. Klein, and S. Behnke, “Mobile bin picking with an anthropomorphic service robot,” in *2013 IEEE International Conference on Robotics and Automation*, pp. 2327–2334, IEEE, 2013.
- [54] M. U. Khalid, J. M. Hager, W. Kraus, M. F. Huber, and M. Toussaint, “Deep workpiece region segmentation for bin picking,” in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 1138–1144, IEEE, 2019.
- [55] K. Rahardja and A. Kosaka, “Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS’96*, vol. 3, pp. 1448–1457, IEEE, 1996.
- [56] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, “Fast graspability evaluation on single depth maps for bin picking with general grippers,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1997–2004, IEEE, 2014.
- [57] M. Pauly, M. Gross, and L. P. Kobbelt, “Efficient simplification of point-sampled surfaces,” in *IEEE Visualization, 2002. VIS 2002.*, pp. 163–170, 2002.
- [58] P. S. Heckbert and M. Garland, “Survey of polygonal surface simplification algorithms,” tech. rep., Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1997.
- [59] P. Cignoni, C. Montani, and R. Scopigno, “A comparison of mesh simplification algorithms,” *Computers & Graphics*, vol. 22, no. 1, pp. 37–54, 1998.
- [60] K. Lee, H. Woo, and T. Suk, “Point data reduction using 3D grids,” *The International Journal of Advanced Manufacturing Technology*, vol. 18, no. 3, pp. 201–210, 2001.
- [61] B.-Q. Shi, J. Liang, and Q. Liu, “Adaptive simplification of point cloud using k-means clustering,” *Computer-Aided Design*, vol. 43, no. 8, pp. 910–922, 2011.
- [62] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

- [63] Y. Miao, R. Pajarola, and J. Feng, “Curvature-aware adaptive re-sampling for point-sampled geometry,” *Computer-Aided Design*, vol. 41, no. 6, pp. 395–403, 2009.
- [64] E.-C. Kang, D.-B. Kim, and K. H. Lee, “Balanced feature-sensitive point sampling for 3D model generation,” *The International Journal of Advanced Manufacturing Technology*, vol. 38, no. 1-2, pp. 130–142, 2008.
- [65] H. Song and H.-Y. Feng, “A global clustering approach to point cloud simplification with a specified data reduction ratio,” *Computer-Aided Design*, vol. 40, no. 3, pp. 281–292, 2008.
- [66] H. Song and H.-Y. Feng, “A progressive point cloud simplification algorithm with preserved sharp edge data,” *The International Journal of Advanced Manufacturing Technology*, vol. 45, no. 5-6, pp. 583–592, 2009.
- [67] X. Ma and R. J. Cripps, “Shape preserving data reduction for 3D surface points,” *Computer-Aided Design*, vol. 43, no. 8, pp. 902–909, 2011.
- [68] G.-J. Wang, C.-C. Wang, and S. Chuang, “Reverse engineering of sculptured surfaces by four-axis non-contacting scanning,” *The International Journal of Advanced Manufacturing Technology*, vol. 15, no. 11, pp. 800–809, 1999.
- [69] D.-Y. Chang and Y.-M. Chang, “A freeform surface modelling system based on laser scan data for reverse engineering,” *The international journal of advanced manufacturing technology*, vol. 20, no. 1, pp. 9–19, 2002.
- [70] I. Budak, M. Soković, J. Kopač, and J. Hodolič, “Point data pre-processing based on fuzzy logic for reverse engineering modelling,” *Strojnski Vestnik/Journal of Mechanical Engineering*, vol. 55, pp. 755–765, 2009.
- [71] I. Budak, M. Sokovic, and B. Barisic, “Accuracy improvement of point data reduction with sampling-based methods by fuzzy logic-based decision-making,” *Measurement*, vol. 44, no. 6, pp. 1188–1200, 2011.
- [72] I. Budak, D. Vukelić, D. Bračun, J. Hodolič, and M. Soković, “Pre-processing of point-data from contact and optical 3D digitization sensors,” *Sensors*, vol. 12, no. 1, pp. 1100–1126, 2012.
- [73] K. Lee, H. Woo, and T. Suk, “Data reduction methods for reverse engineering,” *The International Journal of Advanced Manufacturing Technology*, vol. 17, no. 10, pp. 735–743, 2001.
- [74] Y.-Q. Wang, Y. Tao, H.-J. Zhang, and S.-S. Sun, “A simple point cloud data reduction method based on Akima spline interpolation for digital copying manufacture,” *The International Journal of Advanced Manufacturing Technology*, vol. 69, no. 9-12, pp. 2149–2159, 2013.
- [75] Y. Tao, Y. Li, Y.-Q. Wang, and Y.-Y. Ma, “On-line point cloud data extraction algorithm for spatial scanning measurement of irregular surface in copying manufacture,” *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 5-8, pp. 1891–1905, 2016.
- [76] G. Xu, X. Cheng, L. Wang, K. Tan, and Q. Lou, “An adaptive compression algorithm of scattered point cloud based on wavelet technology,” *Journal of Information and Computational Science*, vol. 11, no. 6, pp. 1917–1927, 2014.
- [77] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, “Pu-net: Point cloud upsampling network,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

