Izgradnja 3D modela drveta na temelju slike snimljene 3D kamerom

Rotim, Josip

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:205249

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-14**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Diplomski studij računarstva

IZGRADNJA 3D MODELA DRVETA NA TEMELJU SLIKE SNIMLJENE 3D KAMEROM

Diplomski rad

Josip Rotim

Osijek, 2016.

SADRŽAJ

1. UVOD 1

1.1. Motivacija izgradnje 3D modela stabla 1

1.2. RGB-D kamera 2

1.3. Zadatak 3

1.4. Metoda primjenjena u radu 3

1.5. Pregled rada po poglavljima 3

2. DETEKCIJA DIJELOVA STABLA METODOM NASUMIČNOG UZORKOVANJA 4

2.1. Općenito o RANSAC metodi 4

2.2. Primjena RANSAC-a na segmentaciju oblaka točaka na valjke 5

2.3. Normale i estimacija normala 8

3. Programska realizacija 12

3.1. Prikaz slike RGB-D kamere 12

3.2. Segmentacija valjkastih objekata iz spremljenih slika 14

4. Rezultati pokusa 18

4.1. Segmentacija jedne grane uz različite vrijednosti parametra DT 19

4.2. Segmentacija jedne grane uz različite vrijednosti parametra NDW 24

4.3. Segmentacija pet grana uz različite vrijednosti parametra DT 28

4.4. Segmentacija pet grana uz različite vrijednosti parametra NDW 31

4.5. Segmentacija deset grana uz različite vrijednosti parametra DT 35

4.6. Segmentacija deset grana uz različite vrijednosti parametra NDW 36

4.7. Segmentacija trideset grana uz različite vrijednosti parametra DT 37

4.8. Segmentacija trideset grana uz različite vrijednosti parametra NDW 38

5. Zaključak 39

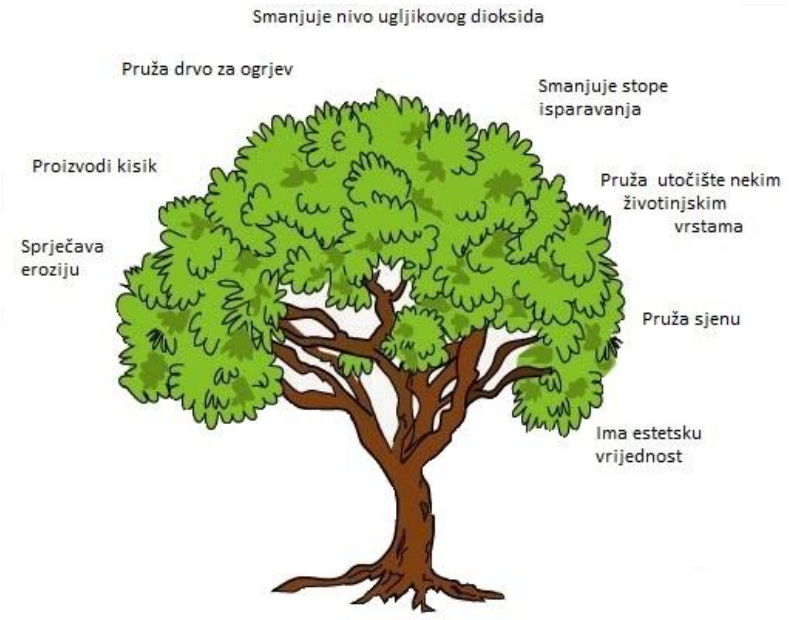
6. Literatura 40

7. Sažetak 41

1. UVOD

1.1. Motivacija izgradnje 3D modela stabla

Stabla obavljaju bitan posao za dobrobit svih vrsta na planeti. Ne samo da uljepšavaju planetu, proizvode hranu te su životni prostor nekih životinja, ona također apsorbiraju ugljični dioksid i daju kisik koji doprinosi održavanju i razvoju gotovo svih živih organizama. Detekcija stabala bitan je zadatak raznovrsnih projekata kao što je urbano planiranje, eko monitoring i automatska navigacija. Brzi tehnološki napredak posljednjih godina nudi nam mogućnost virtualnog gradskog modeliranja. Ugrađivanje komponenata okoliša u model omogućuje bolje planiranje i donošenje odluka. U tom kontekstu, integracija 3D modela stabala daje potpuniji i realniji pogled u virtualnom gradskom modeliranju. Za znanstvenike, općine grada i vladine agencije bitno je pratiti broj stabala i mjeriti njihove fizičke atribute za znanstvene analize kao što su npr. volumen absorbiranja ugljičnog dioksida. U šumarstvu je cilj istraživanja stabala dobivanje parametara koji omogućuju praćenje razvoja stabala u određenoj okolini, kako u ekološkom (npr. reakcija stabla tokom njegovog rasta u okolini te utjecaji vjetra na njegov rast) tako i u gospodarskom smislu (procjena količine stabala za proizvodnju,otkrivanje ozljeda tijekom procesa rasta itd). Jedan od ciljeva takvog istraživanja može biti prikupljanje informacija o duljini i promjeru debla, grana i sl. Tehnologija skeniranja senzorom pruža mogućnost obavljanja svih ovih fizička mjerenja u računalnom okruženju bez ometanja prirode i trošeći manje ljudskog napora.



Slika 1.1. Utjecaj stabla na okoliš (slika preuzeta sa "http://priyasethi.blogspot.hr")

1.2. RGB-D kamera

RGB-D kamera na sebi ima tri tipa senzora. Sliku detektira uz pomoć video kamere visoke razlučivosti, zvuk uz pomoć mikrofona i dubinu uz pomoć infracrvenog projektora i infracrvene kamere. Microsoft Kinect i Asus Xtion Pro Live su neki od modela RGB-D kamera. Za potrebe ovog rada korištena je kamera Asus Xtion PRO LIVE.

Slika 1.2. Asus Xtion PRO LIVE [1]

Vizualni senzori ovakvog tipa su općenito jeftini i daju veliku količinu informacija. Primjena ovih senzora se može naći u različitim granama tehnike npr. robotika, medicina, geodezija itd.

Senzor RGB-D kamere se bazira na projiciranju uzorka točaka u infracrvenom spektru te potom snimanja refleksije svjetlosti na sceni pomoću infracrvene kamere. Uspoređujući položaj snimljene točke s određenim referentnim položajem odnosno modelom vrši se triangulacija i izračunava se dubina te točke. Prema [1], Asus Xtion PRO LIVE daje izuzetno gustu dubinsku sliku rezolucije 640x480 točaka pri osvježavanju slike 30Hz. Asus Xtion PRO LIVE ima širinu slike horizontalnog pogleda 58°, vertikalnog od 45° i dijagonalnog od 70°. Prema specifikacijama proizvođača preporučeno radno područje je od 0.8m do 3.5m. Razlučivost infracrvene kamere iznosi maksimalno 320 x 240 piksela.

1.3. Zadatak

U okviru diplomskog rada potrebno je izraditi 3D model drveta na temelju slike snimljene 3D kamerom.

1.4. Metoda primjenjena u radu

Kao što je već prethodno spomenuto u ovom radu za potrebe snimanja slika korištena je RGB-D kamera Asus Xtion PRO LIVE. Microsoft Visual Studio 2012 Express for Desktop je integrirano razvojno okruženje (eng. IDE) koje je korišteno za implementiranje algoritma napisanog u programskom jeziku C++. Za izvedbu ovoga rada korištena je biblioteka PCL (eng. Point Cloud Library). PCL je biblioteka koja služi za obradu oblaka točaka, 2D i 3D slika. Otvorenog je koda, te je besplatna za komercijalne i istraživačke svrhe. U radu je PCL primjenjen na način da u oblaku točaka detektira valjkaste površine RANSAC metodom.

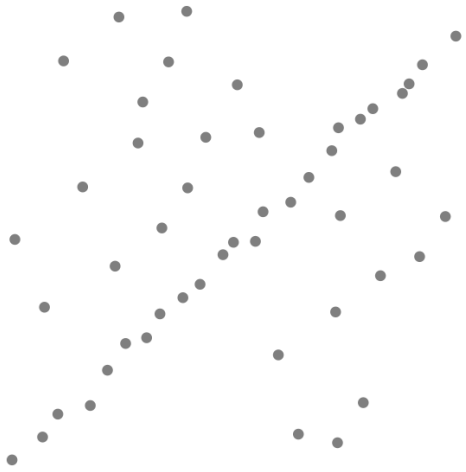
1.5. Pregled rada po poglavljima

U drugom poglavlju opisana je detekcija dijelova stabla metodom nasumičnog uzorkovanja. Prikazan je sažetak rada RANSAC metode, primjena RANSAC-a na segmentaciju oblaka točaka na valjke i normale i njihova estimacija. U trećem poglavlju opisana je programska realizacija. Prikazano je na koji način je realiziran program. U četvrtom poglavlju prikazani su rezultati pokusa korištenjem grafova, te slike koje predstavljaju rezultate implementiranog programa. U petom poglavlju napisan je zaključak ovog rada.

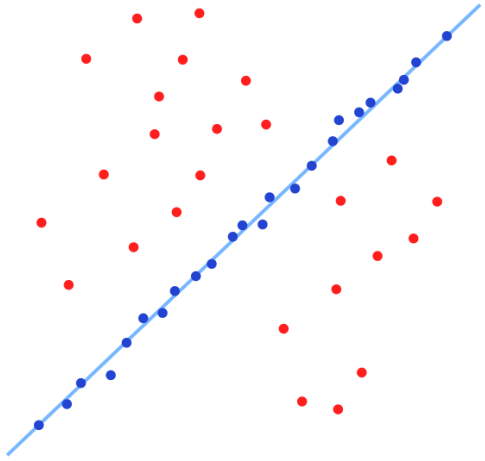
2. DETEKCIJA DIJELOVA STABLA METODOM NASUMIČNOG UZORKOVANJA

2.1. Općenito o RANSAC metodi

Prema [2], RANSAC (eng. Random Sample Consensus) je iterativna metoda za procjenu parametara matematičkog modela iz skupa podataka koja uključuje i outliere. To je nedeterministički algoritam koji pronalazi točne rezultate samo sa određenom sigurnošću, koja se povećava povećanjem iteracija. Algoritam je prvi puta objavljen od strane Fischlera i Bollesa 1981. Osnovna pretpostavka je da se podaci sastoje od inliera (podataka čija se distribucija može objasniti skupom parametara modela) i outliera (oni ne pripadaju populaciji koju modeliramo). Glavna ideja je slučajnim odabirom minimalnog uzorka hipotetizirati traženi model. Hipoteze evaluirati nad svim elementima ulaznog skupa podataka. Algoritam vraća najbolju hipotezu iz skupa inliera. Primjer jednog takvog matematičkog modela je pravac. Na slici 2.1. je prikazan skup točaka kroz koji je potrebno provući pravac, a na slici 2.2. rezultat RANSAC algoritma. Plave točke su inlieri – točke koje otprilike tvore pravac, a crvene outlieri – točke koje nikako ne mogu tvoriti taj pravac. Nekom jednostavnijom metodom za pronalaženje takvoga pravca bi uglavnom generirala lošiji pravac, jer za razliku od RANSAC algoritma ne bi zanemarila točke koje su predaleko od pravca (outliere).



Slika 2.1. Skup podataka kroz koji se treba provući pravac [2]



Slika 2.2. Pravac provučen metodom RANSAC [2]

Vrijednosti parametara *t* i *d* moraju se odrediti iz specifičnih zahtjeva vezanih uz aplikaciju i skup podataka, eventualno na temelju eksperimentalne evaluacije. Međutim, parametar broj iteracija *k* može se odrediti iz teorijskog rezultata. Neka je *p* vjerojatnost da RANSAC algoritam

u nekoj iteraciji odabere samo inliere iz ulaznog seta kada odabere *n* točaka iz kojih se procjenjuju parametri modela. Kada se to dogodi, rezultat će vjerojatno biti koristan, pa *p* daje vjerojatnost da algoritam daje koristan rezultat. Neka je *w* vjerojatnost odabira inliera svaki put kada se odabere jedna točka.

w = broj inliera u podacima / broj točaka u podacima

Uobičajeni slučaj je da *w* nije unaprijed poznat, ali može se dati neka gruba vrijednost. Pod pretpostavkom da su *n* točke koje su samostalno odabrane za potrebu procjene modela, w^n je vjerojatnost da su sve *n* točke inilieri, a $1 - w^n$ je vjerojatnost da je barem jedna od *n* točaka outlier. Vjerojatnost da algoritam neće odabrati skup od *n* točaka u kojemu su sve inlieri je $1 - w^n$,što je jednako $1 - p$, tj.

$$1 - p = (1 - w^n)^k \tag{2-1}$$

Gdje primjena logaritama na obje strane jednadžbe (2-1) dovodi do:

$$k = \frac{\log(1 - p)}{\log(1 - w^n)} \tag{2-2}$$

Ovaj rezultat pretpostavlja da su podatkovne točke odabrane neovisno, to jest, točka koja je odabrana jednom je zamijenjena i može se ponovno odabrati u istoj iteraciji.

2.2. Primjena RANSAC-a na segmentaciju oblaka točaka na valjke

Za razliku od ravnine pronalaženje valjka u oblaku točaka je složenije jer se parametri valjka ne mogu lako odrediti iz nekoliko točaka na njegovoj površini. Valjak u 3D prostoru definiran je jednadžbom (2-3). Pet parametara (ρ, θ, φ, a, r) je potrebno da se odredi valjak.

$$|(p - q) \times a| - r = 0, \tag{2-3}$$

gdje je:

p = točka na površini valjka,

q = točka na osi valjka,

a = jedinični vekor smjera valjka,

r = radijus valjka,

× - vektorski produkt vektora

|·| - duljina vektora

$$q = (\rho + r) * n, \tag{2-4}$$

gdje je:

ρ = udaljenost od ishodišta [0, 0, 0] do najbliže točke na površini valjka,

n = ($\cos\varphi \sin\theta$, $\sin\varphi \sin\theta$, $\cos\theta$) – jedinični vektor koji definira smjer pravca ortogonalnog na os valjka koji prolazi ishodištem,

φ = kut između projekcije n na xy ravninu i osi x,

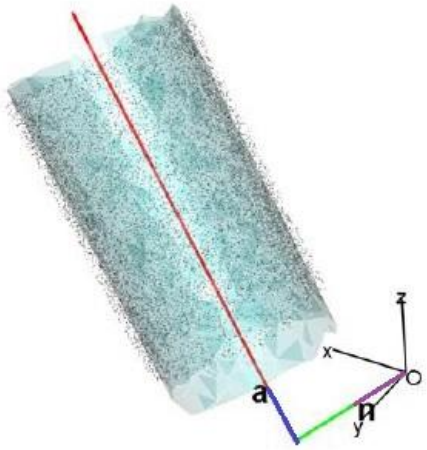
θ = kut između n i osi z.

$$a = n^{\theta} \cos\alpha + n^{\varphi} \frac{\sin\alpha}{\sin\theta}, \tag{2-5}$$

gdje je:

α = kut između a i n^{θ},

n^{θ} i n^{φ}= parcijalne derivacije od n po kutovima θ i φ.