- [78] W. Zhang, H. Jiang, Z. Yang, S. Yamakawa, K. Shimada, and L. B. Kara, “Data-driven upsampling of point clouds,” *Computer-Aided Design*, vol. 112, pp. 1 – 13, 2019.
- [79] L. D. Angelo and P. D. Stefano, “Geometric segmentation of 3D scanned surfaces,” *Computer-Aided Design*, vol. 62, pp. 44 – 56, 2015.
- [80] L. D. Angelo, P. D. Stefano, and A. Morabito, “Fillet, rounds, grooves and sharp edges segmentation from 3D scanned surfaces,” *Computer-Aided Design*, vol. 110, pp. 78 – 91, 2019.
- [81] Z. Jakovljevic, R. Puzovic, and M. Pajic, “Recognition of planar segments in point cloud based on wavelet transform,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 342–352, 2015.
- [82] M. R. Rahayem and J. A. Kjellander, “Quadric segmentation and fitting of data captured by a laser profile scanner mounted on an industrial robot,” *The International Journal of Advanced Manufacturing Technology*, vol. 52, no. 1-4, pp. 155–169, 2011.
- [83] M. Vieira and K. Shimada, “Surface mesh segmentation and smooth surface extraction through region growing,” *Computer Aided Geometric Design*, vol. 22, no. 8, pp. 771 – 792, 2005.
- [84] H.-C. Lai, Y.-H. Chang, and J.-Y. Lai, “Development of feature segmentation algorithms for quadratic surfaces,” *Advances in Engineering Software*, vol. 40, no. 10, pp. 1011 – 1022, 2009.
- [85] J. Wang, D. Gu, Z. Yu, C. Tan, and L. Zhou, “A framework for 3D model reconstruction in reverse engineering,” *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 1189 – 1200, 2012.
- [86] M. do Carmo, *Differential Geometry of Curves and Surfaces*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1976.
- [87] M. Attene, B. Falcidieno, and M. Spagnuolo, “Hierarchical mesh segmentation based on fitting primitives,” *The Visual Computer*, vol. 22, no. 3, pp. 181 – 193, 2006.
- [88] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [89] R. Schnabel, R. Wahl, and R. Klein, “Efficient ransac for point-cloud shape detection,” *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.
- [90] P. Mukhopadhyay and B. B. Chaudhuri, “A survey of Hough transform,” *Pattern Recognition*, vol. 48, no. 3, pp. 993 – 1010, 2015.
- [91] T. Rabbani and F. Heuvel, “Efficient Hough transform for automatic detection of cylinders in point clouds,” *Proc ISPRS Workshop Laser Scan 2005, ISPRS Arch*, vol. 36, 01 2005.
- [92] O. O. Ogundana, J. M. Huntley, C. R. Coggrave, and R. L. Burguete, “Automated detection of planes in 3-D point clouds using fast Hough transforms,” *Optical Engineering*, vol. 50, no. 5, pp. 1 – 12, 2011.
- [93] T. Ogundana, C. R. Coggrave, R. L. Burguete, and J. M. Huntley, “Fast Hough transform for automated detection of spheres in three-dimensional point clouds,” *Optical Engineering*, vol. 46, no. 5, pp. 1 – 11, 2007.
- [94] T.-T. Tran, V.-T. Cao, and D. Laurendeau, “Extraction of reliable primitives from unorganized point clouds,” *3D Research*, vol. 6, no. 4, p. 44, 2015.

- [95] T.-T. Tran, V.-T. Cao, and D. Laurendeau, “Extraction of cylinders and estimation of their parameters from point clouds,” *Computers & Graphics*, vol. 46, pp. 345 – 357, 2015.
- [96] T.-T. Tran, V.-T. Cao, and D. Laurendeau, “eSphere: extracting spheres from unorganized point clouds,” *The Visual Computer*, vol. 32, no. 10, pp. 1205–1222, 2016.
- [97] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [98] Y. Liu, J. Zhang, J. Hou, J. Ren, and W. Tang, “Cylinder detection in large-scale point cloud of pipeline plant,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 10, pp. 1700–1707, 2013.
- [99] A. K. Patil, P. Holi, S. K. Lee, and Y. H. Chai, “An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds,” *Automation in Construction*, vol. 75, pp. 65 – 78, 2017.
- [100] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, (Red Hook, NY, USA), pp. 5105–5114, Curran Associates Inc., 2017.
- [101] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph CNN for learning on point clouds,” *ACM Trans. Graph.*, vol. 38, no. 5, 2019.
- [102] B. Nagy and C. Benedek, “3D CNN-based semantic labeling approach for mobile laser scanning data,” *IEEE Sensors Journal*, vol. 19, pp. 10034–10045, Nov 2019.
- [103] L. Winiwarter, G. Mandlburger, S. Schmohl, and N. Pfeifer, “Classification of als point clouds using end-to-end deep learnings,” *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 87, no. 3, 2019.
- [104] T. Chaperon and F. Goulette, “Extracting cylinders in full 3D data using a random sampling method and the Gaussian image.,” in *Proceedings of the Vision Modeling and Visualization Conference*, pp. 35–42, 01 2001.
- [105] J.-F. Lalonde, N. Vandapel, and M. Hebert, “Automatic three-dimensional point cloud processing for forest inventory,” Tech. Rep. CMU-RI-TR-06-21, Carnegie Mellon University, Pittsburgh, PA, July 2006.
- [106] S.-W. Kwon, F. Bosche, C. Kim, C. T. Haas, and K. A. Liapi, “Fitting range data to primitives for rapid local 3D modeling using sparse range point clouds,” *Automation in Construction*, vol. 13, no. 1, pp. 67 – 81, 2004. The best of ISARC 2002.
- [107] M. Hubert, P. J. Rousseeuw, and K. V. Branden, “Robpca: A new approach to robust principal component analysis,” *Technometrics*, vol. 47, no. 1, pp. 64–79, 2005.
- [108] A. Nurunnabi, Y. Sadahiro, and R. Lindenbergh, “Robust cylinder fitting in three-dimensional point cloud data,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-1/W1, pp. 63–70, 2017.
- [109] E. E. Catmull, *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, The University of Utah, Salt Lake City, UT, USA, 1974.

- [110] H. Hoppe, “Progressive meshes,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’96, (New York, NY, USA), pp. 99–108, Association for Computing Machinery, 1996.
- [111] H.-W. Lin, C.-L. Tai, and G.-J. Wang, “A mesh reconstruction algorithm driven by an intrinsic property of a point cloud,” *Computer-Aided Design*, vol. 36, no. 1, pp. 1 – 9, 2004.
- [112] M. Zwicker and C. Gotsman, “Meshing point clouds using spherical parameterization,” *Eurographics Symposium on Point-Based Graphics*, 2004.
- [113] J. J. McConnell, *Computer graphics: theory into practice*. Jones & Bartlett Learning, 2005.
- [114] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [115] V. Vapnik, *The nature of statistical learning theory*. New York: Springer science & business media, 2013.
- [116] V. Vapnik, *Estimation of dependences based on empirical data*. New York: Reprint of 1982 Edition, Springer Science & Business Media, 2006.
- [117] S. R. Gunn *et al.*, “Support vector machines for classification and regression,” *ISIS technical report*, vol. 14, no. 1, pp. 5–16, 1998.
- [118] MathWorks, “MATLAB - Programming platform designed specifically for engineers and scientists.” <https://www.mathworks.com/products/matlab.html>. Poslednji pristup: Jul 2020.
- [119] School of Electronics and Computer Science of the University of Southampton, Research group ISIS, “MATLAB Support Vector Machine Toolbox.” <http://www.isis.ecs.soton.ac.uk/resources/svminfo/>. Poslednji pristup: Maj 2015.
- [120] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’96, (New York, NY, USA), pp. 303–312, Association for Computing Machinery, 1996.
- [121] Stanford Computer Graphics Laboratory, “The Stanford 3D scanning repository.” <http://graphics.stanford.edu/data/3Dscanrep/>. Poslednji pristup: Maj 2015.
- [122] D. Harlev, “MATLAB function - Stlread.” <https://www.mathworks.com/matlabcentral/fileexchange/6678-stlread>. Poslednji pristup: Novembar 2014.
- [123] S. Petitjean, “A survey of methods for recovering quadrics in triangle meshes,” *ACM Comput. Surv.*, vol. 34, pp. 211–262, June 2002.
- [124] R. Sacchi, J. F. Poliakoff, P. D. Thomas, and K. . Hafele, “Improved extraction of planar segments for scanned surfaces,” in *2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics*, pp. 325–330, 2000.
- [125] J. A. Kjellander and M. Rahayem, “Planar segmentation of data from a laser profile scanner mounted on an industrial robot,” *The International Journal of Advanced Manufacturing Technology*, vol. 45, no. 1-2, pp. 181–190, 2009.
- [126] L. Piegl, “On NURBS: a survey,” *IEEE Computer Graphics and Applications*, vol. 11, no. 1, pp. 55–71, 1991.