2.3. Odnos vektora *n* i $\boldsymbol{a}$ pri određivanju valjka

Slika (2.3) prikazuje valjak koji se pronalazi korištenjem jednadžbe (2-3), gdje je *n* vektor ljubičaste boje koji počinje u ishodištu koodrinatnog sustava i ima duljinu 1. Vektor *n* je ortogonalan na os valjka *a* koji je na slici prikazan plavom bojom. Polazna točka vektora *a* je točka q koja se dobije korištenjem jednadžbe (2-4), što je točka na udaljenosti $(\rho + r)$ od ishodišta do osi valjka u smjeru *n*, prikazan svijetlo zelenom bojom.

U prethodnom poglavlju (2.1.) objašnjen je rad RANSAC algoritma na primjeru pravca, za potrebe ovog rada RANSAC algoritam koristi se za detekciju valjkastih površina korištenjem jednadžbi (2-3), (2-4) i (2-5).

U sklopu rada opisane su funkcije klase sac_model_cylinder.h.

Tablica 2.1. Pripadne funkcije klase sac_model_cylinder.h [3]

Tip varijable	Pripadne funkcije	Opis pripadne funkcije
	SampleConsensusModelCylinder (const PointCloudConstPtr &cloud)	Konstruktor za bazu SampleConsensusModelCylinder
void	setEpsAngle (double ea)	Postavi kut udaljenosti za epsilon (delta) .
double	getEpsAngle ()	Preuzmi kut udaljenosti za epsilon (delta) .
void	setAxis (const Eigen::Vector3f &ax)	Postavi os duž koje će se tražiti smjer valjka
Eigen::Vector3f	getAxis ()	Preuzmi os duž koje je tražen smjer valjka
void	getSamples (int &iterations, std::vector< int > &samples)	Uzmi dvije nasumične točke s njihovim normalama i vrati ih kao indekse točaka
bool	computeModelCoefficients (const std::vector< int > &samples, Eigen::VectorXf &model_coefficients)	Provjeriti mogu li dani uzorci indeksa stvoriti valjani model valjka, izračunati koeficijente modela iz tih uzoraka i pohraniti ih u model_coefficients.
void	getDistancesToModel (const Eigen::VectorXf &model_coefficients, std::vector< double > &distances)	Izračunaj sve udaljenosti od podataka oblaka do zadanog modela valjka.
void	selectWithinDistance (const Eigen::VectorXf &model_coefficients, double threshold, std::vector< int > &inliers)	Odaberi sve točke koje za zadani model_coefficients pripadaju inlierima
void	optimizeModelCoefficients (const std::vector< int > &inliers, const Eigen::VectorXf &model_coefficients, Eigen::VectorXf &optimized_coefficients)	Ponovno izračunaj koeficijente cilindra pomoću zadanog inlier seta i vrati ih korisniku.
void	projectPoints (const std::vector< int > &inliers, const Eigen::VectorXf &model_coefficients, PointCloud &projected_points, bool copy_data_fields=true)	Stvori novi oblak točaka sa inlierima koji projektriraju model valjka

bool	doSamplesVerifyModel (const std::set< int > &indices, const Eigen::VectorXf &model_coefficients, double threshold)	Provjeri provodi li podskup indeksa zadani koeficijent modela valjka.
pcl::SacModel	getModelType () const	Vrati jedinstveni ID za ovaj model (SACMODEL_CYLINDER).

2.3. Normale i estimacija normala

RANASAC koji je korišten u sklopu izrade ovoga rada koristi informacije o normalama. Normale su bitno svojstvo površina i često se koriste u računalnoj grafici. Normala je vektor koji je okomit na objekt o kojem se govori, npr. normala na krvulju, normala na površinu i sl. Dva su dominatna pristupa estimiranju normala u oblaku točaka. Jedna je numerička koja primjenjuje tehnike optimiranja, a druga je uglavnom kombinacija Delaunay/Voronoi svojstava.

Prema [5], u numeričkom pristupu široko korištena tehnika je pronaći odgovarajući skup točaka u lokalnom susjedstvu neke točke p te zatim procjeniti koja je najbolja ravnina koja pripada tim točkama. Normala ravnine je estimirana normala točke *p*. Za određivanje normala površine u nekoj točki p_q koriste se njene susjedne točke $p^k = \{p_1, p_2 ... p_k\}$ (često zvane k-susjedstva, eng. k - neighborhood). Problem određivanja normale neke točke na površini može se aproksimirati problemom određivanja normale ravnine koja je tangentna na površinu gdje se točka nalazi, što ustvari postaje problem određivanja ravnine s najmanjim kvadratnim odstupanjem od točaka u p^k. Ravnina je prikazana kao točka *x* koja leži u danoj ravnini i normala *n*. Udaljenosti d_i od točaka p_i u susjedstvu p^k za ravninu su:

$$di = (pi - x) * n. \tag{2-6}$$

Točka *x* i normala *n* su izračunate metodom najmanjeg kvadrata. To jest, točka *x* se određuje kao 3D centroid susjednih točaka p^k

$$x = \frac{1}{k} * \sum_{i=1}^{k} p_i \tag{2-7}$$

a normala *n* je dana analizom svojstvenih vrijednosti λ_j i svojstvenih vektora v_j (ili PCA- eng. Principial Component Analysis) matrice kovarijance C

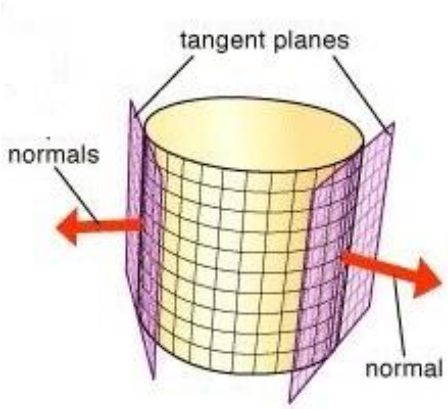
$$C = \frac{1}{k} * \sum_{i=1}^{k} (p_i - x) * (p_i - x)^T, \tag{2-8}$$

$$C * v_j = \lambda_j * v_j \ , \tag{2-9}$$

$$j \epsilon \{0, 1, 2\} \ . \tag{2-10}$$

Za odabir skupa točaka p^k mogu se uzeti sve točke unutar kugle polumjera r ili k najbližih točaka Estimacija normala površine može se sažeti na sljedeće:

1. Pronaći najbliže susjede od točke oblaka *p*.
2. Izračunati normalu površine *n* od točke *p*.



Slika 2.4. Normale valjka

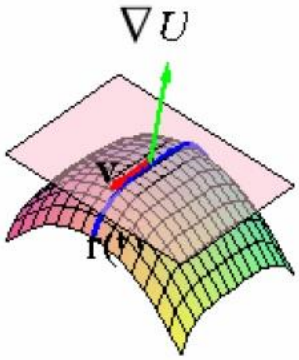
Prema [6], neka ploha *U(x, y, z) = k* je glatka ploha ako je gradijent $\nabla U = \langle U_x \ , \ U_y \ , \ U_z \rangle$ neprekidan i različit od nule u bilo kojoj točki plohe. Pretpostavimo da krivulja $r(t) = \langle x(t), y(t), z(t) \rangle$ leži na glatkoj plohi *U(x, y, z) = k*. Primjeni se derivacija po *t* na obje strane jednadžbe glatke plohe:

$$\frac{dU}{dt} = \frac{d}{dt} k, \tag{2-11}$$

Budući da je *k* konstanta dobije se:

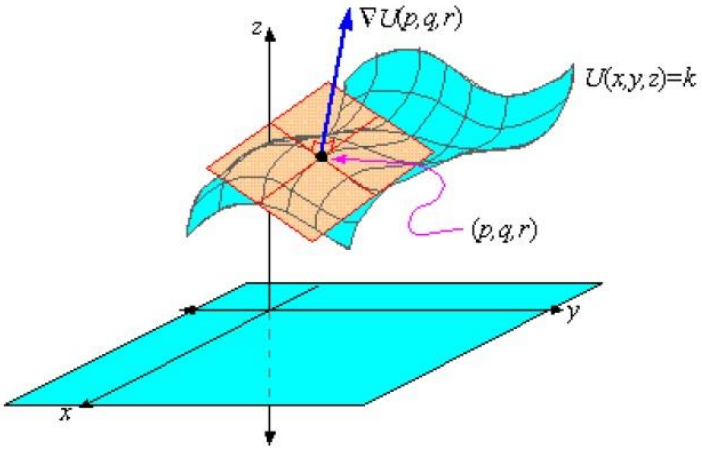
$$\nabla U \cdot v = 0, \tag{2-12}$$

gdje je $v = \langle x'(t), y'(t), z'(t) \rangle$. Vektor v je tangentan na plohu zato što je tangentan na krivulju koja leži na plohi, što implicira da je ∇U ortogonalan na svaki tangentan vektor v u datoj točki na plohi.



Slika 2.5. Krivulja *r(t)*, vektor *v* i gradijent $\boldsymbol{\nabla U}$ [6]

Što znači da je $\nabla U = (p, q, r)$ u danoj točki (p, q, r) normala na tangentnu ravninu glatke plohe $U(x, y, z) = k$ za točku (p, q, r).

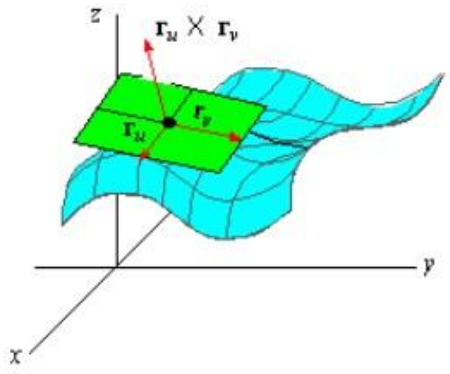


Slika 2.6. Gradijent $\boldsymbol{\nabla U}$ kao normala na površinu [6]

Proizlazi da je gradijent ∇U normala na glatku plohu $U(x, y, z) = k$ za bilo koju točku plohe.

Normala i tangentna ravnina na parametarske površine:

Ako je $r(u, v)$ regularna parametrizacija plohe, tada je $r_u \times r_v$ vektor okomit na vektor r_u i na vektor r_v. Vektori r_u i r_v su radij-vektori točke u kojoj tangencijalna ravnina dira plohu. Također je okomit na ravninu tangente na kojoj se nalaze vektori r_u i r_v.



2.7. Normala na zakrivljenu površinu [6]

Vektor $r_u \times r_v$ predstavlja normalu na plohu, također se može koristiti kao vektor normale u određivanju jednadžbe tangentne ravnine u točki na način da je $(x_1,\ y_1, z_1) = r(p, q)$.
Ako je $r(u, v)$ parametrizacija plohe, tada je jedinica normale plohe difinirana na način:

$$n = \frac{r_u \times r_v}{\|r_u \times r_v\|}, \tag{2-13}$$

Vektor *n* je također normala na plohu.

3. Programska realizacija

U okviru ovog rada izrađena su dva projekta u Microsoft Visual Studiu 2012 integriranom razvojnom okruženju uz korištenje PCL-a. Prvi projekt imena kinect služi za prikaz slike koju snima RGB-D kamera. Projekt je izrađen tako da u prozoru prikazuje RGB dubinsku sliku oblaka točaka. Pritiskom na tipku tipkovnice program sprema dubinsku sliku u .pcd formatu. Drugi projekt imena roller služi za segmentaciju valjkastih objekata RANSAC metodom. Oba projekta koriste PCL biblioteku. Što znači da nakon preuzimanja biblioteke na računalo da bi ju koristili moramo postaviti varijable okoline (eng. Enviroment Variables) na način kako je prikazano u tablici (3.1).

Ime varijable (*eng. Variable name*)	Vrijednost (*eng. Value)*
PCL_ROOT	C:\Program Files\PCL 1.7.2 (ili C:\Program Files (x86)\PCL 1.7.2)
Putanja (*eng. Path*)	;%PCL_ROOT%\bin ;%OPENNI2_REDIST64% (ili %OPENNI2_REDIST%)

Tablica 3.1. Postavljanje varijabli okoline

3.1. Prikaz slike RGB-D kamere

Prvi projekt je izrađen tako da u prozoru prikazuje RGB dubinsku sliku oblaka točaka, te pritiskom na određenu tipku tipkovnice sprema sliku u .pcd formatu.U ovom poglavlju opisane su neke bitne naredbe koje su korištene za izradu projekta. U početku projekta prije inicijalizacije kamere potrebno je uključiti određene datoteke iz PCL-a koje će se koristiti. Te datoteke su:

```
#include <pcl/point_types.h>
#include <pcl/io/pcd_io.h>
#include< pcl/io/boost.h>
#include< pcl/visualization/cloud_viewer.h>
#include< pcl/io/ply_io.h>
```

Da bi dobili dubinsku sliku oblaka točaka u projektu je deklarirano dvodimenzionalno polje 'depthImage' tipa int koji sadrži elemente w(width) i h(height) te su im pridružene vrijednosti 640 za širinu i 480 za visinu. Umnožak elemenata 'depthImage.w' i 'depthImage.h' se sprema u novu varijablu 'nPix' tipa int koja predstavlja rezoluciju kamere.

```
RVL::Array2D<short> depthImage;
depthImage.w = 640;
depthImage.h = 480;
int nPix = depthImage.w * depthImage.h;
depthImage.Element = new short int[nPix];
```

U poglavlju 1.2. objašnjeno je da razlučivost infracrvene kamere iznosi maksimalno 320x240 piksela. Stoga u dvije nove deklarirane varijable wRGB i hRGB tipa int dubinska slika dijeli se sa brojem 2.

```
int wRGB = depthImage.w / 2;
int hRGB = depthImage.w / 2;
```

Slijedećim naredbama kreiraju se dva prozora, gdje jedan prozor prikazuje sliku kamere u boji, a drugi prikazuje dubinsku sliku kamere.

```
IplImage *RGBImage = cvCreateImage(cvSize(wRGB, hRGB), IPL_DEPTH_8U, 3);
IplImage *GSImage = cvCreateImage(cvSize(wRGB, hRGB), IPL_DEPTH_8U, 1);
IplImage *depthDisplayImage = cvCreateImage(cvSize(depthImage.w, depthImage.h),
IPL_DEPTH_8U, 3);
```

Spremanje slike u .pcd formatu napravljeno je korištenjem naredbe switch gdje pritiskom na tipku 'g' dubinsku sliku kamere iz oblaka točaka sprema unutar mape gdje se nalazi projekt pod imenom 'stablo.pcd'.

```
switch (key){
        case 'g':
                camera.GetPointCloud(&(depthImage), RGBImage, PC);
                viewer.showCloud(PC);
                pcl::io::savePCDFile("stablo.pcd", *PC);
                break;
                case 0x0000001B:
                break;
}
```

3.2. Segmentacija valjkastih objekata iz spremljenih slika

U ovom poglavlju je objašnjeno na koji način je izrađen projekt za segmentiranje valjkastih objekata RANSAC metodom korištenjem slika koje su dobivene korištenjem prvog projekta. Objasnit će se neke bitne metode koje su korištene tokom izrade projekta. U radu su korištene datoteke iz PCL-a, koje su uključene u početku izrade projekta.

```
#include <pcl/ModelCoefficients.h>
#include <pcl/io/pcd_io.h>
#include <pcl/point_types.h>
#include <pcl/filters/extract_indices.h>
#include <pcl/filters/passthrough.h>
#include <pcl/features/normal_3d.h>
#include <pcl/sample_consensus/method_types.h>
#include <pcl/sample_consensus/model_types.h>
#include <pcl/segmentation/sac_segmentation.h>
```

Za potrebe ovog projekta deklarirano je šest oblaka točaka.

```
pcl::PointCloud<PointT>::Ptr cloud (new pcl::PointCloud<PointT>);
```

Oblaci točaka služe za spremanje dubinske slike i za razne manipulacije nad podacima koje sadrži taj oblak točaka. Također deklarirana su i tri oblaka točaka koji će poslužiti za estimaciju normala valjkastih objekata.

```
pcl::PointCloud<pcl::Normal>::Ptr cloud_normals (new pcl::PointCloud<pcl::Normal>);
```

Cilj projekta je segmentirati valjkaste objekte RANSAC metodom. Da bi to bilo moguće potrebno je učitati sliku koja je dobivena korištenjem RGB-D kamere. To je urađeno na način da je u projektu deklariran objekt 'reader' klase PCDReader koji se nalazi u biblioteci oblaka točaka. Zatim se metodom 'read' učitava slika i sprema u oblak.

```
reader.read ("stablo.pcd", *cloud);
```

Nakon što je slika učitana kreira se filter koji služi za filtriranje učitanih slika od točaka bez definirane dubine. Također, izdvajaju se samo točke koje su unutar zadanog intervala udaljenosti od kamere. Da bi se to realiziralio deklariran je objekt 'pass' za klasu PassThrough iz datoteke oblaka točaka. Granice filtriranja su postavljene kao parametri funkcije setFilterLimits a iznose (0, 3.5) m. Pozivanjem funkcije 'filter' filtriraju se podaci oblaka točaka i spremaju u novi oblak točaka.

```
pass.filter (*cloud_filtered);
```

Nakon što je slika filtrirana slijedeće što se radi je estimacija normala. U projektu je deklariran objekt 'ne' za klasu NormalEstimation iz datoteke PCL-a. Za metodu pretraživanja najbližih

susjeda odabran je kd-tree algoritam. K-d stablo ili k-dimenzionalno stablo je struktura podataka koja se koristi u računalnoj znanosti za organiziranje određenog broja točaka u prostoru s k dimenzija.To je binarno "stablo" u kojemu je svaki čvor k-dimenzionalna točka.

```
ne.setSearchMethod (tree);
```

Računanje normala točaka vrši se pomoću metode 'compute' i rezultat sprema u oblak za normale.

```
ne.compute (*cloud_normals);
```

Nakon određivanja normala, pristupa se segmentaciji valjkastih objekata na slici RANSAC metodom. Da bi se segmentacija mogla obaviti potrebno je uvrstiti parametre segmentacije:

1. Max Iterations- maksimalan broj iteracija
2. Normal Distance Weight – težinski koeficijent s kojim se uzima u obzir odstupanje smjera normala
3. Distance Threshold - prag udaljenosti za inliere
4. Radius Limits – Granice radijusa valjka

Ideja unošenja parametara segmentacije je da se njihove vrijednosti učitavaju iz tekstualne datoteke 'config.txt' koja se nalazi u mapi gdje i projekt, i pohranjuju u varijable. U projektu je deklarirana varijabla 'MI' tipa int za maksimalan broj iteracija, te četiri varijable tipa double: 'NDW' za udaljenost normala, 'DT' za prag udaljenosti, 'RL1' i 'RL2' za granice radijusa. Uvrštavanje vrijednosti u varijable napravljeno je korištenjem 'ifstream' klase gdje je deklariran objekt 'inFile' koji pomoću metode 'open' otvara tekstualnu datoteku 'config.txt' sa numeričkim vrijednostima i te vrijednosti pohranjuje u varijable.

```
if(inFile.is_open()){
        inFile >> NDW >> MI >> DT >> RL1 >> RL2;
}
else{
        std::cout << "Error!" << std::endl;
}
```

Prednost korištenja tekstualne datoteke je taj što u dijelu koda gdje je potrebno upisati parametre segmentacije postavimo varijablu. Time je omogućeno da parametre segmentacije mijenjamo, a da pri tome ne mijanjamo kod što omogućuje brže testiranje. Pri postavljanju parametara potrebno je postaviti tip modela segmentacije. Za segmentiranje valjkastih objekata koristi se 'sacmodel_cylinder'. Model sadrži sedam koeficijenata valjka koji je određen točkom na svojoj osi(x,y,z), u smjeru osi (x,y,z) i radijusu.

Segmentacijom se pronalazi dominantni valjak na sceni, međutim stablo se sastoji od više grana stoga se i segmentacija mora izvršiti više puta kako bi se pronašle sve grane (dominatni valjci).
To je realizirano na način da korisnik ručno unese broj grana te se korištenjem 'for' petlje segmentacija izvršava onoliko puta koliki je broj grana unešen od strane korisnika. Unutar 'for' petlje nakon što se izvrše metode za segmentaciju, najdominantniji valjak se sprema u novi oblak imena 'cloud_result'. Nakon što je dominatni valjak spremljen u oblak za rezultate pristupa se bojanju spremljenog valjka, tj. bojanju svake točke koja predstavlja taj valjak. Bojanje se vrši na način da su deklarirane tri varijable tipa int imenom 'r' za crvenu, 'g' za zelenu i 'b' za plavu boju. U radu je korištena dubina boje od 8 bita što daje 256 kombinacija. Numeričke vrijednosti od 1 do 256 potrebno je spremiti u varijable koje predstavljaju boje. Spremanje je realizirano unutar varijable gdje se generira slučajni broj korištenjem funkcije rand().

```
r = 256 * rand()/(RAND_MAX + 1.0);
```

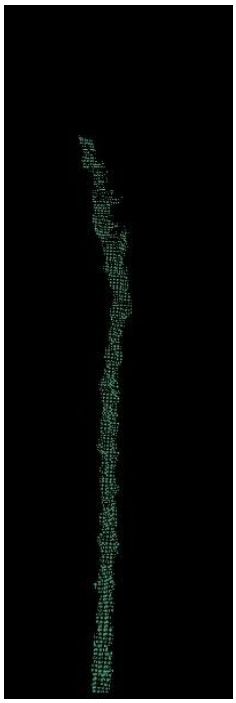
Zatim se unutar nove 'for' petlje vrijednosti varijable pridružuju točkama koje čine granu.

```
for(int i =0; i < cloud_result->points.size();i++)
		{
			cloud_result->points[i].r = r;
			cloud_result->points[i].g = g;
			cloud_result->points[i].b = b;
		}
```

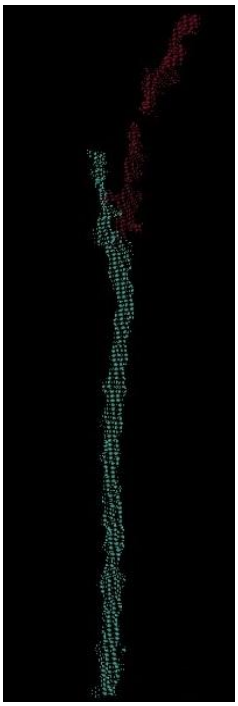
Nakon što se obojaju sve točke koje čine granu, spremaju se u oblak točaka imena 'cloud_tree'.

```
*cloud_tree+=*cloud_result;
```

U oblaku točaka iz kojeg se segmentiraju grane, nakon prve iteracije 'for' petlje, sadrži ukupan broj valjaka – 1, jer je najdominatniji valjak izvađen i spremljen u oblak za rezultate. Nakon toga se traži slijedeća dominantna grana na sceni iz oblaka gdje se nalazi preostali skup točaka.



3.1. Rezultat prve iteracije petlje



3.2. Rezultat druge iteracije petlje

Nakon što 'for' petlja izvrši sve iteracije obavljena je segmentacija svih grana na sceni. Iz oblaka 'cloud_tree' skup svih točaka potrebno je spremiti kao sliku u .pcd formatu. Ta slika predstavlja 3D model stabla snimljen RGB-D kamerom. Snimanje slike vrši se korištenjem klase PCDWriter iz biblioteke oblaka točaka za koju je u projektu deklariran objekt 'writer'. Pozivanjem objekta i korištenjem metode 'write' sprema se slika u .pcd formatu.

```
writer.write ("slikaStabla.pcd", *cloud_tree, false);
```

Prikaz slike u .pcd formatu snimljene Asus Xtion PRO LIVE kamerom izvršava se korištenjem aplikacije 'pcl_viewer_release.exe' koja se nalazi u 'bin' direktoriju biblioteke oblaka točaka. Pokretanje aplikacije 'pcl_viewer_release.exe' izvršava se korištenjem Command Prompt prozora (kratica cmd.exe). Da bi se aplikacija 'pcl_viewer_release.exe' pokrenula potrebno se je nalaziti u direktoriju gdje se aplikacija nalazi korištenjem naredbe 'cd' i upisom putanje do konačnog direktorija.

```
cd “C:\Program Files\PCL 1.7.2\bin“
```

Kada je pokazivač u 'bin' direktoriju, pokretanje aplikacije i prikaz slike u .pcd formatu izvršava se uspisom imena aplikacije i putanjom gdje se slika nalazi.

```
pcl_viewer_release “put\stablo.pcd“
```

4. Rezultati pokusa

U ovom poglavlju ispitat će se mogućnost segmentacije 3D snimke drveta na grane detekcijom valjkastih površina RANSAC metodom. Ispitat će se segmentacija traženjem jedne ili više grana iz skupa točaka uz različite kombinacije parametara RANSAC algoritma. U tablici u kojoj će se spremati rezultati imena parametara će biti skraćeno zapisana i to na način:

NDW – (eng. Normal Distance Weight), predstavljati težinu s kojom se u obzir uzimaju razlike orijentacije normala,

MI - (eng. Maxx Iterations), maksimalan broj iteracija,

DT – (eng. Distance Threshold), prag udaljenosti

RL1, RL2 – (eng. Radius Limits), granice radijusa

Ispitivanje će se izvršiti na stablu, slika (4.1.), snimljenom RGB-D kamerom, slika (4.2). Snimljeno stablo ne sadrži sve svoje grane iz razloga što dubinska kamera(infracrveni projektor i infracrvena kamera) u vanjskim uvjetima zbog svjetlosti i površine koja prekriva stablo ne daje najbolje rezultate.

Prvi dio testiranja bavit će se segmentiranjem jedne grane iz skupa oblaka točaka. Postavljanjem različitih kombinacija parametara, rezultati testiranja bit će prikazani grafovima. Parametri segmentacije prikazani su u tablici 4.1.

c	NDW	MI	DT	RL1	RL2
Vrijednost parametra:	0.0	10000	0.01	0	0.04

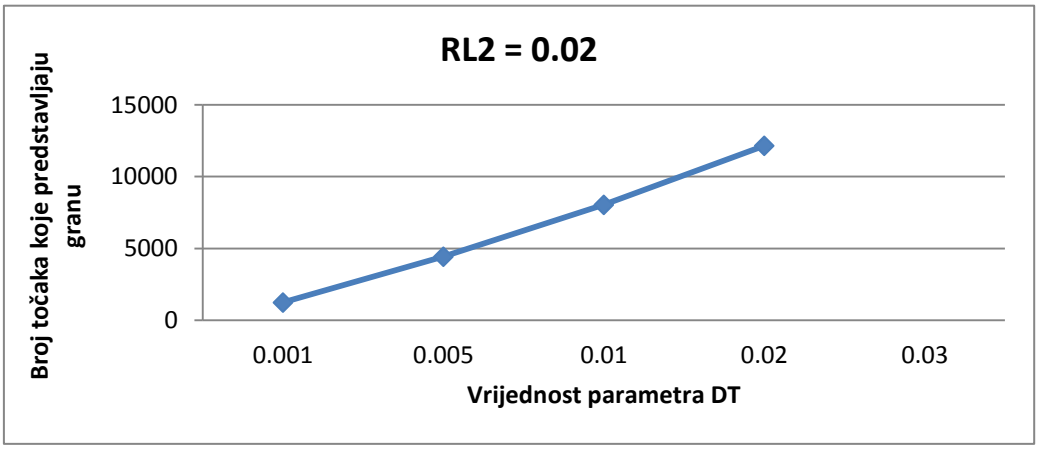
Tablica 4.1. Parametri segmentacije RANSAC metodom

4.1. Stablo snimljeno digitalnim fotoaparatom

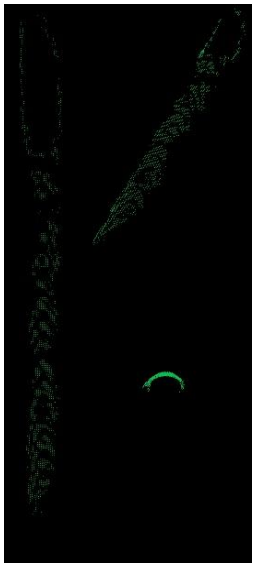
4.2. Stablo snimljeno dubinskom kamerom

4.1. Segmentacija jedne grane uz različite vrijednosti parametra DT

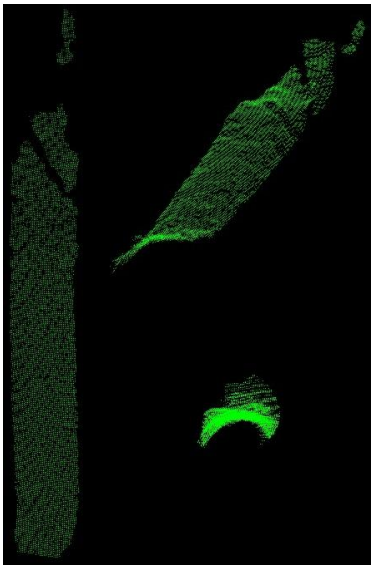
U prvom testiranju vrši se segmentacija jedne najdominantnije grane na sceni.