- [127] V. Markovic, Z. Jakovljevic, and Z. Miljkovic, “Feature sensitive three-dimensional point cloud simplification using support vector regression,” *Tehnicki Vjesnik/Technical Gazette*, vol. 26, no. 4, pp. 985–994, 2019.
- [128] A. Fitzgibbon, M. Pilu, and R. B. Fisher, “Direct least square fitting of ellipses,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 476 – 480, May 1999.
- [129] J. Stanić, *Metod inženjerskih merenja*. Beograd: Mašinski fakultet, 1990.
- [130] R. Halır and J. Flusser, “Numerically stable direct least squares fitting of ellipses,” in *Proceedings of 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*, vol. 98, pp. 125–132, Citeseer, 1998.
- [131] Z. Jakovljevic, V. Markovic, and M. Miladinovic, “Recognition of elliptical segments in scanned lines,” in *Proceedings of the Etikum Conference*, vol. 6, pp. 19–22, 2014.
- [132] V. Marković and Ž. Jakovljević, “Segmetnacija jedne klase površi drugog reda iz struktuiranog oblaka tačaka,” in *Zbornik radova 39. JUPITER konferencije*, pp. 4.13 – 4.22, 2014.
- [133] V. Markovic and Z. Jakovljevic, “Recognition of one class of surfaces from structured point cloud,” *FME Transactions*, vol. 45, no. 4, pp. 481 – 490, 2017.
- [134] Z. Jakovljevic, V. Markovic, and S. Zivanovic, “Recognition of quadrics from 3D point clouds generated by scanning of rotational parts,” *Journal of Production Engineering*, vol. 19, no. 1, pp. 65–68, 2016.
- [135] Z. Jakovljevic and V. Markovic, “Recognition of one class quadric surfaces from unstructured point cloud,” in *Proceedings of the 8th International Working Conference “Total Quality Management – Advanced and Intelligent Approaches*, pp. 353–360, 2015.
- [136] Z. Jakovljevic, V. Markovic, R. Puzovic, and V. Majstorovic, “Recognition of one class of quadrics from 3D point clouds,” *Procedia CIRP*, vol. 57, pp. 292 – 297, 2016.
- [137] P. L. Rosin, “A note on the least squares fitting of ellipses,” *Pattern Recognition Letters*, vol. 14, no. 10, pp. 799–808, 1993.
- [138] X. Ying, L. Yang, J. Kong, Y. Hou, S. Guan, and H. Zha, “Direct least square fitting of ellipsoids,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3228–3231, Nov 2012.
- [139] A. R. Webb, *Statistical pattern recognition - Third Edition*. UK: John Wiley & Sons, 2011.
- [140] K. Fukunaga, *Introduction to statistical pattern recognition- Second Edition*. San Diego: Academic Press, 2013.
- [141] M.-L. Zhang and Z.-H. Zhou, “A k-nearest neighbor based algorithm for multi-label classification,” in *2005 IEEE international conference on granular computing*, vol. 2, pp. 718–721, IEEE, 2005.
- [142] A. Al-sharadqah and N. Chernov, “Error analysis for circle fitting algorithms,” *Electronic Journal of Statistics*, vol. 3, 07 2009.
- [143] “CloudCompare - 3D point cloud and mesh processing software.” <https://www.danielgm.net/cc/>. Poslednji pristup: Dec. 2019.