Slika 4.3. Promjena parametra DT za RL2 = 0.02

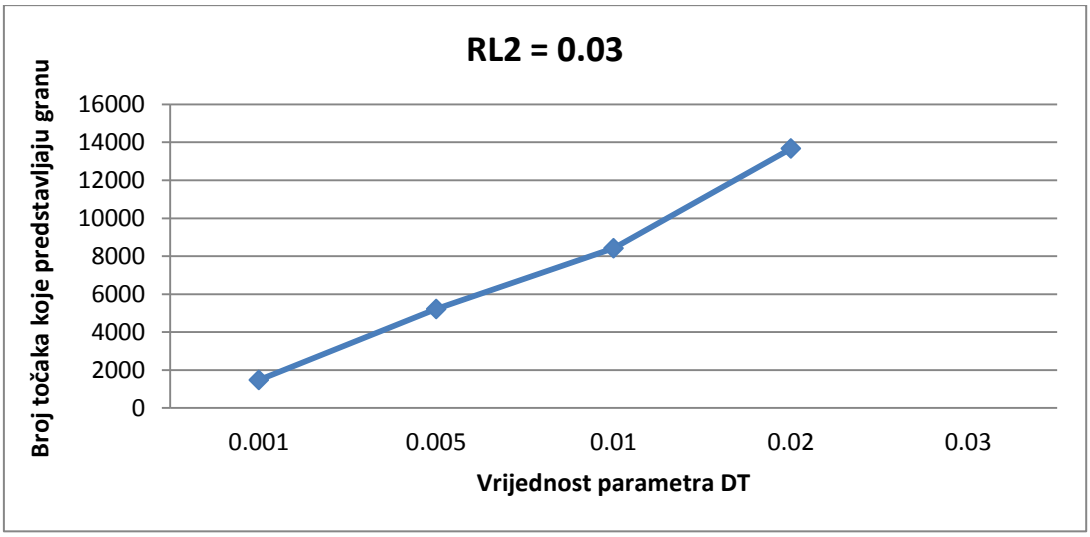


Slika 4.4. DT = 0.001

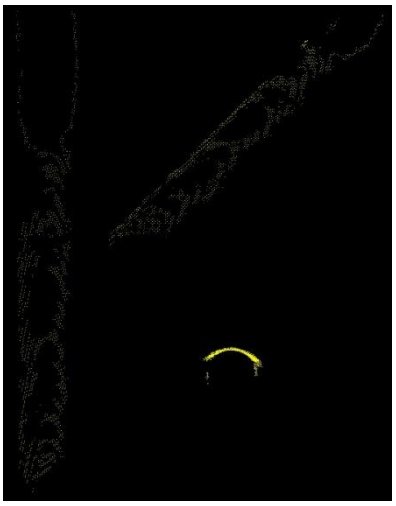


Slika 4.5. DT = 0.02

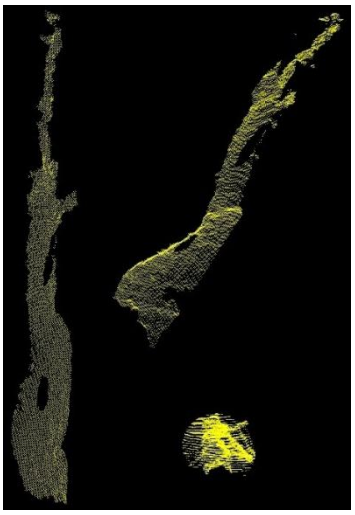
Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.02, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.3). Iz grafa se zaključuje da za manju vrijednost parametra DT segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju granu. Za vrijednost parametra DT = 0.03 RANSAC metoda ne pronalazi nijednu točku. Na slici (4.4) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.001 sa 1245 točaka koje predstavljaju granu. Na slici (4.5) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.02 sa 12154 točaka koje predstavljaju granu.



Slika 4.6. Promjena parametra DT za RL2 = 0.03

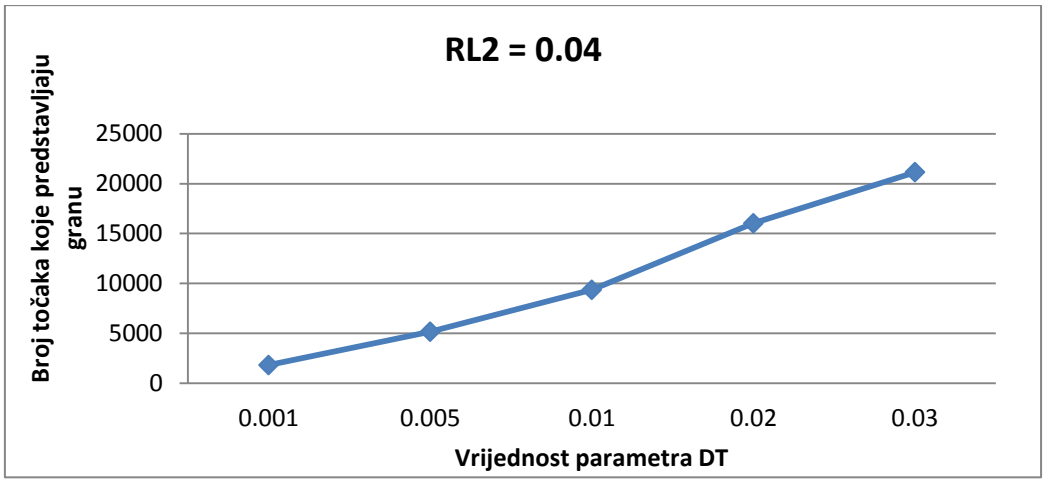


Slika 4.7. DT = 0.001

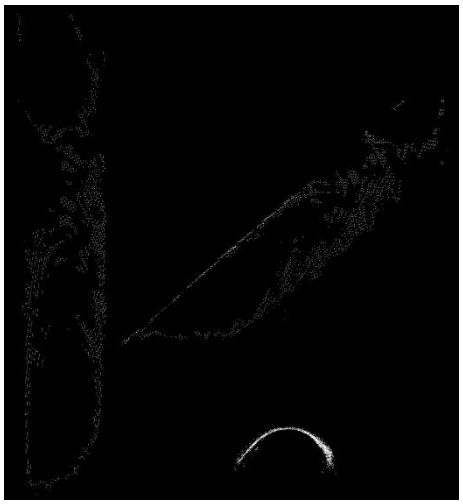


Slika 4.8. DT = 0.02

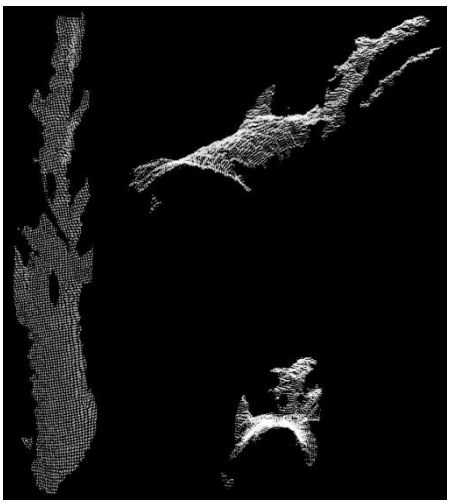
Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.03, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.6). Iz grafa se zaključuje da za manju vrijednost parametra DT segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju granu. Za vrijednost parametra DT = 0.03 RANSAC metoda ne pronalazi nijednu točku. Na slici (4.7) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.001 sa 1476 točaka koje predstavljaju granu. Na slici (4.8) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.02 sa 13676 točaka koje predstavljaju granu.



Slika 4.9. Promjena parametra DT za RL2 = 0.04

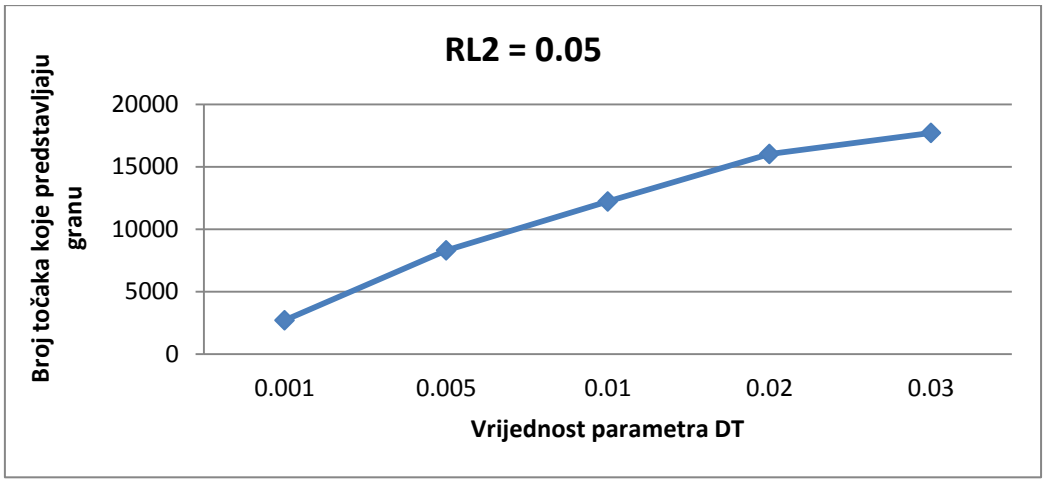


Slika 4.10. DT = 0.001

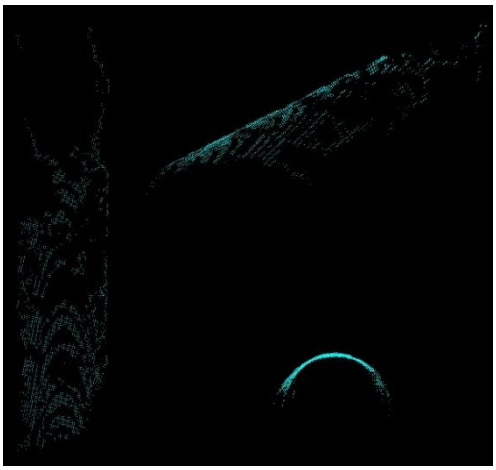


Slika 4.11. DT = 0.03

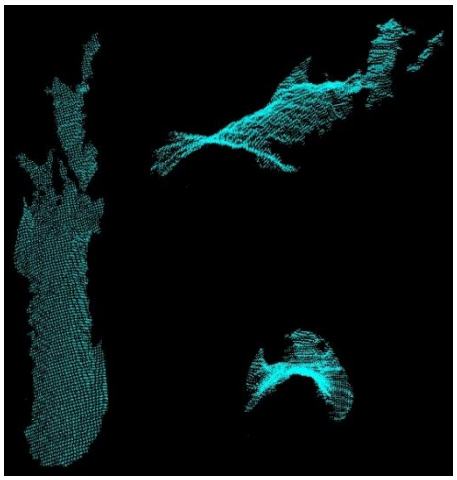
Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.04, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.9). Iz grafa se zaključuje da za manju vrijednost parametra DT segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju granu. Na slici (4.10) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.001 sa 1826 točaka koje predstavljaju granu. Na slici (4.11) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.03 sa 21153 točaka koje predstavljaju granu.



Slika 4.12. Promjena parametra DT za RL2 = 0.05

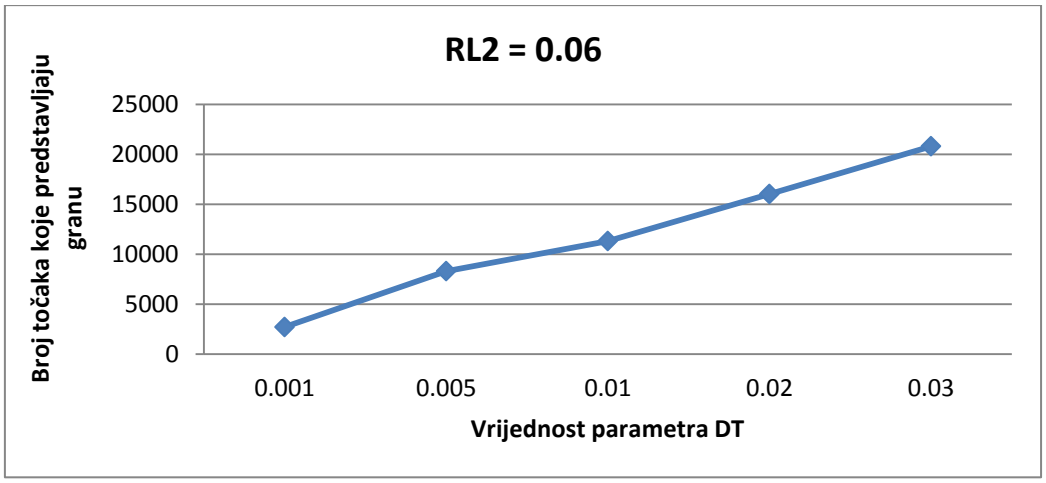


Slika 4.13. DT = 0.001



Slika 4.14. DT = 0.03

Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.05, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.12). Iz grafa se zaključuje da za manju vrijednost parametra DT segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju granu. Na slici (4.13) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.001 sa 2699 točaka koje predstavljaju granu. Na slici (4.14) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.03 sa 17715 točaka koje predstavljaju granu.



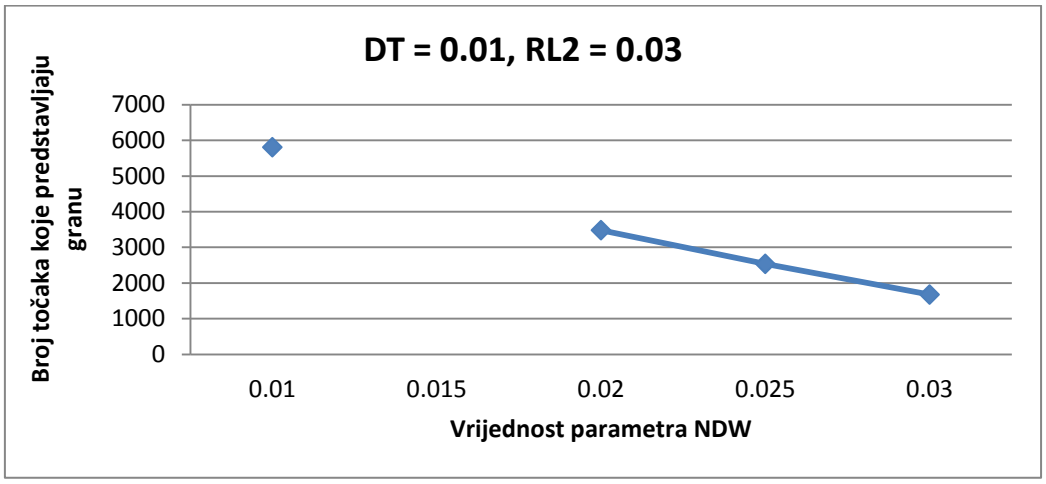
Slika 4.15. Promjena parametra DT za RL2 = 0.06

Slika 4.16. DT = 0.001

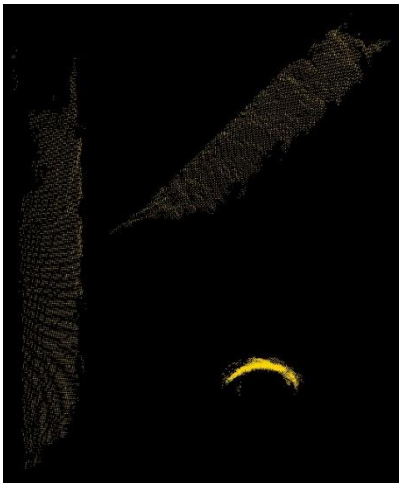
Slika 4.17. DT = 0.03

Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.06, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.15). Iz grafa se zaključuje da za manju vrijednost parametra DT segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju granu. Na slici (4.16) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.001 sa 2699 točaka koje predstavljaju granu. Na slici (4.17) prikazana je najdominantnija grana na sceni za vrijednost parametra DT = 0.03 sa 20813 točaka koje predstavljaju granu.

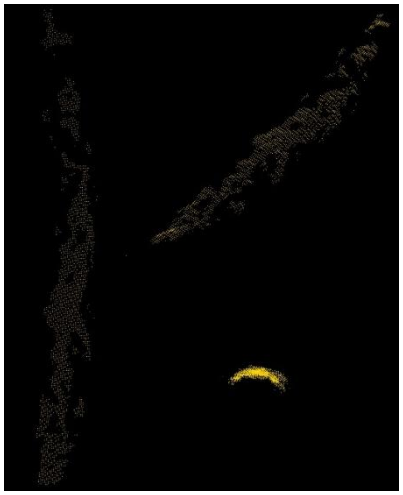
4.2. Segmentacija jedne grane uz različite vrijednosti parametra NDW



Slika 4.18. Promjena parametra NDW za DT = 0.01, RL2 = 0.03

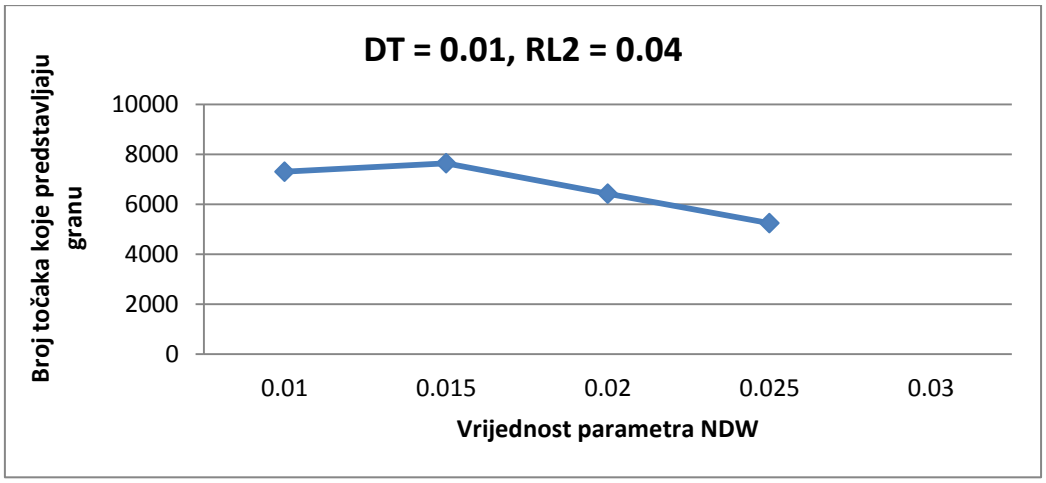


Slika 4.19. NDW = 0.01

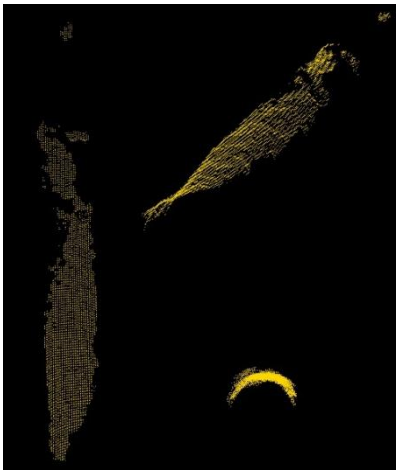


Slika 4.20. NDW = 0.03

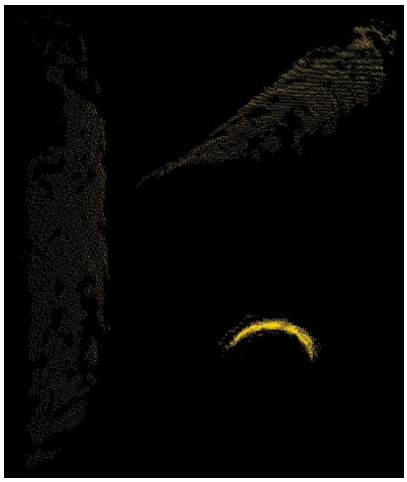
Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.01, RL2 = 0,03. Rezultati testiranja prikazani su na slici (4.18). Iz grafa se zaključuje da postavljanjem manje vrijednosti parametra NDW, segmentacija RANSAC metodom pronalazi više točaka koje predstavljaju granu. Za vrijednost NDW = 0.015, segmentacija RANSAC metodom ne pronalazi točke koje predstavljaju granu. Na slici (4.19) prikazana je najdominantnija grana na sceni za vrijednost parametra NDW = 0.01 sa 5810 točaka koje predstavljaju granu. Na slici (4.17) prikazana je najdominantnija grana na sceni za vrijednost parametra NDW = 0.03 sa 1677 točaka koje predstavljaju granu.



Slika 4.21. Promjena parametra NDW za DT = 0.01, RL2 = 0.04

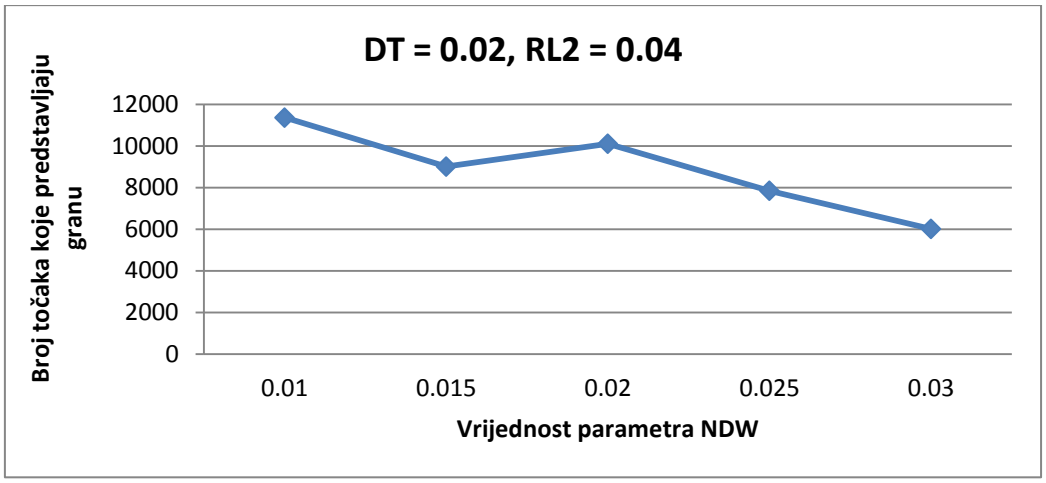


Slika 4.22. NDW = 0.01

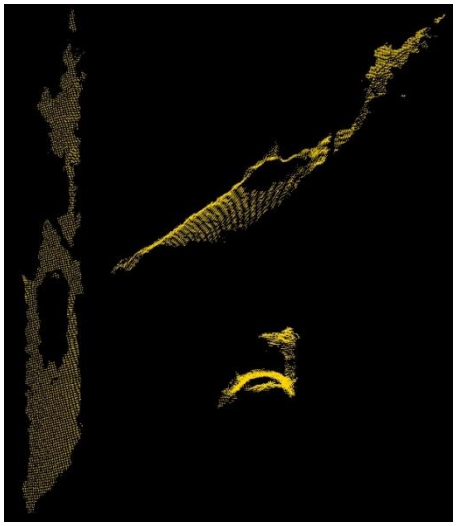


Slika 4.23. NDW = 0.025

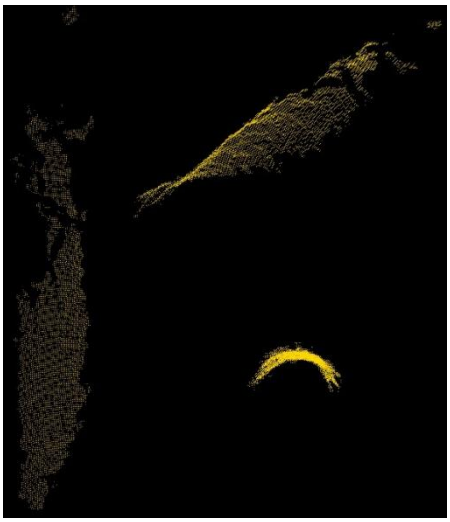
Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.01, RL2 = 0,04. Rezultati testiranja prikazani su na slici (4.21). Iz grafa se zaključuje da postavljanjem manje vrijednosti parametra NDW, segmentacija RANSAC metodom pronalazi više točaka koje predstavljaju granu. Za vrijednost NDW = 0.03, segmentacija RANSAC metodom ne pronalazi točke koje predstavljaju granu. Na slici (4.22) prikazana je najdominantnija grana na sceni za vrijednost parametra NDW = 0.01 sa 7302 točaka koje predstavljaju granu. Na slici (4.23) prikazana je najdominantnija grana na sceni za vrijednost parametra NDW = 0.025 sa 5242 točaka koje predstavljaju granu.



Slika 4.24. Promjena parametra NDW za DT = 0.02, RL2 = 0.04

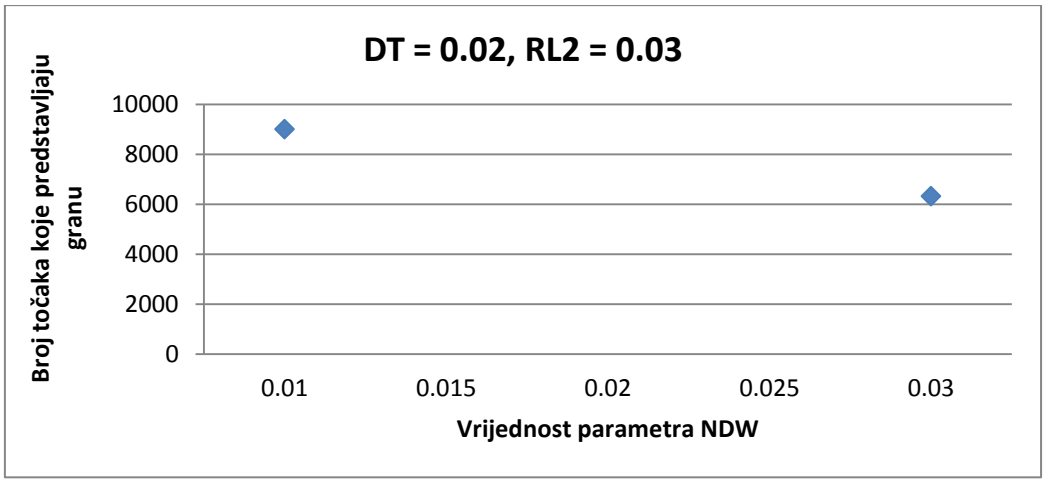


Slika 4.25. NDW = 0.01

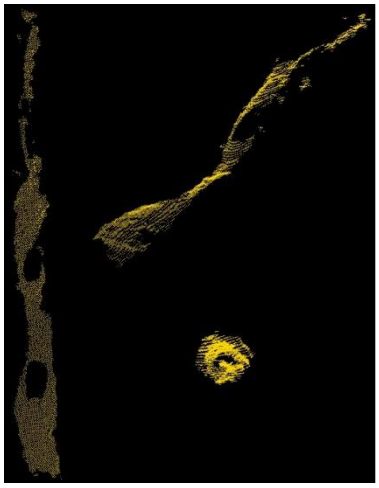


Slika 4.26. NDW = 0.03

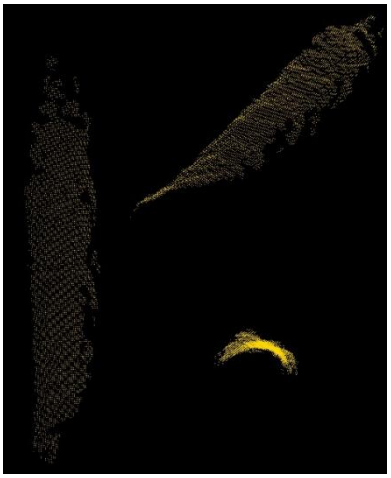
Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.02, RL2 = 0,04. Rezultati testiranja prikazani su na slici (4.24). Iz grafa se zaključuje da postavljanjem manje vrijednosti parametra NDW, segmentacija RANSAC metodom pronalazi više točaka koje predstavljaju granu. Na slici (4.24) prikazana je najdominantnija grana na sceni za vrijednost parametra NDW = 0.01 sa 11361 točaka koje predstavljaju granu. Na slici (4.26) prikazana je najdominantnija grana na sceni za vrijednost parametra NDW = 0.03 sa 6013 točaka koje predstavljaju granu.