- [144] “Range Vision Pro 3D scanner.” <https://rangevision.com/en/products/pro/>. Poslednji pristup: Dec. 2019.
- [145] V. Marković, Ž. Jakovljević, and I. Budak, “Prepoznavanje cilindara i ravni u trodimenzionim oblacima tačaka,” in *Zbornik radova 41. JUPITER konferencije*, pp. 2.9 – 2.17, 2018.
- [146] Z. Miljković, *Sistemi veštačkih neuronskih mreža u proizvodnim tehnologijama*. Beograd: Mašinski fakultet, 2003.
- [147] Z. Miljković, N. Vuković, M. Mitić, and B. Babić, “New hybrid vision-based control approach for automated guided vehicles,” *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 1-4, pp. 231–249, 2013.
- [148] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [149] J. N. Kapur, P. K. Sahoo, and A. K. Wong, “A new method for gray-level picture thresholding using the entropy of the histogram,” *Computer vision, graphics, and image processing*, vol. 29, no. 3, pp. 273–285, 1985.
- [150] B. Karasulu and S. Korukoglu, “A simulated annealing-based optimal threshold determining method in edge-based segmentation of grayscale images,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2246–2259, 2011.
- [151] P.-Y. Yin, “Multilevel minimum cross entropy threshold selection based on particle swarm optimization,” *Applied mathematics and computation*, vol. 184, no. 2, pp. 503–513, 2007.
- [152] Z. Miljković and D. Aleksendrić, *Veštačke neuronske mreže - zbirka rešenih zadataka sa izvodima iz teorije*. Beograd: Mašinski fakultet, 2009.
- [153] V. Marković, Ž. Jakovljević, and Z. Miljković, “Segmetnacija jedne klase površi drugog reda iz struktuiranog oblaka tačaka: problem određivanja pragova,” in *Zbornik radova 40. JUPITER konferencije*, pp. 4.7 – 4.17, 2016.

Biografija autora

Veljko Marković rođen je 16.06.1988. u Čupriji, Republika Srbija. Osnovnu školu „13. oktobar“ u Čupriji završio je odličnim uspehom zbog čega je nagrađen diplomom „Vuk Karadžić“. Srednju tehničku školu u Čupriji završio je odličnim uspehom i zapaženim rezultatima sa stručnih takmičenja zbog čega je proglašen đakom generacije.

Školske 2007/2008 upisao je Osnovne akademske studije (B.Sc.) na Mašinskom fakultetu u Beogradu. Osnovne studije završio je 2010. godine sa prosečnom ocenom 9.54. Diplomске akademske studije (M.Sc.) na Katedri za proizvodno mašinstvo Mašinskog fakulteta u Beogradu upisao je školske 2010/2011, a završio ih je 2012. godine sa prosečnom ocenom 9.95.

Tokom studiranja na Mašinskom fakultetu bio je više puta nagrađivan za odličan uspeh povodom Dana Fakulteta. Takođe, dvostruki je nosilac stipendije „Dositeja“ Fonda za mlade talente Republike Srbije, kao i stipendija Ministarstva prosvete.

Doktorske studije upisuje školske 2013/2014. godine gde polaže sve ispite predviđene Programom usavršavanja. Na osnovu naučno-istraživačkog rada tokom Doktorskih studija kao autor i koautor objavio je 10 radova u domaćim i međunarodnim časopisima kao i na naučnim skupovima. Takođe, učestvovao je u izvođenju nastave iz predmeta „Kompjuterska grafika“ koji se izvodi na Osnovnim studijama.

Veljko Marković poseduje višegodišnje radno iskustvo. Od 2012. godine do 2013. godine bio je zaposlen u preduzeću za projektovanje i proizvodnju medicinske opreme „Visaris“ u Zemunu gde je bio zadužen za procese projektovanja radiografskih uređaja i dodatne opreme. Nakon kraće pauze, 2014. godine ponovo se vraća u istu kompaniju na radno mesto rukovodioca mašinskog sektora razvoja medicinskih aparata. Početkom 2020. godine kao inženjer za podršku serijskoj proizvodnji zapošljava se u kompaniji „ZF“ u Pančevu gde i danas radi.

Prilog 1.