Slika 4.27. Promjena parametra NDW za DT = 0.02, RL2 = 0.03



Slika 4.28. NDW = 0.01

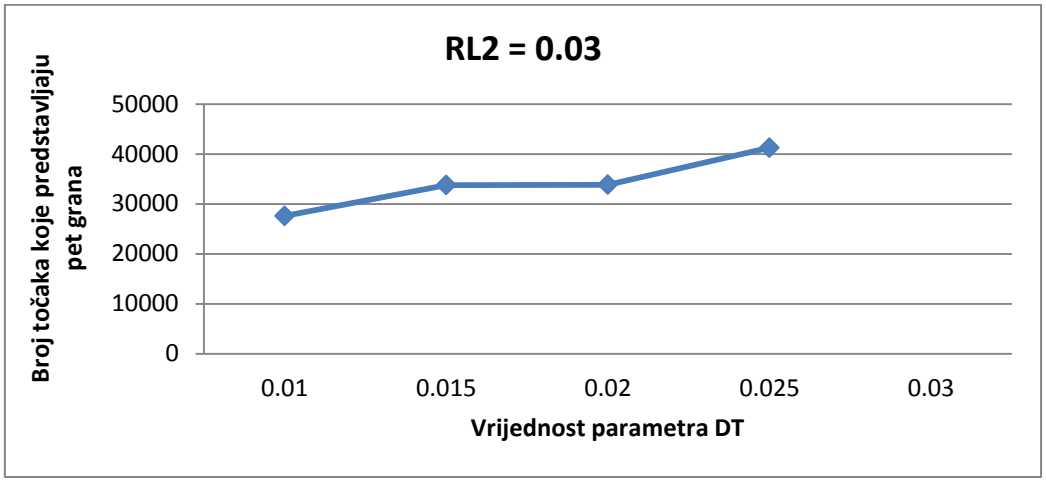


Slika 4.29. NDW = 0.03

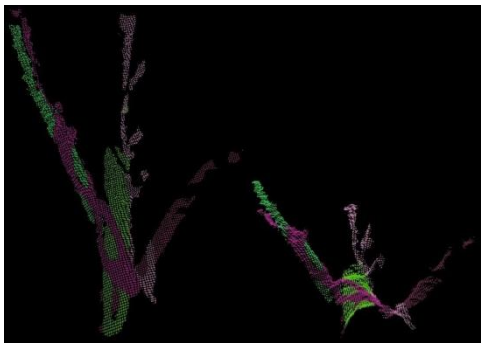
Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.02, RL2 = 0,03. Rezultati testiranja prikazani su na slici (4.27). Iz grafa se zaključuje da postavljanjem manje vrijednosti parametra NDW, segmentacija RANSAC metodom pronalazi više točaka koje predstavljaju granu. Za vrijednost NDW = 0.015, 0.02 i 0.025 segmentacija RANSAC metodom ne pronalazi točke koje predstavljaju granu. Na slici (4.28) prikazana je najdominantnija grana na sceni za vrijednost parametra NDW = 0.01 sa 9007 točaka koje predstavljaju granu. Na slici (4.29) prikazana je najdominantnija grana na sceni za vrijednost parametra NDW = 0.03 sa 6322 točaka koje predstavljaju granu.

4.3. Segmentacija pet grana uz različite vrijednosti parametra DT

U ovom testiranju vrši se segmentacija pet najdominantnijih grana na sceni.



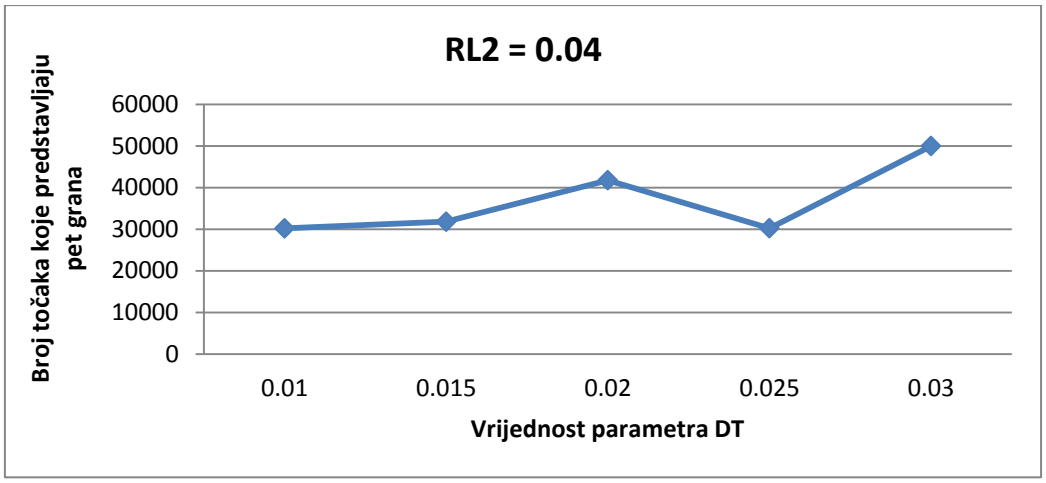
Slika 4.30. Promjena parametra DT za RL2 = 0.03



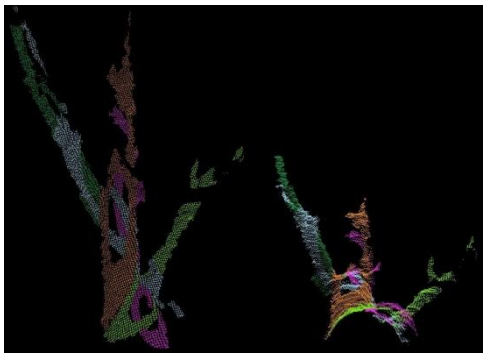
Slika 4.31. DT = 0.01

Slika 4.32. DT = 0.025

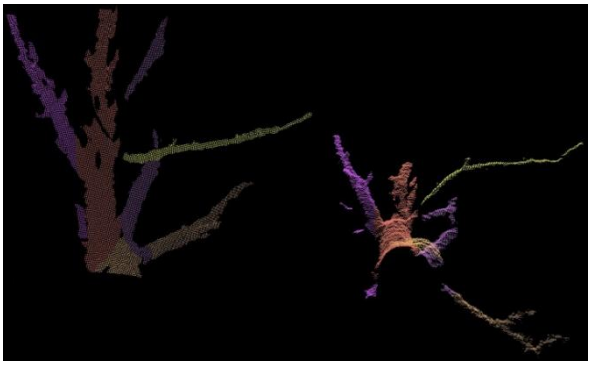
Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.03, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.30). Iz grafa se zaključuje da za manju vrijednost parametra DT segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju pet grana. Za vrijednost parametra DT = 0.03 RANSAC metoda ne pronalazi nijednu točku. Na slici (4.31) prikazano je pet najdominantnijih grana na sceni za vrijednost parametra DT = 0.01 sa 27596 točaka koje predstavljaju grane. Na slici (4.32) prikazano je pet najdominantnijih grana na sceni za vrijednost parametra DT = 0.025 sa 41261 točaka koje predstavljaju grane.



Slika 4.33. Promjena parametra DT za RL2 = 0.04

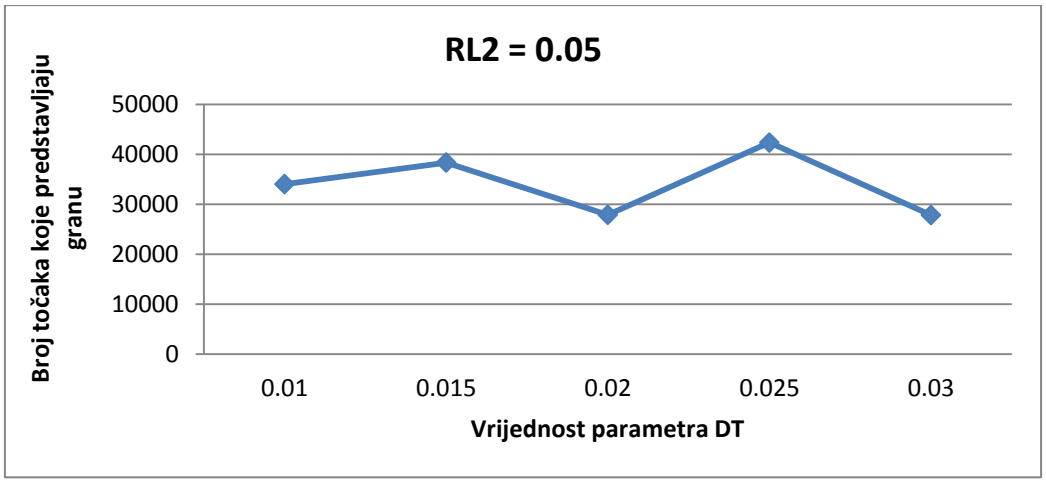


Slika 4.34. DT = 0.01

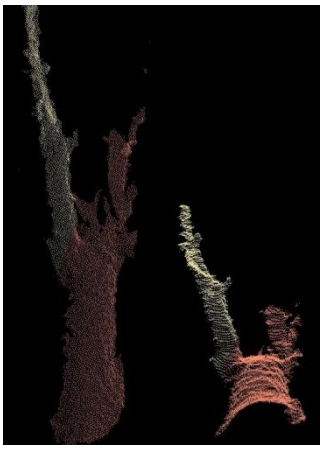


Slika 4.35. DT = 0.03

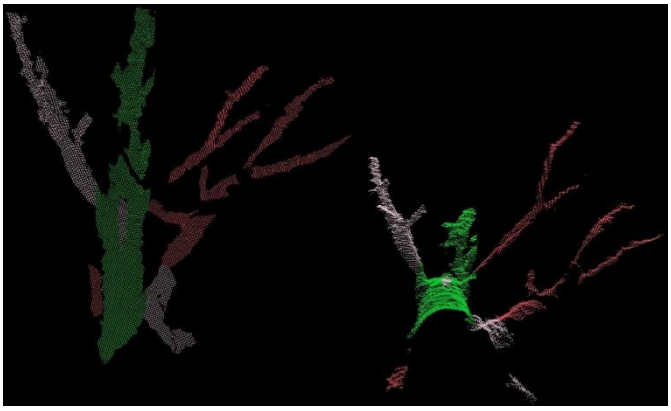
Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.04, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.33). Iz grafa se zaključuje da za manju vrijednost parametra DT segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju pet grana. Za vrijednost DT = 0.025 broj točaka koje predstavljaju pet grana je približan kao i za vrijednost DT = 0.01. Na slici (4.34) prikazano je pet najdominantnijih grana na sceni za vrijednost parametra DT = 0.01 sa 30199 točaka koje predstavljaju grane. Na slici (4.35) prikazano je pet najdominantnijih grana na sceni za vrijednost parametra DT = 0.03 sa 50000 točaka koje predstavljaju grane.



Slika 4.36. Promjena parametra DT za RL2 = 0.05



Slika 4.37. DT = 0.03

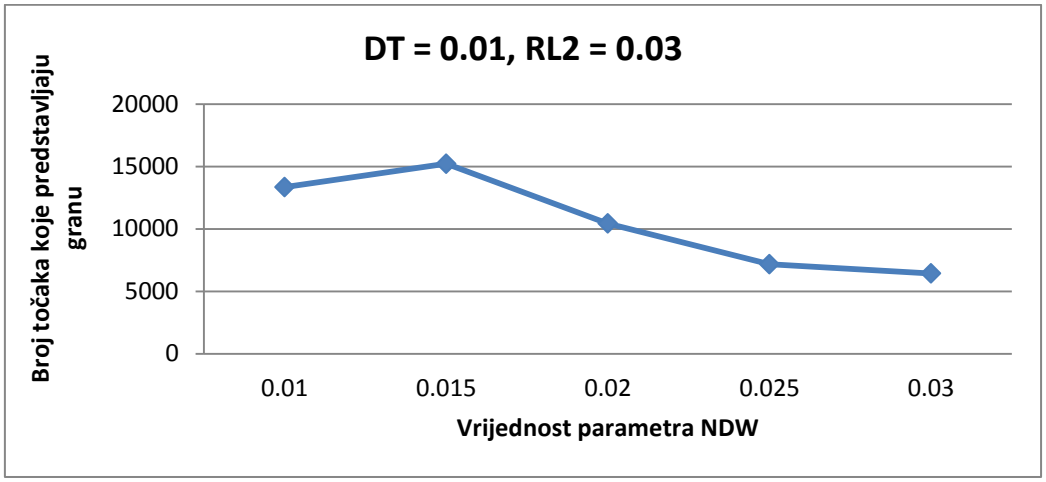


Slika 4.38. DT = 0.025

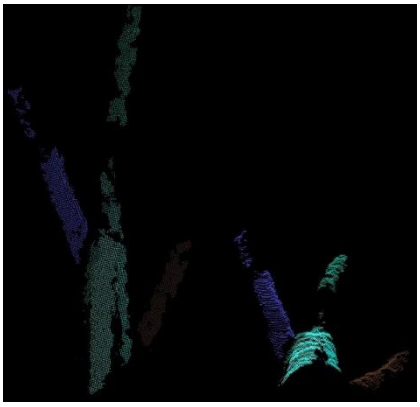
Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.05, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.36). Iz grafa se mogu vidjeti različiti rezultati povećanjem parametra DT. Za vrijednost parametra DT = 0.03 RANSAC metoda pronalazi najmanje točaka koje predstavljaju granu, njih 27819 i prikazani su na slici (4.37) . Na slici (4.38) prikazano je pet najdominantnijih grana na sceni za vrijednost parametra DT = 0.025 sa 42344 točaka koje predstavljaju grane.

4.4. Segmentacija pet grana uz različite vrijednosti parametra NDW

U ovom poglavlju mijenjat će se parametar NDW za različite kombinacije parametra DT i RL2.



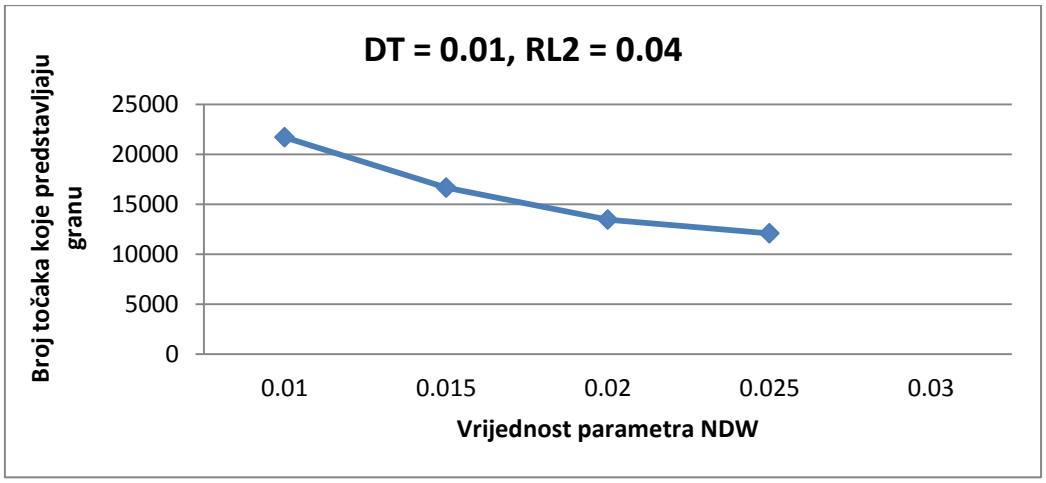
Slika 4.39. Promjena parametra NDW za DT = 0.01, RL2 = 0.03



Slika 4.40. NDW = 0.015

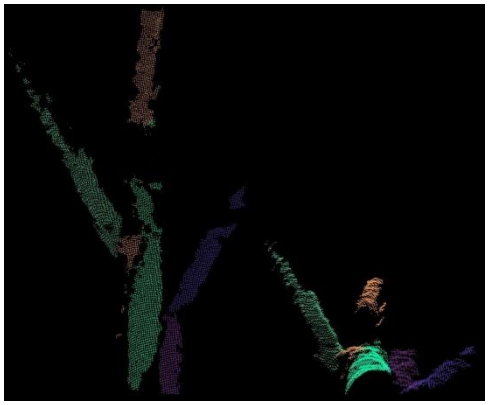
Slika 4.41. NDW = 0.03

Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.01, RL2 = 0,03. Rezultati testiranja prikazani su na slici (4.39). Iz grafa se zaključuje da postavljanjem većih vrijednosti parametra NDW, segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju granu. Na slici (4.40) prikazana su najdominantnije grane na sceni za vrijednost parametra NDW = 0.015 sa 15217 točaka koje predstavljaju grane. Na slici (4.41) prikazani su rezultati za vrijednost parametra NDW = 0.03 kada je pronađeno najmanje točaka, njih 6432 koje predstavljaju grane.



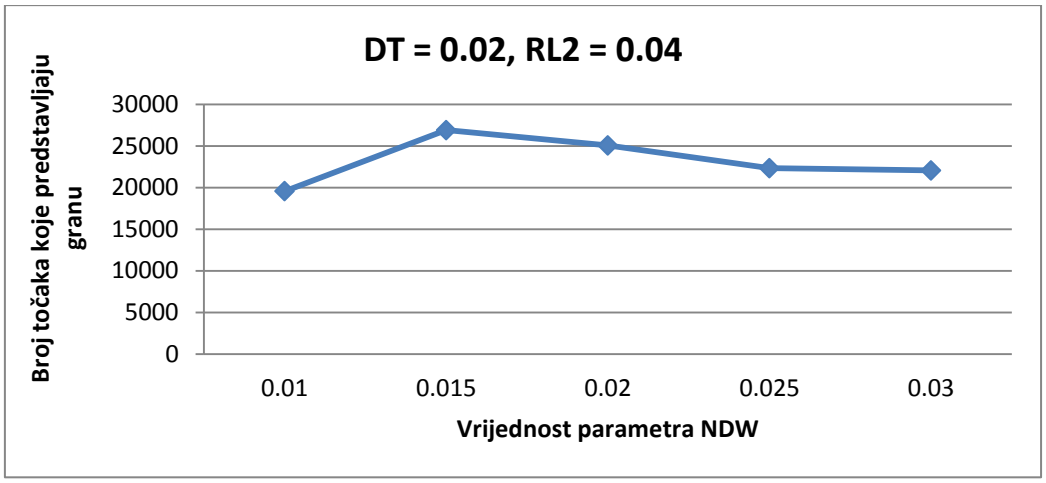
Slika 4.42. Promjena parametra NDW za DT = 0.01, RL2 = 0.04

Slika 4.43. NDW = 0.01

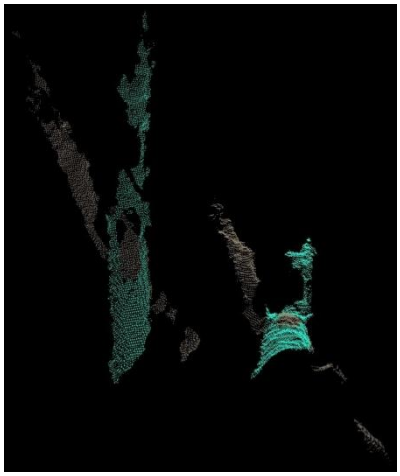


Slika 4.44. NDW = 0.025

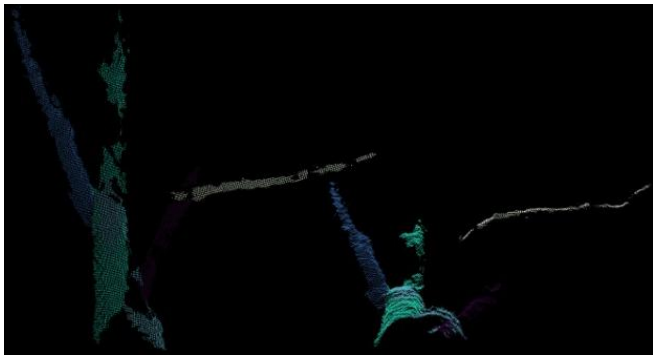
Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.01, RL2 = 0,04. Rezultati testiranja prikazani su na slici (4.42). Iz grafa se zaključuje da postavljanjem većih vrijednosti parametra NDW, segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju grane. Za vrijednost parametra NDW = 0.03 RANSAC metoda ne pronalazi nijednu točku. Na slici (4.43) prikazane su najdominantnije grane na sceni za vrijednost parametra NDW = 0.01 sa 21711 točaka koje predstavljaju grane. Na slici (4.29) prikazane su najdominantnije grane na sceni za vrijednost parametra NDW = 0.025 sa 12087 točaka koje predstavljaju grane.



Slika 4.45. Promjena parametra NDW za DT = 0.02, RL2 = 0.04

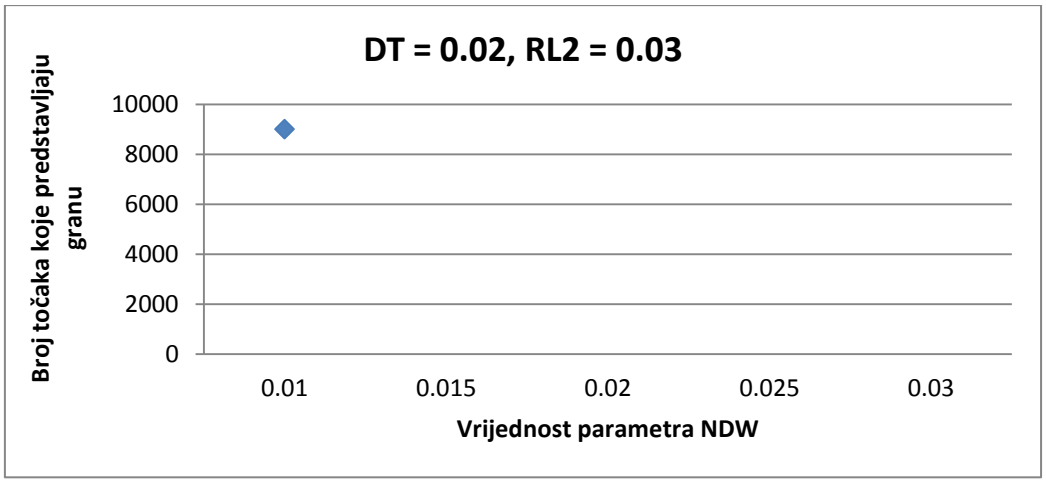


Slika 4.46. NDW = 0.01



Slika 4.47. NDW = 0.015

Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.02, RL2 = 0,04. Rezultati testiranja prikazani su na slici (4.45). Na slici (4.46) prikazane su najdominantnije grane na sceni za vrijednost parametra NDW = 0.01 gdje se RANSAC metodom pronalazi najmanje točaka koje predstavljaju grane, njih 19574. Na slici (4.47) prikazane su najdominantnije grane na sceni za vrijednost parametra NDW = 0.015 kada se RANSAC metodom pronalazi najviše točaka koje predstavljaju grane, njih 26904.



Slika 4.48. Promjena parametra NDW za DT = 0.02, RL2 = 0.03

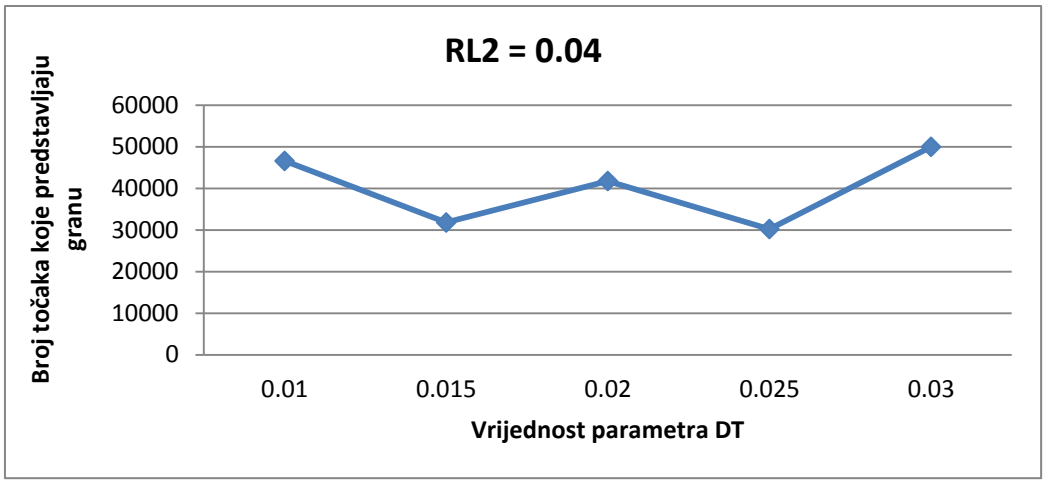
Slika 4.49. NDW = 0.01

Rezultati testiranja prikazani su slikom (4.48).

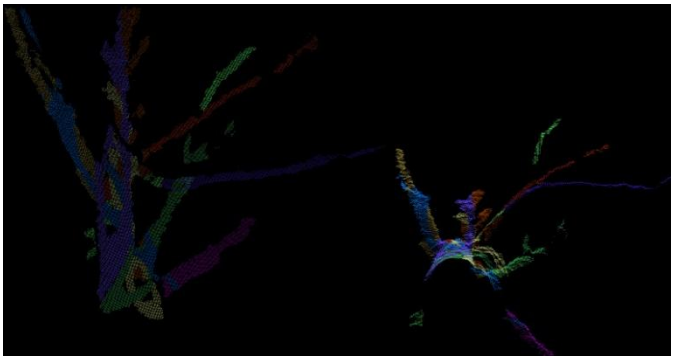
Za vrijednost parametra NDW = 0.01 RANSAC metodom pronalazi se 9007 točaka koje predstavljaju granu, prikazani su slikom (4.49). Za povećanje parametra NDW RANSAC metoda ne pronalazi nijednu točku.

4.5. Segmentacija deset grana uz različite vrijednosti parametra DT

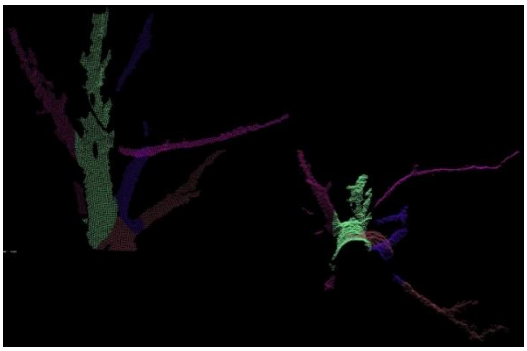
U ovom poglavlju vrši se segmentacija deset dominantnih grana na sceni, promjenom vrijednosti parametra DT.



Slika 4.50. Promjena parametra DT za RL2 = 0.04



Slika 4.51. DT = 0.01

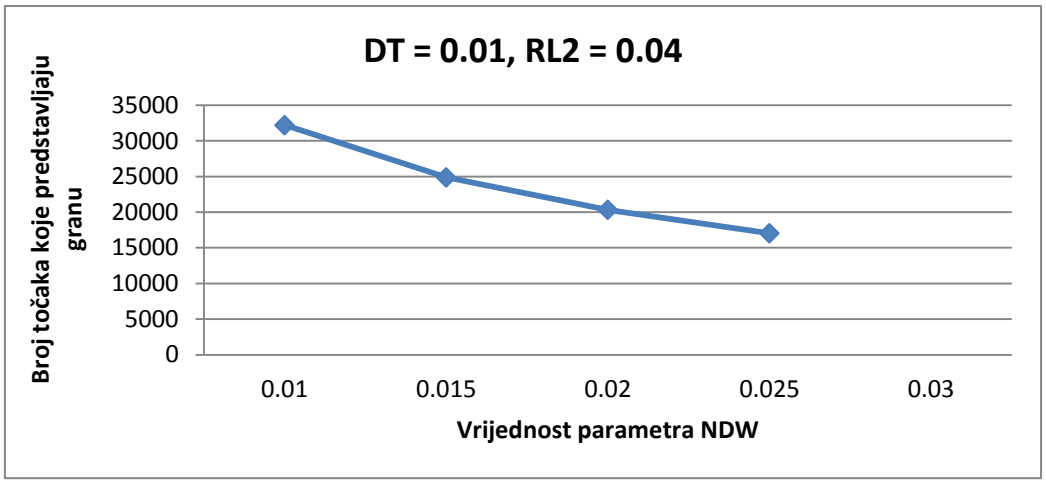


Slika 4.52. DT = 0.03

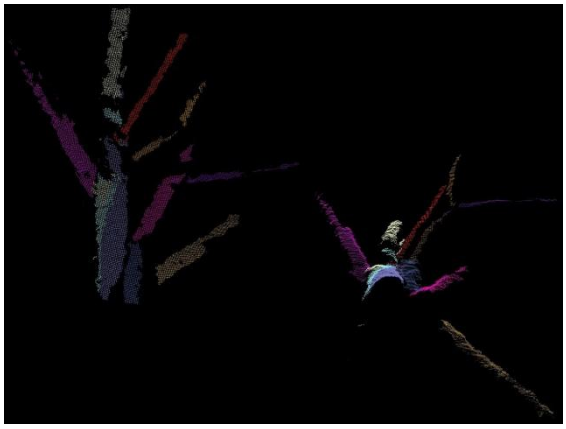
Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.04, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.50), iz koje se može vidjeti da se najmanji broj točaka, njih 30263, koje predstavljaju grane dobije za vrijednost parametra DT = 0.025.. Na slici (4.51) prikazano je deset najdominantnijih grana na sceni za vrijednost parametra DT = 0.01 sa 46603 točaka koje predstavljaju grane. Na slici (4.52) prikazano je deset najdominantnijih grana na sceni za vrijednost parametra DT = 0.03 sa 50000 točaka koje predstavljaju grane.