Izjava o autorstvu

Ime i prezime autora Veljko Marković

Broj indeksa D12/2013

Izjavljujem

da je doktorska disertacija pod naslovom

PREPOZNAVANJE GEOMETRIJSKIH PRIMITIVA IZ TRODIMENZIONALNOG
OBLAKA TAČAKA

- rezultat sopstvenog istraživačkog rada;
- da disertacija u celini ni u delovima nije bila predložena za sticanje druge diplome prema studijskim programima drugih visokoškolskih ustanova;
- da su rezultati korektno navedeni i
- da nisam kršio/la autorska prava i koristio/la intelektualnu svojinu drugih lica.

Potpis autora

U Beogradu, _____

Prilog 2.

Izjava o istovetnosti štampane i elektronske verzije doktorskog rada

Ime i prezime autora Veljko Marković

Broj indeksa D12/2013

Studijski program Doktorske akademske studije - Mašinsko inženjerstvo

Naslov rada PREPOZNAVANJE GEOMETRIJSKIH PRIMITIVA IZ TRODIMENZIONALNOG
OBLAKA TAČAKA

Mentor Prof. dr Živana Jakovljević

Izjavljujem da je štampana verzija mog doktorskog rada istovetna elektronskoj verziji koju sam predao/la radi pohranjivanja u **Digitalnom repozitorijumu Univerziteta u Beogradu**.

Dozvoljavam da se objave moji lični podaci vezani za dobijanje akademskog naziva doktora nauka, kao što su ime i prezime, godina i mesto rođenja i datum odbrane rada.

Ovi lični podaci mogu se objaviti na mrežnim stranicama digitalne biblioteke, u elektronskom katalogu i u publikacijama Univerziteta u Beogradu.

Potpis autora

U Beogradu, _____

Prilog 3.

Izjava o korišćenju

Ovlašćujem Univerzitetsku biblioteku „Svetozar Marković“ da u Digitalni repozitorijum Univerziteta u Beogradu unese moju doktorsku disertaciju pod naslovom:

PREPOZNAVANJE GEOMETRIJSKIH PRIMITIVA IZ TRODIMENZIONALNOG
OBLAKA TAČAKA

koja je moje autorsko delo.

Disertaciju sa svim priložima predao/la sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moju doktorsku disertaciju pohranjenu u Digitalnom repozitorijumu Univerziteta u Beogradu i dostupnu u otvorenom pristupu mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučio/la.

1. Autorstvo (CC BY)
2. Autorstvo - nekomercijalno (CC BY-NC)
3. Autorstvo - nekomercijalno - bez prerade (CC BY-NC-ND)
4. Autorstvo - nekomercijalno - deliti pod istim uslovima (CC BY-NC-SA)
5. Autorstvo - bez prerade (CC BY-ND)
6. Autorstvo - deliti pod istim uslovima (CC BY-SA)

(Molimo da zaokružite samo jednu od šest ponuđenih licenci.
Kratak opis licenci je sastavni deo ove izjave).

Potpis autora

U Beogradu, _____

1. **Autorstvo.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence, čak i u komercijalne svrhe. Ovo je najslobodnija od svih licenci.
2. **Autorstvo – nekomercijalno.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca ne dozvoljava komercijalnu upotrebu dela.
3. **Autorstvo – nekomercijalno – bez prerada.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, bez promena, preoblikovanja ili upotrebe dela u svom delu, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca ne dozvoljava komercijalnu upotrebu dela. U odnosu na sve ostale licence, ovom licencom se ograničava najveći obim prava korišćenja dela.
4. **Autorstvo – nekomercijalno – deliti pod istim uslovima.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence i ako se prerada distribuira pod istom ili sličnom licencom. Ova licenca ne dozvoljava komercijalnu upotrebu dela i prerada.
5. **Autorstvo – bez prerada.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, bez promena, preoblikovanja ili upotrebe dela u svom delu, ako se navede ime autora na način određen od strane autora ili davaoca licence. Ova licenca dozvoljava komercijalnu upotrebu dela.
6. **Autorstvo – deliti pod istim uslovima.** Dozvoljavate umnožavanje, distribuciju i javno saopštavanje dela, i prerade, ako se navede ime autora na način određen od strane autora ili davaoca licence i ako se prerada distribuira pod istom ili sličnom licencom. Ova licenca dozvoljava komercijalnu upotrebu dela i prerada. Slična je softverskim licencama, odnosno licencama otvorenog koda.