4.6. Segmentacija deset grana uz različite vrijednosti parametra NDW

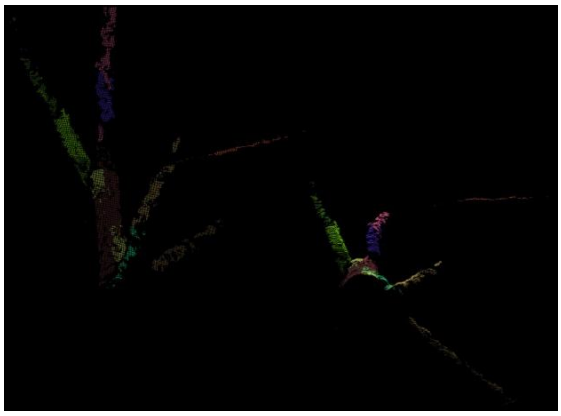
U ovom poglavlju vrši se segmentacija deset dominantnih grana na sceni, promjenom vrijednosti parametra NDW.



Slika 4.53. Promjena parametra NDW za DT = 0.01, RL2 = 0.04



Slika 4.54. NDW = 0.01

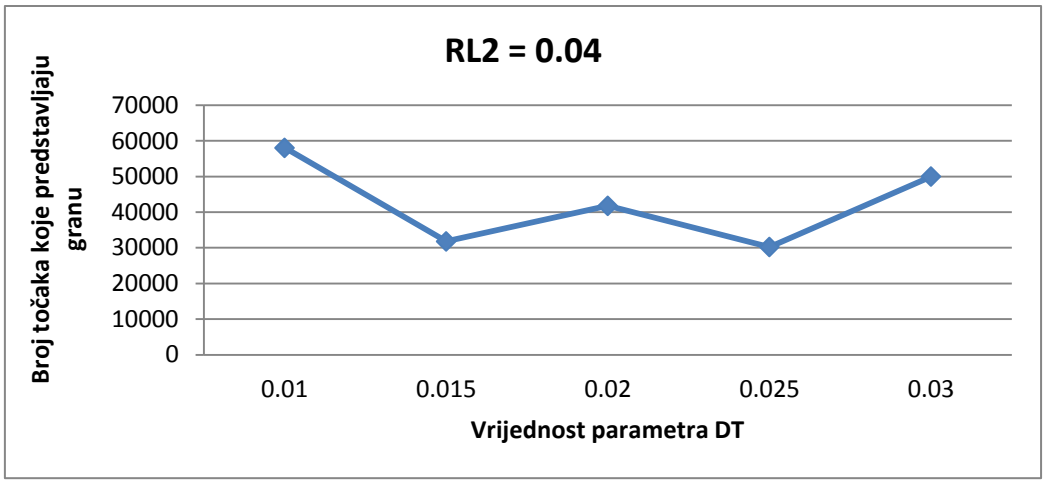


Slika 4.55. NDW = 0.025

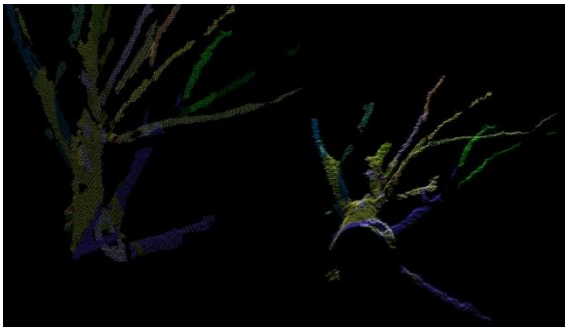
Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.01, RL2 = 0,04. Rezultati testiranja prikazani su na slici (4.53). Za vrijednost parametra NDW = 0.03 segmentacija RANSAC metodom ne pronalazi nijednu točku. Na slici (4.54) prikazane su najdominantnije grane na sceni za vrijednost parametra NDW = 0.01 gdje se RANSAC metodom pronalazi najviše točaka koje predstavljaju grane, njih 32195. Na slici (4.47) prikazane su najdominantnije grane na sceni za vrijednost parametra NDW = 0.015 kada se RANSAC metodom pronalazi najmanje točaka koje predstavljaju grane njih 17039. Povećanjem vrijednosti parametra NDW segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju grane.

4.7. Segmentacija trideset grana uz različite vrijednosti parametra DT

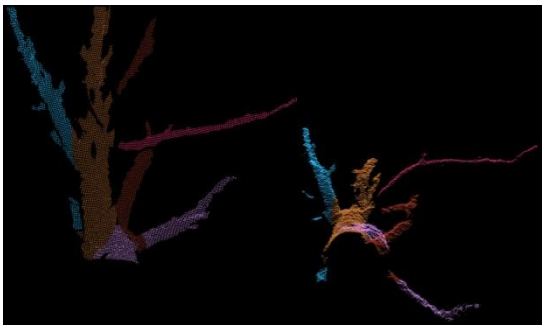
U ovom poglavlju vrši se segmentacija trideset dominantnih grana na sceni, promjenom vrijednosti parametra DT.



Slika 4.56. Promjena parametra DT za RL2 = 0.04



Slika 4.57. DT = 0.01

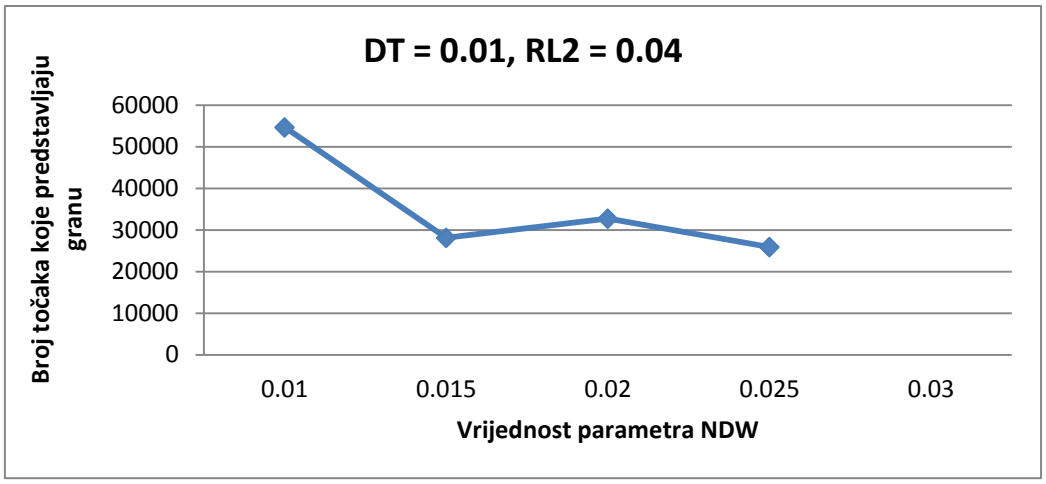


Slika 4.58. DT = 0.03

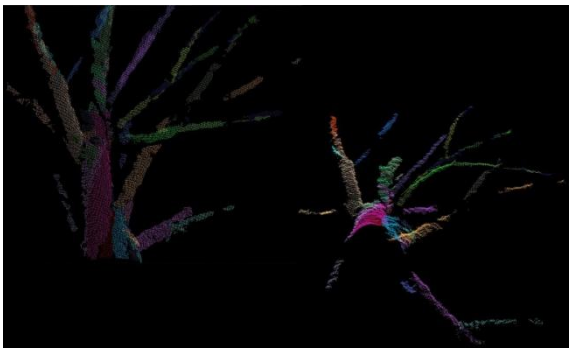
Vrijednost parametra DT se mijenja za konstantnu vrijednost parametra RL2 = 0.04, ostale vrijednosti parametara se ne mijenjaju. Rezultati testiranja prikazani su na slici (4.56), iz koje se može vidjeti da se najmanji broj točaka, njih 30236, koje predstavljaju grane dobije za vrijednost parametra DT = 0.025. Na slici (4.57) prikazano je trideset najdominantnijih grana na sceni za vrijednost parametra DT = 0.01 sa 58033 točaka koje predstavljaju grane. Na slici (4.58) prikazano je trideset najdominantnijih grana na sceni za vrijednost parametra DT = 0.03 sa 50000 točaka koje predstavljaju grane.

4.8. Segmentacija trideset grana uz različite vrijednosti parametra NDW

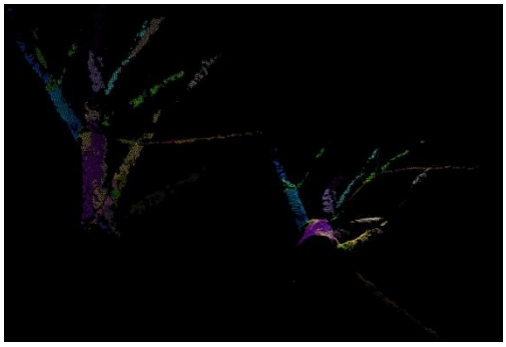
U ovom poglavlju vrši se segmentacija trideset dominantnih grana na sceni, promjenom vrijednosti parametra NDW.



Slika 4.59. Promjena parametra NDW za DT = 0.01, RL2 = 0.04



Slika 4.60. NDW = 0.01



Slika 4.61. NDW = 0.025

Vrijednost parametra NDW se mijenja za konstantne vrijednosti parametra DT = 0.01, RL2 = 0,04. Rezultati testiranja prikazani su na slici (4.59). Za vrijednost parametra NDW = 0.03 segmentacija RANSAC metodom ne pronalazi nijednu točku. Na slici (4.60) prikazane su najdominantnije grane na sceni za vrijednost parametra NDW = 0.01 gdje se RANSAC metodom pronalazi najviše točaka koje predstavljaju grane, njih 54649. Na slici (4.61) prikazane su najdominantnije grane na sceni za vrijednost parametra NDW = 0.025 kada se RANSAC metodom pronalazi najmanje točaka koje predstavljaju grane njih 25886. Povećanjem vrijednosti parametra NDW segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju grane.

5. Zaključak

U ovom radu uspješno je razvijen i testiran algoritam za segmentaciju valjkastih objekata RANSAC metodom. Za razumijevanje osnova dan je teorijski uvod za RGB-D kameru i za RANSAC metodu koja je korištena u radu. Rad je realiziran pomoću programskog jezika C++ u Miscrosoft Visual Studiu 2012 Express for Desktop integriranom razvojnom okruženju uz korištenje PCL biblioteke i Asus Xtion ProLive kamere.

U radu je izvršeno dvadeset testova za različite kombinacije parametara RANSAC metode.

Povećanjem broja grana zadanih za segmentaciju povećavalo se i vrijeme testiranja, tako da za segmentaciju trideset grana promjenom parametra NDW vrijeme izvršavanja algoritma iznosilo je više od trideset minuta za jednu vrijednost parametra. Testiranje je prikazano grafovima za određene promjene parametara i slikama kada je segmentacijom pronađeno najmanje i najviše točaka koje predstavljaju grane.

Iz provedenih ispitivanja može se zaključiti da u većini slučajeva povećanjem vrijednosti parametra DT segmentacija RANSAC metodom pronalazi više točaka koje predstavljaju granu, dok za povećanje parametra NDW segmentacija RANSAC metodom pronalazi manje točaka koje predstavljaju granu. Pri postavljanju parametara potrebno je postaviti tip modela segmentacije. Za segmentiranje valjkastih objekata koristi se 'sacmodel_cylinder'. Model sadrži sedam koeficijenata valjka koji je određen točkom na svojoj osi(x,y,z), u smjeru osi (x,y,z) i radijusu.

Razvojni program se sastoji od dva projekta, prvi imena kinect.cpp snima sliku i sprema je na računalo. Drugi imena roller.cpp učitava sliku sa računala i vrši segmentaciju valjkastih objekata RANSAC metodom na toj slici. Jedan od prijedloga za poboljšanje razvojnog programa je povezati prvi i drugi projekt u jednu cjelinu, tj. u jedan projekt kako bi sve obuhvaćeno bilo na jednom mjestu. Također bilo bi dobro poboljšati vizualizaciju korištenjem PCL biblioteke, kako se snimljene slike nebi otvarale korištenjem cmd.exe aplikacije.

6. Literatura

[1] https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/specifications

[2] https://en.wikipedia.org/wiki/Random_sample_consensus

[3] http://docs.pointclouds.org/1.0.1/classpcl_1_1_sample_consensus_model_cylinder

[4] Rusu, R. B. (2009). Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. PhD thesis, Technischen Universitat Munchen, Munchen.

[5] Martin, M. M., Geir , G. O. T. (2015). Robotic Assembly Using a 3D Camera. Project / Master Thesis, Department of Production and Quality Engineering Norwegian University of Science and Technology

[6] http://math.etsu.edu/multicalc/prealpha/Chap3/Chap3-6/printversion.pdf

7. Sažetak

U ovom diplomskom radu razvijen je program koji izrađuje 3D model drveta na temelju slike snimljene 3D kamerom. Pomoću programskog jezika C++ u razvojnom okruženju Microsoft Visual Studio 2012 i korištenjem programske biblioteke Point Cloud Library (PCL) napravljena su dva algoritma, jedan učitava sliku iz datoteke i segmentira ju na valjkaste površine RANSAC metodom, te drugi koji prikazuje i sprema sliku RGB-D kamere u datoteku. Segmentacijom na valjkaste površine želi se postići prepoznavanje najdominantnijih valjkastih objekata na sceni. Razvijeni program je ispitan na RGB-D slici stabla za različite kombinacije parametara RANSAC algoritma.

Ključne riječi: RGB-D kamera, PCL, RANSAC, segmentacija slike, 3D modeliranje

Abstract

In this graduate thesis, a program has been developed that creates a 3D model of a tree based on images captured with a 3D camera. Two algorithm's were created using the C++ programming language in the development environment of Microsoft Visual Studio 2012 and using the programming library Point Cloud Library (PCL). One algorithm loads an image from a file and segmentes it into cylindrical surfaces using the RANSAC approach. The other algorithm creates dispalys and saves images acquired by a RGB-D camera to a file. The developed program was tested on an RGB-D image of a tree for different combinations of the applied RANSAC algorithm parameters.

Key words: RGB-D camera, PCL, RANSAC, image segmentation, 3D modeling

Životopis

Josip Rotim, rođen 28.06.1990. u Osijeku, Republika Hrvatska. Školovanje započeo 1997. godine u osnovnoj školi Fran Krsto Frankopan u Osijeku. Godine 2005. upisao Strojarsko tehničku školu u Osijeku, smjer Računalni tehničar za strojarstvo. Godine 2009. upisao sveučilišni preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku. Godine 2013. upisuje sveučilišni diplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku, smjer Procesno računarstvo.

Prilog

CD koji sadrži:

- Dokumentacija diplomskog rada u .docx i .pdf verziji
- C++ projekti:
    - kinect.cpp
    - roller.cpp
- Slike stabala u .pcd verziji
- Slike rezultata pokusa